BeSmart2: A multicriteria decision aid application

Anabela Tereso[1] and João Amorim[1],

[1] Department of Production and Systems Engineering
Algoritmi Centre
University of Minho, 4800-058 Guimarães, Portugal
anabelat@dps.uminho.pt, jmrcamorim@outlook.com

Abstract. This paper presents an improved version of an application whose goal is to provide a simple and intuitive way to use multicriteria decision methods in day-to-day decision problems. The application allows comparisons between several alternatives with several criteria, always keeping a permanent backup of both model and results, and provides a framework to incorporate new methods in the future. Developed in C#, the application implements the AHP, SMART and Value Functions methods.

Keywords: Multicriteria Decision Aid, AHP, SMART, Value Functions

1 Introduction and literature review

The problem of decision making has been throughout the human history a topic of reflection by many scholars, being suggested that the ability to decide is what separates us from other species. As we moved to more recent times, decision problems have grown in size, frequency and importance. The immense competition between companies makes that each decision counts, and the slightest advantage can make or break the future success of a company.

However, making a decision can be a very difficult task without adequate tools, either due to uncertainty, the time-span that it will affect, or just due to the sheer amount of information involved. It should also be noted that the human mind is naturally biased when it comes to decision making, as it tends to give more relevance to the first information received, and to make choices in order to justify previous decisions regardless of their current validity [**1**].

In order to help with the decision process, several decision aid software tools were created, some with a specific purpose and others with a more general purpose. Some examples are: tools for multiple objective problems (ADBASE [**2**], Tommix [**3**]); tools for ordering problems (Electre Tri [**4**], IRIS [**5**]); tools for group decision (AGAP [**6**], WINGDSS [**7**]) and tools for multiple criteria problems, like Criterium Decision Plus [**8**] and WebHipre [**9**].

In BeSmart2 we implemented some of the most widely known Multicriteria Decision methods, such as the Analytic Hierarchy Process (AHP) [**10**], Simple MultiAttribute Rating Technique (SMART) [**11**], Value Functions [**12**], and a simple interface to input multicriteria problems of any kind is provided.

We started by making a literature review of multicriteria decision methods and software [**13**], in order to identify the most important features to include in the tool (summarized in section 2).

Then we changed the first version of the BeSmart tool [**14**] and improved it by implementing:

- Unlimited number of alternatives for each comparison.
- Allowing a multilevel hierarchy of objectives and criteria.
- Detailed analysis of results and sensitivity analysis.
- Permanent storing of results or partially filled comparisons.
- Revamped graphical interface.

A description of the implementation details can be found in section 3.

Section 4 describes the results obtained and finally section 5 presents the conclusions and future research.

2 Description of methods used

The software currently implements three multicriteria methods: AHP [**10**], SMART [**11**] and Value Functions [**12**], still allowing for manual introduction of weights in case no method is selected. The methods used will be described next.

2.1 AHP

In the AHP method weights are attributed to alternatives or criteria through a matrix of pairwise comparisons between alternatives for a given criterion, or between criteria for a given intermediate or global objective. The comparison between alternatives or criteria must be made according to the following table.

Table 1. AHP preference values (adapted from [**10**])

If x is … y	**Preference value**
As important as	1
Slightly more important than	3
More important than	5
Much more important than	7
Extremely more important than	9

In case of reverse comparisons, i.e., y being compared to x, the inverse values should be used (1/1, 1/3, 1/5, 1/7, 1/9). Taking an example with three criteria A, B and C, where A is more important than B, and slightly more important than C, and B is slightly less important than C, we would get the matrix shown on table 2.

To calculate the weights for each criterion, the software uses an approximation to the eigenvector with the largest eigenvalue. This approximation consists of normalizing each column (dividing each element by the column sum) and then adding

each line and dividing by the number of criteria. For this example, we would get the weights shown in table 3.

Table 2. AHP matrix example

	A	**B**	**C**
A	1	5	3
B	1/5	1	1/3
C	1/3	3	1

Table 3. Calculated AHP weights

Criterion	**Weight**
A	0.63
B	0.11
C	0.26

To ensure consistency, the software calculates the Consistency Index proposed by Saaty [**10**], and warns the user in case its value is above 0.1.

2.2. SMART

The SMART method [**11**] is a simple and quick way of weighting alternatives or criteria. First the worst alternative or less important criterion is scored with 10 points, and then all the other alternatives or criteria will be scored based on that.

Taking again an example with three criteria, we will rank B as the less important and score it with 10 points. We judge A to be four times more important, and C to be two times more important, so we score them with 40 and 20 points respectively. The weights will be calculated by dividing the points by the sum of the points of every criteria (see table 4).

Table 4. SMART points and calculated weights

Criterion	**Points**	**Weight**
A	40	0.57
B	10	0.14
C	20	0.29

2.3. Value Functions

Value Functions are functions that assign a value to each alternative based on the concept of preference differences [**12**]. Objectively, the value function takes four

parameters, alternative value for the criteria in analysis, range of values for the criteria (maximum and minimum value) and an exponential factor different than 0, if an exponential function, instead of a linear function, best describes the decision maker profile, and will return a score between 0 and 1. Since it uses the alternative values, it can only be used to rank alternatives in criteria with numeric scales, or scales that can be converted to numeric values.

The first step is to calculate the linear value for an alternative, also known as rescaling. Depending on whether the objective for the criteria is to maximize or to minimize, we will use formula (1) or (2) respectively.

$$Linear\ value\ (x) = \frac{x - min}{max - min} \quad (1)$$

$$Linear\ value\ (x) = \frac{max - x}{max - min} \quad (2)$$

The exponential factor α indicates the preferences of the decision maker (his risk profile). Positive values indicate that differences in values closer to the optimal value (either maximum or minimum depending of the objective) are more important than differences in values closer to the worst.

For example, if we were rating hotels, a positive exponential factor indicates that the difference between 4 and 5 stars is more important than the difference between 1 and 2 stars. Negative values indicate the opposite, and the higher the value is (in module) the stronger this preference is.

Formula (3) indicates the full calculations for the value function. In case the exponential factor α is 0, the formula for the value function is the same as the linear value.

$$Value\ Function(x) = \frac{e^{\alpha * Linear\ value(x)} - 1}{e^{\alpha} - 1} \quad (3)$$

Taking an example of a criteria with minimum value 0 and maximum value 10, with a maximization objective, and an exponential factor of 2 we would get the value unction represented in Figure 1.

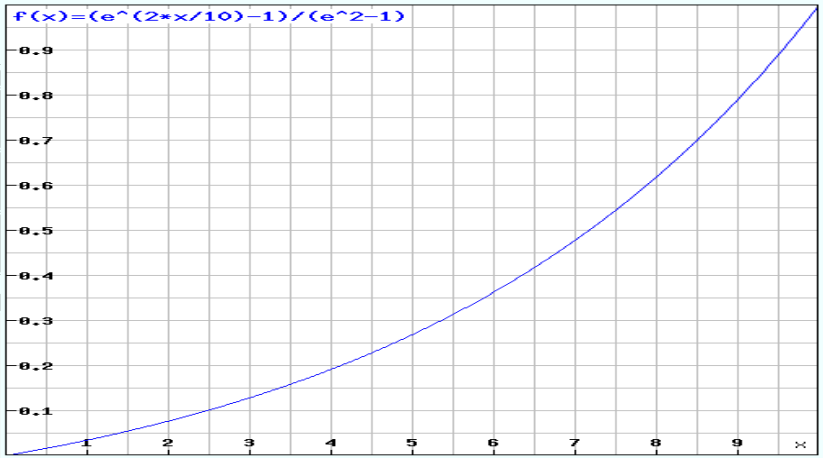

Figure 1. Graph representing a value function with positive exponential factor

2.4. Global calculations

To ensure that we can add the weights or scores of the different alternatives regarding different criteria, we must convert them to the same scale. Therefore, the software allows the user to choose whether the results should be shown in weights (sum of all alternatives scores equal to 1) or in scores (the value represents how close to the "ideal" the alternative is, in a scale from 0 to 1).

The conversions used are identical to the ones used in the Web-Hipre software tool [**9**]. To convert from weights to score, we attribute score 1 to the highest score and divide the remaining weights by the highest weight to get the remaining scores. To convert scores to weights, we divide each score by the sum of all scores (see table 5).

Table 5. Example of conversions between scores and weights

Scores →Weights		**Weights → Scores**	
0.3	0.24	0.5	1
0.55	0.44	0.125	0.25
0.41	0.32	0.375	0.75

After converting everything to the same scale, the scores or weights for each alternative are evaluated for each intermediate objective, all the way to the global objective. The score (or weight) for an alternative is the sum of all the scores in the subcriteria (or subobjective) for that objective, multiplied by the weight of each subcriterion.

2.5. Sensitivity Analysis

The software allows the user to do a sensitivity analysis by changing the weight of a determined criterion or subobjective. If a user changes the weight of a criterion **j** (or subobjective) from $\mathbf{p_j}$ to $\mathbf{p_j'}$, assuming there are **N** criteria (or subobjectives) the remaining weights will be given by expression (4).

$$p(i) = \begin{cases} p_j', & i = j \\ p_i \times \left(1 + \frac{p_j - p_j'}{\sum_{k \in N, k \neq j} p_k}\right), & i \neq j \end{cases} \quad (4)$$

After calculating the new weights, the global calculations are redone the same way as shown in the previous section.

3 Software implementation

Since the software was made with the goal of upgrading existing software [**14**] , it was not necessary a traditional requirement analysis phase. Nevertheless, the key points we focused on were creating a more user friendly interface and flexibility to include new methods. This section summarizes the process of software development, from modelling to implementation.

3.1. Use Case Diagram

The Use Case Diagram (Figure 2) represents various possible interactions between the user and the system. We divided the uses cases in three different groups: model persistency (blue), model data (criteria, categories, alternatives) edition (green) and comparisons (orange).

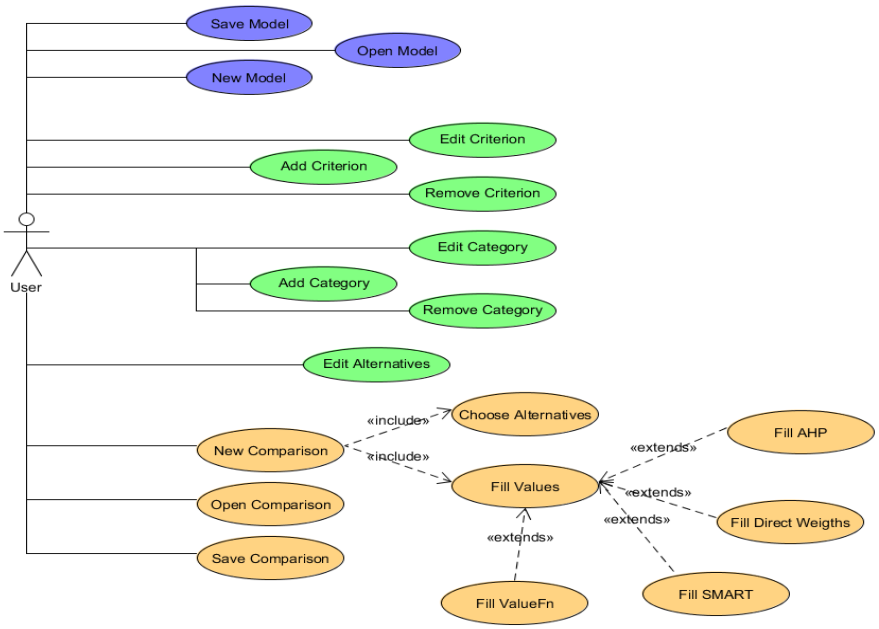

Figure 2. Use case diagram of BeSmart2

3.2. Class Diagram

The class diagram (Figure 3) describes the structure of the system, displaying its classes, attributes, methods and operations, as well as the relationships between the various objects.
The software can be divided in two groups of classes: *Business* (orange) and *Interface* (green). The *Business* group has all the classes that are related to data management (saving, loading), data structures (criteria, categories) and calculations. The *Interface* group includes all the classes that deal with the user interface.
Functionally, we can also divide the tool in two groups: classes related with the data model (left half of the diagram) and classes related with comparisons and calculations (right half of the diagram).

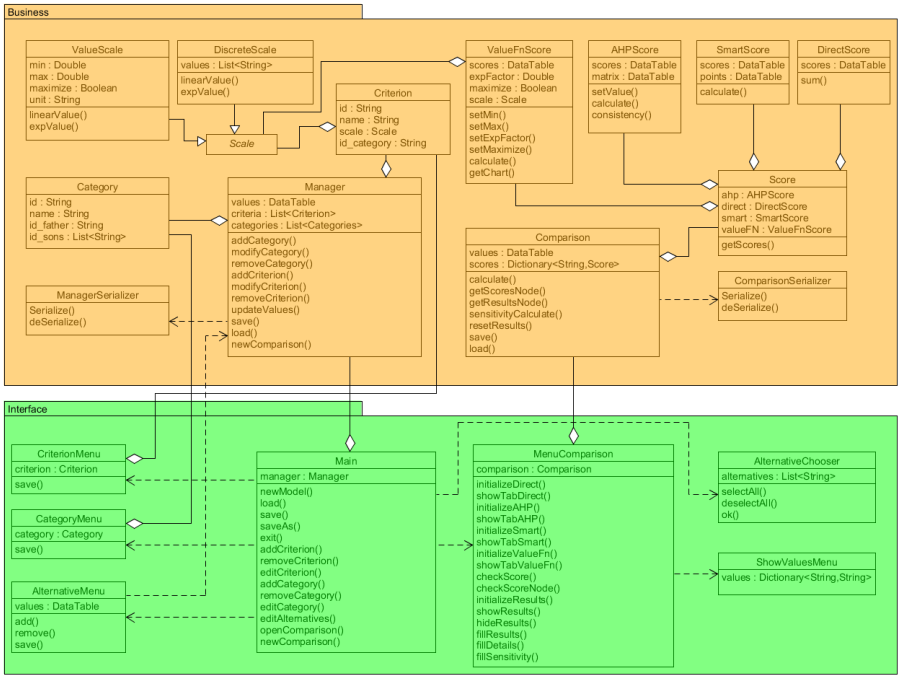

Figure 3. Class diagram of BeSmart2

3.3. Development

After modelling, the implementation of the software was done using C# [**15**] in order to reuse the code from the first version of BeSmart. The visual interface was done using Windows Forms [**16**].

To ensure the extensibility of the software, the code was developed modularly in order to easily accommodate new multicriteria methods. To integrate a new model, one just needs to:

- Create a new MethodScore class that does the calculations and has an attribute that will be a table with the final results for that method.
- Modify the Score class to accommodate the new class (include a new attribute and change the switch portions of the code).
- Include a new tab in the interface to input the values for the new method, and update the tab switching mechanism.

To implement a completely different method, namely some outranking methods like Electre [**17**] or Promethee [**18**], a new comparison interface can be built, since the model management and comparison interfaces of the software are completely independent.

4 Results

The result of this work was a fully working multicriteria decision support software application, directed to solve generic decision problems. The software and source code can be downloaded at https://code.google.com/p/besmart2/.

In Figure 4 we can see the results of a comparison between three potential candidates (alternatives) for an entry-level job opening, being evaluated based on their grades, degree and immediate availability.

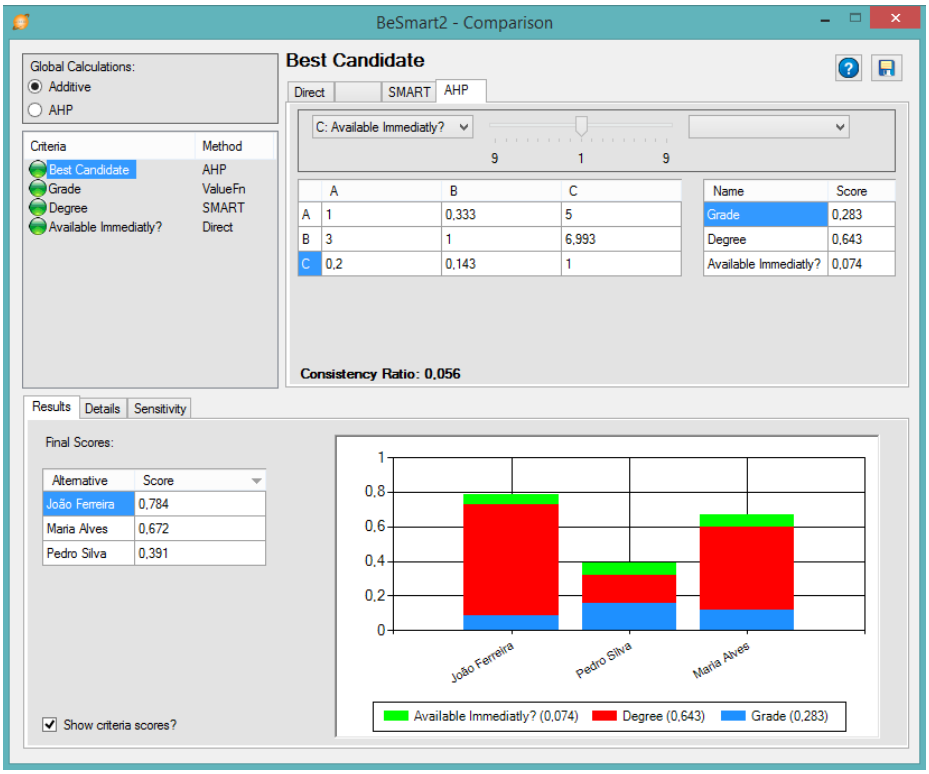

Figure 4. Results interface

The detailed scores for each criterion are shown in order to easily identify what is weighting the most in the final results. The user can also use the sensitivity interface (Figure 5) to see what effect on the final result will have changes on a specific variable.

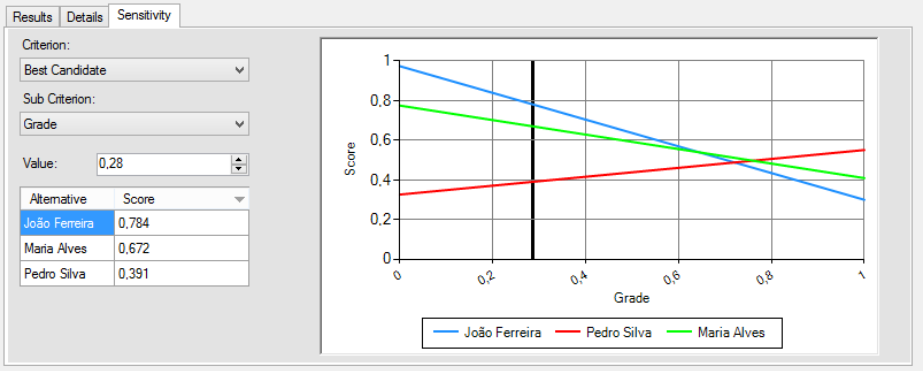

Figure 5. Sensitivity interface

5 Conclusions and future research

In this paper we present the description of the research that led to the improvement of a decision aid application, BeSmart [**14**]. As the first version, BeSmart2 allows the ranking of up to 16 alternatives in multicriteria problems using AHP, SMART and Value Functions. In this improved version, the application now allows the definition of multilevel objectives, detailed results, sensitivity analysis and permanent backup of comparisons and results.

A comparative analysis between this tool and existing software lead us believe that this application is useful to help decision making in different scenarios. Compared to similar tools, like Web-Hipre [**9**] and the original BeSmart [**14**], one of the points that stands out is the ability to see real time changes in the results as the parameters are changed. Another feature not present in many tools is the possibility of saving different comparisons and the results for the same model, without the need of recreating the model or making a copy of its backup, since the backup of the model and results are independent files.

In the future, we hope to implement more methods with different scopes (like ordering methods). It is also on the horizon the development of a more detailed report system, that allows the user to see the result data as needed, and a graphical help module, that guides the user step by step.

6 References

1. J. S. Hammond, R. L. Keeney and H. Raiffa, *Smart choices: A practical guide to making better life decisions*, Random House LLC1999.
2. R. E. Steuer, *Manual for the adbase multiple objective linear programming package*, Department of Management Science and Information Technology, University of Georgia (1992).

3. C. H. Antunes, M. J. Alves, A. L. Silva and J. N. Climaco, *An integrated molp method base package—a guided tour of tommix*, Computers & operations research **19** (1992), no. 7, 609-625.
4. W. Yu, *Electre tri(aspects méthodologiques et manuel d'utilisation)*, Document- Université de Paris-Dauphine, LAMSADE (1992).
5. L. C. a. M. V. Dias, *Iris: A dss for multiple criteria sorting problems*, Journal of Multi-Criteria Decision Analysis **12** (2003), no. 4-5, 285--298.
6. J. P. Costa, P. Melo, P. Godinho and L. s. C. Dias, *The agap system: A gdss for project analysis and evaluation*, European Journal of Operational Research **145** (2003), no. 2, 287-303.
7. P. Csáki, T. Rapcsák, P. Turchányi and M. Vermes, *R and d for group decision aid in hungary by wingdss, a microsoft windows based group decision support system*, Decision Support Systems **14** (1995), no. 3, 205-217.
8. W. Haerer, *Software review: criterium decision plus 3.0*, OR/MS Today **27** (2000), no. 1.
9. J. Mustajoki and R. P. Hamalainen, *Web-hipre: Global decision support by value tree and ahp analysis*, INFOR J **38** (2000), no. 3, 208-220.
10. T. L. Saaty, *The analytic hierarchy process : Planning, priority setting, resource allocation*, McGraw-Hill, New York ; London, 1980.
11. J. a. H. R. P. a. S. A. Mustajoki, *Decision support by interval smart/swing—incorporating imprecision in the smart and swing methods*, Decision Sciences **36** (2005), no. 2, 317--339.
12. J. S. Dyer and R. K. Sarin, *Measurable multiattribute value functions*, Operations Research **27** (1979), no. 4, 810-822.
13. A. Tereso and C. Seixedo, "Multi-criteria decision aid: Evaluation and comparison of main techniques/tools", *24th European Conference on Operational Research (EURO XXIV)*, Lisbon - Portugal, July 11-14, 2010.
14. A. Tereso, A. Sampaio, H. Frade, M. Costa and T. Abreu, "Besmart: A software tool to support the selection of decision software," *International Conference on Engineering UBI2011 (ICEUBI2011)*, Covilhã – Portugal, 2011.
15. Microsoft visual c# webpage - http://msdn.Microsoft.Com/en-us/vstudio/hh341490.Aspx.
16. Microsoft windows forms webpage - http://msdn.Microsoft.Com/en-us/library/dd30h2yb(v=vs.110).Aspx.
17. B. Roy, *Classement et choix en présence de points de vue multiples*, RAIRO-Operations Research-Recherche Opérationnelle **2** (1968), no. V1, 57-75.
18. J. P. Brans and P. Vincke, *A preference ranking organisation method: The promethee method for mcdm*, Management Science **31** (1985), no. 6, 647-656.