Konradin Maier and Volker Stix

A Semi-Automated Approach for Structuring Multi Criteria Decision Problems

A Semi-Automated Approach for Structuring Multi Criteria Decision Problems

Konradin Maier[*,a], Volker Stix[a]

[a] *WU - Vienna University of Economics and Business, Institute for Information Business, Augasse 2-6, A 1090 Vienna, Austria*

Abstract

This article seeks to enhance multi criteria decision making by providing a scientific approach for decomposing and structuring decision problems. We propose a process, based on concept mapping, which integrates group creativity techniques, card sorting procedures, quantitative data analysis and algorithmic automatization to construct meaningful and complete hierarchies of criteria. The algorithmic aspect is covered by a newly proposed recursive cluster algorithm, which automatically generates hierarchies from card sorting data. Based on comparison with another basic algorithm and empirical engineered and real-case test data, we validate that our process efficiently produces reasonable hierarchies of descriptive elements like goal- or problem-criteria.

Key words: Problem structuring, Multiple criteria analysis, Concept Mapping, Hierarchical decomposition

1. Introduction

The first steps of multi criteria decision making (MCDM) are typically the decomposition and structuring of the decision problem at hand. The disaggregation enables the implementation of "divide-and-conquer" decision strategies, similar to expert decision making (Shanteau, 1988). The basic idea underlying problem decomposition is that smaller parts of the problem can be more

*Corresponding author. Tel.: +431313365203; fax: +43131336739.

Email addresses: `konradin.maier@wu.ac.at` (Konradin Maier), `volker.stix@wu.ac.at` (Volker Stix)

easily handled by human information processing capabilities than the entire problem at once. Furthermore, these smaller parts enable decision makers to structure the decision problem (typically a hierarchy composed of objectives and criteria) which increases their problem understanding and capacity to process information (Aschenbrenner et al., 1980). The decomposition and structuring of decision problems are of particular importance for the accuracy of the applied MCDM process (Von Winterfeldt, 1980; Saaty, 1990). If a decision maker evaluates only a subset of all relevant objectives and criteria, he cannot be sure that this evaluation identifies the most valuable alternative. That is also the meta-decision which objectives and criteria are relevant for the decision at hand may be biased which in turn leads to biased decisions (Pitz and Riedel, 1984). The structure itself has also a significant effect on the outcome of the decision process (Borcherding and Von Winterfeldt, 1988; Brugha, 1998). Although the initial activities of analytical decision making are usually considered as the most important, valuable and also difficult steps (Von Winterfeldt and Fasolo, 2009) the questions how to derive a complete list of criteria and how to reveal the latent structure of such a list does not receive much attention within the MCDM literature. At the same time, most methods for decomposing and structuring decision problems have been criticized for being “artistic” and for lacking methodical accuracy (Von Winterfeldt, 1980). Although some researchers expressed their optimism that decision structuring will advance from art to science quite a while ago (Borcherding and Von Winterfeldt, 1988), there has been only little progress towards this goal.

In this paper we propose a new approach to support the conceptualization and structuring of multi criteria decision problems to overcome this research gap. The proposed process integrates several techniques to balance the requirements of science (validity, reliability and objectivity) on the one hand and the demands of the practical field (efficiency, understandability and accuracy) on the other hand. We apply brainstorming techniques and structuring methods to cover the creative aspect of conceptualizing and structuring decision problems (Saaty, 2009), we utilize small group techniques to support group decisions and to mitigate biases resulting from the current perspective of a single decision maker (Pitz and Riedel, 1984; Saaty and Shih, 2009) and we apply quantitative data analysis and automatic data processing to ensure methodical accuracy and efficiency.

The remainder of this article is structured as follows. In the next section we shortly give an overview on current decision structuring processes and related techniques, which form the basis for the design of our approach introduced in Section 3. In Section 4 we present a new algorithm for the automatic construction of hierarchical representations of decision problems and compare it with another simple one. Section 5 demonstrates the validity and efficiency of the proposed algorithms using several test cases. Finally, we summarize the main findings, discuss further research issues and close with a concluding remark in Section 6.

2. State of the art

Conceptualization and structuring of decision problems are mainly creative tasks. Consequently, to overcome the challenge of conceptualizing and structuring decision problems, some researchers proposed applying "creativity techniques" to decision problems (e.g. Mackenzie et al., 2006; Saaty, 2009). "Creativity techniques" are methods which seek to foster divergent, creative and original thinking to solve a given problem. One of the most prominent methods is brainstorming, which is a group process focused on maximizing the quantity of ideas generated (Osborn, 1963). However, brainstorming has been criticized for being ineffective due to group effects like free riding, evaluation apprehension and production blocking. Electronic brainstorming, where the participants interact only via an electronic meeting system, tries to overcome these limitations by ensuring the anonymity of the participants and are therefore more productive than the original brainstorming technique (Dennis and Valacich, 1993).

In the context of MCDM, brainstorming and similar techniques are especially useful for identifying criteria relevant for the decision at hand. But they offer only little support for (hierarchically) structuring brainstormed criteria. To reveal the structure of a list of criteria, card sorting procedures can be used. Sorting procedures are typically based on a set of cards with related terms and participants who are asked to form clusters of cards based on their "relatedness" or "similarity". The sorting of a sample of participants serves as a measure of psychological distance which can be used in multivariate analysis like multi-dimensional scaling or clustering. While there are more accurate ways to measure psychological distances, the main advantage of card sorting is its economy, especially if the number of items is large (Rosenberg and Kim, 1975).

Recently, another method, interpretive structural modeling, gained popularity for structuring decision problems in the context of MCDM (e.g. Feng et al., 2010). Interpretive structural modeling builds on matrix representations and graph-theoretic methods to model problem domains. The modeling process starts with the identification of all problem elements followed by qualitative judgments about the strength of the relationships between these elements. This assessment is used to fill a reachability matrix which is then converted into a graph or tree with the help of graph-theoretic methods (Warfield, 1974).

Beside such "general purpose" problem structuring techniques, there exists some research on decomposing and structuring decision problems within the field of MCDM. For example (Saaty, 1990) proposed a qualitative top-down approach for the hierarchical structuring of decision problems. This process begins with the specification of an overall-objective, which is then iteratively decomposed until the level of criteria is reached. The disaggregation is guided by questions like "Which subgoals must be satisfied to fulfill this objective?". Several similar qualitative approaches have been suggested (e.g. Keeney and Raiffa, 1993), which differ mainly in details like direction of analysis (bottom-up versus top-down) or viewpoint (focus on objectives of the decision maker versus focus on qualitative differences of the alternatives). However, most of these qualitative approaches are rather vague and have been criticized for being "artistic" and for lacking methodical accuracy (Von Winterfeldt, 1980). Another problem related to such structuring processes is that there is no straightforward way to adapt them to yield a structure which reflects a group's thinking about a given decision problem. To overcome the latter limitation, it has been suggested to build distinct hierarchies for homogeneous subgroups of decision makers (Saaty and Shih, 2009). However, we argue that even within homogeneous subgroups individual decision makers can have different internal representations of a problem domain.

A somewhat outstanding structuring technique is concept mapping (CM), which combines brainstorming, card sorting procedures and multivariate data analysis (Trochim, 1989a). W. Trochim's CM is a structured conceptualization process which aims to organize a group's thinking about a domain of interest and to represent it in the form of pictures (concept maps). From the perspective of MCDM, concept mapping offers two desirable features: (1) it produces hierarchical clusters of similar concepts and therefore could be used to construct hierarchical representations of a decision problem (2) it is based on small groups of participants and thus supports group decisions and avoids biases due to the current perspective of a single decision maker. Furthermore, as opposed to many other conceptualization approaches, CM follows a rather quantitative than a qualitative paradigm and thus provides quite better ways to assess its methodical accuracy (Trochim, 1993, 1989b).

3. A process for semi-automated hierarchy generation

In this Section we introduce our process for creating goal-criteria hierarchies. The aim of the process is to support decision makers in constructing valid representations of a multi criteria decision problem at hand. Although we focus on structuring objectives, the processes can easily be adapted to build any hierarchical structure. The process design of our approach is based on CM, which we have adapted to meet the requirements of MCDM.

Step 1: Preparation. First of all, a qualified facilitator is selected who guides the process of hierarchy construction by providing knowledge, organizing workshops and resolving conflicts. Participants of the workshops are selected from decision makers, stakeholders, domain specific experts and/or consultants. The main activity is the development of a brainstorming focus, which will be used as a stimulus for the brainstorming session. The brainstorming focus should be a short statement, which describes the decision problem as well as the intended contributions of the participants. For example, a simple statement for an ERP software selection might be worded as: "Generate short statements which describe criteria relevant for the selection problem: Which ERP software should your organization use in the future?".

Step 2: Identification of criteria. The next process-step is the generation of a list of criteria, which should include all criteria relevant for the decision at hand along with short statements defining the meaning of each criterion. Typically, this list of criteria is generated within a brainstorming session. To avoid the aforementioned drawbacks of traditional brainstorming, we suggest the use of electronic brainstorming. However, other techniques like document analysis or interview procedures can be used as well. The resulting list of criteria has to be reviewed and edited to ensure its completeness and to avoid redundancies (e.g. from synonyms). Beside redundancy, other quality related aspects like understandability and measurability of the criteria can be assessed as well (Keeney and Gregory, 2005). These quality assurance activities are usually done by discussing the criteria one by one.

Step 3: Sorting of criteria. The finalized list from Step 2 forms the input for structuring the decision problem via an open card sorting procedure. Each criterion is written on a card together with a concise explanation. The participants are asked to sort these cards into piles according to a pre-defined sorting dimension (e.g. importance or semantic relatedness of criteria). The explicit specification of a sorting dimension is necessary to gain data of high quality (see Section 5.1). The choice of an appropriate sorting dimension can be adapted to the requirements of the respective MCDM problem. The sorting of criteria is restricted by the following rules: 1) each card can only be placed in one pile; 2) there have to be at least two piles; and 3) at least one pile need to have more than one card. Beside this "paper and pencil" approach, electronic card sorting can be applied which eases the sorting task as well as data collection.

Step 4: Automatic construction of a preliminary hierarchy. In this process step algorithmic data processing is used to efficiently construct a preliminary hierarchy of the decision problem which serves as a "good starting point" for the further refinement of the problem representation in step 5. A two-stage cluster algorithm for this process-step is described in detail in the next Section. The cluster analysis is performed on a distance matrix D which can be obtained from the card sorting data by calculating $d_{ij} = n - s_{ij}$, where n is the number of participants and s_{ij} is the count of participants sorting items i and j into the same pile.

Step 5: Finalization of the hierarchy. This process-step finalizes the hierarchy by refining the raw structure. The workshop participants are asked to inspect and discuss the hierarchy to identify potential cluster names and inappropriate assignments of criteria to clusters. Special emphasis is placed on examining unbalanced subtrees of the hierarchy and "dummy nodes" since both are indicators of possible inconsistencies like missing criteria, different levels of granularity or similar issues (see Section 4.3 for more details on dummy nodes, unbalanced subtrees and their potential interpretations). The output of this process-step is the final hierarchy, which can be utilized in the subsequent MCDM process-steps.

4. Algorithmic construction of hierarchies based on card sorting data

The algorithm we present in this section constructs a hierarchy considering some constraints using the output of a hierarchical cluster analysis. The result of such a cluster algorithm can be illustrated using a dendrogram. Figure 1 (a) shows an example of a dendrogram with 8 objects. Whereas equation (1) formulates the the very same clustering process. Let $C(t)$ be the set of all clusters with respect to the given threshold-distance t and let C_i denote the extraction

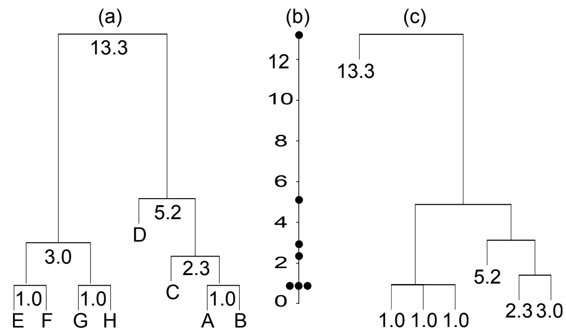

Figure 1: (a) A simple dendrogram of eight objects (A-H); (b) the respective merge threshold-values; (c) The clustering of the clusters (merge values).

of the i-th element from the set C. With increasing threshold t the algorithm always joins exactly two clusters from previous sets. Starting with $t = 0$, there are 8 clusters composed of all elements, with e.g. $t = 1.5$ there are 5 clusters, and so on, ending with e.g. $t = 15$ in one cluster containing all objects. Clearly this result is achieved always as $t \to \infty$. Let m be the vector of all these merge threshold-values in descending order and let m_i denote the i-th element. Thus $m = (13.3, 5.2, 3, 2.3, 1, 1, 1)^T$ for this example as shown in Figure 1 (b). It can be seen that there may exist multiple mergers having the same threshold-value (e.g. here for $t = 1$).

$$
\begin{aligned}
C(0) &= \{E, F, G, H, D, C, A, B\} \\
C(1.5) &= \{\{E, F\}, \{G, H\}, D, C, \{A, B\}\} \\
&\vdots \\
C(8) &= \{\{\{E, F\}, \{G, H\}\}, \{D, \{C, \{A, B\}\}\}\} \\
C(15) &= \{\{\{\{E, F\}, \{G, H\}\}, \{D, \{C, \{A, B\}\}\}\}\} = C(m_1)
\end{aligned} \tag{1}
$$

In many applications only the elements and not their history are important. This can be achieved by flattening $C(\cdot)$ as shown exemplarily in (2).

$$C(15) \quad \hat{=} \quad \{\{E, F, G, H, D, C, A, B\}\} \tag{2}$$

The exact values of m, where two clusters are merged, depend on the clustering method. We opted for Ward's method (Ward, 1963) because it seems to be the natural choice to build homogeneous hierarchies due to its strategy to locally minimize in-group variance for each merge of two clusters (here variance represents the threshold). Furthermore, Ward's algorithm does not suffer from the "chaining" problem of other hierarchical clustering methods. However, any hierarchical clustering method can be used to determine the values of m.

In classical cluster analysis, one of the main problems is to determine an adequate value for t namely to find the appropriate number of clusters. In the case of hierarchy construction the problem is more complex: we need to determine (i) how many levels the final hierarchy should have and (ii) how many siblings each level should have. Most researcher in MCDM suggest to keep the cluster size (number of siblings) rather low. Usually, this is justified by referencing to the limitations of human information processing. The results of research in human cognition indicate that human short term memory can store 7 ± 2 chunks (Miller, 1956), newer research suggests that our memory span is even smaller (Cowan, 2001). Additionally, researchers who favor relative measurement of criteria weights argue that only similar objects can be meaningfully compared (Saaty, 1990). At the same time, a large number of small clusters is not a major concern because the number of criteria, which can be arranged in a hierarchy, increases by C_{max}^h, where C_{max} is the cluster size and where h is the height of the hierarchy. That is, even a hierarchy with small clusters needs only a few levels to arrange dozens or even hundreds of criteria. For example, a hierarchy with cluster size 7 and height 3 may contain up to 343 leaves. In summary, these arguments support the notion to restrict cluster sizes to 6 (± 2). However, hierarchies with larger clusters are possible if necessary to adequately represent the decision problem. In the following, we present two algorithms for the construction of hierarchies from card sorting data which are based on the choice of a desired cluster size and on a clustering algorithm.

4.1. A structure based algorithm

This simple algorithm constructs a hierarchy purely based on the merging-sequence of the clustering process, i.e. the merge of two clusters to a new cluster. Let C_a and C_b be the two respective clusters, then the algorithm joins those clusters as long as the size $|C_a \cup C_b|$ does not exceed the maximum cluster size C_{max}. If $|C_a \cup C_b| > C_{max}$, the algorithm inserts a level into the hierarchy and makes C_a and C_b siblings, i.e. inserting $\{C_a, C_b\}$, forming a new cluster on the next higher level. This leads to a hierarchical structure where no cluster exceeds the specified cluster size C_{max}. The placement of the hierarchy levels, however, is only based on the sequencing information and on the desired cluster size, the distances between clusters remain unconsidered.

Algorithm 1 $StructureCollapse(C)$ – a recursive approach

1: **if** $C =$ isElement **then**
2:     **return** C
3: **end if**
4: $C_a \leftarrow StructureCollapse(C_1)$
5: $C_b \leftarrow StructureCollapse(C_2)$
6: **if** $|C_a \cup C_b| > C_{max}$ **then**
7:     **return** $\{C_a, C_b\}$
8: **else**
9:     **return** $C_a \cup C_b$
10: **end if**

Algorithm 1 shows this approach using the recursive function "Structure-Collapse". It recursively collapses set-structures to sets with the maximal size of C_{max}. We start the algorithm with the set of the last remaining two clusters: $StructureCollapse(C(m_2))$. Lines 1–3 represent the recursive escape condition. If C is not a set but an element it clearly fulfills the size condition and is returned. In all other cases a cluster is always constructed out of exactly two ancestors.

Line 6 checks whether the union of the already collapsed ancestors (lines 4–5) exceeds C_{max}. If so, they build a new cluster (level), which is returned. If they do not exceed the size, they are joined and returned as the result. We shall discuss this simple algorithm together with our newly proposed one at the end of this section.

4.2. A distance based algorithm

We propose the following algorithm to construct hierarchies from card sorting data, which is based on the idea to place groups of similar distant items within one level of the hierarchy. To achieve this, we perform a clustering of clusters as follows. The ordinate in Figure 1 (b), represents the respective threshold-values m at which clusters are merged. These values can be interpreted as similarity between mergers. So our algorithm clusters these merge-values in order to find levels which have a similar level of semantic abstraction. Let $C'(t)$ be the recursive set representing this clustering result (see Figure 1 (c)). The objects are now identified through numbers instead of letters since they correspond with the merge-values of the original objects. The dendrogram illustrates all possible clusters. For example $C'(m'_2)\hat{=}\{\{13.3\},\{5.2,3.0,2.3,1.0,1.0,1.0\}\}$ equals two clusters (of mergers) which correspond to the smallest hierarchy of two levels (excluding the root-node level zero), flattening sets above the threshold level 5.2 into hierarchy level one and sets up to 5.2 into level two. The result $C'(m'_3)\hat{=}\{\{13.3\},\{5.2,3.0,2.3\}\ ,\{1.0,1.0,1.0\}\}$ equals three clusters (of mergers) which correspond with three levels in the hierarchy: level one in $(5.2,\infty)$, level two in $(1.0,5.2]$ and level three in $(0,1.0]$. Let $l(C)$ be the function, which returns the level of the hierarchy into which the set C belongs, where $l=0$ represents the top-level (root-level), $l=1$ the next sub level, and so on. This level is derived from the respective results $C'(m'_2),\dots,C'(m'_i)$ (where i represents the number of hierarchy-levels). Again, an optimal threshold-value needs to be found. We decided to look at all possible clusters of clusters $C'(m'_2),\dots,C'(m'_i)$. From these alternatives of hierarchies, the algorithm chooses the hierarchy with the minimal absolute deviation from the desired cluster size C_{max} (considering all levels greater zero).

Algorithm 2 *DistanceCollapse*(C) – a recursive approach

1: **if** C = isElement **then**
2:     **return** C
3: **end if**
4: $C_a \leftarrow DistanceCollapse(C_1)$
5: $C_b \leftarrow DistanceCollapse(C_2)$
6: **for** $i = 2$ **to** $l(C_1) - l(C)$ **do**
7:     $C_a \leftarrow \{C_a\}$
8: **end for**
9: **for** $i = 2$ **to** $l(C_2) - l(C)$ **do**
10:     $C_b \leftarrow \{C_b\}$
11: **end for**
12: **if** $l(C) < \max(l(C_1), l(C_2))$ **then**
13:     **return** $\{C_a, C_b\}$
14: **else**
15:     **return** $C_a \cup C_b$
16: **end if**

Algorithm 2 covers this approach. It looks similar to the structure based algorithm (SBA), with some small differences. Again lines 1–3 represent the recursive escape condition. One of the differences is the level assignment function $l(\cdot)$ introduced above which builds on the results of the second clustering. The second difference concerns dummy nodes. In contrast to the SBA), our distance based algorithm (DBA) knows (due to the second clustering) to which level each cluster belongs. So it might happen that there are missing parents on some levels. For example we merge two clusters A and B into AB, but that $l(A) = l(B) = 3$ whereas $l(AB) = 1$, thus there is no explicit parent node on level two. In such cases we insert dummy nodes to maintain the levels, which is done in lines 6–11 (for illustration see also Figure 4 in the next section). Whenever the merged set C belongs to a hierarchy level strict smaller than at least one of the mergers, then C_a and C_b represent siblings since the join of both reaches into the next level (line 13) otherwise the two mergers C_a and C_b are joined, since they belong to the same level (line 15).

4.3. Comparison of the algorithms

Our DBA considers more information than the SBA and thus acts more "intelligent". This allows our DBA to construct hierarchies with more homogeneous levels than the SBA. This is explicated in Figure 2, where the SBA fails to form the most reasonable hierarchy. The upper part of the Figure shows the behaviors of both algorithms and the respective hierarchies for $C_{max} = 4$. The SBA (thin boxes) is not able to form a common cluster for the elements $E1$ through $E5$ because this would exceed the specified cluster size. Thus, $E5$ is placed as a singleton on the first level of the hierarchy which is therefore rather inhomogeneous. As shown in the lower part of this figure, a cluster limit of five would solve this problem. This size, however, leads to a similar problem in another branch of the dendrogram. The elements $E6$ through $E10$ are placed in one cluster although $E10$ does not fit well into this cluster which leads to a inhomogeneous cluster on the second level of the hierarchy. The DBA does not suffer from such problems. In both cases (cluster size four and five), the DBA identifies the most reasonable cutting line and thus leads to homogeneous hierarchies. Another advantage of our DBA is that it reacts less sensitive to changes of the cluster size specified by the workshop facilitator since the placement of the cutting lines is based on distance information only. The desired cluster size effects solely the number of levels formed by the DBA. Thus, the analyst's influence on the outcome of the algorithm is reduced. The latter advantage, however, comes with the disadvantage that the DBA does not necessarily respond to the specified cluster size. While the resulting clusters approach the specified cluster size, some of the clusters will exceed C_{max} (see Figure 2). We argue that homogeneity is more important than strict upper limits of cluster sizes due to three reasons: Firstly, inhomogeneity can render hierarchies completely useless since, instead of being supportive, such hierarchies hamper human information processing and measurement of preferences (Brugha, 1998). Secondly, there are ways to deal with disadvantages of larger clusters in MCDM settings (Bernroider et al., 2010). Finally, homogeneity is a precondition to reasonably interpret hierarchies. For instance, think of a well balanced tree, where each branch has the same depth, with the exception of one leaf which is directly assigned to the root (like $E10$ in Figure 2). If this hierarchy has been generated with our DBA, there could be a meaningful reason for this unbalanced subtree:

- Granularity: There are two possible interpretations: Either the respective criterion has not been broken up into the same level of granularity as the other criteria or there are missing abstract concepts (parent-nodes), depending whether this criterion is a semantic high- or low-level concept.
- Irrelevance: The respective criterion does not fit to any other criterion because it is irrelevant for the decision at hand but has mistakenly passed the review of the brainstormed criteria.
- Dissent: The workshop participants are either dissent in their interpretation of the respective criterion or in their judgments regarding its group membership.
- Dissimilarity: None of the other reasons apply. The criterion is simply not similar to any other criterion and not required in more detailed or abstracted views. Thus the unbalanced subtree is justified.

Therefore, unbalanced subtrees as well as dummy nodes can serve as stimuli in process step 5 to discuss the hierarchy, its validity and whether revisions are necessary or not, which enhances the quality of the final hierarchy further. In contrast, the SBA does not allow such interpretations since unbalanced subtrees occur arbitrarily which makes the resulting hierarchies hard to interpret.

5. Testing of the proposed algorithm

In this Section we report on test cases to evaluate the main contribution of this article, i.e. the (automatic) construction of preliminary hierarchies. We

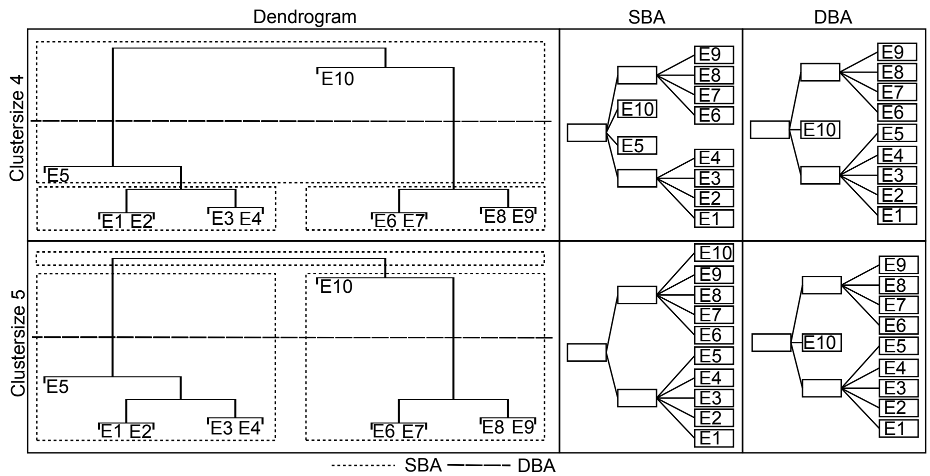

Figure 2: Comparison of the SBA and the DBA based on a hypothetical data set.

focus on algorithmic validity because processes similar to our overall-approach (e.g. concept mapping) as well as single process steps (e.g. card sorting) have already been extensively tested by other researchers (see Section 2).

5.1. Pre-test

To test whether card sorting procedures are useful for structuring decision problems as well as for planing and parametrization we performed a pre-test with 16 students who were asked to complete a paper based card sorting task. We selected an ordinary decision problem which did not require expert knowledge: "Select a job offer from several alternatives". Our research group identified 30 criteria for this decision problem using brainstorming. The set was reduced to 26 criteria by eliminating redundant statements. These criteria, together with a short explanation of each, were written on cards. The students were briefed, asked to read through the written instructions and to sort the statements into piles "in a way which makes sense to them". The sorting of the criteria was followed by unstructured interviews to gather the students' opinions on the card sorting procedure as well as to identify the applied sorting strategies.

The interviews showed that the students used two sorting dimensions: importance and semantic relatedness of the criteria. Three students reported that they did not know how to decide on one of these two dimensions. Some of the participants even mixed these dimensions by separating important from unimportant criteria and sorting each of them semantically. Some put the remaining cards in a "not relevant"-group or into two groups of medium and low importance. Other participants piled semantic related criteria together, and labeled cards which did not fit their categorization scheme as unimportant. Eleven participants reported that semantic relatedness was their main sorting dimension, while the others chose importance as their sorting criterion. As a consequence, the resulting clusters were difficult to interpret. The general feedback on the card sorting procedure was favorable. All students felt comfortable with the sorting task and the card sorting procedure turned out to be time efficient, since no student required more than ten minutes (seven in average). From these primary results we found that a card sorting procedure is adequate for structuring decision problems but that a sorting dimension needs to be explicitly and precisely specified afore to yield meaningful clusters.

5.2. Settings of the test cases

To test our algorithms we implemented the process described in Section 3. As our tests were focused on algorithmic validity, we decided to perform the time consuming process steps one and two, i.e. preparation and identification of criteria (brainstorming), within our research team. The resulting concepts (cards) were given to volunteering students for sorting (step three: structuring). To substitute real decision makers with students is adequate in this context because the data gathered was not used to make inferences about decision makers but to evaluate whether the proposed algorithms are able to build meaningful hierarchies out of card sorting data. Furthermore, the ability to recognize groups of related items is not unique to real decision makers but a fundamental process of human cognition. After the card sorting procedure, preliminary hierarchies were constructed out of sorting data using our algorithm (step four). Finally, the results were discussed again by our research team (step five: finalization).

test case	problem domain	criteria generation	#criteria	#participants	avg. time
1	Flat selection	Brainstorming	33	26	9 min
2	Job offers	Brainstorming	26	30	11 min
3	Transport infrastructure	Document analysis	19	18	10 min
4	Fuel selection	Document analysis	17	16	9 min

Table 1: Overview of empirical tests.

For the card sorting procedure we briefed the participants to form piles of semantically related criteria. To ease data gathering, the card sorting was computer-aided. Each test included a short (about five minutes) verbal and a written instruction explaining the purpose of the test, the sorting procedure and the web-based card sorting tool. Overall, four tests with different criteria-sets were arranged. Table 1 gives an overview of these test cases. The first two cases were based on brainstormed criteria-sets (a flat-selection problem and the job-set from the pre-test). Again, neither of these two required expert knowledge. The criteria-sets of the tests three and four were based on MCDM studies reported in the literature (Dodgson et al., 2009; Winebrake and Creswick, 2003). The rationale for this approach was to get a glimpse of the validity of our structuring technique by comparing the criteria hierarchies reported in the literature with the hierarchies build by our algorithms. Furthermore, this allowed us to test the card sorting procedure in a "hostile" environment where the participants face unfamiliar decision problems. From the vast amount of MCDM literature we selected two decision problems (appraisal of transport projects and evaluation of fueling systems for transportation) based on the following considerations: (1) the MCDM problem should offer a reasonable number of criteria, (2) the criteria itself should be comprehensible for non-experts and (3) the publication should offer a hierarchical structuring of the criteria. The next section outlines the major empirical results.

algo	S	Δ	short	level 1		level 2		level 3	
		depth	branches	inter	inner	inter	inner	inter	inner
SBA	4	1.48	67%	45.63	65.00	73.33	88.00	61.67	88.00
DBA	4	0.48	3%	9.06	34.00	42.23	39.00	46.00	n.a.
SBA	5	1.83	55%	n.a.	38.00	49.50	81.00	85.33	88.00
DBA	5	0.83	3%	9.06	34.00	42.23	39.00	46.00	n.a.
SBA	6	3.05	73%	n.a.	38.00	77.45	96.00	65.76	96.00
DBA	6	1.05	3%	9.06	34.00	42.23	39.00	46.00	n.a.
SBA	7	2.20	73%	n.a.	38.00	77.45	96.00	83.00	88.00
DBA	7	0.20	0%	9.06	34.00	88.00	n.a.	n.a.	n.a.

Table 2: Numerical results of test cases one (flat selection).

5.3. Results of the test cases

All four test cases were rendered using the structure based algorithm (SBA) and the proposed distance based algorithm (DBA) described in Section 4. Additionally, we built four sub cases for each case, targeting on different (maximal) cluster sizes $S = 4, \ldots, 7$. Figure 4 and Figure 5 show examples of such a hierarchy for size=4. Table 2 gives some specific results of the respective algorithms for test case one. The supplementary material provides an extended version of Table 2, which includes more measures as well as the other test cases.

The first two columns of Table 2 reflect basic structural properties of the hierarchies. Column Δ depth shows the absolute difference between the hierarchy's depth and the theoretical depth ($\log_S n$, where n is the number of criteria). In 14 out of 16 cases, the DBA approaches the theoretical maximal depth closer than the SBA. This indicates that the latter one constructs results with unnecessary many levels which eventually could hamper the use and interpretation of the hierarchy. The column % short branches gives the percentage of criteria which are not on the lowest level of the hierarchy. Such short branches have missing intermediate level(s) (e.g. see Figure 5 (b)). Our DBA places nearly all criteria on the lowest level, maintaining intermediate hierarchic levels, while the SBA fails to build a common level for atomic elements, which suggests that it is

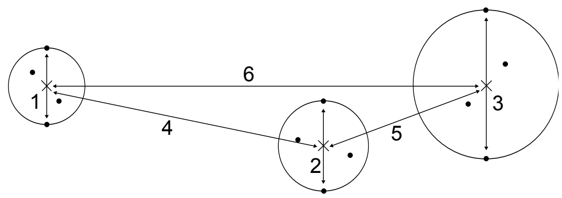

Figure 3: Inner and inter cluster homogeneity. The former is calculated as the difference of the largest (3) and the smallest (1) cluster diameter. The latter is calculated as the difference of the largest (6) and the smallest (5) distance of cluster centers.

not able to identify levels of elements which are of similar semantical distance.

The last 6 columns of Table 2 report on two different measures of partial homogeneity for the first 3 levels of the hierarchies. The first measure captures inter-cluster homogeneity as the difference of the largest and the smallest distance of siblings (clusters assigned to the same parent node), where this inter-cluster distance $d(a, b)$ is measured as the average pairwise distance from any element of cluster a to any element of cluster b. We interpret this measure as the degree to which the children assigned to a parent node are semantically equidistant. Table 2 lists the largest inter-cluster homogeneity per hierarchical level. The smaller this value the more homogeneous are the inter-clusters distances for that respective level. The second indicator, inner-cluster homogeneity, is the difference of the largest and the smallest cluster diameter within each level of the hierarchy. We interpret this measure as the degree to which a level contains clusters with the same level of semantic abstraction. Again, the smaller the difference, the more homogeneous is the respective level. Figure 3 illustrates inter and inner homogeneity. Short branches, which can not be assigned distinctly to levels, where resolved using a top-down assignment approach (starting from the root=top). As can be seen from Table 2, the DBA results in levels with higher inter- and inner-cluster homogeneity than the SBA. The only exception is inter-cluster homogeneity of level 2 for cluster size 7, where the SBA performs better than the DBA.

5.4. Face validity

To assess the hierarchies on a semantic level, we follow a “face validity” approach by visually examining the raw structures resulting from the proposed algorithms. Furthermore, we compare the raw structures of test cases three and four with hierarchies proposed in the literature. In the following, we present the results of the two test cases with a cluster size $S = 4$, however, we want to note that the other test cases and larger cluster sizes support the results reported here.

Figure 4 shows the results of test case three for both algorithms, as well as the corresponding hierarchy taken from the literature. A visual examination shows that both algorithms form clusters of semantically related criteria. However, it could be difficult to find reasonable identifiers for some of the clusters (e.g. for the cluster consisting of “heritage”, “landscape” and “regeneration”). This demonstrates that a manual revision of raw structures is usually necessary to derive a final hierarchy (see step 5 in Section 3). A comparison of the SBA’s output with the original structure shows that the SBA reconstructs the cost-benefit structure of the original hierarchy. However, as the desired cluster size has been set to four, it requires an additional sub cluster to merge all cost-related criteria into one cluster. With the criteria “water”, “noise”, “biodiversity” and “air quality” within one cluster, also the “environment” cluster is reconstructed quite well. Compared to the original hierarchy, the other clusters seem to be a mixture of the remaining criteria, however, these clusters are semantically justifiable. In comparison, the DBA does not reconstruct the cost-benefit structure on the first level of the hierarchy. While there is a “cost” cluster, the benefits are broken up into a cluster directly related to transportation issues and a cluster related to economic and environmental issues. As the DBA does not strictly adhere to the specified cluster size, it is able to perfectly reconstruct the “cost” cluster of the original hierarchy, the other clusters are identical with the clusters formed by the SBA. Test case three also demonstrates the DBA’s insertion of dummy-clusters. The “cost” cluster is on a similar level of semantic abstraction like the clusters on the second level of the hierarchy but the “cost” branch of

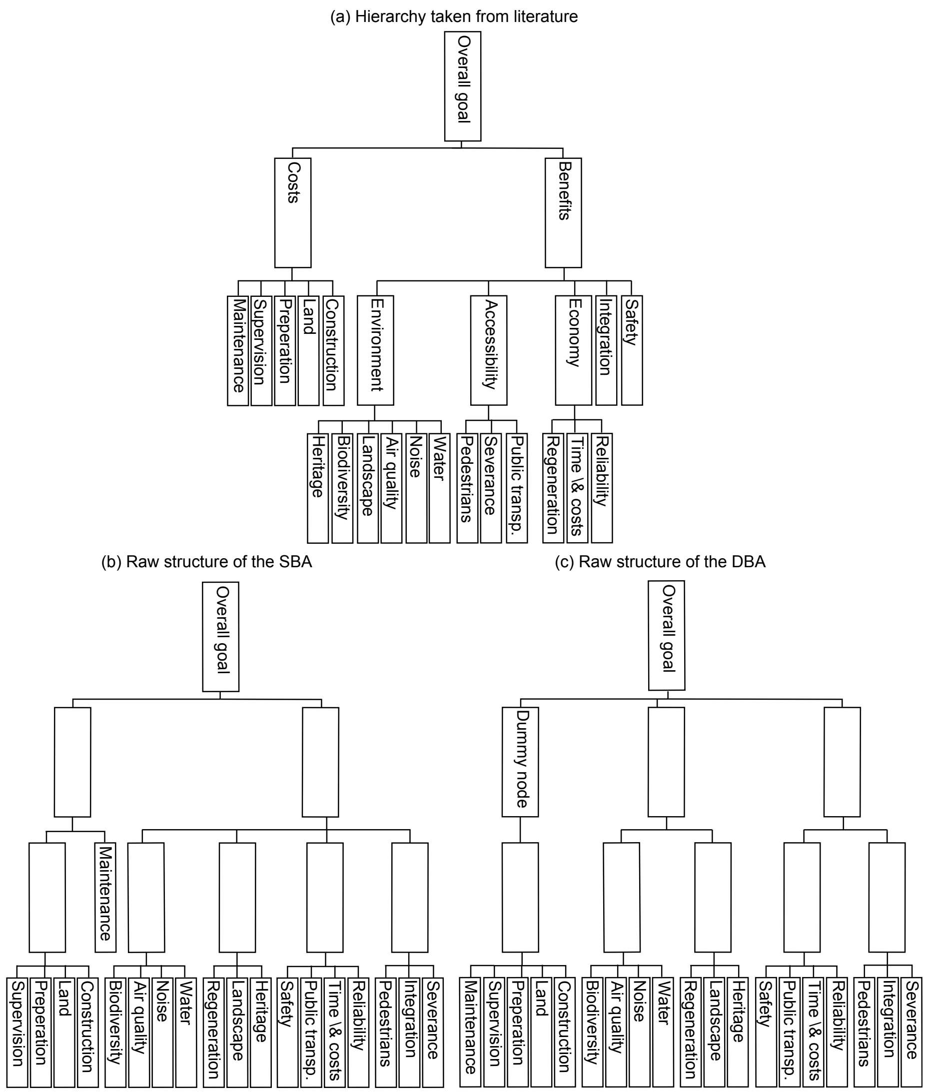

Figure 4: Hierarchies of test case three for cluster size four.

the hierarchy has not the same height as the other branches. To form levels of similar abstract criteria a dummy cluster is inserted. Interestingly, this dummy node flags the unbalanced subtree of the hierarchy taken from literature, which could indicate that the cost-benefit structure on the first level of the original hierarchy is not appropriate. This and other interpretations of the dummy node would be discussed in process step 5 where dummy nodes serve as stimuli to evaluate whether the resulting hierarchy is complete and valid or if a revision of the hierarchy is necessary.

Figure 5 shows the results of test case four for both algorithms, as well as the corresponding hierarchy taken from the literature. Again, the resulting hierarchies are to a large extent semantically reasonable. Also the benchmark hierarchy is partly reconstructed. The DBA exactly rebuilds the clusters "vehicle operation" and "economics", and there is also a cluster similar to the 'environment' goal of the original hierarchy. However, the DBA forms only four instead of five clusters on the first level of the hierarchy and thus there are some "mixed" clusters. The SBA forms identical clusters for vehicle operation and for environmental issues but the cluster "vehicle operation" is split into two clusters. The SBA builds a hierarchy with five levels, while the DBA requires only three levels to arrange all criteria. Furthermore, the branches of the hierarchy formed by the SBA have different heights, while all branches built by the DBA have the same height. Additionally, the SBA forms some counter-intuitive clusters. For example the parent node of "sustainability" also covers the criteria related to vehicle operation. The semantics of this cluster is unclear and thus it is difficult to find a meaningful name for this cluster. In sum, the hierarchy of the SBA seems to be more complex and the criteria are less clearly arranged while the hierarchy of the DBA is more balanced and more easily interpretable. The reasons for this are that the SBA strictly adheres to the cluster limit and considers fewer information than the DBA and therefore it is not able to identify levels of criteria which are on the same level of semantic abstraction.

We did not expect that either of the two algorithms is able to perfectly reconstruct the hierarchies taken from literature. However, we found that the DBA

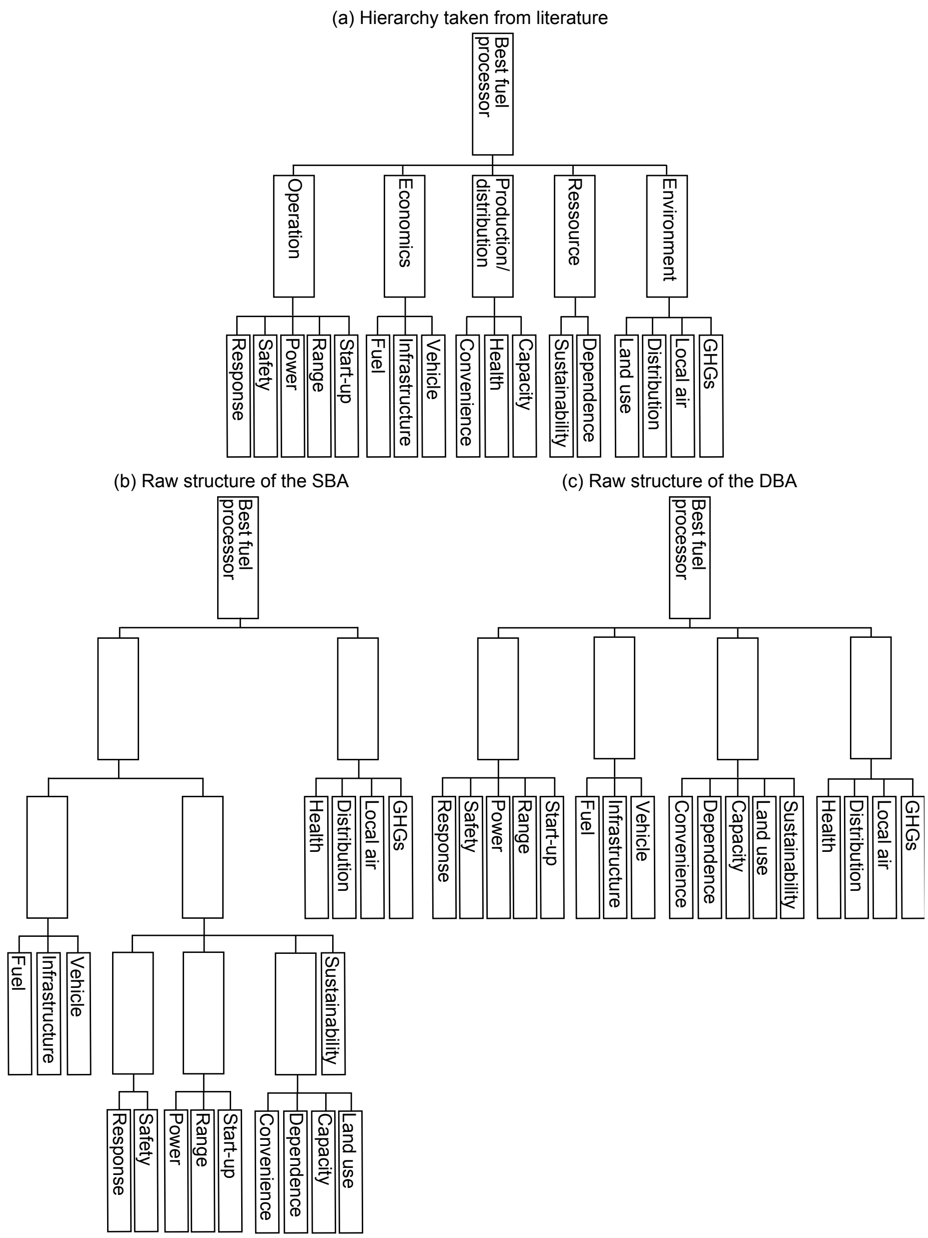

Figure 5: Hierarchies of test case four for cluster size four.

rebuild the hierarchies well. Beside the large number of possible hierarchies we identified three plausible explanations why the results did not exactly match the original structures: (1) inaccuracies introduced by translating the criteria from English to German (2) inconsistent interpretations of criteria due to participants who ignored the descriptions of the criteria and (3) the participants who lacked domain knowledge. However, in a practical setting, these potential problems are of little relevance because the brainstorming session ensures that the participants share a common interpretation of the criteria and the careful selection of participants guarantees that they have an adequate domain expertise.

6. Conclusions and further research

The main contribution of this paper is the development of a new algorithm for the automatic construction of hierarchies from card sorting data. The SBA analyzes the structure of dendrograms, that is the sequence of mergers of hierarchical clustering procedures, to derive hierarchies, while the DBA is based on the distance information of dendrograms, that is the height of each merge, to construct hierarchies. Since the DBA processes more relevant information, we hypothesized that our DBA will construct more reasonable hierarchies compared to the SBA. We conducted four test cases which confirmed this assumption. In the test cases our DBA formed hierarchies which are more balanced and more easily interpretable than the SBA. Furthermore, the DBA flags areas which require more discussion of the participants by inserting dummy nodes and thus facilitates the construction of high-quality hierarchies. Since the SBA is a rather naïve algorithm we should be careful when examining the results reported here. Nevertheless, we think that the DBA proved to be a useful and valid instrument to support the construction of hierarchies within the context of MCDM.

Regarding the efficiency of our overall-process, the preparation of the card sorting procedure took us about two hours (step 2), the sorting lasted about 10 minutes (step 3) and the automated construction of a preliminary hierarchy requires a few seconds (step 4). In a practical setting, we would also have to reserve some time for step one, preparing the workshop, and step five, finalization of the hierarchy. In sum, we expect that a workshop (step two through five) takes at least four hours, which is a reasonable amount of time for preparing an important decision. However, we also found that media discontinuities within the process significantly increase these time requirements. Thus, to take full advantage of the automatized procedure the use of computer-aided card sorting and brainstorming is strongly recommended. Despite of such measures it might be necessary to build the hierarchy in a more traditional way if the decision maker is not able or not willing to invest some time in making his decision.

Besides efficiency, the effectiveness of the proposed process is also of interest. A factor potentially limiting the effectiveness of our approach is the use of clustering procedures. These are often considered as "soft techniques" because there is no agreed on procedure to determine the number of clusters and because cluster analysis always generates clusters, whether they exist in reality or represent noise only. Nevertheless, our test cases show that the proposed clustering procedure is a useful technique for constructing meaningful and valid representations of MCDM problems. Regardless of pro and cons of clustering, our process offers two major advantages. Firstly, it is well structured and should thereby guide the decision maker to an adequate representation of his decision problem. Secondly, our process encourages extensive reflection and discussion on the decision at hand and thus should increase the decision maker's understanding about the given problem. In sum, we conclude that our approach for structuring MCDM problems is effective.

Further research could address the question whether our process is more valid than more traditional approaches for structuring MCDM problems. A simple method to answer this question would be to generate several hierarchies using different techniques and then to ask decision makers to identify the structure which reflects the decision problem at best. Another method, adapted from Trochim (1989b), would be to generate hierarchies and then to randomly permute or replace some of the criteria. Only if decision makers are able to identify the original, non-randomized hierarchy then one could speak of a valid approach to structure MCDM problems. The percentage of decision makers able to identify the non-randomized hierarchy could serve as a measure to compare the validity of different structuring techniques. These and similar methods to validate structuring techniques offer several interesting areas for research, which might further advance the structuring of MCDM problems from art to science.

Acknowledgment

The research was funded by the Austrian Science Fund (FWF): TRP111-G11.

References

Aschenbrenner, K., Jaus, D., Villani, C., 1980. Hierarchical goal structuring and pupils' job choices: Testing a decision aid in the field. Acta Psychologica 45, 35–49.

Bernroider, E., Maier, K., Stix, V., 2010. Incomplete Information within Relative Pairwise Comparisons as Utilized by the AHP, in: Business Information Systems Workshops, Springer. pp. 39–50.

Borcherding, K., Von Winterfeldt, D., 1988. The effect of varying value trees on multiattribute evaluations. Acta Psychologica 68, 153–170.

Brugha, C., 1998. Structuring and weighting criteria in multi criteria decision making (MCDM). Lecture notes in economics and mathematical systems , 229–242.

Cowan, N., 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and brain sciences 24, 87–114.

Dennis, A., Valacich, J., 1993. Computer brainstorms: More heads are better than one. Journal of Applied Psychology 78, 531.

Dodgson, J., Spackman, M., Pearman, A., Phillips, L., 2009. Multi-criteria analysis: a manual. Department for Communities and Local Government. London.

Feng, C., Wu, P., Chia, K., 2010. A hybrid fuzzy integral decision-making model for locating manufacturing centers in china: A case study. European Journal of Operational Research 200, 63–73.

Keeney, R., Gregory, R., 2005. Selecting attributes to measure the achievement of objectives. Operations Research 53, 1–11.

Keeney, R., Raiffa, H., 1993. Decisions with multiple objectives: Preferences and value tradeoffs. Cambridge University Press.

Mackenzie, A., Pidd, M., Rooksby, J., Sommerville, I., Warren, I., Westcombe, M., 2006. Wisdom, decision support and paradigms of decision making. European Journal of Operational Research 170, 156–171.

Miller, G., 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological review 63, 81.

Osborn, A., 1963. Applied imagination. Charles Scribner's Sons, New York.

Pitz, G., Riedel, S., 1984. The content and structure of value tree representations. Acta Psychologica 56, 59–70.

Rosenberg, S., Kim, M., 1975. The method of sorting as a data-gathering procedure in multivariate research. Multivariate Behavioral Research 10, 489–502.

Saaty, T., 1990. How to make a decision: the analytic hierarchy process. European journal of operational research 48, 9–26.

Saaty, T., 2009. Decisions, Structure, and Natural Law. International Journal of the Analytic Hierarchy Process 1.

Saaty, T., Shih, H., 2009. Structures in decision making: On the subjective geometry of hierarchies and networks. European Journal of Operational Research 199, 867–872.

Shanteau, J., 1988. Psychological characteristics and strategies of expert decision makers. Acta Psychologica 68, 203–215.

Trochim, W., 1989a. An introduction to concept mapping for planning and evaluation. Evaluation and program planning 12, 1–16.

Trochim, W., 1989b. Concept mapping:: Soft science or hard art? Evaluation and Program Planning 12, 87–110.

Trochim, W., 1993. Reliability of concept mapping. Paper presented at: Annual Conference of the American Evaluation Association.

Von Winterfeldt, D., 1980. Structuring decision problems for decision analysis. Acta Psychologica 45, 71–93.

Von Winterfeldt, D., Fasolo, B., 2009. Structuring decision problems: A case study and reflections for practitioners. European journal of operational research 199, 857–866.

Ward, J., 1963. Hierarchical grouping to optimize an objective function. Journal of the American statistical association 58, 236–244.

Warfield, J., 1974. Developing interconnection matrices in structural modeling. Systems, Man and Cybernetics, IEEE Transactions on 4, 81–87.

Winebrake, J., Creswick, B., 2003. The future of hydrogen fueling systems for transportation: An application of perspective-based scenario analysis using the analytic hierarchy process. Technological Forecasting and Social Change 70, 359–384.