Practical work

Use of diviz to help Thierry to choose a car

Miłosz Kadziński* Patrick Meyer†

Introduction

The goal of this practical work is to learn how to use the diviz software[1] and to explore the various possibilities which diviz opens on MCDA.

The diviz tool facilitates the creation of complex algorithmic MCDA workflows via a graphical user interface which allows you to drag and drop the calculation elements in a workspace and to connect them in a coherent manner. After the execution of the workflow, you can easily study the final and intermediate results, or adapt the input parameters of the algorithms, in order to obtain a valid final recommendation.

Further details on the example (data set) which is used in this document can be found in [BMP+00].

The decision problem

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 three viewpoints 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 the five criteria are presented in Table 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.

In order to help Thierry to choose the car which is the most appropriate for him, we propose to rank these cars according to his preferences. To do so, we will use 4 techniques: an additive value aggregation from MAVT (steps and materials marked with V) and preference disaggregation UTA method (ACUTA; marked with U), Electre III (marked with E) and PROMETHEE (marked with P) from the outranking models. In case you wish to sort the cars to the pre-defined and ordered classes, you can find some hints in section A. All materials needed for conducting the practical exercises are available at http://www.cs.put.poznan.pl/mkadzinski/MCDASummerSchool/02-practicalWork/.

*Poznan University of Technology, Poland, milosz.kadzinski@cs.put.poznan.pl
†IMT Atlantique, France, patrick.meyer@imt-atlantique.fr
[1] http://www.diviz.org

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: Data for Thierry's car selection problem

Step V.1 : The performance table

The performance table of Table 1 can be found in a "comma separated value" (csv) format in the file `performanceTable.csv` on the website of this lecture[2]. Files in this format can be edited, e.g., in Microsoft Excel or Notepad.

Download this file and store it on your computer. In the first step, this file has to be converted to the `XMCDA` format, to make it compatible with the various web-services which can be called via diviz.

diviz exercise 1. *If it is not already done, install diviz. For conducting the exercises, use the version available at* *`http://www.cs.put.poznan.pl/mkadzinski/MCDASummerSchool/02-practicalWork/`*. *Download the jar file, which allows you to run diviz immediately.Later on you can install the software from* *`http://www.diviz.org/download`* *so that to receive its updates in an automatic way (for the users of Mac OS it is recommended to use the jar file).*

Then start diviz, create a new workflow (`Workflow - New`*), and drag and drop the module* `csvToXMCDAperformanceTable` *from the right pane of diviz into the empty workspace. Add the file containing the performance table to diviz by dragging and dropping the element* `File(s)` *from the right pane into the workspace and by selecting the csv file you just downloaded on your hard drive. Connect the csv file to the* `csvToXMCDA-performanceTable` *module, and execute the workflow.*

The `csvToXMCDA-performanceTable` module has generated three `XMCDA` outputs: one containing a description of the alternatives (ids), another with the description of the criteria (ids), and the last one with the performances of the alternatives on the criteria.

At this stage, it could be interesting to show a summary of the data contained in the performance table to help Thierry in the preference elicitations steps. The module `criteriaDescriptiveStatistics` calculates some basic statistical indicators from the data contained in the performance table, whereas `plotStarGraphPerformanceTable` generates graphical representations of these data.

[2] http://www.cs.put.poznan.pl/mkadzinski/MCDASummerSchool/02-practicalWork/data/

diviz exercise 2. *Use* `criteriaDescriptiveStatistics` *and* `plotStarGraphPerformanceTable` *to show a summary of the performance table to Thierry* [3]*. Add these modules to your workspace, connect them correctly to the outputs of* `csvToXMCDA-performanceTable`*, and execute the workflow.*

Thierry should become aware of the various values ranges of the criteria, as well as the distribution of the alternatives on the scales of the criteria.

Step V.2 : Elicitation of Thierry's value functions

MCDA exercise 1. *Question Thierry to elicit his preferences on the 5 criteria via value functions. For this purpose, you can use, e.g., a bisection method presented during the lecture.*

diviz exercise 3. *Store these value functions in a csv file according to the following format:*

g1	*Cost*	*25000*	*0*
		19000	*0.5*
		10000	*1*
g2	*Acceleration*	*32*	*0*
		29	*0.5*
		26	*1*
...	...	...	...

In this example, for g1 *Thierry considers that a price of 25000 is not acceptable, a price of 10000 satisfies him fully, and a price of 19000 satisfies him at 50%. Intermediate values are determined by linear interpolation.*
Use `valueFunctions.csv` *as a starting point for your input.*

This csv file needs to be transformed into an `XMCDA` file in order to be usable by the MCDA algorithms available via diviz.

diviz exercise 4. *Drag and drop the module* `csvToXMCDA-valueFunctions` *from the right pane of diviz into your workspace. Add the file containing the value functions to diviz. Connect the csv file to the* `csvToXMCDA-valueFunctions` *module, and execute the workflow.*

The output of this first workflow is an `XMCDA` file which contains the definition of the criteria (id, name) together with the value functions representing Thierry's preferences on the five criteria. It can be viewed in the bottom pane of diviz by clicking on the output of the `csvToXMCDA-valueFunctions` module.

These value functions should be validated by Thierry. To do so, confront him with their graphical representation.

diviz exercise 5. *The module* `plotValueFunctions` *plots a graphical representation of value functions. Drag and drop this module into your workspace and connect it to the output of* `csvToXMCDA-valueFunctions`*. Double click on* `plotValueFunctions` *in your workspace to tune some visual parameters (colors, etc.) before executing your workflow.*

[3]Whenever you wish to extend the workflow, you need to come back to the edit mode from the mode of viewing the results. This can be done by clicking the name of the developed workflow in the menu to the left.

Step V.3 : Transforming the original performances via Thierry's preferences

The value functions representing Thierry's preferences need to be applied on the original performances of the alternatives of Table 1.

diviz exercise 6. *Add the module* `computeNormalisedPerformanceTable` *to your workspace and connect it correctly to the existing workflow in order to determine the value of each alternative on each criterion, according to Thierry's preferences.*

Step V.4 : Eliciting Thierry's criteria weights

MCDA exercise 2. *Elicit Thierry's criteria weights. If you wish to use the ROC method, you can enter the ranking of criteria in ROC-weights.xls files and derive the weights from this sheet.*

diviz exercise 7. *Store the elicited weights in a csv file according to the following format:*

	g1	*g2*	*g3*	*g4*	*g5*
weights	*0.457*	*0.257*	*0.040*	*0.090*	*0.157*

Use `weights.csv` *as a starting point for your input. Use the* `csvToXMCDAcriteriaValues` *module to create an* `XMCDA` *output containing these criteria weights.*

Validate these weights with Thierry by explaining him that the greater the criterion's weight, the greater will be its share in the alternatives' comprehensive values.

diviz exercise 8. *Use modules like* `plotCriteriaValues` *or* `plotCriteriaValuesPreorder` *to help you validate these weights with Thierry. A double click on these modules allows you to tune the graphical representations of the weights according to your needs.*

Step V.5 : The aggregation procedure

In order to obtain an overall score (comprehensive value) for each car, we use the weighted sum to aggregate the partial (marginal) values of the cars on different criteria.

diviz exercise 9. *Use the module* `generalWeightedSum` *to determine the overall values of the alternatives with respect to the value functions and the weights given by Thierry.*

The car with the highest overall value will be recommended to Thierry. However, Thierry is also interested in the ranking of all cars.

diviz exercise 10. *Determine the ranking of the cars via the* `rankAlternativesValues` *module. Double click on this module to specify that the highest value has to be ranked first. Combine* `rankAlternativesValues` *with* `plotAlternativesPreorder` *to show Thierry the preorder of the cars.*

Having determined a recommendation for Thierry on the basis of an MAVT technique, we will now try with UTA.

Step U.1 : Elicitation of Thierry's preferences for the criteria

diviz exercise 11. *Construct a new workflow (Workflow - New). Add* `XMCDA` *files with specification of alternatives, criteria, their preference directions, and performance table. The set of alternatives is enriched with two fictive ones: the ideal alternative with the best performances on all criteria and the anti-ideal alternative with the worst performances on all criteria.*

MCDA exercise 3. *Elicit from Thierry the number of linear pieces (segments) that should be used for the marginal value function for each criterion. Explain that a single piece is equivalent to using a linear value function and that the greater the number of pieces, the greater flexibility of an additive value function in representing non-linear preferences.*

diviz exercise 12. *Store the numbers of linear pieces in a csv file according to the following format:*

	g1	*g2*	*g3*	*g4*	*g5*
numberOfSegments	*2*	*2*	*2*	*3*	*3*

Use `segments.csv` *as a starting point for your input. In the last line, 2 indicates that a marginal value functions will be composed of 2 linear pieces (i.e., it can change the slope in the middle), while 3 means that the slope of the function can change in the 1/3 and 2/3 of the criterion's range. Use the* `csvToXMCDA-criteriaValues` *module to create an* `XMCDA` *output containing the numbers of segments.*

Step U.2 : Elicitation of Thierry's reference ranking

MCDA exercise 4. *Ask Thierry to select a subset of about 3 to 5 cars that he knows best or these which can be compared by him most easily. Elicit from Thierry a complete ranking for these reference alternatives. Include the ideal alternative (id: fictiveBest) as the first in the reference ranking and the anti-ideal one (id: fictiveWorst) as the last one. The latter is needed due to an internal implementation of the procedure that will be used to derive the ranking.*

diviz exercise 13. *Store the preference ranking in a csv file according to the following format:*

	fictiveBest	*a11*	*a03*	*a13*	*a09*	*a14*	*fictiveWorst*
rank	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Use `ranking.csv` *as a starting point for your input. In the last line, "1" means that fictiveBest is first in the reference ranking, "2" means that a11 is second, etc. Use the* `csvToXMCDAalternativesValues` *module to create an* `XMCDA` *output containing these reference ranks.*

Step U.3 : Use a pre-defined procedure for selecting a value function compatible with Thierry's preferences

In order to select a single value function compatible with the Thierry's reference ranking, we will use ACUTA.

diviz exercise 14. *Use the* `ACUTA` *module to select marginal value functions compatible with Thierry's ranking. Double click the module and uncheck the options regarding using i) delta, ii) alternativesPreferences, and iii) alternativesIndifferences as inputs. Having configured the module in this way (i.e., using only alternativesRanks), provide it with all inputs it requires.*

diviz exercise 15. *Use the module* `plotValueFunctions` *to plot a graphical representation of value functions selected by ACUTA. Validate them with Thierry.*

diviz exercise 16. *Use the module* `computeNormalisedPerformanceTable` *to read off marginal values assigned to the performances of different cars according to the marginal value functions selected by ACUTA.*

diviz exercise 17. *Use the module* `generalWeightedSum` *to compute a comprehensive value of each alternative. Double click the module and set the aggregation operator as "Sum" (we do not need to account separately for weights as these are incorporated into the maximal values of the marginal functions). Remember to provide the marginal values (rather than the original performance table) at the input of this module.*

diviz exercise 18. *Determine the ranking of the cars via the* `rankAlternativesValues` *module. Double click this module to specify that the highest value has to be ranked first. Combine* `rankAlternativesValues` *with* `plotAlternativesPreorder` *to show Thierry the preorder of the cars obtained with ACUTA.*

Now that we have determined a recommendation for Thierry on the basis of ACUTA, let us outrank (first with Electre and then with PROMETHEE).

Step E.1 : Elicitation of Thierry's discrimination thresholds and criteria weights

MCDA exercise 5. *Question Thierry to obtain the performance thresholds (indifference, preference, veto) as used in the ELECTRE methods. Determine also the weights of the criteria representing his inter-criteria preferences.*

diviz exercise 19. *Store the thresholds in a csv file according to the following format:*

	g1	*g2*	*g3*	*g4*	*g5*
ind	*250*	*1*	*1*	*1*	*1*
pref	*1000*	*2*	*2*	*1.5*	*1.5*
veto	*2000*	*3*	*3*	*10*	*10*
preferenceDirection	*min*	*min*	*min*	*max*	*max*

Use `thresholdsElectre.csv` *as a starting point for your input. The last line represents the preference directions on the criteria, as expressed by Thierry.*

This csv file needs to be transformed into a valid `XMCDA` file to be compatible with the web-services callable from diviz.

diviz exercise 20. *Start a new workflow. Add* `XMCDA` *files with the specification of alternatives and performance table. Use the module* `csvToXMCDAcriteriaThresholds` *to transform the csv file containing the thresholds into* `XMCDA` *outputs.*

diviz exercise 21. *Store Thierry's weights in a csv file* `weightsOutranking.csv` *and transform it into an* `XMCDA` *output with the* `csvToXMCDAcriteriaValues` *module.*

Step E.2 : Construction of the Electre III outranking relation

The construction of the valued outranking relation requires two steps: the concordance test and the discordance test.

diviz exercise 22. *Use the modules* `ElectreConcordance (PUT)` *and* `ElectreDiscordance (PUT)` *to calculate the concordance and discordance indices linked to this problem. Since you wish to compare alternatives against each other, double click the modules and ensure that a parameter* `comparison_with` *is set to "alternatives vs alternatives" and that parameters use* `classes_profiles` *and* `profiles_performance_table` *are not checked.*

When providing these modules with the specification of criteria use the output of `csvToXMCDAcriteriaThresholds` *rather than* `csvToXMCDAcriteriaValues`*.*

Combine these results (concordance and discordance) with the module `ElectreCredibility (PUT)` *to calculate the fuzzy (valued) outranking relation according to the Electre III rule (use a default setting; ensure that* `classes_profiles` *are not required at the input).*

Step E.3 : Determination of the ranking with Electre III

In the first step, the downward and upward distillation procedures need to be performed. Then, combine them into a final ranking (intersection).

diviz exercise 23. *Use two instances of the module* `ElectreDistillation`*. Parameterize one of them (double click) to conduct a downward distillation and the other to perform an upward distillation.*
Combine their outcomes (rankings) using `ElectreDistillationRank` *module. Use the module* `plotAlternativesHasseDiagram` *to plot the output final ranking (intersection). When drawing it, this module applies a transitive reduction (to remove unnecessary arcs that can be obtained from transitivity).*

Which car(s) should be recommended to Thierry according to this approach?

Step E.4 : Alternative Electre-based ranking (NFS)

If you wish to obtain a ranking based on yet different exploitation algorithm than offered by Electre III, you can use, e.g., the Net Flow Score procedure. It considers for each alternative arguments in favor of its strength and weakness, deriving in this way a comprehensive score of its desirability.

diviz exercise 24. *Use the module* `ElectreNFSOutranking`*. Parameterize it (double click) to exploit fuzzy (valued) outranking relation (crisp_outranking not checked) and use information only about outranking (use non_outranking as input not checked). Parameterize the module to exploit the matrix of credibilities (output of the* `ElectreCredibility (PUT)` *module).*

Which car(s) should be recommended to Thierry using NFS?

Step P.1 : Reminder on PROMETHEE I and II

PROMETHEE I and PROMETHEE II are outranking methods which deliver, respectively, partial and complete rankings. They were developed by J.P. Brans and presented for the first time in 1982.

Preference information. The preference information consists in a preference function $\pi_j(a,b)$ associated to each criterion g_j (e.g., parameterized with indifference q_j and preference p_j thresholds) as well as weights w_j describing their relative importance.

Marginal preference functions. Although six types of particular preference functions have been proposed in PROMETHEE, we will consider the one that is most frequently used (see Figure 1). It is defined so that $\pi_j(a,b) = 0$ if $g_j(a) - g_j(b) \leq q_j$, $\pi_j(a,b) = 1$ if $g_j(a) - g_j(b) \geq p_j$, whereas if $g_j \leq g_j(a) - g_j(b) \leq p_j$, a is weakly preferred to b on g_j and $\pi_j(a,b) \in (0,1)$.

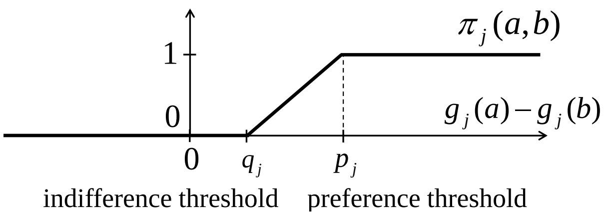

Figure 1: Marginal preference index $\pi_j(a,b)$

Comprehensive preference. To express the degree in which a is preferred to b over all criteria, PROMETHEE refers to a comprehensive preference index:

$$\pi(a,b) = \sum_{j=1}^{m} w_j \pi_j(a,b) \text{ for all } (a,b) \in A \times A,$$

where w_j is a weight associated with g_j.

Positive, negative and comprehensive flows. The positive outranking flow $\Phi^+(a)$ expresses how much alternative a is outranking all the other $n-1$ alternatives:

$$\Phi^+(a) = 1/(n-1) \sum_{b \in A} \pi(a,b),$$

whereas the negative outranking flow $\Phi^-(a)$ expresses how much alternative a is outranked by all the others:

$$\Phi^-(a) = 1/(n-1) \sum_{b \in A} \pi(b,a).$$

These flows are graphically represented in Figure 2. Obviously, the higher the positive flow and the smaller the negative flows, the better the alternative.
The balance between the positive and the negative flows is reflected in the comprehensive (net) outranking flow:

$$\Phi(a) = \Phi^+(a) - \Phi^-(a).$$

Obviously, the higher the comprehensive flow, the better.

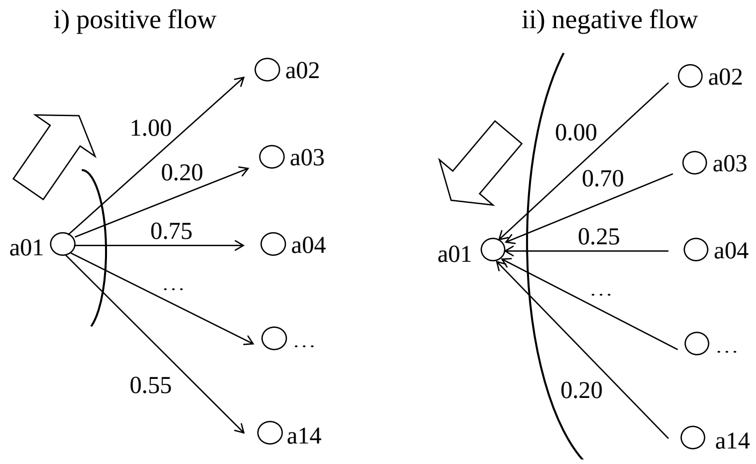

Figure 2: Positive and negative flows.

PROMETHE II. A complete ranking in PROMETHEE II is determined by $\Phi(a)$, i.e., a is preferred to b (aPb) if $\Phi(a) > \Phi(b)$, whereas a is indifferent with b (aIb) if $\Phi(a) = \Phi(b)$. No incomparability can occur.

PROMETHE I. A partial ranking in PROMETHEE I is determined through positive $\Phi^+(a)$ and negative $\Phi^-(a)$ flows, thus, admitting incomparability. Precisely, a is preferred to b (aPb) if one of its flows (i.e., positive or negative one) is strictly better while the other flow is not worse:

$$\begin{aligned} aPb \Leftrightarrow & \ \Phi^+(a) > \Phi^+(b) \text{ and } \Phi^-(a) < \Phi^-(b), \\ & \text{or } \Phi^+(a) > \Phi^+(b) \text{ and } \Phi^-(a) = \Phi^-(b), \\ & \text{or } \Phi^+(a) = \Phi^+(b) \text{ and } \Phi^-(a) < \Phi^-(b). \end{aligned}$$

The indifference between a and b (aIb) occurs if they have the same positive and negative flows:

$$aIb \Leftrightarrow \Phi^+(a) = \Phi^+(b) \text{ and } \Phi^-(a) = \Phi^-(b).$$

Finally, two alternatives are considered incomparable (aRb) if the results of a comparison between their positive and negative flows are contradictory:

$$\begin{aligned} aRb \Leftrightarrow & \ \Phi^+(a) > \Phi^+(b) \text{ and } \Phi^-(a) > \Phi^-(b), \\ & \text{or } \Phi^+(a) < \Phi^+(b) \text{ and } \Phi^-(a) < \Phi^-(b). \end{aligned}$$

Step P.2 : Construct PROMETHEE workflows to help Thierry

MCDA exercise 6. *Question Thierry to obtain the discrimination thresholds (indifference and preference) representing his intra-criteria preferences, and weights of the criteria.*

diviz exercise 25. *Store the thresholds in a csv file according to the following format:*

	g1	*g2*	*g3*	*g4*	*g5*
ind	*250*	*1*	*1*	*1*	*1*
pref	*1000*	*2*	*2*	*1.5*	*1.5*
preferenceDirection	*min*	*min*	*min*	*max*	*max*

Use `thresholdsPromethee.csv` *as a starting point for your input.*

diviz exercise 26. *Start a new workflow (Workflow - New). Add* `XMCDA` *files with the specification of alternatives and performance table. Use the module* `csvToXMCDA-criteriaThresholds` *to transform the csv file containing the thresholds into* `XMCDA` *outputs.*

diviz exercise 27. *Store Thierry's weights in a csv file* `weightsOutranking.csv` *and transform it into an* `XMCDA` *output with the* `csvToXMCDA-criteriaValues` *module.*

diviz exercise 28. *Extend the workflows to deliver and visualise the ranking obtained with PROMETHEE I and II. Use the following modules (you need to discover how to parameterize and combine them on your own):*

- `PrometheePreference (J-MCDA)` *for marginal preference indices;*
- `PrometheeFlows (J-MCDA)` *for positive, negative or comprehensive flows - you need its three suitably parameterized instances, one for each type of flow;*
- `Promethee1Ranking (RXMCDA)` *for constructing the ranking with Promethee I;*
- `rankAlternativesValues (RXMCDA)` *for deriving ranks of the alternatives based on their values (flows);*
- `plotAlternativesValuePreorder (ITTB)` *for visualising a complete ranking;*
- `plotAlternativesHasseDiagram (PUT)` *for visualising some relation (in particular, a partial ranking) while applying a "transitive reduction".*

Step C.1 : Comparing the rankings obtaing with MAVT and PROMETHEE II

Assist Thierry in comparing the rankings obtained with MAVT and PROMETHEE II.

diviz exercise 29. *To compare the two rankings graphically use* `XYPlotAlternatives-Values` *module. Provide the ranks of alternatives (obtained with* `rankAlternativesValues` *module) at its input.*

Discuss with Thierry the similarities and major differencies in the ranks attained by some cars. Can you explain why there is a difference in the rankings obtained via the MAVT approach and the PROMETHEE one?

diviz exercise 30. *To objectively measure the similarity between the two rankings use* `alternativesValuesKendall` *module. Provide the ranks of alternatives at its input. Explain to Thierry that a value close to* 1 *means that the two rankings are very similar, whereas when the Kendall's* τ *is close to* -1*, most pair-wise preference relations from one ranking are inverse in the other.*

Step A.1 : Multiple criteria sorting – just in case you need it

In the exercises conducted so far, you have ranked a set of cars using various methods implementing different MCDA schools. In case you need some examples with multiple criteria sorting methods, have a look at how the "Thierry's sorting problem" can be approached with:

- Electre TRI-B with boundary class profiles;
- Electre TRI-C and Electre TRI-rC with characteristic (central) class profiles;
- FlowSort-I and FlowSort-II with characteristic class profiles.

These are complete workflows that you can just import to diviz and execute.

References

[BMP+00] 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. 1