

Implementation of an Online Software Tool for the Analytic Hierarchy Process – Challenges and Practical Experiences

Dr. Klaus D. Goepel^a

^aBPMSG

Business Performance Management Singapore
2 Bedok Reservoir View #17-02, Singapore 479232
E-mail drklaus@bpmsg.com

Abstract

The analytic hierarchy process (AHP) remains a popular multi-criteria decision method. The author has implemented a free, web-based AHP online system with noteworthy features, allowing for the detailed analysis of decision problems. Beside standard functions like flexible decision hierarchies, support to improve inconsistent judgments, alternative evaluation and sensitivity analysis, the software can handle group inputs, calculate group consensus based on Shannon α and β -entropy, and estimate weight uncertainties based on randomized small variations of input judgments. In addition different AHP judgment scales can be applied *a posteriori*, and alternative evaluation can be done using the weighted sum (WSM) or weighted product model (WPM). This flexibility opens up opportunities to study decision projects under various parameters. The author's intention was to provide a complete and free software tool for educational and research purposes, where calculations and algorithms are well documented and all input data and results can be exported in an open format for further processing or presentation. The article describes the basic concept and structure of the software and the underlying mathematical algorithms and methods. Challenges and practical experiences during the implementation, validation and productive phase of the software are highlighted.

Keywords

Multi-criteria decision making, Analytic hierarchy process, AHP Software, AHP online system, AHP Group decision making.

1. Introduction

The analytic hierarchy process (AHP) was developed by T. L. Saaty in the 1980 (Saaty, 1980) and remains a widespread multi-criteria decision method (MCDM). Based on pairwise comparison inputs, weights are calculated by finding the dominant right eigenvector (EV) of a positive reciprocal decision matrix. Calculations for simple decision problems can be done with a spreadsheet program (Goepel, 2013), but for more complex decision problems a software tool is needed.

There are several AHP software packages available on the market (see annex), and some of them have been described or were compared in the literature (Ossadnik, Kaspar 2013, Ishizaka, Labib 2009, Siraj *et al.* 2015), but most of these packages are targeting companies and pursue a business objective. Their focus lies on the application, and less on the methods and algorithms, which are often not fully transparent to users. The author's intention was to provide a complete and free software tool for educational and research purposes, where methods and algorithms are well documented and validated. As a result, a web-based AHP online system (AHP-OS) was developed and is available in its full functionality to any user for non-commercial purposes.

Implementation of the software was done in PHP, an open source general-purpose, object oriented scripting language that is especially suited to web development. Database functions are implemented using Structured Query Language (SQL), the most widely used database language with available open source database engines like SQLite or MariaDB. The whole package was developed from scratch, using only a few other available open source packages for general supporting functions like sending of e-mails or user login and registration (annex 1).

The functions and features of AHP-OS include

- Flexible definition of decision hierarchies as text input, following a simple syntax with multi-language support using Unicode character coding.
- Weight calculation (hierarchy mode) and alternative evaluation (alternative mode) using the AHP eigenvector method.
- Pairwise comparison input, highlighting the top-3 most inconsistent judgments.
- Partial judgments.
- *A posteriori* application of different AHP judgment scales.
- Group decision making using weighted geometric mean aggregation of individual judgments (WGM-AIJ).

- Group consensus calculation based on Shannon α and β -entropy.
- Weight uncertainty estimation using Monte Carlo simulation.
- Sensitivity analysis.
- Weighted sum model (WSM) and weighted product model (WPM) for the aggregation of alternatives.
- Export of input and result data as comma separated value (CSV) files for further processing or presentation in a spreadsheet program.

The software has a modular structure and is implemented using object oriented programming (OOP). It consists of five major objects (classes):

1. *ahp* class for all basic AHP calculations,
2. *ahpHierarchy* class for decision hierarchy related functions,
3. *ahpDb* class for database handling,
4. *ahpGroup* class for result calculations and
5. *ahpAdmin* class for all administrative functions like user and project administration.

These classes contain all AHP methods, and in the following we will describe the major functions in more detail.

2. Decision Hierarchies and Eigenvector Calculations

The definition of a decision hierarchy in the program is done with plain text input, consisting of a sequence of statements following a simple syntax. This provides different advantages:

- Hierarchy definitions in plain text form can be modified or duplicated by simple copy and paste functions
- They can be archived easily as text files.
- Unicode character coding allows for multi-language support; languages like Chinese, Korean, Thai *etc.* are supported.
- Weights can be completely or partly predefined within the hierarchy text. This is useful when weights are already known, *e.g.* from a previous project or derived from other MCDM methods, and AHP-OS is only used for alternative evaluation.

The syntax is defined as follows:

```
<hierarchy> → <branch>; [{<branch>;}]
<branch>    → <node>: <leafs>, <leafs> [,<leafs>]
<leafs>     → {<leaf> [= <weight>]}
```

For all <leafs> in a <branch> the sum of <weight> has to be one. A <node> of the second and any further <branch> has to be one of the <leaf> in <leafs>. Each <node> represents a decision matrix, and the corresponding priority vector's dimension is the number of <leafs>. Figure 1 shows a simple example of a two level hierarchy. In the first level (main criteria), weights for the two criteria are predefined as 30% and 70%. For the sub-criteria weights are automatically set to the default value $1/n_{leaf}$.

Decision Hierarchy			
Level 0	Level 1	Level 2	Global Priorities
AHP-Project	Criterion-1 0.3	Sub-criterion A 0.5	15.0 %
		Sub-criterion B 0.5	15.0 %
	Criterion-2 0.7	Sub-criterion C 0.5	35.0 %
		Sub-criterion D 0.5	35.0 %
			1.0

Fig. 1 Example of a decision hierarchy. This hierarchy has 2 levels, 3 nodes (AHP priority vectors) and 4 leafs.

This hierarchy in fig. 1 is defined by the following hierarchy text:

```
AHP-project: Criterion-1=0.3, Criterion-2=0.7;
Criterion-1: Sub-criterion A, Sub-criterion B;
Criterion-2: Sub-criterion C, Sub-criterion D;
```

All methods related to the decision hierarchy are coded in the *ahpHierarchy* class of the software. The class includes the parser to translate the hierarchy text into a multidimensional array. The parser checks for syntax errors and cleans the input text from redundant and impermissible characters. Supporting functions to extract nodes, branches or leafs from the hierarchy are included in this class.

AHP Eigenvector Method

Once a hierarchy is defined and stored in the database, weights can be calculated going through the following steps. Each step is coded as a method in the *ahp* class of AHP-OS:

1. Get pairwise comparisons (pwc): `get_pair_comp()`

2. Apply selected AHP judgment scale: `calcScale()`
3. Fill decision matrix from pairwise comparisons: `getMatrixFromPwc()`
4. Find the eigenvector using the power method: `set_evm_evec()`
5. Calculate the dominant eigenvalue from the eigenvector `setEvmEval()`
6. Calculate the consistency ratio *CR*: `setAlonsoCr()`
7. Calculate the inconsistency matrix: `get_incon_matrix()`
8. Identify and highlight the top three inconsistencies: `get_inconsistency()`
9. Go back to step 1 until the user submits his judgments.

Figure 2 shows an example of the display of pairwise comparisons to the user.

	A - wrt Buy tablet computer - or B?	Equal	How much more?
1	<input checked="" type="radio"/> display size or <input type="radio"/> battery life	<input checked="" type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8 <input type="radio"/> 9
2	<input checked="" type="radio"/> display size or <input type="radio"/> weight	<input checked="" type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8 <input type="radio"/> 9
3	<input checked="" type="radio"/> display size or <input type="radio"/> design	<input checked="" type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8 <input type="radio"/> 9

Figure 2. Pairwise Comparison Input.

All pairwise comparisons are internally stored in the format

$$pwc = (a_1, a_2, \dots, a_{npc}), (x_1, x_2, \dots, x_{npc}) \quad (2.1)$$

With integers $a_i \in [0,1]$, $x_i \in [1, M]$, $M = 9$ and $i = 1 \dots npc$ with npc is the number of pairwise comparisons.

$$npc = \frac{n^2 - n}{2}. \quad (2.2)$$

For n criteria the $n \times n$ decision matrix is then filled from pwc . For $a_i = 0$ we take x_i , for $a_i = 1$ we have to take the reciprocal of x_i . For example, for three criteria with $pwc = (0,0,1), (3,5,7)$ the decision matrix is

$$M = \begin{pmatrix} 1 & 3 & 5 \\ \frac{1}{3} & 1 & \frac{1}{7} \\ \frac{1}{5} & 7 & 1 \end{pmatrix} \quad (2.3)$$

The selected format minimizes redundancy and uses less memory than storing the complete positive reciprocal matrix. The dominant eigenvector (EV) of M is calculated using the power method (Larsen, 2013). The number of iterations is limited to 20, this is sufficient for an accepted approximation error of $1.E-7$.

Calculation of Consistency Ratio

Instead of Saaty's original consistency ratio calculation, based on the average random consistency index RI_n

$$CR = \frac{\lambda_{\max} - n}{(n-1)RI_n} \quad (2.4)$$

we use the linear fit proposed by Alonso and Lamata (2006) to calculate the consistency ratio CR .

$$CR = \frac{\lambda - n}{2.7699 \cdot n - 4.3513 - n} \quad (2.5)$$

It can also be used for matrices larger than 10 x 10. In case of a decision hierarchy with more than one node, CR of each hierarchy node is calculated, and for the global weights the maximum of all CR s is shown by the program.

Calculation of the top-3 inconsistencies

If the consistency ratio exceeds 10%, the software calculates the top-3 inconsistent judgments. We construct the inconsistency matrix

$$e_{ij} = d_{ij} \cdot \frac{w_j}{w_i} \quad (2.6)$$

and identify the three largest e_{ij} . The corresponding pairwise comparisons are then highlighted to the decision makers, allowing them to make some adjustments (Saaty, 2003).

AHP Judgment Scales

Over the last decades a variety of judgment scales, different from Saaty's fundamental 1 to 9 scale, have been proposed. In AHP-OS pairwise comparisons are stored with their original judgment values (eq. 2.1), we therefore can apply different scales *a posteriori* and study the effect of different scales on the resulting weights. Ten different scales are implemented and summarized in table 1.

A comparison of these scales, based on weight boundaries and weight ratio, weight uncertainty and weight dispersion, is given by Goepel (2017).

No	Name	Short	Scale function	Comment
0	Linear 1 to 9 AHP scale	AHP	$c = x$	Saaty (1980)
1	Logarithmic Scale	Log	$c = \log_a(x + a - 1)$	Ishizaka et. al (2010) $a=2$
2	Root square scale	Root	$c = \sqrt[a]{x}$	Harker, Vargas (1987) we use $a=2$
3	Inverse linear scale	Inv-lin	$c = \frac{9}{10 - x}$	Ma-Zheng (1991)
4	Balanced scale	Bal	$c = \frac{0.45 + 0.05x}{1 - 0.45 + 0.05x}$	Saalo, Hämäläinen (1997) for [0.1,0.9]
5	Balanced-n scale	Bal-n	$w_{bal} = \frac{1}{n} + \frac{w_{max} - \frac{1}{n}}{M - 1}(x - 1)$ $c = \frac{w_{bal}}{1 - w_{bal}}(n - 1)$	Corrected scale no. 4 n number of criteria
6	Adaptive-bal scale*	Adapt-bal	Same as 5 with $w_{max} = 0.9$	n number of criteria
7	Adaptive scale*	Adapt	$c = x^{1 + \frac{\ln(n-1)}{\ln 9}}$	n number of criteria
8	Power scale	Power	$c = x^a$	Harker, Vargas (1987) we use $a=2$
9	Geometric scale	Geom	$c = a^{x-1}$	Lootsma (1994), we use $a=2$

Table 1. AHP judgment scales implemented in the software.

3. AHP-OS Database and data structure

The software uses the PHP data object (PDO) interface to SQL databases. The current implementation can handle two SQL servers, either open source MariaDB, or the public domain SQLite database engine. The implemented database structure comprises of four database tables:

1. *Users*: table for user registration and login. Registered users can be project authors and can store and manage their own AHP projects.
2. *Projects*: table of AHP projects with unique session code, project name, project description and hierarchy definition. Its foreign key is the user name from the table *Users*.
3. *Judgments (pwc)*: table containing all pairwise comparisons with the decision makers' name and nodes of the decision hierarchy. Foreign key is the session code from the table *Projects*.
4. *Alternatives*: table with alternative names. Foreign key is the session code from the table *Projects*.

This structure minimizes redundancies and keeps the database slim (table 2). In order to identify projects, each project gets a unique 6 character session code. As described before, decision hierarchies are defined with plain text input, and pairwise comparisons are stored in a simple integer format. For group decision making participants' judgments are stored in the *pwc* table. Participants don't need to register as users; they just need to follow a link with the unique session code provided by the AHP project author (registered user). They then can start to input their judgments by opening the link and providing their name. (Field *participant* in table *pwc*).

1. Table *user*

User Id	user_name	User_email	reg_date_time
3487	Klaus	drklaus@...			2017-07-30
Key	Unique	Unique			

2. Table *projects*

session_code	project_name	description	hierarchy_text	date_time	user_name
yEyKfc	AHP Project	... project ...	P: A, B; A: ...	2017-07-30	Klaus
Unique	Varchar(64)	Text(400)	Text(6000)		Foreign key

3. Table *pwc* (Judgments)

session_code	participant	timestamp	node	pwc_a	pwc_i
yEyKfc	John	14356787	Criterion-1	001011	143795
Foreign key	Varchar(64)		Varchar(30)	Int 0 or 1	Int 1 to 9

4. Table *alternatives*

session_code	alternatives
yEyKfc	Alternative 1
Foreign key	Varchar(64)

Table 2. SQL database structure.

The *ahpDb* software class handles the interfacing with the SQL database server. As an example, the following SQL statement:

```
SELECT session_code, project_name, count(alt) AS Alt, count(pwpart) AS Part, description,
date(date_time) FROM projects
LEFT JOIN
(SELECT DISTINCT pwc.session_code AS pwsc, pwc.participant AS pwpart FROM pwc)
ON projects.session_code = pwsc
LEFT JOIN
(SELECT alternatives.session_code AS altsc, projects.session_code AS alt FROM projects,
alternatives WHERE projects.session_code = alternatives.session_code
GROUP BY alternatives.session_code
) ON projects.session_code = altsc
WHERE projects.user_name = :user_name
GROUP BY session_code ORDER BY projects.date_time DESC;
```


will result in a table of all projects for a specified user name, showing the unique project session code, the project name, number of alternatives (if any), number of participants (decision makers) with pairwise comparison inputs, a short project description and the creation date/time of the project. The database does not store results; all results are calculated on the fly using methods in the software class *ahpGroup*.

Basic Data structure

Each node in the decision hierarchy represents a decision matrix with the number of branches as number of criteria. For the example given in fig.1 *AHP project*, *criterion-1* and *criterion-2* are the nodes of the hierarchy with two branches each, and *sub-criterion A* to *D* build the leafs of the decision hierarchy. In the software we need to calculate the weights of each branch or leaf of all nodes and for all decision makers (participants). The weights (local priorities) are stored in a 3-dimensional array with array keys [*participant*], [*node*] and [*leaf*]. The array key [*participant = 0*] holds the consolidated group results; the array key [*node = "pTot"*] holds the global priorities. This data structure applies to both, hierarchy and alternative evaluation, where for the latter the hierarchy's leafs become nodes with alternatives as branches. For any calculation the software has to loop through all nodes.

Once the weight calculations are done, the results (local priorities) are inserted into the hierarchy definition text, for example *Criterion-2* becomes *Criterion-2=0.7*. The modified hierarchy text can then be used to define alternatives and be stored as a new project in the database for alternative evaluation.

Partial Judgments

As pairwise comparisons for each node are stored in the pairwise comparisons database table, it is possible for participants to do a partial evaluation, i.e. only give their input for a part of the hierarchy or the alternatives. This can be useful, if a project requires inputs from different experts with specific expertise in selected nodes of the decision hierarchy.

5. Group decision making

Aggregation of individual judgments

One of the major advantages of a web based AHP software is that it allows worldwide access. Once a decision hierarchy is defined and stored as a project on the server, decision makers can give their inputs from any location, as long as they have internet access. In AHP-OS each project is identified by a unique

session code, and participants only need a link with this session code to start their pairwise comparisons and submit their judgments. The number of participants is practically unlimited.

In order to consolidate all participants' judgments, several group AHP aggregation techniques are available, but not all group AHP methods are equally convenient and the selection of the method might depend on the specific application (Grošelj, P *et al.*, 2015). In the current version of the software we decided to use the weighted geometric mean aggregation of individual judgments (WGM-AIJ), as it is the only method that meets several required axiomatic conditions, such as the reciprocal property. We calculate the geometric mean and standard deviation of all K participant's individual judgments pwc_k to form the consolidated decision matrix a_{ij}^{cons} .

Function `pwcGeoMean($pwcArray)`

$$\text{Sum over } K \text{ participants} \quad pwcx = \sum_{k=1}^K \ln(pwc_k) \quad (5.1)$$

$$\text{Square sum over } K \text{ participants} \quad pwcx2 = \sum_{k=1}^K [\ln(pwc_k)]^2 \quad (5.2)$$

$$\text{Geometric mean} \quad pwc_{\text{CONS}} = \exp\left(\frac{pwcx}{K}\right) \quad (5.3)$$

$$\text{Standard deviation} \quad pwc_{\text{SD}} = \exp\left(\sqrt{\frac{pwcx2 - \frac{1}{K} pwcx \cdot pwcx}{K-1}}\right) \quad (5.4)$$

Using the pairwise comparison data from eq. 5.3 to fill the consolidated decision matrix we get

$$a_{ij}^{\text{cons}} = \left(\prod_{k=1}^K a_{ij}\right)^{\frac{1}{K}} \quad (5.5)$$

The standard deviation pwc_{SD} of individual judgments (eq. 5.4) is used for the estimation of weight variations based on judgment variations.

Group Consensus

Although it is always possible to calculate a group result using WGM-AIJ, it does not makes sense in all cases. For example, if we have two opposite judgments for two criteria (x and $1/x$), an aggregation will result in equal weights for both criteria. In fact, there is no consensus (agreement), and equal weights may result in a deadlock situation to solve a decision problem. Therefore, it is necessary to measure the consensus for the aggregated group result. We use Shannon entropy and its partitioning in two independent components (alpha and beta diversity) to derive the AHP consensus indicator. We analyze the priority distribution of criteria among different decision makers. We use the following relations (Goepel, 2013):

```
Function getShannonBeta($prio, $node)
```

$$\text{Shannon beta-entropy} \quad H_{\beta} = H_{\gamma} - H_{\alpha} \quad (5.6)$$

$$\text{Shannon alpha-entropy} \quad H_{\alpha} = \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^n (-w_j \ln w_j) \quad (5.7)$$

$$\text{Shannon gamma-entropy} \quad H_{\gamma} = \sum_{j=1}^n (-w_{\text{avg}} \ln w_{\text{avg}}) \quad (5.7)$$

$$\text{with} \quad w_{\text{avg}} = \frac{1}{k} \sum_{j=1}^k w_{jk}$$

The similarity measure S (eq. 5.8) depends on the number of criteria, and we used a linear transformation to map it to a range from 0 to 1 (eq. 5.9)

$$S = \exp(-H_{\beta}) = \frac{1}{D_{\beta}} \quad (5.8)$$

$$S^* = \frac{S - D_{\alpha \min} / D_{\gamma \max}}{1 - D_{\alpha \min} / D_{\gamma \max}}$$

In general $D_{\alpha \min} = 1$ and $D_{\gamma \max} = n$. In the analytic hierarchy process (AHP) minimum alpha entropy $H_{\alpha \min} = \ln(D_{\alpha \min})$ is a function of the maximum scale value M ($M = 9$ for the fundamental AHP scale) and the number of criteria n (eq. 5.9).

$$H_{\alpha \min} = -\left(\frac{M}{n+M-1}\right) \ln \left[\frac{M}{n+M-1}\right] - \left(\frac{n-1}{n+M-1}\right) \ln \left[\left(\frac{1}{n+M-1}\right)\right] \quad (5.9)$$

Maximum gamma entropy $H_{\gamma \max} = \ln(D_{\gamma \max})$ is

$$H_{\gamma \max} = \ln(n) \quad (5.10)$$

```
Function hamin($nCrit, $mScale)
```

The correction factor cor is

```
Function ahpShannonCor($nCrit, $mScale)
```

$$cor = \exp[H_{\gamma \max} - H_{\alpha \min}]^{-1} = D_{\alpha \min} / D_{\gamma \max} = \frac{\exp(H_{\alpha \min})}{n} \quad (5.11)$$

The AHP consensus indicator is based on the relative homogeneity S

```
Function relHomogeneity($dBeta, $dBetaMin)
```

$$S = \frac{1}{D_{\beta}} - \frac{1}{D_{\beta \min}} \quad (5.12)$$

AHP Group Consensus Indicator

Function `getConsensus($prio, $node, $iScale)`

$$S^* = \left(\frac{1}{\exp(H_\beta)} - \frac{1}{\exp(H_{\gamma_{\max}} - H_{\beta_{\min}})} \right) / \left(1 - \frac{1}{\exp(H_{\gamma_{\max}} - H_{\beta_{\min}})} \right) \quad (5.13)$$

$$S^* = \frac{1/D_\beta - cor}{1 - cor} \quad (5.14)$$

The consensus of the complete hierarchy is calculated as weighted arithmetic mean of the consensus of all hierarchy nodes (categories); for alternatives we take the consensus indicator of the consolidated alternative evaluation matrix weighted with the global priorities.

The consensus indicator ranges from 0% (no consensus) to 100% (full consensus). The whole range is categorized into five categories according table 3.

S^*	Consensus
$\leq 50\%$	Very low
50% - 65%	low
65% - 75%	moderate
75% - 85%	high
$\geq 85\%$	Very high

Table 3. Interpretation of AHP consensus.

6. Weight Uncertainties

As shown by Goepel (2017) weight uncertainties due to rounding of the judgment to integers can exceed 10%, and could impact the results of a decision. In order to get an estimation of uncertainties, we generate *NVAR* randomized variations of the original judgments with

Function `generateRandPwc($pwc, $n, $dx)`

$$pwc_j = (a_1, \dots, a_i, \dots, a_{npc}), (x_1 + \Delta x_1, \dots, x_i + \Delta x_i, \dots, x_{npc} + \Delta x_{npc}) \quad (6.1)$$

With $a_1 = 0$ for $x_i + \Delta x_i \geq 1$, and $a_1 = 1$ for $x_i + \Delta x_i < 1$,

where ΔX are randomly uniform distributed values (-0.5, 0., +0.5).

$$\Delta X \sim 0.5 \cdot U(-1, +1) / \sqrt{K} \quad (6.2)$$

and K is the number of participants.

For each variation $pwcr_j$ the eigenvector solution w_j and consistency ratio CR is calculated. We then capture the maximum and minimum of the weights for all $CR < 0.25$.

Function `uncertainty($pwcr, $ahpH, $iScale, $sdFlg, $hierMode)`

$$\Delta w_{(+)} = (w_{max} - w) = \left(\max[w_j = EV(pwcr_j)]_{j=1}^{NVAR} - w \right) \tag{6.3}$$

$$\Delta w_{(-)} = (w - w_{min}) = \left(w - \min[w_j = EV(pwcr_j)]_{j=1}^{NVAR} \right) \tag{6.4}$$

Figure 3 shows a typical example of weight uncertainties of a project with x criteria using the standard AHP 1 to 9 scale.

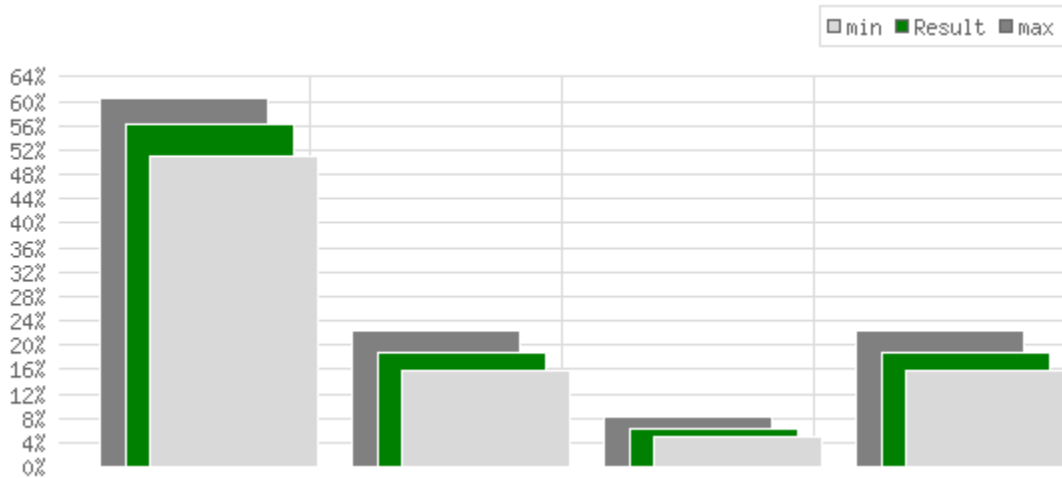


Figure 3. Calculated weights for a project with 7 criteria with weight uncertainties due to rounding.

Alternatively, users can also display uncertainties using the standard deviation of judgments $pwcs_{SD}$ (eq. 5.4), if there are more than one participant. It gives, in addition to the AHP consensus, an idea about the weight variations based on participants' judgment variations.

$$pwcr_j = (a_1, \dots, a_i, \dots, a_{npc}), (x_1 \Delta x_1, \dots, x_i \Delta x_i, \dots, x_{npc} \Delta x_{npc}) \tag{6.5}$$

$$\Delta X \sim N(\mu, \sigma^2) \tag{6.6}$$

$$\Delta x_i = \exp\left(\frac{1}{2} \log pwcs_{SD} \cdot \Delta X\right) \tag{6.7}$$

Overlap

The program provides a list of criteria or alternatives overlapping within the estimated weight uncertainties. Overlapping criteria or alternatives should get the same ranking, as they cannot be differentiated within the uncertainties.

7. Alternative Evaluation

Weighted Sum and Weighted Product Model

Function `calcPrioTotal($k,$pGlb)`

In AHP the preference P_i of alternative A_i is calculated using the weighted sum model (WSM):

$$P_i = \sum_{j=1}^n a_{ij} w_j \quad (7.1)$$

with W_j the weight of criterion C_j , and a_{ij} the performance measure of alternative A_i with respect to criterion C_j . Performance values are normalized.

$$\sum_{j=1}^n a_{ij} = 1 \quad (7.2)$$

In the software users can also select the weighted product model (WPM), where alternatives are aggregated using the product instead of the sum

$$P_i = \prod_{j=1}^n a_{ij}^{w_j} \quad (7.3)$$

Some of the first references to this method are due to Bridgman (1922) and Miller and Starr (1969).

8. Sensitivity Analysis

Sensitivity analysis is a fundamental concept in the effective use and implementation of quantitative decision models, whose purpose is to assess the stability of an optimal solution under changes in the parameters. Sensitivity analysis (Triantaphyllou, 1997) will answer two questions:

- Which is the most critical criterion, and
- Which is the most critical performance measure,

changing the ranking between two alternatives?

The *most critical criterion* is defined as the criterion C_k , with the smallest absolute ($\delta_{k,i,j}^{abs}$) or relative ($\delta_{k,i,j}^{rel}$) change of the current weight W_k by the amount of δ_{kij} changing the ranking between the alternatives A_i and A_j .

The *most critical measure of performance* is defined as the minimum absolute or relative change of the current value of a_{ij} such that the current ranking between alternative A_i and A_j will change. Calculation is different for WSM and WPM.

Weighted Sum Model (WSM)

To find the *most critical criterion* we calculate for each pair of alternatives A_i, A_j , with $i = 1$ to n and $i < j$

$$\delta_{i,j,k}^{\text{abs}}(A_i, A_j, W_k) = \frac{P_j - P_i}{a_{jk} - a_{ik}} \quad (8.1)$$

with $|\delta_{i,j,k}^{\text{abs}}(A_i, A_j, W_k)| < W_k$

and $\delta_{i,j,k}^{\text{rel}}(A_i, A_j, W_k) = \delta_{i,j,k}^{\text{abs}}/W_k$

To find the *most critical measure of performance* we calculate for all alternatives A_i and A_k with $i \neq k$ and each criterion

$$\delta_{i,j,k}^{\text{abs}}(W_j, a_{kj}, a_{ij}) = \frac{P_i - P_k}{P_i - P_k + W_j(a_{kj} - a_{ij} + 1)} \quad (8.2)$$

with $|\delta_{i,j,k}^{\text{abs}}(W_j, a_{kj}, a_{ij})| < W_j$

and $\delta_{i,j,k}^{\text{rel}}(W_j, a_{kj}, a_{ij}) = \delta_{i,j,k}^{\text{abs}}/a_{ij} \quad (8.3)$

Weighted Product Model (WPM)

To find the *most critical criterion* for each pair of alternatives A_i, A_k , with $i = 1$ to n and $i < k$ we calculate

$$\log\left(\prod_{y=1}^n (a_{iy}/a_{ky})^{W_y}\right) \quad (8.4)$$

and $\delta(A_i, A_k, W_j) = \frac{\log\left(\prod_{y=1}^n (a_{iy}/a_{ky})^{W_y}\right)}{\log(a_{ij}/a_{kj})} \quad (8.5)$

with $\delta^{\text{abs}}(A_i, A_k, W_j) < W_j$

Then we look at the smallest $\delta^{\text{rel}}(A_i, A_k, W_j) = \frac{\delta^{\text{abs}}}{W_j} \quad (8.6)$

To find the *most critical measure of performance* for the WPM we calculate for all alternatives A_i and A_k with $i \neq k$ and each criterion

$$\delta^{\text{rel}}(A_i, A_k, W_j) = \left(1 - \sqrt[W_j]{\frac{A_k}{A_i}}\right) \quad (8.7)$$

and look for the smallest $\delta^{\text{rel}}(A_i, A_k, W_j) < 1$ (8.8)

and $\delta^{\text{abs}} = a_{ij} \delta^{\text{rel}}$ (8.9)

Program Output for Sensitivity Analysis

Depending on the results, the program will then display the following text output

- The solution for the top alternative A_x robust.
- The percent-top critical criterion is C_y , a change from $y\%$ by absolute $dy\%$ will change the ranking between alternatives A_i and A_k .
- The percent-any critical criterion is the same as above.
- The percent-any critical criterion is C_z , a change from $y\%$ by absolute $dz\%$ will change the ranking between alternatives A_i and A_j .
- The percent-any critical performance measure is for alternative A_i under criterion C_z . A change from $x\%$ by absolute $dx\%$ will change the ranking between A_i and A_j .

When exporting the results, the complete sensitivity tables (eq. 8.1 and 8.2 or eq. 8.5 and 8.9) will be exported.

9. Data Export

AHP-OS is intended for educational and research purposes, therefore it is important to provide an extensive possibility of data export for further analysis or presentation. The most common and universal format is the comma separated value (csv) text format. It can be easily imported into most spreadsheet programs, independent from the operation system. The software provides download functions for input data (hierarchy, decision matrices), local and global weights with estimated uncertainties, alternative tables and sensitivity analysis.

10. Software Validation

Coding and implementation of the complete AHP-OS software package resulted in approximately ten thousand lines of code; therefore testing and validation became essential. With each new release the author goes through a pre-defined checklist for unit, integration and system testing. In addition, in order to ensure that the user can rely on the accuracy of the program results, test cases were designed for black box testing. We compare program results with manually calculated results and results published in the literature.

A simple test case with known results is the case, where we judge one criterion x -times more important than all others. The weight for this criterion has to be (Goepel, 2017)

$$w_{\text{AHP}} = \frac{x}{x+n-1} \quad (10.1)$$

For example, if all criteria are equal ($x = 1$), it follows that $w_{\text{AHP}} = 1/n$, and for four criteria, with one criterion nine times more important than all others, it follows $w_{\text{AHP}} = \frac{9}{12}$ or 75%.

In a next step we can apply different *scales functions* according table 1 by replacing x with the scaled value c .

Group decision making can be validated by verifying that the elements of the consolidated group matrix show the geometric mean of the elements of the individual matrices.

Testing the *AHP consensus indicator* is done by simulating k participants for a hierarchy with k leafs, and each participant judges one different leaf extreme more important than all others. The resulting consensus should show 0% for all scales.

As a test case for *weight uncertainties* we use the derivation of eq. 10.1

$$\frac{dw}{dx} = \frac{n-1}{(x+n-1)^2} \quad (10.2)$$

With $n = 2$, $dx = 0.5$ and $x' = x \pm dx/2$, $x > 1$ we get

$$\Delta w_{(+)} = \frac{0.5}{(x+1.25)^2} \quad (10.3)$$

$$\Delta w_{(-)} = \frac{0.5}{(x-0.75)^2} \quad (10.4)$$

For example, for two criteria A and B with A three times more important than B, we get $w = 75\%$ with $\Delta w_{(+)} = 2.8\%$ and $\Delta w_{(-)} = 3.6\%$.

Alternative evaluation using the *weighted sum* or *weighted product* model can be verified exporting the results and using a spreadsheet program and its SUMPRODUCT function.

As a test case for *sensitivity analysis* we use a spreadsheet program with exported data and subtract δ^{abs} from the original weight W_j

$$W'_k = W_k - \delta^{\text{abs}} \quad (10.5)$$

Using WSM eq. 7.1 or WPM eq. 7.3 for alternative aggregation the performance measure of alternative A_i should now equal that of alternative A_j . In a similar way we can calculate

$$a'_{ij} = a_{ij} - \delta^{\text{abs}} \quad (10.6)$$

and alternative A_i should have the same performance measure as alternative A_k . For WSM we have to renormalize a_{ij} according eq. 7.2.

11. Experiences and Outlook

AHP-OS was originally released in 2014. Features described in this paper were developed gradually over the last three years. With the currently implemented functionality the software tool has reached a state, where it covers most of the possible options for the classical analytic hierarchy process. It allows for a wide range of parameter variations to analyze and study a specific project under different aspects. This and the possibility of transparent data export makes it an ideal tool for study, education and further research.

Up to the date of writing this paper more than 5000 users have registered for the software, and on average there are at least 500 active users over a three months period. The AHP projects handled with AHP-OS cover a wide range of applications like healthcare, climate, risk assessment, supplier selection, hiring, IT, marketing, environment, transport, project management, manufacturing or quality assurance. Numerous projects are used to acquire group inputs, the number of participants goes up to 320 in single projects. Multi-language support is utilized for languages like Chinese, Korean, Russian, Hebrew, Greece, Thai, Vietnamese or Arabic. Lecturers at universities make use of AHP-OS as a tool for their students when teaching multi criteria decision making methods.

Overall the number of software bugs could be kept reasonable low, more often users' feedback and questions relate to the understanding of the AHP and can be solved straightforwardly. The main challenges in maintaining the software are updates of the underlying open source tools due to security reasons, or smaller updates requested by users. Because of the complexity of the software they always require extensive testing and validation.

A few extensions of AHP-OS are under consideration for the future, for example a selection of different group aggregation techniques in addition to WGM-AIJ. As a major extension a cluster analysis of group consensus is planned. The concept of Shannon entropy and its partitioning in alpha- and beta-entropy opens the possibility to identify clusters of sub-groups with high consensus within a group of decision makers. This could help to gain a deeper insight into group decision making processes.

References

Alonso, J. A., Lamata, T. (2006). *Consistency in the Analytic Hierarchy Process: A New Approach*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 14, No. 4, 445–459.

Bridgman, P.W. (1922). *Dimensional Analysis*. New Haven, CT, U.S.A.: Yale University Press.

Goepel, K. D. (2017). *Comparison of Judgment Scales of the Analytical Hierarchy Process - A New Approach*, submitted for consideration in International Journal of Information Technology and Decision Making.

Goepel, K. D. (2013). *Implementing the Analytic Hierarchy Process As A Standard Method For Multi-Criteria Decision Making In Corporate Enterprises – A New AHP Excel Template with Multiple Inputs*, Proceedings of the International Symposium on the Analytic Hierarchy Process, Kuala Lumpur.

Grošelj, P. Stirn, L. Z., Ayrilmis, N., Kuzman, M. K. (2015) *Comparison of some aggregation techniques using group analytic hierarchy process*. Expert Systems with Applications 42, 2198–2204.

Ishizaka, A., Nemery, P. (2013). *Multi-Criteria Decision Analysis, Methods and Software*. John Wiley & Sons Ltd, UK, 2013, ISBN 978-1-119-97407-9.

Ishizaka, A., Labib, A. (2009) *Analytic Hierarchy Process and Expert Choice: Benefits and Limitations*, OR Insight, 22(4), p. 201–220.

Larsen, R. (2013). *Elementary Linear Algebra*, chapter 10, p 551 ff, ISBN 978-1-305-65800-4, Cengage Learning, Boston, MA.

Miller, D.W.; M.K. Starr (1969). *Executive Decisions and Operations Research*. Englewood Cliffs, NJ, U.S.A.: Prentice-Hall, Inc.

Ossadnik, W., Kaspar, R. (2013). *Evaluation of AHP software from a management accounting perspective*. Journal of Modelling in Management, Vol. 8 No. 3, pp. 305-319

Saaty, T.L. (2003). *Decision-making with the AHP: Why is the principal eigenvector necessary*, European Journal of Operational Research 145, 85–91.

Saaty, T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, ISBN 0-07-054371-2, McGraw-Hill.

Siraj, S., Mikhailov, L., Keane, J. A., PriEsT (2015). *An Interactive Decision Support Tool To Estimate Priorities From Pairwise Comparison Judgments*, International Transactions in Operational Research, Vol. 22, Issue 2, 217–235.

Triantaphyllou, E., Sánchez, A. (1997). *Sensitivity Analysis Approach for Some Deterministic Multi-Criteria Decision Making Methods*. Decision Sciences, Vol. 28, No. 1, pp. 151-194.

Wen-Hsiang Wu, Chang-tzu Chiang, Chin-tsai Lin (2008). *Comparing the aggregation methods in the analytic hierarchy process when uniform distribution*. WSEAS Transactions On Business And Economics, Issue 3, Volume 5, p 82 – 87, March 2008.

Annex

1. Open Source Software Packages used for AHP-OS

PHP, Version 5.6.31, <http://php.net/>

MariaDB, MariaDB Foundation, Version 5.5.54, <https://mariadb.org/>

SQLite, Version 3.8.10.2, <https://sqlite.org/index.html>

jQuery JavaScript library, <https://jquery.com/>

PHPGraphLib Graphing Library, Elliot Brueggeman, <http://www.ebrueggeman.com/phpgraphlib>

PHPMailer, the classic email sending library for PHP, <https://github.com/PHPMailer/PHPMailer>

2. Other AHP Software Packages

- 123AHP, <http://123ahp.com/>
- AHP-OS, <http://bpmsg.com/academic/ahp.php>
- MakeItRational, <http://makeitrational.com>
- ChoiceResults, <http://www.filebuzz.com/fileinfo/91865/ChoiceResults.html>
- HIPRE 3+, <http://sal.aalto.fi/en/resources/downloadables/hipre3>
- SuperDecisions, <https://superdecisions.com/>
- Expert Choice, <http://expertchoice.com/>
- Decision Lens, <http://decisionlens.com>
- CDP, Criterium® DecisionPlus®, <http://www.infoharvest.com/ihroot/index.asp>
- RightChoiceDSS, <http://www.updatestar.com/publisher/tgkconsulting-2369>
- PriEsT, AHP Priority Estimation Tool, <https://sourceforge.net/projects/priority/>