An algorithm for ordinal sorting based on ELECTRE with categories defined by examples

Clara Rocha · Luis C. Dias

Abstract This work proposes a Progressive Assisted Sorting Algorithm (PASA) based on a multicriteria evaluation ELECTRE-type method. The purpose of the PASA is to aid a decision maker to progressively sort a set of alternatives into a set of categories, which we considered are ordered (ordinal sorting), following a consistency principle. We consider the principle that if an alternative outranks (is as good as) a second one, then it must belong to the same category or to a better category. The set of alternatives already sorted by the decision maker will implicitly define the categories, and will constrain the range of categories where other alternatives may be sorted. We show how the same idea may be used in an aggregation/disaggregation approach, considering some parameters of ELECTRE are not fixed a priori, but are constrained only by the examples provided. In this context, we establish a "convex-shape property" stating that the range of possible categories for an alternative is always an interval of categories. A discussion contrasting this approach with ELECTRE TRI is included in the conclusions.

Keywords Multi-criteria decision aiding · Sorting problem · ELECTRE · Aggregation/disaggregation approaches

1 Introduction

Sorting problems are concerned with evaluating a set A of alternatives in order to assign them to mutually exclusive categories $C^1, C^2, \ldots, C^{n_{cat}}$. In multicriteria analysis, the assignment

C. Rocha
Escola Superior de Tecnologia da Saúde de Coimbra, Instituto Politécnico de Coimbra, Rua 5 de Outubro,
S. Martinho do Bispo, Ap. 7006, Coimbra 3040-162, Portugal

C. Rocha
INESC Coimbra, R. Antero de Quental 199, Coimbra 3000-033, Portugal

L. C. Dias (✉)
INESC Coimbra and Faculdade de Economia, Universidade de Coimbra, Av. Dias da Silva 165,
Coimbra 3004-512, Portugal
e-mail: lmcdias@fe.uc.pt

of an alternative a_i to a category results from its intrinsic evaluation on different criteria, as well as from the definition of the categories, sometimes also provided in terms of multiple criteria. Among the methods that have been proposed to handle multiple criteria sorting problems we can cite, e.g., Trichotomic Segmentation [12], N-TOMIC [11], ORCLASS [10], ELECTRE TRI [19], PROAFTN [1], UTADIS [21], a general class of filtering methods [17], rough sets [6] or the Koksalan-Ulu method [9].

In some cases, the categories are not ordered (nominal sorting or classification). For instance, one may sort the members of a database of persons offering themselves to work in a company into the categories "technical profile", "commercial profile", "leadership profile", etc., without stating one category is the best or the most important. In other cases (ordinal sorting), the categories are ordered. For instance, one may sort the same job candidates into the categories "low potential", "average potential", "high potential", etc., where the candidates placed in higher categories are supposedly better than the ones placed in lower categories. When the categories are not ordered, they are usually defined through prototypical elements (e.g., [1]), and the elements of A are sorted according to their similarity with such prototypes. When the categories are ordered, they can also be defined through prototypical elements, but typically they are defined through category limits, i.e., lower and upper bounds, where usually the upper bound of a category is the lower bound of the next better category (e.g., ELECTRE TRI [19]).

We are interested in ordinal sorting problems, with the categories ordered from worst (C^1, by convention) to best ($C^{n_{cat}}$). For these problems, there have been some proposals based on aggregation/disaggregation approaches [3,8,21] to overcome the difficulties often felt by the Decision Maker (DM) when asked to provide values for the method's parameters. In some of the best-known approaches, either based on utility functions [21] or based on ELECTRE TRI [3,14,16], the DM is asked to provide prototypical elements (i.e., sorting examples), although there exist parameters defining the category limits. In UTADIS [21], the examples are used to infer all of the method's parameters, including the category limits; in ELECTRE, inferring all of the method's parameters requires solving difficult non-linear problems [14]. As simplifications, [3] assumes the category limits are given, whereas [16] addresses the difficult problem of inferring these limits from sorting examples. In all of these approaches, the examples are used to infer a sorting model, which may then be used to sort any other alternatives.

A different new direction consists in proposing ordinal sorting procedures without explicitly defining category limits [5,9]. Rather, the DM is asked to sort some alternatives from A as examples which will define the categories of the remaining alternatives. Doumpos and Zopounidis [5] consider the examples as a learning set that can be used to define a sorting model based on the net flow of PROMETHEE, solving a linear program to minimize the violations of the classification rule. Koksalan and Ulu [9] propose an interactive procedure to help the DM add successively new examples until all the alternatives are sorted. Their interactive procedure, based on the additive aggregation model for multi-attribute utility functions, ensures the examples are provided in a consistent manner, namely, imposing the natural principle that if the utility of an alternative a_i is equal to or higher than the utility of a_j, then the category of a_i must be equal to or better than the category of a_j.

The method of Koksalan and Ulu is to our knowledge a first example of what we call a PASA (Progressive Assisted Sorting Algorithm). The purpose of a PASA is to assist a DM in progressively sorting alternatives by assigning them to categories $C^1, C^2, \ldots, C^{n_{cat}}$. The DM exercises his or her judgement concerning each alternative to be sorted, with the aid of a PASA that constrains the categories where it may be placed if consistency is to be maintained concerning previous judgements; namely, the DM is aided by learning the minimum and maximum category where an alternative might be sorted, given the examples already

sorted. Sometimes, the minimum and maximum will coincide. Otherwise, it is up to the DM to make a decision. In these cases, it is the DM who can assist the procedure by choosing one among an interval of potential categories for an alternative. This decision will in turn become a precedent for future ones. The spirit behind a PASA is hence that of aiding a DM to perform a series of sorting decisions in a consistent manner, rather than deriving a model to substitute the DM.

The purpose of this paper is to make a proposal of a PASA based on the ELECTRE methodology, rather than utility functions. However, unlike utility functions, there is not an undisputed principle of consistency to be followed, namely when we note that the outranking relation S defined by ELECTRE is not complete (alternatives that do not outrank each other are called incomparable), it is not transitive (e.g., $a_i S a_j$ and $a_j S a_k$, but not $a_i S a_k$), and it may present cycles (e.g., $a_j S a_k S a_i S a_j$). These difficulties stem form the strengths of ELECTRE, namely the partial absence of compensation among criteria (a very low performance on one criterion may not be compensated by an excellent performance on another criterion) and the identification of incomparability (an alternative may not be sorted into a precise category because it is too different from any category limits or prototypes).

In this paper we put forward the suggestion (also advocated by [2]) of using the semantic meaning of the outranking relation ($a_i S a_j$ means a_i is at least as good as a_j, i.e., a_i is not worse than a_j) to define a consistency principle: "if an alternative outranks a second one, then it must belong to the same category or to a better category". Although seemingly logical and weak, this is a relatively strong requirement. For instance, it is not imposed by the ELECTRE TRI method. Let us note that the word consistency in this work refers to not contradicting the principle when outranking relations are used. Hence, lack of consistency may be due to misjudgments of the DM (i.e., judgements the DM is willing to change in retrospect), or may be due to the inadequacy of the principle, or may be due to the way the outranking relation is defined in ELECTRE.

This paper also proposes how the same procedure may be used even when not all of the method's parameters have been fixed, using an aggregation/disaggregation approach. More concretely, we will assume that the criteria weights and the cutting level of ELECTRE have not been defined and will compute, for each potential sorting decision, a vector of weights and an interval of values for the cutting level that make the decision compatible with the previous ones. Therefore, we may conclude that there is more than one possible category for a given alternative because either it is incomparable to those already sorted, or because of the accepted variability of the weights and the cutting level, or for both of these reasons at the same time. We will consider the parameters that do not interrelate the criteria (i.e., the thresholds associated to each criterion individually) as fixed.

This paper is structured as follows. The next section briefly reminds how the [0,1]-valued outranking relation used by ELECTRE TRI is defined. Section 3 presents an ELECTRE-type PASA without explicit category limits, assuming all of the method's parameters are fixed beforehand. Section 4 presents the same procedure in the context of an aggregation/disaggregation approach where the weights and the cutting level do not need to be precisely fixed. Section 5 presents two illustrative examples. Finally, in Sect. 6 we highlight the main characteristics of the proposed approach, discussing how it compares with ELECTRE TRI.

2 Valued outranking relations in ELECTRE

Let $A = \{a_1, \ldots, a_m\}$ denote a set of alternatives (actions, objects, projects) represented by a vector of evaluations on n criteria. Let $g_1(.), \ldots, g_n(.)$ denote the set of criteria functions,

such that $g_t(a_i)$ indicates the evaluation (performance) of the ith alternative according to the tth criterion. Without loss of generality, we will assume that the higher the performance value $g_t(a_i)$, the better the alternative will be.

A valued outranking relation is used by methods such as ELECTRE III [18] and ELEC-TRE TRI [19,20] when comparing one alternative against another. Given any ordered pair $(a_i, a_j) \in A^2$, one may compute a credibility degree $S(a_i, a_j)$ indicating the degree to which a_i outranks a_j. This degree may then be compared with a cutting level λ, to decide whether the outranking holds or not:

$$a_i \text{ outranks } a_j (\text{denoted } a_i S a_j) \Leftrightarrow S(a_i, a_j) \geq \lambda$$

In ELECTRE, the word outranking means "is at least as good as", or "is not worse than". When comparing $S(a_i, a_j)$, $S(a_j, a_i)$, and λ, four situations may occur:

- $a_i S a_j$ and $\neg(a_j S a_i) \Leftrightarrow a_i P a_j$ (a_i is preferable to a_j)
- $\neg(a_i S a_j)$ and $a_j S a_i \Leftrightarrow a_j P a_i$ (a_j is preferable to a_i)
- $a_i S a_j$ and $a_j S a_i \Leftrightarrow a_i I a_j$ (a_i is indifferent to a_j)
- $\neg(a_i S a_j)$ and $\neg(a_j S a_i) \Leftrightarrow a_i R a_j$ (a_i is incomparable to a_j)

The remainder of this section briefly reminds the computation of a credibility degree $S(a_i, a_j)$ for any given ordered pair $(a_i, a_j) \in A^2$. For justifications and more details see [13,19].

2.1 Computation of single-criterion concordance indices

The single-criterion concordance index $c_t(a_i, a_j)$ indicates the degree to which the tth criterion ($t = 1, \ldots, n$) agrees with the conclusion that a_i outranks a_j. This index is computed taking into account the difference of performances on the criterion considered, as well as two thresholds: indifference q_t and preference p_t ($0 \leq q_t \leq p_t$):

$$c_t(a_i, a_j) = \begin{cases} 0 & \text{if } g_t(a_j) - g_t(a_i) \geq p_t \\ \frac{p_t - g_t(a_j) + g_t(a_i)}{p_t - q_t} & \text{if } q_t < g_t(a_j) - g_t(a_i) < p_t \\ 1 & \text{if } g_t(a_j) - g_t(a_i) \leq q_t \end{cases} \tag{1}$$

2.2 Computation of the global concordance index

A global concordance index $c(a_i, a_j)$ is computed by aggregating the n single-criterion concordance indices obtained before. It represents the level of majority among the criteria in favor of the conclusion that a_i outranks a_j. The computation $c(a_i, a_j)$ takes into account a vector of criteria weights. Each of these weights k_t ($t = 1, \ldots, n$) can be interpreted as the voting power of the respecting criterion. $c(a_i, a_j)$ can be written as follows:

$$c(a_i, a_j) = \frac{\sum_{t=1}^{n} k_t c_t(a_i, a_j)}{\sum_{t=1}^{n} k_t}$$

Usually, the weights are normalized such that $\sum_{t=1}^{n} k_t = 1$, therefore allowing to write:

$$c(a_i, a_j) = \sum_{t=1}^{n} k_t \cdot c_t(a_i, a_j) \tag{2}$$

2.3 Computation of single-criterion discordance indices

The single-criterion discordance index $d_t(a_i, a_j)$ indicates the degree to which the tth criterion ($t = 1, \ldots, n$) disagrees with the conclusion that a_i outranks a_j. This index is computed taking into account the difference of performances on the criterion considered, as well as two thresholds: discordance u_t and veto v_t ($p_t \leq u_t \leq v_t$) [13]:

$$d_t(a_i, a_j) = \begin{cases} 1 & \text{if } g_t(a_j) - g_t(a_i) \geq v_t \\ \frac{g_t(a_j) - g_t(a_i) - u_t}{v_t - u_t} & \text{if } u_t < g_t(a_j) - g_t(a_i) < v_t \\ 0 & \text{if } g_t(a_j) - g_t(a_i) \leq u_t \end{cases} \tag{3}$$

2.4 Computation of the credibility degree

The computed global concordance index and single-criterion discordance indices are aggregated into a credibility degree $S(a_i, a_j)$ indicating the degree to which a_i outranks a_j. Originally, [18] proposed the following expression:

$$S(a_i, a_j) = c(a_i, a_j). \prod_{\substack{t \in \{1, \ldots, n\}: \\ d_t(a_i, a_j) > c_t(a_i, a_j)}} \frac{1 - d_t(a_i, a_j)}{1 - c(a_i, a_j)}.$$

Two simpler variants have been proposed afterwards. The paper that introduced the possibility of using a discordance threshold u_t different than the preference threshold p_t [13] suggests:

$$S(a_i, a_j) = c(a_i, a_j). \prod_{t \in \{1, \ldots, n\}} [1 - d_t(a_i, a_j)].$$

Another possibility proposed by [13] is:

$$S(a_i, a_j) = c(a_i, a_j)[1 - d^{max}(a_i, a_j)]$$

with

$$d^{max}(a_i, a_j) = \max_{t \in \{1, \ldots, n\}} d_t(a_i, a_j).$$

In the remainder of this paper we will consider that this latter variant has been chosen.

3 An ELECTRE-type PASA without explicit category limits

Usually, sorting methods require an a priori definition of categories by indicating profiles, i.e., n-dimensional vectors of evaluations $(g_1(b_h), \ldots, g_n(b_h))$, that are either limits separating the categories or prototypes for the categories. The idea of using profiles as limits is implemented by ELECTRE TRI, in which each category C^h ($h = 1, \ldots, n_{cat}$) is defined by a lower-bound profile b_{h-1} and an upper-bound profile b_h. The pessimistic variant of ELECTRE TRI, for instance, sorts alternatives as follows:

$$a_i \in C^h \Leftrightarrow a_i S b_{h-1} \wedge \neg a_i S b_h.$$

The idea of using profiles as prototypes of the categories is used by PROAFTN [1], which does not require the categories to be ordered (hence applying to the more general category

of nominal classification methods). In this method, alternatives are sorted according to their similarity with the examples.

The idea we propose is to use exemplary alternatives (i.e., alternatives that a DM has already placed into one of the categories judging their merits holistically) to indirectly constrain the range of possible categories for the remaining alternatives, in the context of a PASA. Like ELECTRE TRI, this approach is meant for ordinal sorting problems only. However, unlike ELECTRE TRI, there are no profiles acting as category limits. On the other hand, like PROAFTN, it requires that some alternatives (or at least some fictitious profiles) are provided as examples for the categories. However, unlike PROAFTN, the approach we propose is designed for ordinal sorting problems, and it is not based on the idea of a similarity relation.

Let us define a set of categories $C^1, \ldots, C^{n_{cat}}$ in increasing preference order (C^1 is the worst category and $C^{n_{cat}}$ is the best one); formally, we will consider that each category is the set of alternatives that have been sorted into that category. Therefore, $a_i \in C^h$ means that the alterative a_i has been sorted into the category C^h.

As inputs, let us consider the DM has a set of alternatives $A^* \subset A$ that have been previously sorted. For instance, the alternatives may have been sorted by the DM based on an holistic evaluation of their absolute merit, or they may be fictitious alternatives imagined to fit the DM's notion of each category, or they may correspond to past decisions. A^* should contain at least one example for each category, so that each category has at least one reference. Let us emphasize that, contrarily to ELECTRE TRI, there is no a priori definition for the categories, which are indirectly defined using examples. This means that we can consider that the categories are vaguely defined in the DMs mind and can be seen as qualitative evaluation levels. Furthermore, contrarily to ELECTRE TRI, sorting one alternative may depend on how other alternatives were sorted.

Formally, we can write the inputs as:

$$A^* = C^1 \cup C^2 \cup \ldots \cup C^{n_{cat}} \tag{4}$$

This notation allows us to write a statement like "a_i is assigned to category C^h as an example" concisely as $a_i \in C^h$. We propose to base the sorting of the remaining actions $A \backslash A^*$ on one simple principle: "*if an alternative a_i outranks an alternative a_j, then the category of a_i must be at least as good as the category of a_j*":

$$a_i S a_j \wedge a_i \in C^h \wedge a_j \in C^s \Rightarrow h \geq s. \tag{5}$$

This principle can be rephrased as stating that "*alternatives belonging to a given category cannot be outranked by any alternative belonging to a lower category, and cannot outrank any alternative belonging to a higher category*". Although this requirement is not respected by ELECTRE TRI, it is a reasonable and logic principle given that $a_i S a_j$ means "a_i is at least as good as (or is not worse than) a_j".

The following corollaries result form (5):

(1) $a_i P a_j \wedge a_i \in C^h \wedge a_j \in C^s \Rightarrow h \geq s$;
(2) $a_i I a_j \wedge a_i \in C^h \wedge a_j \in C^s \Rightarrow h = s$;

The latter corollary stems from the fact that $a_i I a_j$ if and only if a_i and a_j outrank each other. Similarly, if there exists a cycle in the outranking relation ($a_i S a_j S a_k S \ldots S a_i$), then (5) also implies that all the alternatives involved in the cycle must be placed in the same category.

Considering the inputs (4) and principle (5), one may try to find an interval of potential categories for each alternative remaining to be sorted. For each alternative $a_i \in A \backslash A^*$, let us

denote this interval as $[C_{min}(a_i), C_{max}(a_i)]$. Hence, $C_{min}(a_i)$ is the lowest category where a_i may be placed, and $C_{max}(a_i)$ is the highest category where a_i may be placed, without violating (5) and given the current set of examples. Given a_i, principle (5) allows one to bound $C_{min}(a_i)$ and $C_{max}(a_i)$ as follows:

- If a_i outranks any alternative from A^*, say $a_j \in C^h$ for some h, then a_i should be placed into a category at least as good as C^h, i.e., $C_{min}(a_i) \geq C^h$.
- If a_i is outranked any alternative from A^*, say $a'_j \in C^{h'}$ for some h', then a_i should be placed into a category at most as good as $C^{h'}$, i.e., $C_{max}(a_i) \leq C^{h'}$.

Taking this into account we may write:

$$C_{min}(a_i) = C^h, \text{ with } h = \begin{cases} 1, & \text{if } \nexists a_j \in A^* : a_i S a_j \\ \max\{h \in \{1, \ldots, n_{cat}\} : a_i S a_j \wedge a_j \in C^h\}, & \text{otherwise;} \end{cases} \quad (6)$$

$$C_{max}(a_i) = C^{h'}, \text{ with } h' = \begin{cases} n_{cat}, & \text{if } \nexists a_j \in A^* : a_j S a_i \\ \min\{h \in \{1, \ldots, n_{cat}\} : a_j S a_i \wedge a_j \in C^h\}, & \text{otherwise.} \end{cases} \quad (7)$$

For some $a_i \in A \backslash A^*$ it may result that $C_{min}(a_i) = C_{max}(a_i)$. This means that one of the following cases holds, and the DM should include a_i in the only possible category for it:

- a_i is outranked by an alternative from C^1, which means that a_i should be placed in C^1 also;
- a_i outranks an alternative from $C^{n_{cat}}$, which means that a_i should be placed in $C^{n_{cat}}$ also.
- a_i outranks an alternative from a given category and is outranked by another (or the same) alternative from the same category, which means that a_i should be placed in that category.

For some $a_i \in A \backslash A^*$ it may result that $C_{min}(a_i) < C_{max}(a_i)$. In these situations, the DM may either decide to place the alternative in one of the categories in the interval $[C_{min}(a_i), C_{max}(a_i)]$, or may postpone this decision to a later stage, turning his/her attention to another unsorted alternative.

Finally, for some $a_i \in A \backslash A^*$ it may result that $C_{min}(a_i) > C_{max}(a_i)$. In these situations, the way the set of example alternatives A^* has been sorted is inconsistent. The cause of the inconsistency is that a_i outranks another alternative $a_j \in C^h$ and is at the same time outranked by an alternative $a_k \in C^{h'}$, with $h > h'$ (there may or not exist a cycle $a_j S a_k S a_i S a_j$). The DM now will have two options. The first solution is to reconsider the way the alternatives involved (a_k and a_j) were sorted, possibly changing the categories where they have been assigned to, in a manner that respects the principle (5). The other solution is to consider merging the categories C^h, $C^{h'}$, and all the categories in between, into a single category, especially if the number of categories is high.

To summarize, if all the parameters defining an outranking relation are set, then the following PASA may be followed to sort alternatives in a set A without explicitly defining the characteristics of the categories:

1. For each $h = 1, \ldots, n_{cat}$, select an alternative from A (or invent fictitious alternatives) to serve as examples of the category C^h. Let A^* be defined as in (4).
2. Choose an element $a_i \in A \backslash A^*$
3. Determine $C_{min}(a_i)$ and $C_{max}(a_i)$. Then,

   - if $C_{min}(a_i) = C_{max}(a_i) = C^h$, then add a_i to C^h (hence adding it to A^*);
   - if $C_{min}(a_i) < C_{max}(a_i)$, then either choose a category $C^h \in [C_{min}(a_i), C_{max}(a_i)]$ and add a_i to that category (hence adding it to A^*), or do nothing (a_i will remain in $A \backslash A^*$);

- if $C_{min}(a_i) > C_{max}(a_i)$ (inconsistency), then either revise the sorting judgements performed until this moment, or merge the categories $[C_{max}(a_i), C_{min}(a_i)]$ into a single one.

4. Choose a different alternative $a_i \in A \backslash A^*$ and return to 3.
5. If all the alternatives have been analyzed, either stop the procedure or reanalyze the set of alternatives remaining to be sorted, returning to 2.

After completing this procedure, a subset of alternatives $A^* \subseteq A$ will have been sorted. For the remaining alternatives, an interval of possible categories is determined. Considering the alternatives sorted in each of the categories, we may state that:

- No alternative outranks another alternative placed in a higher category;
- An alternative may or not outrank alternatives placed in lower categories;
- An alternative may or not outrank, or be outranked, by an alternative in the same category; it may happen that one is preferable to the other, or that they are indifferent, or that they are incomparable;
- If two alternatives are indifferent (they outrank each other), then they must belong to the same category;
- if there is a cycle in the outranking relation, then all the alternatives forming the cycle must belong to the same category (this idea of considering as indifferent alternatives forming a cycle in the outranking relation was already present in the first ELECTRE method, ELECTRE I [19]).

4 Extension of the idea towards an aggregation/disaggregation procedure

We believe that the kind of procedure presented in the previous section is particularly suited to the ideas of aggregation/disaggregation approaches [8], in which the DM provides information in the form of results that the method should yield. In such approaches, all or part of the parameter values are to be inferred, rather than directly asking the DM to provide them. The present section has the objective of presenting how the ELECTRE-based PASA may integrate an aggregation/disaggregation approach.

4.1 Constraints of the parameter values

Previous research [14] shows that inferring all the parameters of ELECTRE methods is a difficult mathematical problem. Hence, there exists some work devoted to inferring only a subset of the parameters [13,15,16]. Here, we will consider that all the parameters that are independent from one criterion to another have already been fixed. Indeed, to set the indifference, preference, discordance, or veto thresholds of one criterion, the DM may focus on that criterion only and needs not think about the remaining criteria. Contrarily to this, to set the criteria weights or the cutting level, the DM must compare and interrelate the multiple criteria, all at the same time. For this reason, and also because this will lead to manageable linear programming problems, we will consider that the variables (the parameters to be inferred) are only the weights $k_1, \ldots, k_n$ and the cutting level λ.

To avoid situations where some criteria have a null weight or an overwhelming weight, we will constrain the weights to an interval such that no criterion can have so little weight that it might almost be discarded, or so much weight that it would become a "dictator". We propose

that no criterion should weigh more than half of the total number of criteria, by defining the following set K of acceptable weight vectors:

$$K = \{(k_1, \ldots, k_n) : (1/n - x) \leq k_t \leq (1/n + x)(t = 1, \ldots, n) \quad \text{and} \quad \sum_{t=1}^{n} k_t = 1\},$$

with x such that:

$$1/n + x = \lceil n/2 \rceil \times (1/n - x)$$

This will allow each weight to vary within an interval centered at the value corresponding to an equal weights situation ($k_1 = k_2 = \ldots = k_n = 1/n$), while ensuring that no criterion weighs more than half of other criteria (rounded to the next integer, if n is odd). We therefore propose that the maximum value for the weights should not be greater than $\lceil n/2 \rceil$ times the minimum value for the weights. For instance, $n = 7$ leads to $k_t \in [2/35; 8/35]$.

Concerning the cutting level λ, we will accept the interval $\lambda \in [0, 1]$, although one would usually expect that $\lambda \geq 0.5$.

4.2 Inference of the weights and the cutting level

The inference of the weights $k_1, \ldots, k_n$ and the cutting level λ will be performed following the iterative and interactive process proposed in Sect. 3. The main difference will be that considering some parameters as variables, rather than having all the parameters fixed, will result in larger intervals of categories where each alternative may be placed.

As before, let us consider the DM is able to provide a set of alternatives $A^* = C^1 \cup C^2 \cup \ldots \cup C^{n_{cat}}$ as examples for the different categories. The process should start with $C^h \neq \emptyset$ for $h = 1, \ldots, n_{cat}$. These examples implicitly define a set J of compatible vectors $(k_1, \ldots, k_n, \lambda)$. By compatible, we mean that $\forall (k_1, \ldots, k_n, \lambda) \in J$, the principle (5) is not violated considering A^*. As a note, let us mention that formally the sets J and A^* should be written as J_r and A_r^*, with r being an index identifying the iteration, since these sets will change during the interactive process. Nevertheless, we will omit these indices, to keep a simpler notation.

$$J = \{(k, \lambda) : \lambda - S(a_p, a_i) \geq \varepsilon, \forall a_p, a_i \in A^* : a_p \in C^h, a_i \in C^s, h < s$$
$$\wedge \sum_{t=1}^{n} k_t = 1 \wedge k_t \geq 0(t = 1, \ldots, n)\}$$

where ε is an arbitrarily fixed near-zero positive quantity, needed to model the strict inequality $\lambda - S(a_p, a_i) > 0$, i.e., $\neg(a_p S a_i)$.

We will now present a linear programming (LP) formulation to find the parameter values allowing an alternative $a_i \in A \backslash A^*$ to be sorted in a given category C^h, subject to the constraints $(k_1, \ldots, k_n, \lambda) \in J$. In other words, we will solve an LP to test whether a_i can be sorted in a given category, given the examples defining A^*. Repeating this test for all the categories will indicate the interval of categories $[C_{min}(a_i), C_{max}(a_i)]$ where it would be reasonable to sort a_i.

All the constraints implied by out principle (5) refer to "negative" outranking statements $\neg(a_p S a_i)$, whenever a_p has been placed in a category lower than that of a_i. Hence, for very high values of λ, namely for $\lambda = 1$, it is likely that no outranking holds (the exceptions might come from alternatives dominating other alternatives), thereby satisfying all the constraints. For this reason, the approach we suggest is based on finding the minimum value for λ that

allows an alternative a_i to be placed in a category C^h. It is then up to the DM to decide whether that minimum λ is still too high (meaning that placing a_i in C^h is unacceptable) or not.

The following LP can be solved to find the minimum λ allowing an alternative $a_i \in A \backslash A^*$ to be sorted in a given category C^h, with parameter values $(k_1, \ldots, k_n, \lambda) \in J$ and respecting (5):

(LP1)   Min $\lambda_{i,h}$
  s.t. $\lambda_{i,h} - S(a_i, a_m) \geq \varepsilon, \forall a_m \in C^t, with\ t > h$
      $\lambda_{i,h} - S(a_p, a_i) \geq \varepsilon, \forall a_p \in C^s, with\ s < h$ //Constraints ensuring(5)
      $k_j \leq (1/n + x), (j = 1, \ldots, n)$ // Upper bound for k_j
      $k_j \geq (1/n - x), (j = 1, \ldots, n)$ // Lower bound k_j
      $(k, \lambda_{i,h}) \in J$

Let $\lambda_{i,h}^*$ denote the optimal value of LP1. This value indicates that it is possible to sort a_i into C^h for cutting level values in the interval $[\lambda_{i,h}^*, 1]$. The cutting level, let us remind, indicates whether the credibility of the outranking relations is or not sufficient to establish an outranking conclusion. It can be interpreted as meaning the required majority of the criteria weights in favor of an outranking (possibly weakened in case of significant discordance) needed to accept that conclusion. If $\lambda_{i,h}^*$ is low, i.e., below 0.5 (meaning a required majority of 50%) or not much greater than 0.5, then the DM can conclude that sorting a_i into C^h is perfectly acceptable. In contrast, if $\lambda_{i,h}^*$ is high, i.e., equal to 1 or near 1, then the DM can conclude that sorting a_i into C^h is unacceptable.

For each $a_i \in A \backslash A^*$, it is possible to solve n_{cat} linear programs (although it may not be necessary to solve as many, as we show next), to compute the following vector:

$$s_i^{\lambda} = (\lambda_{i,1}^*, \lambda_{i,2}^*, \ldots, \lambda_{i,n_{cat}}^*). \tag{8}$$

Based on this information, the DM may decide to sort a_i into one of the categories (hence adding a_i to $A \backslash A^*$), or may decide to postpone a decision on a_i and proceed to analyze a different alternative. For instance, if there are three categories and $s_i^{\lambda} = (0.25, 0.85, 0.90)$, then the DM could easily decide to place a_i into C^1, otherwise a cutting level as high as 0.85 would be required. On the other hand, if $s_i^{\lambda} = (0.25, 0.33, 0.40)$, then these computations would be of no help to the DM, because usually the DM considers $\lambda \geq 0.5$, therefore greater than the minimum required for each of the three possibilities of assignment. As in the procedure presented in Sect. 3, it is also possible to reach a situation of inconsistency. This would occur if all the values of s_i^{λ} were considered too high, e.g., $s_i^{\lambda} = (0.95, 0.82, 0.90)$. In such cases, following the procedure in Sect. 3, the DM would either have to revise previous judgements or to merge categories.

In summary, the PASA proposed in Sect. 3 may be adapted to the case where the weights and cutting level are not fixed, based on computing the minimum cutting level that allows placing each action into each category. If the DM specifies a value λ for the cutting level, then this will define the minimum and maximum category to which each $a_i \in A \backslash A^*$ may be assigned to, given the assignments made previously:

$$C_{min}(a_i) = \min\{C^h : \lambda_{i,h}^* \leq \lambda\}, \quad and\ C_{max}(a_i) = \max\{C^h : \lambda_{i,h}^* \leq \lambda\}.$$

For instance, consider there are 5 categories and one has $s_i^{\lambda} = (0.85, 0.53, 0.40, 0.78, 0.92)$ for a given a_i. In this situation, if the DM indicates the cutting level is $\lambda = 0.6$ implies that $C_{min}(a_i) = C^2$ and $C_{max}(a_i) = C^3$. However, the DM does not have to commit to a precise value for λ. The same interval of possible categories for a_i would be reached if the DM simply expressed that 0.78 was an excessive value for the cutting level.

We will now show that s_i^λ (8) obeys to what we could call a "convex-shape property": $\lambda_{i,h}^* \leq \max\{\lambda_{i,h-1}^*, \lambda_{i,h+1}^*\}$. Therefore, it may not be necessary to compute all the elements of s_i^λ to find $C_{min}(a_i)$ and $C_{max}(a_i)$ ($a_i \in A \backslash A^*$). For instance, one may proceed as follows:

$h \leftarrow 1$
WHILE $h \leq n_{cat}$ and $\lambda_{i,h}^*$ is unacceptably high DO
$\qquad h \leftarrow h + 1$
END WHILE
$C_{min}(a_i) \leftarrow h$
IF $h > n_{cat}$ THEN STOP {Inconsistent situation}
$h' \leftarrow n_{cat}$
WHILE $h' \geq 1$ and $\lambda_{i,h'}^*$ is unacceptably high DO
$\qquad h' \leftarrow h' - 1$
END WHILE
$C_{max}(a_i) \leftarrow h'$ {Inconsistent if $h' < 1$ or $h' < h$}.

For instance, suppose that a DM considers a cutting level is too high when it exceeds 0.7. In this case, if s_i^λ=(0.25, [0.25], [0.34], 0.54, 0.72) then only the first, the fourth, and the fifth values would have to be computed: since $0.25 < 0.7$ and $0.54 < 0.7$, all the values in between must be lower than 0.7.

The following Proposition establishes the "convex-shape property":

Proposition 1 *If a_i can be placed in C^x and a_i can be placed in C^z ($x < z - 1$) given J then, for all $y \in \{x + 1, x + 2, \ldots, z - 1\}$, a_i can be placed in C^y.*

Proof Since a_i can be placed in C^z, it must hold:

$$\exists (k, \lambda) \in J : \nexists a_\alpha \in C^w, w < z : a_\alpha \, S \, a_i. \tag{9}$$

On the other hand, Since a_i can be placed in C^x, it must hold:

$$\exists (k, \lambda) \in J : \nexists a_\beta \in C^b, b > x : a_i \, S \, a_\beta. \tag{10}$$

We will now suppose the proposition is false and show this leads to a contradiction. If the proposition was false, then there would exist an index y greater than x and lower than z such that a_i could not be placed in C^y, i.e.:

$$\forall (k, \lambda) \in J, [(\exists \, a_{\alpha'} \in C^{w'}, w' < y : a_{\alpha'} \, S \, a_i)$$
$$\vee \, (\exists \, a_{\beta'} \in C^{b'}, b' > y : a_i \, S \, a_{\beta'})] \tag{11}$$

The previous expression states that, for every vector of parameter values in J (the set of acceptable vector of parameter values given the examples provided so far), either a) there exists an action $a_{\alpha'}$ in a category lower than C^y that outranks a_i; or b) there exists an action $a_{\beta'}$ in a category higher than C^y that is outranked by a_i. However, since $w' < y \wedge y < z \Rightarrow w' < z$, then the case a) contradicts (9), whereas since $b' > y \wedge y > x \Rightarrow b' > x$, then the case b) contradicts (10). Hence, (11) leads to a contradiction. □

5 Illustrative examples

In this section we will illustrate the use of the procedure proposed in this paper using two sets of data. In both cases, we considered the weights and cutting level as variables, and to impose

Table 1 Evaluations on six criteria for 20 stocks

	g_1	g_2	g_3	g_4	g_5	g_6		g_1	g_2	g_3	g_4	g_5	g_6
a_0	0.82	0.45	0.26	−4.7	−100	0.45	a_{10}	0.8	0.58	0.62	13.7	34.6	1.54
a_1	0.41	0.63	0.03	2.28	−20	2.04	a_{11}	1.23	0.37	0.64	8.97	45.9	0.96
a_2	0.57	0.2	0.1	6.08	−33.3	1.08	a_{12}	0.24	0.28	0.73	−1.75	0	0.72
a_3	0.24	0.02	0.08	2.41	−53.5	0.62	a_{13}	0.26	0.65	0.58	4.88	7.14	0.9
a_4	0	0.46	0.62	5.04	−76.5	3.02	a_{14}	1.1	0.76	0.54	0.29	0	0.73
a_5	0.93	0.02	0.14	2.82	6.38	0.72	a_{15}	1.79	0.55	0.73	5.88	−100	2.69
a_6	0.01	0.69	0.77	7.55	−40	3.23	a_{16}	1.02	1.06	0.82	5.5	6.38	0.73
a_7	0.86	0.86	0.86	4.28	3.71	0.57	a_{17}	1.32	1.12	0.94	12.06	−61	2.69
a_8	2.16	0.6	0.12	2.11	56.3	0.51	a_{18}	1.36	0.04	1.02	1.79	110	2.31
a_9	1.24	0.12	0.62	11.65	12.5	1.17	a_{19}	0.57	0.17	0.23	−11.5	0	0.52

Table 2 Indifference and preference thresholds

	g_1	g_2	g_3	g_4	g_5	g_6
q_j	0.05	0.05	0	0.1	8.72	0.05
p_j	0.25	0.2	0.2	0.5	10	0.25

strict inequalities in LP1 we used $\varepsilon = 0.00001$. Since the DMs of the original studies were not present, and since the results obtained in these studies are not intended to be compared with the ones to be obtained with our procedure, we followed a simple "rule-of-thumb" to provide examples. This rule is to choose for some alternatives the category corresponding to the minimum value in the vector s_i^λ corresponding to them. Of course, in the presence of a real DM, his/her preferences and requirements, based on holistic judgement and experience, should strongly influence the choice of the examples. Hence, these illustrative examples should not be seen as indicating a way to automatically sort the alternatives.

5.1 First example

We will use data form an application for sorting stocks listed in the Athens Stock Exchange [7], namely 20 alternatives from the commercial sector, which were evaluated on 6 criteria (Table 1). The criteria names are not relevant here, therefore we will note only that all criteria are to be maximized, except $g_3(.)$, where the lower the values, the better. Three categories are to be used: C^1 ("not attractive"), C^2 ("uncertain"), and C^3 ("attractive", the best category).

We will use the original values [7] for the indifference and preference thresholds (Table 2), but we will not use the veto thresholds, i.e., we will ignore discordance. This occurs, in this particular example, because the original values for the veto thresholds led to many veto situations. Since our procedure is based on varying weights and the cutting level to see if it is possible to avoid all the outrankings forbidden by (5), having many veto situations is undesirable to the extent that many of the forbidden outrankings will not hold regardless of the weights and the cutting level values.

The remaining parameters are considered as variables. The cutting level λ can assume values in [0, 1], whereas the weight vectors are constrained to a polytope K defined as suggested in Sect. 4.1. (no criterion is preponderant and no criterion is negligible):

$$K = \{(k_1, \ldots, k_6) : 1/12 \leq k_j \leq 3/12 (j = 1, \ldots, 6) \wedge k_1 + k_2 + \ldots + k_6 = 1\}.$$

Let us suppose the DM starts by stating one example for each of the three categories: $a_3 \rightarrow C^1$, $a_{14} \rightarrow C^2$, and $a_{11} \rightarrow C^3$, i.e., $A^* = C^1 \cup C^2 \cup C^3$, with $C^1 = \{a_3\}$, $C^2 = \{a_{14}\}$, and $C^3 = \{a_{11}\}$. Table 3 depicts the optimal values $\lambda_{i,h}^*$ of (LP1) for each potential way of sorting the remaining alternatives in $A \setminus A^*$.

If as usual in ELECTRE the DM states that λ will be at least 0.5, then any sorting possibility with $\lambda_{i,h}^* < 0.5$ will be feasible (the sorting is possible for any $\lambda \in [\lambda_{i,h}^*, 1]$). For instance, we may see that alternative a_0 can perfectly be sorted into C^1 or C^2. However, sorting a_0 into C^3 would require accepting $\lambda \geq 0.75$, which might be considered too high. If the DM considered that $\lambda \geq 0.60$ would already be too high, then the alternative a_{13} could be placed only in C^2. Concerning the remaining alternatives, there are typically two categories (sometimes three) where each one could easily be placed. This lack of constraints is natural given the scarce information used up to this moment. In the absence of a DM, we will choose as a set of new examples those that had only one element in s_i^λ lower than 0.5: $a_0 \rightarrow C^1$, a_5, a_{13}, and a_{16} are sorted into C^2, a_8, a_9, and a_{10} are sorted into C^3. The results are now those depicted in Table 4, where it can be seen that each value for $\lambda_{i,h}^*$ is not lower than the respective value in Table 3. This is natural, since the minimum value of the linear program (LP1) does not decrease as new constraints are added.

Again, we remind that this procedure is not to be followed in such an automatic way. Rather, the holistic evaluations of the DM should play the main role. As a matter of fact, when the most obvious choices are made (e.g., considering at this point adding $a_7 \rightarrow C^2$ and $a_{19} \rightarrow C^1$, two situations where $\lambda_{i,h}^*$ is clearly low compared with the other sorting possibilities for the same alternatives), they are not likely to add much information. Therefore, we will continue now considering $a_4 \rightarrow C^1$, a_2, $a_7 \rightarrow C^2$, and $a_{17} \rightarrow C^3$, where $a_2 \rightarrow C^2$ would not be an obvious choice for sorting a_2, but would be the result of a request by the DM. The corresponding results are provided in Table 5.

Table 3 Minimum cutting levels $\lambda_{i,h}^*$ for sorting the alternatives at iteration 1

	C^1	C^2	C^3		C^1	C^2	C^3
a_0	**0.342**	0.500	0.750	a_{10}	0.500	0.500	**0.379**
a_1	0.410	0.350	0.463	a_{11}			✓
a_2	0.371	0.342	0.463	a_{12}	0.356	0.438	1.000
a_3	✓			a_{13}	0.610	**0.349**	0.600
a_4	0.355	0.417	0.504	a_{14}		✓	
a_5	0.600	**0.348**	0.529	a_{15}	0.440	0.440	0.379
a_6	0.369	0.344	0.504	a_{16}	0.713	**0.342**	0.500
a_7	0.442	0.342	0.667	a_{17}	0.500	0.500	0.351
a_8	0.604	0.542	**0.371**	a_{18}	0.529	0.500	0.342
a_9	0.650	0.500	**0.342**	a_{19}	0.355	0.397	0.750

Table 4 Minimum cutting levels $\lambda_{i,h}^*$ for sorting the alternatives at iteration 2

	C^1	C^2	C^3		C^1	C^2	C^3
a_0	✓			a_{10}			✓
a_1	0.500	0.500	0.542	a_{11}			✓
a_2	0.542	0.517	0.500	a_{12}	0.465	0.510	1.000
a_3	✓			a_{13}		✓	
a_4	0.417	0.500	0.713	a_{14}		✓	
a_5		✓		a_{15}	0.583	0.4480	0.417
a_6	0.419	0.417	0.542	a_{16}		✓	
a_7	0.600	**0.417**	1.000	a_{17}	0.600	0.542	0.417
a_8			✓	a_{18}	0.558	0.533	0.417
a_9			✓	a_{19}	**0.500**	0.688	0.833

Table 5 Minimum cutting levels $\lambda_{i,h}^*$ for sorting the alternatives at iteration 3

	C^1	C^2	C^3		C^1	C^2	C^3
a_0	✓			a_{10}			✓
a_1	0.613	**0.500**	0.546	a_{11}			✓
a_2		✓		a_{12}	**0.465**	0.522	1.000
a_3	✓			a_{13}		✓	
a_4	✓			a_{14}		✓	
a_5		✓		a_{15}	0.583	0.474	0.486
a_6	0.542	**0.417**	0.542	a_{16}		✓	
a_7		✓		a_{17}			✓
a_8			✓	a_{18}	0.558	0.575	**0.500**
a_9			✓	a_{19}	0.515	0.688	0.833

Table 6 Minimum cutting levels $\lambda_{i,h}^*$ for sorting the alternatives at iteration 4

	C^1	C^2	C^3		C^1	C^2	C^3
a_0	✓			a_{10}			✓
a_1		✓		a_{11}			✓
a_2		✓		a_{12}	✓		
a_3	✓			a_{13}		✓	
a_4	✓			a_{14}		✓	
a_5		✓		a_{15}	0.750	0.750	0.700
a_6		✓		a_{16}		✓	
a_7		✓		a_{17}			✓
a_8			✓	a_{18}			✓
a_9			✓	a_{19}	0.556	0.693	0.833

Table 7 Minimum cutting levels $\lambda_{i,h}^*$ for sorting the alternatives after removing example $a_{18} \to C^3$

	C^1	C^2	C^3		C^1	C^2	C^3
a_0	✓			a_{10}			✓
a_1		✓		a_{11}			✓
a_2		✓		a_{12}	✓		
a_3	✓			a_{13}		✓	
a_4	✓			a_{14}		✓	
a_5		✓		a_{15}	0.583	0.542	0.700
a_6		✓		a_{16}		✓	
a_7		✓		a_{17}			✓
a_8			✓	a_{18}	0.617	0.577	0.542
a_9			✓	a_{19}	0.556	0.693	0.833

Table 8 Performances of the alternatives to be sorted (second example)

	g_1	g_2	g_3	g_4	g_5	g_6	g_7		g_1	g_2	g_3	g_4	g_5	g_6	g_7
a_0	35.8	67.0	19.7	0.0	0.0	5.0	4.0	a_{20}	15.5	8.5	56.2	5.5	1.8	4.0	2.0
a_1	16.4	14.5	59.8	7.5	5.2	5.0	3.0	a_{21}	13.2	9.1	74.1	6.4	5.0	2.0	2.0
a_2	35.8	24.0	64.9	2.1	4.5	5.0	4.0	a_{22}	9.1	4.1	44.8	3.3	10.4	3.0	4.0
a_3	20.6	61.7	75.7	3.6	8.0	5.0	3.0	a_{23}	12.9	1.9	65.0	14.0	7.5	4.0	3.0
a_4	11.5	17.1	57.1	4.2	3.7	5.0	2.0	a_{24}	5.9	−27.7	77.4	16.6	12.7	3.0	2.0
a_5	22.4	25.1	49.8	5.0	7.9	5.0	3.0	a_{25}	16.9	12.4	60.1	5.6	5.6	3.0	2.0
a_6	23.9	34.5	48.9	2.5	8.0	5.0	3.0	a_{26}	16.7	13.1	73.5	11.9	4.1	2.0	2.0
a_7	29.9	44.0	57.8	1.7	2.5	5.0	4.0	a_{27}	14.6	9.7	59.5	6.7	5.6	2.0	2.0
a_8	8.7	5.4	27.4	4.5	4.5	5.0	2.0	a_{28}	5.1	4.9	28.9	2.5	46.0	2.0	2.0
a_9	25.7	29.7	46.8	4.6	3.7	4.0	2.0	a_{29}	24.4	22.3	32.8	3.3	5.0	3.0	4.0
a_{10}	21.2	24.6	64.8	3.6	8.0	4.0	2.0	a_{30}	29.5	8.6	41.8	5.2	6.4	2.0	3.0
a_{11}	18.3	31.6	69.3	2.8	3.0	4.0	3.0	a_{31}	7.3	−64.5	67.5	30.1	8.7	3.0	3.0
a_{12}	20.7	19.3	19.7	2.2	4.0	4.0	2.0	a_{32}	23.7	31.9	63.6	12.1	10.2	3.0	2.0
a_{13}	9.9	3.5	53.1	8.5	5.3	4.0	2.0	a_{33}	18.9	13.5	74.5	12.0	8.4	3.0	3.0
a_{14}	10.4	9.3	80.9	1.4	4.1	4.0	2.0	a_{34}	13.9	3.3	78.7	14.7	10.1	2.0	2.0
a_{15}	17.7	19.8	52.8	7.9	6.1	4.0	4.0	a_{35}	−13.3	−31.1	63.0	21.2	29.1	2.0	1.0
a_{16}	14.8	15.9	27.9	5.4	1.8	4.0	2.0	a_{36}	6.2	−3.2	46.1	4.8	10.5	2.0	1.0
a_{17}	16.0	14.7	53.5	6.8	3.8	4.0	4.0	a_{37}	4.8	−3.3	71.1	8.6	11.6	2.0	2.0
a_{18}	11.7	10.0	42.1	12.2	4.3	5.0	2.0	a_{38}	0.1	−9.6	42.5	12.9	12.4	1.0	1.0
a_{19}	11.0	4.2	60.8	6.2	4.8	4.0	2.0	a_{39}	13.6	9.1	76.0	17.1	10.3	1.0	1.0

Sorting many alternatives at each iteration increases the risk of reaching an inconsistency situation. For instance, let us suppose the DM decided to sort all the remaining alternatives except a_{15} and a_{19} to the categories where the value $\lambda_{i,h}^*$ was minimum. This would lead to the results in Table 6, where the DM must accept a cutting level of at least 0.7 to sort a_{15}. If we suppose this was considered too high by the DM, then some examples would have to be withdrawn. To inform this decision, one may check which constraints of (LP1)

are binding at the optimal solutions concerning a_{15}, which would show that the example concerning a_{18} is responsible for the high values in $\lambda^*_{15,1}$ and $\lambda^*_{15,2}$. Withdrawing only the example a_{18} (removing the alternative from C^3) leads to Table 7. At this point the DM could decide $a_{15} \to C^2$, which would not change s^*_{19}. Then, $a_{19} \to C^1$ would be a natural continuation for this process, yielding $s^*_{18} = (0.617, 0.577, 0.750)$. Finally, $a_{18} \to C^2$ could be the natural ending for this process.

5.2 Second example

We will now present a second example with more alternatives to sort and more categories. We will use data from [4], referring to the evaluation of 40 companies to be sorted into 5 cate-

Table 9 Thresholds associated with the criteria

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
q_j	1	4	1	1	0	0	0
p_j	2	6	3	2	3	0	0
u_j	35	90	45	20	35	3.5	3.5
v_j	45	120	55	30	45	6.5	6.5

Table 10 Minimum cutting levels $\lambda^*_{i,h}$ at iteration 1 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0	1.00	1.00	1.00	1.00	**0.41**	a_{20}	1.00	0.77	0.43	0.41	0.78
a_1	1.00	1.00	0.47	0.41	0.77	a_{21}		✓			
a_2	1.00	1.00	0.77	0.77	**0.41**	a_{22}	0.54	0.44	0.41	0.44	0.54
a_3	0.73	0.73	0.44	0.41	0.41	a_{23}			✓		
a_4	0.84	0.68	0.41	0.41	0.54	a_{24}	0.41	0.54	0.77	1.00	1.00
a_5	0.97	0.97	0.54	0.41	0.41	a_{25}	0.95	0.54	0.41	0.41	0.95
a_6	0.96	0.96	0.54	0.44	0.41	a_{26}	0.77	**0.41**	0.70	0.70	1.00
a_7	1.00	1.00	0.77	0.77	**0.41**	a_{27}	0.95	0.54	0.68	0.68	1.00
a_8	0.77	0.54	0.47	0.48	0.41	a_{28}	✓				
a_9	1.00	0.77	0.77	0.47	0.41	a_{29}					✓
a_{10}	0.77	0.73	0.44	0.53	0.53	a_{30}	0.89	0.77	0.41	0.41	0.47
a_{11}	1.00	0.77	0.54	0.50	0.52	a_{31}	0.41	0.41	0.41	1.00	1.00
a_{12}	1.00	0.77	0.76	0.54	0.43	a_{32}	0.56	0.50	0.43	0.51	0.56
a_{13}	0.54	0.54	0.54	0.54	1.00	a_{33}	0.56	0.48	0.41	0.53	0.77
a_{14}	0.54	0.41	0.53	0.53	0.77	a_{34}	0.42	0.53	1.00	1.00	1.00
a_{15}	1.00	1.00	0.80	**0.43**	0.71	a_{35}	**0.41**	0.77	0.77	0.89	1.00
a_{16}	1.00	0.77	0.73	0.41	0.54	a_{36}	**0.41**	0.71	0.71	0.71	0.71
a_{17}				✓		a_{37}	0.50	0.77	0.77	0.77	1.00
a_{18}	0.73	0.73	0.54	0.49	0.54	a_{38}	**0.41**	0.77	0.77	0.77	0.77
a_{19}	0.67	0.56	0.53	0.53	1.00	a_{39}	0.41	0.53	1.00	1.00	1.00

gories (in the original application there were only 3 categories) representing their risk, based on their performances on 7 criteria. Criteria g_1, g_2, g_6, and g_7 are to be maximized, whereas g_3, g_4, and g_5 are to be minimized. Table 8 depicts the performances of the alternatives on these criteria.

We will consider as fixed the indifference, preference, discordance, and veto thresholds associated with each criterion, indicated in Table 9. We now enable discordance to occur, although we have chosen values for u_j and v_j that do not allow veto situations to occur frequently when comparing the alternatives.

The remaining parameters are again considered as variables. The cutting level λ can assume values in $[0, 1]$, whereas the weight vectors are constrained to a polytope K defined as suggested in Sect. 4.1:

$$K = \{(k_1, \ldots, k_7) : 2/35 \leq k_j \leq 8/35 (j = 1, \ldots, 7) \wedge k_1 + k_2 + \ldots + k_7 = 1\}.$$

Let us suppose the DM starts by stating one example for each of the five categories: $a_{28} \rightarrow C^1$, $a_{21} \rightarrow C^2$, $a_{23} \rightarrow C^3$, $a_{17} \rightarrow C^4$, and $a_{29} \rightarrow C^5$. Table 10 depicts the optimal values $\lambda_{i,h}^*$ of (LP1) for each potential way of sorting the remaining alternatives in $A \backslash A^*$.

Despite the scarce information used, there are some possible assignments that are very natural to accept, such as sorting a_0 into C^5. If we hypothesize that the DM wants to consider a cutting level lower than 0.7, then the following examples could be also added: $a_{35}, a_{36}, a_{38} \rightarrow C^1, a_{26} \rightarrow C^2, a_{15} \rightarrow C^4, a_0, a_2, a_7 \rightarrow C^5$. Adding these examples would lead to the results

Table 11 Minimum cutting levels $\lambda_{i,h}^*$ at iteration 2 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}	1.00	0.77	0.48	0.46	0.78
a_1	1.00	1.00	0.52	0.45	0.77	a_{21}		✓			
a_2					✓	a_{22}	0.54	0.46	0.45	0.48	0.54
a_3	0.73	0.73	0.51	0.45	0.45	a_{23}			✓		
a_4	0.84	0.68	0.49	0.49	0.54	a_{24}	**0.45**	0.59	0.77	1.00	1.00
a_5	0.97	0.97	0.63	0.45	0.45	a_{25}	0.95	0.55	0.50	0.50	0.95
a_6	0.96	0.96	0.63	0.50	0.45	a_{26}		✓			
a_7					✓	a_{27}	0.95	0.55	0.68	0.68	1.00
a_8	0.77	0.54	0.50	0.48	0.50	a_{28}	✓				
a_9	1.00	0.77	0.77	0.50	0.45	a_{29}					✓
a_{10}	0.77	0.73	0.51	0.53	0.55	a_{30}	0.89	0.77	0.52	0.45	0.47
a_{11}	1.00	0.77	0.54	0.50	0.53	a_{31}	0.50	0.52	0.52	1.00	1.00
a_{12}	1.00	0.77	0.77	0.54	**0.45**	a_{32}	0.77	0.51	**0.45**	0.51	0.56
a_{13}	0.55	0.54	0.56	0.54	1.00	a_{33}	0.77	0.50	**0.45**	0.53	0.95
a_{14}	0.54	**0.45**	0.54	0.54	0.77	a_{34}	**0.47**	0.53	1.00	1.00	1.00
a_{15}				✓		a_{35}	✓				
a_{16}	1.00	0.77	0.73	**0.46**	0.55	a_{36}	✓				
a_{17}				✓		a_{37}	**0.53**	0.77	0.77	0.79	1.00
a_{18}	0.76	0.73	0.54	0.54	0.54	a_{38}	✓				
a_{19}	0.67	0.56	0.53	0.53	1.00	a_{39}	**0.45**	0.59	1.00	1.00	1.00

Table 12 Minimum cutting levels $\lambda^*_{i,h}$ at iteration 3 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}	1.00	0.77	0.57	**0.53**	1.00
a_1	1.00	1.00	0.55	0.53	0.77	a_{21}		✓			
a_2					✓	a_{22}	0.58	0.57	0.57	0.57	0.62
a_3	0.98	0.98	0.54	0.53	0.53	a_{23}			✓		
a_4	0.84	0.68	0.53	0.53	0.73	a_{24}	✓				
a_5	1.00	1.00	0.63	0.53	0.53	a_{25}	0.95	0.61	0.54	0.54	0.95
a_6	1.00	1.00	0.63	0.56	**0.53**	a_{26}		✓			
a_7					✓	a_{27}	0.95	0.60	0.68	0.68	1.00
a_8	0.77	0.59	0.53	0.53	0.77	a_{28}	✓				
a_9	1.00	1.00	0.77	0.59	0.57	a_{29}					✓
a_{10}	0.77	0.77	0.56	0.58	0.57	a_{30}	0.89	0.77	0.54	0.53	0.56
a_{11}	1.00	1.00	0.68	**0.53**	0.57	a_{31}	**0.54**	0.57	0.58	1.00	1.00
a_{12}					✓	a_{32}			✓		
a_{13}	0.61	0.59	0.77	0.77	1.00	a_{33}			✓		
a_{14}		✓				a_{34}	✓				
a_{15}				✓		a_{35}	✓				
a_{16}				✓		a_{36}	✓				
a_{17}				✓		a_{37}	✓				
a_{18}	0.76	0.73	0.57	0.56	0.77	a_{38}	✓				
a_{19}	0.68	0.60	0.77	0.77	1.00	a_{39}	✓				

in Table 11. Tables 12–16 present a possible sequence based on a systematical choice of the lower $\lambda^*_{i,h}$ values.

After the last iteration, only two alternatives remain to be sorted: a_{22} and a_{25}. There remains some freedom concerning how to sort them. For instance, sorting a_{22} into C^1, C^2, C^3, or C^4 (C^5 would require a higher cutting level) does not change the values of s^λ_{25}, meaning it allows sorting a_{25} into one of the three categories in the interval $[C^2, C^4]$. Likewise, sorting a_{25} into one of the three categories in the interval $[C^2, C^4]$ does not constrain the sorting possibilities for a_{22}. Therefore, the procedure would not be of further help to the DM concerning how to sort these two alternatives.

6 Conclusion

This work presented a PASA based on the ELECTRE methodology. Its purpose is to support a DM who does not wish to provide a formal and explicit definition for the categories in an ordinal sorting problem. Rather, the categories can be seen as subjective qualitative evaluation levels vaguely established in the DM's mind. The PASA supports the DM by enforcing the principle (5): "if an alternative outranks (is as good as) a second one, then it must be placed on the same category or in a better category".

We have shown that if the outranking relation is completely defined (all the parameter values are fixed), then it is possible to constrain the range of categories where an alternative

Table 13 Minimum cutting levels $\lambda^*_{i,h}$ at iteration 4 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}				✓	
a_1	1.00	1.00	0.56	0.54	0.77	a_{21}		✓			
a_2					✓	a_{22}	0.58	0.57	0.59	0.57	0.62
a_3	0.98	0.98	0.58	0.55	0.54	a_{23}			✓		
a_4	0.84	0.68	0.63	0.54	0.73	a_{24}	✓				
a_5	1.00	1.00	0.77	0.58	0.54	a_{25}	0.95	0.62	0.55	0.55	0.95
a_6					✓	a_{26}		✓			
a_7					✓	a_{27}	0.95	0.60	0.68	0.68	1.00
a_8	0.77	0.62	0.59	0.54	0.77	a_{28}	✓				
a_9	1.00	1.00	0.85	0.59	0.59	a_{29}					✓
a_{10}	0.77	0.77	0.58	0.58	0.58	a_{30}	0.89	0.77	0.59	0.55	0.59
a_{11}				✓		a_{31}	✓				
a_{12}					✓	a_{32}			✓		
a_{13}	0.62	0.59	0.77	0.77	1.00	a_{33}			✓		
a_{14}		✓				a_{34}	✓				
a_{15}				✓		a_{35}	✓				
a_{16}				✓		a_{36}	✓				
a_{17}				✓		a_{37}	✓				
a_{18}	0.76	0.73	0.58	0.57	0.77	a_{38}	✓				
a_{19}	0.68	0.60	0.77	0.77	1.00	a_{39}	✓				

may be sorted given sorting examples previously expressed. We have also shown how the same idea may be used in an aggregation/disaggregation approach, considering the weights and the cutting level are not fixed a priori, but constrained by the examples provided. In this context, we established a "convex-shape property" that shows that there are no "holes" in the range of possible categories for an alternative: if it can be placed in C^x and C^y, then it can be placed in any category in $[C^x, C^y]$.

Finally, we have proposed, and we have illustrated with some examples, a process based on the computation of the minimum cutting level $\lambda^*_{i,h}$ needed to allow an alternative a_i to be sorted into a category C^h.

The spirit of the approach proposed in this paper is that of aiding a DM to perform sorting decisions in a consistent manner, rather than deriving a model to substitute the DM. Each sorting decision will constrain the subsequent ones if consistency with principle (5) is to be maintained. Therefore, the DM is aided when the procedure shows what is the minimum and maximum category where an alternative might be sorted. Sometimes, the minimum and maximum will coincide. Some other times, it is up to the DM to make the decision. In this sense, the procedure both assists the DM and is assisted by the DM. The DM assists the procedure by choosing one among a range of potential categories for an alternative. On the other hand, the DM will be assisted subsequently as the newly sorted alternative may contribute to narrow the interval of possible categories for future alternatives.

Comparing this approach with ELECTRE TRI and its aggregation/disaggregation extensions [3], the most salient feature of the approach proposed in this paper is that the explicit

Table 14 Minimum cutting levels $\lambda^*_{i,h}$ at iteration 5 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}				✓	
a_1	1.00	1.00	0.59	0.54	0.77	a_{21}		✓			
a_2					✓	a_{22}	0.58	0.58	0.59	0.58	0.73
a_3	0.98	0.98	0.58	0.58	0.54	a_{23}			✓		
a_4				✓		a_{24}	✓				
a_5					✓	a_{25}	0.95	0.62	0.56	0.56	0.95
a_6					✓	a_{26}		✓			
a_7					✓	a_{27}	0.95	0.60	0.68	0.68	1.00
a_8				✓		a_{28}	✓				
a_9	1.00	1.00	0.85	0.67	0.59	a_{29}					✓
a_{10}	0.77	0.77	0.59	0.60	0.61	a_{30}	0.89	0.77	0.61	0.57	0.59
a_{11}				✓		a_{31}	✓				
a_{12}					✓	a_{32}			✓		
a_{13}	0.62	0.61	0.77	0.77	1.00	a_{33}			✓		
a_{14}		✓				a_{34}	✓				
a_{15}				✓		a_{35}	✓				
a_{16}				✓		a_{36}	✓				
a_{17}				✓		a_{37}	✓				
a_{18}	0.76	0.73	0.62	0.57	0.77	a_{38}	✓				
a_{19}	0.68	0.62	0.77	0.77	1.00	a_{39}	✓				

definition of category limits (profiles) is avoided. However, contrarily to other approaches that define categories on the basis of exemplary alternatives (e.g., [1]), we use the outranking relation in an asymmetrical way, rather than being interested in an indifference relation.

Avoiding the definition of category limits may be considered as an advantage of this procedure, as inferring the category limits is not an easy task without imposing a few simplifications [16]. In any case, we are indirectly defining multiple limits. In ELECTRE TRI (pessimistic variant), an alternative is sorted into a category if it outranks its lower limit and does not outrank its upper limit, with lower and upper limits defined a priori. In this paper, an alternative is sorted into a category if it is not outranked by a lower limit and does not outrank an upper limit, with lower and upper limits being any alternative already sorted by the DM in worse or better categories, respectively.

Both the procedure proposed here and ELECTRE TRI have the common characteristic of not sorting an alternative into a precise category if that alternative shows to be atypical. In ELECTRE TRI this is reflected in different results between the pessimistic and optimistic variants, which appear when the alternative to be sorted is incomparable to one or more limits. In this paper, the procedure in Sect. 3 may also result in $C_{min}(a_i) \neq C_{max}(a_i)$, if the alternative is incomparable to all the examples placed in a given category. However, if the DM then decides to consider that alternative as an example for one of the categories in $[C_{min}(a_i), C_{max}(a_i)]$, then subsequent similar alternatives will use that example to justify a more precise classification. As the number of sorted alternatives increases, the chance of

Table 15 Minimum cutting levels $\lambda_{i,h}^*$ at iteration 6 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}				✓	
a_1				✓		a_{21}		✓			
a_2					✓	a_{22}	0.63	0.63	0.65	0.65	0.73
a_3					✓	a_{23}			✓		
a_4				✓		a_{24}	✓				
a_5					✓	a_{25}	1.00	0.63	0.63	0.67	0.95
a_6					✓	a_{26}		✓			
a_7					✓	a_{27}		✓			
a_8				✓		a_{28}	✓				
a_9					✓	a_{29}					✓
a_{10}	0.77	0.77	0.69	0.68	0.64	a_{30}				✓	
a_{11}				✓		a_{31}	✓				
a_{12}				✓		a_{32}			✓		
a_{13}	0.68	0.63	0.77	0.77	1.00	a_{33}			✓		
a_{14}		✓				a_{34}	✓				
a_{15}				✓		a_{35}	✓				
a_{16}				✓		a_{36}	✓				
a_{17}				✓		a_{37}	✓				
a_{18}				✓		a_{38}	✓				
a_{19}	0.71	0.64	0.77	0.77	1.00	a_{39}	✓				

finding an atypical alternative decreases. The downside is that as the number of examples increases, so does the chance of being inconsistent.

Since the examples can be fictitious vectors of performances invented by the DM (as usually are the limits in ELECTRE TRI), our paper can be interpreted as proposing an extension of ELECTRE TRI to allow multiple limits (probably incomparable ones) separating the categories. The main difference would then be the fact that the condition of outranking the lower limit is replaced by the condition of not being outranked by a lower limit. This use of this approach would place each alternative in the lowest category allowed by the principle (5) (pessimistic variant), or in the highest category possible (optimistic variant). If the principle (5) is to be strictly followed, as sustained in this paper, then it must apply to all the alternatives that are sorted by the DM. Nevertheless, the DM can select not to consider all alternatives as examples, i.e., not to sort all alternatives to precise categories. This means that the set A^* would strictly follow the principle (5), serving as prototypes to constrain the sorting of the remaining alternatives $a_i \in A \backslash A^*$ to an interval $[C_{min}(a_i), C_{max}(a_i)]$ ($C_{min}(a_i)$ being the result of a pessimistic procedure, and $C_{max}(a_i)$ being the result of an optimistic procedure).

Comparing the aggregation/disaggregation approach proposed here with the one proposed for ELECTRE TRI, another difference is to have only negative outranking (i.e., non-outranking) constraints of the type $\lambda - S(a_p, a_i) \geq \varepsilon$, whereas for ELECTRE TRI we have positive ($\lambda - S(a_p, a_i) \leq 0$) as well as negative outranking constraints. Therefore, in this paper we do not infer a value for the cutting level λ (we know that $\lambda = 1$ would trivially solve almost all of the constraints), but we infer a minimum value for this parameter. This forces the DM

Table 16 Minimum cutting levels $\lambda_{i,h}^*$ at iteration 7 (2nd example)

	C^1	C^2	C^3	C^4	C^5		C^1	C^2	C^3	C^4	C^5
a_0					✓	a_{20}				✓	
a_1				✓		a_{21}		✓			
a_2					✓	a_{22}	0.66	0.65	0.67	0.67	0.73
a_3					✓	a_{23}			✓		
a_4				✓		a_{24}	✓				
a_5					✓	a_{25}	1.00	0.65	0.65	0.67	0.95
a_6					✓	a_{26}		✓			
a_7					✓	a_{27}		✓			
a_8				✓		a_{28}	✓				
a_9					✓	a_{29}					✓
a_{10}					✓	a_{30}				✓	
a_{11}				✓		a_{31}	✓				
a_{12}					✓	a_{32}			✓		
a_{13}		✓				a_{33}			✓		
a_{14}		✓				a_{34}	✓				
a_{15}				✓		a_{35}	✓				
a_{16}				✓		a_{36}	✓				
a_{17}				✓		a_{37}	✓				
a_{18}				✓		a_{38}	✓				
a_{19}		✓				a_{39}	✓				

to deal explicitly with the meaning of the cutting level when deciding in which categories may each alternative be sorted into, which may be seen as the price to pay in this approach to avoid the definition of category limits.

Finding appropriate ways of incorporating positive outranking constraints in this approach is an interesting issue to research in the future. Future research is also needed to evaluate the implications of the number and choice of examples in the results of the procedure. This involves not only to continue experimenting with examples such as the ones presented here, but also possibly performing simulation studies.

References

1. Belacel, N.: Multicriteria assignment method PROAFTN: methodology and medical application. European J. Oper. Res. **125**(1), 175–183 (2000)
2. Bilgin, S., Köksalan, M., Mousseau, V., Özpeynirci, O.: A new outranking-based approach for assigning alternatives to ordered classes. Communication to the 18th International Conference on Multiple Criteria Decision Making. pp. 19–23. Chania, Greece (2006)
3. Dias, L.C., Mousseau, V., Figueira, J., Clímaco, J.N.: An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI. European J. Oper. Res. **138**(2), 332–348 (2002)
4. Dimitras, A., Zopounidis, C., Hurson, C.: A multicriteria decision aid method for the assessment of business failure risk. Found. Comput. Decision Sci. **20**(2), 99–112 (1995)

5. Doumpos, M., Zopounidis, C.: A multicriteria classification approach based on pairwise comparisons. European J. Oper. Res. **158**, 378–389 (2004)
6. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. European J. Oper. Res. **138**, 247–259 (2002)
7. Hurson, C., Zopounidis, C.: On the use of multicriteria decision aid methods to portfolio selection. In: Climaco, J.N. (ed.) Multicriteria Analysis., pp. 496–507. Springer-Verlag, Berlin (1997)
8. Jacquet-Lagrèze, E., Siskos, Y.: Preference disaggregation: 20 years of MCDA experience. European J. Oper. Res. **130**(2), 233–245 (2001)
9. Koksalan, M., Ulu, C.: An interactive approach for placing alternatives in preference classes. European J. Oper. Res. **144**, 429–439 (2003)
10. Larichev, O.I., Moskovich, H.M.: An approach to ordinal classification problems. Int. Trans. Oper. Res. **1**(3), 375–385 (1994)
11. Massaglia, R., Ostanello, A.: N-tomic: a support system for multicriteria segmentation problems. In: Korhonen, P., Lewandowski, A., Wallenius, J. (eds.) Multiple Criteria Decision Support., pp. 167–174. Springer Verlag, LNEMS 356 (1991)
12. Moscarola, J., Roy, B.: Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiples. RAIRO Oper. Res. **11**(2), 145–173 (1977)
13. Mousseau, V., Dias, L.C.: Valued outranking relations in ELECTRE providing manageable disaggregation procedures. European J. Oper. Res. **156**(2), 467–482 (2003)
14. Mousseau, V., Slowinski, R.: Inferring an ELECTRE TRI model from assignment examples. J. Global Optim. **12**(2), 157–174 (1998)
15. Mousseau, V., Slowinski, R., Zielniewicz, P.: A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support. Comput. Oper. Res. **27**(7–8), 757–777 (2000)
16. Ngo The, A., Mousseau, V.: Using assignment examples to infer category limits for the ELECTRE TRI method. J. Multi-Criteria Decis. Anal. **11**(1), 29–43 (2002)
17. Perny, P.: Multicriteria filtering methods based on concordance/non-discordance principles. Ann. Oper. Res. **80**, 137–167 (1998)
18. Roy, B.: 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 (1978)
19. Roy, B., Bouyssou, D.: Aide Multicritère à la Décision: Méthodes et Cas. Economica, Paris (1993)
20. Yu, W.: Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications. PhD thesis, LAMSADE, Université Paris Dauphine (1992)
21. Zopounidis, C., Zanakis, S., Doumpos, M.: Multicriteria preference disaggregation for classification problems with an application to global investing risk. Decis. Sci. **32**(2), 333–385 (2001)