Jason Papathanasiou & Nikolaos Ploskas

Multiple Criteria Decision Making: Methods, Examples and Python Implementations

May 8, 2017

Springer

To Dimitra, Alexander and Charicleia, always
For their love and patience during sailing in uncharted waters
– Jason Papathanasiou

To my family
– Nikolaos Ploskas

Foreword

I am particularly happy to write this foreword, for several reasons. First, I appreciate the work done by Jason Papathanasiou and Nikolaos Ploskas to propagate multicriteria decision aid in classes, to make students think about the consequences of their decisions, and to promote the use of multicriteria methods in actual decision problems. Inspired by the famous quote by the French writer and politician André Malraux, I am sure that the twenty-first century will be ethical or it will not be. Indeed, ethics is essential for the survival of our world. Decisions shape our future. Decisions are often difficult to make. Decisions can be good or bad. Decisions have often been poorly made, especially during the last decades. Today, we can observe the resulting crises all around the world. Many decisions are bad because they rely on a single, often economic (cost or profit) criterion. They can be made on a qualitative basis (experience, expertise, etc.) or using unicriterion optimization models and methods. Anyway, they usually fail because they are shortsighted and biased: they do not take into account all the stakeholders, all the objectives that are essential for our future. As a crucial example, achieving sustainable development is impossible with a unicriterion approach. It calls for a multicriteria approach encompassing economic as well as social and environmental objectives. Multicriteria decision aid can help us to achieve better, more sustainable decisions. This book is an important step toward a more widespread use of multicriteria models and methods.

A second point is that the authors do not focus on a single method but rather review different multicriteria methods. It is important to understand that there exists no ideal multicriteria decision making method. Instead many different methods have been proposed over the last fifty years. Each of them has advantages and limits. Each of them is making specific assumptions about the type of decision problem and about the preferences of the decision makers. Choosing the "right" method also depends on the problem at hand; the availability of data, the quality of data, the type and number of decision makers, the requirements of the different stakeholders, etc. All these parameters have an impact on the choice of an appropriate method. With this in mind, the authors present the principles and the characteristics of six families of methods. This set of methods is not exhaustive; other interesting methods exist as

well. However, the choice they have made reflects as well as possible the different types of methods currently available.

TOPSIS and VIKOR are two compensatory aggregation methods that have been widely used during the recent years. They are simple to use and rather straightforward to understand. PROMETHEE is a well-known outranking method. It allows for partial compensation between criteria and has many extensions including among others advanced sensitivity analysis tools and the GAIA visual descriptive model. It thus provides decision makers with a much richer information at the expense of a more complex preference modeling. The SIR method is another interesting extension of PROMETHEE. On the other hand, AHP is quite different as it is designed to work with qualitative input data and it is based on a very specific pairwise comparison principle. Many people have been critical about the theoretical basis of AHP. Yet, it is one the most widely-used multicriteria decision aid method. Finally, goal programming methods are based on the mathematical programming model. They are an appropriate choice when decisions are related to decision variables whose best values should be determined under a set of constraints.

The lecture of this book will provide the reader with a quite good overview of existing multicriteria decision making methods and with guidelines for selecting an appropriate method for each decision problem. It is helpful for the practitioner as well as for the researcher wishing to compare different approaches and methods.

Finally, the authors provide reusable Python computer code for each of the methods presented in the book. The availability of computer solutions is essential for the use of multicriteria methods. When I started working on multicriteria methods back in the 1980's, people had to rely on programming mainframe computers to implement methods. It was difficult, time consuming, and only possible for people with confirmed computer skills. Interaction was very limited and computer graphics were nearly non-existent. The advent of personal computers started a new era. Around 1990, interactive computer software, such as the PROMCALC implementation of PROMETHEE, appeared. They made it possible for individual users without programming skills to apply multicriteria decision aid methods in actual decision processes. Progressively, more applications appeared and methods evolved based on the feedback from actual applications. At the turn of the century, graphical interfaces became widely available and more advanced, and user-friendly software appeared, boosting even more the use of multicriteria methods. Today, most multicriteria methods are supported by specific software such as Visual PROMETHEE or Expert Choice (for AHP). Practitioners have powerful tools available to perform comprehensive and detailed analysis. However, these specialized pieces of software do not make it easy to perform comparisons between different methods, to adapt methods to special decision problems, or to test new developments or extensions of the methods. The Python code provided by the authors is an elegant and original solution to these shortcomings. It is open-source, can be modified easily, and relies on a modern and easy-to-learn language available on multiple platforms. Only moderate computer skills are necessary to take advantage of it. Students will be able to use it to better understand the inner works of the methods and to develop more advanced projects. Practitioners will have a way to adapt the methods to their specific needs.

Researchers will have the basis for analyzing the characteristics of the methods, for validating theoretical experiences, or for developing new modules or methods.

In the compact format of this book, the authors manage to transmit essential information needed to learn, understand, and apply multicriteria decision aid. They also provide interested readers with the means to practice, to test, and to validate their acquired knowledge.

Πολλές ευχαριστίες στους Ιάσωνα και Νικόλαο για αυτό το υπέροχο βιβλίο!
(Many thanks to Jason and Nikolaos for this wonderful book!)

Brussels, May 2017 *Bertrand Mareschal*

Preface

The multiple criteria decision aid (MCDA) discipline is growing rapidly as the number of publications in the literature soared during the recent years. It is successfully applied in all kinds of scientific domains and numerous MCDA methodologies exist with an even larger number of variations available. The rationale behind this book is something that the authors were able to identify at a quite early stage on their engagement in the field of MCDA. And that is that, as already said, there is a lot of research available on the literature, however much of it is difficult to interpret or be exploited by the non-expert researcher, practitioner, or academic. This is an applied field of research and people need to be able to fully understand each step of the methodologies and implement them easily, coherently, and more importantly correctly.

The main feature of this book is the presentation of a variety of MCDA methods. This book includes the thorough theoretical and computational presentation of six MCDA methods:

- TOPSIS
- VIKOR
- PROMETHEE
- SIR
- AHP
- Goal Programming

The novelty of this book is that the presentation of each method is focused on three aspects:

- Initially, the theoretical background is presented for each method including variants that have been proposed in the literature.
- Secondly, a thorough numerical example is presented for each method.
- Finally, a Python code is presented to fully cover the presentation of each method. The Python implementations that are presented in this book are as simple as possible.

This book is addressed to students, scientists, and practitioners. Students will learn various MCDA methods through illustrative examples, while they will be able to solve the examples using the Python codes given in this book. This book covers thoroughly a course on Multicriteria Decision Analysis whether Python is used or not. Scientists and practitioners will have a book in their library that presents many different MCDA methods and their variants.

The book is organized in six chapters plus an appendix. A simple, yet important method in the MCDA pantheon is presented in the first chapter, namely TOPSIS. The following chapter is about a very similar method called VIKOR; a method that recently has gained much popularity. The algorithms in both cases have a number of distinct steps that offer a great opportunity to modify the classical versions of the methods, which are also naturally presented. The fuzzy variations are included too, as well as the group decision making extensions. The third chapter focuses on PROMETHEE, a well known member of the outranking family of methods. PROMETHEE I and II are fully analyzed and then the text proceeds with PROMETHEE V as well as short references to the GAIA method and the group decision making PROMETHEE procedure. SIR is next, a method that integrates both TOPSIS and PROMETHEE ideas and could be actually considered as a generalization of PROMETHEE. This is a relative new member of the MCDA portfolio, but the authors feel that SIR could in time make a difference. A book about MCDA cannot be complete if it does not include the AHP method. Despite its somewhat controversial status within the broader MCDA community, AHP has proven to be a very popular method, perhaps the most popular. This is a reason enough for the authors to include a chapter on this volume including a discussion on the variations and some shortcomings of the method. The final chapter is about goal programming, which is based actually on the linear programming principles; the classical approach is complemented on this book by the weighted, lexicographic, and Chebyshev versions. The appendix includes the revised Simos method for the calculation of the weights associated with the criteria in an MCDA model; a sometimes rather difficult step during the model formulation. Important methodologies (like ELECTRE, MAUT, ANP, and multiple objective programming) are not included and the authors fully acknowledge this shortcoming; if all goes well, this is expected to be rectified in a future, second edition of the present volume.

Various software packages exist for many MCDA methods, both proprietary and free. In all the chapters of this book, code is provided for each and every step of the procedures, and this is also readily available on the book website. The programming language used is Python; a modern language with a numerous and robust user base. The authors assume a basic knowledge of programming principles and especially of the Python language by the reader; however, they have tried to keep the code as simple as possible. Early on, they decided that important capabilities of the language like the object oriented programming features, decorators or iterators, are not actually important for the purpose of this book. This is a book not only intended for computer science specialists but for researchers from every field of science that need tools in order to model first and then solve a particular problem that they need to tackle using an MCDA method. Python is very well-suited for this task; the au-

thors pondered at the beginning of this endeavor using another fully object oriented programming language like Java, but this would needlessly complicate the code. The code is kept as simple as possible but is not simplistic; it is divided in modules and the reader is indeed required to have some knowledge of Python and its peculiarities. Comments are included in the code, as well as commentary on the main text and this combined with the ability of Python to produce very clearly defined and readable code should result on easily read, understood, and applied code listings. Moreover, the readers are very welcome and actually are encouraged to use, modify, and extend the code to better suit their particular demands.

All of the Python packages included in the book are free and open source as is the language itself; this was another prerequisite firmly set by the authors early on. There is no proprietary code anywhere; the idea is that everyone interested should be able to use the code in this book without restrictions. The Python packages included and necessary for some of the code listings in this book are:

- *graphviz*: a library for creating and rendering graph drawings
- *Matplotlib*: a library used for plotting the results of the MCDA methods
- *PuLP*: an open source Python-based linear programming solver that allows the user to optimize mathematical programming models
- *Pyomo*: an open-source collection of Python software packages for formulating optimization models

The authors would like to thank the publisher's team for their help and patience during this endeavor and Professor Bertrand Mareschal for agreeing to write the foreword to this book. They also hope that the international research community will find the book interesting and of high standards. For any mistakes in the text the authors, of course, acknowledge exclusive and full responsibility.

Thessaloniki,
April 2017

Jason Papathanasiou
Nikolaos Ploskas

Contents