

Scilab Programming for Solving Problems of Multi Attribute Decision Making Using Analytical Hierarchical Process (AHP)

Neethu Fernandes¹, Rakhi Bhadkamkar²

¹Fr. C. Rodrigues Institute of Technology

²Fr. Agnel Technical Education Complex, Sec 9A, VashiNavi Mumbai - 400703

Abstract: In many engineering application and even the real life problems the final decision is based on the evaluation of a number of alternatives in terms of a number of criteria. The Analytic Hierarchy Process (AHP) is an effective approach in dealing with this kind of decision problems. In this paper one of the numerical examples of AHP is illustrated and the corresponding user friendly Scilab program is proposed.

Keywords: Analytical hierarchical process, priority vector, multiattribute, decision making techniques, Multi criteria decision making

1. Introduction

The Analytic Hierarchy Process (AHP) is a multi-criteria decision-making approach and was introduced by Saaty (1977 and 1994). It uses a multi-level hierarchical structure of objectives, criteria, sub criteria and alternatives. The pertinent data are derived by using a set of pair wise comparisons. These comparisons are used to obtain the weights of importance of the decision criteria, and the relative performance measures of the alternatives in terms of each individual decision criterion.

2. The AHP Divides the Decision Problem into the Following Main Steps

A) Problem Structuring

The AHP decision problem is structured hierarchically at different levels, each level consisting of a finite number of decision elements. Consider one such problem with four criteria C1, C2, C3 and C4 at level 1 and three alternatives A, B and C. As in Fig 1.

B) Computing the Vector (Priority Vector) of Criteria Weights

In order to compute the weights for the different criteria, the AHP starts creating a pair wise comparison matrix G . The matrix ' G ' is a $n \times n$ real matrix, where n is the number of evaluation criteria considered. Each entry $g_{j,k}$ of the matrix G , represents the importance of the j th criterion relative to the k th criterion. If $g_{j,k} > 1$, then the j th criterion is more important than the k th criterion, while if $g_{j,k} < 1$, then the j th criterion is less important than the k th criterion. If two criteria have the same importance, then the entry $g_{j,k}$ is 1.

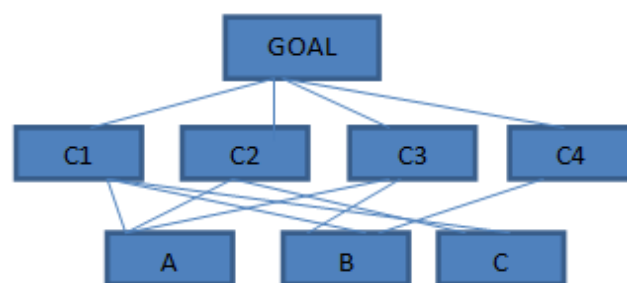


Figure 1

The entries $g_{j,k}$ and $g_{k,j}$ satisfy the following constraint:

$1 = g_{j,k} g_{k,j}$ Obviously, $g_{j,j} = 1$ for all j . The relative importance between two criteria is measured according to a numerical scale from 1 to 9, as shown in Table 1, where it is assumed that the j th criterion is equally or more important than the k th criterion. The phrases in the "Interpretation" column of Table 1 are only suggestive, and may be used to translate the decision maker's qualitative evaluations of the relative importance between two criteria into numbers. It is also possible to assign intermediate values which do not correspond to a precise interpretation. The values in the matrix G are by construction pairwise consistent. On the other hand, the ratings may in general show slight inconsistencies. However these do not cause serious difficulties for the AHP.

Table 1

Value of g_{jk}	Interpretation
1	j and k are equally important
3	j is slightly more important than k
5	j is more important than k
7	j is strongly more important than k
9	j is absolutely more important than k

Once the matrix G is built, it is possible to derive from G the normalized pairwise comparison matrix G_{norm} by making equal to 1 the sum of the entries on each column, i.e. each entry \bar{g}_{jk} of the matrix G_{norm} is computed as.

$$\bar{g}_{j,k} = \frac{g_{j,k}}{\sum_{j=1}^n g_{j,k}} \quad S_j = \sum_{j=1}^n g_{j,k}$$

Finally, the w (that is an n -dimensional column vector) is built by averaging the entries on each row of G_{norm} ,

$$i.e. w_j = \frac{\sum_{i=1}^m \bar{g}_{ji}}{n} \text{ (priority vector)}$$

$$\lambda_{max} = w \times S$$

$$CI = (\lambda_{max} - n) / (n - 1)$$

C. Checking the Consistency of the Comparison Matrix:

However, perfect consistency rarely occurs in practice. In the AHP the pairwise comparisons in a judgment matrix are considered to be adequately consistent if the corresponding consistency ratio (CR) is less than 10% (Saaty, 1980). The CR coefficient is calculated as follows. First the consistency index (CI) needs to be estimated. This is done by adding the columns in the judgment matrix and multiply the resulting vector S by the vector of priorities W (i.e., the approximated eigenvector) obtained earlier. This yields an approximation of the maximum eigenvalues, denoted by λ_{max} . Then, the CI value is calculated by using the formula: $CI = (\lambda_{max} - n) / (n - 1)$. Next the consistency ratio CR is obtained by the Random Consistency index (RI) as given in Table 2.

Table 2: Random Consistency Index (RI)

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

A perfectly consistent decision maker should always obtain $CI=0$, but small values of inconsistency may be tolerated. In particular, if $CI/RI < 0.1$ the inconsistencies are tolerable, and a reliable result may be expected from the AHP.

Numerical Example: Suppose that the three alternative computer systems need to be evaluated in terms of the four decision criteria: hardware expandability, hardware maintainability, finance availability and user friendly characteristic of the operating system.

Suppose the following is the judgment matrix when the three alternative configurations are examined in terms of the first criterion

C1: Hardware expandability	A	B	C	Priority vector
A	1	6	8	0.7391
B	1/6	1	4	0.1915
C	1/8	1/4	1	0.0694
S	1.291	7.25	13	

$$\lambda_{max} = 3.246 \quad CI = 0.123$$

Judgment matrix when the three alternative configurations are examined in terms of the second criterion

C2: Hardware maintainability	A	B	C	Priority vector
A	1	7	1/5	0.2504
B	1/7	1	1/8	0.06
C	5	8	1	0.6895
S	6.1428	16	1.325	

$$\lambda_{max} = 3.405 \quad CI = 0.2025$$

Judgment matrix when the three alternative configurations are examined in terms of the third criterion

C3: Finance availability	A	B	C	Priority vector
A	1	8	6	0.7391
B	1/8	1	1/4	0.0694
C	1/6	4	1	0.1915
S	1.291	13	7.25	

$$\lambda_{max} = 3.24 \quad CI = 0.12$$

Judgment matrix when the three alternative configurations are examined in terms of the four criterion

C4: User friendly characteristic	A	B	C	Priority vector
A	1	5	4	0.665
B	1/5	1	1/3	0.1038
C	1/4	3	1	0.231
S	1.45	9	5.333	

$$\lambda_{max} = 3.13 \quad CI = 0.065$$

Judgment matrix for the case of comparing the importance of four decision criteria

The four criteria	C1	C2	C3	C4	Priority vector
C1	1	5	3	7	0.5438
C2	1/5	1	1/3	5	0.1412
C3	1/3	3	1	6	0.2656
C4	1/7	1/5	1/6	1	0.0442
S	1.676	9.2	4.5	19	

$$\lambda_{max} = 4.245 \quad CI = 0.08166$$

Calculating the adjusted weight:

$$\text{Adjusted weight wrt to } C_j (W_{adj}) = W_j / \sum W_j$$

Over all composite weight of the alternatives

	Criteria a C1	Criteria a C2	Criteria a C3	Criteria a C4	Composite weight
Adjusted weight	0.5466	0.1485	0.2669	0.0444	
Alt A	0.7391	0.2504	0.7391	0.665	0.6663
Alt B	0.1915	0.06	0.0694	0.1038	0.1363
Alt C	0.0694	0.6895	0.1915	0.231	0.1917

Scilab programming

```
clear
clc
//comparison matrix for the level1(goal)
g=input('Enter the comparison matrix for the criterion')
s(1)=0;s(2)=0;s(3)=0;s(4)=0;
fori=1:4;
for j=1:4;
s(i)=s(i)+g(j, i);
end
end
fori=1:4;
```

```

for j=1:4;
g(j, i)=g(j, i)/s(i);
end
end

wg(1)=0;wg(2)=0;wg(3)=0;wg(4)=0;
fori=1:4;
for j=1:4;
wg(i)=wg(i)+g(i, j);
end
wg(i)=wg(i)/4;
end
lambdamax=wg(1)*s(1)+wg(2)*s(2)+wg(3)*s(3)+wg(4)*s(
4);
CI=(lambdamax-4)/3;
CR=CI/0.9;
printf('\nlambdamax at level 1 %f, lambdamax)
printf('\nconsistency ratio for comparison of the factors wrt
goal is %f, CR)
//comparison matrix for level2(with respect to the factor A
xa=[1 6 8;1/6 1 4;1/8 1/4 1];
sa(1)=0;sa(2)=0;sa(3)=0;
fori=1:3;
for j=1:3;
sa(i)=sa(i)+xa(j, i);
end
end
fori=1:3;
for j=1:3;
xa(j, i)=xa(j, i)/sa(i);
end
end

wa(1)=0;wa(2)=0;wa(3)=0;
fori=1:3;
for j=1:3;
wa(i)=wa(i)+xa(i, j);
end
wa(i)=wa(i)/3;
end
lambdamax=wa(1)*sa(1)+wa(2)*sa(2)+wa(3)*sa(3);
CI=(lambdamax-3)/2;
CR=CI/0.58;
printf('\nlambdamax at level 2 for the wrt factor A %f,
lambdamax)
printf('\nconsistency ratio for comparison of the factors wrt
factor A is %f, CR)
//comparison matrix for level2(with respect to the factor B
xb=[1 7 1/5;1/7 1 1/8;5 8 1];
sb(1)=0;sb(2)=0;sb(3)=0;
fori=1:3;
for j=1:3;
sb(i)=sb(i)+xb(j, i);
end
end
fori=1:3;
for j=1:3;
xb(j, i)=xb(j, i)/sb(i);
end
end
wb(1)=0;wb(2)=0;wb(3)=0;
fori=1:3;
for j=1:3;
wb(i)=wb(i)+xb(i, j);
end
end

end
wb(i)=wb(i)/3;
end
lambdamax=wb(1)*sb(1)+wb(2)*sb(2)+wb(3)*sb(3)
CI=(lambdamax-3)/2;
CR=CI/0.58;
printf('\nlambdamax at level 2 for the wrt factor B %f,
lambdamax)
printf('\nconsistency ratio for comparison of the factors wrt
factor B is %f, CR)
xc=[1 8 6;1/8 1 1/4;1/6 4 1];
sc(1)=0;sc(2)=0;sc(3)=0;
fori=1:3;
for j=1:3;
sc(i)=sc(i)+xc(j, i);
end
end
fori=1:3;
for j=1:3;
xc(j, i)=xc(j, i)/sc(i);
end
end

wc(1)=0;wc(2)=0;wc(3)=0;
fori=1:3;
for j=1:3;
wc(i)=wc(i)+xc(i, j);
end
wc(i)=wc(i)/3;
end
lambdamax=wc(1)*sc(1)+wc(2)*sc(2)+wc(3)*sc(3)
CI=(lambdamax-3)/2
CR=CI/0.58
xd=[1 5 4;1/5 1 1/3;1/4 3 1];
sd(1)=0;sd(2)=0;sd(3)=0;
fori=1:3;
for j=1:3;
sd(i)=sd(i)+xd(j, i);
end
end
fori=1:3;
for j=1:3;
xd(j, i)=xd(j, i)/sd(i);
end
end
wd(1)=0;wd(2)=0;wd(3)=0;
fori=1:3;
for j=1:3;
wd(i)=wd(i)+xd(i, j);
end
wd(i)=wd(i)/3;
end
lambdamax=wd(1)*sd(1)+wd(2)*sd(2)+wd(3)*sd(3)
CI=(lambdamax-3)/2
CR=CI/0.58
disp(CR)
adjwa=wg(1)/(wg(1)+wg(2)+wg(3)+wg(4))
adjwb=wg(2)/(wg(1)+wg(2)+wg(3)+wg(4))
adjwc=wg(3)/(wg(1)+wg(2)+wg(3)+wg(4))
adjwd=wg(4)/(wg(1)+wg(2)+wg(3)+wg(4))
x=adjwa*wa(1)+adjwb*wb(1)+adjwc*wc(1)+adjwd*wd(1)
y=adjwa*wa(2)+adjwb*wb(2)+adjwc*wc(2)+adjwd*wd(2)
z=adjwa*wa(3)+adjwb*wb(3)+adjwc*wc(3)+adjwd*wd(3)
printf('composite weight for A %f\n', x)

```

```
printf('composite weight for B %f\n', y)
```

```
printf('composite weight for C %f\n', z)
```

Output of the result

Enter the comparison matrix for the criterion [1 5 3 7;1/5 1
1/3 5;1/3 3 1 6;1/7 1/5 1/6 1];

lambdamax at level 1 4.341409

consistency ratio for comparison of the factors wrt goal is
0.126448

lambdamax at level 2 for the wrt factor A **3.245477**

consistency ratio for comparison of the factors wrt factor A
is **0.211618**

lambdamax at level 2 for the wrt factor B **3.412413**

consistency ratio for comparison of the factors wrt factor B
is 0.355528

consistency ratio for comparison of the factors wrt factor C
is 0.2116180

consistency ratio for comparison of the factors wrt factor B
is 0.1132911

composite weight for A**0.666367**

composite weight for B**0.136193**

composite weight for C**0.197440.**

3. Conclusion

Input for the program will be only the comparison matrix entered by the user according to his/her preference. The comparison matrices for the alternative with respect to criterions will be given by the experts in the field. Here as the composite weight for A is more followed by C and B, the best alternative is A. The manual results and the programming result matches. However modifications in the program can be made according to the number of criterion and alternatives.

References

- [1] Saaty, T.L. (1977). A Scaling Method for Priorities in Hierarchical Structures, Journal of Mathematical Psychology, 15: 57-68.
- [2] Saaty, T.L. (1980). The Analytic Hierarchy Process. McGraw-Hill International, New York, NY, U.S.A.
- [3] Saaty, T.L. (1983). Axiomatic Foundations of the Analytic Hierarchy Process, Management Science, 32: 841-855.
- [4] Saaty, T.L. (1990). An Exposition of the AHP in Reply to the Paper 'Remarks on the Analytic Hierarchy Process', Management Science, 36: 259-268.
- [5] Saaty, T.L. (1994). Fundamentals of Decision Making and Priority Theory with the AHP. RWS Publications, Pittsburgh, PA, U.S.A.
- [6] Triantaphyllou, E., and Mann, S.H. (1989). An Examination of the Effectiveness of Four Multi-Dimensional, Decision-Making Methods: A Decision-Making Paradox. International Journal of Decision Support Systems, 5:303-312.
- [7] Triantaphyllou, E., Pardalos, P.M., and Mann, S.H. (1990a). A Minimization Approach to Membership, Evaluation in Fuzzy Sets and Error Analysis, Journal of Optimization Theory and Applications, 66: 275-287