

Chapter 1

Illustrating the support of the MCDA process by the diviz workbench

Sébastien Bigaret and Patrick Meyer

Abstract In this chapter, we illustrate how the MCDA process can be supported by the use of a decision aid software called diviz. The diviz workbench allows to build, execute and share complex *workflows* of MCDA algorithms, and as such, is a convenient tool to help the analyst to perform the decision aid process. We start by a presentation of diviz, before switching to the detailed description of a fictive MCDA process, based on a classical example from the MCDA literature. We show how each major step of this process can be backed up by diviz, and how the software can help to build the final recommendation.

1.1 Introduction

The diviz workbench is one of the initiatives of the [Decision Deck Consortium](#) [2009]. It facilitates the use of algorithmic resources from the field of Multicriteria Decision Aid (MCDA). Before the birth of diviz and the tools it relies on, the analyst who had to perform a decision aid process for the decision maker was regularly facing the following difficulties :

Patrick Meyer
Institut Télécom, Télécom Bretagne
UMR CNRS 6285 Lab-STICC
Technopôle Brest-Iroise CS 83818, F-29238 Brest Cedex 3
Université européenne de Bretagne, e-mail: patrick.meyer@telecom-bretagne.eu

Sébastien Bigaret
Institut Télécom, Télécom Bretagne
UMR CNRS 6285 Lab-STICC
Technopôle Brest-Iroise CS 83818, F-29238 Brest Cedex 3
Université européenne de Bretagne, e-mail: sebastien.bigaret@telecom-bretagne.eu

1. different MCDA techniques were generally implemented in separate software products, with *heterogeneous* user interfaces;
2. testing multiple MCDA algorithms on one problem instance was not easy, because of the various input *data formats* required by the software applications –this problem is specifically addressed in Chapter ??;
3. a lot of MCDA algorithms which were presented and published in scientific articles were not easily *available* and consequently often only used by their authors;
4. several MCDA software products were not *free* (neither from a financial, nor from an open-source point of view), which can be considered as a weakness for their large dissemination.

In other scientific research fields, as, e.g., statistics or data mining, there exist software platforms which allow to easily compare different analysis methods and to test them on a given data set inside a common framework. Among the most famous ones, one can cite platforms such as the GNU R statistical system by the [R Development Core Team \[2005\]](#) or the Weka suite of machine learning software by [Hall et al. \[2009\]](#). Both of these suites are open-source and independent from the operating system, which has certainly contributed to their large dissemination and acceptance among many researchers and users.

In order to overcome the earlier mentioned difficulties linked to the software situation in the field of MCDA, a group of researchers has got together to create the [Decision Deck Consortium \[2009\]](#). Its objective is to collaboratively develop open-source software tools implementing MCDA techniques. As such, its purpose is to provide effective tools for at least three types of users:

- practitioners who use MCDA tools to support actual decision makers involved in real world decision problems;
- teachers who present MCDA algorithms in courses;
- researchers who want to test, share and compare algorithms or to develop new ones.

In this chapter we focus on *diviz*, one of the software initiatives of the Decision Deck project, which eases the use of algorithmic resources from the field of MCDA. The *diviz* tool is an easy to use software to build, execute and share complex workflows of MCDA algorithms. In the literature, such workflows are often called *methods* (consider, e.g., the ELECTRE method by [Roy \[1968\]](#), the UTA method by [Jacquet-Lagrèze and Siskos \[1982\]](#), etc). One of the main features of *diviz* is that it facilitates the construction of these classical MCDA *methods*, as well as derivatives and original ones, by combining various elementary calculation components via an intuitive graphical user interface.

The *diviz* tool uses extensively two other outcomes of the Decision Deck project, which we also present shortly in the sequel:

- XMCDA: a standardized XML recommendation to represent objects and data structures coming from the field of MCDA. Its main objective is to allow different MCDA algorithms to *interact* and to analyze a problem instance stored in XMCDA by various MCDA algorithms;
- XMCDA web-services: distributed open-source computational MCDA resources.

The goal of this chapter is to show how diviz can be used to support the MCDA process. We therefore present in Section 1.2 the workbench together with the resources it relies on. Then, in Section 1.3 we illustrate the use of diviz to back up a fictive decision aid process concerning a classical MCDA problem which deals with the choice of a sports car.

1.2 diviz for dummies

In this section we present the main features of diviz, along with its practical usage, as well as the external resources it relies on. We also outline the original work methodologies which arise from its use.

1.2.1 Use of diviz

The diviz workbench allows to build, execute and share complex *workflows* of MCDA algorithms. The design of these workflows is done via an intuitive graphical user interface, where each algorithm is represented by a box which can be linked to data files or supplementary calculation elements by using connectors. Thus, the construction of complex sequences of algorithms does not require any programming skills, but only necessitates to understand the functioning of each calculation module.

Figure 1.1 shows the diviz workbench.

- On the left side, a tree presents the list of the opened workflows, along with their execution results.
- The upper-middle panel contains the currently selected workflow: it shows either the *design panel*, i.e. the workflow while it is designed with its input files and programs, or the workflow that has been executed when consulting an execution result.
- The lower-middle panel appears only when viewing an execution result, it shows the results of every program in the workflow.
- On the right side, all available programs are organized by theme (e.g. aggregation, outranking, elicitation).

The diviz software can be downloaded from <http://www.diviz.org>.

[h!]

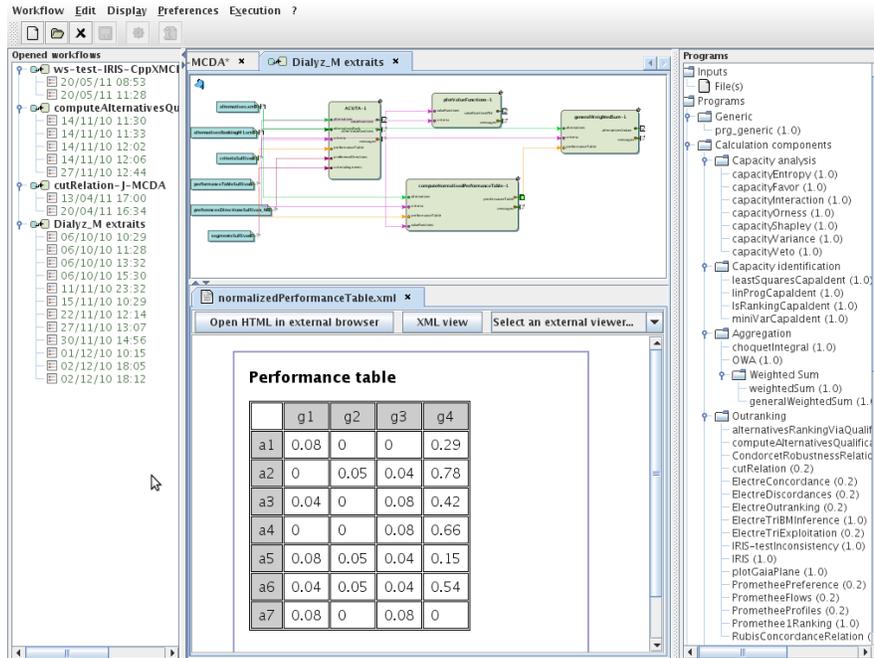


Fig. 1.1 A typical diviz workbench, here showing a workflow and one of its execution results.

Workflow design

The design of the MCDA workflows is performed via an intuitive graphical user interface, where each algorithm is represented by a box which can be linked to data files or supplementary calculation elements by using connectors (see Figure 1.2 for a close view on the design panel).

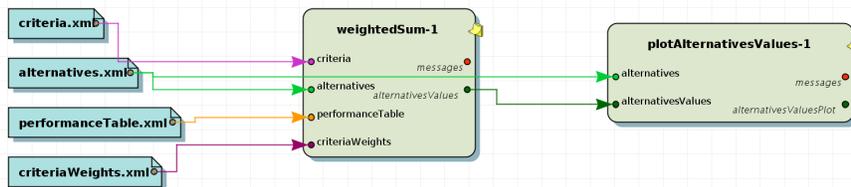


Fig. 1.2 An MCDA workflow representing the input data (left) and a weighted sum (middle) combined with a module which plots a graphical representation of the output (right).

The inputs and outputs of these elementary components can be manifold and can correspond to various MCDA concepts or data elements. To illustrate this, consider the following example.

Example

diviz allows to use a component called `weightedSum`. This element calculates the weighted sum of alternatives' performances with respect to a set of weights associated with a list of criteria. Consequently, `weightedSum` requires four inputs: the description of the criteria, the description of the alternatives, the performance table containing the numerical evaluation of each alternative on each of the criteria, and the numerical weights associated with the criteria. The main output of this component are the overall values of the input alternatives via the weighted sum aggregation operator (see Figure 1.2 for an example of the use of the `weightedSum` module).

To construct a new MCDA workflow, the user chooses the modules in the list of the available calculation elements and he drags and drops them into the design panel. Then he adds data files to the workspace and connects them appropriately to the inputs of the elements. Finally he connects the inputs and outputs of the components to complete the structure of the workflow.

Execution and results

Once the design of the MCDA workflow is finished, the user can execute it in order to obtain the possibly multiple outputs of the algorithms. These calculations are performed on high performance computing servers through the use of Decision Deck's XMCDA web-services (see Section 1.2.2 describing the external resources used by diviz). As a consequence, diviz does not *physically* contain any calculation modules, and requires a connection to the Internet to access these resources.

After the execution of the workflow, the outputs of each of the components can be viewed and analyzed by the user. Some of these outputs might represent intermediate results of the various calculation steps of the workflow. This feature helps tuning the parameters of the various algorithms of the workflow. We illustrate this in the following example.

Example

Consider the following workflow (typically a UTA-like disaggregation method): a first module determines piecewise linear value functions on basis of a ranking of alternatives provided by the user; a second module transforms a performance table by applying these value functions on the performances of the

alternatives; a third module calculates the sum of these performances for each of the alternatives; a fourth module draws a ranking of the alternatives on basis of the overall values previously computed. The intermediary results are the value functions, the transformed performance table and the overall values of the alternatives. As each of these elements is explicitly available for the user, first he can gain a deeper understanding of the decision aid method which he has constructed, and second the fine-tuning of the input parameters (here, the number of segments of the value functions to be constructed, the ranking provided by the user, etc.) is facilitated.

In diviz the history of the past executions is kept in the software and can at any moment be viewed by the user. More precisely, if a workflow is modified, the former executions' results and their associated workflows are still available –this includes the values of the programs' parameters at the time the workflow was executed. This also contributes to the good understanding of the constructed chain of algorithms and helps calibrating the parameters of the workflow's elementary components.

Available algorithmic components

The algorithmic elements available in diviz are web-services proposed by the Decision Deck Consortium. At the time of writing, about 100 such components can be used, which can be divided into four main categories:

1. *calculation components* containing aggregation operators, disaggregation techniques, post-analysis elements, pre-processing tools, etc.;
2. *methods* containing full MCDA methods;
3. *visualization components* containing modules allowing to represent graphically certain input and output data elements;
4. *reporting components* containing techniques to create aggregated reports of multiple output data pieces.

These programs allow to reconstruct for example classical MCDA methods like the ELECTRE series by Roy [1968], the PROMETHEE series by Brans and Vincke [1985] or UTA-like techniques by Jacquet-Lagrèze and Siskos [1982]. Next to that, more recent techniques linked to the elicitation of capacities in Choquet integral-based MCDA can also be used [see Grabisch et al., 2008], as well as some inverse analysis techniques by Bisdorff et al. [2009].

The complete list of the available calculation components is given on diviz's website, where each of them is documented and details are given on their inputs, outputs and parameters.

Comparison of “methods”

Next to designing and executing MCDA workflows, diviz can also be a convenient tool to compare the outputs of various methods and algorithms on the same input data.

Earlier, such a task has been far from easy, as no unified software platform for MCDA techniques existed. However, with diviz and its possibility to construct complex workflows, it is easy to connect a dataset of a specific decision problem to various workflows in a single workspace, each of them representing a different MCDA method, and to compare their outputs. This is clearly a very simple way to check the robustness of the output recommendation of an analysis with respect to the choice of the decision aid technique. Note that in practice this possibility has to be used carefully, as the preferential parameters used by two MCDA methods may have very different meanings (for example, the tradeoff weights used by an additive model may not be compatible with an outranking-based technique).

Workflow sharing and dissemination

The diviz software enables to export any workflow, with or without the data, as an archive. The latter can then be shared with any other diviz user, who can import it (by loading the archive) into the software and continue the development of the workflow or execute it on the original data.

Consequently, diviz can be used as a convenient dissemination tool: first, in combination with a research article, the authors of a new MCDA technique or an experiment could propose the corresponding diviz workflow together with an appropriate data set as supplementary electronic material with their article. Second, in a practical context, MCDA analysts could also be willing to share the algorithmic treatment they have performed with the various stakeholders of the process. This feature may contribute to a larger dissemination of new algorithms and might facilitate their acceptance among many researchers and users.

In this context, the examples which are presented in Sections [1.3.1](#), [1.3.2](#), [1.3.3](#) and [1.3.4](#) are available as downloadable archives from the diviz website, and can be tested by any interested reader in their own copy of diviz.

1.2.2 Resources used by diviz

We made mention earlier of the fact that diviz relies on further outcomes of the Decision Deck Consortium, as for example the XMCDA web-services, which are the calculation resources implementing MCDA algorithms. In this

section we briefly recall the purpose of XMCDA, present the XMCDA web-services, and show how these two initiatives contribute to the diviz tool.

The XMCDA data format has been proposed by the Decision Deck Consortium to solve the problem of the heterogeneity of the data formats used by the available software programs. Indeed, this problem prevented existing tools to inter-operate, and it also made it necessary to re-encode all existing data each time one wants to use a new tool, so that they comply to the new expected data format.

The XMCDA markup language is written in XML (<http://www.w3.org/XML/>), a general-purpose syntax for defining markup languages. XMCDA allows to store data and concepts related to a MCDA problem, as for example descriptions of the criteria and alternatives, performances of alternatives on the criteria, preferences of decision makers, . . . For further details on the use of XMCDA, we recommend that the reader refers to Chapter of this book.

The various calculation modules which are available in diviz use the XMCDA standard to represent their input and output data. Consequently, they are interoperable, and they can be chained in complex workflows.

The XMCDA web-services are an initiative of the Decision Deck project and they are all available as calculation resources in diviz. From a general point of view, a web-service is an application which can be accessed via the Internet and is executed on a remote system. One of the great advantages of such online programs is their availability to anyone at any time and any place and on any computer which is connected to the Internet; they are accessible through the same mechanism, without requiring the (sometimes complicated) installation of each of them, and their dependencies, on the user's computer. Furthermore, the user of web-services is always sure to use the latest available version of the programs.

XMCDA web-services have furthermore the following properties:

- they are released under an open-source licence;
- they “speak” XMCDA: their inputs and outputs are formatted using this standard. This guarantees that all web-services are able to inter-operate;
- they are asynchronous: each of them exposes a method for submitting a problem and an other one for retrieving the results. Consequently, one can submit long-running tasks (hours, or even days) and retrieve them afterwards, without having to stay connected in-between (in this case, the user regularly polls for available results until their are available);
- they can be made with any programming language, with the current limitation that the program should be runnable on a Linux machine, because they are deployed on Linux servers (we really mean *runnable* here: they can be developed on the user's favorite operating system as long as the language used is also available for Linux, which is in fact the case for most languages nowadays). This way everyone can participate to the web-services effort using their favorite language.

From an algorithmic point of view, the XMCDA web-services propose elementary calculation steps, which, if properly chained, can rebuild MCDA methods in complex workflows, as for example in the diviz workbench.

*

This introductory presentation of the diviz workbench and its underlying resources should show the coherence and the versatility of this ecosystem, which among other things simplifies the use of MCDA algorithms.

1.3 diviz to support the MCDA process

In this section we present the use of diviz on a classical 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 workbench can be used in the various steps of the MCDA process. In a real-world decision aid 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. The goal of this section being to highlight the advantages of using diviz, we will not discuss the decision aid process and its variants here, which are thoroughly presented in Chapter .

We first present the context of the example and then illustrate the use of diviz in a fictitious decision aid process involving four main phases: preparatory analysis of the problem, preference elicitation, aggregation, and finally analysis of the results. This process 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.

The problem we are dealing with here takes place in 1993, where 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 selects five criteria related to: cost (criterion $g1$), performance of the engine (criteria $g2$ and $g3$) and safety (criteria $g4$ and $g5$). The list of alternatives and their evaluations on these five criteria is presented in Table 1.1. 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. Note that 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].

We will play the role of the analyst in this decision aid process.

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

Table 1.1 Data for Thierry’s car selection problem

1.3.1 Analysis of the problem and the underlying data

In this first step, Thierry 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 rules are kept :

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

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.

To help Thierry, in *diviz*, we construct a simple workflow which only uses the calculation element `performanceTableFilter`, which is connected to the XMCDAs files containing the definitions of the alternatives and the criteria and a file containing the performance table. Alternative a14 has been left out in this filtering. The workflow is represented on Figure 1.3.

The resulting performance table is shown on Figure 1.4.

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

In *diviz*, we complete the previous workflow with the calculation element `plotStarGraphPerformanceTable`, which is connected to the output of the filtering module and the definitions of the alternatives and criteria. In the XMCDAs file containing the criteria, we furthermore specify the preference directions on the various criteria. This allows to put the preferred values in

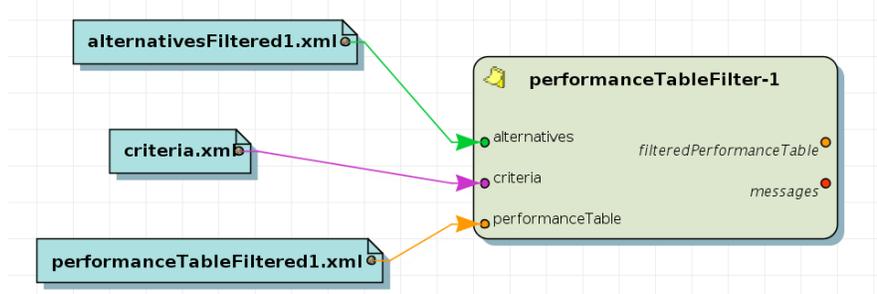


Fig. 1.3 The workflow representing the input data (left) and the module to filter out alternatives according to the above rules (right).

Filtered performance table

	g1	g2	g3	g4	g5
a11	17537	28.3	34.8	2.33	2.75
a03	16973	29	34.9	2.66	2.5
a07	18971	28	35.6	2.33	2
a12	15980	29.6	35.3	2.33	2.75

Fig. 1.4 The performance table resulting from the filtering step.

the star graphs on the outside of the star, whereas the less preferred values are situated in the center of the graph. The workflow is represented on Figure 1.5.

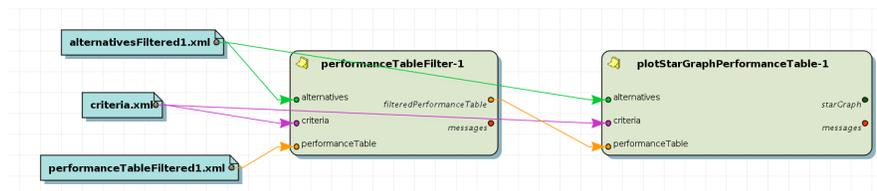


Fig. 1.5 The workflow representing the input data (left), the filtering module (center) and the module to plot star graphs of the alternatives (right).

The resulting plots are shown to Thierry. On Figure 1.6 we show these star graphs for the 4 remaining alternatives. Thierry can easily notice that a12 (P309) is the best car (among the 4 remaining cars) 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 for the acceleration criterion.

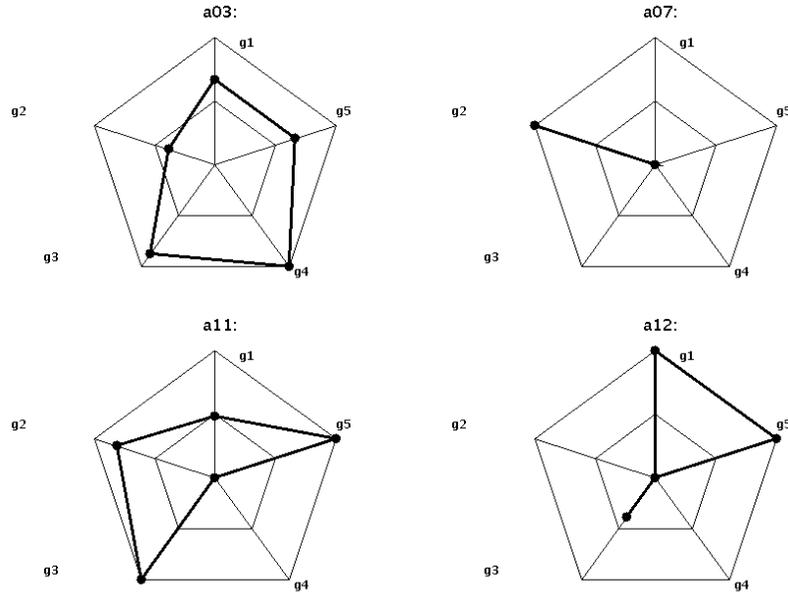


Fig. 1.6 Star graphs of alternatives a03 (Sunny), a07 (Civic), a11 (P309-16) and a12 (P309).

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. He now proposes to see how the alternatives compare one to another on each criterion. Among other things, he wishes to determine which alternatives have the best and worst evaluations on the criteria. We therefore construct him in *diviz* a new workflow which uses the whole performance table as input and plots the values taken by the alternatives in barcharts, for each of the criteria. The corresponding workflow is shown on Figure 1.7.

Thierry analyzes the resulting plots. They are shown on Figure 1.8 for the 3 criteria to be minimized, and on Figure 1.9 for the 2 criteria to be maximized. The alternatives on the abscissa are ordered from left to right according to the preferential direction.

Thierry observes, among other things, that alternative a11 (P309-16) seems to be a good alternative, as it has good values on a lot of criteria (except g_1 (price)). He seems to be very interested by this alternative, and says that the rather bad performance on the price criterion could be compensated by the good performances on the other criteria.

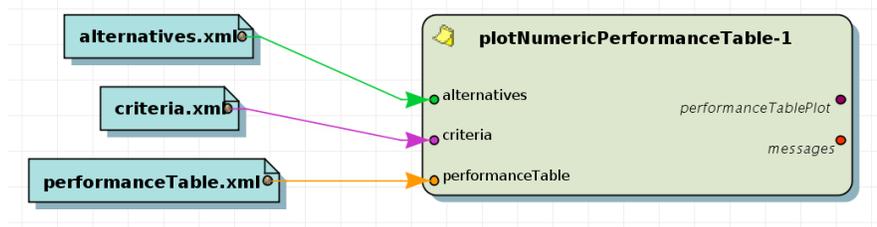


Fig. 1.7 The workflow representing the input data (left) and the module to plot the performances of the alternatives on the criteria (right).

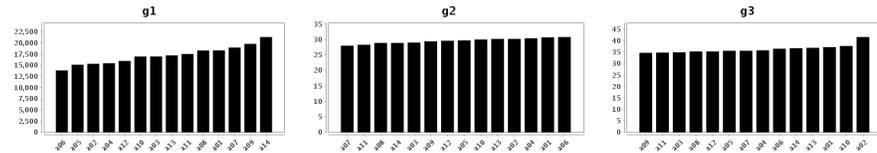


Fig. 1.8 Bar plots of the performances, for the 3 criteria to be minimized (cost, acceleration and pick up).



Fig. 1.9 Bar plots of the performances, for the 2 criteria to be maximized (brakes and road-hold).

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 aid 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 the values $u(a)$, $u(b)$ of the alternatives a and b in the following way:

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

The value $u(a)$ depends naturally on the evaluations $\{g_i(a), i = 1, \dots, n\}$ of alternative a (where n is the number of criteria), and we choose to construct a quite simple 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 some trade-offs between the criteria).

1.3.2 Preference elicitation

Now that a motivated choice has been made on the preference model, the next step of this decision aid 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 order to identify his preferences in the chosen aggregation model, we construct a new workflow in `diviz`, by using the calculation module called UTA. The workflow is represented in [Figure 1.10](#). Next to the definitions of the alternatives and the criteria, and the performance table, it uses as input the ranking provided by Thierry, as well as the number of segments for each marginal value function which has to be determined. For arguments of parsimony, we decide to search for piecewise linear value functions with 2 segments. Thierry’s input ranking is completed by two fictive alternatives (also sometimes called “ideal” and “anti-ideal” points), which simply are the best possible (ranked first) and the worst possible (ranked last) alternatives, given the ranges of values taken by the alternatives on the criteria.

Thierry’s ranking is compatible with the chosen model, and we plot the obtained value functions, by completing the previous workflow with the `plotValueFunctions` module (see [Figure 1.11](#)).

The resulting marginal value functions are shown on [Figure 1.12](#). Only those for criteria `g1`, `g2` and `g5` are represented, as the ones for `g3` and `g4` do not intervene in the aggregation (the marginal value equals 0 for any evaluation on `g3` and `g4`). The maximal value on the ordinate axis represents the trade-off weight in the aggregation.

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 pick up and brakes criteria should also be considered to discriminate between alternatives. He decides to modify his a priori ranking by adding two alternatives :

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

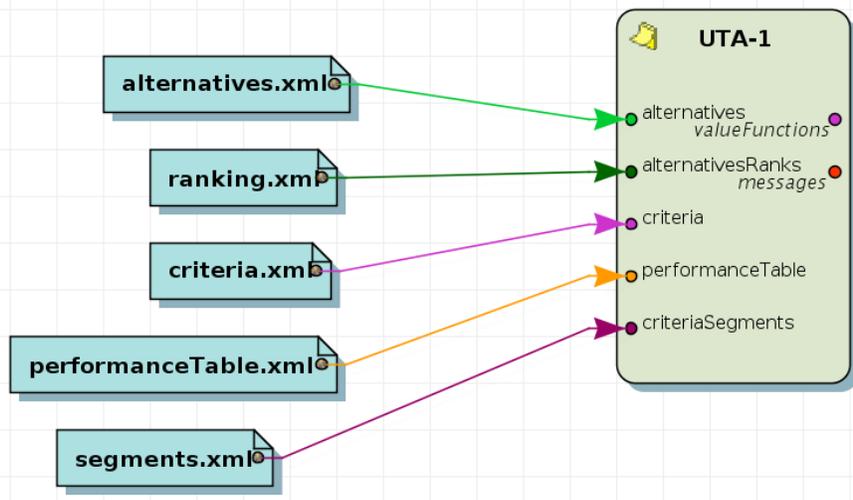


Fig. 1.10 The workflow representing the input data (left) and the UTA module to determine the piecewise linear value functions compliant with Thierry’s input preferences (right).

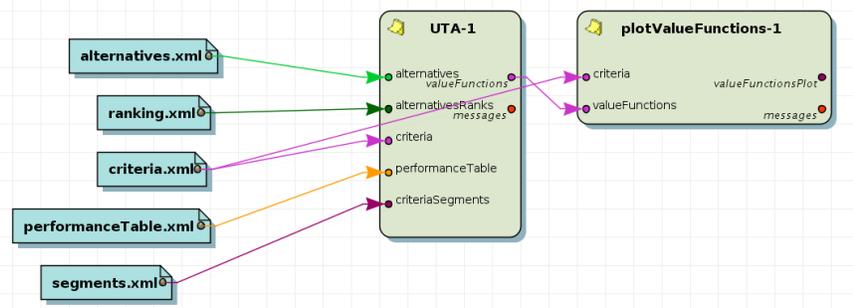


Fig. 1.11 The workflow representing the input data (left), the UTA module (center), and the plot module (right).

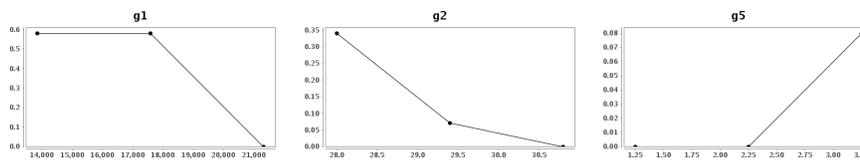


Fig. 1.12 Marginal value functions for criteria g1 (price), g2 (acceleration) and g5 (roadhold).

The new calculations generate the value functions represented on Figure 1.13. This time Thierry validates the model, as at least the pick up criterion plays a significant role in the aggregation.

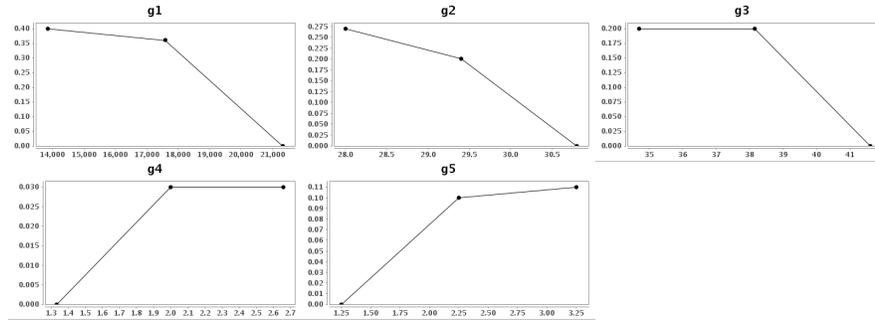


Fig. 1.13 The marginal value functions after the update of the a priori ranking of Thierry.

1.3.3 The aggregation phase

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 diviz, we therefore add a the module `computeNormalizePerformanceTable`, which takes as input the marginal value functions, the performance table, and the descriptions of the criteria and the alternatives. We then combine the result of this module with the `generalWeightedSum` one, to get the overall score of each car. This workflow is shown on Figure 1.14.

The output of the `generalWeightedSum` module is the “super-scale” we were mentioning earlier. It indicates the value of each car, according to the preference model (and consequently, if it can be considered as accurate, according to Thierry). These overall scores are represented on Figure 1.15.

We can observe that the car which obtains the highest score is a11 (P309-16) (after the fictive “ideal” car). This confirms Thierry’s preliminary analysis.

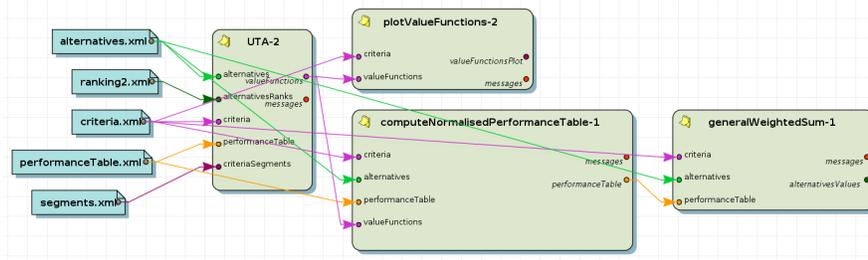


Fig. 1.14 The workflow representing the input data (left), the UTA module, the plot module and the normalization module (center), as well as the summation module (right).

	0.95	0.92	0.8	0.88	0.82	0.8	0.81	0.68	0.64	0.68	0.66	0.6	0.63	0.56	0	1.0
a11															antIdeal	ideal
a03																
a07																
a12																
a10																
a08																
a05																
a04																
a01																
a06																
a13																
a02																
a09																
a14																

Fig. 1.15 The overall scores of the cars.

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

1.3.4 Analysis of the results

A last step of the decision aid process could be 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.

We mainly confront Thierry to the ranking of the cars according to their overall scores, and compare the output obtained by his preference model to the one calculated through a weighted sum (with more or less arbitrary weights and normalized data). This latter model is extensively discussed in [Bouyssou et al. \[2000\]](#) on pages 103 to 109.

In a first step, we complete the workflow of Figure 1.14 by two elements: first a module to obtain the ranks of the alternatives according to their overall scores, and second a module to represent this ranking graphically. Figure 1.16 shows the first 7 positions of this ranking (plus the “ideal” fictive car).

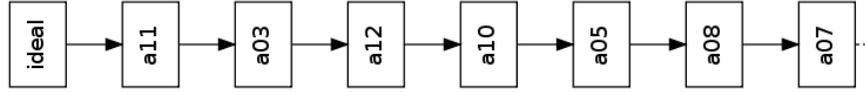


Fig. 1.16 The ranking obtained by the additive value model.

In a second step, Thierry wishes to see what would have happened if, instead of this preference elicitation phase, he had used a more simpler weighted sum model. 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 1.2. The first three criteria receive negative weights since they have to be minimized.

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

Table 1.2 Thierry’s naive weights for the weighted sum model.

The workflow corresponding to this aggregation in diviz is presented in Figure 1.17. We first use the performanceTableTransformation module to normalize the data according to the method described above. Then we use the weightedSum module to calculate the weighted sum of each alternative, before deriving their ranks via rankAlternativesValues.

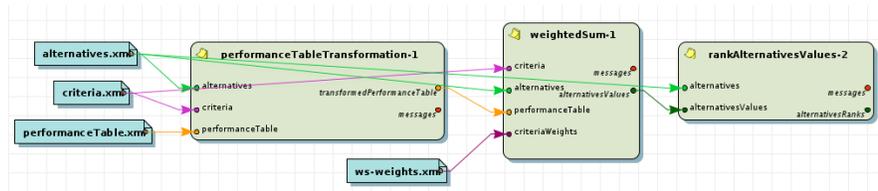


Fig. 1.17 The workflow representing the input data (left), data normalization module and the weighted sum module (center), and a module to obtain the ranks of the alternatives (right).

According to this model, car a03 is ranked first, before car a11 and a12. This is quite similar compared to the model obtained by eliciting Thierry's preferences in an additive value model (compare to Figure 1.16). This similarity (but not equality) is also confirmed by a quite high Kendall's rank correlation index (0.73) between the two rankings, obtained by adding the `alternativesValuesKendall` module in the end of the workflow, as shown on Figure 1.18.

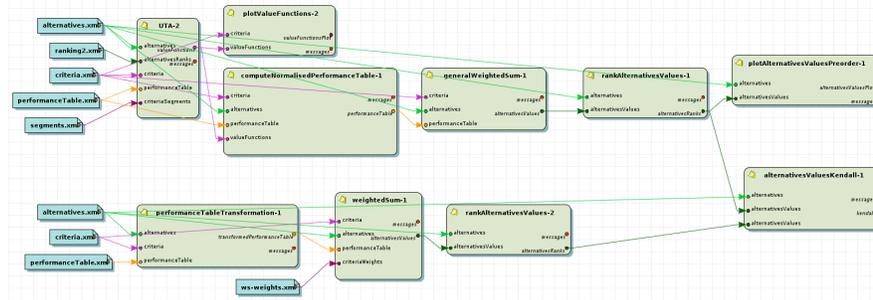


Fig. 1.18 The workflow comparing the two models : on top the additive value model, and in the bottom the weighted sum model. In the end, both rankings are compared by calculating their Kendall's rank correlation index (on the right).

Thierry however has a higher confidence in the output of the additive value function model, as it confirms his initial feeling about a11, and the determination of the parameters of the model seems less arbitrary to him.

1.4 Concluding remarks

In this chapter we have shown the flexibility and the main advantages of the diviz workbench. We have highlighted how it can be used to design and execute algorithmic MCDA workflows and to disseminate research results. The diviz software is being constantly improved, and the number of available components is quickly growing.

The example detailed in Section 1.3 underlines the big potential of the software for the analyst, and shows how diviz can be adapted to various practical decision aid situation. Next to that, diviz is also an easy to use pedagogical tool for teachers who need to present and compare classical MCDA methods.

All in all, diviz gives rise to an innovative work methodology in MCDA, which no longer considers the methods as static and immutable black boxes, but rather as dynamic workflows which can be changed and adapted for the current purpose.

Currently diviz is already used in a lot of universities and engineering schools through Europe in MCDA courses. It has clearly proven its great potential as a pedagogical tool via its large adoption by the students.

References

- R. Bisdorff, P. Meyer, and T. Veneziano. Inverse analysis from a condorcet robustness denotation of valued outranking relations. In F. Rossi and A. Tsoukiás, editors, *Algorithmic Decision Theory*, LNAI, pages 180–191. Springer-Verlag Berlin Heidelberg, 2009.
- D. Bouyssou, T. Marchant, M. Pirlot, P. Perny, A. Tsoukias, and P. Vincke. *Evaluation and decision models, A critical Perspective*. Kluwer’s International Series. Kluwer, Massachusetts, 2000.
- J.-P. Brans and P. Vincke. A preference ranking organization method. *Management Science*, 31(6):647–656, 1985.
- Decision Deck Consortium. Strategic manifesto of the Decision Deck project, 2009. <http://www.decision-deck.org/manifesto.html>.
- M. Grabisch, I. Kojadinovic, and P. Meyer. A review of capacity identification methods for Choquet integral based multi-attribute utility theory; Applications of the Kappalab R package. *European Journal of Operational Research*, 186(2):766–785, 2008. doi: 10.1016/j.ejor.2007.02.025.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.
- E. Jacquet-Lagrèze and Y. Siskos. Assessing a set of additive utility functions for multicriteria decision making: the UTA method. *European Journal of Operational Research*, 10:151–164, 1982.
- R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL <http://www.R-project.org>. ISBN 3-900051-00-3.
- B. Roy. Classement et choix en présence de points de vue multiples (la méthode ELECTRE). *Revue française d’informatique et de recherche opérationnelle (RIRO)*, 2:57–75, 1968.