A multiple criteria sorting method in a web-based platform

Portuguese public hospital performance assessment

Ricardo Jorge Tomás Oliveira

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisors: Prof. Dr. José Rui de Matos Figueira
Prof. Dr. José Luís Brinquete Borbinha

Examination Committee

Chairperson: Prof. Dr. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. Dr. José Rui de Matos Figueira
Member of the Committee: Prof. Dr. Diogo Filipe da Cunha Ferreira

November 2019

Preface

The work presented in this thesis was performed at the Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento (INESC-ID), Civil Engineering Research and Innovation for Sustainability (CERIS) and Centro de Estudos de Gestão (CEG-IST) of Instituto Superior Técnico (Lisbon, Portugal), during the period February-September 2019. This work was part of the hSNS FCT – Research Project (PTDC/EGE-OGE/30546/2017).

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank the members of a large multidisciplinary group who made this work possible: my supervisors professor José Rui Figueira and professor José Borbinha for their continuous support, Rúben Rodrigues and André Xiang for their excellent work in developing DECSPACE, Ana Sara Costa for the support as a multicriteria expert, Diogo Ferreira, Miguel Pereira, António Rocha and Sara Ferreira for their work on the project, and professor Alexandre Nunes for the support as decision maker and knowledge in health administration.

Thanks to the Núcleo de Engenharia Biomédica (NEBM) for the opportunities and teachings provided that were very helpful in the development of this work. Thank you Alexandre and Leonor for this adventure.

Thank you to all my friends and colleagues from Biomedical Engineering who have accompanied me on this long journey. They were all very important but I have to mention some of them: Carlota, Clara, Cláudia, João, Sara, Valentina and Vanda.

Last but not least, thanks also to my longtime friends Raul, Bruno and Frango for the unconditional support and funny moments. It would be impossible to mention unconditional support without thanking my family. Thank you Mom, Dad and Brother.

To all of you, thank you!

Abstract

Multiple Criteria Decision Analysis (MCDA) methods are widely used in several fields and disciplines, such as healthcare, and their use can be quite complex. Therefore, a Decision Support System (DSS) can be useful to compute these methods and visualize their results. However, most of the available solutions are desktop based applications with usability issues that not fulfill the requirements of a case study. For this purpose, this work describes the design and implementation of ELECTRE TRI-nC method, a multiple criteria sorting method for handling categorical classification problems taking into account several reference actions to characterize each category, in DECSPACE, a web-based innovative platform for supporting decision aiding processes using one or more MCDA methods in a user-friendly interface. The method implementation was validated by testing it against a numerical example. The demonstration of its use in the platform to hospital performance assessment allows to draw conclusions and recommendations and demonstrates how a DSS can be used to facilitate the process of applying the method in real cases. DECSPACE aggregates the best features of the previous tools, although it stills has margin to be improved in future works, either related to this method or to new ones. Multiple applications of this method in different areas can be carried out by making use of this successfully implementation in DECSPACE.

Keywords: Decision Support System, DecSpace, ELECTRE TRI-nC, Hospital Performance Assessment, Multiple Criteria Decision Analysis.

Resumo

Métodos de Análise de Decisão Multicritério (ADMC) são amplamente usados em diversas áreas e disciplinas, como nos cuidados de saúde, e o seu uso pode ser bem complexo. Assim, um Sistema de Apoio à Decisão (SAD) pode ser útil para aplicar estes métodos e visualizar os seus resultados. No entanto, a maioria das soluções disponíveis são baseadas no computador do utilizador, com problemas de usabilidade, que não satisfazem os requisitos de um caso de estudo. Para este propósito, este trabalho descreve a concepção e implementação do método ELECTRE TRI-nC, um método de classificação de múltiplos critérios para lidar com problemas de classificação categórica tendo em conta várias ações de referência para caracterizar cada categoria, no DECSPACE, uma plataforma inovadora online para apoiar processos de auxílio à decisão usando um ou mais métodos de ADMC numa interface de fácil utilização. A implementação do método foi validada testando-o com um exemplo numérico. A demonstração do seu uso na plataforma para avaliação de desempenho hospitalar permite esboçar conclusões e recomendações e demonstra como um SAD pode ser usado para facilitar o processo de aplicação do método em casos reais. DECSPACE reúne as melhores características das ferramentas anteriores, embora ainda tenha margem para ser melhorado em trabalhos futuros, quer relacionados com este método quer com novos. Múltiplas aplicações deste método em diferentes áreas podem ser levadas a cabo, fazendo uso desta implementação com sucesso no DECSPACE.

Palavras-chave: Análise de Decisão Multicritério, Avaliação de Desempenho Hospitalar, DecSpace, ELECTRE TRI-nC, Sistema de Apoio à Decisão.

Contents

1 **Introduction** 1

1.1 Motivation . . . 3

1.2 Objectives . . . 3

1.3 Outline . . . 4

2 **Multiple Criteria Decision Analysis (MCDA)** 5

2.1 Introduction to MCDA . . . 7

2.2 MCDA methods . . . 7

2.2.1 Outranking methods . . . 8

2.2.2 Multiattribute utility and value theories . . . 8

2.2.3 Non-classical approaches . . . 9

2.3 MCDA in healthcare . . . 10

2.4 MCDA-ULAVAL . . . 10

2.5 Summary . . . 13

3 **ELECTRE TRI-nC method** 15

3.1 Overview . . . 17

3.2 Concepts, definitions, and notation . . . 17

3.3 Assignment procedures . . . 20

3.4 Flowchart . . . 21

3.5 Summary . . . 23

4 **Implementation** 25

4.1 DECSPACE overview . . . 27

4.2 ELECTRE TRI-nC implementation . . . 30

4.3 Building an ELECTRE TRI-nC workflow . . . 35

4.4 Validating the implementation . . . 37

4.5 Summary . . . 41

5 **Demonstration** 43

5.1 Case description . . . 45

5.1.1 Background . . . 45

5.1.2 Performance assessment in healthcare . . . 46

5.2 Data and sample . . . 46

5.2.1 Actions . . . 47

5.2.2 Criteria 47
5.2.3 Performance table 50
5.3 Preference parameters 51
5.3.1 Categories and reference actions 52
5.3.2 Criteria weights 52
5.3.3 Thresholds 53
5.4 Results and discussion 54
6 Conclusion and future work 57
6.1 Conclusion 59
6.2 Future work 59
A ELECTRE TRI-nC algorithm code 65
B DECSPACE demonstration 75

List of Figures

2.1 MCDA-ULAVAL interface. 11
2.2 DIVIZ workflow. 12

3.1 Relations between two actions according to a criterion. 18
3.2 Partial discordance and concordance indices according to a criterion. 19
3.3 Flowchart of ELECTRE TRI-nC method. 22

4.1 DECSPACE homepage (pre-alpha version). 27
4.2 DECSPACE method catalog. 29
4.3 DECSPACE workspace. 30
4.4 Directory structure of DECSPACE source code. 31
4.5 DECSPACE module of ELECTRE TRI-nC method. 32
4.6 DECSPACE modal tab of ELECTRE TRI-nC method. 33
4.7 DECSPACE workflow of ELECTRE TRI-nC method. 37
4.8 Results obtained for the numerical example ($\lambda = 0.65$). 40

5.1 Chromatic classification system for occupancy rate. 50

6.1 DECSPACE workflow of DECK CARDS METHOD and ELECTRE TRI-nC method. 60

B.1 Actions in DECSPACE. 77
B.2 Criteria in DECSPACE. 78
B.3 Performance table in DECSPACE. 78
B.4 Reference actions in DECSPACE. 79
B.5 DECK CARDS METHOD (SCENARIO 1) in DECSPACE. 80
B.6 DECK CARDS METHOD (SCENARIO 2) in DECSPACE. 81
B.7 Criteria weights (SCENARIO 1) in DECSPACE. 82
B.8 Criteria weights (SCENARIO 2) in DECSPACE. 82
B.9 Thresholds in DECSPACE. 82
B.10 Credibility level in DECSPACE. 82

List of Tables

2.1 Summary of MCDA-ULAVAL characteristics. 12

4.1 Criteria weights and thresholds for the numerical example. 37
4.2 Characteristic reference actions for the numerical example. 38
4.3 Set of projects for the numerical example. 39

5.1 Portuguese public hospitals under assessment. 48
5.2 Criteria, indicators and corresponding direction preferences. 49
5.3 Performance table. 51
5.4 Categories, reference actions and their performance for each criterion. 52
5.5 Ranking of criteria and white cards. 53
5.6 Criteria weights. 53
5.7 Preference, indifference and veto thresholds for each criterion. 54
5.8 Assignment results for both scenarios ($\lambda = 0.65$). 54

Acronyms

AHP	Analytic Hierarchy Process
ANP	Analytic Network Process
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DEA	Data Envelopment Analysis
DM	Decision Maker
DSS	Decision Support System
ELECTRE	ELimination and Choice Expressing REality
EPE	Public Enterprise
HTA	Health Technology Assessment
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
MACBETH	Measuring Attractiveness by a Categorical Based Evaluation Technique
MAUT	Multiple Attribute Utility Theory
MCDA	Multiple Criteria Decision Analysis
MEVN	MongoDB, Express, Vue.js and Node.js
NHS	National Health Service
PPP	Public–Private Partnerships
PROMETHEE	Preference Ranking Organization METHod for Enrichment of Evaluations
UTA	Utilité Additive

1

Introduction

Contents

1.1 Motivation 3
1.2 Objectives 3
1.3 Outline 4

1.1 Motivation

Multiple Criteria Decision Analysis (MCDA) is a sub-discipline of Operations Research that studies the process of using multiple criteria methods, techniques and tools to assist a Decision Maker (DM) for obtaining responses to the questions in a decision process. MCDA methods are widely used in several fields and disciplines (such as healthcare, where decisions are complex and involve confronting trade-offs between various, often conflicting, criteria) and their use can be quite complex [1]. Therefore, a Decision Support System (DSS) can be useful to compute these methods and visualize their results. For a real case, most of the available software solutions are desktop based applications, which usually makes available just few methods. An increasing trend is to reduce the gap between researchers and practitioners by taking into account requirements related to accessibility, usability and user friendly issues in the development of this kind of software solutions [2].

For this purpose, in this thesis, the focus is the implementation of the ELimination and Choice Expressing REality (ELECTRE) TRI-nC method, a multiple criteria sorting method which takes into account several reference actions to characterize each category [3], in DecSpace (Decision Space), a web-based innovative platform that gives the possibility of building decision models in an intuitive graphical user interface [4], and a detailed demonstration in the case of Portuguese public hospital performance assessment to draw conclusions and recommendations and to support the application of ELECTRE TRI-nC in DecSpace for real cases. This case was motivated by the project hSNS[1] that focus on the Portuguese public hospitals performance assessment.

The ELECTRE TRI-nC method allows the DM to assign *actions* to predefined *categories* considering several *criteria* and according to the DM preferences. To obtain the results assignment, the method needs data and preference parameters to be used as input for the computations. The preference parameters to be defined include reference actions, criteria weights, thresholds and credibility level.

The DecSpace aims to be user-friendly to explore solutions for decision situations to either non-expert users or MCDA expert users, with researching, teaching or even for consulting purposes.

1.2 Objectives

The following objectives can be defined:

1. Implement the ELECTRE TRI-nC method in DecSpace;
2. Apply the implemented method to a real case in health sector and draw conclusions and recommendations.

The accomplishment of these objectives will be addressed in Chapter 6.

[1] https://hsns.eu/

1.3 Outline

This thesis is composed by six chapters aligned to the objectives referred above. In this first chapter, an introduction to this work is presented with a focus on motivation that led to do it. The second chapter comprises a literature review on MCDA, the use of this methodology in healthcare and details software tools that are being used. In this analysis the need for a new platform comes naturally. The third chapter describes the ELECTRE TRI-nC method formally and in a perspective of computational implementation, by providing all the required input and computation steps in a flowchart. In the fourth chapter, the DecSpace platform is presented, the implementation of the method itself is detailed and also validated. Next, the fifth chapter demonstrates the use of the implemented method in a case of Portuguese public hospital performance assessment, describing all the data, sample and variables used, followed by a briefly discussion of the results obtained. In the sixth and last chapter, the conclusions of this work and hints for future work are presented.

2

Multiple Criteria Decision Analysis (MCDA)

Contents

2.1 Introduction to MCDA 7

2.2 MCDA methods 7

2.3 MCDA in healthcare 10

2.4 MCDA-ULAVAL 10

2.5 Summary 13

This chapter presents a literature review of MCDA, namely a briefly introduction (Section 2.1), a description of MCDA methods (Section 2.2), the use of MCDA in healthcare (Section 2.3), a description of MCDA softwares namely MCDA-ULAVAL (Section 2.4) and lastly a brief summary of the chapter (Section 2.5).

2.1 Introduction to MCDA

In daily life, multiple criteria decision situations are frequent. For instance, when buying a new product, price is usually one of the main criteria, and quality are typically another criterion, easily in conflict with the price: it is unusual that the cheapest product has the highest quality.

MCDA is a sub-discipline of Operations Research that studies the process of using multiple criteria models to help obtain responses to the questions in a decision process. MCDA is not just a collection of formulations and methodologies, but a specific perspective to deal with decision problems [5].

MCDA perspective is closely related to the humans intuition to make decisions. Therefore, the main steps of MCDA can be defined as follows: 1. Defining the decision problem; 2. Selecting and structuring criteria; 3. Measuring performance; 4. Assessing actions/alternatives; 5. Weighting criteria; 6. Aggregating; 7. Dealing with uncertainty; 8. Reporting and examining of findings [6]. This process helps making decisions mainly in terms of choosing, sorting or ranking the alternatives, briefly described as follows:

- *Choice problematic* is focused on the selection of a small number as possible of alternatives in such a way that a single alternative may finally be chosen;
- *Sorting problematic* is focused on an assignment of each action to the most appropriate category among those of a family of predefined categories;
- *Ranking problematic* is focused on a complete or partial order of the actions by comparing actions among each other.

Due to this variety, next section presents the different classes of MCDA methods, mainly based on a MCDA state-of-the-art book [5].

2.2 MCDA methods

MCDA methods are widely used in several fields and disciplines. Most of the research focuses on the improvement and development of new methods. However, the appropriate method selection for the given decision problem is also very important. For that, a methodological and practical framework for selecting suitable MCDA methods for a particular decision situation can be useful [7].

2.2.1 Outranking methods

The class of outranking based multiple criteria decision methods is presented in this subsection. Given the DM preferences, the performances of the actions and the nature of the problem, an outranking relation between two actions, to decide if an action is at least as good as other action, can be defined.

The ELECTRE methods strictly apply the definition of outranking relation. However, the methods which are based on pairwise comparison of actions are considered in the outranking methods class. Thus, the Preference Ranking Organization METHod for Enrichment of Evaluations (PROMETHEE) methods, are also included in this part.

The conception of ELECTRE methods started in the 1960s by B. Roy and the theoretical research on their foundations has been intensive all this time. These methods are handling the preference modelling with outranking relations and concepts of concordance and discordance [8].

The family of ELECTRE methods includes several methods designed for the three main problematic defined previously. For sorting problematic, that is the object of study in this work, the following methods are developed: ELECTRE TRI-B (originally called just ELECTRE TRI, read 'ELECTRE tree'), ELECTRE TRI-nB, ELECTRE TRI-C, ELECTRE TRI-rC and ELECTRE TRI-nC. The ELECTRE TRI-nC method is presented in Chapter 3.

The first PROMETHEE methods were developed by J.P. Brans in the 1980s and few years later the same author proposed the visual interactive module GAIA which provides a graphical representation to support PROMETHEE methods.

The preference modelling structure of PROMETHEE is based on pairwise comparisons: the deviation between the performances of two actions on a particular criterion is considered. For small deviations, the DM will allocate a small preference to the best alternative (or no preference if is considered that this deviation is negligible), and for larger deviations, larger will be the preference.

Beside these well known methods, other outranking methods were proposed. All these methods (QUALIFLEX, REGIME, ORESTE, ARGUS, EVAMIX, TACTIC and MELCHIOR) propose definitions and computations of particular binary relations, linked to the basic idea of the ELECTRE methods. Other methods (MAPPAC, PRAGMA, IDRA and PACMAN) have been developed in the framework of the Pairwise Criterion Comparison Approach methodology, as in PROMETHEE methods.

2.2.2 Multiattribute utility and value theories

The class of multiattribute utility and value theories is presented in this subsection.

The Multiple Attribute Utility Theory (MAUT) is a model of preference suitable for applications involving risky choice by trying to assign a utility value to each action. The utility is a real number representing the preference of the considered action under risk, very often calculated by the sum of the marginal utilities in each criterion. Thus, this approach very often coincides with the traditional weighted sum.

The Utilité Additive (UTA) methods models the DM preferences using linear programming techniques in order to optimally infer additive value/utility functions.

Analytic Hierarchy Process (AHP) is a theory of measurement that uses pairwise comparisons along with expert judgments to deal with the measurement of qualitative or intangible criteria. The Analytic Network Process (ANP) is a general theory of relative measurement used to derive composite priority ratio scales from individual ratio scales that represent relative measurements of the influence of elements that interact with respect to control criteria. This feedback structure does not have the top-to-bottom form of a hierarchy but looks more like a network, with cycles and loops connecting elements.

Measuring Attractiveness by a Categorical Based Evaluation Technique (MACBETH) is an MCDA approach that requires only qualitative judgements about differences of values of attractiveness of one action over another action to help an individual or a group to quantify the relative preference of different actions. To ease the judgemental process, six semantic categories of difference of attractiveness: “very weak”, “weak”, “moderate”, “strong”, “very strong” or “extreme” are possible answers. By pairwise comparing the elements, a matrix of qualitative judgements is filled in. The software M-MACBETH[1] automatically detects “inconsistency”, even for an incomplete matrix of judgements. This approach has been used in many public and private applications of multicriteria additive value analysis.

2.2.3 Non-classical approaches

Some approaches have been proposed in MCDA besides outranking methods and multiattribute utility and value theories to deal with uncertainty: fuzzy and rough approaches.

The modelling approaches for internal and external uncertainties may often become qualitatively different in nature. The boundary between external uncertainty and imprecision is fuzzy. Fuzzy set approaches for choice, ranking and sorting problems consider conflicting systems uncertain and imprecise knowledge in a preference model with the appropriate granularity. The Choquet integral is an example of it.

In a rough set approach, preferences can be modelling conditionally in terms of ”if... then...”, following decision rules. This allows to express the decision model in a very intuitive way. The set of decision rules defined by the DM can be applied to a certain set of alternatives in order to assign them to categories (sorting problem) or to obtain specific preference relations in the set of actions (choice and ranking problems).

[1] M-MACBETH is available at http://m-macbeth.com/

2.3 MCDA in healthcare

Healthcare decisions are complex and involve confronting trade-offs between various, often conflicting, criteria. The iron triangle describes how the three fundamental criteria in healthcare delivery (cost, quality and access) interact: in order to increase one criterion, one or both of the remaining criteria must decrease [9]. Using structured models to improve the quality of decision making involving multiple criteria are useful for this purpose. Examples of its use in healthcare are: benefit risk analysis, Health Technology Assessment (HTA), resource allocation, portfolio decision analysis, shared patient clinician decision making and prioritizing patients' access to services [1].

Health interventions may be chosen to maximize general population health and to reduce health inequalities, all with respect to practical and budgetary constraints. This is the type of problem that policy makers need help to solve rationally. This stresses the need for MCDA to support priority setting, which has recently been identified as one of the most important issues in health system research [6].

One of the areas where MCDA is mostly used is HTA, that concerns the way that the costs and benefits of new healthcare technologies are identified and compared. Cost effectiveness is widely used as the criterion, to choose about where to invest on or not. This approach just tells to DMs about the cost per unit of benefit produced. However, it is not sufficient to make the best decisions: this criterion is just one in many others such as need, appropriateness, quality and ethical or social values [10].

2.4 MCDA-ULAVAL

MCDA-ULAVAL[2] is a free, standalone and open-source desktop application. It only makes available methods from ELECTRE family (specifically ELECTRE II, III, TRI-B, TRI-C, TRI-rC and TRI-nC) and it is the most used tool in the application of methods of this family.

A MCDA-ULAVAL project has different types of objects:

- Alternatives are stored in a single set on which you can define and name subsets.
- Criteria are stored in a single set and implement the same subset mechanism as alternatives.
- Performance tables hold performance values of the alternatives in such criteria.
- Method and criterion parameters. Parameters that have a value for each criterion are referred to as criterion parameters and parameters that are not function of a criterion are referred to as method parameters.
- Decision configurations hold the parameters needed by decision methods. Criterion parameters are edited in a matrix where the rows correspond to the parameters and the columns to the criteria.

[2] MCDA-ULAVAL is available at `http://cersvr1.fsa.ulaval.ca/mcda-ulaval`

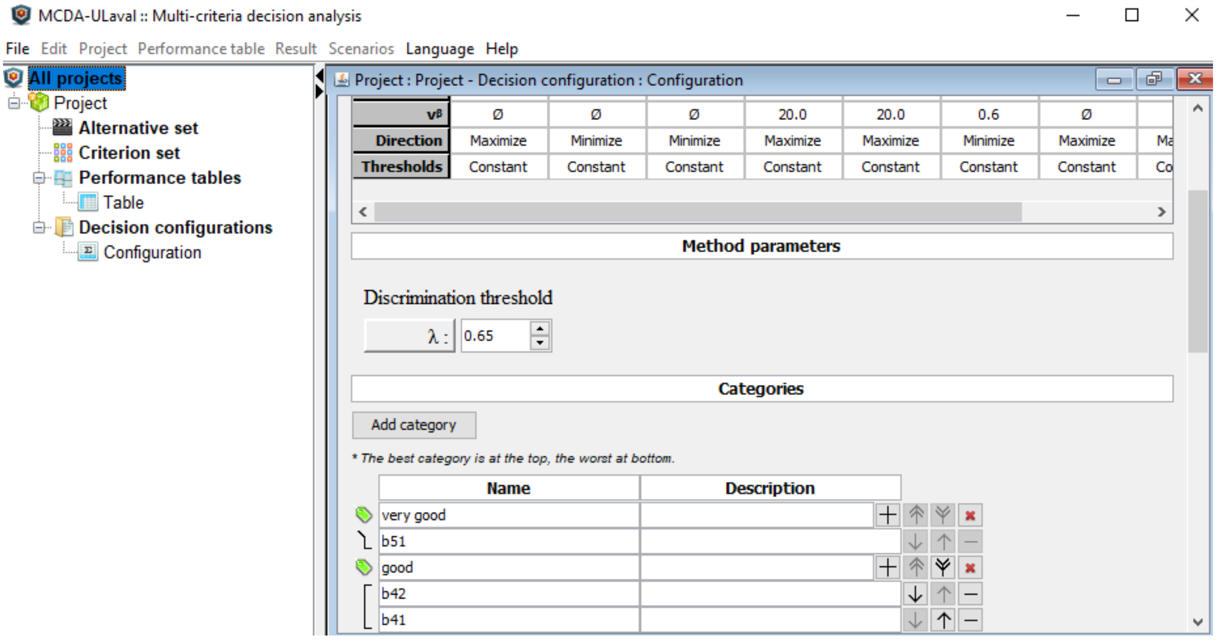
- Decision tuples consist of alternative and criteria sets, a performance table and a decision configuration. Together, these items set the stage for the execution of a decision method with the all the data and parameters.

- Results are the product of executing a decision tuple. Its validity is checked before execution. If it contains an error, a message describing the problem will be displayed.

- Scenario analysis. The starting point for a scenario analysis is a decision tuple. The parameters are selected to vary and how they vary. All the parameters that are not included in the analysis have as default values those specified by the initial configuration.

A main feature of MCDA-ULAVAL is the possibility of insert data by importing Comma-Separated Values (CSV) format files.

As shown in Figure 2.1, the list of objects described before are displayed in the left side of the screen and their windows overlap in the right side of the screen, which sometimes makes it difficult to access them. The interface itself is not very innovative and graphical. Even though, this tool makes available various data insertion techniques (as can be seen in Figure 2.1): pre-defined tables with all the necessary parameters and buttons to easily add, move and delete data.


MCDA-ULaval :: Multi-criteria decision analysis
File Edit Project Performance table Result Scenarios Language Help
All projects
Project
Alternative set
Criterion set
Performance tables
Table
Decision configurations
Configuration
Project : Project - Decision configuration : Configuration
Method parameters
Categories
Add category

vᵝ	Ø	Ø	Ø	20.0	20.0	0.6	Ø	
Direction	Maximize	Minimize	Minimize	Maximize	Maximize	Minimize	Maximize	Ma
Thresholds	Constant	Constant	Constant	Constant	Constant	Constant	Constant	Co

Discrimination threshold

λ : 0.65

** The best category is at the top, the worst at bottom.*

Name	Description
very good	
b51	
good	
b42	
b41	

Figure 2.1: MCDA-ULAVAL interface.

A summary of MCDA-ULAVAL main characteristics (pros and cons) is displayed in Table 2.1.

Aside from this tool, there are dozens of other DSSs developed with different features and methods available. None of these software tools are such that users without any prior experience of MCDA could use it. An increasing trend is to reduce the gap between researchers and practitioners [2].

Table 2.1: Summary of MCDA-ULaval characteristics.

Pros	Cons
✓ Open Source	× Desktop application
✓ Data importation in CSV format	× Not modern user interface
✓ Predefined tables and buttons	× Not adaptable for different types of devices
✓ Data validation	× Just one family of methods available
✓ Scenario analysis	× Not allow multi-user projects

The Decision Deck project was created to collaboratively develop open source software tools implementing MCDA techniques to support complex decision aiding processes. Diviz[3], one of the software initiatives of the project, eases the use of algorithmic resources to build, execute and share complex workflows of MCDA algorithms (Figure 2.2) as ELECTRE TRI family. It is a pedagogical tool for teachers who need to present and compare classical MCDA methods and it may also be used by practitioners who wish to solve decision problems with a given method [11]. However, it also suffers from the flaw of being a desktop standalone application, as MCDA-ULaval.

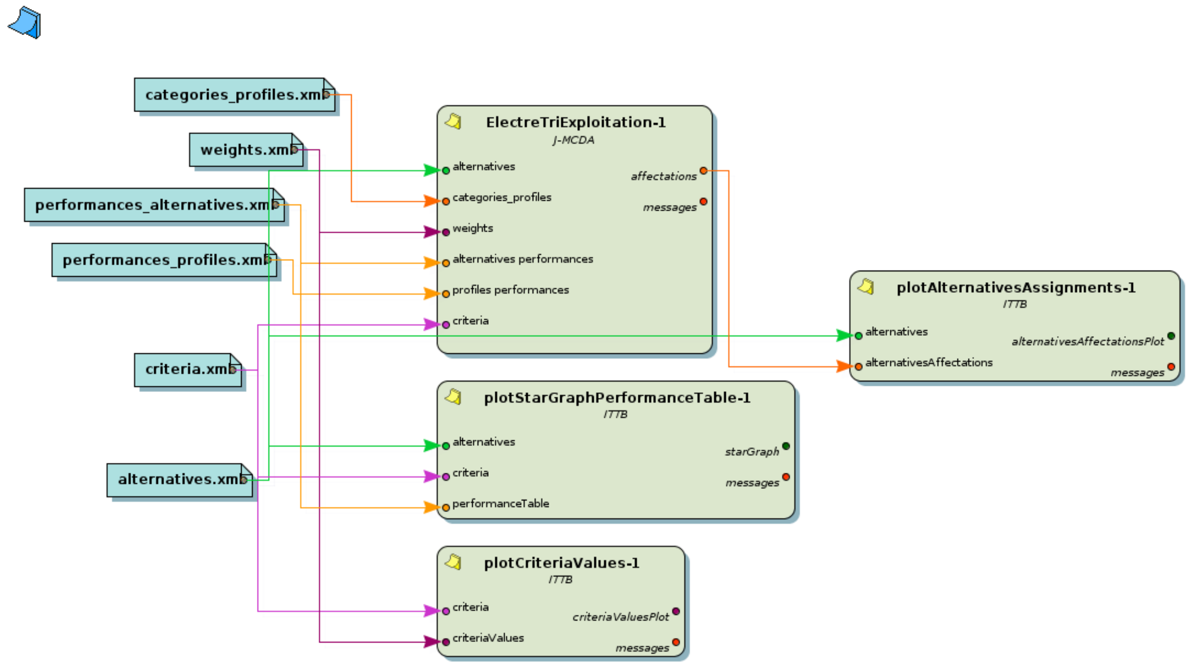


Figure 2.2: Diviz workflow.

[3] Diviz is available at https://www.diviz.org/

A newly tool in development, DecSpace[4], inspired by the workflow design of Diviz, is an open source web-based platform optimized to different types of devices using state-of-the-art technology in terms of user interface and back-end that allows a project to be accessed by multi-users. Chapter 4 is fully dedicated to this platform.

2.5 Summary

In this chapter, the area of MCDA is presented as a sub-discipline of Operations Research that studies the process of using multiple criteria models to help obtain responses to the questions in a decision process. The decision problematic can be a choice, sorting or ranking problematic.

There are different classes of MCDA methods such as outranking methods (where the ELECTRE family belongs to), multiattribute utility and value theories and non-classical approaches.

MCDA models are useful to make healthcare decisions, because they are complex and involve confronting trade-offs between various, often conflicting, criteria.

To apply a MCDA method, a software tool can be needed due to the complex calculations that are involved many times. MCDA-ULaval is a free, standalone and open-source desktop application and it is the most used tool in the application of methods of ELECTRE family. Diviz is one of the software initiatives of Decision Deck project that eases the use of algorithmic resources to build, execute and share complex workflows of MCDA algorithms.

A newly tool in development, DecSpace, has the best features of the existent tools: is an open source web-based platform optimized to different types of devices using state-of-the-art technology in terms of user interface and back-end that allows a project to be accessed by multi-users. Currently, DecSpace has a very small number of methods available, therefore there is a lot of opportunities and work to do.

[4] DecSpace pre-alpha is available at http://app.decspacedev.sysresearch.org

3

ELECTRE TRI-nC method

Contents

3.1 Overview 17

3.2 Concepts, definitions, and notation 17

3.3 Assignment procedures 20

3.4 Flowchart 21

3.5 Summary 23

This chapter presents a overview of the ELECTRE TRI-nC method (Section 3.1), concepts, definitions, and notation necessary (Section 3.2), as well as the assignment procedures of the actions into categories (Section 3.3). Afterwards, a flowchart of the method, useful for implementation, is presented (Section 3.4) and lastly a brief summary of the chapter (Section 3.5).

3.1 Overview

As referred in Section 2.1, ELECTRE TRI family methods (sorting methods) are used in contexts where the objects of a decision must be assigned to a category according to several criteria. Traditionally, categories definition makes use of limiting profiles (boundary actions) [12]. However, in real-life decision aiding context is more useful to define those categories using central profiles (reference actions) [3]. The actions to be assigned are not compared to boundary actions that represent lower and upper bounds of the categories, but instead they are compared to representative characteristic actions of each category.

The ELECTRE TRI-nC method was proposed as a multiple criteria sorting method which takes into account several reference actions to characterize each category [3]. To apply this method, the set of categories to which the actions must be assigned to has to be completely ordered (from the worst to the best, for instance) and defined *a priori*. Each category has to be characterized by several reference actions judged by the DM as representative or informative. This method follows a decision aiding constructive approach, through the interaction between the analyst(s) and the DM(s). In the case of each category being only defined by a single reference action, the method is equivalent to ELECTRE TRI-C method [13].

Many applications of these methods were verified in the last years. For instance, the ELECTRE TRI-C method was applied to assisted reproduction [14], for erosion risk assessment [15] and for government performance assessment [16]. The ELECTRE TRI-nC method was used for identifying favourable climates for tourism [17], for supplier classification [18] and for water utilities performance assessment [19]. In this thesis, a demonstration of the ELECTRE TRI-nC method for Portuguese public hospital performance assessment is presented (Chapter 5).

3.2 Concepts, definitions, and notation

This section is dedicated to the main concepts, definitions, and notation concerning the ELECTRE TRI-nC method, adapted from [3] and [13].

Let $A = \{a_1, a_2, ..., a_i, ...\}$ denote the *set of actions*. These actions can be known *a priori* or it may appear during the decision aiding process. The main objective of the method is to assign these actions to a *set of categories* $C = \{C_1, C_2, ..., C_h, ...C_q\}$ with $q \geqslant 2$ (with $q = 1$ there is no sorting problem). Note that as referred in Section 3.1, the set of categories has to be completely ordered, such that C_1 is the worst category and the C_q is the best one. To evaluate any action, a *set of criteria* $G = \{g_1, g_2, ..., g_j, ...g_n\}$ is defined. $g_j(a_i)$ denotes the performance of the action a_i in the criterion g_j.

In what follows, assume, without loss of generality, that all criteria $g_j \in G$ are to be maximized, which means that the preference increases when the criterion performance increases too. Due to imperfect character of the data as well as the arbitrariness that affects the definition of the criteria, two thresholds are associated to g_j: an *indifference threshold*, q_j, and a *preference threshold*, p_j. A *veto threshold*, v_j, can also be associated, such that $v_j \geqslant p_j \geqslant q_j \geqslant 0$. Based on the definition of such thresholds, the following relations between two actions, a and a', can be derived:

(i) $\mid g_j(a) - g_j(a') \mid \leqslant q_j$ represents a non-significant advantage of one of the two actions over the other, meaning that a is indifferent to a' according to g_j, denoted aI_ja'.

(ii) $g_j(a) - g_j(a') > p_j$ represents a significant advantage of a over a', meaning that a is preferred to a' according to g_j, denoted aP_ja'.

(iii) $q_j < g_j(a) - g_j(a') \leqslant p_j$ represents an ambiguity zone: the advantage of a over a' is enough to not have indifference, but not enough to conclude about a preference in favor of a, meaning that a is *weakly preferred* to a', denoted aQ_ja'.

These zones can be represented graphically as in Figure 3.1.

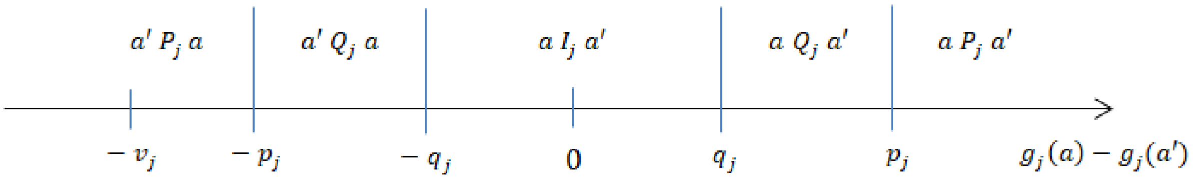


Figure 3.1: Relations between two actions according to a criterion.

To quantify the relations between two actions, a and a', according to a criterion g_j, a *partial concordance index*, denoted $c_j(a, a')$, is defined as follows:

$$c_j(a,a') = \begin{cases} 1 & \text{if } g_j(a) - g_j(a') \geqslant -q_j, \\ \dfrac{g_j(a) - g_j(a') + p_j}{p_j - q_j} & \text{if } -p_j \leqslant g_j(a) - g_j(a') < -q_j, \\ 0 & \text{if } g_j(a) - g_j(a') < -p_j. \end{cases} \tag{1}$$

And a similar definition of *partial discordance index*, denoted $d_j(a, a')$, is defined as follows:

$$d_j(a,a') = \begin{cases} 0 & \text{if } g_j(a) - g_j(a') \geqslant -p_j, \\ \dfrac{g_j(a) - g_j(a') + p_j}{p_j - v_j} & \text{if } -v_j \leqslant g_j(a) - g_j(a') < -p_j, \\ 1 & \text{if } g_j(a) - g_j(a') < -v_j. \end{cases} \tag{2}$$

According to Equations 1 and 2, these partial indices $\in [0, 1]$. A graphical representation is displayed in Figure 3.2.

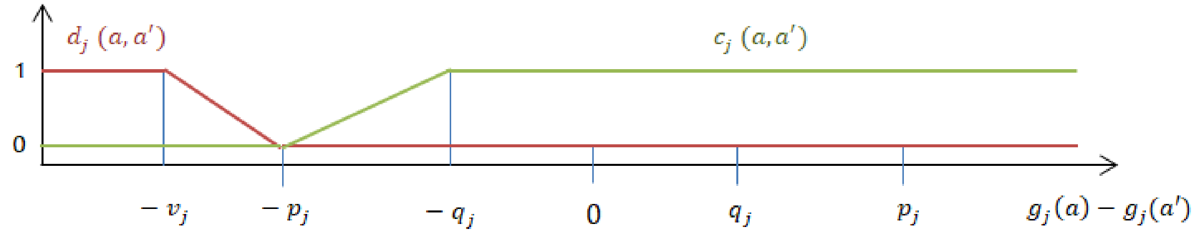


Figure 3.2: Partial discordance and concordance indices according to a criterion.

A *criterion weight*, denoted w_j, with $w_j > 0$ is associated to each criterion. Assume, without loss of generality, that $\sum_{j=1}^{n} w_j = 1$. Thus, the *global concordance index*, denoted $c(a, a')$, is defined as follows:

$$c(a, a') = \sum_{j=1}^{n} w_j c_j(a, a'). \tag{3}$$

Finally, the *credibility index*, denoted $\sigma(a, a')$, which quantifies the relation between a and a' when taking all the criteria from G into account, is defined as follows:

$$\sigma(a, a') = c(a, a') \prod_{j=1}^{n} T_j(a, a'), \tag{4}$$

where

$$T_j(a, a') = \begin{cases} \dfrac{1 - d_j(a, a')}{1 - c(a, a')} & \text{if } d_j(a, a') > c(a, a'), \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

Let $B = \{B_1, B_2, ..., B_h, ...B_q\}$ denote the *set of subsets of reference actions*. Each *subset of reference actions* to characterize category C_h, is defined as $B_h = \{b_h^r, r = 1, ..., m_h\}$, such that $m_h \geqslant 1$. The set $B \cup \{B_0, B_{q+1}\}$ denotes the *set of all reference actions*, such that $B_0 = \{b_0^1\}$ and $B_{q+1} = \{b_{q+1}^1\}$ contain two reference actions defined as follows: $g_j(b_0^1)$ is the worst possible performance on criterion g_j and $g_j(b_{q+1}^1)$ is the best possible performance on criterion g_j for all $g_j \in G$. Using this definition, for any action and criterion, the relation $g_j(b_0^1) < g_j(a) < g_j(b_{q+1}^1)$ is verified.

The comparison of an action a to the characteristic reference actions b_h^r, provides m_h credibility indices $\sigma(a, b_h^r)$ and m_h credibility indices $\sigma(b_h^r, a)$, calculated according to Equation 4. In order to find a representative credibility index for each action a with respect to each subset of reference actions, B_h, a *categorical index* is defined as follows:

$$\begin{aligned} (i) \quad & \sigma(a, B_h) = \max_{r=1,...,m_h} \sigma(a, b_h^r), \\ (ii) \quad & \sigma(B_h, a) = \max_{r=1,...,m_h} \sigma(b_h^r, a). \end{aligned} \tag{6}$$

Let λ denote a *credibility level* as the minimum degree of credibility, which is considered necessary by the DM for accept (or not) the outranking statement "a outranks B_h" taking into account all the criteria from G. Normally, $\lambda \in [0.5, 1]$. The assignment procedures, by comparing each categorical index to the credibility level, are described in Section 3.3.

3.3 Assignment procedures

This section presents the two assignment rules, which must be used conjointly. To apply the two rules, a credibility level is chosen and, to select between two consecutive categories, a *selecting function*, denoted $\rho(a, B_h)$, is defined. Due to the role played by this function, it must fulfill the following properties:

- $\rho(a, B_h)$ is a function of $\sigma(a, B_h)$ and $\sigma(B_h, a)$.
- The chosen condition for selecting the category C_h rather than $C_{h\pm1}$ (which depends on one of the two rules where the pre-selection is made) must be meaningful, such as $\rho(a, B_h) > \rho(a, B_{h\pm1})$.
- Let a and a' be two actions that allow to pre-select the same category. If a strictly dominates a', then $\rho(a, B_h) > \rho(a, B_{h\pm1}) \implies \rho(a', B_h) > \rho(a', B_{h\pm1})$.

According to these properties, the selecting function can be defined as follows:

$$\rho(a, B_h) = \min\{\sigma(a, B_h), \sigma(B_h, a)\}. \tag{7}$$

(Descending rule). Decrease h of B_h from $(q + 1)$ until the first value, t, such that $\sigma(a, B_t) \geqslant \lambda$ (C_t is the descending pre-selected category):

(i) For $t = q$, select C_q as a possible category to assign action a.

(ii) For $0 < t < q$, if $\rho(a, B_t) > \rho(a, B_{t+1})$, select C_t as a possible category to assign a, otherwise, select C_{t+1}.

(iii) For $t = 0$, select C_1 as a possible category to assign a.

(Ascending rule). Increase h of B_h from zero until the first value, k, such that $\sigma(B_k, a) \geqslant \lambda$ (C_k is the ascending pre-selected category):

(i) For $k = 1$, select C_1 as a possible category to assign action a.

(ii) For $1 < k < (q + 1)$, if $\rho(a, B_k) > \rho(a, B_{k-1})$, select C_k as a possible category to assign a, otherwise, select C_{k-1}.

(iii) For $k = (q + 1)$, select C_q as a possible category to assign a.

The assignment procedures lead to select two possible categories to which an action can be assigned to by using the descending and ascending rules conjointly. Therefore, the ELECTRE TRI-nC method provides as a possible assignment of an action:

(i) One category, when the two selected categories are the same.

(ii) Two categories, when the two selected categories are consecutive.

(iii) A range of categories, delimited by the two selected categories when they are non-consecutive.

3.4 Flowchart

This section presents a flowchart of the method, inspired by the design and implementation of another method in the platform [20]. This flowchart (Figure 3.3) contains three main blocks: input, calculations and output. They are briefly described as follows:

1. *Input.* It can be divided in two main types:
    - (a) *Data.* The basic data is composed by the set of actions, the set of criteria and the performance of the actions in the criteria (performance table).
    - (b) *Preference parameters.* They are chosen by the DM and are composed by the reference actions per category, the criteria weights, the thresholds per criterion and the credibility level.
2. *Computations.* These steps are described in Section 3.2.
3. *Output.* The output of the method are the assignment results. The assignment procedures, namely the application of the ascending and descending rules, are described in Section 3.3.

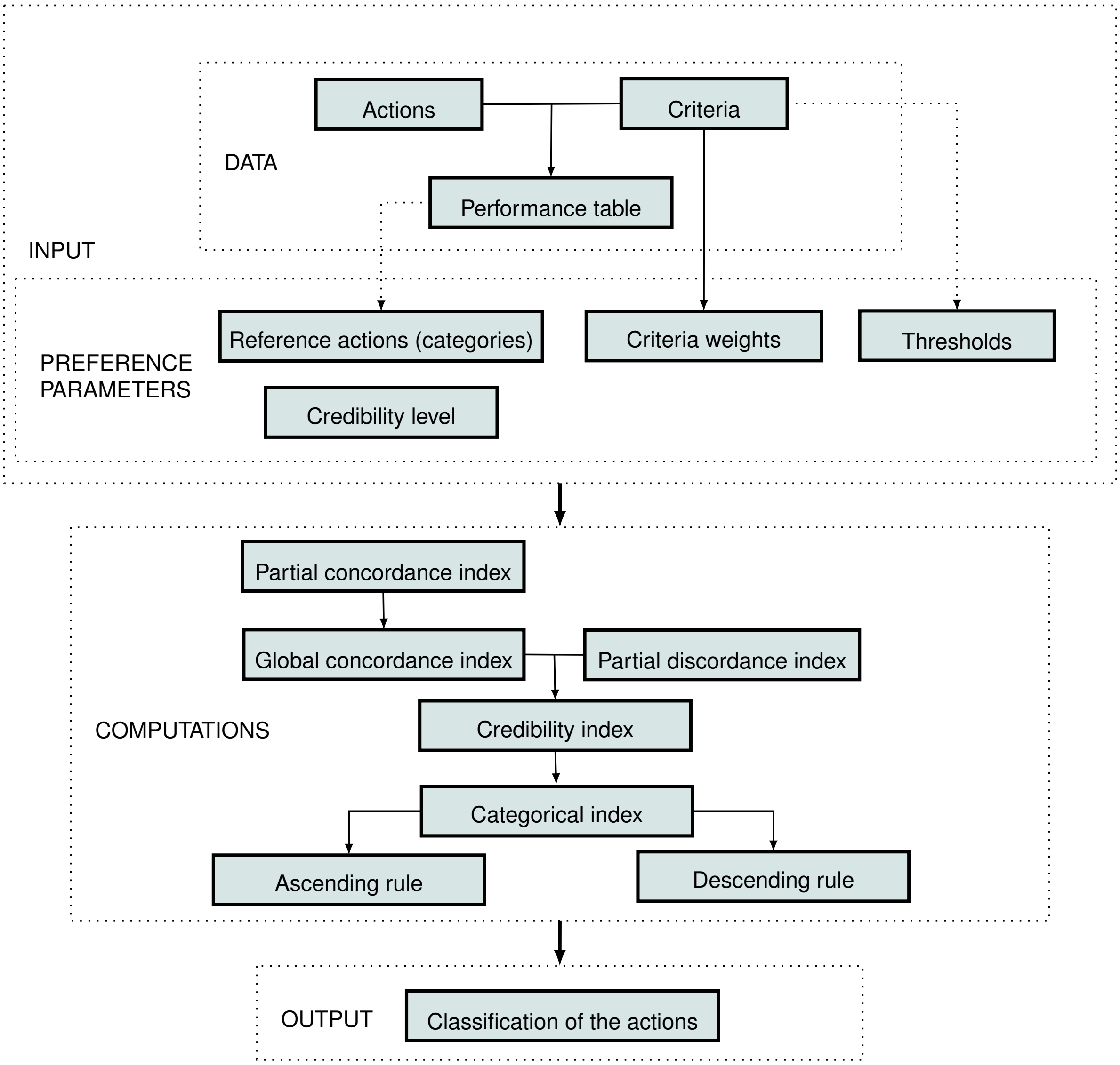


Figure 3.3: Flowchart of ELECTRE TRI-nC method.

3.5 Summary

The ELECTRE TRI-nC method is a multiple criteria sorting method which takes into account several reference actions to characterize each category. These categories have to be defined *a priori* and completely ordered. The main objective of the method is to assign actions to these set of categories according to their performance in a set of criteria.

Due to imperfect character of the data as well as the arbitrariness that affects the definition of criteria, two thresholds (*preference* and *indifference*) are associated to criteria and allow to define relations between two actions according to their performance in criteria. For each criterion, a *partial concordance* and *discordance indices* can be defined. Taking into account all the criteria, a *credibility index* can be defined. To find a representative index for each action with respect to each subset of reference actions of a category, a *categorical index* can be defined. A *credibility level*, which is considered the minimum degree of credibility to accept (or not) the outranking, is necessary to apply the two assignment rules, *ascending* and *descending rules*, that must be used conjointly.

All these steps of the method can be represented in a flowchart, useful for computational implementation purposes.

4

Implementation

Contents

4.1 DecSpace overview 27

4.2 ELECTRE TRI-nC implementation 30

4.3 Building an ELECTRE TRI-nC workflow 35

4.4 Validating the implementation 37

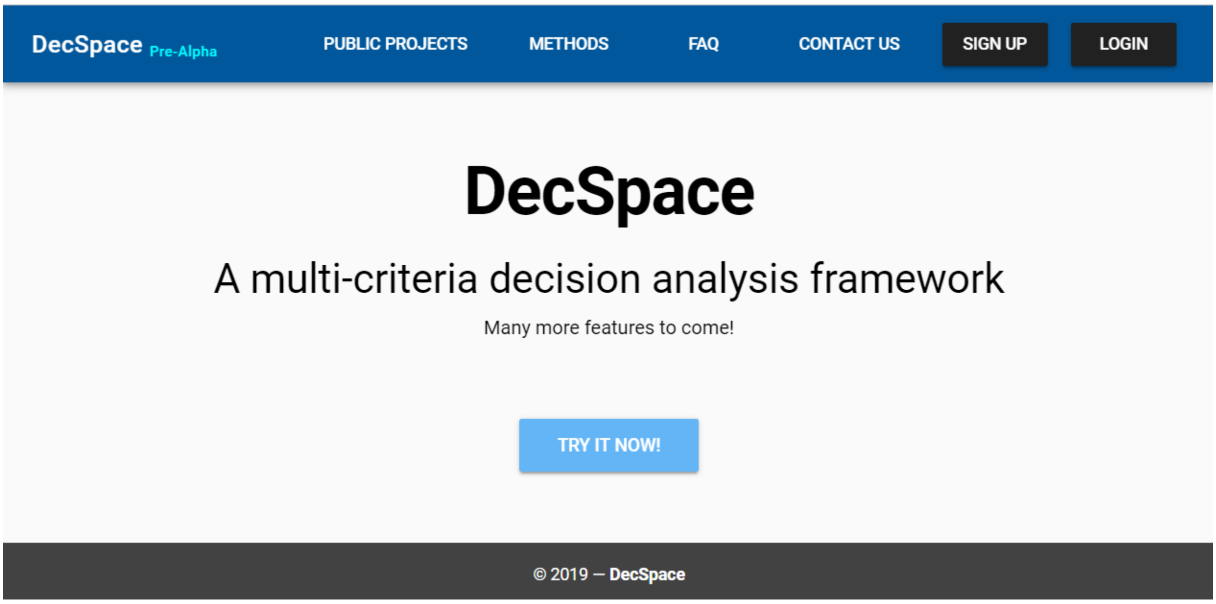
4.5 Summary 41

This chapter presents a overview of the DecSpace platform (Section 4.1), necessary steps for ELECTRE TRI-nC implementation (Section 4.2), explains how to build a ELECTRE TRI-nC workflow in the platform (Section 4.3), validates this implementation (Section 4.4) and lastly a brief summary of the chapter (Section 4.5) is presented.

4.1 DecSpace overview

As referred in Section 2.4, DecSpace, a new MCDA tool that is in development, aggregates the best features of other tools. It is inspired in Diviz, in the sense that it permits to construct workflows that may use multiple MCDA methods and data, connected in a way that allows to design complete decision aiding processes.

DecSpace is an innovative web-based platform that allows the efficient use of MCDA methods to support the user (or DM) during the decision process by giving the possibility of building decision models in an intuitive graphical user interface (in any web browser) [4]. The interface is optimized for different types of devices, including mobile devices as tablets and smartphones. Figure 4.1 displays DecSpace homepage. DecSpace is currently in pre-alpha, meaning that the platform has been in development, and this is an early release. More recently, several user experience improvements and front-end migration were carried out [21].



DecSpace

A multi-criteria decision analysis framework

Many more features to come!

© 2019 — DecSpace

Figure 4.1: DecSpace homepage (pre-alpha version).

DecSpace consists of a three-tiers architecture:

1. *Client tier*: it implements the user interface and sends user simple HTTP requests to the application

tier. After these requests are processed a response is returned to the client.

2. *Application tier*: confines the great complexity of the system, where most of the computational activity is performed. It processes the user requests sent by the client tier, and also carries out the connections to the data tier.

3. *Data tier*: it stores all the information of the application in a database. It receives and replies to any data requests sent by the application tier.

Regarding the technology, a combination of programming languages and libraries is used, through a MongoDB, Express, Vue.js and Node.js (MEVN) stack:

- `MongoDB`: document database used by the back-end application to store its data as JavaScript Object Notation (JSON) documents.

- `Express`: framework running on top of `Node.js` that provides a robust set of features for web and mobile applications.

- `Vue.js`: front-end web app framework that runs the JavaScript code in the user's browser, allowing the application user interface to be dynamic.

- `Node.js`: environment that lets implement the application back-end in JavaScript.

Aside the MEVN stack technology, HyperText Markup Language (HTML) is the web core language used for creating structured content in web pages and Cascading Style Sheets (CSS) is used for describing the visual style of the web pages. Due to the flexible nature of the three-tier architecture, it is possible to migrate the technology of just one of the layers. For instance, the recently front-end migration involved the migration from `AngularJS` to `Vue.js` [21].

To start using DECSPACE, there are four types of users with different permissions:

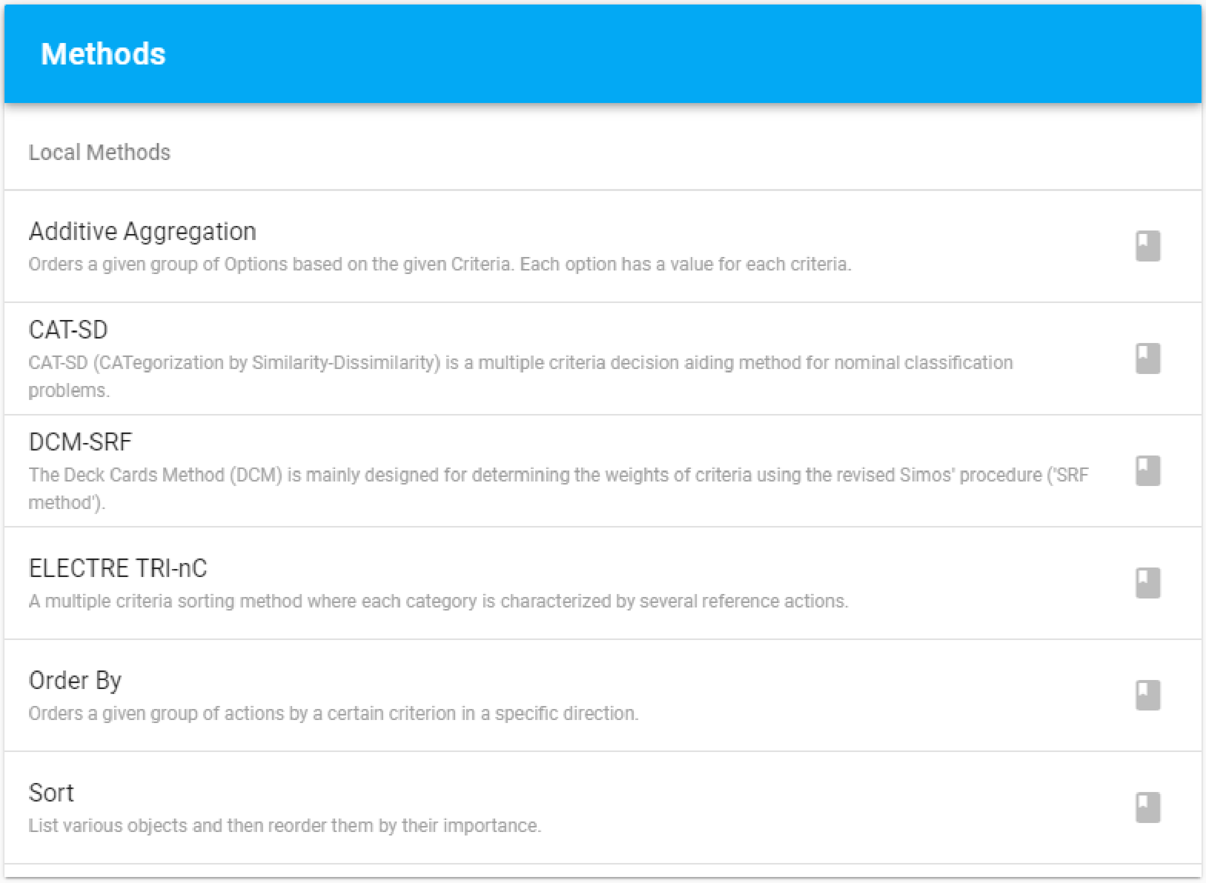
1. *Registered users*: they have an account to create and manage projects with all the features available in a project area.

2. *Anonymous users*: they can test the platform with temporary projects that cannot be saved but can be exported.

3. *Administrators*: they can manage the platform having permission to modify or delete any registered users and projects.

4. *Developers*: they can implement and add new methods to the catalog.

The *Method Catalog* (Figure 4.2) has the available methods created by the developers of the framework to use in projects. Each project has its own information, including the privacy setting, public or private. The *My Projects* area allows the user to create and manage projects. The *Public Projects* area provides access to public projects that were shared by other users. They can be opened and duplicated to other projects. Private projects are just available for the own user.



Methods

Local Methods

Additive Aggregation
Orders a given group of Options based on the given Criteria. Each option has a value for each criteria.

CAT-SD
CAT-SD (CATegorization by Similarity-Dissimilarity) is a multiple criteria decision aiding method for nominal classification problems.

DCM-SRF
The Deck Cards Method (DCM) is mainly designed for determining the weights of criteria using the revised Simos' procedure ('SRF method').

ELECTRE TRI-nC
A multiple criteria sorting method where each category is characterized by several reference actions.

Order By
Orders a given group of actions by a certain criterion in a specific direction.

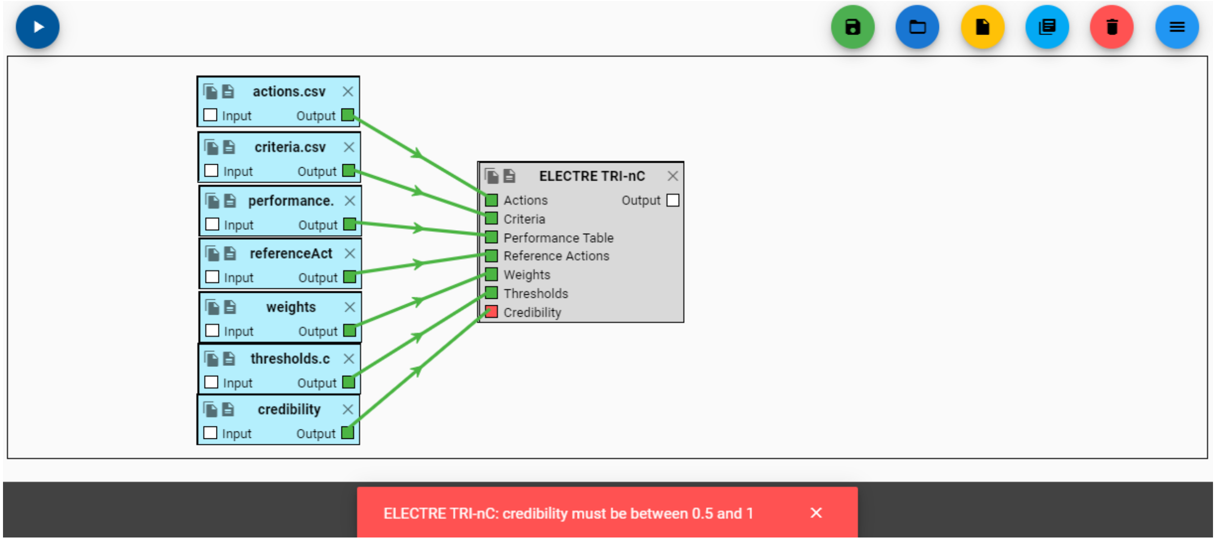
Sort
List various objects and then reorder them by their importance.

Figure 4.2: DECSPACE method catalog.

Any method on the catalog can be used in the workspace area, by dragging, dropping and properly connecting themselves and data modules in an intuitive and interactive graphical user interface that allows the user to get feedback (Figure 4.3). This is the most important area, where all the technical work happens. These workflows can be saved, executed, deleted and exported. The user can also select the project version to retrieve a previous version saved.

The input data and preference information can be manually provided in the modules or be imported as a CSV and JSON files. A `.zip` file can also be imported, containing a workflow that was already used and exported. A workflow example is presented in Section 4.3.

As a developer user, that can implement and add new methods to the catalog, the ELECTRE TRI-nC implementation is presented in Section 4.2.



ELECTRE TRI-nC: credibility must be between 0.5 and 1 ×

Figure 4.3: DecSpace workspace.

4.2 ELECTRE TRI-nC implementation

A MCDA method in DecSpace can be split in three main components:

1. *Module*: the module is a small, draggable box that can be moved and dropped inside the workspace. A module has a name, a delete button, input slots and an output slot.

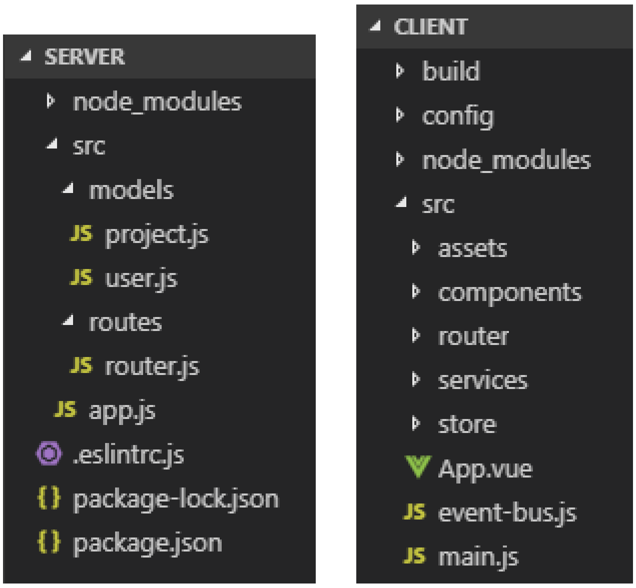
2. *Modal*: the modal is a window with a graphical interface specifically designed for each method. It can be accessed by clicking one of the input slots of the module. For most methods, an interactive table should be enough for manual data input, but more complex and dynamic interfaces can be implemented.

3. *Service*: the service represents the algorithm of the MCDA method itself. It receives as input the parameters of that method and when the user executes the workflow it returns an output.

The module is the virtual representation of the method. The modal is triggered when the user interacts with the module and then the user can insert data by interacting with the interface. These components are what makes DecSpace a user-friendly platform. The service is not visible to the user, just the final result (the output of the method) appears.

To implement a method, the structure of the platform has to be known, in order to change the code and add files in the right place. A high-level view of the directory structure of the DecSpace source code is displayed in Figure 4.4.

Average knowledge of the JavaScript programming language and `Vue.js` library should be enough to develop a method for the platform. For a MCDA method developer, the folders and files that have to be taken into account are the following:



```
◢ SERVER
  ▹ node_modules
  ◢ src
    ◢ models
      JS project.js
      JS user.js
    ◢ routes
      JS router.js
    JS app.js
  .eslintrc.js
  {} package-lock.json
  {} package.json
```

(a) Back-end directory.

```
◢ CLIENT
  ▹ build
  ▹ config
  ▹ node_modules
  ◢ src
    ▹ assets
    ▹ components
    ▹ router
    ▹ services
    ▹ store
    App.vue
    JS event-bus.js
    JS main.js
```

(b) Front-end directory.

Figure 4.4: Directory structure of DECSPACE source code.

- `/client/src/components/Workspace/Methods/`: this is where the module and modal files of the method are placed.

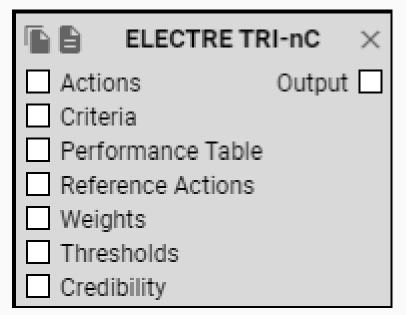
- `/client/src/services/`: this is where the algorithm implementation of the method is placed.

- `/client/src/components/Workspace/Workspace.vue`: this is the main workspace controller, where few lines of code to link the method are added.

- `/client/src/components/Catalog/Catalog.vue`: this is where the method and basic information about it are added to the method catalog.

Firstly, a file named `ELECTRETRINCModule.vue` is created and added to `/client/src/components/Workspace/Methods/`. In this file the input parameters of the box are defined. According to the flowchart presented in Section 3.4, the module has to receive endpoints for the data and preference parameters, as displayed in Figure 4.5.

To create the seven inputs, buttons are needed to state. For each button a `<button>` tag and a `<span>` tag (for the text) are needed. A part of the code, for the actions entry for instance, is presented in Listing 4.1.



ELECTRE TRI-nC

□ Actions Output □

□ Criteria

□ Performance Table

□ Reference Actions

□ Weights

□ Thresholds

□ Credibility

Figure 4.5: DECSPACE module of ELECTRE TRI-nC method.

Listing 4.1: Part of DECSPACE module code of ELECTRE TRI-nC method.

```
<button type='button'
    class='module-circle module-circle-input'
    style='margin-top: 28px; left: 5px'
    :id="id+'-actions'"
    @click="showModal('actions')">
</button>
<span class='module-circle-span'
    style='margin-top: 26px; left: 22px'
    @click="showModal('actions')">
    Actions
</span>
```

For the other entries, the code is easily adaptable. The margin-top property should have 17px more for each button. The output button is automatically generated because is on all modules by default.

Each input on-click should open the respectively tab of the modal. A tab for each of the parameters should be created. For that, a file named `ELECTRETRINCModal.vue` is created and added to `/client/src/components/Workspace/Methods/`. To receive and validate the data, a table for each tab is mainly used, as displayed in Figure 4.6.

Each tab of the modal contains in general the following components: headers, data entries, option to remove an entry and option to add an entry. Each component has to be coded. Note that the available templates and methods already implemented turn this process easier.

The actions tab code part is presented in Listing 4.2. For each tab of the modal the procedure is identical, adapted to the required parameters.

In the modal file, a `close()` function is also defined to run whenever the modal is closed. The function above calls the `saveModuleData()` function that sends the updated data to the workspace. All the modules generated have an object in the workspace that must be correctly updated by the modal.

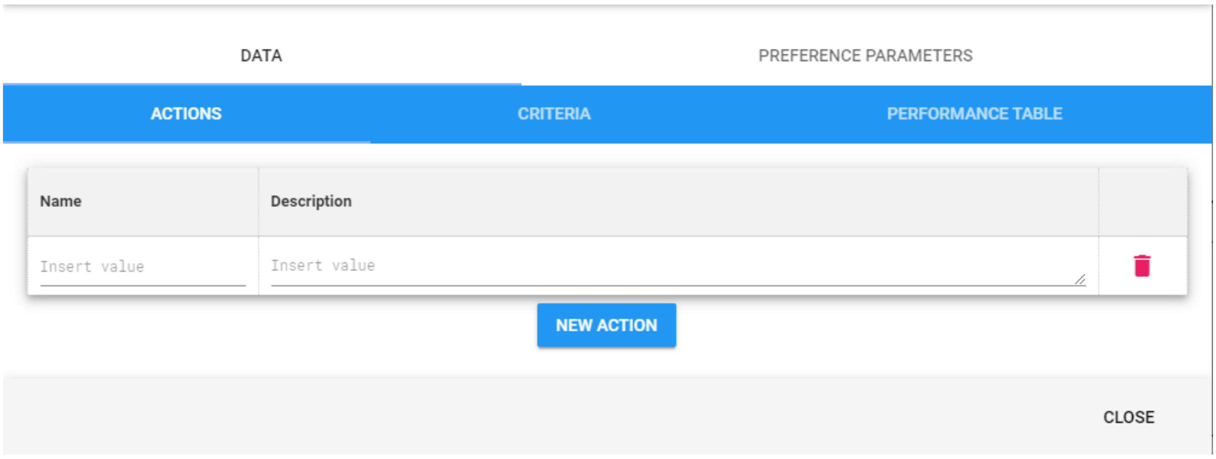


Figure 4.6: DecSpace modal tab of ELECTRE TRI-nC method.

The `saveModuleData()` function saves in the `input` property the data of the input parameters of that method. In the `output` property, `headers` and `description` are the information that will appear on the resulting output data module. In this file, error messages are also defined to appear if the data provided are not the expected, in order to have an interactive graphical user interface.

To implement the method itself, a file named `ELECTRETRINCService.js` is created and added to `/client/src/services/`. In this file, after receiving and verifying all the inputs, the calculations are done and the results are saved. This algorithm is implemented in JavaScript and follows the calculation steps presented in Section 3.4. The code is available on Appendix A.

To add the developed method to the catalog, in the file `/client/src/components/Catalog/Catalog.vue`, a new object is added to the methods array in the variables. The `active` property defines whether the method is available or not, the `name` property is the name of the method, the `type` property is a lower case code of the method used to identify it (and generate IDs), the `reference` property is a link to the method catalog page and the `description` property is a short description of the method.

Finally, to link everything and make the workspace know about the method, the file `/client/src/components/Workspace/Workspace.vue` is edited: few lines are added to the `<script>` part to import the module, modal and service and then these components are associated to the method `type` property.

Listing 4.2: Part of DecSpace modal code of ELECTRE TRI-nC method.

```
<v-tab-item key="actions">
  <div id="subheader">
    <v-data-table id="actions-table" :items="actions" hide-actions class="elevation-5">
      <template slot="no-data"><tr><td colspan="100%" class="text-xs-center">Empty</td></tr></template>
      <template slot="headers" slot-scope="props">
        <th class="text-xs-left" style="width: 20%">Name</th>
        <th class="text-xs-left">Description</th>
        <th style="width: 70px;"></th>
      </template>
      <template slot="items" slot-scope="props">
        <td>
          <v-text-field placeholder="Insert value" v-model="props.item['Name']" hide-details></v-text-field>
        </td>
        <td>
          <v-textarea placeholder="Insert value" v-model="props.item['Description']" hide-details rows="1" maxlength="150"></v-textarea>
        </td>
        <td>
          <v-btn icon @click="removeActionsLine(props.item)"><v-icon color="pink">delete</v-icon></v-btn>
        </td>
      </template>
    </v-data-table>
    <v-btn @click="addActionsLine(null)" color="info">New Action</v-btn>
  </div>
</v-tab-item>
```

4.3 Building an ELECTRE TRI-nC workflow

In order to use ELECTRE TRI-nC in DecSpace, the user after creating a project, that is displayed in the workspace, must choose the ELECTRE TRI-nC method among the available methods by clicking on the *Add Method* button of the *Method Selection* menu. Immediately, the method module appears in the workspace area. To create the workflow, all required data and preference parameters need to be linked to that module. As explained in Section 4.1, this can be done by manually inserting the data in the modal, after clicking on the buttons of the module, or by uploading correctly structured CSV or JSON files, which appear as data modules. These files must have the needed data and table headers (if applied), to be then properly connected to the method module.

As presented in Section 3.4, the input block contains the main data (actions, criteria and performance table) and the preference parameters (reference actions, criteria weights, thresholds and credibility level), which are contained in the method module.

More specifically, the user can provide the following input information to the method module:

1. *Data*:

   (a) *Actions*. In this table, the main information about an action is needed:

   - *Name*: it is the name of the action.
   - *Description*: it is the respective action description (not mandatory).

   (b) *Criteria*. In this table, the following columns appear by default:

   - *Name*: it is the name of the criterion.
   - *Description*: it is the respective criterion description (not mandatory).
   - *Direction*: it corresponds to the criterion preference direction (*Maximize* or *Minimize*).
   - *Scale Type*: it is related to criterion performance levels (*Cardinal* or *Ordinal*).
   - *Min*: if the scale type is *Cardinal*, then a minimum value for the performance levels should be provided.
   - *Max*: if the scale type is *Cardinal*, then a maximum value for the performance levels should be provided.
   - *Num Levels*: If the scale type is *Ordinal*, then the total number of scale levels should be provided.

   (c) *Performance Table*. The rows correspond to the actions names and the columns corresponds to the criteria names. Performance levels to each criterion must be provided for each action (the platform verifies if they respect the criteria scales characteristics).

2. *Preference Parameters*:

    (a) *Reference Actions*. For each reference action, the following information is needed:

    - *Category*: the name of the category it belongs to.
    - *Name*: the name of the reference action.
    - *Performance Levels*: performance levels have to be fulfilled in each criterion column.

    (b) *Criteria Weights*. For each criterion, a value of the criterion weight must be given.

    (c) *Thresholds*. For each criterion, the preference and indifference thresholds must be defined. The veto threshold is not mandatory.

    (d) *Credibility Level*. A value between $[0.5, 1]$ has to be given.

The constructed workflow can be executed when all connections are properly done and the required data are provided. If the required data has errors, a message appears to the user (as displayed in Figure 4.3). Executing the constructed workflow, the results obtained (method output) appear as a data module in the workspace that contains a table with the classification of the actions into the considered categories, according to the possible assignments described in Section 3.3. By clicking on the output module box, the corresponding results can be visualized and analyzed by the user. Note that this output module can be used as input module of another method. The resulting workflow is presented in Figure 4.7.

The workflow, including input files and results, can be exported as a `.zip` file containing CSV files. It can only be saved by a registered user.

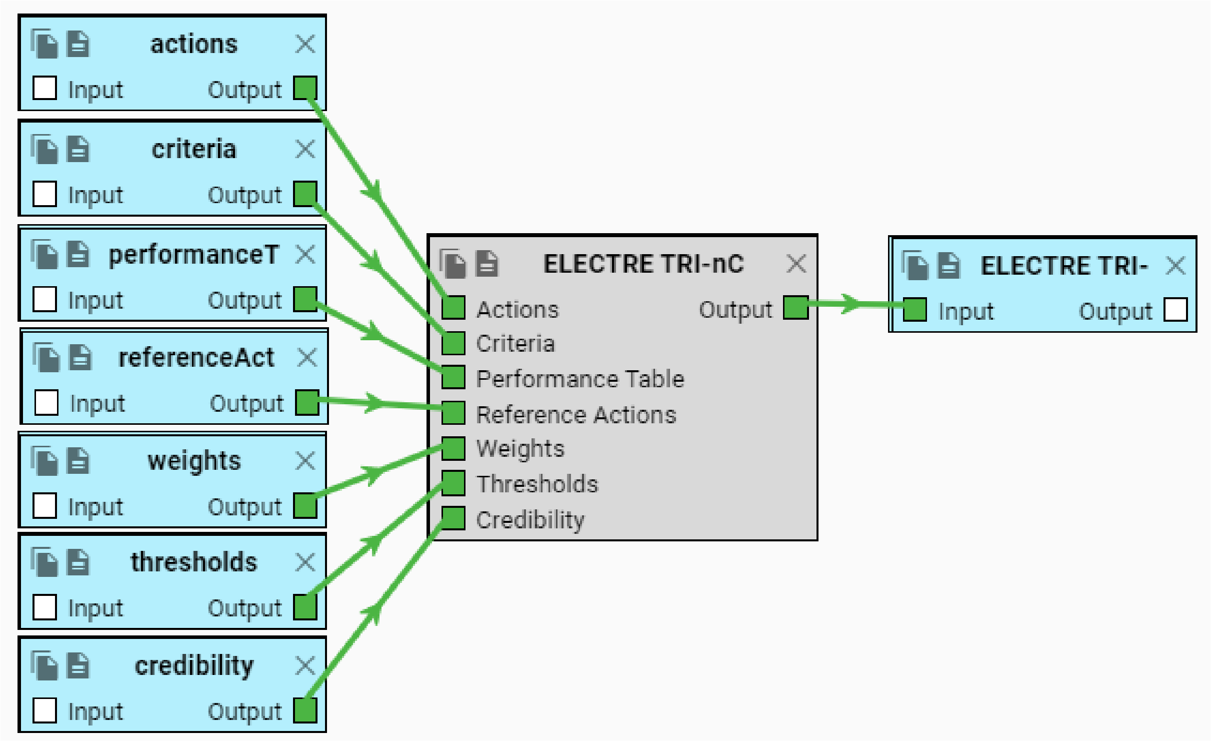


Figure 4.7: DECSPACE workflow of ELECTRE TRI-nC method.

4.4 Validating the implementation

In this section, the validation of the implemented method in DECSPACE is presented by testing it against a numerical example [3]. The assignment results and intermediate results (partial concordance, global concordance, partial discordance, credibility and categorical indices) obtained in MCDA-ULAVAL and DECSPACE have to be equal, by providing the same data and preference parameters from the numerical example.

The numerical example consists on the assignment of projects, in order to allocate them to different categories according to their performance in a set of criteria. The 5 predefined categories are the following: *Excellent*, *Good*, *Moderate*, *Weak* and *Bad*. Each project has to be evaluated on a set of 7 criteria, within the range $[0, 100]$. The set of criteria weights and thresholds are presented in Table 4.1.

Table 4.1: Criteria weights and thresholds for the numerical example.

Parameters	g_1	g_2	g_3	g_4	g_5	g_6	g_7
Weight, w_j	20	15	10	10	10	15	20
Indifference, q_j	5	5	5	5	5	5	5
Preference, p_j	10	10	10	10	10	10	10
Veto, v_j	35	35	$-$	$-$	$-$	35	35

The set of categories can be characterized as following (Table 4.2):

- A *Bad* project by the performances of 5 on all the criteria.
- A *Weak* project by the performances of 20 on all the criteria as representative of the lower part, and 30 on all the criteria as representative of the upper part of this category.
- A *Moderate* project by the performances of 40 on all the criteria as representative of the lower part, and 65 on all the criteria as representative of the upper part of this category.
- A *Good* project by the performances of 85 on the first 4 or last 4 criteria, and at least 70 on the remaining 3 criteria. A project with performances of 75 on all the criteria can also characterize this category.
- An *Excellent* project by the performances of 95 on all the criteria.

Table 4.2: Characteristic reference actions for the numerical example.

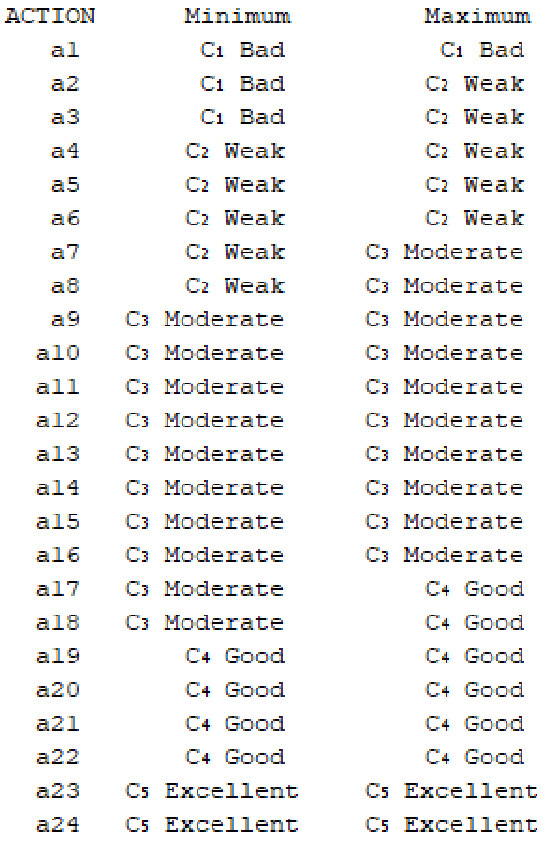
Category		g_1	g_2	g_3	g_4	g_5	g_6	g_7
Bad	b_1	5	5	5	5	5	5	5
Weak	$b_{2,1}$	20	20	20	20	20	20	20
	$b_{2,2}$	30	30	30	30	30	30	30
Moderate	$b_{3,1}$	40	40	40	40	40	40	40
	$b_{3,2}$	65	65	65	65	65	65	65
Good	$b_{4,1}$	70	70	70	85	85	85	85
	$b_{4,2}$	85	85	85	85	70	70	70
	$b_{4,3}$	75	75	75	75	75	75	75
Excellent	b_5	95	95	95	95	95	95	95

For this numeric example, 24 projects having easily interpretative performances on the set of criteria are presented in Table 4.3.

The assignment results for the set of projects, $\lambda = 0.65$, are displayed in Figure 4.8. Besides that, the following variations were successfully tested: criteria direction (maximize/minimize), criteria scale type (cardinal/ordinal) and veto thresholds (with/without).

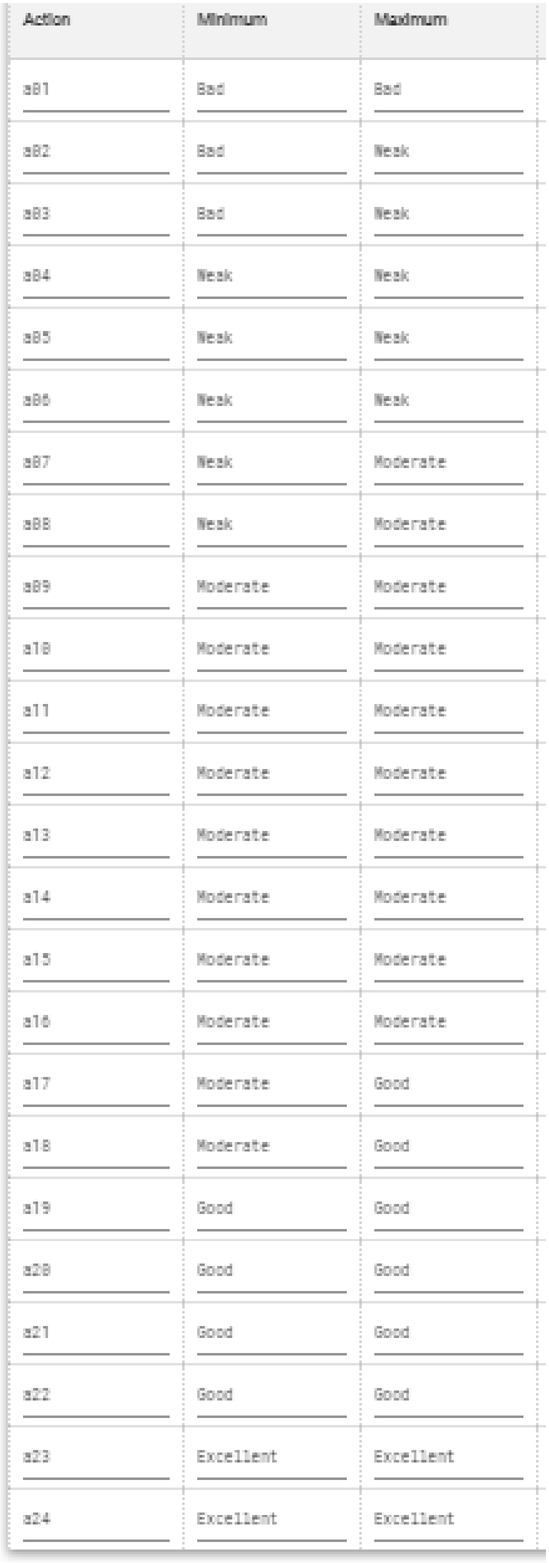
Table 4.3: Set of projects for the numerical example.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
a_1	10	10	10	10	10	10	10
a_2	10	20	20	10	20	20	10
a_3	15	5	10	15	10	5	15
a_4	15	15	15	15	15	15	15
a_5	20	20	50	50	50	20	20
a_6	30	30	45	45	45	30	30
a_7	5	50	50	50	50	50	90
a_8	35	35	35	35	35	35	35
a_9	25	25	25	50	50	50	50
a_{10}	30	30	30	40	40	40	40
a_{11}	30	45	45	45	45	45	30
a_{12}	35	35	35	45	45	45	45
a_{13}	35	35	35	70	70	70	70
a_{14}	45	45	30	30	30	45	45
a_{15}	65	25	25	25	25	25	65
a_{16}	85	85	50	50	50	15	15
a_{17}	65	65	85	85	85	65	65
a_{18}	70	70	70	70	70	70	70
a_{19}	70	70	70	95	95	95	95
a_{20}	75	75	75	80	80	80	80
a_{21}	80	80	80	80	80	80	80
a_{22}	85	50	85	85	85	50	85
a_{23}	75	75	75	95	95	95	95
a_{24}	90	90	80	80	80	90	90



```
ACTION      Minimum          Maximum
   a1          C1 Bad          C1 Bad
   a2          C1 Bad        C2 Weak
   a3          C1 Bad        C2 Weak
   a4         C2 Weak        C2 Weak
   a5         C2 Weak        C2 Weak
   a6         C2 Weak        C2 Weak
   a7         C2 Weak    C3 Moderate
   a8         C2 Weak    C3 Moderate
   a9     C3 Moderate    C3 Moderate
  a10     C3 Moderate    C3 Moderate
  a11     C3 Moderate    C3 Moderate
  a12     C3 Moderate    C3 Moderate
  a13     C3 Moderate    C3 Moderate
  a14     C3 Moderate    C3 Moderate
  a15     C3 Moderate    C3 Moderate
  a16     C3 Moderate    C3 Moderate
  a17     C3 Moderate        C4 Good
  a18     C3 Moderate        C4 Good
  a19         C4 Good        C4 Good
  a20         C4 Good        C4 Good
  a21         C4 Good        C4 Good
  a22         C4 Good        C4 Good
  a23    C5 Excellent   C5 Excellent
  a24    C5 Excellent   C5 Excellent
```

(a) MCDA-ULAVAL output.



Action	Minimum	Maximum
a01	Bad	Bad
a02	Bad	Weak
a03	Bad	Weak
a04	Weak	Weak
a05	Weak	Weak
a06	Weak	Weak
a07	Weak	Moderate
a08	Weak	Moderate
a09	Moderate	Moderate
a10	Moderate	Moderate
a11	Moderate	Moderate
a12	Moderate	Moderate
a13	Moderate	Moderate
a14	Moderate	Moderate
a15	Moderate	Moderate
a16	Moderate	Moderate
a17	Moderate	Good
a18	Moderate	Good
a19	Good	Good
a20	Good	Good
a21	Good	Good
a22	Good	Good
a23	Excellent	Excellent
a24	Excellent	Excellent

(b) DECSPACE output.

Figure 4.8: Results obtained for the numerical example ($\lambda = 0.65$).

The objective of the validation is verified: the results are correctly obtained in DECSPACE. Therefore, in the next chapter, a demonstration of ELECTRE TRI-nC in DECSPACE for a real case is provided. This case was motivated by the project hSNS focused on the Portuguese public hospitals performance assessment.

4.5 Summary

DECSPACE is an innovative web-based platform that allows the efficient use of MCDA methods by giving the possibility of building decision models in an intuitive graphical user interface. The MCDA methods available are presented in *Method Catalog* and can be used in projects by dragging, dropping and properly connecting themselves and data modules. This can be done in *Workspace*, the most important area, where the user can build, save and execute their workflows.

A MCDA method in DECSPACE can be split in three main components: *Module*, *Modal* and *Service*. The module is a small draggable box, the virtual representation of the method. The modal is the interface that allows the user to insert manual data triggered when the user interacts with the module. The service is the algorithm of the method itself, not visible to the user.

Average knowledge of JavaScript programming language, `Vue.js` library, folders and files that have to be changed, should be enough to develop a method for the platform. Each one of the MCDA components and their linkage has to be implemented. Afterwards, the developed method can be added to the catalog and used in projects.

By testing the implemented method in DECSPACE against a numerical example, the implementation can be validated if the intermediate and final results are correctly obtained. The ELECTRE TRI-nC was successfully implemented in DECSPACE.

5

Demonstration

Contents

5.1 Case description 45

5.2 Data and sample 46

5.3 Preference parameters 51

5.4 Results and discussion 54

This chapter presents the case adopted to demonstrate the use of ELECTRE TRI-nC method in DECSPACE. It starts with a brief description of the case (Section 5.1), then presents the data (Section 5.2) and procedures to define all the preference parameters (Section 5.3). Finally, the results are analysed and aim of discussion in Section 5.4. The interfaces of the case in DECSPACE can be seen in Appendix B.

5.1 Case description

This section defines the real case of Portuguese public hospital performance assessment. It is divided in two subsections, the first introduces the Portuguese health paradigm (mainly based on a Portuguese health system review book [22]) and the second describes performance assessment in healthcare.

5.1.1 Background

The current Portuguese National Health Service (NHS) was established in 1979 in order to provide the appropriate and equitable care to all citizens and is tendentiously free (because co-payments are charged to some customers). This means that the system must be financially sustainable, in particular note that the hospitals consume more than half of the public health expenses. In view of that, several health reforms have been introduced to reduce costs and waste of public funds, while the efficiency and the effectiveness of healthcare providers were expected to improve. These reforms include the creation of Public Enterprise (EPE) hospitals, hospital centers, primary care local health units and Public–Private Partnerships (PPP).

Primary care reform local health units were created in 1999 to enhance greater and better communication between primary care and hospitals, through a vertical integration of different levels of care. At the start of the twenty-first century, the government increased private sector involvement in the building, maintaining and operating of health facilities under PPP. The creation of hospital centers allows better coordination between institutions providing hospital care in the same geographical area. Both hospital centers and hospitals are EPEs, meaning that hospital boards have some autonomy.

Portugal is progressively becoming more transparent with real time information and an extensive information infrastructure, which plays a central role in monitoring the health system performance.

5.1.2 Performance assessment in healthcare

Performance in healthcare, as in other services industry, can be defined as as appropriate combination of efficiency and effectiveness. To be efficient a healthcare provider must has a certain level of quality care using the minimum combination of resources. Many times, productivity and efficiency are used with the same purpose. Effectiveness, more specifically, evaluates the outcomes of medical care [23].

A way of assessing performance in healthcare is by measuring the quality of medical care according to the three interrelated categories: outcomes, process and structure [24]. CARE APPROPRIATENESS can be identified as an indicator of outcomes and process quality. Appropriateness of healthcare regards the ability of providing patient-centred care services supported by evidence-based medicine [25].

In other hand, the Portuguese NHS provides universal care, therefore a citizen should has access to a particular service whenever it is necessary or intended. To measure access, some dimensions should be taken into account, such as TIMELINESS OF SERVICES and SERVICES AVAILABILITY. Timeliness of services refers to the capacity of delivering healthcare services whenever required in proper time, and services availability regards the existence of disposable resources to be used when necessary [25].

The system must be financially sustainable, hence the problem of ECONOMIC EFFICIENCY can also be considered as a measurable dimension of performance.

Data Envelopment Analysis (DEA) is a linear programming methodology that is the leading performance evaluation approach commonly used to measure the efficiency of hospitals, by evaluating the relative efficiency of hospitals using multiple inputs to produce multiple outputs. A DEA model was used to assess Portuguese public hospital performance in a previous thesis [26]. In this chapter, a MCDA sorting method to assign each hospital to the most appropriate category is presented. For this context, the DM chosen is specialized in health administration.

5.2 Data and sample

All data (hospitals, indicators and performance) are collected from the official benchmarking database, maintained by the Portuguese Central Administration of the Health System at http://benchmarking.acss.min-saude.pt/. This database has a total of 43 institutions in 5 groups using hierarchical clustering and a total of 34 indicators grouped in 6 benchmarking dimensions.

The process of data selection to define the sample of interest to this case is presented in the following subsections.

5.2.1 Actions

The actions to be assigned to the predefined categories are defined as Portuguese public hospitals. The clustering groups are not considered in this analysis because the object of study are hospitals as individual institutions. The local health units include primary care units, so they are not considered. Hospitals managed as PPP are also not considered because they have a considerable lack of public data. The oncology institutions (IPOs), despite being public hospitals, are very specialized institutions with their specific technology of production [27], so they are not considered, as well.

This process leads to a sample composed by a total of 28 institutions (7 hospitals and 21 hospital centers). The Portuguese public hospitals under assessment and the corresponding actions notation, a_i with $i = 1, ..., 28$, are presented in Table 5.1.

5.2.2 Criteria

The indicators in the database are organized in individual benchmark dimensions: ACCESS, PERFORMANCE ASSISTANCE, SAFETY, VOLUME AND USAGE, PRODUCTIVITY, and ECONOMIC-FINANCIAL. There are, at least, two different procedures to deal with these indicators: 1. Considering individual benchmark dimensions as criteria and indicators as sub criteria (criteria are structured hierarchically [28]); 2. Mixing indicators considering that each one describes a criterion, not considering dimensions, and applying the method once to classify the performance in general (criteria at the same level). A parallel study [29] is using the first methodology. Therefore, the second procedure will be adopted in this thesis to method demonstration.

In this demonstration, the indicators from SAFETY and VOLUME AND USAGE dimensions, as well as childbirth related indicators, are not considered due to their values being almost null. In the ECONOMIC-FINANCIAL dimension the most representative indicator, OPERATION COSTS, was chosen.

This process leads to a sample composed by a total of 9 indicators describing 9 criteria to assess the Portuguese public hospitals as presented in Table 5.2.

The indicators are summary described as follows:

- *Number of non-urgent first medical appointments within adequate time per* 100 *first medical appointments* is the amount of first medical appointments that occur in guaranteed maximum response times compared to the total number of first medical appointments;

- *Number of scheduled surgeries within adequate time per* 100 *first medical appointments* is the amount of patients enrolled to surgeries in guaranteed maximum response times compared to the total number of patients enrolled to surgeries;

- *Number of scheduled hip surgeries performed in the first* 48 *hours per* 100 *hip surgeries* assesses

Table 5.1: Portuguese public hospitals under assessment.

Action	Hospital
a_1	Centro Hospitalar Barreiro/Montijo, EPE
a_2	Centro Hospitalar de Leiria, EPE
a_3	Centro Hospitalar de Lisboa Ocidental, EPE
a_4	Centro Hospitalar de Setúbal, EPE
a_5	Centro Hospitalar do Baixo Vouga, EPE
a_6	Centro Hospitalar do Médio Ave, EPE
a_7	Centro Hospitalar do Oeste, EPE
a_8	Centro Hospitalar e Universitário de Coimbra, EPE
a_9	Centro Hospitalar Entre Douro e Vouga, EPE
a_{10}	Centro Hospitalar Médio Tejo, EPE
a_{11}	Centro Hospitalar Póvoa de Varzim/Vila do Conde, EPE
a_{12}	Centro Hospitalar Tâmega e Sousa, EPE
a_{13}	Centro Hospitalar Tondela-Viseu, EPE
a_{14}	Centro Hospitalar Trás-os-Montes e Alto Douro, EPE
a_{15}	Centro Hospitalar Universitário Cova da Beira, EPE
a_{16}	Centro Hospitalar Universitário de Lisboa Central, EPE
a_{17}	Centro Hospitalar Universitário de São João, EPE
a_{18}	Centro Hospitalar Universitário do Algarve, EPE
a_{19}	Centro Hospitalar Universitário do Porto, EPE
a_{20}	Centro Hospitalar Universitário Lisboa Norte, EPE
a_{21}	Centro Hospitalar Vila Nova de Gaia/Espinho, EPE
a_{22}	Hospital da Senhora da Oliveira, Guimarães, EPE
a_{23}	Hospital Distrital da Figueira da Foz, EPE
a_{24}	Hospital Distrital de Santarém, EPE
a_{25}	Hospital Espírito Santo de Évora, EPE
a_{26}	Hospital Fernando Fonseca, EPE
a_{27}	Hospital Garcia de Orta, EPE
a_{28}	Hospital Santa Maria Maior, EPE

the percentage of elderly patients ($\geqslant 65$ years old) with hip surgeries within 48 hours after fracture (important to prevent infections);

- *Average number of days after admission waiting before surgery* is the waiting time, inside the hospital, before the surgery;

- *Number of outpatient surgeries per* 100 *potential outpatient procedures* concerns the amount of outpatient surgeries within the total of outpatient procedures (important to reduce inpatients);

- *Number of readmissions in* 30 *days after discharge per* 100 *inpatients* is the percentage of patients readmitted within 30 days after discharge;

Table 5.2: Criteria, indicators and corresponding direction preferences.

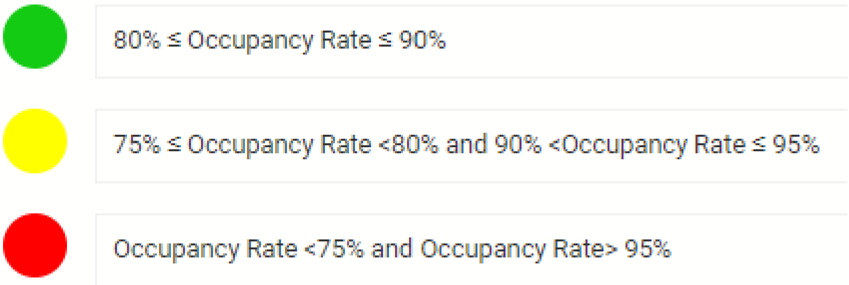
Criterion	Indicator	Direction
g_1: TIMELINESS OF MEDICAL APPOINTMENTS	Number of non-urgent first medical appointments within adequate time per 100 first medical appointments	Maximization
g_2: TIMELINESS OF SURGERIES	Number of scheduled surgeries within adequate time per 100 surgeries	Maximization
g_3: READINESS OF HIP SURGERIES	Number of scheduled hip surgeries performed in the first 48 hours per 100 hip surgeries	Maximization
g_4: WAITING TIME BEFORE SURGERY	Average number of days after admission waiting before surgery	Minimization
g_5: OUTPATIENT SURGERIES ADEQUACY	Number of outpatient surgeries per 100 potential outpatient procedures	Maximization
g_6: READMISSIONS	Number of readmissions in 30 days after discharge per 100 patients	Minimization
g_7: DELAY OF CARE	Number of inpatients staying more than 30 days per 100 admissions	Minimization
g_8: BED OCCUPANCY	Absolute difference in inpatient bed annual occupancy rate to a reference value of 85%	Minimization
g_9: OPERATION COSTS	Operation costs per standard patient	Minimization

- *Number of inpatients staying more than* 30 *days per* 100 *admissions* handles the quantity of inpatient admissions longer than 30 days in comparison with the total number of inpatient episodes;

- *Absolute difference in inpatient bed annual occupancy rate to a reference value of* 85% measures how far is the occupancy of inpatient beds for a reference value (note that the original indicator *Inpatient bed annual occupancy rate* can not be maximized or minimized due to the trade-off between PRODUCTIVITY and ACCESS illustrated in the benchmarking database, Figure 5.1: for a occupancy bed rate $< 75\%$ the hospital is inefficient and for a value $> 95\%$ it is almost full);

- *Operating costs per standard patient* is rather self-explanatory.

The considered criteria follow the established in Subsection 5.1.2: TIMELINESS OF MEDICAL APPOINTMENTS and TIMELINESS OF SURGERIES are associated to TIMELINESS OF SERVICES; READINESS OF HIP SURGERIES, WAITING TIME BEFORE SURGERY, OUTPATIENT SURGERIES ADEQUACY, READMISSIONS and DELAY OF CARE are associated to CARE APPROPRIATENESS; BED OCCUPANCY is associated to SERVICES AVAILABILITY and OPERATION COSTS to ECONOMIC EFFICIENCY.



🟢 80% ≤ Occupancy Rate ≤ 90%

🟡 75% ≤ Occupancy Rate <80% and 90% <Occupancy Rate ≤ 95%

🔴 Occupancy Rate <75% and Occupancy Rate> 95%

Figure 5.1: Chromatic classification system for occupancy rate.

5.2.3 Performance table

The time interval established to be subject of this analysis was the last year: 2018. Just some exceptions occurred due to the lack of data and in the following cases the data used are from the year 2017: $g_8(a_{28})$, $g_3(a_{19})$, $g_6(a_7)$, $g_9(a_i)$ with $i = 1, ..., 28$. Note that $g_j(a_i)$ denotes the performance of the action a_i in the criterion g_j.

The performance of 28 hospitals in 9 criteria results in a 28×9 matrix, with 252 entries, as displayed in Table 5.3.

Table 5.3: Performance table.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
a_1	82.90	72.16	45.98	0.98	73.14	8.38	4.91	0.11	3328
a_2	56.10	76.34	27.21	0.75	90.37	8.43	2.29	4.52	2828
a_3	70.05	61.39	51.22	1.34	82.04	6.88	4.35	4.65	3259
a_4	74.16	60.82	75.00	1.00	90.26	5.91	3.67	1.92	3658
a_5	66.31	85.52	66.11	0.47	82.02	6.95	2.95	0.26	3208
a_6	75.86	91.09	26.45	0.75	83.86	7.31	3.94	3.41	3265
a_7	55.37	68.71	37.65	0.88	75.48	7.73	3.00	1.71	3138
a_8	65.97	61.39	41.03	1.29	82.15	8.72	4.01	6.84	3006
a_9	58.81	80.77	21.96	0.62	84.72	6.92	2.65	4.62	3053
a_{10}	83.86	72.94	28.74	0.62	86.78	9.43	4.02	8.32	3619
a_{11}	95.93	98.39	84.38	0.55	71.31	6.39	1.53	7.61	3255
a_{12}	59.84	83.82	60.38	0.84	85.97	6.68	3.48	4.94	2781
a_{13}	80.27	48.21	35.42	1.71	95.55	5.31	3.37	3.42	2852
a_{14}	64.08	69.61	73.72	1.08	86.47	11.25	2.89	3.22	3071
a_{15}	76.06	67.70	53.04	0.65	70.24	7.05	3.02	6.86	3716
a_{16}	75.08	58.13	35.90	1.28	84.38	6.13	4.66	7.12	3190
a_{17}	49.94	72.29	65.05	0.83	84.06	4.80	3.01	5.16	2740
a_{18}	73.14	71.01	12.45	1.39	80.21	6.84	5.63	4.18	3558
a_{19}	72.72	78.91	30.00	0.30	85.40	6.87	1.90	11.8	3090
a_{20}	63.54	63.59	44.78	1.00	84.48	8.76	4.36	1.53	3173
a_{21}	55.52	80.06	71.91	0.95	84.50	7.27	3.71	1.87	2947
a_{22}	57.12	75.66	41.42	0.58	79.65	8.03	4.61	3.20	2921
a_{23}	79.47	97.93	39.47	0.90	87.07	10.00	3.12	8.12	2802
a_{24}	67.22	61.50	31.36	0.47	82.20	9.19	2.93	8.55	3863
a_{25}	61.68	70.58	11.11	0.42	78.50	6.07	3.99	3.44	3241
a_{26}	76.35	65.95	31.49	0.53	82.49	6.47	5.40	3.74	2779
a_{27}	86.47	46.93	23.55	1.29	90.50	7.04	4.48	3.52	2772
a_{28}	85.95	99.01	40.52	0.50	86.53	8.36	1.75	13.45	2740

5.3 Preference parameters

As referred in Section 3.1, this method follows a decision aiding constructive approach, through the interaction between the analyst(s) and the DM(s). This interaction is necessary to define the following preference parameters: categories and their reference actions, criteria weights, thresholds and credibility level.

5.3.1 Categories and reference actions

The DM defined five categories a priori: *Very Weak*, *Weak*, *Neutral*, *Good*, *Very Good* and for each one, one or more (characteristic) reference actions and their performance in the criteria were defined. This information is presented in Table 5.4.

Table 5.4: Categories, reference actions and their performance for each criterion.

Category		g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
Very Weak	b_1	55	65	30	1.4	70	10	6	12	3500
Weak	$b_{2,1}$	70	70	40	1.0	75	9	5	9	3250
	$b_{2,2}$	65	75	40	1.0	75	9	5	9	3250
Neutral	b_3	80	80	50	0.8	80	8	4	6	3100
Good	$b_{4,1}$	90	85	70	0.5	85	7	3	3	2950
	$b_{4,2}$	85	90	70	0.5	85	7	3	3	2950
Very Good	b_5	95	95	80	0.4	90	6	2	0	2800

5.3.2 Criteria weights

The DECK CARDS METHOD is used to determine the weights of criteria in the ELECTRE type methods using the revised Simos' procedure [30], described as follows:

1. The cards are provided to the DM. Each card has the identification of a criterion.
2. The DM should rank these cards in order of importance. If some criteria have the same importance/weight, a set of cards in the same rank should be built.
3. To establish the difference of importance between successive criteria, white cards can be placed between them. The greater the difference, greater the number of white cards.
4. The last information required is the ratio z, by asking the DM to state how many times the first criterion is more important than the last one in the ranking.

Two different scenarios were taken into account:

- SCENARIO 1: more importance to indicator g_9, because operating costs are the main concern of the NHS sustainability;
- SCENARIO 2: less importance to indicator g_9, because in an ethical and politically correct view of the NHS the operating costs are the last issue.

The preference information in terms of ranking cards, white cards and ratio z for both scenarios provided by the DM is presented in Table 5.5.

The DECK CARDS METHOD is already implemented in DECSPACE and available in *Method Catalog* (Figure 4.2). The interface of this method in DECSPACE for both SCENARIOS 1 and 2 are displayed in Appendix B. The results obtained for the criteria weights are presented in Table 5.6. Note that is not mandatory to obtain normalized weights because the ELECTRE TRI-nC service implemented (Appendix A) normalizes the input weights.

Table 5.5: Ranking of criteria and white cards.

(a) SCENARIO 1.

Rank	Set of cards	White cards
1	g_9	
		3
2	g_1, g_2, g_6	
		1
3	g_5, g_7, g_8	
		2
4	g_3, g_4	
	$z = 3$	

(b) SCENARIO 2.

Rank	Set of cards	White cards
1	g_1, g_2, g_6	
		1
2	g_5, g_7, g_8	
		2
3	g_3, g_4	
		3
4	g_9	
	$z = 3$	

Table 5.6: Criteria weights.

Weights	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
SCENARIO 1	12.91	12.91	6.12	6.12	10.22	12.92	10.22	10.22	18.36
SCENARIO 2	13.97	13.98	8.81	8.81	11.93	13.98	11.93	11.93	4.66

5.3.3 Thresholds

The DM considered that, to associate an action to a category in a criterion, the difference between the action and the reference action of that category must be less than half of the distance between the reference performance of successive categories. The additional veto threshold was assigned with the objective of increasing the power of certain criteria (the most important ones were considered) and to mitigate compensatory effects. Therefore, this process leads to the defined thresholds displayed in Table 5.7.

Table 5.7: Preference, indifference and veto thresholds for each criterion.

Threshold	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
Indifference, q_j	5	5	5	0.1	2.5	0.5	0.5	1.5	75
Preference, p_j	10	10	10	0.2	5.0	1.0	1.0	3.0	150
Veto, v_j	30	30	$-$	$-$	$-$	3.0	$-$	$-$	$-$

5.4 Results and discussion

The chosen credibility level was $\lambda = 0.65$. The results obtained from DecSpace for both considered scenarios are displayed in Table 5.8.

Table 5.8: Assignment results for both scenarios ($\lambda = 0.65$).

Hospital	Scenario 1	Scenario 2
a_1	Weak	Weak
a_2	Neutral	Neutral
a_3	Weak	Neutral
a_4	Neutral	Neutral
a_5	Good	Good
a_6	Neutral	Neutral
a_7	Weak	Weak
a_8	Neutral	Weak
a_9	Neutral	Neutral
a_{10}	Weak	Neutral
a_{11}	Good	Very Good
a_{12}	Good	Neutral
a_{13}	Neutral	Neutral
a_{14}	[Weak,Neutral]	[Weak,Neutral]
a_{15}	Neutral	Neutral
a_{16}	Neutral	Neutral
a_{17}	Neutral	Neutral
a_{18}	Weak	Weak
a_{19}	Neutral	Good
a_{20}	Weak	Weak
a_{21}	Neutral	Neutral
a_{22}	Neutral	Neutral
a_{23}	Neutral	Neutral
a_{24}	Weak	Weak
a_{25}	Neutral	Neutral
a_{26}	Good	Neutral
a_{27}	[Weak,Neutral]	[Weak,Neutral]
a_{28}	Good	Good

As mentioned earlier (Section 3.3), ELECTRE TRI-nC provides a range of possible categories that a hospital should be assigned to (hospitals a_{14}, a_{27}) and when the two possible categories are the same, they are assigned to a unique category (remaining hospitals).

The robustness of the results is demonstrated by analyzing its stability to the change of the preference parameters, namely criteria weights and credibility level. The two scenarios vary in terms of weights, so it allows to test the robustness with the variation of criteria weights. By varying the criteria weights, it was verified that 25% of the assignment results have changed. The importance of the criterion OPERATION COSTS has a small impact in the assignments. For each scenario, the results stability by changing the credibility level to $\lambda = 0.60$ and $\lambda = 0.70$ is tested. For $\lambda = 0.60$, it was verified that only 7.14% and 0% of the assignment results have changed for SCENARIO 1 and SCENARIO 2, respectively. For $\lambda = 0.70$, it was verified that only 3.57% and 0% of the assignment results have changed for SCENARIO 1 and SCENARIO 2, respectively. Through this analysis, it can be concluded that the results obtained are robust and conclusions can be drawn from this.

The hospitals with the worst performance are the following: Centro Hospitalar Barreiro/Montijo, Centro Hospitalar do Oeste, Centro Hospitalar Universitário do Algarve, Centro Hospitalar Universitário Lisboa Norte, Hospital Distrital de Santarém; and the hospitals with the best performances are the following: Centro Hospitalar do Baixo Vouga, Centro Hospitalar Póvoa de Varzim/Vila do Conde, Hospital Santa Maria Maior. The profile of high performing hospitals can be used by other hospitals for benchmarking purposes, allowing them to pursuit continuous improvement, as a conceptual framework [31]. In addition, and according to the objectives of the hSNS project, the results can also be used to: improve the quality of healthcare services delivered, support management by monitoring performance indicators and optimize hospitals funding based on their performance.

The results obtained are consistent with a previous study [26] that uses a DEA methodology, where Centro Hospitalar Barreiro/Montijo, Centro Hospitalar Universitário do Algarve and Centro Hospitalar Universitário Lisboa Norte were also classified as inefficient and Centro Hospitalar do Baixo Vouga, Centro Hospitalar Póvoa de Varzim/Vila do Conde as efficient, too. Note that in that study the year under analysis was 2016 and the hospitals: Centro Hospitalar do Oeste and Hospital Santa Maria Maior were not considered due to lack of data. The results obtained are not comparable with the results from the parallel study [29] that considers criteria structured in a hierarchical way because of the differences in the amount of criteria.

An important issue that should be addressed in future studies is the possible correlation and interaction between some criteria [32].

A possible interesting study could include the hospitals managed as PPP, in order to compare their performance with the public hospitals performance. However, the definition of criteria where the PPP have public data can be a difficult issue.

6

Conclusion and future work

Contents

6.1 Conclusion 59

6.2 Future work 59

6.1 Conclusion

The main focus of this thesis, manifested in Chapter 1, was the implementation of the ELECTRE TRI-nC method in DecSpace and its demonstration to support its application in real cases. To do so, a literature review was carried out in Chapter 2 to be aware of the MCDA concept and the use of this methodologies in healthcare context and in software tools. In that moment, the need for DecSpace was justified, and the conditions to implement the ELECTRE TRI-nC in this platform were fulfilled. This method was described and detailed in Chapter 3 and its implementation in DecSpace was presented in Chapter 4. A real case, in which this framework is used, was presented in Chapter 5 to demonstrate the application of this implementation. The demonstration comprises the Portuguese public hospital performance assessment, with real data from the official benchmarking database, maintained by the Portuguese Central Administration of the Health System, followed by a briefly discussion of the results obtained.

In conclusion, the objectives of this thesis were successfully achieved:

1. The ELECTRE TRI-nC method was implemented in DecSpace and validated against a numerical example;
2. The implemented method was applied to a real case in health sector and conclusions and recommendations were drawn.

Afterwards, multiple applications of this method in different areas can be applied in the future, taking advantage of its implementation in DecSpace, thus continuing the research performed in the last years, as mentioned in Section 3.1.

6.2 Future work

By comparing ELECTRE TRI-nC implementation in DecSpace versus MCDA-ULaval, the first one solves all the cons that was presented to MCDA-ULaval (Table 2.1). However, the following pros: data validation and scenario analysis are not yet in their full potential in DecSpace. Therefore, for the future work of ELECTRE TRI-nC in DecSpace the following hints are presented:

- The data validation of inputs should be improved in order to allow the workflow to be successfully executed only when everything is correctly provided.
- The robustness analysis should be facilitated to allow the user to easily change parameters and see the changes originated.

- The indifference, preference, and veto thresholds have been presented as constants, however, in practice, these thresholds can vary according to the performances $g_j(a)$ [13]. This should be further studied, to understand if the variation should be just linear, and then be implemented in DecSpace with a user-friendly interface.

- The Deck Cards Method, because is used to determine the criteria weights for ELECTRE methods, should has its output consistent with the ELECTRE TRI-nC implementation to allow it to be the input of criteria weights, by connecting it to ELECTRE TRI-nC module, as presented in Figure 6.1. For now, this data input is not recognized, because both tables headers do not match.

- A button to export DecSpace tables to documents should be developed. As can be seen in this thesis, the dimension of data interfaces is not suitable to take screenshots and besides that, is not good practice to do so.

- More ways to visualize the output data, besides the generic data table, should be available to the user. This should be done using `D3.js`, a JavaScript library to manipulate documents based on data [21].

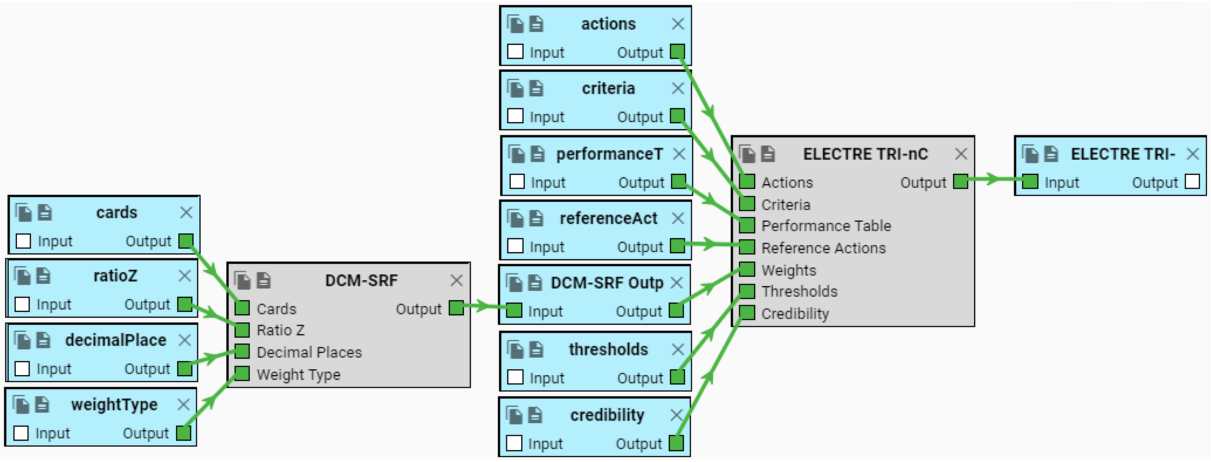


Figure 6.1: DecSpace workflow of Deck Cards Method and ELECTRE TRI-nC method.

More MCDA methods and theses should be developed with this platform to continue the work performed and to increase the DecSpace method catalog.

References

[1] P. Thokala, N. Devlin, K. Marsh, R. Baltussen, M. Boysen, Z. Kalo, T. Longrenn, F. Mussen, S. Peacock, J. Watkins, and M. Ijzerman, "Multiple criteria decision analysis for health care decision making - An introduction: Report 1 of the ISPOR MCDA Emerging Good Practices Task Force," *Value in Health*, vol. 19, no. 1, pp. 1–13, 2016.

[2] J. Mustajoki and M. Marttunen, "Comparison of multi-criteria decision analytical software for supporting environmental planning processes," *Environmental Modelling and Software*, vol. 93, pp. 78–91, 2017.

[3] J. Almeida-Dias, J. R. Figueira, and B. Roy, "A multiple criteria sorting method where each category is characterized by several reference actions: The Electre Tri-nC method," *European Journal of Operational Research*, vol. 217, no. 3, pp. 567–579, 2012.

[4] A. D. Barbosa, "DecSpace : A Multi-Criteria Decision Analysis Framework," *MSc Dissertation, Instituto Superior Técnico, Universidade de Lisboa*, 2017.

[5] J. Figueira, S. Greco, and M. Ehrgott, Eds., *Multiple Criteria Decision Analysis, State of the Art Surveys*. Springer's International Series, 2005.

[6] P. Thokala, R. Baltussen, T. Lönngren, M. Boysen, N. Devlin, S. Peacock, J. Watkins, K. Marsh, Z. Kaló, M. IJzerman, and F. Mussen, "Multiple Criteria Decision Analysis for Health Care Decision Making—Emerging Good Practices: Report 2 of the ISPOR MCDA Emerging Good Practices Task Force," *Value in Health*, vol. 19, no. 2, pp. 125–137, 2016.

[7] J. Wątróbski, J. Jankowski, P. Ziemba, A. Karczmarczyk, and M. Zioło, "Generalised framework for multi-criteria method selection," *Omega*, vol. 86, pp. 107–124, 2019.

[8] J. Figueira, S. Greco, B. Roy, and R. Slowinski, "An overview of ELECTRE methods and their recent extensions," *Journal of Multi-Criteria Decision Analysis*, vol. 20, pp. 61–85, 2013.

[9] D. N. Van Der Goes, N. Edwardson, V. Rayamajhee, C. Hollis, and D. Hunter, "An iron triangle roi model for health care," *ClinicoEconomics and Outcomes Research*, vol. 11, pp. 335–348, 2019.

[10] N. J. Devlin and J. O. N. Sussex, *Incorporating multiple criteria in HTA*. Office of Health Economics, 2011.

[11] P. Meyer and S. Bigaret, "Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA," *Intelligent Decision Technologies*, vol. 6, no. 4, pp. 283–296, 2012.

[12] D. Bouyssou and T. Marchant, “On the relations between ELECTRE TRI-B and ELECTRE TRI-C and on a new variant of ELECTRE TRI-B,” *European Journal of Operational Research*, vol. 242, no. 1, pp. 201–211, 2015.

[13] J. Almeida-Dias, J. R. Figueira, and B. Roy, “Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions,” *European Journal of Operational Research*, vol. 204, no. 3, pp. 565–580, 2010.

[14] J. R. Figueira, J. Almeida-Dias, S. Matias, B. Roy, M. J. Carvalho, and C. E. Plancha, “Electre Tri-C, a multiple criteria decision aiding sorting model applied to assisted reproduction,” *International Journal of Medical Informatics*, vol. 80, no. 4, pp. 262–273, 2011.

[15] F. Macary, J. A. Dias, J. R. Figueira, and B. Roy, “A Multiple Criteria Decision Analysis Model Based on ELECTRE TRI-C for Erosion Risk Assessment in Agricultural Areas,” *Environmental Modeling and Assessment*, vol. 19, no. 3, pp. 221–242, 2014.

[16] A. S. Costa, J. Rui Figueira, C. R. Vieira, and I. V. Vieira, “An application of the ELECTRE TRI-C method to characterize government performance in OECD countries,” *International Transactions in Operational Research*, vol. 26, pp. 1935–1955, 2017.

[17] D. Mailly, I. Abi-Zeid, and S. Pepin, “A Multi-Criteria Classification Approach for Identifying Favourable Climates for Tourism,” *Journal of Multi-Criteria Decision Analysis*, vol. 21, no. 1, pp. 65–75, 2014.

[18] A. S. Costa, K. Govindan, and J. R. Figueira, “Supplier classification in emerging economies using the ELECTRE TRI-nC method: A case study considering sustainability aspects,” *Journal of Cleaner Production*, vol. 201, pp. 925–947, 2018.

[19] F. S. Pinto, A. S. Costa, J. R. Figueira, and R. C. Marques, “The quality of service: An overall performance assessment for water utilities,” *Omega*, vol. 69, pp. 115–125, 2017.

[20] A. S. Costa, J. R. Figueira, and J. Borbinha, “A multiple criteria nominal classification method based on the concepts of similarity and dissimilarity,” *European Journal of Operational Research*, vol. 271, no. 1, pp. 193–209, 2018.

[21] R. Rodrigues, “Front-end Migration and User Experience Improvements in the DecSpace MCDA Framework,” *MSc Dissertation, Instituto Superior Técnico, Universidade de Lisboa*, 2019.

[22] J. D. A. Simões, G. F. Augusto, and C. Hernández-Quevedo, *Portugal: Health system review*, vol. 19, no. 2, 2017.

[23] Y. A. Ozcan, *Health Care Benchmarking and Performance Evaluation*, ser. International Series in Operations Research & Management Science. Springer, vol. 120, 2014.

[24] A. Donabedian, "Evaluating the quality of medical care," *The Milbank Quarterly*, vol. 83, no. 4, pp. 691–729, 2005.

[25] D. C. Ferreira and R. C. Marques, "Do quality and access to hospital services impact on their technical efficiency?" *Omega*, vol. 86, pp. 218–236, 2018.

[26] M. A. Pereira, "Preference information incorporation for decision-making in a DEA model using the Choquet integral," *MSc Dissertation, Instituto Superior Técnico, Universidade de Lisboa*, 2018.

[27] D. C. Ferreira, R. C. Marques, and A. M. Nunes, "Economies of scope in the health sector: The case of Portuguese hospitals," *European Journal of Operational Research*, vol. 266, no. 2, pp. 716–735, 2018.

[28] S. Corrente, S. Greco, and R. Słowiński, "Multiple Criteria Hierarchy Process for ELECTRE Tri methods," *European Journal of Operational Research*, vol. 252, no. 1, pp. 191–203, 2016.

[29] A. Rocha, "Quality assessment of the Portuguese public hospitals: A multicriteria approach," *MSc Dissertation, Instituto Superior Técnico, Universidade de Lisboa*, 2019.

[30] J. Figueira and B. Roy, "Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure," *European Journal of Operational Research*, vol. 139, no. 2, pp. 317–326, 2002.

[31] M. Augusto, J. Lisboa, M. Yasin, and J. Figueira, "Benchmarking in a multiple criteria performance context: An application and a conceptual framework," *European Journal of Operational Research*, vol. 184, pp. 244–254, 2008.

[32] J. R. Figueira, S. Greco, and B. Roy, "ELECTRE methods with interaction between criteria: An extension of the concordance index," *European Journal of Operational Research*, vol. 199, no. 2, pp. 478–495, 2009.

A

ELECTRE TRI-nC algorithm code

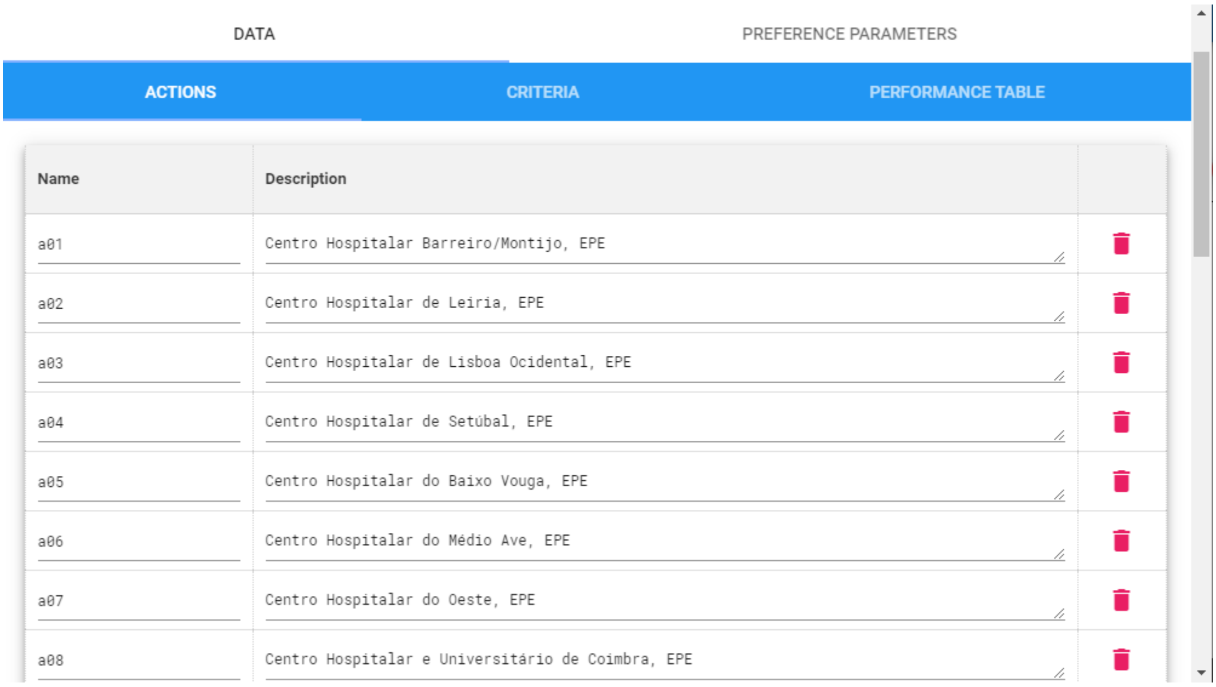
Listing A.1: DecSpace service code of ELECTRE TRI-nC method.

```
export default {
  execute(input) {
    var Parser = require('expr-eval').Parser;
    var parser = new Parser();

    // verify if input has content
    for(var prop in input) {
      if(input[prop].length==0) {
        var res = {};
        res.err = true;
        res.msg = prop + ' values missing';
        return res;
      }
    }
    // **************************************************************
    // ******************** variable declaration ********************

    // input data
    var actions = input.actions;
    var criteria = clone(input.criteria);
    var performanceTable = clone(input.performanceTable);
    var referenceActions = clone(input.referenceActions);
    var weights = clone(input.weights[0]);
    var thresholds = input.thresholds;
    var credibilityLevel = input.credibility;

    // initialize data
    var distanceValues = [];
    var categories = [];
    var assignedActions = [];

    // **************************************************************
    // ******************** main algorithm flow *********************

    // normalize the input weights
    normalizeWeights();

    // invert the values of criteria to minimize
    minimizeCriteria();

    // calculate the distance between actions and reference actions
    distanceReference();

    // calculate the best C(0), worst C(q+1) categories and the distance
    extentReference();

    // define the array of categories and eliminate the duplicates
    defineCategories();

    // assign the actions to the categories
    assignActions();

    // output
    return assignedActions;

    // **************************************************************
    // ******************** auxiliary functions *********************

    function clone(object) {
      return JSON.parse(JSON.stringify(object));
    }

    function normalizeWeights(){
      var sum = 0;
      for(var crit of criteria){
        weights[crit['Name']] = Number(weights[crit['Name']]);
        sum += weights[crit['Name']];
      }
      for(var crit of criteria){
        weights[crit['Name']] /= sum;
      }
    }

    function minimizeCriteria() {
      for(var crit of criteria) {
        if(crit['Direction'] == 'Minimization'){
          if(crit['Scale Type'] =='Cardinal'){
            var aux = crit['Min'];
            crit['Min'] = - crit['Max'];
            crit['Max'] = - aux;
          } else if(crit['Scale Type']=='Ordinal'){
            crit['Num Levels'] = - crit['Num Levels'];
          }
          for(var act of performanceTable)
            act[crit['Name']] = - act[crit['Name']];
            for(var ref of referenceActions)
            ref[crit['Name']] = - ref[crit['Name']];
        }
      }
    }

    function distanceReference() {
      for(var perf of performanceTable){
        var action = perf['Name'];
        for(var ref of referenceActions){
          var reference = ref['Name'];
          for(var crit of criteria){
            var criterion = crit['Name'];
            var distance = perf[criterion] - ref[criterion];
            distanceValues.push({'Action':action, 'Reference':reference,
            'Value':distance, 'Criteria':criterion});
          }
        }
      }
    }

    function extentReference(){
      for(var crit of criteria){
        var criterion = crit['Name'];
        if(crit['Scale Type']=='Cardinal'){
          var firstValue = crit['Min'];
          var lastValue = crit['Max'];
        } else if(crit['Scale Type']=='Ordinal' && crit['Direction'] == '
           Maximization'){
          var firstValue = 1;
          var lastValue = crit['Num Levels'];
          } else if(crit['Scale Type']=='Ordinal' && crit['Direction'] == '
             Minimization'){
            var firstValue = crit['Num Levels'];
            var lastValue = -1;
          }
        for(var performance of performanceTable){
          var action = performance['Name'];
          var firstDistance = performance[criterion] - firstValue;
          var lastDistance = performance[criterion] - lastValue;
          distanceValues.push({'Action':action, 'Reference':'first',
          'Value':firstDistance, 'Criteria':criterion});
          distanceValues.push({'Action':action, 'Reference':'last',
          'Value':lastDistance, 'Criteria':criterion});
        }
      }
    }

    function partialConcordance(crit,action,reference){
      var preference = Number(thresholds[0][crit]);
      var indifference = Number(thresholds[1][crit]);
      for(var aux of distanceValues){
        if(aux['Criteria'] == crit && aux['Action'] == action && aux['
           Reference'] == reference){
          var value = Number(aux['Value']);
        } else if(aux['Criteria'] == crit && aux['Action'] == reference &&
           aux['Reference'] == action){
          var value = - Number(aux['Value']);
        }
      }
        if(value >= - indifference){
          return 1;
        } else if(value >= - preference){
          return (preference + value) / (preference - indifference);
        } else return 0;
    }
```

```

    function concordanceIndex(action,reference){
      var x = 0;
      for(var crit of criteria){
        var weight = Number(weights[crit['Name']]);
        x += weight * partialConcordance(crit['Name'],action,reference);
      }
      return x;
    }

    function partialDiscordance(crit,action,reference){
      var preference = Number(thresholds[0][crit]);
      var veto = Number(thresholds[2][crit]);
      for(var aux of distanceValues){
        if(aux['Criteria'] == crit && aux['Action'] == action && aux['Reference'] == reference){
          var value = Number(aux['Value']);
        } else if(aux['Criteria'] == crit && aux['Action'] == reference && aux['Reference'] == action){
          var value = - Number(aux['Value']);
        }
      }
        if(value >= - preference || isNaN(veto) || veto == 0){
          return 0;
        } else if(value >= - veto){
          return (preference + value) / (preference - veto);
        } else return 1;
    }

    function credibilityIndex(action,reference){
      var prod = 1;
      var c = concordanceIndex(action,reference);
      for(var crit of criteria){
        var d = partialDiscordance(crit['Name'],action,reference);
        if(d > c){
          prod *= (1 - d) / (1 - c);
        }
      }
      return c * prod;
    }

    function categoricalIndex(x,y){
      var ans = 0;
      for(var ref of referenceActions){
        if(ref['Category'] == y){
          var credibility = credibilityIndex(x,ref['Name']);
          ans = Math.max(ans,credibility);
        } else if(ref['Category'] == x){
          var credibility = credibilityIndex(ref['Name'],y);
          ans = Math.max(ans,credibility);
        }
      }
      return ans;
    }

    function selectingFunction(action,category){
      return Math.min(categoricalIndex(action,category),categoricalIndex(category,action));
    }

    function defineCategories(){
      categories.push('first');
      for(var ref of referenceActions){
        categories.push(ref['Category']);
      }
      categories.push('last');
      categories = Array.from(new Set(categories));
    }

    function descendingRule(action){
      var aux = categories.length;
      for(var i = 2; i <= aux; i++){
        var cat = categories[aux - i];
        var catIndex = categoricalIndex(action,cat);
        if(catIndex >= credibilityLevel){
          if(cat == categories[aux - 2]){
            return cat;
          } else if(cat == categories[0]){
            return categories[1];
          } else if(Math.min(catIndex,categoricalIndex(cat,action)) >
          selectingFunction(action,categories[aux - i + 1])){
            return cat;
          } else return categories[aux - i + 1];
        }
      }
    }

    function ascendingRule(action){
      var aux = categories.length;
      for(var i = 1; i < aux; i++){
        var cat = categories[i];
        var catIndex = categoricalIndex(cat,action);
        if(catIndex >= credibilityLevel){
          if(cat == categories[1]){
            return cat;
          } else if(cat == categories[aux - 1]){
              return categories[aux - 2];
          } else if(Math.min(catIndex,categoricalIndex(action,cat)) >
          selectingFunction(action,categories[i - 1])){
            return cat;
          } else return categories[i - 1];
        }
      }
    }

    function assignActions(){
      for(var action of actions){
        var name = action['Name'];
        var option1 = descendingRule(name);
        var option2 = ascendingRule(name);
        if(categories.indexOf(option1) < categories.indexOf(option2)){
          assignedActions.push({'Action':name,'Minimum':option1,'Maximum':option2});
        } else assignedActions.push({'Action':name,'Minimum':option2,'
            Maximum':option1});
      }
    }

  }
}
```

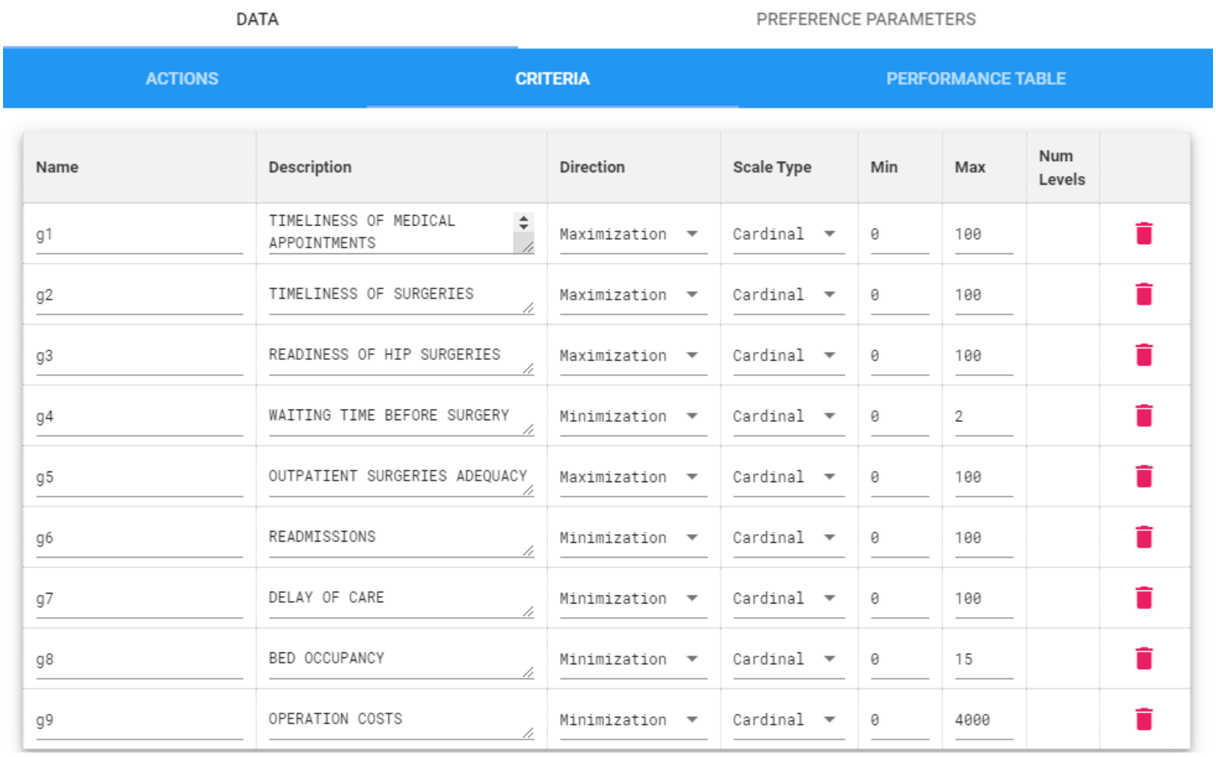
B

DECSPACE demonstration



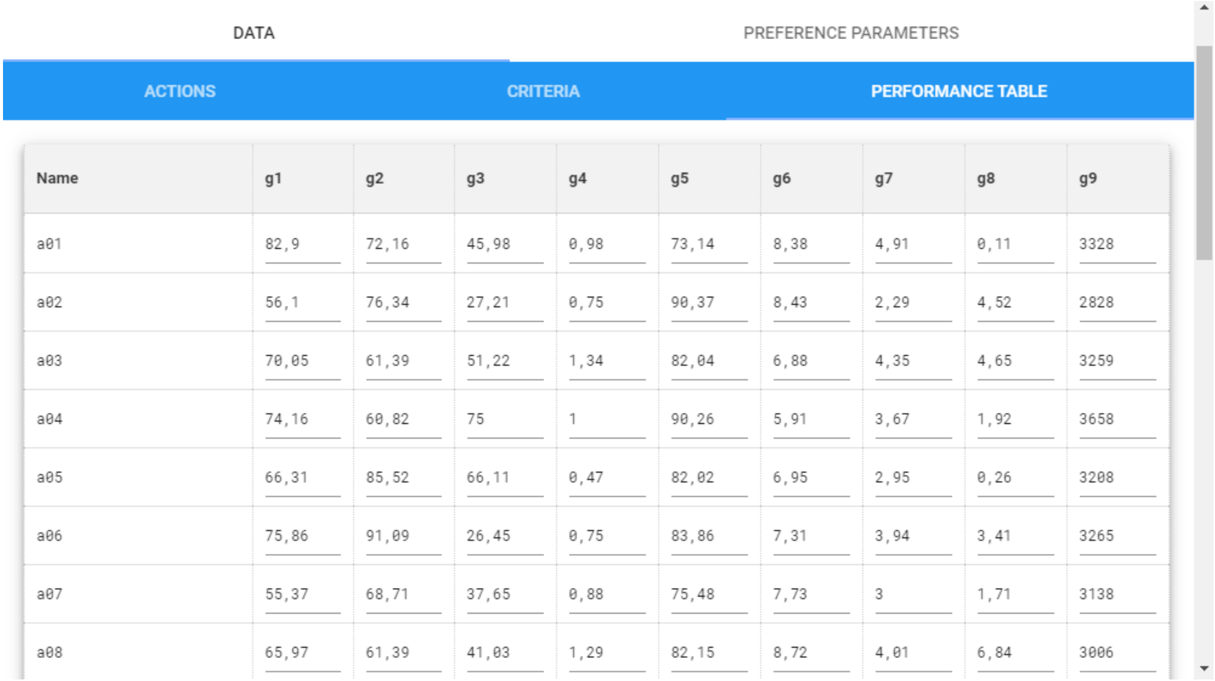
Name	Description	
a01	Centro Hospitalar Barreiro/Montijo, EPE	🗑
a02	Centro Hospitalar de Leiria, EPE	🗑
a03	Centro Hospitalar de Lisboa Ocidental, EPE	🗑
a04	Centro Hospitalar de Setúbal, EPE	🗑
a05	Centro Hospitalar do Baixo Vouga, EPE	🗑
a06	Centro Hospitalar do Médio Ave, EPE	🗑
a07	Centro Hospitalar do Oeste, EPE	🗑
a08	Centro Hospitalar e Universitário de Coimbra, EPE	🗑

Figure B.1: Actions in DecSpace.



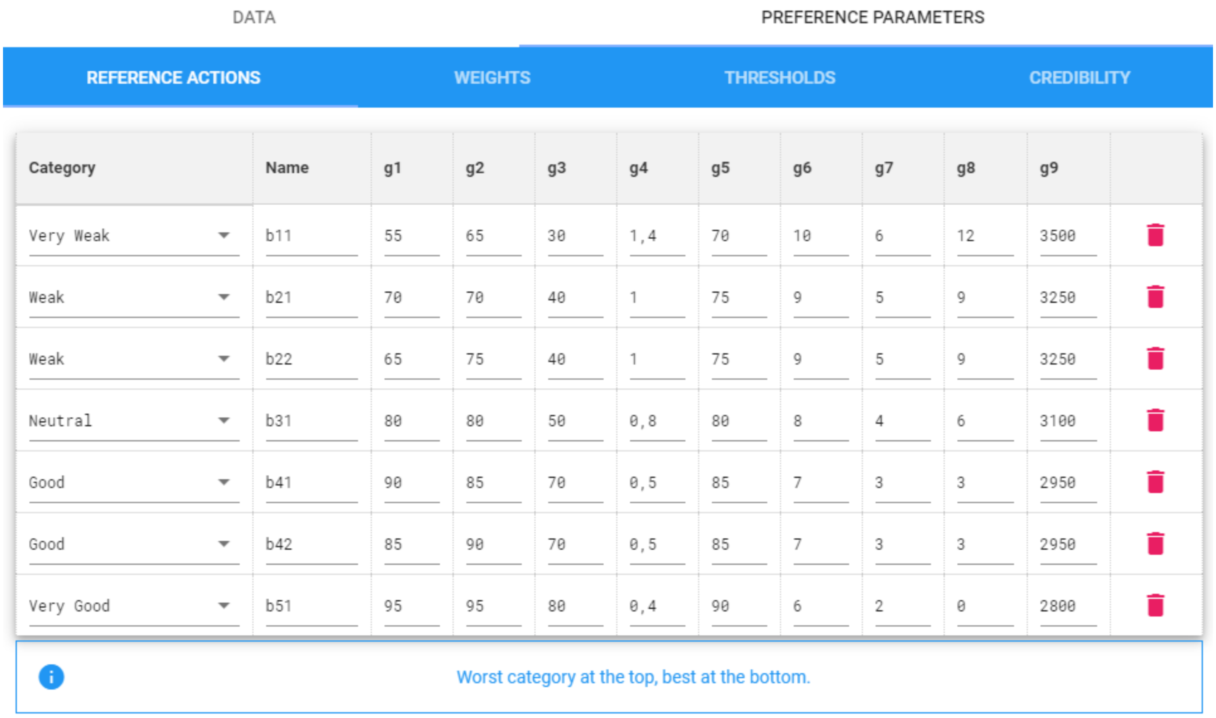
Name	Description	Direction	Scale Type	Min	Max	Num Levels	
g1	TIMELINESS OF MEDICAL APPOINTMENTS	Maximization	Cardinal	0	100		
g2	TIMELINESS OF SURGERIES	Maximization	Cardinal	0	100		
g3	READINESS OF HIP SURGERIES	Maximization	Cardinal	0	100		
g4	WAITING TIME BEFORE SURGERY	Minimization	Cardinal	0	2		
g5	OUTPATIENT SURGERIES ADEQUACY	Maximization	Cardinal	0	100		
g6	READMISSIONS	Minimization	Cardinal	0	100		
g7	DELAY OF CARE	Minimization	Cardinal	0	100		
g8	BED OCCUPANCY	Minimization	Cardinal	0	15		
g9	OPERATION COSTS	Minimization	Cardinal	0	4000		

Figure B.2: Criteria in DECSPACE.



Name	g1	g2	g3	g4	g5	g6	g7	g8	g9
a01	82,9	72,16	45,98	0,98	73,14	8,38	4,91	0,11	3328
a02	56,1	76,34	27,21	0,75	90,37	8,43	2,29	4,52	2828
a03	70,05	61,39	51,22	1,34	82,04	6,88	4,35	4,65	3259
a04	74,16	60,82	75	1	90,26	5,91	3,67	1,92	3658
a05	66,31	85,52	66,11	0,47	82,02	6,95	2,95	0,26	3208
a06	75,86	91,09	26,45	0,75	83,86	7,31	3,94	3,41	3265
a07	55,37	68,71	37,65	0,88	75,48	7,73	3	1,71	3138
a08	65,97	61,39	41,03	1,29	82,15	8,72	4,01	6,84	3006

Figure B.3: Performance table in DECSPACE.



Category	Name	g1	g2	g3	g4	g5	g6	g7	g8	g9	
Very Weak	b11	55	65	30	1,4	70	10	6	12	3500	🗑
Weak	b21	70	70	40	1	75	9	5	9	3250	🗑
Weak	b22	65	75	40	1	75	9	5	9	3250	🗑
Neutral	b31	80	80	50	0,8	80	8	4	6	3100	🗑
Good	b41	90	85	70	0,5	85	7	3	3	2950	🗑
Good	b42	85	90	70	0,5	85	7	3	3	2950	🗑
Very Good	b51	95	95	80	0,4	90	6	2	0	2800	🗑

ⓘ Worst category at the top, best at the bottom.

Figure B.4: Reference actions in DecSpace.

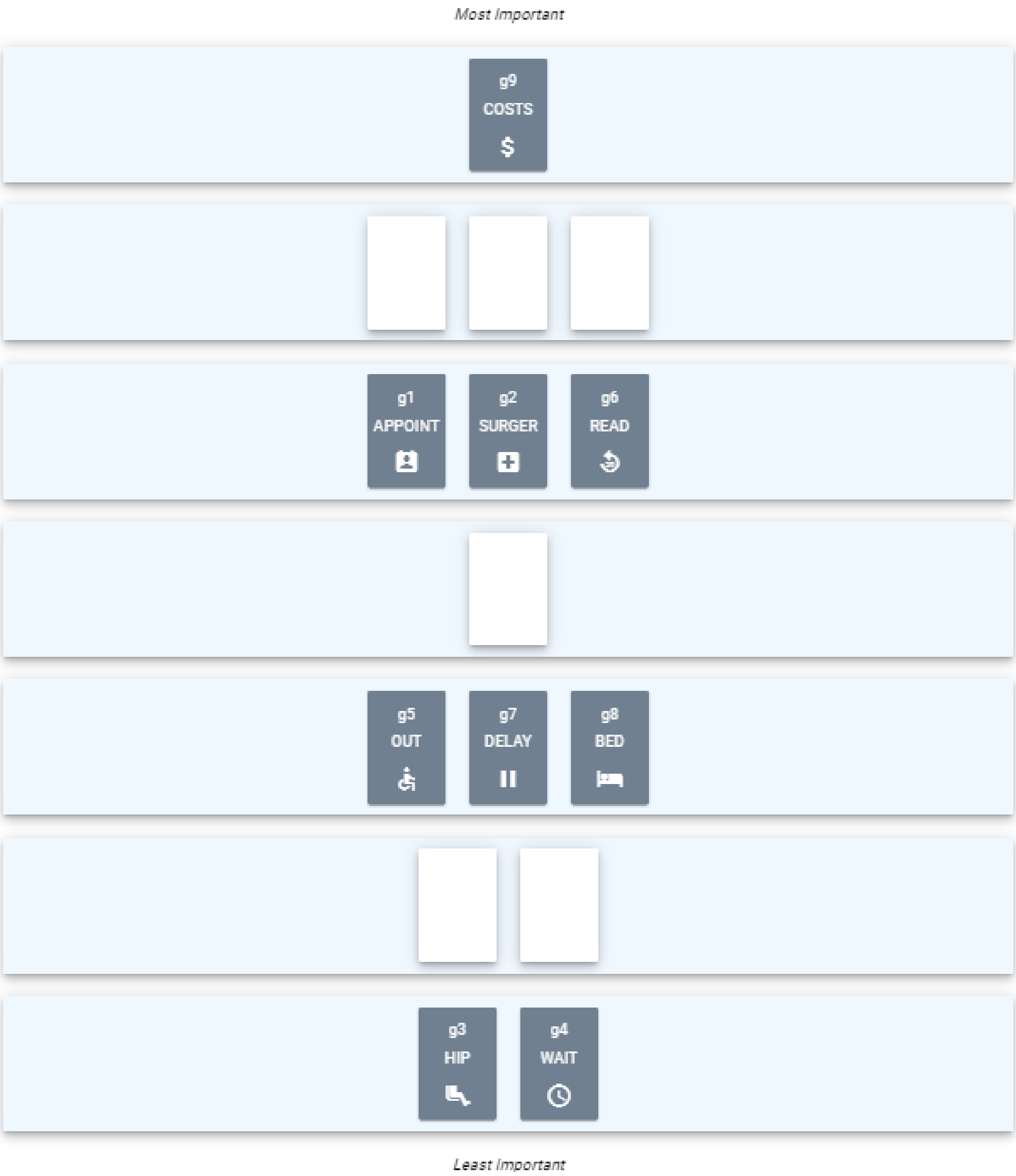


Figure B.5: DECK CARDS METHOD (SCENARIO 1) in DECSPACE.

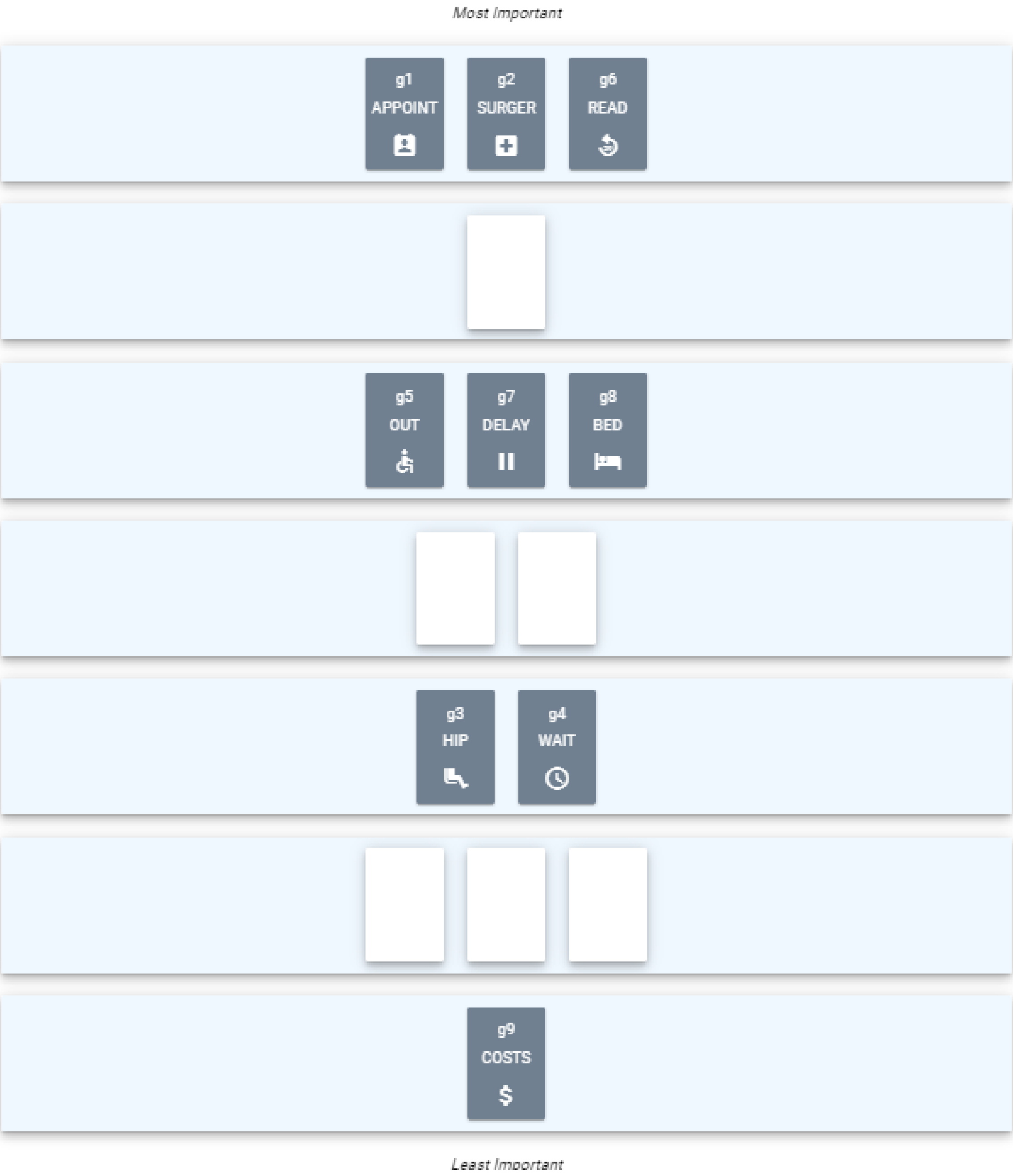
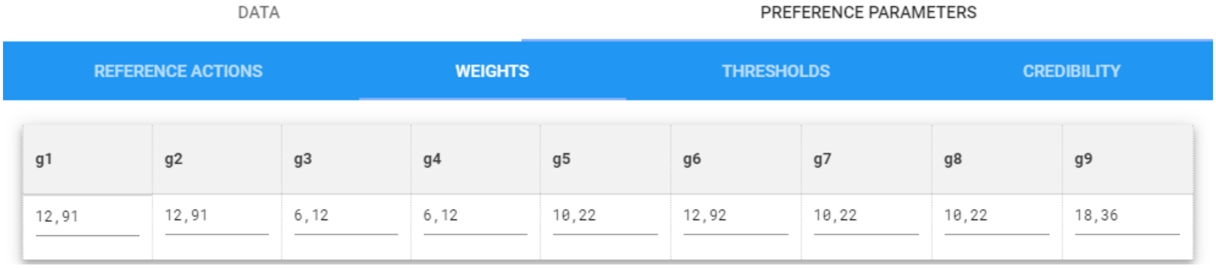
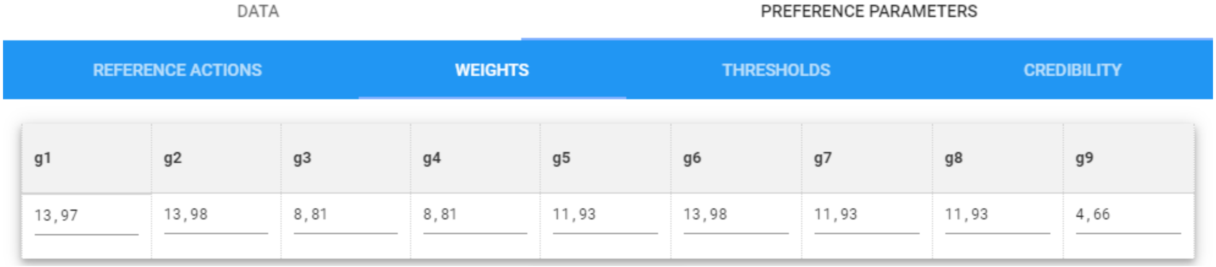


Figure B.6: DECK CARDS METHOD (SCENARIO 2) in DECSPACE.



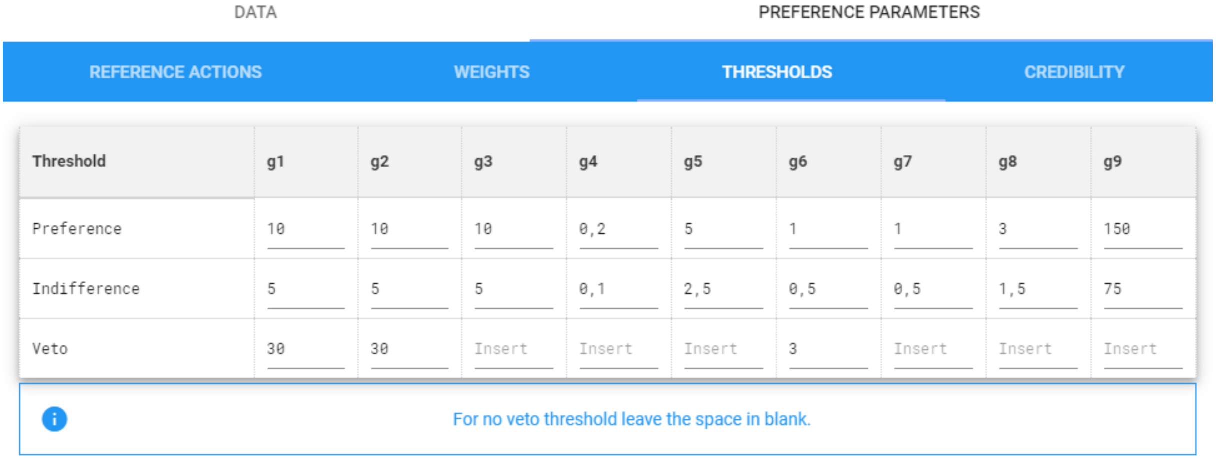
g1	g2	g3	g4	g5	g6	g7	g8	g9
12,91	12,91	6,12	6,12	10,22	12,92	10,22	10,22	18,36

Figure B.7: Criteria weights (Scenario 1) in DecSpace.



g1	g2	g3	g4	g5	g6	g7	g8	g9
13,97	13,98	8,81	8,81	11,93	13,98	11,93	11,93	4,66

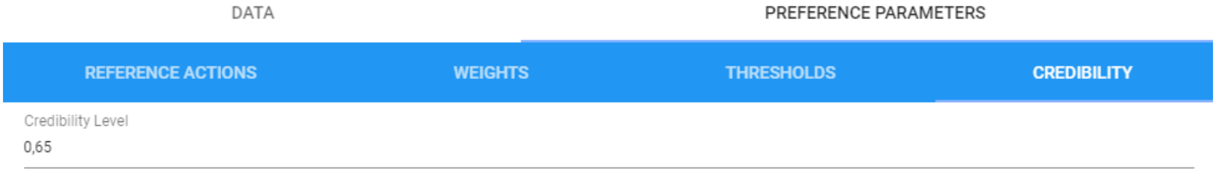
Figure B.8: Criteria weights (Scenario 2) in DecSpace.



Threshold	g1	g2	g3	g4	g5	g6	g7	g8	g9
Preference	10	10	10	0,2	5	1	1	3	150
Indifference	5	5	5	0,1	2,5	0,5	0,5	1,5	75
Veto	30	30	Insert	Insert	Insert	3	Insert	Insert	Insert

For no veto threshold leave the space in blank.

Figure B.9: Thresholds in DecSpace.



Credibility Level
0,65

Figure B.10: Credibility level in DecSpace.