



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

European Journal of Operational Research 153 (2004) 271–289

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/dsw

Using ELECTRE TRI outranking method to sort MOMILP nondominated solutions

Rui Pedro Lourenço *, João Paulo Costa

*Faculty of Economics, University of Coimbra, Av. Dias da Silva 165, 3004-512 Coimbra, Portugal
INESC, R. Antero de Quental 199, 3000-033 Coimbra, Portugal*

Abstract

Several interactive methods exist to identify nondominated solutions in a Multiple Objective Mixed Integer Linear Program. But what if the Decision Maker is also interested in sorting those solutions (assigning them to pre-established ordinal categories)? We propose an interactive “branch-and-bound like” technique to progressively build the nondominated set, combined with ELECTRE TRI method (Pessimistic procedure) to sort identified nondominated solutions. A disaggregation approach is considered in order to avoid direct definition of all ELECTRE TRI preference parameters. Weight-importance coefficients are inferred and category reference profiles are determined based on assignment examples provided by the Decision Maker. A computation tool was developed with a twofold purpose: support the Decision Maker involved in a decision process and provide a test bed for research purposes.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Sorting; Multiple Objective Mixed Integer Linear Program

1. Introduction

Considering the growing complexity of today’s decision processes, it is more and more essential to support Decision Makers (DMs). The keyword is “support”: it is not about replacing the DM but it is about helping him/her in every phase of the decision process.

This support starts with the initial analysis and structuring phase and may lead to a mathematical

model defining admissible values for the decision variables (expressed through constraints) and identifying the relevant, usually multiple and conflicting, criteria (expressed through objective functions). Solutions to the problem are implicitly defined by the model and may be made explicit afterwards using some exploratory method. This kind of models is usually referred to as Multiple Objective Programming. In the scope of this paper we are particularly interested in those models where decision variables may assume integer or continuous values and both constraints and objective functions are linear: Multiple Objective Mixed Integer Linear Programming (MOMILP) problems. Benson (1995), for instance, proposes to

* Corresponding author. Tel.: +351-239790584; fax: +351-239403511.

E-mail address: ruiloure@sonata.fe.uc.pt (R.P. Lourenço).

address the weekly surgery scheduling problem in a hospital using a model of this type (in this specific example all variables are integer).

Once such a model is established, support should focus on assisting the DM to explore it in the context of a specific decision problematic or problem statement. In our work this means interactively identify the solutions resulting from a MOMILP problem and assign them to a set of ordinal categories. Among all solutions we are only interested in those for which no other solution exists with, at least, equal performance in all objective functions and better performance in, at least, one objective function. These are the *non-dominated* solutions and they represent, in the scope of this paper, the alternatives to consider during the inference process. Once the DM's preference structure is established, it also is possible to classify the remaining MOMILP solutions.

Assigning those alternatives to a set of categories corresponds to one of the three problem statements proposed by Roy (1996) (choice, sorting and ranking). More specifically, since we consider an order structure over the entire set of pre-defined categories, it constitutes what Mousseau et al. (2000) designate by Ordered Multiple Criteria Sorting Problems (Ordered MCSP) and Zopounidis and Doumpos (2002) designate simply by Sorting Problems¹. Among the several existing approaches to this type of problems (again see Mousseau et al., 2000; Zopounidis and Doumpos, 2002), namely using utility functions (for instance UTADIS, Jacquet-Lagrèze, 1995), we opted to use an approach based on the outranking relation, more specifically the ELECTRE TRI method (Yu, 1992; Roy and Bouyssou, 1993). ELECTRE TRI method puts several challenging technical issues because the veto threshold makes its *formulae* nonlinear.

As in most multiple criteria methods, ELECTRE TRI requires many preference parameters. Instead of asking the DM for a direct definition of all parameters, which is often a difficult task, a

preference disaggregation approach (Jacquet-Lagrèze and Siskos, 2001) takes solution assignment examples based on holistic evaluations to infer some of these parameters. Examples of this disaggregation approach applied to the ELECTRE TRI method can be found in the literature (Mousseau and Slowinski, 1998; Ngo The and Mousseau, 2000; Mousseau et al., 2001; Dias et al., 2002) although they all apply to cases where the alternatives are fully identified in advance.

Our work aims at supporting DMs involved in decision processes in which the problem is already structured (as a MOMILP model) and related to the sorting problematic. Since, in these cases, the set of alternatives is defined implicitly by a set of linear constraints, the first step is to gradually present the DM every alternative to consider (which in the particular case of MOMILP problems are represented by nondominated solutions). We use an interactive process to search the non-dominated region of the MOMILP problem, allowing the DM to choose the next solutions. This way the DM is faced with the solutions he/she feels more comfortable to classify. This is the main advantage on integrating both methods: the MOMILP search with the ELECTRE TRI method. As each solution is presented to the DM, he/she is required to classify it (with a provisory status) in an holistic manner. By this way, a set of assignment examples is progressively built reflecting the DM's preferences. These examples will be used to infer some parameters of an ELECTRE TRI type structure underlying the DM's expressed preferences. We are particularly interested in inferring the weight-importance coefficients and determining the categories reference profiles, taking into account the veto mechanism. The preference structure established at a certain point reflects the preferences expressed so far by the DM and will be used to "advise" him/her as he/she will be asked to classify the next solutions the DM chose to classify (through the interactive search method). This aggregation/disaggregation approach requires less cognitive effort from the DM as compared to the direct definition of all parameters. As soon as the DM considers that the inferred ELECTRE TRI type structure reflects his/her preferences (the "advises" for each new classification are consistent

¹ A distinction is made by these authors between classification (categories defined in a nominal way) and sorting (categories defined in an ordinal way).

with his/her preferences), it is possible to stop the interactive process and use the structure to classify the remaining solutions (including the dominated solutions of the MOMILP problem).

The LinearTri algorithm was presented in Lourenço and Costa (2001b) and it was applied to a Surgery Scheduling decision problem using the software tool built to support it (see Lourenço and Costa, 2001a). It is possible by now to identify/confirm some of the algorithm limitations. For instance, the simplifications introduced in the weight-importance coefficients inference program may reject certain assignment examples (alternative/classification) although, according to the current preference structure, they would get precisely the same classification. However, the main limitation of the algorithm lies in the fact that it uses nondominated solutions as category reference profiles. This is not in accordance with the ELECTRE TRI method and seems to make even more difficult the weight inference process.

All considered, there was a need to improve LinearTri: replace the nondominated solutions as category inferior reference profiles and distinguish between inconsistent assignments and the algorithm inability to infer a preference structure. The purpose of this paper is to present the algorithm latest developments together with some reflections about category reference profiles in the context of MOMILP nondominated solutions classification.

A trial-and-error procedure is proposed to determine the reference profiles, leaving some of the effort to the DM him/herself. This procedure was developed under the assumption that it would be better to separate, in two consecutive steps, the inference of weight importance coefficients and the determination of category reference profiles. It is important to start with “good” initial category reference profiles in order to “minimise” the “discrepancies” left to be solved in the next step. Once the weight importance coefficients are inferred, it is necessary again to analyse the results and adjust the reference profiles accordingly. The procedure has been applied to an example problem and the results were presented in Lourenço and Costa (2002).

It is also necessary to distinguish between the limitations of the algorithm and those situations

where the assignments were inconsistent and therefore impossible to reproduce by an ELECTRE TRI structure. We propose to establish some pair wise comparison conditions between alternatives that can be used to identify inconsistent assignment examples (Section 6). Although this approach cannot be used to infer the preference structure, it may be very useful to eliminate doubts about the possibility to infer those structures.

The LinearTri software, developed to support the use of the LinearTri algorithm, as also been improved to incorporate some of this latest developments. These additional test bed features aim to give the DM more freedom when constructing and analysing the preference structure that will reflect his/her assignment examples. It is possible to test different combinations of assignment examples, different values for the reference profiles and combine this with the LinearTri algorithm. It is also a research tool for any Analyst interested in this kind of decision problems.

The next two sections give a brief methodological presentation of Multiple Objective Mixed Integer Linear Programming and the ELECTRE TRI method. Section 4 is dedicated to the LinearTri algorithm, including some remarks about its application and limitations. These limitations were the main source of inspiration for the developments presented in Sections 5 and 6 and, ultimately, to the LinearTri software test bed described in Section 7. Finally some concluding remarks are presented in Section 8.

2. A brief introduction to MOMILP problem formulation

A MOMILP problem can be formulated as

$$\begin{aligned} \max \quad & \{z_1 = \mathbf{c}^1 \mathbf{x}, \dots, z_t = \mathbf{c}^t \mathbf{x}\} \\ \text{s.t.} \quad & \mathbf{x} \in S = \{\mathbf{x} \in R^n \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, x_i \text{ integer}, i \in I\}, \\ & I \subset \{1, \dots, n\}, I \neq \emptyset, \end{aligned} \tag{P1}$$

where t is the number of objective functions (criteria), n is the number of decision variables, \mathbf{A} is a $m \times n$ matrix, \mathbf{b} is a column m -vector and \mathbf{c}^j , $j = 1, \dots, t$, are row n -vectors. Let Z denote the set

of the criterion points (solutions) $\mathbf{z} \in R^t$, i.e., images of $\mathbf{x} \in S$. It is assumed that S is closed and bounded.

Any solution $\bar{\mathbf{z}} \in Z$ is said to be *nondominated* iff there does not exist another $\mathbf{z} \in Z$ such that $\mathbf{z} \geq \bar{\mathbf{z}}$ and $\mathbf{z} \neq \bar{\mathbf{z}}$. Associated with *ideal point* $\mathbf{z}^* \in R^t$ (composed of all objective functions maximum values), there are, at most, t nondominated solutions (one for each function maximum). These may be determined using a linear program derived from the original one (P1). Using an augmented Tchebycheff program it is possible to determine an “extra” nondominated solution, $\mathbf{z}^{\text{at}} \in Z$, that is closest to the ideal point \mathbf{z}^* (Steuer, 1986).

The LinearTri algorithm will use these derived problem formulations to interactively and progressively identify the nondominated solutions associated with the original problem. These solutions represent the alternatives available to the DM, that is, the ones he/she wants to classify. Similarly, the objective functions will act as quantitative criteria for the ELECTRE TRI method described in the next section.

3. Multiple criteria sorting methods

It is possible to distinguish between two large “families” of Multiple Criteria Decision Aiding (MCDA) methods based on their criteria aggregation models (Zopounidis and Doumpos, 2002): the outranking relation and the utility function. Within both “families” it is possible to find not only sorting/classification specific methods but also methods that use a preference disaggregation approach.

The utility function disaggregation approach is used in the UTADIS method (Jacquet-Lagrèze and Siskos, 1982; Jacquet-Lagrèze, 1995). This method defines category lower bounds as utility thresholds and assigns each alternative to the right category by comparing the global utility of the alternative with those thresholds. To estimate the weights necessary to determine the global utility function (disaggregation approach) the UTADIS method uses a linear program.

The ELECTRE TRI method (Yu, 1992; Roy and Bouyssou, 1993) belongs to a group of meth-

ods, called ELECTRE (“ELimination Et Choix Traduisant la REalité”), which have their foundations in the outranking relation as a mean to aggregate different criteria. Among the several preference parameters necessary to this method, the veto threshold is perhaps the most characteristic one: it is a straightforward way to say that if an alternative is “much worse” than another alternative in one criteria, then it cannot be better than that alternative, regardless of all other criteria. The next subsections present the ELECTRE TRI fundamentals and some disaggregation approaches related to it.

3.1. ELECTRE TRI fundamentals

ELECTRE TRI was developed to deal with ordered MCSP. It starts with a set of alternatives, evaluated by quantitative and/or qualitative criteria (g_j), and a set of categories, and uses an outranking relation to assign each alternative to a category. This method requires several preference parameters such as preference (p_j), indifference (q_j) and veto (v_j) thresholds, and weight-importance coefficients $\mathbf{k} = (k_1, \dots, k_t; k_j > 0; j = 1, \dots, t)$. Each outranking relation (aSb) between an alternative a and a category lower reference profile b is established in four stages:

1. criteria concordance indexes $c_j(a, b)$ and global concordance indexes $C(a, b)$ calculation;
2. discordance index $d_j(a, b)$ calculation;
3. credibility degree $\sigma_S(a, b)$ calculation;
4. establishing the outranking relation through the cutting level.

Within the ELECTRE TRI method the categories are defined in an ordered way, and may be represented as C^1, C^2, \dots, C^r ($r \geq 2$), where C^r is the best and C^1 the worst category. Each category C^i ($1 \leq i \leq r$) is limited by two reference profiles (b^i —upper and b^{i-1} —lower reference profile), where the upper reference profile of a category corresponds to the lower reference profile of the category immediately above.

The ELECTRE TRI Pessimistic process (used in the LinearTri Algorithm) aims to assign each alternative to the highest category for which that

alternative outranks the category inferior reference profile. The followed procedure is:

1. Compare successively alternative a with reference profile b^i , with i varying from $r - 1$ to q_j (from higher to lower categories);
2. Let b^p be the first reference profile for which (aSb^p) ; then a is assigned to category C^{p+1} .

In our work we consider all thresholds to be fixed for each criteria, so they are simply noted q_j (indifference threshold for criteria j), p_j (preference threshold for criteria j) and v_j (veto threshold for criteria j). Furthermore, the performance of alternative a and reference profile b , respectively $g_j(a)$ and $g_j(b)$, will simply be noted a_j and b_j .

3.2. ELECTRE TRI disaggregation approaches

Contrary to the “traditional way” of decision-making (define the object of the decision, determine the problematic, define a consistent family of criteria, develop a global preference model and finally exploit the model to support a decision), the disaggregation approach takes a final decision and aims to infer the DM’s preference model that “best” matches the decision. This type of approach has been applied widely in MCDA (Jacquet-Lagrèze and Siskos, 2001) and particularly to the ELECTRE TRI method.

Mousseau and Slowinski (1998) describe an algorithm that starts from a set of assignment examples produced by the DM and tries to determine a model (preference structure) that best matches those examples. The inferred parameters are the reference profiles, the weight-importance coefficients and the preference and indifference thresholds. Even without inferring veto thresholds this approach leads to a nonlinear programming problem. In Mousseau et al. (2001) it was only considered the sub-problem of the determination of the weight-importance coefficients (the thresholds and reference profiles being fixed and without veto) thus leading to a linear programming problem. A complementary work can be found in Ngo The and Mousseau (2000), in which the importance coefficients are fixed and the inferred parameters are the category limits (excluding the veto

threshold). Finally, Dias et al. (2002) integrates the disaggregation approach with robustness analysis combining them in an interactive method.

Contrary to these previous authors, we aim to integrate the initial phase of nondominated solutions identification with the preference parameters inference (weight-importance coefficients and determination of categories reference profiles), thus not starting with an initial complete set of assignment examples. We use an interactive process to search the nondominated region of the MOMILP problem, allowing the DM to choose the next solutions. This way the DM is faced with the solutions he/she feels more comfortable to classify. This is the main advantage on integrating both methods: the MOMILP search with the ELECTRE TRI method. Although we do not infer the veto threshold, we consider it during the inference process: this is particularly important since this is one of the most distinguishing characteristics of the ELECTRE TRI method. Finally, we propose to use a linear inference program combined with an interactive procedure leaving some of the effort to the DM him/herself.

4. LinearTri algorithm

Having presented the MOMILP considerations and the ELECTRE TRI method, this section will present the LinearTri algorithm, which combines elements from both approaches. Consequently, the terms *alternativenondominated solution* and *criticalobjective function* will be used indistinctly.

The LinearTri algorithm is based on an algorithm presented by Marcotte and Soland (1986) and afterwards modified by Durso (1992).

Once the MOMILP problem is formulated and the remaining (not inferred) parameters are set, the LinearTri algorithm begins with the creation of an initial node representing the entire problem. Then it uses a branch-and-bound technique to build an enumeration tree where each node represents a sub-problem of the original problem and, consequently, a subset of nondominated solutions.

For any generic node (including the first one) the algorithm calculates $t + 1$ (at most) nondominated solutions, resulting from the programs

referred to in Section 2. The DM must then choose one of these solutions and classify it taking into consideration the classification “proposed” by the previously established ELECTRE TRI type structure. The algorithm will then try to infer a new set of preference parameters that accommodates this new example of the DM’s preferences.

The chosen solution is compared with the node ideal point: for each objective function where the difference is considered “significant” a new node is created. These new nodes are characterised by an additional constraint (to the programming problem represented by the parent node) defining a new lower limit on the objective function for which the node was created. This branching mechanism allows for the complete identification of the original problem nondominated solutions. Among all nodes created but not yet analysed, the DM must choose the one to proceed (the others will be analysed afterwards). The algorithm ends when all nodes are created and analysed, or at any time the DM so wishes.

The algorithm has the scheme presented in Fig. 1, with the most important steps detailed in the following sections.

4.1. Calculation of nondominated “characteristic” solutions: The alternatives

Any generic node \mathbf{n}^h from the enumeration tree (representing a sub-problem) is characterised by:

- a lower limit vector $\mathbf{L}^h \in R^t$ for the objective functions and an ideal point \mathbf{z}^{*h} .
- a subset of feasible (Z^h) and nondominated (N^h) solutions ($N^h \subseteq Z^h$).

For each node \mathbf{n}^h the ideal point \mathbf{z}^{*h} is calculated simultaneously with t (at most) nondominated solutions associated with it. These are “characteristic” solutions since they “represent” the entire subset of nondominated solutions associated with the node and show the maximum reachable value for each objective function. Another “characteristic” nondominated solution (Tchebycheff solution) results from an augmented unweighted Tchebycheff problem and represents a “balance” between all objective functions. In conclusion,

$t + 1$ nondominated solutions are calculated and presented to the DM in each node \mathbf{n}^h . Fig. 2 presents a screen from the LinearTri software tool, with four² nondominated “characteristic” solutions associated with node \mathbf{n}^4 .

The left side of the screen shows the enumeration tree representing the hierarchy of sub-problems being defined along the LinearTri algorithm. On the right we can see which is the current node and some information about it, particularly the node ideal point that will determine which new nodes will be created once a nondominated solution is chosen by the DM. The purpose of the grid on the right side is to present to the DM the $t + 1$ nondominated solutions for him/her to choose from. The chosen solution is afterwards classified and will be another example of the DM’s preferences.

4.2. Testing the selected solution against the ideal point and creation of new nodes

Having selected and classified a nondominated solution, the question is “How far is that solution from the ideal point determined for the respective node?” or, to put it in another way, “Is it possible/desirable to improve any of the objective functions?”. That depends on Delta values defined, in the beginning of the algorithm, for each objective function. These values will determine which nodes are created from the current one. Generally, they determine the granularity of solutions found.

4.3. Inference of weight-importance coefficients

In the LinearTri algorithm the weight-importance coefficients are inferred through a linear programming problem whose formulation will be presented below.

According to the ELECTRE TRI Pessimistic assignment procedure (see Yu, 1992, for details), in a preference structure with r categories (C^1, \dots, C^r) inferior limited by reference profiles b^0, \dots, b^{r-1} respectively, where C^r represents the highest

² This example has three objective functions.

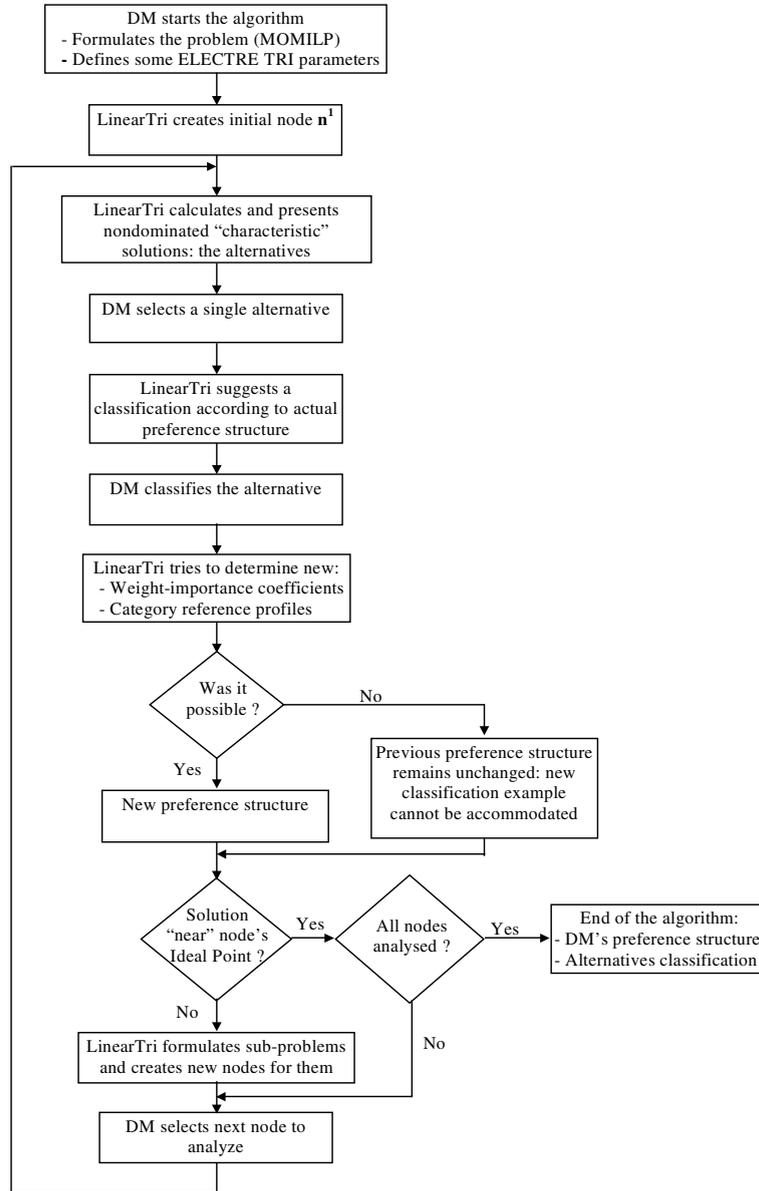


Fig. 1. LinearTri scheme for identification and classification of nondominated solutions in MOMILP.

category, each solution a assigned to category i ($1 \leq i \leq r$) must respect the following propositions:

(I) NOT (aSb^{m-1}) , for $i < m \leq r$

A solution cannot outrank any of the inferior reference profile representing the categories higher than the category to which it is assigned.

(II) (aSb^{i-1})

A solution must outrank the inferior reference profile representing the category to which it is assigned.

By respecting the above propositions, solution a will be assigned to category i since, as we run down

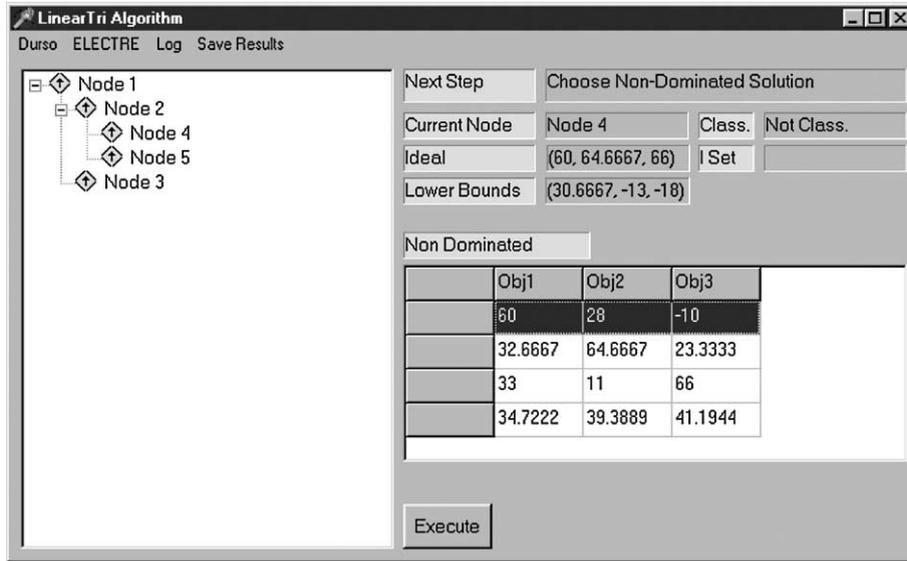


Fig. 2. LinearTri software; nondominated “characteristic” solutions presentation.

the reference profiles in descending order b^{r-1}, \dots, b^0 , reference profile b^{i-1} is the first one for which (aSb^{i-1}) .

Each proposition type (I and II) will give origin to functional constraints to be incorporated in the different linear programming problems used to infer the weight-importance coefficients. In order to work only with linear constraints, it is necessary to simplify them without compromising the end results.

As to proposition type I, we begin by ensuring that

$$\bar{F}(a, b) = \{j \in F | d_j(a, b) > C(a, b)\} \neq \emptyset \quad (4.1)$$

that is, considering F as the set of the indices of the objective functions ($F = \{1, \dots, t\}$), there is at least one criteria for which the discordance index $d_j(a, b)$ is greater than the global concordance index $C(a, b)$.

Consider $D_M(a, b) = \text{Max}_{j \in F} d_j(a, b)$. Then, in order to have at least one discordance index inferior to the global concordance index ($\#\bar{F}(a, b) \geq 1$), the *first constraint* is

$$C(a, b) < D_M. \quad (4.2)$$

Under this condition, the calculation of the credibility degrees $\sigma_S(a, b)$ necessary to establish the

outranking relation (aSb) gives origin to the constraint

$$\sigma_S(a, b) = C(a, b) * \prod_{j \in \bar{F}} \frac{1 - d_j(a, b)}{1 - C(a, b)} < \lambda. \quad (4.3)$$

Since we aim to solve linear programming problems it is necessary to make this constraint linear on k_j (this variable appears on the expression of $C(a, b)$). Consider $\bar{t} = \#\bar{F}(a, b)$. Since we do not determine \bar{t} , the constraint (4.3) is simplified considering that $t \geq \bar{t}$. Taking also in consideration that $0 \leq C(a, b) \leq 1$, it is sufficient to assure that

$$\frac{\prod_{j \in \bar{F}} [1 - d_j(a, b)]}{[1 - C(a, b)]^t} < \lambda. \quad (4.4)$$

Since we can only be sure for one objective function that $d_j(a, b) > C(a, b)$, it is sufficient to assure that

$$\frac{1 - D_M}{[1 - C(a, b)]^t} < \lambda. \quad (4.5)$$

In conclusion, the *second constraint* becomes

$$C(a, b) < 1 - \sqrt[t]{\frac{1 - D_M}{\lambda}}. \quad (4.6)$$

From constraints (4.2) and (4.6) it will result a single constraint of type

$$C(a, b) < \text{Min} \left\{ D_M, 1 - \sqrt[1]{\frac{1 - D_M}{\lambda}} \right\}. \quad (4.7)$$

In a similar way, from proposition type II we arrive to another constraint:

$$C(a, b) \geq \text{Max} \{ D_M, \lambda \}. \quad (4.8)$$

The linear programming problem, which constitutes the core of the inference procedure, will take constraints of type (4.7) and (4.8) as functional constraints. It will be used to validate a certain preference structure by inferring the corresponding importance coefficients, and has the following generic formulation.

The goal is to minimise the increment ε_j (always nonnegative) of the importance coefficients, starting from its previous value k_j . The linear programming problem to be solved to determine the increments ε_j of the importance coefficients k_j , for a certain preference structure, can then be stated as

$$\text{Min} \quad \sum_j \varepsilon_j \quad (P2)$$

s. t.

$$\sum_j \varepsilon_j (c_j - M_{\text{in}}) < \sum_j k_j (M_{\text{in}} - c_j), \quad (I)$$

$$\sum_j \varepsilon_j (c_j - M_{\text{ax}}) \geq \sum_j k_j (M_{\text{ax}} - c_j), \quad (II)$$

$$\varepsilon_j \geq 0, \quad \forall_j,$$

where: $c_j = c_j(a, b)$, $D_M = \text{Max} d_j(a, b)$, $\forall_j \in F$, $M_{\text{in}} = \text{Min} \left\{ 1 - \sqrt[1]{\frac{1 - D_M}{\lambda}}, \lambda \right\}$, $M_{\text{ax}} = \text{Max} \{ D_M, \lambda \}$;

- (I) for all classified solutions—including the category reference ones— a , for each inferior reference profile b of the upper categories;
- (II) for all classified solutions a , for the inferior reference profile b of the corresponding category.

4.4. Determination of category reference profiles

In the LinearTri algorithm reference profiles are not inferred. Instead, a simple heuristic was used to choose, among the nondominated solutions classified in a certain category, the one that will represent the category as reference profile.

According to this heuristic, if a new solution is the first one to be classified in a particular category, then there is no doubt that this solution should be the reference profile for that category. On the other hand, if that category has already a reference profile, it is necessary to know which of the two solutions is the best one to act as reference profile. Two alternative structures are in analysis and, if it is possible to infer the importance coefficients for both structures, the selected structure will be the one that implies a smaller variation of the original importance coefficients.

4.5. Final remarks

The LinearTri algorithm has already been applied to a decision problem (see Lourenço and Costa, 2001a) allowing for the identification and classification of alternatives and, at the same time, making explicit the DM's preference structure. The problem was to find efficient weekly surgery schedules in a hospital. To address it, Benson (1995) proposed a Multiple Objective Integer Programming model which required the DM to directly establish different sets of weights for the objective functions. For each set it was then possible to determine one or several efficient scheduling solutions. The use of LinearTri avoided the need for direct elicitation of the relative importance of the different types of surgery (as proposed in Benson, 1995). Instead, through the holistic classification of some schedule examples, the algorithm was able to infer the weight-importance coefficients of an ELECTRE TRI type structure and establish reference solutions for the categories proposed by the DM.

It is possible by now to identify/confirm some of the algorithm limitations. For instance, the simplifications introduced in the weight-importance coefficients inference program (described in Section 4.3) to make the functional restrictions linear, may block out some solutions. This means that a certain assignment example (alternative/classification) may not be included in the final DM's preference structure (because it was not possible to infer a set of weights to accommodate it) although, according to the same structure, it would get precisely the same classification. However, the main

limitation of the algorithm lies in the fact that it uses nondominated solutions as category reference profiles. This is not in accordance with the ELECTRE TRI method. Furthermore, it seems to make even more difficult the weight inference process.

All considered, there was a need to improve LinearTri and replace the nondominated solutions as category inferior reference profiles and to distinguish between inconsistent assignments and the algorithm inability to infer a preference structure. The resulting developments are presented in the next sections.

5. A trial-and-error procedure to determine reference profiles

Using nondominated solutions as category reference profiles was a major drawback of the LinearTri algorithm. Not only because it was not in full compliance with ELECTRE TRI but also because it seemed to make the weight inference process much harder. This led to the development of a trial-and-error procedure with two separate but consecutive major steps: the determination of reference profiles and the inference of weight-importance coefficients. Each step begins with the analysis of the previous step results and tries to solve the remaining “issues”.

A simple scheme of the procedure is presented in Fig. 3. The next subsections will give more details about each step of the procedure.

5.1. Establish initial category reference profiles

The first step of the procedure consists of finding an initial set of values for the category reference profiles. Ideally each component (associated with each criteria) of each reference profile (associated with each category) would be greater than the values of all alternatives in the lower categories and lower than the values of all alternatives classified in the category itself or in categories above. That is not possible when the alternatives are represented by nondominated solutions. The goal is then to determine for each b_j^{i-1} (category i in-

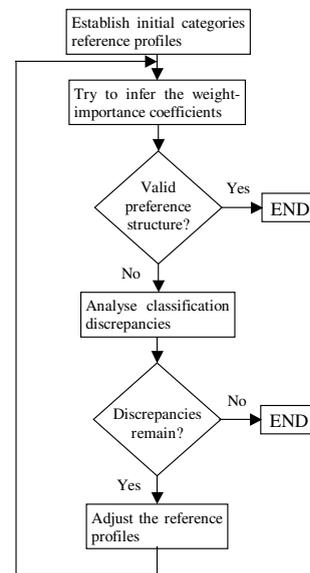


Fig. 3. Simple scheme of a trial-and-error procedure to determine reference profiles.

ferior reference profile, for criteria j) a value that minimise “classification discrepancies”. Those discrepancies occur when:

- an alternative is classified in a category below category i , but it has a performance in criteria j greater than b_j^{i-1} , or
- an alternative is classified in or above category i , but it has a performance in criteria j lower than b_j^{i-1} .

Additionally, some “measures” of the discrepancy are considered: how many categories away is the “discrepant alternative” from the category being considered, and the ratio number of “discrepant alternatives”/total number of alternatives in a category.

The value of each component— b_j^{i-1} (category C^i inferior reference profile, criteria j)—is calculated independently, minimising the following expression:

$$\sum |a_j^k - b_j^{i-1}| / N \times Q \tag{5.1}$$

- for all alternatives a^k classified in categories below C^i , for which $a_j^k > b_j^{i-1}$;

- for all alternatives a^k classified in category C^i or above, for which $a_j^k < b_j^{i-1}$;
 where: N = number of alternatives classified in the same category as a^k ; Q = “distance” (measured by the number of categories between a^k category and category C^i).

All profiles determined through the expression above must also respect the following rules:

- $\forall_j \forall_i b_j^i \geq b_j^{i-1}$ ($j = 1, \dots, t$; $i = 1, \dots, r$),
- $\forall_j b_j^{i-1} < a_j^k + v_j$ ($j = 1, \dots, t$), for each alternative— a^k —classified in category C^i ,

where t represents the number of criteria and r the number of categories.

5.2. Try to infer the weight-importance coefficients

Once a set of reference profiles is determined, it is necessary to infer a set of weight-importance coefficients. A new approach was also developed for this part of the procedure. The starting point is the same as it was for the LinearTri algorithm (see Section 4.3) where each solution a assigned to category i ($1 \leq i \leq r$) must respect the following propositions:

- (I) NOT (aSb^{m-1}), for $i < m \leq r$,
- (II) (aSb^{i-1}).

Again, each proposition type (I and II) will give origin to functional constraints to be incorporated in the linear programming problem used to infer the weight-importance coefficients.

As to proposition type I, that is, solution a is not assigned to a category whose inferior reference profile is b , it is sufficient that

$$C(a, b) < \lambda$$

$$\vee$$

$$\exists_j d_j(a, b) > C(a, b) \wedge C(a, b) \times \frac{(1 - d_j(a, b))}{(1 - C(a, b))} < \lambda,$$

$$1 \geq j \geq t. \tag{5.2}$$

Solving in order to $C(a, b)$ yields

$$\exists_j C(a, b) < d_j(a, b) \wedge C(a, b) < \frac{\lambda}{1 - d_j(a, b) + \lambda}, \quad 1 \geq j \geq t, \tag{5.3}$$

or, in another way,

$$\exists_j C(a, b) < \text{Min} \left\{ d_j(a, b); \frac{\lambda}{1 - d_j(a, b) + \lambda} \right\},$$

$$1 \geq j \geq t. \tag{5.4}$$

The complete condition is then

$$C(a, b) < \text{Max} \left\{ \lambda; \text{Min} \left\{ d_j(a, b); \frac{\lambda}{1 - d_j(a, b) + \lambda} \right\} \right\},$$

$$1 \geq j \geq t. \tag{5.5}$$

Likewise, to make a proposition type II valid, that is, solution a is assigned to a category whose inferior reference profile is b , it is necessary that

$$C(a, b) \geq \lambda. \tag{5.6}$$

To simplify the definition of $C(a, b)$ it was considered that $\sum_{j=1}^t k_j = 1$, which leads to

$$C(a, b) = \sum_{j=1}^t k_j \times c_j(a, b). \tag{5.7}$$

Using the set of reference profiles determined in the previous step, we start out by calculating all local concordance and discordance indexes. It is then possible to infer a corresponding set of weight-importance coefficients— k_j —using the following linear program:

$$\text{Max} \quad \sum_{j=1}^t k_j \tag{P3}$$

s.t.

$$\sum_{j=1}^t k_j \times c_j(a, b) < \text{Max} \{ \lambda; m_1; \dots; m_t \}, \tag{I}$$

$$\sum_{j=1}^t k_j \times c_j(a, b) \geq \lambda, \tag{II}$$

$$\sum_{j=1}^t k_j = 1,$$

$$k_j \geq 0,$$

where

$$m_j = \text{Min} \left\{ d_j(a, b); \frac{\lambda}{1 - d_j(a, b) + \lambda} \right\}, \quad 1 \geq j \geq t,$$

- (I) for all classified solutions—including the category reference ones— a , for each inferior reference profile b of the upper categories;
- (II) for all classified solutions a , for the inferior reference profile b of the corresponding category.

The main difference between this linear program and the one presented in Section 4.3 (P2) is that the later program uses necessary conditions as constraints. This means that even if it is possible to find a solution for the linear program (a set of weights) now it is not guaranteed that the corresponding preference structure is valid. On the other hand, using necessary instead of sufficient conditions makes it possible to reach more solutions (set of weights) that were blocked out in the initial LinearTri algorithm.

Since this is a trial-and-error procedure, if the linear program is impossible it does not mean the end of the procedure. Instead, the weight-importance coefficients, valued up to the point where the impossibility was detected, are used in the next step. It is always necessary to validate the resulting preference structure and analyse the corresponding weight-importance coefficients.

5.3. Validate the resulting preference structure

Once a set of weight-importance coefficients is determined, it is always necessary to check if the resulting preference structure is valid. This means that each classification example provided by the DM is matched against the classification obtained using the inferred preference structure and the ELECTRE TRI pessimistic assignment procedure. If all classifications match, then the preference structure really reflects the DM's preferences and the procedure ends. Otherwise it is necessary to check where are the discrepancies (alternatives assigned to different categories) and what is the dimension of those discrepancies (outranking relation credibility degree obtained *versus* necessary).

5.4. Analyse classification discrepancies

This analysis step starts by calculating all global concordance indexes— $C(a, b)$ —and outranking

relation credibility degrees— $\sigma_S(a, b)$ —for each alternative with regard to the relevant reference profiles. It is possible to detect which alternatives do not fit the inferred structure (discrepancy) just by observing the obtained values:

- if $\sigma_S(a, b) < \lambda$ when it is intended to be (aSb) ;
- if $\sigma_S(a, b) \geq \lambda$ when it is intended to be NOT (aSb) .

The degree of each discrepancy is given by expression $\Delta\sigma_S = |\lambda - \sigma_S(a, b)|$.

Once the discrepancies are detected and analysed, the DM may reconsider some classifications. It is possible that, by this action, all discrepancies disappear and the search ends. If some discrepancies remain, it is necessary to adjust the reference profiles and try again.

5.5. Adjust reference profiles

Taking into consideration the information resulting from the analysis of classification discrepancies it is now necessary to adjust the relevant reference profiles. In order to better control the effect of the adjustment process, only one reference profile component— b_j^{i-1} —is adjusted on each iteration. The first task is to choose which profile and then which component to adjust.

The idea is to start by eliminating the biggest discrepancy, which means the reference profile to adjust is the one with a greater degree of discrepancy— $\Delta\sigma_S$. Consider b to be that profile. As to the component of b to adjust (increase or decrease) in order to eliminate the discrepancy ($\Delta\sigma_S \rightarrow 0$), it is necessary to analyse each local concordance and discordance index. It is intended that the adjustment made would contribute to one of two situations (depending on the type of discrepancy):

- (aSb) . In this case b_j should diminish to $b_j = p_j - \lambda(p_j - q_j) + a_j$.
- NOT (aSb) . In this case b_j should increase to $b_j = p_j - \lambda(p_j - q_j) + a_j$.

The idea behind these values is that a criteria “strongly” contributes to the establishment of an

outranking if its local concordance index is greater than the cutting level threshold.

Once adjusted, the new proposed values for each component of b would have to comply with the global rules mentioned before:

- $\forall_j \forall_i b_j^i \geq b_j^{i-1}$ ($j = 1, \dots, t$; $i = 1, \dots, r$),
- $\forall_j b_j^{i-1} < a_j^k + v_j$ ($j = 1, \dots, t$), for each alternative— a^k —classified in category C^i ,

where t represents the number of criteria and r the number of categories.

Finally, considering all components of b , we choose to adjust the one that implies a smaller relative (to the previous value) change. The next step is to go back to the weight-importance coefficients inference stage with the new reference profiles.

5.6. End of the trial-and-error procedure

This procedure ends when the inferred ELECTRE TRI preference structure entirely reflects the DM's preferences, expressed through the assignment examples. This means that no discrepancies occur after the inference of weight-importance coefficients and establishment of reference profiles. That is, it is necessary that all outranking relations established (or not) between the alternatives and the reference profiles agree with the propositions inherent to ELECTRE TRI assignment method.

This is a trial-and-error procedure without guarantee that a “discrepancy free” situation will occur. It is possible that, at some iteration, the DM reconsiders some assignment examples previously given when confronted with the “dimension” of discrepancies detected and accepts the classifications proposed by the procedure. If not, it is up to the DM to decide when to stop the procedure considering that it is not possible to accommodate some assignment examples.

5.7. Final remarks

This trial-and-error procedure was developed under the assumption that it would be better to separate the inference weight importance

coefficients from the determination of category reference profiles. These inference processes are performed in two separate but consecutive steps, where each step begins by analysing the results of the previous step. It is important to start with “good” initial category reference profiles in order to “minimise” the “discrepancies” left to be solved in the next step. Once the weight importance coefficients are inferred, it is necessary again to analyse the results and, if some “discrepancies” remain, to adjust the reference profiles accordingly.

The procedure has been applied to an example problem and the results were presented in Lourenço and Costa (2002). Starting from a set of classification examples provided by the DM it was possible to determine a set of weight-importance coefficients and reference profiles that reflected the DM's preferences through an ELECTRE TRI structure.

6. Inconsistent classifications proposed by the Decision Maker

During the development of the LinearTri algorithm and the trial-and-error procedure a question arose. When it is not possible to infer an ELECTRE TRI structure to match the DM expressed preferences, is this due to the inference method or are the assignment examples incompatible? The following subsection presents a set of necessary conditions that detect incompatibilities among the DM's expressed preferences thus avoiding the effort to find a matching ELECTRE TRI preference structure. A simple example is presented afterwards to better understand those conditions.

6.1. Compatibility conditions

Consider two alternatives, a^1 and a^2 , which were assigned by the DM to two different categories in a problem with t criteria. Alternative a^1 was assigned to a category characterized by inferior reference profile b and a^2 was assigned to a lower category. According to the ELECTRE TRI Pessimistic assignment method:

$$(a^1 S b) \text{ AND NOT } (a^2 S b) \Rightarrow \sigma_S(a^1, b) > \sigma_S(a^2, b). \tag{6.1}$$

Suppose that $\forall_j a_j^1 \leq a_j^2, j = 1, \dots, t$. Then

$$\forall_j c_j(a^1, b) \leq c_j(a^2, b) \wedge d_j(a^1, b) \geq d_j(a^2, b) \Rightarrow \sigma_S(a^1, b) \leq \sigma_S(a^2, b). \tag{6.2}$$

This means that

$$(a^1 S b) \text{ AND NOT } (a^2 S b) \Rightarrow \exists_j c_j(a^1, b) > c_j(a^2, b) \vee d_j(a^1, b) < d_j(a^2, b) \Rightarrow \exists_j a_j^1 > a_j^2. \tag{6.3}$$

Consider then a criteria j for which $a_j^1 > a_j^2$. Table 1 shows a comparison of both alternatives local concordance indexes across the range of possible values for the reference profile, divided into intervals ³.

In conclusion,

$$b_j \in]q_j + a_j^2, p_j + a_j^1[\iff c_j(a^1, b) > c_j(a^2, b).$$

A similar line of reasoning applied to $d_j(a^1, b)$ and $d_j(a^2, b)$ yields

$$b_j \in]p_j + a_j^2, v_j + a_j^1[\iff d_j(a^1, b) < d_j(a^2, b).$$

Finally, considering that $a_j^1 > a_j^2$,

$$b_j \in]q_j + a_j^2, v_j + a_j^1[\iff c_j(a^1, b) > c_j(a^2, b) \vee d_j(a^1, b) < d_j(a^2, b) \tag{6.4}$$

or

$$(a^1 S b) \text{ AND NOT } (a^2 S b) \Rightarrow \exists_j a_j^1 > a_j^2 \wedge b_j \in]q_j + a_j^2, v_j + a_j^1[. \tag{6.5}$$

This later proposition should hold for all combinations of alternatives classified in different categories. Additionally, according to ELECTRE TRI,

$$\forall_j b_j^{i-1} < a_j^k + v_j \quad (j = 1, \dots, t), \text{ for each alternative—}a^k\text{—classified in category } C^i \tag{6.6}$$

and

$$\forall_j \forall_i b_j^i \geq b_j^{i-1} \quad (j = 1, \dots, t; i = 1, \dots, r). \tag{6.7}$$

³ This line of reasoning assumes that $p_j - q_j > a_j^1 - a_j^2$ without any loss of generality.

6.2. Inconsistent classification example

The following example will illustrate how the propositions presented in the last subsection may be used to detect inconsistent classifications.

Consider a problem with three criteria and three categories (C^1, C^2, C^3) defined by three inferior reference profiles (b^0, b^1, b^2 respectively). Table 2 presents the indifference, preference and veto thresholds defined for this problem.

Suppose now that the DM expressed his/her preferences assigning three alternatives to those three categories as presented in Table 3.

Consider alternatives a^2 and a^3 . The DM has assigned a^2 to category C^2 and a^3 to a lower category C^1 . This means that $(a^2 S b^1)$ AND NOT $(a^3 S b^1)$ must hold ⁴ and consequently

$$(a^2 S b^1) \text{ AND NOT } (a^3 S b^1) \Rightarrow \exists_j a_j^2 > a_j^3 \wedge b_j \in]q_j + a_j^3, v_j + a_j^2[, \tag{6.8}$$

$j = 1, 2, 3.$

This means that, if alternative a^2 is to be classified in a category above a^3 category, it is necessary that a^2 is better in at least one criteria. Furthermore, the difference must be such that allows for either $c_j(a^2, b^1) > c_j(a^3, b^1)$ or $d_j(a^2, b^1) < d_j(a^3, b^1)$. Only then it is possible that $C(a^2, b^1) > C(a^3, b^1)$ which finally makes possible that $(a^2 S b^1)$ AND NOT $(a^3 S b^1)$.

Since only criteria 1 meets the requirement that $a_j^2 > a_j^3$, it is necessary that $b_1^1 \in]63, 78[$.

Looking now at the veto conditions, alternative a^1 may be assigned to category C^3 if and only if $(a^1 S b^2)$. This means that $\forall_j b_j^2 < a_j^1 + v_j$ ($j = 1, 2, 3$). Then, for criteria 1, $b_1^2 < 45.6$.

Another general condition from ELECTRE TRI is $\forall_j \forall_i b_j^i \geq b_j^{i-1}$ ($j = 1, 2, 3; i = 1, 2, 3$).

It is now easy to see that $b_1^1 \in]63, 78[\wedge b_1^2 \geq b_1^1 \wedge b_1^2 < 45.6$ is an impossible condition. In conclusion, there is no ELECTRE TRI structure that accommodates these assignment examples: they are inconsistent.

⁴ b^1 is the inferior reference profile of category C^2 .

Table 1
Comparison of local concordance indexes value

Interval	Local concordance indexes value	Comparison
$b_j < q_j + a_j^2$	$c_j(a^1, b) = 1 \wedge c_j(a^2, b) = 1$	$c_j(a^1, b) = c_j(a^2, b)$
$q_j + a_j^2 < b_j < q_j + a_j^1$	$c_j(a^1, b) = 1 \wedge c_j(a^2, b) = \frac{p_j - (b_j - a_j^2)}{p_j - q_j} < 1$	$c_j(a^1, b) > c_j(a^2, b)$
$q_j + a_j^1 < b_j < p_j + a_j^2$	$c_j(a^1, b) = \frac{p_j - (b_j - a_j^1)}{p_j - q_j} \wedge c_j(a^2, b) = \frac{p_j - (b_j - a_j^2)}{p_j - q_j}$	$c_j(a^1, b) > c_j(a^2, b)$
$p_j + a_j^2 < b_j < p_j + a_j^1$	$c_j(a^1, b) = \frac{p_j - (b_j - a_j^1)}{p_j - q_j} \wedge c_j(a^2, b) = 0$	$c_j(a^1, b) > c_j(a^2, b)$
$b_j > p_j + a_j^1$	$c_j(a^1, b) = 0 \wedge c_j(a^2, b) = 0$	$c_j(a^1, b) = c_j(a^2, b)$

Table 2
Indifference, preference and veto thresholds

	Criteria 1	Criteria 2	Criteria 3
Indifference (q)	6	8.77	8.6
Preference (p)	12	17.53	17.2
Veto (v)	18	26.3	25.8

Table 3
DM's assignment examples

	Criteria 1	Criteria 2	Criteria 3	
a^1	27.6	38	63	C^3
a^2	60	28	-10	C^2
a^3	57	33	-6	C^1

6.3. Final remarks

As a first remark, it is necessary to point out that the conditions used are necessary but not sufficient. Even if it is possible to establish a set of intervals for all reference profiles that does not mean it is possible to find an ELECTRE TRI structure that fits the assignment examples.

Although this is a simple process to understand, it may be complex to implement. This is especially true when the number of criteria increases because they increase the number of disjunctive conditions. One way to deal with the growing complexity might be to start by analysing those pairs of alternatives assigned to different categories but “very close” in performance according to the criteria.

It is also noteworthy to point out that the process may be improved considering not

only pair wise alternative analysis but also alternative/profile analysis (besides the veto conditions).

Although this approach cannot be used to infer the preference structure, it may be very useful to eliminate doubts about the capacity of other methods to infer those structures.

7. LinearTri software test bed

A software package was developed to support any DM wanting to use the LinearTri algorithm. The software implements every step of the original algorithm and drives the DM through those steps as depicted in Fig. 1 flowchart. It uses a visual enumeration tree to keep track of all sub-problems being defined and several grids to present the nondominated solutions (alternatives) in an organized way (see Fig. 2).

Afterwards, the software was extended, in order to constitute a test bed where the DM may “toy around” with the problem outside the LinearTri algorithm, and provide a research tool for the Analyst. The extensions made were inspired in the developments presented in Sections 5 and 6. The next subsections will present the most important aspects of the test bed.

7.1. Outranking indexes calculation

A very simple feature of the software package is to allow the DM to calculate all local and global indexes of the outranking relation between any pair of nondominated solutions (alternatives) already identified. An example is presented in Fig. 4.

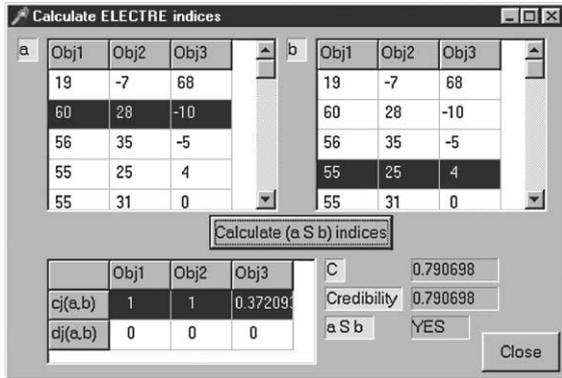


Fig. 4. LinearTri screen to calculate outranking relation indexes.

7.2. Direct manipulation of ELECTRE TRI preference structure

Another useful characteristic of the test bed is that it allows the DM to “manipulate” at will the preference structure determined by the LinearTri algorithm. Fig. 5 shows the interface built for that purpose.

The left side grid shows the current preference structure (classified alternatives and reference profiles). The right side grid presents all nondominated solutions (alternatives) identified but not

included in the preference structure (either because it was not possible to find a set of weights and reference profiles or due to DM’s option). Upon entering this screen, the preference structure is always valid since it comes from the LinearTri algorithm. The DM may then manipulate the structure by removing alternatives, changing alternatives category, inserting alternatives or changing category reference profiles. This last possibility includes the definition of “automatic profiles” according to Section 5.1 or “manual profiles”, that is, profiles with values directly inserted by the DM. This is of particular importance since it makes possible to have preference structures without nondominated solutions as category reference profiles.

Whatever changes are made, the resulting preference structure will only be accepted if it is possible to infer a set of weight importance coefficients that, together with the reference profiles and the preference parameters defined in the beginning of the algorithm, define a preference structure that “matches” (see Section 5.3) the classification examples in the left side grid. This inference process is started through the “Test Structure” button. However, as it was mentioned in Section 5.3, whatever the result of the weights inference process it is always necessary to analyse it.

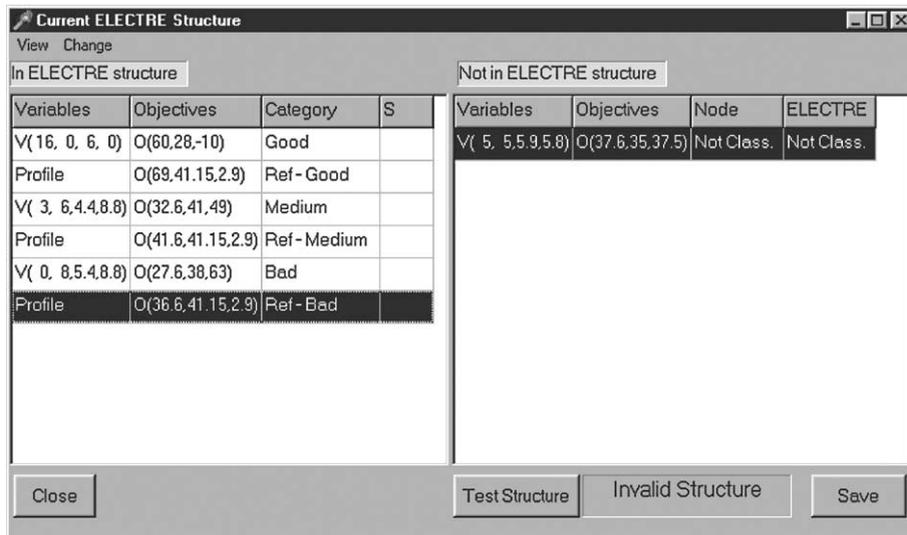


Fig. 5. LinearTri screen to manipulate the preference structure.

Desc.	Objectives	c1	c2	c3	d1	d2	d3	C	Sigma	Delta
a S b	O(60,28,-10)	0.5	0.5	0.5	0	0	0	0.5	0.5	
NOT (a S b)	O(32.6,41,49)	0	1	1	1	0	0	0.5	0	
NOT (a S b)	O(27.6,38,63)	0	1	1	1	0	0	0.5	0	
Good	O(69,41,15,2,9)									
a S b	O(32.6,41,49)	0.5	1	1	0	0	0	0.75	0.75	
NOT (a S b)	O(27.6,38,63)	0	1	1	0.3333	0	0	0.5	0.5	
Medium	O(41.6,41,15,2,9)									
a S b	O(27.6,38,63)	0.5	1	1	0	0	0	0.75	0.75	
Bad	O(36.6,41,15,2,9)									

Close Profile should increase in this objective Correct
 Profile should decrease in this objective

Fig. 6. LinearTri screen to analyse weight-importance coefficients inference.

7.3. Weights inference analysis

The screen in Fig. 6 is used to analyse the results of the weights inference process as it was proposed in Sections 5.3 and 5.4.

This screen shows all outranking relations necessary (or not necessary) to assign each alternative to the correspondent category according to the ELECTRE TRI pessimistic assignment method. All local concordance and discordance indexes, global concordance indexes, and credibility degrees are presented for each relation. The last column presents, if relevant, the degree of discrepancy as defined in Sections 5.4. Additionally, for each discrepancy detected, it also pointed out to the DM what type of adjustment (increase or decrease) should be made on the relevant reference profile to eliminate the discrepancy. These adjustments could be made according to Section 5.5, manually, in the screen of Fig. 5.

7.4. Final remarks

These additional test bed features of the LinearTri software aim to give the DM more freedom when constructing and analysing the preference structure that will reflect his/her assignment examples. It is possible to test different combinations

of assignment examples, different values for the reference profiles and combine this with the LinearTri algorithm. It is also a research tool for any Analyst interested in this kind of decision problems. The developments presented in Sections 5 and 6 inspired these additional features.

8. Conclusion

This paper presents an algorithm developed to help the DM to identify and classify nondominated solutions resulting from MOMILP problems, with a high degree of integration between the procedures of identification and classification, and preference elicitation. The node generation mechanism used to identify the complete nondominated set is based upon the establishment of lower bound values for the objective functions. The ELECTRE TRI method was chosen among the various sorting methods to express the DM's preferences and it was used within an aggregation/disaggregation framework. This approach tries to overcome the natural difficulties to precisely quantify every preference parameter necessary to the ELECTRE TRI method. Instead, the DM is asked to provide assignment examples and the algorithm itself tries to infer those parameters. In the particular case of

the LinearTri algorithm, the DM's ELECTRE TRI preference structure is modified step-by-step, as each new nondominated solution (representing an alternative) is identified and classified.

The weight-importance coefficients are inferred using a linear programming problem and the category reference solutions were established through an heuristic. The presence of a veto mechanism in the ELECTRE methods, although very useful to model the DM's preferences, poses several difficulties when it is necessary to infer the preference parameters. This initial proposal used very strict sufficient conditions that guaranteed, when possible, a set of parameters reflecting the DM's classification examples. However this proved, in some occasions, to block out perfectly good sets of weight-importance coefficients. Also, the use of nondominated solutions as category reference was not in strict accordance with the ELECTRE TRI method.

An attempt was made to overcome these limitations. A trial-and-error procedure was proposed to determine the reference profiles, leaving some of the effort to the DM him/herself. The sufficient conditions were relaxed, thus enabling more sets of weight-importance coefficients to be achieved. As a drawback, there was no more assurance that a set of weight-importance coefficients inferred this way would lead to an ELECTRE TRI structure that fully reflected the DM's expressed preferences. It was then necessary to try and distinguish between the limitations of the algorithm to reproduce the example assignments made by the DM and those situations where the assignments were inconsistent and therefore impossible to reproduce by an ELECTRE TRI structure. We proposed to establish some pair wise comparison conditions between alternatives that can be used to identify inconsistent assignment examples. On a future work we are planning to use this approach in order to know which alternatives should have a different classification to give some support to the DM in order that he/she can better resolve his/her inconsistencies.

The LinearTri software, developed to support the use of the LinearTri algorithm, as also been improved to incorporate some of this latest developments. These additional test bed features aim

to give the DM more freedom when constructing and analysing the preference structure that will reflect his/her assignment examples. It is possible to test different combinations of assignment examples, different values for the reference profiles and combine this with the LinearTri algorithm. It is also a research tool for any Analyst interested in this kind of decision problems.

Acknowledgements

This paper was supported by FEDER, FCT, POCTI/32405/GES/2000.

References

- Benson, H.P., 1995. Scheduling surgeries for patients requiring post-operative intensive care: A multiple objective integer programming approach. In: Pardalos, P.M., Siskos, Y., Zopounidis, C. (Eds.), *Advances in Multicriteria Analysis*. Kluwer Academic Publishers, Dordrecht, pp. 233–247.
- Dias, L.C., Mousseau, V., Figueira, J., Clímaco, J., 2002. An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI. *European Journal of Operational Research* 138, 332–348.
- Durso, A., 1992. An interactive combined branch-and-bound/Tchebycheff algorithm for multiple criteria optimization. In: *Proceedings of the 9th Conference on MCDM 5-8/8*, Goicoechea, Springer.
- Jacquet-Lagrèze, E., 1995. An application of the UTA discriminant model for the evaluation of R & D projects. In: Pardalos, P.M., Siskos, Y., Zopounidis, C. (Eds.), *Advances in Multicriteria Analysis*. Kluwer Academic Publishers, Dordrecht, pp. 203–211.
- Jacquet-Lagrèze, E., Siskos, Y., 1982. Assessing a set of additive utility functions for multicriteria decision-making. *European Journal of Operational Research* 10 (2), 151–164.
- Jacquet-Lagrèze, E., Siskos, Y., 2001. Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research* 130, 233–245.
- Lourenço, R., Costa, J.P., 2001a. Applying LinearTri to a Multiple Objective Surgery Scheduling Model. In: *Proceedings of the 6th International Conference of the Decision Sciences Institute, Tec. de Monterrey, Chihuahua, Mexico*.
- Lourenço, R., Costa, J.P., 2001b. Classification of Nondominated Solutions in Multiple Objective Linear Integer Programming Problems. In: Koksalan M., Zionts, S. (Eds.), *Multiple Criteria Decision Making in the New Millennium: Proceedings of 15th International Conference on Multiple Criteria Decision Making (MCDM)*, Middle East Technical University, Ankara, Turkey, Springer 507, pp. 132–141.

- Lourenço, R., Costa, J.P., 2002. Determinação de perfis de referência para classificação de soluções não-dominadas em Programação Linear Inteira Multi-Objectivo. Relatório de Investigação, Faculdade de Economia, Universidade de Coimbra, Portugal.
- Marcotte, O., Soland, R.M., 1986. An interactive branch-and-bound algorithm for multiple criteria optimization. *Management Science* 32 (1), 61–75.
- Mousseau, V., Figueira, J., Naux, J.-P., 2001. Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. *European Journal of Operational Research* 130, 263–275.
- Mousseau, V., Slowinski, R., 1998. Inferring an ELECTRE TRI Model from assignment examples. *Journal of Global Optimization* 12, 157–174.
- Mousseau, V., Slowinski, R., Zielniewicz, P., 2000. A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support. *Computers & Operations Research* 27, 757–777.
- Ngo The, A., Mousseau, V., 2000. Using assignment examples to infer category limits for the ELECTRE TRI method. *Cahier du Lamsade* 168, Université Paris-Dauphine.
- Roy, B., 1996. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic Publishers, Dordrecht.
- Roy, B., Bouyssou, D., 1993. *Aide Multicritère à la Décision: Méthodes et Cas*. Economica, Paris.
- Steuer, R.E., 1986. *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, New York.
- Yu, W., 1992. *ELECTRE TRI—Aspects méthodologiques et guide d'utilisation*. Document du Lamsade 74, Université Paris-Dauphine.
- Zopounidis, C., Doumpos, M., 2002. Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research* 138, 229–246.