



# Diviz: an MCDA workflow design, execution and sharing tool

Sébastien Bigaret, Patrick Meyer

## ► To cite this version:

Sébastien Bigaret, Patrick Meyer. Diviz: an MCDA workflow design, execution and sharing tool. Newsletter of the EURO Working Group Multicriteria Aid for Decisions, 2010, pp.10-13. hal-00926569

**HAL Id: hal-00926569**

**<https://hal.science/hal-00926569>**

Submitted on 13 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## **diviz: an MCDA workflow design, execution and sharing tool**

**Sébastien Bigaret, Patrick Meyer**  
**Institut Télécom, Télécom Bretagne**  
**UMR CNRS 3192 Lab-STICC**  
**Technopôle Brest-Iroise CS 83818**  
**F-29238 Brest Cedex 3**

**Université européenne de Bretagne**

`{sebastien.bigaret,patrick.meyer}@telecom-bretagne.eu`

**ABSTRACT.** The diviz software is a tool for designing, executing and sharing Multiple Criteria Decision Aid (MCDA) techniques. The target user community of diviz reaches from researchers who require to construct algorithmic workflows from elementary MCDA components to teachers who need to present MCDA methods and let their students experiment them. Besides, one of the principal features of diviz is to ease the comparison of results issued from different methods on the same decision problem. This allows among others to check the robustness of the output of a decision aid process with respect to the choice of the decision aid technique.

**KEYWORDS.** algorithmic MCDA workflows, open-source software, method comparison.

### **1. INTRODUCTION**

Research activities in and around the field of Multiple Criteria Decision Aid have developed quite rapidly over the past years, and they have resulted in various streams of thought and methodological formulations for the resolution of complex decision problems. In particular, many so-called MCDA *methods* have been proposed in the literature and are very often available as software programs.

Unfortunately, at least four major difficulties arise when it comes to using these procedures in practice:

1. Different techniques are generally implemented in separate software products, with heterogeneous user interfaces;
2. Testing multiple MCDA algorithms on one problem instance is not easy, because of the various input data formats required by the software applications;
3. A lot of MCDA algorithms which are presented and published in scientific articles are not easily available and consequently often only used by their authors;
4. Several MCDA software products are 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 dataset inside a common framework. Among the most famous ones, one can cite platforms such as the GNU R statistical system (R Development Core Team, 2009) or the Weka suite of machine learning software (Hall et al., 2009). Both of these suites are open-source and OS independent, which has certainly contributed to their large dissemination and acceptance among many researchers and users.

The diviz software is one of the initiatives of the Decision Deck<sup>1</sup> project and is developed at the LUSI department of Télécom Bretagne in France. Its essential purpose is to overcome the main difficulties of the current software situation in MCDA research mentioned above. diviz is an open-source software which allows to design complex workflows<sup>2</sup> of MCDA algorithms in a very intuitive way and to execute them. One of the key characteristics of these executions is that they are performed via web-services on distant servers.

---

1 <http://www.decision-deck.org>

2 By algorithmic workflow we mean a sequence of connected algorithmic steps.

In this article, we present diviz, its properties and its latest developments in Section 2. Then, in Section 3 we detail its use on a well-known example from the literature. Finally, we conclude in Section 4 and present the perspectives of diviz.

## 2. DESCRIPTION AND PROPERTIES OF DIVIZ

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 (Roy, 1968), the UTA method (Jacquet-Lagrèze and Siskos, 1982), etc). diviz enables to conveniently combine programs implementing MCDA algorithms, which can originate from various methodological schools, researchers and programmers.

In this section we present the main features of diviz and show how it allows to rebuild classical MCDA *methods*, to develop new ones by combining various elementary calculation components<sup>3</sup>, and how it can be used as a research and dissemination tool.

### *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 connectors. Thus, the design of complex algorithmic workflows does not require any programming skills, but only necessitates to understand the functioning of each calculation module.

The inputs and outputs of these elementary components can be manifold and are *typed*. To illustrate this, consider the following example.

**Example:** *diviz allows to use a component called `weightedSum`. This element calculates the weighted sum of alternatives' profiles with respect to a set of weights associated with a list of criteria. Consequently, `weightedSum` requires four inputs: the description of the criteria (at least their ids), the description of the alternatives (at least their ids), the performance table containing the numerical evaluation of each alternative on each of the criteria and the set of numerical weights associated with the criteria. The main output of this component is a list of alternatives' ids associated with the weighted sum of their evaluations. Note that we present later further details on the data typing.*

To construct a new MCDA workflow, the user chooses one or more modules from a list of available calculation elements which he can drag and drop in a dedicated workspace. 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 by connectors to define the structure of the workflow.

### *Execution and results*

Once the design of the MCDA workflow is finished, it is possible to execute it in order to obtain the possibly multiple outputs of the algorithms. In diviz, these calculations are performed on computing servers through the use of web-services<sup>4</sup> published by the Decision Deck project. The idea behind these web-services is to allow anyone who is connected to the Internet to access a large amount of MCDA algorithms without having to install them on their personal computer. As a consequence, diviz does not contain any algorithmic modules, but requires a connection to the Internet to access the calculation resources.

If the execution of the workflow is successful, the outputs of each of the components can be viewed and analysed by the user. Some of these outputs might represent results of intermediate calculation steps of the decision aid workflow which has been built by the user. This allows to help to tune the parameters of the algorithms as shown 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*

---

<sup>3</sup> Note that in the sequel we will unthinkingly use the words *component*, *program* or *module* to describe the algorithms which can be combined in diviz.

<sup>4</sup> <http://www.decision-deck.org/ws>

*performances for each of the alternatives; a fourth module draws a ranking of the alternatives on basis of the overall value 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 is facilitated.*

Among other things, this enables to check and validate the robustness of the output of a decision aid process with respect to the choice of the decision aid technique and the input parameters.

Furthermore, the history of the past executions is kept in diviz and can be viewed by the user. In particular, if a workflow is modified, the former executions' results and their associated workflows are still available. Again this is a key feature which helps calibrating the parameters of the algorithms which are used in a workflow, as any past execution can be recalled and reanalysed.

#### *Available algorithmic components*

As already mentioned earlier, the algorithmic elements are available in diviz via web-services. At the time of writing, there are 32 components present which are divided into 4 main categories:

1. *calculation components* containing aggregation operators, disaggregation techniques, analysis elements, 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.

Currently diviz contains, among many others, functions from the kappalab library (Grabisch et al., 2009) which can be used for Choquet integral-based MCDA, disaggregation techniques based on outranking relations (Bisdorff et al., 2009) and additive value functions (ACUTA, Bous et al., 2010), as well as the SMAA method (Lahdelma et al., 1998). A detailed list of all available components can be found on the website of diviz<sup>5</sup>.

Note that at the time of writing, various researchers of the Decision Deck project are working on the implementation of MCDA techniques in view of publishing them as web-services which can then be accessed from diviz. In particular, *classical* methods like the ELECTRE (Roy, 1968) and PROMETHEE (Brans and Vincke, 1985) series will be available soon. This should lead to a high interest in the diviz software and the web-services offered by Decision Deck.

#### *XMCD: the key to interactions and data visualization*

One of the key points of diviz is the possibility to combine various MCDA algorithms originating from heterogeneous researchers in complex workflows. To facilitate this interoperability, the Decision Deck project suggests to use a data standard called XMCD<sup>6</sup> to store the inputs and outputs of MCDA programs. The XMCD markup language is written in XML, a general-purpose syntax for defining markup languages.

It follows that the web-services proposed by Decision Deck accept data written according to XMCD and generate data in this same format.

In diviz this has three direct consequences:

- first, all components available in diviz can interoperate. Typically, the output of one algorithm can be injected into other elementary modules without requiring data transformations;
- second, the inputs and the outputs of the elementary components in diviz are typed with respect to the different data types defined in XMCD. This allows a very precise control of the various algorithms, and facilitates the creation of complex combinations of components;

---

<sup>5</sup> <http://www.decision-deck.org/diviz>

<sup>6</sup> <http://www.decision-deck.org/xmcd>

- third, we exploit a powerful feature of XML based markup languages to convert XMCDAs documents into HTML pages for a convenient visualisation of their contents in a web browser integrated in diviz.

#### *“Methods” comparison*

Next to designing and executing MCDA workflows, diviz can also be a convenient tool to compare the outputs of various methods on the same input data. Up to now, 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 set of data to various workflows in a single workspace, each representing a different MCDA method, and to compare their outputs. Again, this is an interesting feature to check the robustness of the output of an analysis with respect to the choice of the decision aid technique.

#### *Workflow sharing*

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

Consequently, diviz can be used as a dissemination tool in combination with a research article. Indeed, the authors of a new MCDA technique could propose the corresponding diviz workflow together with an appropriate data set as supplementary electronic material with their article. This would certainly contribute to a larger dissemination of new MCDA algorithms and facilitate their acceptance among many researchers.

#### *A quick look at the architecture*

Technically speaking, diviz is a classical 3-tier application made of: the client which has been described in this section, a component accessing the Decision Deck web-services, and a server. The diviz server's main task consists in planning and controlling the execution of the submitted workflows, making it possible for different components in the workflow to be executed in parallel, when appropriate. diviz users download the client only; the latter connects to the server which takes care of distributing the computations to the dedicated web-services; it gathers all intermediary and final results which are ultimately sent back to the user.

### **3. APPLICATION**

In this section, we illustrate the use of diviz on a classical MCDA problem which has been widely discussed in the literature, namely the choice of a sports car (Bouyssou et al., 2000, chapter 6). Let us briefly recall the main features of this example and the underlying data.

In 1993, 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 viewpoints related to cost (criterion 1), performance of the engine (criteria 2 and 3) and safety (criteria 4 and 5). The list of alternatives and their evaluations on these five criteria is presented in Table 1. The cost criterion (€) and the performance criteria acceleration (seconds) and pick up (seconds) have to be minimised, whereas the safety criteria brakes and road-hold have to be maximised. 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, chapter 6).

car ID	car name	cost	acceleration	pick up	brakes	road-h
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

<b>a08</b>	<b>Astra</b>	18319	28.9	35.3	1.66	2
<b>a09</b>	<b>Escort</b>	19800	29.4	34.7	2	1.75
<b>a10</b>	<b>R19</b>	16966	30	37.7	2.33	3.25
<b>a11</b>	<b>P309-16</b>	17537	28.3	34.8	2.33	2.75
<b>a12</b>	<b>P309</b>	15980	29.6	35.3	2.33	2.75
<b>a13</b>	<b>Galant</b>	17219	30.2	36.9	1.66	1.25
<b>a14</b>	<b>R21t</b>	21334	28.9	36.7	2	2.25

Table 1: Data for Thierry's car selection problem

As done in (Bouyssou et al., 2006, chapter 7), we suppose that Thierry has already some knowledge about the 14 cars, and that he is able to express the following ranking on a few of them:

P309-16 > Sunny > Galant > Escort > R21t.

To present the use of diviz, we detail the construction of a variant of UTA (Bous et al., 2010) named ACUTA to determine a preference model that will allow us to evaluate the 14 cars and to rank them. The screenshot of the corresponding workflow is given in the upper part of Figure 1 below (area 1).

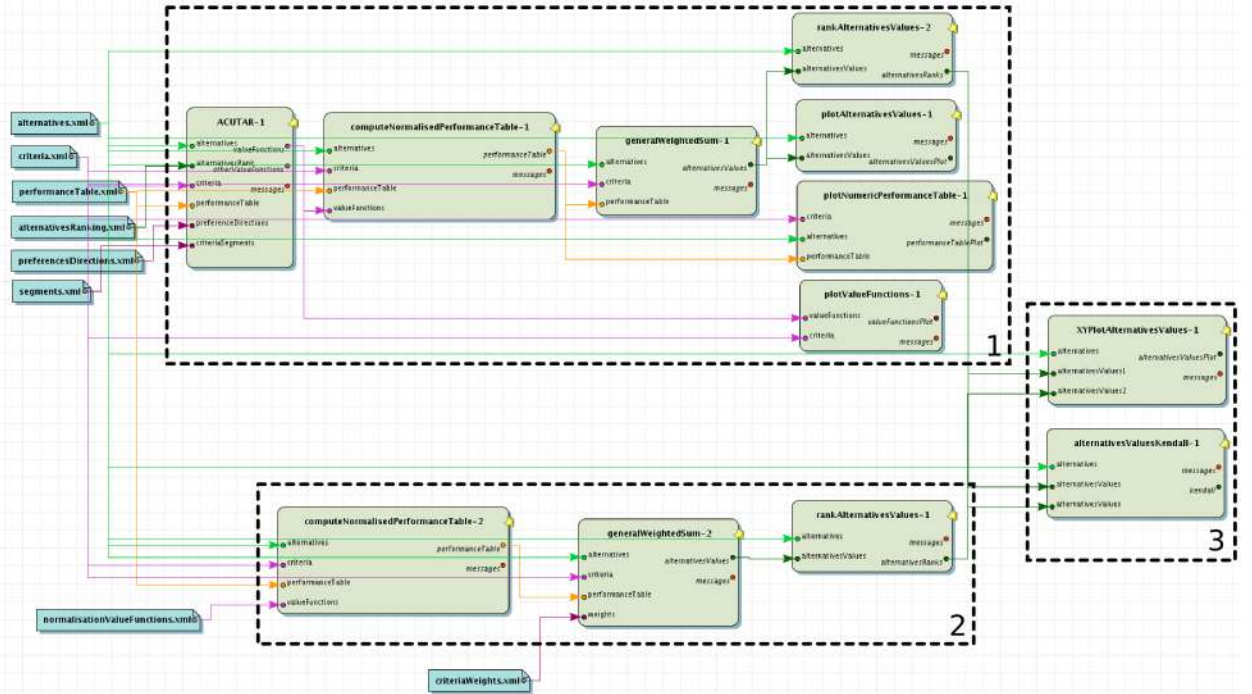


Figure 1: The initial workflow for the car selection problem

The workflow contains 7 elementary components, including the one implementing the ACUTA method. For short, ACUTA determines piecewise linear partial value functions on basis of an input ranking of a subset of alternatives, compatible with an additive value model. Consequently, the ACUTA module requires as inputs the description of the alternatives and the criteria, a performance table evaluating each alternative on the criteria, a ranking of a subset of alternatives, the preference directions on the values of the criteria and the number of segments of each piecewise linear value function. One of the outputs of this component is a description of a set of compatible value functions with the input ranking.

These value functions are then used by computeNormalisedPerformanceTable to transform the original performance table into a *normalized* version, where all the evaluations of the alternatives are in the real unit interval.

The generalWeightedSum module is used to calculate the aggregated value of each of the alternatives of the performance table by a simple sum. This output is then plotted in a bar chart via the plotAlternativesValues component.



Finally, the component `plotNumericPerformanceTable` is used to graphically represent the normalized version of the performance table, `plotValueFunctions` is used to plot the value functions outputted by the ACUTAR module, and the `rankAlternativesValues` module is used to obtain the ranks of the alternatives according to their overall values.

On Figure 2, we show the output of `plotAlternativesValues`, where we can observe that alternative a11(P309-16) obtains the highest overall value.

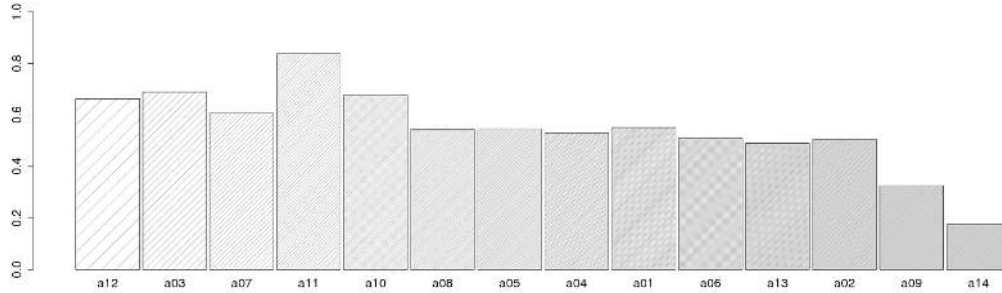


Figure 2: Bar chart representing the overall values of the alternatives

Besides, on the output of `plotNumericPerformanceTable` on Figure 3, one can easily understand why a11 scores so well. Each bar represents an alternative, whose evaluations for the 5 criteria are stacked up. The partial values of a11 for the various criteria are generally very good, which produces by summation a high overall value.

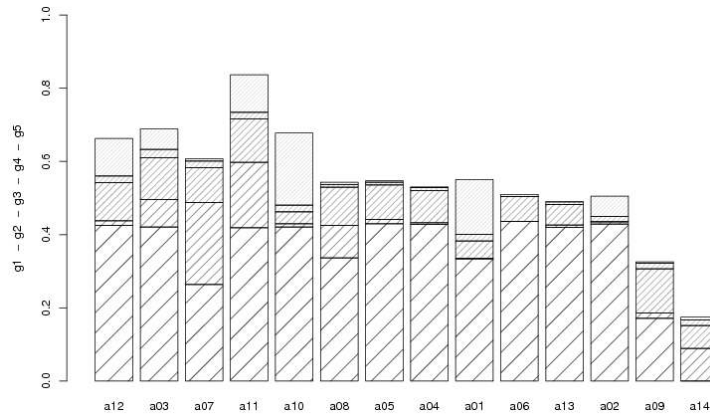


Figure 3: Graphical representation of the normalized performance table.

Finally, on Figure 4, one can observe the preference model (via the partial value functions) of Thierry based on the ranking of the few alternatives. Clearly, the cost criterion (g1) has the biggest influence in his choice and Thierry has a preference for cars which cost less than 18000€.

These outputs (as well as many other results) can easily be viewed in diviz in an associated viewer window.

Let us now exploit the flexibility of diviz in order to compare these results with the output of a much simpler MCDA technique, namely the weighted sum. This workflow is drawn in the lower part of Figure 1 (area 2).

The 3 components `computeNormalisedPerformanceTable`, `generalWeightedSum` and `rankAlternativesValues` are used for this aggregation. In order to normalise the data of the performance table, we divide the performances on each of the criteria by the maximal values they take. This is equivalent to using appropriate linear value functions, which can be done via `computeNormalisedPerformanceTable`.

The set of weights used by the weighted sum is given hereafter (the negative values are used for criteria

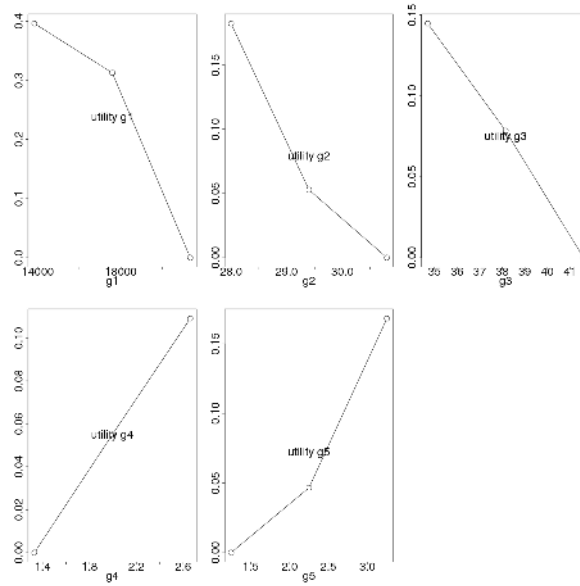


Figure 4: Plots of the partial value functions

which are to be minimised):

$$\{-0.4, -0.18, -0.14, 0.11, 0.17\}.$$

Note that the absolute values of these weights correspond to the maximal values taken by the partial value functions (see Figure 4). The ranks of the alternatives obtained by this technique are then compared to those obtained by the earlier described workflow based on the ACUTA disaggregation.

This comparison is performed in the right part of Figure 1 (area 3), where Kendall's tau (rank correlation coefficient) between both rankings is computed via the `alternativesValuesKendall` module and a graphical representation of both rankings is plotted via the `XYPlotAlternativesValues` component (see Figure 5).

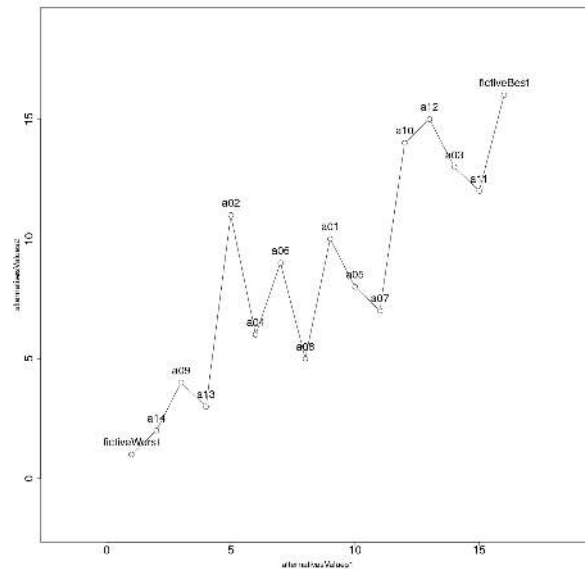


Figure 5: XY plot representing both rankings (abscissa: ranks by ACUTA, ordinate: ranks by the weighted sum)

Kendall's tau coefficient equals 0.68, which shows that there are a lot of inversions between both rankings.



On Figure 5<sup>7</sup> we can see that Thierry's initial preferences ( $a_{11} > a_{03} > a_{13} > a_{09} > a_{14}$ ) are satisfied via the ACUTA model, whereas this is not the case with the weighted sum model ( $a_{03} > a_{11} > a_{09} > a_{13} > a_{14}$ ).

#### 4. Conclusion and perspectives

In this short article we have presented diviz which 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 very simple example detailed in Section 3 shows the big potential of the software for the various targeted user communities. Researchers can easily share and test MCDA workflows or experiments, explore robustness issues linked to the choice of the algorithm or the values of the input parameters and spread their own algorithms very conveniently. diviz is also an easy to use pedagogical tool for teachers who need to present and compare classical MCDA methods. Last but not least, diviz may also be used by consultants who wish to solve real world decision problems with a given method.

To conclude this article, we would like to make a call for participation: your MCDA algorithm is welcome in diviz, and if you would like to make it available as a web-service via diviz, please have a look at the developers corner on the website of Decision Deck's web-services<sup>8</sup> and contact the authors of this article for any help request or question.

#### REFERENCES

- Bisdorff R, Meyer P, Veneziano T, *Inverse analysis from a Condorcet robustness denotation of valued outranking relations*. In F. Rossi and A. Tsoukiàs (Eds.), *Algorithmic Decision Theory*. Springer-Verlag Berlin Heidelberg, LNAI 5783, pp. 180-191, 2009.
- Bous G, Fortemps P, Glineur F, Pirlot M, ACUTA: A novel method for eliciting additive value functions on the basis of holistic preference statements, *European Journal of Operational Research*, submitted, 2010.
- Bouyssou D, Marchant T, Pirlot M, Perny P, Tsoukiàs A, Vincke P. *Evaluation and decision models: A critical perspective*, Kluwer, 2000.
- Bouyssou D, Marchant T, Pirlot M, Tsoukiàs A, Vincke P. *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*, International Series in Operations Research and Management Science Volume 86, Springer, 2006.
- Brans J-P, Vincke P. *A preference ranking organization method*. *Management Science*, 31(6):647-656, 1985.
- Grabisch M, Kojadinovic I, Meyer P, kappalab: Non additive measure and integral manipulation functions, R package version 0.4-4, 2009.
- Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. *The WEKA Data Mining Software: An Update, SIGKDD Explorations*, Volume 11, Issue 1, 2009.
- Jacquet-Lagrèze E, Siskos Y, *Assessing a set of additive utility functions for multicriteria decision making: The UTA method*, *European Journal of Operational Research*, 10(2): 151–164, 1982.
- Lahdelma R, Hokkanen J, Salminen P, *SMAA - Stochastic multiobjective acceptability analysis*, *European Journal of Operational Research*, 106 (1): 137-143, 1998.
- R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, <http://www.R-project.org>, 2009.
- Roy B, *Classement et choix en présence de points de vue multiples (la méthode ELECTRE)*, *RIRO*, 8:57–75, 1968.

---

<sup>7</sup> On Figure 5, the alternatives called *fictiveBest* (resp. *fictiveWorst*) are fictive alternatives having the best (resp. worst) performances on all the criteria. They are used by the ACUTA algorithm to correctly scale the domains of the partial value functions.

<sup>8</sup> <http://www.decision-deck.org/ws/howtos.html>