Analytical Hierarchal Process for Complex Decisions

New Mexico

Supercomputing Challenge

Final Report

April 7, 2010

Team 112

Sandia Preparatory School

Team Member:

Caitlyn Scharmer

Teacher:

Mr. Neil McBeth

Project Mentors:

Minga Banks

Carol Scharmer

Executive Summary

The Analytical Hierarchal Process is important to most people because they can make decisions easily. Though there are programs available for people to use, it lacks its user-friendly qualities due to the fact that the criteria are pre-chosen for the user and the scales are restricted. For my program, it is quite simple to change the scale in the program, although I did choose a fairly common one and one that entails versatility.

My goal was to be able to change the criteria and college names easily. The scale is not as versatile, but it can be changed. The college names are prompted for the user, and it is quite simple for the user to go to the program and change it in the program. The next step would be to have the user be able to change the criteria and number of criteria, but for the nature of the process, it would have to be a square. But we decided on five for the program due to time constraints.

This program was done in C++ because it was easy to do mathematical models using the programmer. It was also easy because one of my mentors is an instructor of C++ at Regis University and could instruct me in how to write this program and could "debug" it with ease.

Problem Statement

As rational beings, we usually like to quantify variables and options to make objective decisions. Decision making is simple when the criteria are easily measured and consistent results are required. But what happens when the criteria cannot be measured, there are a significant number of criteria, the value of the criteria may vary with different individuals, or all of the above?

This Supercomputing project was to develop an easy-to-use, mathematical, multi-criteria, decision-making (MCDM) application, using the Analytical Hierarchal Process (or AHP). Although AHP programs exist and can be purchased, each are highly specific to a particular application and are not typically a household application – such as Microsoft Excel. As this paper explains the process, the reader will understand that the application could be used for many types of common decisions, such as deciding on an investment portfolio, deciding on which job to take, or choosing a laptop computer. The reader will also understand that the results of the application will vary from individual to individual. For example, if the AHP is used to choose a laptop computer, the results for a user who is a “gamer” will be much different that the results for a user who is a small business owner.

Introduction

For this project, I chose to write a C++ program implementing the AHP. To explain the process I decided to use selection of a college as an example. First, I will provide a brief background of the AHP method and how the process works. I will then describe how the program gets the criteria, sets up the matrices, does the calculations and presents the results. The code is provided as an attachment to this report. As of the writing of this paper, four of the five criteria is hard-coded with one criteria entered by the user. Also, the program only compares four colleges. The program includes several outputs that provide insight into the process. The next phase of the project would be to allow for additional criteria and alternatives to be compared and to improve the user interface.

Method – Background

The AHP process, also referred to as Pair-wise Comparison Method, was first introduced in 1977 by Thomas Saaty and orders objects with many attributes. AHP was developed with the intent of providing a simple and repeatable MCDM process. The process is really useful for criteria for which the alternatives are not easily quantified. The method derives both the weights for the criteria and, subsequently, calculates the ordered value of the alternatives based on sets of pair-wise comparisons. Although other scales may be applied, Saaty introduced the 9-grade value scale for the comparisons, which is provided below.

Scale of Relative Importance (where r: row; c: column)

1 Criteria r and c are of equal importance, and when r and c are the same

3 Criteria r is somewhat more importance than c

5 Criteria r is more important than c

7 Criteria r is slightly more important than c

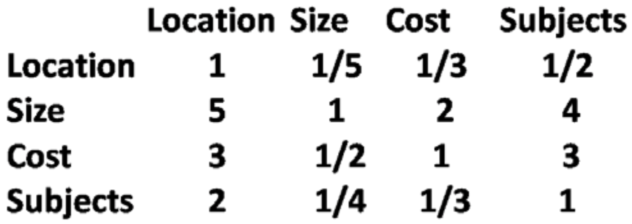
9 Criteria r is much more important than c

2,4,6,8 are intermediate values that can be used between adjacent judgments – when compromise is needed.

The reciprocal is used with Criteria c is more important than r

The AHP begins by listing the alternatives and the evaluation criteria. Then all evaluation criteria are compared to each other (pairs) and assigned a value for each comparison.

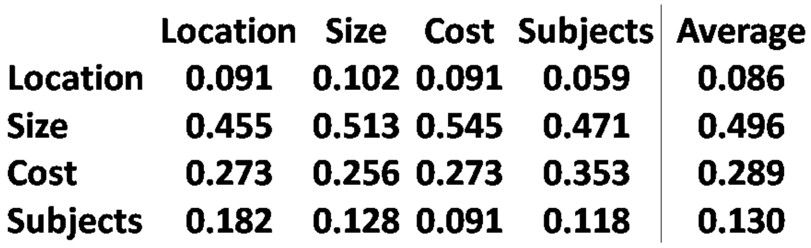
The easiest way to work with the assigned values is to log the value in arrays that look like the matrix below.



	Location	Size	Cost	Subjects
Location	1	1/5	1/3	1/2
Size	5	1	2	4
Cost	3	1/2	1	3
Subjects	2	1/4	1/3	1

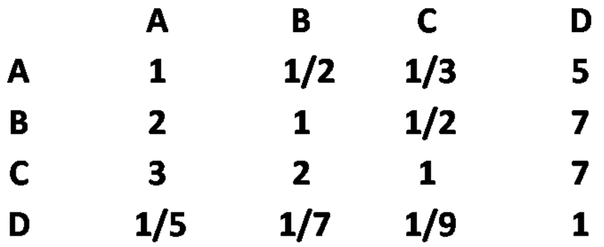
So in this example, Size is "strongly more important" than Location and is, therefore, logged as a 5. Additionally, once Location and Size have been compared, the upper portion of the matrix is be completed by logging the inverse of the corresponding comparison in the lower portion of the matrix.

Then, the weight for each criterion is calculated by taking each entry and dividing by the sum of the column (normalizing). Averaging across each row effectively corrects for any small inconsistencies in the decision making process. The resulting matrix is given below:



	Location	Size	Cost	Subjects	Average
Location	0.091	0.102	0.091	0.059	0.086
Size	0.455	0.513	0.545	0.471	0.496
Cost	0.273	0.256	0.273	0.353	0.289
Subjects	0.182	0.128	0.091	0.118	0.130

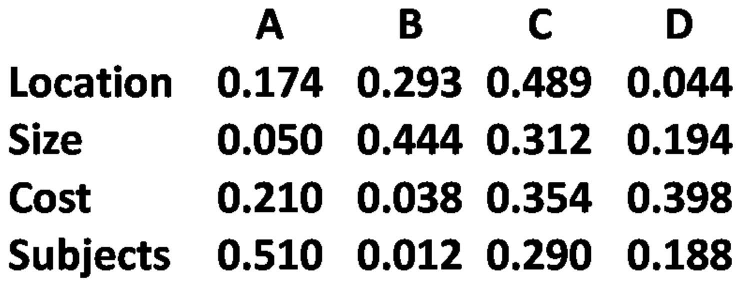
The next step is to compare the Location of colleges being considered. For this example we'll use colleges in: Anchorage (A), Boston (B), Chicago (C) and Denver (D). Now, as done for the criteria, compare the Location for each pair of colleges using the scale and log the value in arrays as illustrated in the matrix below.



	A	B	C	D
A	1	1/2	1/3	5
B	2	1	1/2	7
C	3	2	1	7
D	1/5	1/7	1/9	1

As with the criteria, the columns are normalized (divide by the sums of the columns and average across rows to get the values for each college with regards to location.)

Next, compare the Size, Cost and Subject for each pair of colleges the college pair-wise comparisons and calculations for the other criteria. For example purposes the following values are provided.



	A	B	C	D
Location	0.174	0.293	0.489	0.044
Size	0.050	0.444	0.312	0.194
Cost	0.210	0.038	0.354	0.398
Subjects	0.510	0.012	0.290	0.188

Using the weights determined from the first pair-wise comparison matrix and the example values for Location, Size, Cost and Subjects the value for Anchorage is calculated:

$$(.174)(.086) + (.050)(.496) + (.210)(.289) + (.510)(.130) = .164$$

Similarly, the values are calculated for: Boston as .256, Chicago as .335, Denver as .238, So, Chicago it is.

Method – Program

Although the project is to write a generic AHP program, for clarity the program was written with specific criteria and alternatives for selection of a college. The project does, however, allow for the user to enter an additional criterion and to enter the names of the colleges. The program is attached to the end of this report. Note that sections for debugging the program, have been retained in the program.

Of course, the program must be properly set up, the integer and non-integer constants and variable, arrays and strings are established.

The program then outputs a description of the program and proceeds to provide options and request data input: display option, names of the colleges, the "user choice" criteria and provides the directions for the comparisons using, Saaty's 1-9 graded value scale. The program populates the arrays as the data is entered. A difficult portion of the programming was to ensure that there were no duplications in the comparisons when populating the array with the appropriate reciprocals. The program then performs the calculations to normalize the array and average each row, obtaining the weights for the criteria.

The bulk of the program is for the pair-wise comparison of the colleges with respect to each of the 5 criteria. The user is requested to enter the graded value for each pair of colleges compared. This is completed 5 times for each criterion. The program populates the arrays as the data is entered. The program then performs the calculations to normalize the array, obtaining the value for each college.

Then those numbers in the second normalized array (which are indicated by the column/row of criteria and the column/row of colleges) are multiplied against the weight of the location, which is the average of the normalized array. This is done to every college using the two normalized arrays.

The program outputs a list of the colleges with the calculated value. The highest value is the preferred choice.

Results and Conclusions

The program successfully ranks the alternatives based on the weighted values of the established criteria and have been specifically applied to selection of a college. The program can be modified to allow the user to not only input the criteria but define number of criteria and alternatives to be used in the decision. Having several people use the program also demonstrates that the results will be as individual as the individual entering the data.

Acknowledgements

I would like to thank Mr. McBeth for his unfailing support throughout the year throughout all activities of Supercomputing Challenge. I’d like to thank my mom, Carol Scharmer, for introducing me to and helping me understand the AHP. And, I’d like to give special thanks to my aunt, Minga Banks a C++ instructor at Regis College, who was dedicated to helping me learn C++ and to illustrate how to implement debugging code when the results just would not display correctly.