April 1-3, Azores, Portugal

University of Minho
School of Engineering

1

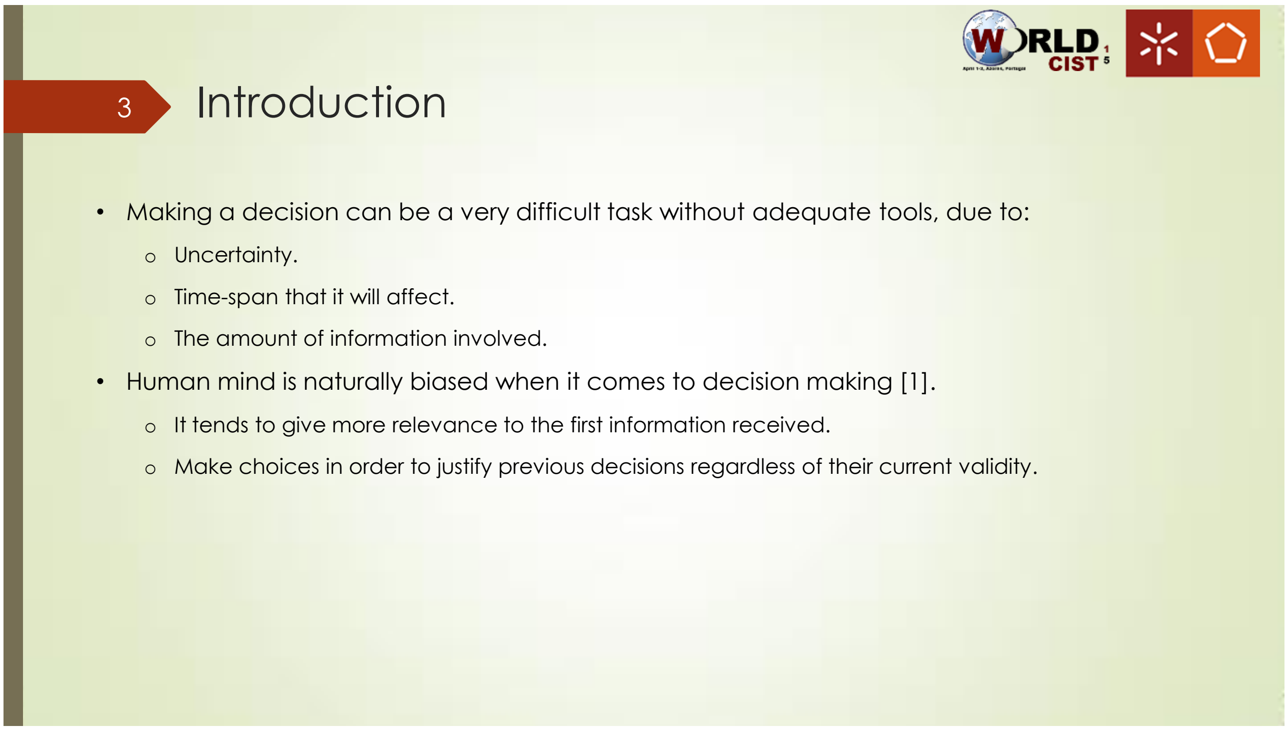
BeSmart2: A multicriteria decision aid application

Anabela Tereso and João Amorim
Department of Production and Systems Engineering
Algoritmi Centre – University of Minho - Portugal

2

Introduction

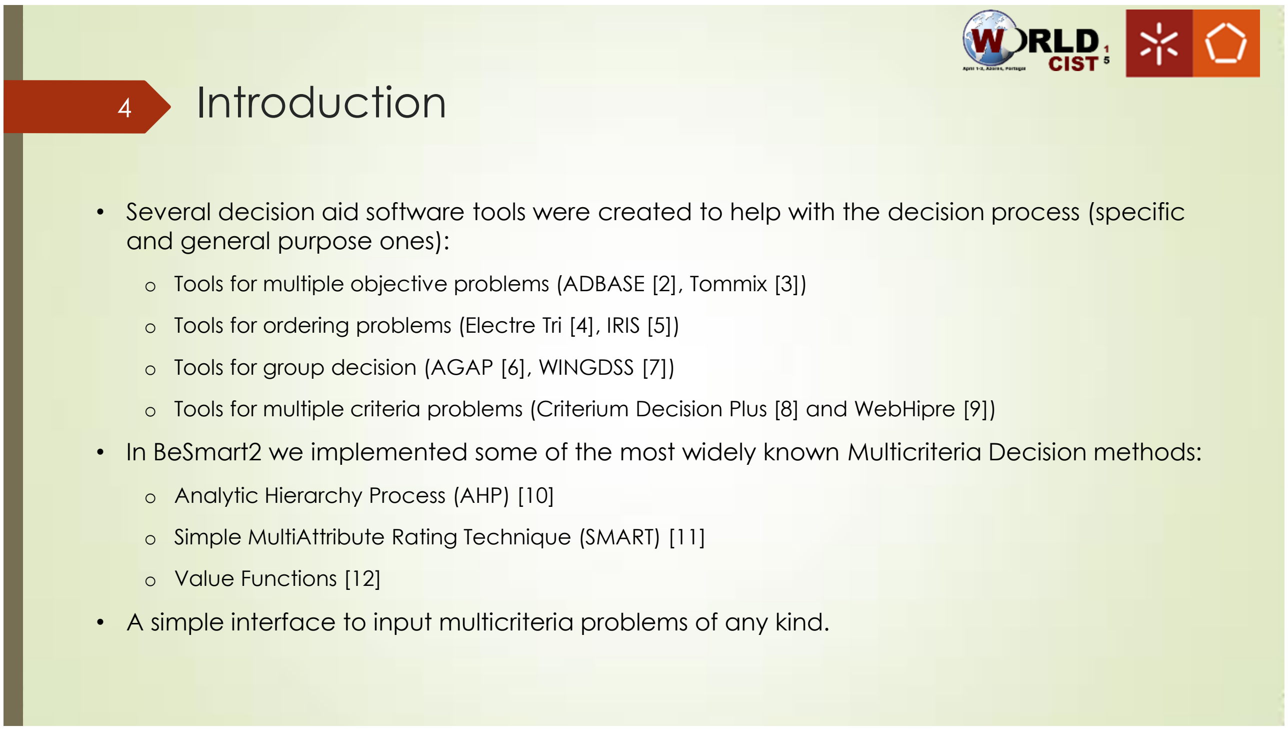
- Improved version of BeSmart [14].
- Goal: provide a simple and intuitive way to use multicriteria decision methods.
- Allows comparisons between several alternatives with several criteria.
- Permanent backup of both model and results.
- Provides a framework to incorporate new methods in the future.
- Developed in C#.
- AHP, SMART and Value Functions.



3

Introduction

- Making a decision can be a very difficult task without adequate tools, due to:
    - Uncertainty.
    - Time-span that it will affect.
    - The amount of information involved.
- Human mind is naturally biased when it comes to decision making [1].
    - It tends to give more relevance to the first information received.
    - Make choices in order to justify previous decisions regardless of their current validity.



4

Introduction

- Several decision aid software tools were created to help with the decision process (specific and general purpose ones):
    - 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])
    - Tools for multiple criteria problems (Criterium Decision Plus [8] and WebHipre [9])
- In BeSmart2 we implemented some of the most widely known Multicriteria Decision methods:
    - Analytic Hierarchy Process (AHP) [10]
    - Simple MultiAttribute Rating Technique (SMART) [11]
    - Value Functions [12]
- A simple interface to input multicriteria problems of any kind.

5

Introduction

- We changed the first version of the BeSmart tool [14] and improved it by implementing:
  - Unlimited number of alternatives for each comparison
  - A multilevel hierarchy of objectives and criteria
  - Detailed analysis of results and sensitivity analysis
  - Permanent storing of results or partially filled comparisons
  - Graphical interface improvement

6

Description of methods used - AHP

- Based on pairwise comparisons between alternatives for a given criterion, or between criteria for a given intermediate or global objective.

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).

7

Description of methods used - AHP

- Example with three criteria A, B and C: A is more important than B, and slightly more important than C, and B is slightly less important than C.
- 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 and then adding each line and dividing by the number of criteria.
- 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.

AHP matrix example

	A	B	C
A	1	5	3
B	1/5	1	1/3
C	1/3	3	1
Sum	1.53	9.00	4.33

Calculated AHP weights

	A	B	C	Sum	Weight
A	0,65	0,56	0,69	1,90	**0.63**
B	0,13	0,11	0,08	0,32	**0.11**
C	0,22	0,33	0,23	0,78	**0.26**

8

Description of methods used - 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.
- Example with three criteria A, B and C: B – the less important; A – 4X more important than B; C – 2X more important than B.
- The weights will be calculated by dividing the points by the sum of the points of every criteria.

SMART points and calculated weights

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

9

Description of methods used - Value Functions

- Value Functions are functions that assign a value to each alternative based on the concept of preference differences [12].
- The value function takes four parameters:
  - Alternative value for the criteria in analysis
  - Maximum value of the criteria
  - Minimum value of the criteria
  - An exponential factor different than 0, if an exponential function, instead of a linear function, best describes the decision maker profile
- It returns a score between 0 and 1.
- 1st Step: Calculate the linear value for an alternative
  - Objective: maximize criteria $\longrightarrow$ $\boxed{Linear\ value\ (x) = \frac{x - min}{max - min}}$
  - Objective: minimize criteria $\longrightarrow$ $\boxed{Linear\ value\ (x) = \frac{max - x}{max - min}}$

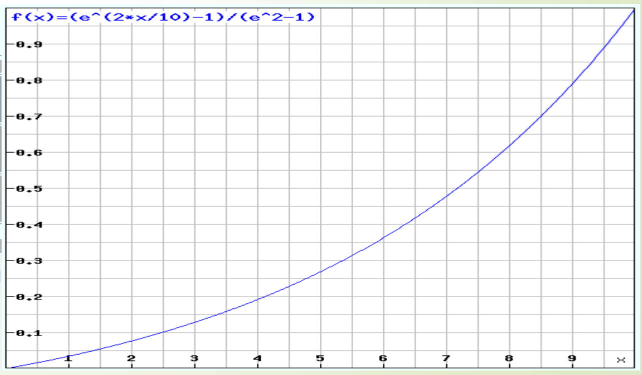
10

Description of methods used - Value Functions

- The exponential factor $\boldsymbol{\alpha}$ indicates the preferences of the decision maker (his risk profile).
- Positive values indicate that differences in values closer to the optimal value are more important than differences in values closer to the worst.
- Negative values indicate the opposite.
- The higher the value is (in module) the stronger the preference is.
- Ex: criteria with minimum value 0 and maximum value 10, with a maximization objective, and an exponential factor 2:

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

Graph representing a value function with positive exponential factor



Positive values indicate that differences in values closer to the optimal value are more important than differences in values closer to the worst: P.e. 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.

11

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 weights equal to 1), or
  - Scores (the value represents how close to the "ideal" the alternative is, in a scale from 0 to 1).
- To convert scores to weights, we divide each score by the sum of all scores.
- 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.

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

12

Global calculations / Sensitivity Analysis

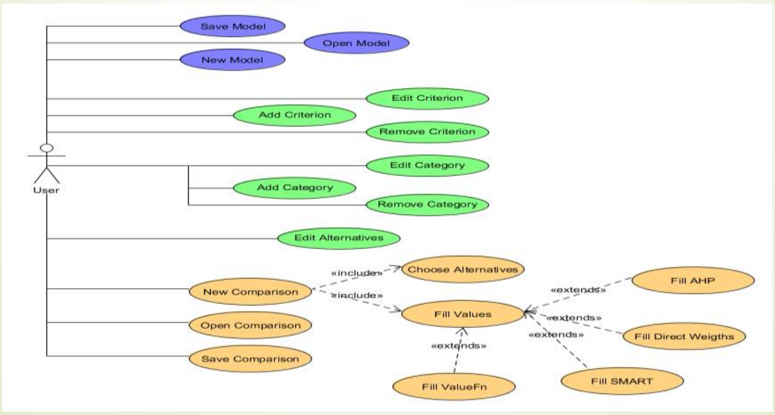
- After converting everything to the same scale, the scores or weights for each alternative are evaluated for each intermediate objective, all the way to the global objective.
- The score for an alternative is the sum of all the scores in the subcriteria for that objective, multiplied by the weight of each subcriterion.
- The software allows the user to do a sensitivity analysis by changing the weight of a criterion or subobjective.
- If a user changes the weight of a criterion j from pj to pj', assuming there are N criteria, the remaining weights will be given by:

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

13

Software implementation

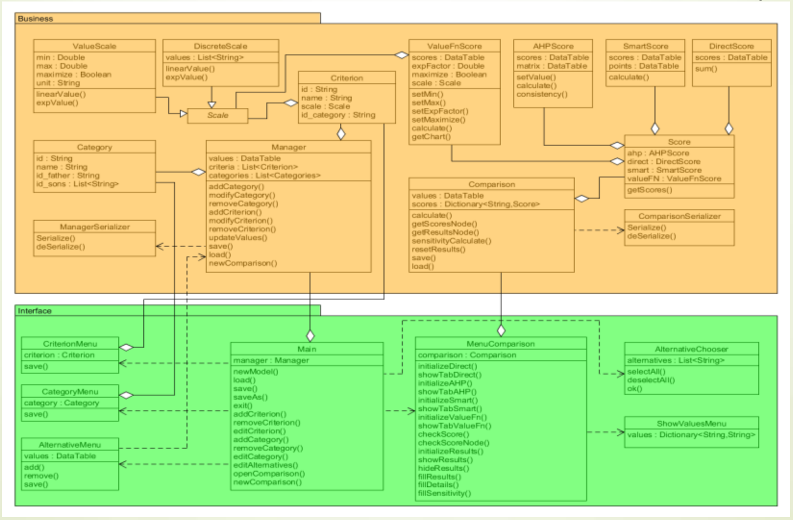
- Since the software was made with the goal of upgrading existing software [14] , it was not necessary a traditional requirement analysis phase.
- We focused on creating a more user friendly interface and flexibility to include new methods.
- A **Use Case Diagram** represents various possible interactions between the user and the system.



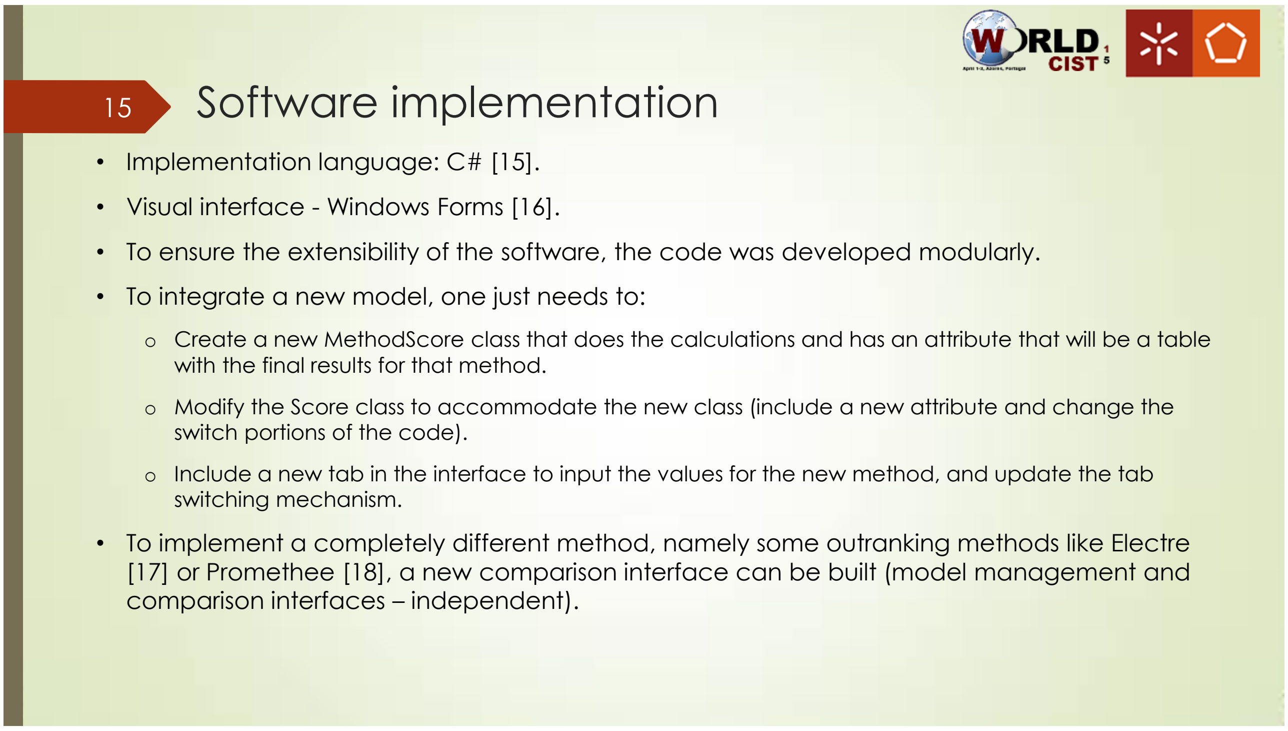
14

Software implementation

- The **class diagram** describes the structure of the system, displaying its classes, attributes, methods and operations, as well as the relationships between the various objects.



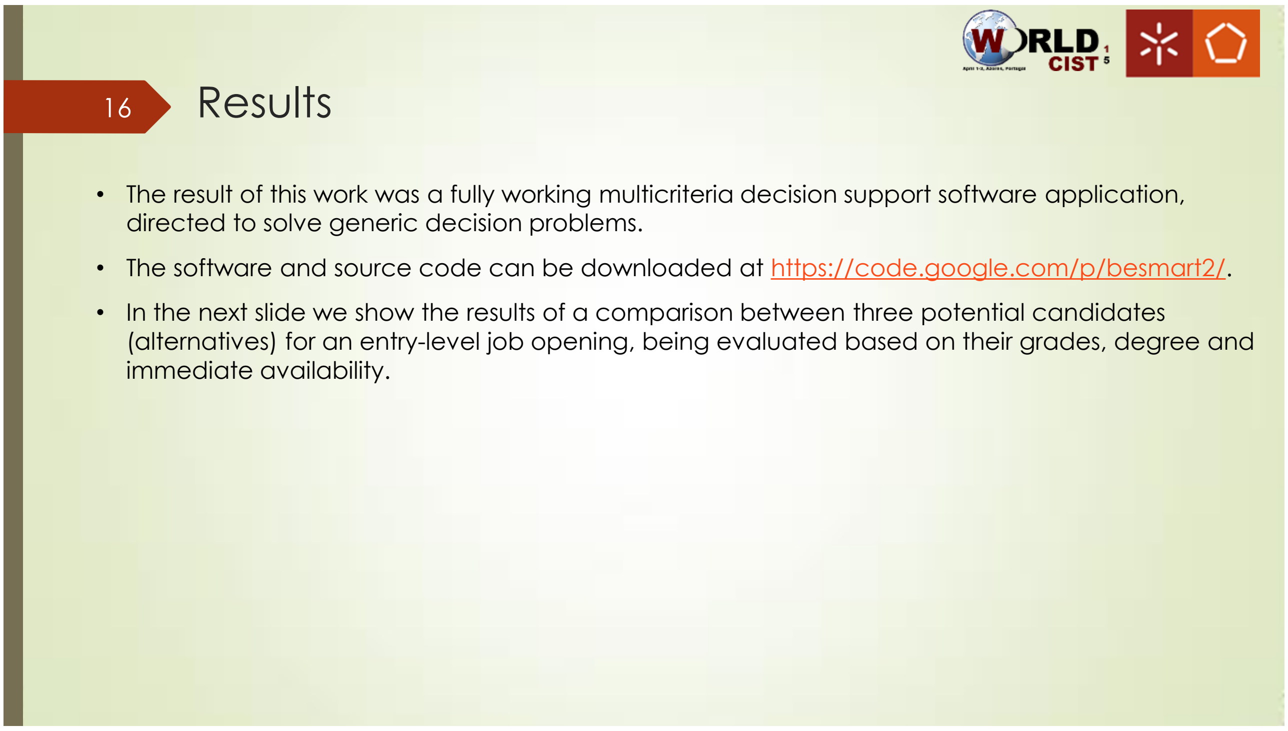
The software can be divided in two groups of classes: Business (orange) and Interface (green). The Business group has all the classes that are related to data management (saving, loading), data structures (criteria, categories) and calculations. The Interface group includes all the classes that deal 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).



15

Software implementation

- Implementation language: C# [15].
- Visual interface - Windows Forms [16].
- To ensure the extensibility of the software, the code was developed modularly.
- 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 (model management and comparison interfaces – independent).



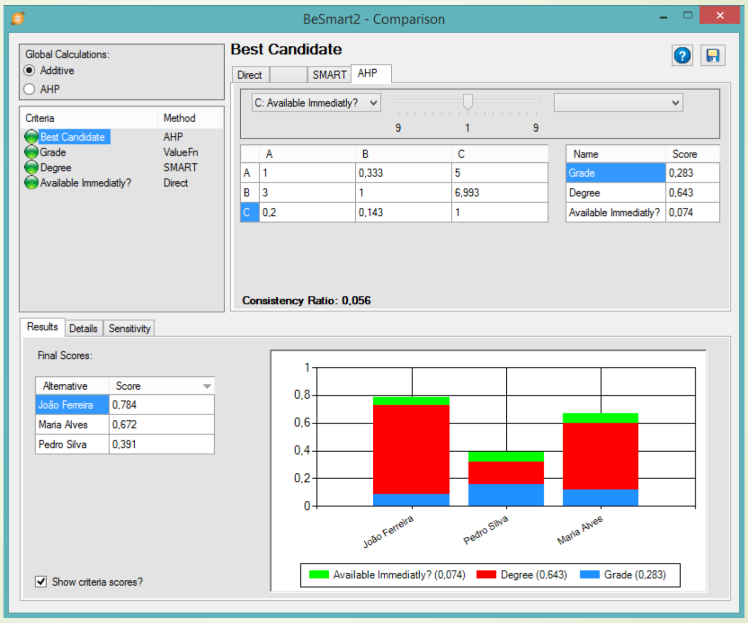
16

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 the next slide we show 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.

17

Results


BeSmart2 - Comparison

Global Calculations:
(●) Additive
( ) AHP

Criteria	Method
Best Candidate	AHP
Grade	ValueFn
Degree	SMART
Available Immediatly?	Direct

Best Candidate

C: Available Immediatly?

	A	B	C
A	1	0,333	5
B	3	1	6,993
C	0,2	0,143	1

Name	Score
Grade	0,283
Degree	0,643
Available Immediatly?	0,074

Consistency Ratio: 0,056

Final Scores:

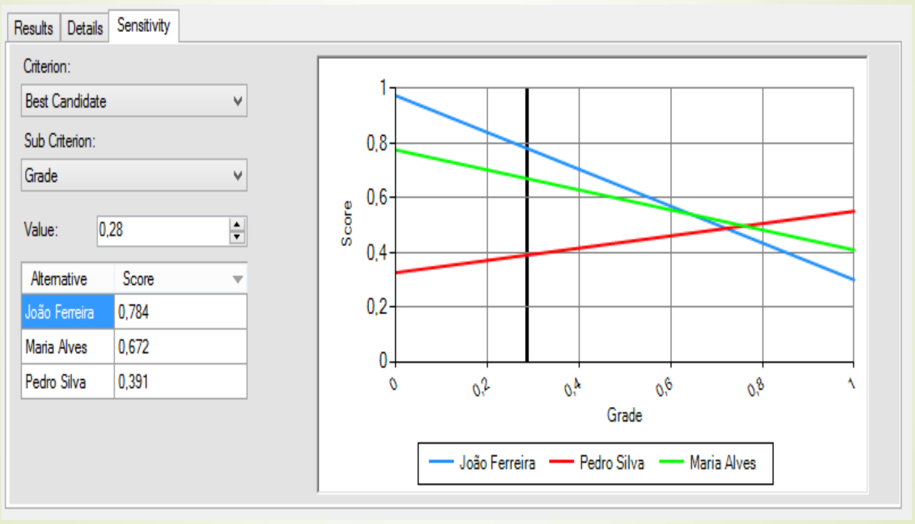
Alternative	Score
João Ferreira	0,784
Maria Alves	0,672
Pedro Silva	0,391

☑ Show criteria scores?

The detailed scores for each criterion are shown in order to easily identify what is weighting the most in the final results.

18

Results


Criterion: Best Candidate

Sub Criterion: Grade

Value: 0,28

Alternative	Score
João Ferreira	0,784
Maria Alves	0,672
Pedro Silva	0,391

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.

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.
- 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.

Conclusions and future research

- In the future, we hope to implement more methods with different scopes (like outranking 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.

Thank you

and

BeSmart2