Noname manuscript No.
(will be inserted by the editor)

Supporting the Multi-Criteria Decision Aiding process: R and the MCDA package

Sébastien Bigaret · Richard E. Hodgett · Patrick Meyer · Tatiana Mironova · Alexandru-Liviu Olteanu

the date of receipt and acceptance should be inserted later

Abstract Reaching a decision when multiple, possibly conflicting, criteria are taken into account is often a difficult task. This normally requires the intervention of an analyst to aid the decision maker in following a clear methodology with respect to the steps that need to be taken, as well as the use of different algorithms and software tools. Most of these tools focus on one or a small number of algorithms, some are difficult to adapt and interface with other tools, while only a few belong to dynamic communities of contributors allowing them to expand in use and functionality. In this paper, we address these issues by proposing to use the R statistical environment and the MCDA package of decision aiding algorithms and tools. This package is meant to provide a wide range of MCDA algorithms that may be used by an analyst to tailor a decision aiding process to their needs, while the choice of R takes advantage of the yet poorly explored opportunity to interface data analysis and decision aiding. We additionally demonstrate the use of this tool on a practical application following a well-defined decision aiding process.

Keywords R · MCDA · decision aiding process

1 Introduction

Over the past 50 years, many articles and books have covered the topic of Multi-Criteria Decision Aiding (MCDA) with many different methods and algorithms being proposed. The interested reader can for example refer to Roy (1991); Bouyssou et al (2006); Belton and Stewart (2002). Within the MCDA framework

Sébastien Bigaret · Patrick Meyer · Tatiana Mironova · Alexandru-Liviu Olteanu
IMT Atlantique, Lab-STICC, Univ. Bretagne Loire, F-29238 Brest, France

Richard E. Hodgett
Leeds University Business School, Maurice Keyworth Building, The University of Leeds, Leeds, LS2 9JT, United Kingdom

Patrick Meyer
Corresponding author
Tel.: +33 (0)2 29 00 14 63, Fax.: +33 (0)2 29 00 10 30
E-mail: patrick.meyer@imt-atlantique.fr

we generally identify at least one decision-maker (DM), who is in charge of and responsible for the decision to be made. He is confronted with several decision alternatives which are evaluated on a set of criteria or points of view, which typically are conflicting. The DM usually expresses some preferences related to these alternatives and criteria, which are usually used as parameters by MCDA algorithms in order to provide a solution to the decision problem. The decision problem may also fall into different categories, as e.g., choice (determine the "best" alternative), ranking (order the alternatives from the "best" to the "worst" one) and sorting (assign the alternatives to predefined and ordered classes). In order to illustrate these concepts, let us present a short example. A school committee is tasked with allocating a fixed number of scholarships to students based on their performances on the subjects they are being taught (e.g. mathematics, computer science, biology, etc.). In this case, the school committee is the DM, the students are the decision alternatives, while the subjects represent the criteria. The decision problem, in this case, is to rank all students from best to worst (ranking problem) and to select the top students as recipients of a scholarship. This ranking has to be done according to the preferences of the school committee.

MCDA has been applied to many different fields, such as health (Wahlster et al, 2015), finance and banking (Figueira et al, 2005, p. 799), environmental management (Lahdelma et al, 2014), urban planning using geographical information systems (Coutinho-Rodrigues et al, 2011), robotics (Taillandier and Stinckwich, 2011), energy planning (Figueira et al, 2005, p. 859), nuclear emergency management (Papamichail and French, 2013), equipment selection (Hodgett, 2016) etc. The process of decision aiding is often complex, depending on the specific field of application and the preferences of the DM. As a result, many MCDA algorithms have been developed over the years (see for example (Figueira et al, 2005; Keeney and Raiffa, 1976)). In order to overcome the difficulties linked to the decision problem, an analyst may be included in the decision aiding process. (S)he is an expert of MCDA, whose purpose is to guide the DM by choosing the correct formalization of the problem, the appropriate methods and algorithmic approaches, in order to support him/her in reaching a decision recommendation. In order to simplify and streamline the decision aiding process, several studies have already dealt with the topic of selecting the best suited algorithm for a decision problem (Guitouni et al, 1998; Ishizaka and Nemery, 2013), while others, as, e.g., Tsoukias (2007), have divided this process into multiple steps. Many software solutions have been proposed to help the analyst in the decision aiding process, however, in most cases they hold several limitations. Plenty of them focus on only a small number of algorithms, raising the need to use multiple software tools throughout the decision aiding process and the potential difficulties linked to their coupling. Other tools limit the capacity of the user to adapt their algorithms to their needs, while only a few belong to dynamic communities of contributors allowing them to grow in use and functionality.

These remarks provide the key motivation for this contribution. The MCDA package (Meyer et al, 2017) for the R statistical software (R Development Core Team, 2008) that we propose is meant to provide a wide range of algorithms that may be used by an analyst across an entire decision aiding process. The choice of R is also motivated by the ease in adapting the different functions to one's needs, the large community of contributors that may aid in extending the MCDA package, as well as the as of yet poorly explored opportunity of interfacing data analysis and decision aiding. Both the data analysis community may benefit from the possibility of applying decision aiding algorithms after the data analysis stage, as well as the decision aiding community from the possibility of applying data analysis during the decision aiding process.

The remainder of this article is organized in the following way. In Section 2 we provide a state of the art, starting with the MCDA process, the different algorithms that have been proposed and finishing with an overview of the most notable supporting software tools. In Section 3 we present and discuss our proposal to use the R statistical environment combined with our contribution, the MCDA package for R. In Section 4 we provide a very detailed illustrative example showcasing how R and the package may be used in practice. Finally, in Section 5 we finish with several conclusions and perspectives for future work.

2 State of the art

We start by providing a state of the art on the existing MCDA literature, covering the decision aiding process, the most commonly used algorithmic approaches, as well as some of the existing supporting software tools.

2.1 The multi-criteria decision aiding process

As mentioned in the introduction, decisions and the objects they are concerned with may be very diverse. In fact, each of us are faced with a multitude of decisions every day, from which route to take in order to get to work in the morning, to selecting what to have for lunch. There are numerous factors which influence these decisions, such as our preferences, our prior experiences, different constraints, etc. In certain cases, balancing these factors can be difficult. MCDA serves as an interface between DMs and analysts, guiding them in reaching a decision when multiple and often conflicting criteria are involved. The process generally starts with the analyst and DMs focusing on defining the problem, their goals and how the final decision should be reached (Bouyssou et al, 2006). One key aspect of MCDA is that the final decision may not need to be the best possible one, but one that is acceptable by all the stakeholders. Hence, when multiple DMs are involved, conflicts need to be handled in order to reach a consensus on the final decision. The term “decision frame”, used by Tversky and Kahneman (1981), supports the fact that DMs often base their decision on subjective judgments. Furthermore, nowadays real-world decision problems have become increasingly complex.

The following steps have been identified to structure a MCDA process: identify the problem, formulate the problem, construct the evaluation model and then reach a final recommendation (Bouyssou et al, 2006; Bisdorff et al, 2015; Figueira et al, 2005). Each of these steps contains additional sub-steps, which depend on the many factors that define a decision problem. Figure 1 illustrates an example decision aiding process and the complexity of the various steps involved. More precisely, the step of structuring the problem includes sub-steps such as identifying the stakeholders (or actors), identifying the context of the problem, the objectives of the decision and its respective properties. The second step of formulating the problem, involves identifying the decision alternatives and their criteria, the type of decision problem, as well as managing multiple DMs and their different perspectives. The third step involves the choice of a mathematical model and its tuning so that it reflects the perspective of the DM. Furthermore, a resolution method also needs to be selected in order to provide a recommendation to the decision problem. Finally, in the last step, this recommendation is presented to the DM, who then either validates the recommendation, asks for additional supporting analyses or revisits previous steps in order to refine the solution. We would like to highlight that the structure of the process is nonlinear, complex and iterative. This means that any point in the process we may choose different paths to follow, in some cases going back to previous ones. We highlight this, and the fact that the decision aiding process is decomposed into multiple sub-steps, as shown in Figure 1. At each of these steps we may have an interaction between the DM and the analyst, the extraction of an important piece of information, the use of an algorithm, or a visual representation of alternative, etc.

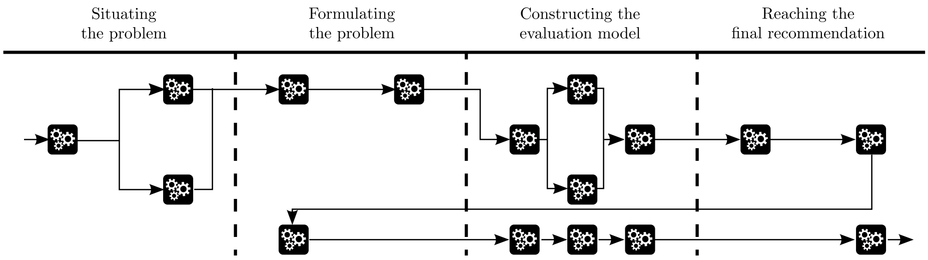

Fig. 1 The decision aiding process - example of one possible instance.

2.2 MC aggregation algorithms

The resolution step of the previously described MCDA process involves the use of an MC aggregation algorithm. Various such algorithmic approaches have been proposed in the literature (Bouyssou et al, 2006; Keeney and Raiffa, 1976; Roy, 1996). Roughly speaking, two main methodological schools can be identified, namely the outranking school of thought and the value-based theories.

The main idea behind *outranking methods* is to compare any two alternatives pair-wise on basis of their evaluations on the set of criteria, according to a majority rule. For two alternatives x and y of A, if for the DM there are enough arguments in favor of the statement "x is at least as good as y", then x outranks y (xSy) (Roy, 1996). These arguments are based on differences of evaluations on the various criteria which are compared to discrimination thresholds determined in accordance with the DM's preferences. Furthermore, a weight is associated with each criterion, which allows giving these local arguments more or less importance in the majority rule. A concordance index then aggregates these partial arguments via a weighted sum to obtain a credibility degree of the outranking. Three preference situations can be derived from this outranking relation. x and y are considered as indifferent if simultaneously xSy and ySx, they are considered as incomparable with respect to the available information if no outranking can be confirmed between them (neither

xSy nor ySx), and x (resp. y) is strictly preferred to y (resp. x) if xSy and not ySx (resp. ySx and not xSy). As this outranking relation is not necessarily complete or transitive, its exploitation in view of building a decision recommendation is in general quite difficult. Many exploitation procedures have been proposed in the literature to solve the three main types of multi-criteria decision problems mentioned in Section 2.1.

Methods based on *multiattribute value theory* aim to construct a numerical representation of the DM's preferences on the set of alternatives A. More formally, those techniques seek at modeling the preferences of the DM, supposed to be a weak order represented by the binary relation $\succsim$ on A, by means of an overall value function $U : A \to \mathbb{R}$ such that $x \succsim y \iff U(x) \geq U(y), \forall x, y \in A$. The overall value function U can be determined via many different methods, presented for example in von Winterfeldt and Edwards (1986, Chapter 8) in the context of an additive value function model. Ideally, such methods should consist of a discussion with the DM in the language of his/her expertise, and avoid technical questions linked to the model which is used. Note that the preference relation induced by such an overall value function is necessarily a complete weak order, which means that only two preference situations can occur : either x and y are considered indifferent (if $U(x) = U(y)$) or x (resp. y) is strictly preferred to y (resp. x) if $U(x) > U(y)$ (resp. $U(y) > U(x)$). Once the overall value function has been properly determined, its exploitation for the decision recommendation is usually straightforward, as all the alternatives have become comparable.

The main differences between these two methodological schools lie in the way the alternatives are compared and in the type of information which is required from the decision maker. Furthermore, outranking methods might be preferable if the evaluations of the alternatives on the criteria are mainly qualitative and if the DM would like to include some impreciseness about his/her preferences in the model, whereas value-based methods can be favored if a compensatory behavior of the DM should be modeled.

2.3 MCDA software

As we have previously discussed, many MCDA methods have been proposed in order to solve different types of decision problems. In order to help applying these methods to real decision problems, a wide range of software have been developed. Some of these software packages are either free (as in beer or as in speech) or only commercially available, while some of them are either stand-alone or web-based. Some software allow to be extended and therefore also gather a community of developers around them. Next to that, many of these software tools provide only a limited number of algorithms, in some cases only single methods (e.g. IRIS by Dias and Mousseau (2003), ELECTRE TRI by Mousseau et al (1999, 2000), MakeItRational by Make It Rational (2016), TransparentChoice by TransparentChoice Ltd. (2016), TOPSIS by Statistical Design Institute (2016), UTA Plus by Kostkowski and Slowinski (1996), JSMAA by Tervonen (2012)), while in other cases multiple methods (e.g. the diviz ecosystem with the XMCDA web-services (Meyer and Bigaret, 2012a) developed under the impetus of the Decision Deck Consortium, or Decernes MCDA by Yatsalo et al (2015)). For a broader review of the existing software tools, the reader may refer to Mustajoki and Marttunen (2013); Ishizaka and Nemery (2013); Weistroffer et al (2005); Mayag et al (2011); Baizyldayeva et al (2013) and International Society on Multiple Criteria Decision Making (2014).

We notice from these reviews of MCDA software, that no software tool is currently able to support the entire complex decision aiding process from start to finish. Additionally, according to Clemen and Reilly (2001), decomposition plays a crucial role in the decision process, while multi-method platforms appear to be more useful due to the possibility of choosing among different algorithms for solving the same problem. There are, nevertheless tools that allow for a wide degree of tuning of the methods they implement, such as for example diviz (Meyer and Bigaret, 2012a). The diviz workbench provides an interface for constructing complex MCDA algorithms from smaller components (available as the XMCDA web-services of the Decision Deck Consortium), which can be interconnected in the form of work-flows. In line with this notion of being able to tailor different methods and tools to one's needs are the R statistical environment (R Development Core Team, 2008) and the Python programming language. In both cases, we find some of the fastest growing communities of contributors and the ability to easily interconnect their contributions in order to solve different problems (Piatetsky, 2016). R, in particular, is widely used in the data science discipline, where we find a significant opportunity of adding MCDA approaches to be used after the data analysis stages. Similarly, integrating MCDA and data analysis can reveal new challenges for the MCDA community.

It should be nevertheless noted that R packages linked to MCDA methods or that may be used in conjunction with them in the decision aiding process also exist: Kappalab by Grabisch et al (2006, 2015), RXMCDA by Meyer and Bigaret (2012b), UTAR by Leistedt (2011), Benchmarking by Bogetoft and Otto (2015) or Rgraphiz by (Gentry et al, 2009, 2016).

All of these remarks serve as motivation for our proposal of the MCDA R package. Our aim is to provide as many different MCDA methods and tools as possible and to make them available to the R and the data analysis communities. In line with the philosophy of R, the package will allow the analyst to construct their own decision aiding process from start to finish, by applying the methods provided by the package, adapting them to their needs as well as making use of other methods and packages linked to data analysis. The functions of the MCDA R package are also easily integrable in the XMCDA web-services proposal of Decision Deck, and consequently in the diviz workbench. Last but not least, we hope that by proposing a library of MCDA functions in an environment like R will create a community of contributors which will participate in its dissemination and the general development effort.

3 R and the MCDA package

We present, in this section, our contribution, by first giving a brief presentation of the philosophy behind R and the package of MCDA functions that we propose, followed by a slightly more in-depth description of the contents of the package, namely the currently implemented functions.

3.1 Philosophy

R is an open-source functional programming language and environment mainly centered around data analysis (Venables et al, 1998; Ihaka and Gentleman, 1996). In recent years it has grown in popularity with the IEEE identifying it as the 9th most popular programming language in 2014, the 6th most popular in 2015 and the 5th most popular in 2016 (IEEE Spectrum, 2016). Due to the large community of R users, many tools in the form of functions within packages have been proposed, many dealing with handling different data formats, data pre-processing, data filtering and interactive visualizations. Although users need to have some basic programming experience they also first need to familiarize themselves with the R programming language. Once this is done, however, users can easily combine functions from different packages in order to solve their problem. Nevertheless, the majority of functions and packages are aimed at data analysis, and while there are a few packages linked to MCDA, there is plenty still to be done in this regard.

The MCDA package that we propose follows the philosophy of R, by encompassing a growing array of MCDA algorithms that may be used to decompose the decision aiding process into sub-steps. The package mainly targets MCDA practitioners that are familiar with the decision aiding process, giving them the possibility to construct this process as they see fit. As very often during a decision aiding process the DM does not have a clear picture of his/her problem (Simon, 1976), being able to quickly adapt the process as new information is made available is of great importance. Finally, the MCDA package may benefit both MCDA practitioners and data analysts, as MCDA practitioners could further apply methods linked to data analysis throughout the decision aiding process, while data analysts could use their data for reaching an objective in addition to analyzing it.

3.2 Currently implemented functions

At the time of writing, the package is very young and consequently is far from covering all of the algorithms from the classical MCDA literature. However, functions supporting various steps of the MCDA process have been implemented in the MCDA R package. They can be categorized as follows :

- state of the art aggregation algorithms;
- state of the art preference elicitation algorithms;
- tool and data manipulation functions;
- plot functions.

The implemented algorithms originate from the two main methodological schools presented in Section 2.2.

With respect to the *aggregation algorithms*, in the outranking paradigm, the currently implemented functions focus on a majority-rule sorting technique commonly called MR-Sort (Leroy et al, 2011; Sobrie et al, 2013), which is a simplified version of the classical Electre TRI method. The `MRSort` function allows to assign alternatives to a set of predefined categories according to a DM's preferences. This method has recently been extended to take into account so-called large performance differences by Meyer and Olteanu (2017). This extension is implemented in the `LPDMRSort` function. Concerning multiattribute value theory, the aggregation can be done with a weighted sum through the `weightedSum` function, which calculates the weighted sum of the evaluations of alternatives on criteria with respect to some criteria weights. To apply piece-wise linear value functions on a performance table, the `applyPiecewiseLinearValueFunctions-OnPerformanceTable` can be used. The package also proposes to use the `AHP` function, which implements the Analytic Hierarchy Process proposed by Saaty (1980), as well as the `pairwiseConsistencyMeasures` function which calculates four pairwise consistency checks for AHP (Siraj et al, 2015). Furthermore, the package proposes an implementation of the TOPSIS method originally proposed by Hwang and Yoon (1981) (`TOPSIS` function) and the MARE method by Hodgett et al (2014) (`MARE` function)

In terms of *preference elicitation algorithms*, in the outranking school, the parameters for the MR-Sort technique can be learned from assignment examples provided by the DM, either via the `MRSortInferenceExact` function (exact elicitation via linear programming), or the `MRSortInferenceApprox` function (approximate elicitation, adapted for large sets of assignment examples). The `MRSort-IdentifyIncompatibleAssignments` function can be used to identify assignment examples which are not compatible with an MR-Sort model. In a context of large performance differences, the `LPDMRSortInferenceExact` function allows to learn the preferential parameters from assignment examples. In case some assignments are incompatible with the large performance differences sorting model, they can be found via the `LPDMRSortIdentifyIncompatibleAssignments` function. Concerning multiattribute value theory, the package currently proposes preference elicitation methods related to the UTA technique originally proposed by Jacquet-Lagrèze and Siskos (1982). The `UTA` and `UTASTAR` functions allow to learn piece-wise linear value functions from rankings of alternatives, whereas the `UTADIS` function identifies such value functions together with category limits from assignment examples. The `additiveValueFunctionElicitation` function elicits a general additive value function from a ranking of alternatives.

Next to these algorithms which represent the heart of the MCDA process, the package provides some *tool and data manipulation* functions. Evaluations in a performance table can be normalized according to various normalization schemes in function `normalizePerformanceTable`. Alternatives can be assigned to categories with respect to some separation thresholds via the `assignAlternatives-ToCategoriesByThresholds` function.

Finally, to show the DM results or intermediary elements of the decision aiding process, a certain number of *plot* functions have been implemented. `plotRadar-PerformanceTable` allows to represent the alternatives very synthetically as radar plots. In the outranking context, `plotMRSortSortingProblem` plots the profiles of the alternatives and the categories for a sorting problem. In multiattribute value theory, `plotPiecewiseLinearValueFunctions` can be used to plot the piece-wise linear value functions (learned for example from a UTA-like method), whereas `plotAlternativesValuesPreorder` shows the pre-order of the alternatives obtained from their overall scores. Finally, the `plotMare` function presents a synthetic vision of the output of the Mare method.

The work on the package is ongoing, and we encourage the interested reader to contribute to this collective effort.

4 Illustrative example

In this section we present the use of the MCDA R package on a didactic MCDA problem which has been widely discussed in the literature, namely the choice of a sports car (see Bouyssou et al (2000), Chapter 6). We show how the package can be used in the various steps of the MCDA process, which was described in (2.1). In a real-world decision aiding process, there might be round-trips between these different steps, in order, for example, to tune the input and output parameters of the various algorithms.

4.1 Problem description

This example is inspired from Chapter 6 of Bouyssou et al (2000), but in order to illustrate all the steps which we wish to highlight, we take the liberty of slightly modifying the original description. As an illustration of the step "situating the problem" we have the following information. The problem takes place in 1993, when Thierry, a student aged 21, is passionate about sports cars and wishes to buy a middle range, 4 years old car with a powerful engine. He asks an analyst to help him to find the best alternative for his needs. We will play the role of the analyst in this decision aiding process. In a first step, we identify the alternatives and the criteria in a dialogue with Thierry. Three points of view appear to be important to Thierry, which are expressed through five criteria: cost point of view (criterion *g1*), performance of the engine point of view (criteria *g2* and *g3*) and safety point of view (criteria *g4* and *g5*). The list of alternatives and their evaluations on these five criteria is presented in Table 1. Thierry is then asked to express the preferential direction on each of the criteria. He considers that the "cost" criterion (€) and the performance criteria "acceleration" (seconds) and "pick up" (seconds) have to be minimized, whereas the safety criteria "brakes" and "road-hold" have to be maximized. The values of the latter two criteria are average evaluations obtained from multiple qualitative evaluations which have been re-coded as integers between 0 and 4. Further details on these data can be found in Bouyssou et al (2000), Chapter 6. Note here that, in comparison to Bouyssou et al (2000) we removed a10 on purpose from these data, as it will be used later in our decision aiding scenario.

The initial meeting between Thierry and the analyst, as well as the session of identifying the decision alternatives, the criteria on which they are defined and the decision problem correspond each to one activity within the decision aiding process. The first activity is contained within the first step of situating the problem, while the second is contained within the second step of formulating the problem. We illustrate these steps within the decision aiding process through 1.a and 1.b in Figure 2.

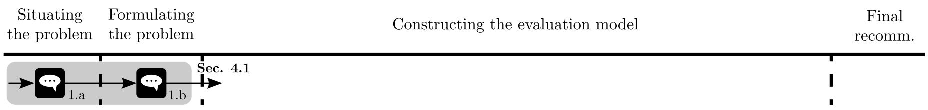

Fig. 2 First part of Thierry's decision aiding process.

Table 1 Data for Thierry's car selection problem.

car ID	car name	cost ($g1$, €)	accel. ($g2$, s)	pick up ($g3$, s)	brakes ($g4$)	road-holding ($g5$)
a01	Tipo	18342	30.7	37.2	2.33	3
a02	Alfa	15335	30.2	41.6	2	2.5
a03	Sunny	16973	29	34.9	2.66	2.5
a04	Mazda	15460	30.4	35.8	1.66	1.5
a05	Colt	15131	29.7	35.6	1.66	1.75
a06	Corolla	13841	30.8	36.5	1.33	2
a07	Civic	18971	28	35.6	2.33	2
a08	Astra	18319	28.9	35.3	1.66	2
a09	Escort	19800	29.4	34.7	2	1.75
a11	P309-16	17537	28.3	34.8	2.33	2.75
a12	P309	15980	29.6	35.3	2.33	2.75
a13	Galant	17219	30.2	36.9	1.66	1.25
a14	R21t	21334	28.9	36.7	2	2.25

4.2 Use of the MCDA R package to support the decision aiding process

Below, we continue by illustrating the use of R and the MCDA package throughout the rest of the decision aiding process. We will divide the discourse further based on the type of evaluation model that will be used. Note also that a file containing the code which we detail step by step hereafter can be found in the directory of the package after its installation. To retrieve the path, the following code can be used:

```
# path to the R script of the example

system.file("examples","articleExample.R",package="MCDA")
```

4.2.1 Filtering rules

First of all, the performances of the cars on the various criteria are loaded into an R data frame. To achieve this, the following code is used:

```
# load performance table csv file
# provided with the MCDA package

f <- system.file("datasets","performanceTable2.csv",package="MCDA")

pT <- read.csv(file = f, header=TRUE, row.names=1)
```

Thierry first wishes to set some rules on the evaluations in order to filter out certain cars. Consequently he asks that only cars respecting the following set of constraints are kept:

$$\begin{aligned} \text{brakes } (g4) &\geq 2 \\ \text{road-hold } (g5) &\geq 2 \\ \text{acceleration } (g2) &< 30 \end{aligned}$$

To achieve this in R, the following steps are proposed:

```
# filter out cars which do not
# respect Thierry's initial rules

fPT <- pT[(pT$g4>=2 & pT$g5>=2 & pT$g2 < 30), ]
```

Furthermore, Thierry notices that car a11 (P309-16) is at least as good as car a14 (R21t) on all the criteria, and thus he wishes to remove the latter.

```
# drop car a14 from the table

fPT <- fPT[!(rownames(fPT) %in% "a14"), ]
```

The resulting filtered performance table is shown by typing `fPT` on the command prompt:

```
                g1   g2   g3   g4   g5
a03          16973 29.0 34.9 2.66 2.50
a07          18971 28.0 35.6 2.33 2.00
a11          17537 28.3 34.8 2.33 2.75
a12          15980 29.6 35.3 2.33 2.75
```

Thierry now asks for a graphical representation of the data. We choose to show him first the performances of the remaining alternatives as radar plots. This allows him to compare their performances in a very synthetic way and to become aware of their conflicting evaluations.

To achieve this in R, we first store the preference directions of the criteria (”min” if the criterion has to be minimized, ”max” if it has to be maximized) in a vector:

```
criteriaMinMax <- c("min","min","min","max","max")

names(criteriaMinMax) <- colnames(pT)
```

Radar plots can display the preferred values on the outside of the radar and the less preferred values in the center of the graph. We can use the following code to create a radar plot of the alternatives:

```
library(MCDA)
plotRadarPerformanceTable(fPT, criteriaMinMax,
                          overlay=FALSE, bw=TRUE, lwd =5)
```

The resulting plots (Figure 3) are shown to Thierry. He notices that a12 (P309) is the best car in terms of price and road-hold, but that it has quite bad evaluations for the acceleration, pick-up and brakes criteria. a03 (Sunny) and a11 (P309-16) seem to be much more well-balanced, whereas a07 (Civic) is only good on the acceleration criterion.

All in all, Thierry considers that his filtering rules have probably been too strict, and that he wishes to continue the analysis with all the initial alternatives.

We continue illustrating the decision aiding process in Figure 4. We have now entered the third stage of the process, that of constructing the evaluation model. We denote with 2.a Thierry's decision to use filtering rules and with 2.b the definition of these rules. The construction of the radar plots are depicted through step 2.c, while the decision to not validate the model is given by step 2.d.

4.2.2 Weighted sum

Thierry now proposes to see how the alternatives compare to each other with respect to each criterion. Among other things, he wishes to determine which alternatives have the best and worst evaluations on the criteria.

We therefore suggest to plot the values taken by the alternatives in barcharts, for each of the criteria. Such a function is not implemented in the MCDA package

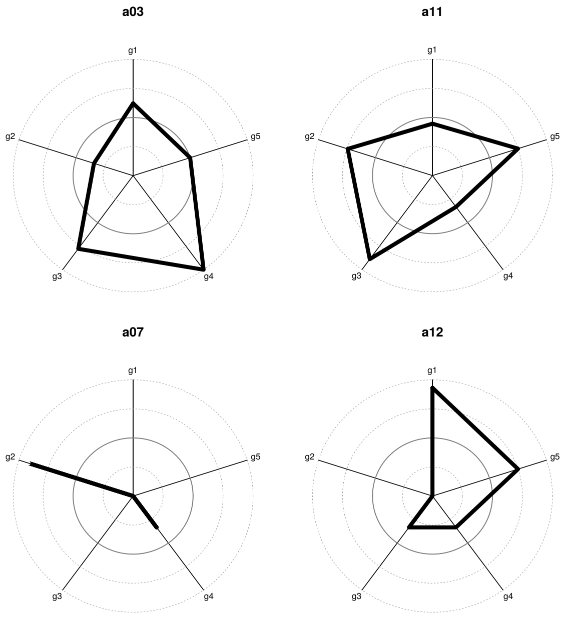

Fig. 3 Radar graphs of the 4 alternatives obtained after the filtering.

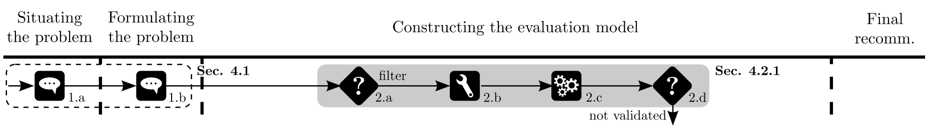

Fig. 4 Second part of Thierry's decision aiding process.

because base R provides this functionality already. We therefore use the following code to generate the 5 plots:

```
par(mfrow=c(2,3))
for (i in 1:dim(pT)[2]){
  yaxis <- range(pT[,i])*c(0.99,1.05)
  if (criteriaMinMax[i] =="min")
    oPT <- pT[order(pT[,i],decreasing=FALSE),]
  else
    oPT <- pT[order(pT[,i],decreasing=TRUE),]
  name <-paste(colnames(pT)[i]," (",criteriaMinMax[i],")", sep="")
  barplot(oPT[,i], main=name, names.arg = rownames(oPT),
          density = i*10, ylim = yaxis, xpd=FALSE)
}
```

Thierry analyzes the resulting plots, shown in Figure 5. The alternatives labeled on the horizontal axis are ordered from left to right according to the preferential direction. He observes, among other things, that alternative a11 (P309-16) seems to be a good alternative, as it performs well on many of criteria (except $g1$ (price)). He seems to be very interested in this alternative and suggests that the rather bad performance on the price criterion could be compensated by the good performances on the other criteria.

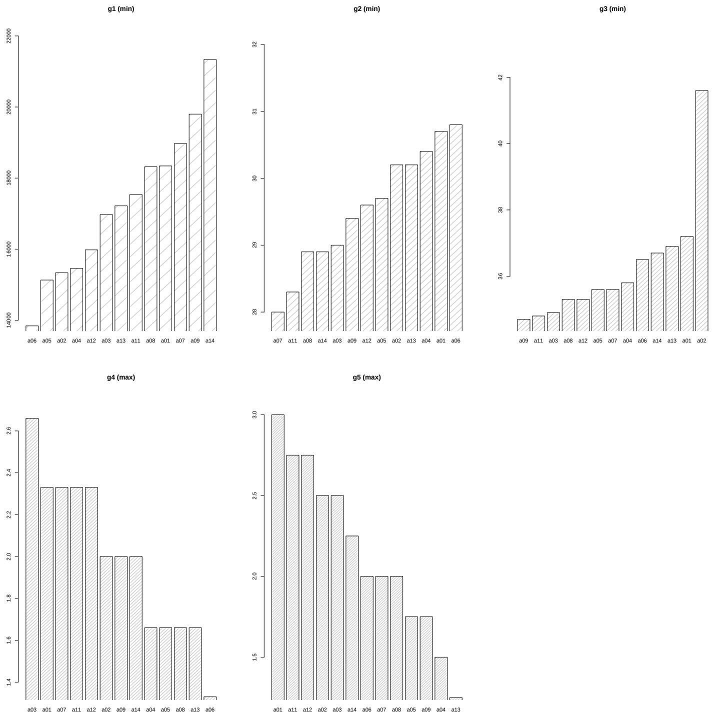

Fig. 5 Bar plots of the performances for each of the 5 criteria.

We deduce from this first discussion with Thierry that he wishes to maximize a quantity which we could call the "value" of the cars. Consequently, our goal in the next steps of the decision aiding process will be to construct a single "super-scale" which reflects the value system of Thierry and his preferences. If we write $\succsim$ for the overall preference relation of Thierry on the set of cars, the goal will be to determine a value function u that allows us to rank the alternatives and represent Thierry's preferences, i.e., which satisfies

$$a \succsim b \iff u(a) \geq u(b).$$

for all alternatives a and b.

The value $u(a)$ depends naturally on the evaluations $\{g_i(a), i = 1, \ldots n\}$ of alternative a (where n is the number of criteria).

Thierry suggests to use a weighted sum to aggregate the various evaluations of the alternatives on the criteria. As described in Bouyssou et al (2000), he chooses to normalize the data (each criterion at a time) by dividing each evaluation by the highest value obtained on the corresponding criterion. He then assigns weights to the criteria according to Table 2. The first three criteria receive negative weights since they have to be minimized.

The above described normalization is done via a function from the MCDA package for R:

	cost ($g1$, €)	accel. ($g2$, s)	pick up ($g3$, s)	brakes ($g4$)	road-hold ($g5$)
weight	-1	-2	-1	0.5	0.5

Table 2 Thierry's naive weights for the weighted sum model.

car ID	car name	cost ($g1$, €)	accel. ($g2$, s)	pick up ($g3$, s)	brakes ($g4$)	road-holding ($g5$)
a10	R19	16966	30	37.7	2.33	3.25

Table 3 Supplementary car for Thierry's car selection problem.

```
# normalization of the data from the performance table

normalizationTypes <- c("percentageOfMax","percentageOfMax",
                         "percentageOfMax","percentageOfMax",
                         "percentageOfMax")

names(normalizationTypes) <- c("g1","g2","g3","g4","g5")

nPT <- normalizePerformanceTable(pT,normalizationTypes)
```

Then, the weighted sum is calculated as follows :

```
# weights and the weighted sum

w <- c(-1,-2,-1,0.5,0.5)
names(w) <- colnames(pT)
ws<-weightedSum(nPT,w)
```

The ranks of the alternatives can be derived from `ws` by typing:

```
# rank the scores of the alternatives
rank(-ws)
```

This produces :

```
a01 a02 a03 a04 a05 a06 a07 a08 a09 a11 a12 a13 a14
  5   6   2  10   7   9   4   8  11   1   3  13  12
```

Thierry observes that the best car, according to this aggregation method, is a11, before a03. He however discovers that one potential car has been forgotten in this decision aiding process. It is given in Table 3.

Note that this car is labelled a10, in accordance with the data from Bouyssou et al (2000).

This car is added to the performance table as follows:

```
# add supplementary car to pT

missing <- c(16966,30,37.7,2.33,3.25)
pT<-rbind(pT,missing)
rownames(pT)[14] <- "a10"
```

This new performance table is then normalized and a weighted sum is calculated on each alternative:

```
# normalization

```

```
nPT <- normalizePerformanceTable(pT,normalizationTypes)

# weighted sum

ws<-weightedSum(nPT,w)
```

The ranking of the alternatives is then shown to Thierry as follows:

```
# rank the scores of the alternatives
rank(-ws)
```

This produces :

```
a01 a02 a03 a04 a05 a06 a07 a08 a09 a11 a12 a13 a14 a10
  6   8   1  11   7  10   5   9  12   2   3  14  13   4
```

This time, car a03 is considered as the best, before car a11. Thierry is surprised that adding alternative a10 to the performance table produced a rank reversal between the first two alternatives of the ranking. This is due to the normalization method, which depends on the data which is present in the performance table. We recommend Thierry to use a more complex model of his preferences, which is independent of the data of the performance table.

We fill the previously presented steps in the decision aiding process in Figure 6.

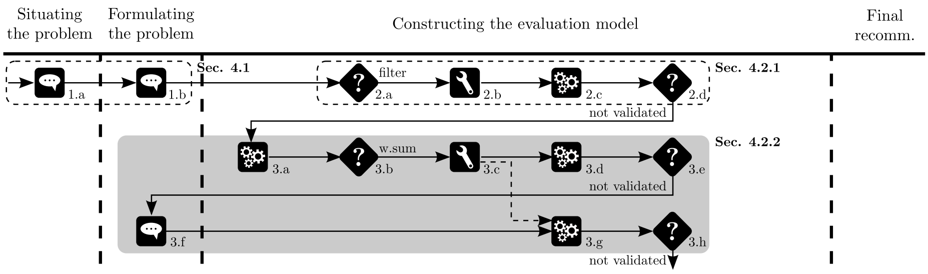

Fig. 6 Third part of Thierry's decision aiding process.

After not validating the previous model, Thierry looks closer at the existing data in 3.a. Based on his remarks, the analyst decides in 3.b to use a weighted sum in order to model his preferences. Thierry gives his relative preferences in 3.c, which are then used in 3.d to compute a ranking of the cars. Before validating this model in 3.e, Thierry realizes that he forgot to include a car in his decision. This takes us back to the second stage of the decision aiding process, as we are identifying other alternatives that need to be included in the model (step 3.f). We then return to the third stage and use the previously constructed model to generate a new ranking of the alternatives in step 3.g. Thierry notices a rank reversal, which prompts him to not validate this model in step 3.h.

4.2.3 MAVT

We choose to construct a model of Thierry's preferences through an additive model, aggregating some marginal value functions on the original evaluations via a weighted sum (the weights representing trade-offs between the criteria).

Now that a motivated choice has been made on the preference model, the next step of this decision aiding process is to elicit the preferences of Thierry (with respect to this additive value model). To determine the marginal value functions, a direct method could be used (by direct numerical estimations, or by indifference judgements). However, as he seems to be quite an expert in sports cars, we decide to switch to an indirect elicitation method, where the shapes of the marginal value functions and the trade-offs are inferred from Thierry's overall preferences on some cars.

The chosen disaggregation method is UTA and was described by Jacquet-Lagrèze and Siskos (1982). It searches for piecewise linear marginal value functions which respect the input preferences expressed by the decision maker. In our case, these a priori preferences are represented by a preorder on a subset of cars, that Thierry knows quite well (the learning set). Thierry chooses to rank 5 cars as follows:

$$a11 \succ a03 \succ a13 \succ a09 \succ a14.$$

In the MCDA package for R, the UTA disaggregation method can be called through the UTA function. Its arguments are the performance table, the preference directions for each criterion, the number of breakpoints for the piecewise linear value functions, a separation threshold (representing the minimal difference in value between two consecutive alternatives from the learning set) and the lower and upper bounds of the criteria scales. For arguments of parsimony, we decide to search for piecewise linear value functions with 1 segment.

```
# ranks of the alternatives

alternativesRanks <- c(1,2,3,4,5)
names(alternativesRanks) <- c("a11","a03","a13","a09","a14")

# number of break points for each criterion : 1 segment = 2 breakpoints

criteriaNumberOfBreakPoints <- c(2,2,2,2,2)
names(criteriaNumberOfBreakPoints) <- colnames(pT)

# lower bounds of the criteria for the determination of value functions

criteriaLBs=apply(pT,2,min)
names(criteriaLBs) <- colnames(pT)

# upper bounds of the criteria for the determination of value functions

criteriaUBs=apply(pT,2,max)
names(criteriaUBs) <- colnames(pT)

# the separation threshold

epsilon <- 0.01

x<-UTA(pT, criteriaMinMax,
       criteriaNumberOfBreakPoints, epsilon,
       alternativesRanks = alternativesRanks,
       criteriaLBs = criteriaLBs, criteriaUBs = criteriaUBs)
```

The calculation is successful, and the result is shown by typing x on the command prompt:

```
$optimum
[1] 0

$valueFunctions
$valueFunctions$g1
   [,1]        [,2]
x 21334 1.38410e+04
y     0 4.61114e-01

$valueFunctions$g2
  [,1] [,2]
x 30.8   28
y  0.0    0

$valueFunctions$g3
  [,1]       [,2]
x 41.6 34.7000000
y  0.0  0.2049873

$valueFunctions$g4
  [,1] [,2]
x 1.33 2.66
y 0.00 0.00

$valueFunctions$g5
  [,1]      [,2]
x 1.25 3.2500000
y 0.00 0.3338987


$overallValues
    a03     a09     a11     a13     a14
0.67611 0.38286 0.68611 0.39286 0.31252

$ranks
a03 a09 a11 a13 a14
  2   4   1   3   5

$errors
a03 a09 a11 a13 a14
  0   0   0   0   0

$Kendall
[1] 1
```

The structure returned by the UTA function is a list / dictionary containing the following main elements:

- `optimum` : the value of the objective function in the UTA algorithm;
- `valueFunctions` : a list containing the value function for each criterion;
- `overallValues` : the overall values of the learning set;
- `ranks` : the ranks of the elements of the learning set;
- `error` : the errors which have to be added to the overall values of the alternatives of the learning set in order to respect the input order;
- `Kendall` : Kendall's rank correlation index between the input and the output ranking of the learning set.

We can observe that Thierry's ranking is compatible with the chosen model (Kendall's rank correlation index equals 1, there are no errors, and the optimal value of the objective function equals 0). We plot the obtained value functions as follows:

```
# plot the piecewise linear value functions

plotPiecewiseLinearValueFunctions(x$valueFunctions)
```

The resulting marginal value functions are shown on Figure 7. The maximal value on the ordinate axis represents the trade-off weight in the aggregation.

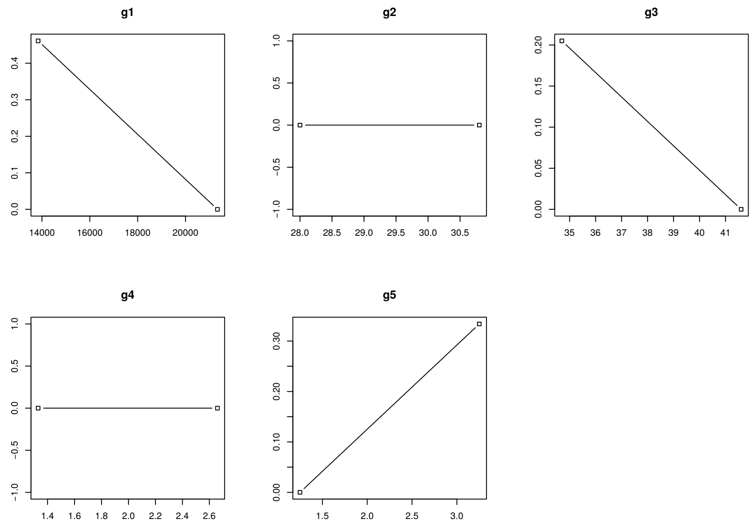

Fig. 7 Marginal value functions for the criteria with respect to the initial reference ranking.

Thierry is not totally convinced by this preference model. He agrees that the price is very important in the aggregation, but he considers that the acceleration should also be considered to discriminate between alternatives. He decides to modify his a priori ranking by adding two alternatives:

$$a11 \succ a03 \succ a08 \succ a04 \succ a13 \succ a09 \succ a14.$$

The following lines of code are entered in R :

```
# ranks of the alternatives for the second try

alternativesRanks <- c(1,2,3,4,5,6,7)
names(alternativesRanks) <- c("a11","a03","a08","a04","a13","a09","a14")

x2<-UTA(pT, criteriaMinMax,
       criteriaNumberOfBreakPoints, epsilon,
       alternativesRanks = alternativesRanks,
       criteriaLBs = criteriaLBs, criteriaUBs = criteriaUBs)

# plot the piecewise linear value functions

plotPiecewiseLinearValueFunctions(x2$valueFunctions)
```

The new calculations generate the value functions represented in Figure 8. This time Thierry validates the model, as the acceleration criterion plays a significant role in the aggregation.

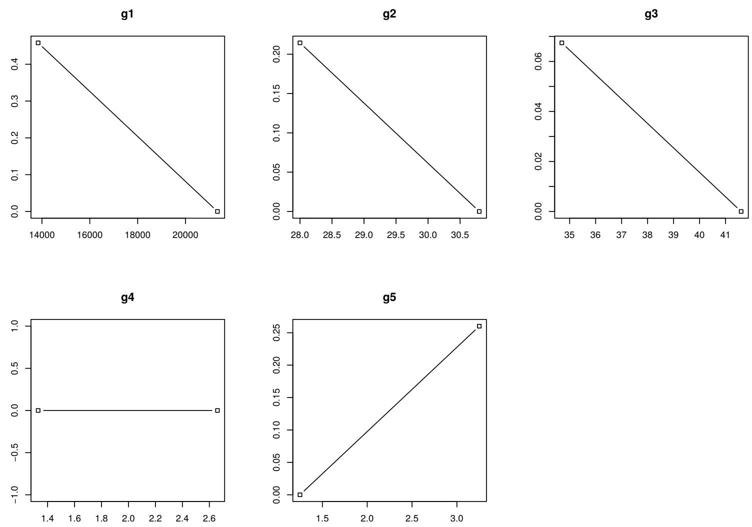

Fig. 8 Marginal value functions for the criteria after the update of the a priori ranking of Thierry.

Now that a model of Thierry's preferences has been found, these marginal value functions can be used to rank all the cars. This is done by applying the value functions on the original performance table, and by performing an additive aggregation of the marginal values vector, for each alternative. In the MCDA package for R, this is done as follows:

```
# apply the value functions on the original performance table

tPT <- applyPiecewiseLinearValueFunctionsOnPerformanceTable(
               x2$valueFunctions,
               pT
               )

# calculate the overall score of each alternative

mavt <- weightedSum(tPT,rep(1,5))
```

The second argument of the `weightedSum` function is a vector of equi-important weights, as the trade-off weight is already contained in the value functions. The output of the `weightedSum` function is the "super-scale" we were mentioning earlier (page 13). It indicates, provided it can be considered as accurate, the value of each car, according to Thierry's preference model.

These overall scores can be obtained by typing `mavt` in the command prompt:

```
      a01       a02       a03       a04       a05       a06       a07
0.4611504 0.5752482 0.6324617 0.4788993 0.5870830 0.6054313 0.5150286
      a08       a09       a10       a11       a12       a13       a14
```

```
0.4888993 0.3334222 0.6265008 0.6850774 0.6758266 0.3434222 0.3234222
```

We can observe that the car which obtains the highest score is a11 (P309-16). This confirms Thierry's preliminary analysis.

Note here that after the confrontation of the decision maker to the overall scores, one could easily imagine a scenario where Thierry is not satisfied with the result, and that he wishes to update the preference model. To avoid adding complexity to this fictive decision aiding process, we suppose that Thierry is satisfied with the scores.

A further step of the decision aiding process is to analyze the result, and to plot some graphical summaries of the outputs. In a more complex process, this phase could also be completed by a sensitivity or robustness analysis. It could also be the right place to compare the outputs of various aggregation models (for example, the ELECTRE methods, see Bouyssou et al (2000), Chapter 6, or Meyer and Bigaret (2012a) for the PROMETHEE methods).

Here, we mainly confront Thierry with the ranking of the cars according to their overall scores,

We complete the R code by calling a function to plot the ranking of the cars:

```
plotAlternativesValuesPreorder(mavt, decreasing=TRUE)
```

Figure 9 shows the first 7 positions of this ranking.

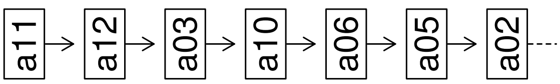

Fig. 9 The ranking obtained by the additive value model.

According to this model, car a11 is ranked first, before car a03 and a12.

We finalize the depiction of the decision aiding process of this illustrative example in Figure 10.

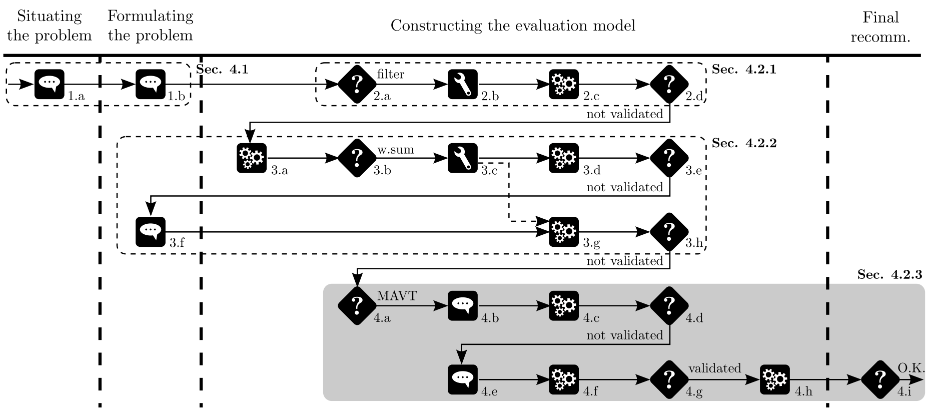

Fig. 10 Last part of Thierry's decision aiding process.

Following the decision to use another preference model, the choice of MAVT is given in step 4.a. The preference elicitation step is depicted in step 4.b, while the application of UTA to generate the marginal value functions is depicted in step 4.c. The illustration of these functions and the subsequent decision of Thierry to not validate this result is given in 4.d. In 4.e we update the partial ranking given by Thierry, in 4.f we generate the updated marginal value functions, while in 4.g Thierry validates the model. We then continue with generating the final ranking in step 4.h, using this model, while in 4.i we step into the final recommendation phase, where Thierry is confronted with this ranking. Thierry validates the recommendation and therefore the process is finished.

5 Discussion and conclusion

In this paper, we proposed to support the MCDA process throughout all of its steps by use of a single environment, the R statistical software. Currently, analysts and the DMs have to resort to using multiple tools at different stages of the decision aiding process, moving from one to the other, adding an additional level of difficulty. The choice of using R throughout the process was motivated by its focus towards data analysis, its open-source and package-based philosophy, as well as its large community of users and contributors. Furthermore, we have developed the MCDA package which seeks to encompass as many of the MCDA algorithms as possible in order to provide additional support. We have illustrated the use of R and the MCDA package using a well-known illustrative example from the literature and in addition highlighting the different steps that were undertaken within the MCDA process. We have shown that, even when the process is complex, by using R and the MCDA package we are able to successfully achieve a solution.

While currently, the MCDA package contains algorithms linked to only a few methods, we wish to continue developing it in the future so that as many of the MCDA algorithms can be found within it. Furthermore, functions linked to the presentation of the results, for instance graphically, will also be added to complement the existing ones. We additionally wish to continue applying this methodology and these tools to other practical applications.

References

Baizyldayeva U, Vlasov O, Kuandykov AA, Akhmetov TB, et al (2013) Multicriteria decision support systems: Comparative analysis. Middle-East Journal of Scientific Research 16(12):1725–1730

Belton V, Stewart T (2002) Multiple criteria decision analysis: an integrated approach. Springer Science & Business Media

Bisdorff R, Dias LC, Meyer P, Pirlot M, Mousseau V (2015) Evaluation and Decision Models With Multiple Criteria: Case Studies. International Handbooks on Information Systems, Springer, Berlin

Bogetoft P, Otto L (2015) Benchmarking-Benchmark and Frontier Analysis Using DEA and SFA. https://cran.r-project.org/package=Benchmarking

Bouyssou D, Marchant T, Pirlot M, Perny P, Tsoukiàs A, Vincke P (2000) Evaluation and decision models: A critical perspective. Kluwer, Dordrecht

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

Clemen RT, Reilly T (2001) Making hard decision with decision tools. South-Western Cengage Learning, Mason, Ohio

Coutinho-Rodrigues J, Simão A, Antunes CH (2011) A gis-based multicriteria spatial decision support system for planning urban infrastructures. Decision Support Systems 51(3):720–726

Dias LC, Mousseau V (2003) Iris: A dss for multiple criteria sorting problems. Journal of Multi-Criteria Decision Analysis 12(4-5):285–298

Figueira J, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys, vol 78. Springer Science & Business Media

Gentry J, Long L, Gentleman R, Falcon S, Hahne F, Sarkar D, Rgraphviz KH (2009) Provides plotting capabilities for r graph objects. R package version 2(0)

Gentry J, Gentleman R, Huber W (2016) How to plot a graph using rgraphviz. http:// wwwbioconductororg/packages/release/bioc/vignettes/R-graphviz/inst/doc/Rgraphviz pdf

Grabisch M, Kojadinovic I, Meyer P (2006) Using the Kappalab R package for capacity identification in choquet integral based maut. In: Proceedings of the 11th international conference on information processing and management of uncertainty in knowledge-based systems, pp 1702–1709

Grabisch M, Kojadinovic I, Meyer P (2015) kappalab - Non-Additive Measure and Integral Manipulation Functions. https://cran.r-project.org/package=kappalab

Guitouni A, Martel JM, Vincke P, North P, Val-bblair O (1998) A framework to choose a discrete multicriterion aggregation procedure. Defence Research Establishment Valcatier (DREV)

Hodgett RE (2016) Comparison of multi-criteria decision-making methods for equipment selection. The International Journal of Advanced Manufacturing Technology 85(5):1145–1157, DOI 10.1007/s00170-015-7993-2

Hodgett RE, Martin EB, Montague G, Talford M (2014) Handling uncertain decisions in whole process design. Production Planning & Control 25(12):1028–1038, DOI 10.1080/09537287.2013.798706

Hwang CL, Yoon K (1981) Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey (Lecture Notes in Economics and Mathematical Systems). Springer

IEEE Spectrum (2016) The 2016 top programming languages, available from: http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages

Ihaka R, Gentleman R (1996) R: a language for data analysis and graphics. Journal of computational and graphical statistics 5(3):299–314

International Society on Multiple Criteria Decision Making (2014) Multiple criteria decision making website, available from: http://www.mcdmsociety.org/content/software-related-mcdm

Ishizaka A, Nemery P (2013) Multi-method platforms. Multi-Criteria Decision Analysis: Methods and Software pp 275–287

Jacquet-Lagrèze E, Siskos Y (1982) Assessing a set of additive utility functions for multicriteria decision making: the UTA method. European Journal of Operational Research 10:151–164

Keeney R, Raiffa H (1976) Decisions with multiple objectives: Preferences and value tradeoffs. J. Wiley, New York

Kostkowski M, Slowinski R (1996) UTA+ application (v. 1.20)-user's manual. Document du LAMSADE 95

Lahdelma R, Salminen P, Hokkanen J (2014) Using Multicriteria Methods in Environmental Planning and Management. Environmental Management 26(6):595–605, DOI 10.1007/s002670010118

Leistedt B (2011) UTAR library for MCDA. https://cran.r-project.org/package=UTAR

Leroy A, Mousseau V, Pirlot M (2011) Learning the parameters of a multiple criteria sorting method. In: Brafman RI, Roberts FS, Tsoukiàs A (eds) ADT, Springer, Lecture Notes in Computer Science, vol 6992, pp 219–233

Make It Rational (2016) Make it rational website, available from: makeitrational.com/

Mayag B, Cailloux O, Mousseau V (2011) Mcda tools and risk analysis: the decision deck project. Advances in Safety, Reliability and Risk Management: ESREL 2011 p 377

Meyer P, Bigaret S (2012a) diviz: a software for modeling, processing and sharing algorithmic workflows in MCDA. Intelligent Decision Technologies 6(4):283–296, doi:10.3233/IDT-2012-0144

Meyer P, Bigaret S (2012b) RXMCDA - Functions to Parse and Create XMCDA Files. https://cran.r-project.org/package=RXMCDA

Meyer P, Olteanu AL (2017) Integrating large positive and negative performance differences into multicriteria majority-rule sorting models. Computers & Operations Research pp 216 – 230

Meyer P, Bigaret S, Hodgett R, Olteanu AL (2017) MCDA : Functions to Support the Multicriteria Decision Aiding Process. https://cran.r-project.org/package=MCDA

Mousseau V, Slowinski R, Zielniewicz P (1999) ELECTRE TRI 2.0 a methodological guide and user's manual. Document du LAMSADE 111:263–275

Mousseau V, Slowinski R, Zielniewicz P (2000) A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. Computers & Operations research 27(7):757–777

Mustajoki J, Marttunen M (2013) Comparison of multi-criteria decision analytical software. Finnish Environment Institute

Papamichail KN, French S (2013) 25 years of MCDA in nuclear emergency management. IMA Journal of Management Mathematics 24(4):481 – 503

Piatetsky G (2016) R, Python duel as top analytics, data science software – kdnuggets 2016 software poll results, available from: http://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html

R Development Core Team (2008) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org, ISBN 3-900051-07-0

Roy B (1991) The outranking approach and the foundations of electre methods. Theory and Decision 31(1):49–73, DOI 10.1007/BF00134132

Roy B (1996) Multicriteria Methodology for Decision Aiding. Kluwer Academic, Dordrecht

Saaty TL (1980) The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation (Decision Making Series). Mcgraw-Hill (Tx)

Simon HA (1976) Administrative behavior; a study of decision-making processes in administrative organization-3

Siraj S, Mikhailov L, Keane JA (2015) Contribution of individual judgments toward inconsistency in pairwise comparisons. European Journal of Operational Research 242(2):557–567, DOI 10.1016/j.ejor.2014.10.024

Sobrie O, Mousseau V, Pirlot M (2013) Learning a majority rule model from large sets of assignment examples. In: ADT, Springer, Lecture Notes in Computer Science, vol 8176, pp 336–350

Statistical Design Institute (2016) Topsis website, available from: http://www.stat-design.com/Software/TOPSIS.html

Taillandier P, Stinckwich S (2011) Using the promethee multi-criteria decision making method to define new exploration strategies for rescue robots. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, pp 321–326, DOI 10.1109/SSRR.2011.6106747

Tervonen T (2012) JSMAA: open source software for smaa computations. Int J Syst Sci 2012:1–13

TransparentChoice Ltd (2016) Transparent choice website, available from: https://www.transparentchoice.com

Tsoukias A (2007) On the concept of decision aiding process: an operational perspective. Annals of Operations Research 154:3 – 27

Tversky A, Kahneman D (1981) The framing of decisions and the psychology of choice. Science 211(4481):453–458

Venables B, Smith D, Gentleman R, Ihaka R (1998) Notes on R: a programming environment for data analysis and graphics. University of Auckland

von Winterfeldt D, Edwards W (1986) Decision Analysis and Behavorial Research. Cambridge University Press, Cambridge

Wahlster P, Goetghebeur M, Kriza C, Niederländer C, Kolominsky-Rabas P (2015) Balancing costs and benefits at different stages of medical innovation: a systematic review of multi-criteria decision analysis (MCDA). BMC Health Services Research 15(1):1 – 12

Weistroffer HR, Smith CH, Narula SC (2005) Multiple criteria decision support software. In: Multiple criteria decision analysis: state of the art surveys, Springer, pp 989–1009

Yatsalo B, Didenko V, Gritsyuk S, Sullivan T (2015) Decerns: A framework for multi-criteria decision analysis. International Journal of Computational Intelligence Systems 8(3):467–489