International Journal of Computer Applications (0975 – 8887)
Volume 36– No.11, December 2011

A Qualitative Assessment of Educational Software

Vandana Bagla
Maharaja Agrasen Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India.

Anjana Gupta
Maharaja Agrasen Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India.

Deepika Kukreja
Maharaja Agrasen Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India.

ABSTRACT
With the ever-growing number of choices of educational software for children scholastic programs, the task of choosing software can be a nerve-wracking one. As computer use becomes more substantial in home and classroom learning, the selection of software endeavors even more importance. The key aspects for providing a better learning experience lies in choosing software that successfully combines education and entertainment. This article inspects the prerequisite of children's software evaluation in the light of dynamic nature of edutainment perspective. To obtain empirical evidence of pupils' performance, choosing software can be productive, if it accede a set of well accomplished criteria. Key issues are discussed such as the ways to evaluate the appropriateness of software for children and the most efficient means of utilizing this information. This paper utilizes Rank Order Centroid (ROC) Methodology as well as Ratio Method to confront the concerns raised by academics, instructional designers and faculty administration about the teaching/learning software delivered via Information Technologies. The employed methodology utilizes systematic approach to gauge and ultimately select the most suitable software. The factors considered here scout Technical as well as Non-Technical aspects of the problem.

Keywords
Multi Criteria Decision Making (MCDM), Rank Order Centroid (ROC) Methodology, Ratio Method, Audio-Visual (AV), Trademark Electronic Search System (TESS).

1. INTRODUCTION
Software industry is gearing up to be one with high growth potential and forgoing supply and demand of talent makes. The task of evaluating an interactive electronic experience with some degree of reliability and validity presents a unique set of challenges which were previously manipulated traditionally. Because the software experience is an interactive and multi-dimensional one, the evaluation of children's software must look at the pedagogy used in its development as well as design features. Decision analysis looks at the paradigm in which an individual decision maker (or decision group) contemplates a choice of action in an uncertain environment.

The first and largest systematic software evaluation effort was conducted by the Educational Products Information Exchange (EPIE). Since 1967, the non-profit group has been reviewing textbooks, Audio-Visual (AV) materials and other educational resources. In the 1980s, the group started reviewing software, and kept the reviews in the form of a database known as Trademark Electronic Search System (TESS). It was available from 1983 to 1993 in print form and on CD-ROM (Compact Disc, Read-Only-Memory) in subsequent years. EPIE was one of the first groups to apply the searchable database as a means of helping educators find specific products.

In the early 1980s, the romance and novelty of computers software led to a flurry of software evaluation activity. Lathrop & Goodson [8], Rucker et al. [9] and Jones & Vaughan [7] did preeminent work in the said field.

The 1980s also saw the first standardized evaluation instruments, designed for children's software testing. Buckleitner [2] and Haugland & Shade [5] remarkably promoted the domain. These checklist-based software evaluation forms were an attempt at quantifying the factors in software design that have been associated with the effectiveness of software products for young children.

In 1996, the first review databases became available online. Amazon.com, the online bookstore, demonstrated the potential for a commercial scenario, for merging evaluation information with the purchasing process. These Internet-related technologies have helped to lay the blueprint for the futuristic software evaluation information in the 21^{st} century.

More studies need to be accomplished in this field, as that done by Escobedo & Evans [4], where the ratings assigned by the published software methods are compared with actual child selection or Tammen & Brock [13], where middle school students are asked to identify issues they feel are important for the evaluation of software programs.

Key indices for emerging software market crave the desideratum for the methodology which can comprehensively accomplish the requirements of children software acquisition. A general opinion database containing relevant questionnaires was crafted and surveys were conducted in educational institutions for searchable criteria exploring children software requirements. Issues concerning the regulation of conducted surveys to ordain the exigency were submitted to Multi Criteria Decision Making (MCDM). The results were accomplished using two methodologies namely Rank Order Centroid (ROC) Method and Ratio Method. Codes for these two methodologies were developed and comparative results have been endowed.

The paper is organized as follows. Section 1 is introductory. Section 2 provides an overview of the proposed methodologies. A hierarchal structure is designed to model the problem mathematically in section 3. Section 4 explains the implementation of the introduced methodologies to establish the results. Section 5 illustrates the methodologies via an example. Section 6 gives an empirical comparison of the prescribed methodologies. The paper concludes for further application in investment mechanism and many other important fields in section 7.

2. INTRODUCTION TO MULTI CRITERIA DECION MAKING (MCDM)

MCDM is a sub-discipline of decision sciences that explicitly considers multiple criteria in decision-making environments. Whether in our daily lives or in professional settings there are typically multiple conflicting criteria that need to be evaluated in decision making. For casual resolutions, we usually weigh multiple criteria implicitly and we may be comfortable with the consequences of such decisions that are made intuitively. On the other hand, when stakes are high, it is important to properly structure the problem and explicitly evaluate the engrossed criteria to gear the predicament. Ignizio [6], Serafini [10], Serafim [11] and Steuer [12] have utilized MCDM in various application scenarios. A typical MCDM problem is cited in Figure 1.

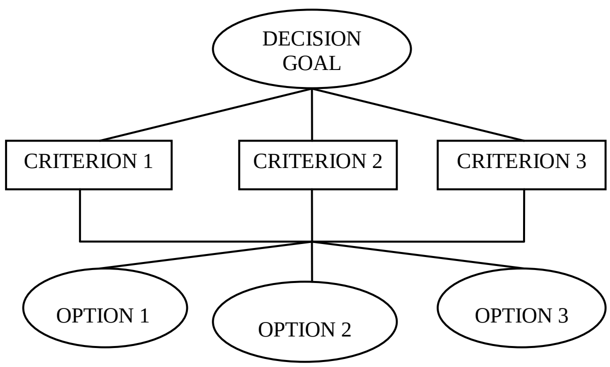


Figure 1: MCDM problem

The Weighted Decision Approach consists of selecting a few important issues relating project goals and rating their importance in achieving project objectives. Many formal procedures have been developed in management science that facilitate this process and make it more objective. This paper describes two such methods that combine subjective and objective methods to arrive at weighted values for the key selection factors. These methods have been selected for their simplicity and effectiveness. These methods help decision-makers to develop more consistent and transparent weights. The two methods described here are: (1) Rank Order Centroid Method and (2) Ratio Method which are commonly used in elementary statistics for averaging purposes.

2.1 Overview of Rank Order Centroid (ROC) Methodology

The term "rank order centroid" was coined by F. H. Barron and B. E. Barrett [1], who also argued for its use in multi-attribute decision problems. The idea is to convert ranks (first, second, third, fourth) into values that are normalized on a 0.0 to 1.0 interval scale. An obvious way to achieve this is to assume each rank is distributed evenly within the unit interval. So, first ≈ 0.80, second ≈ 0.60, third ≈ 0.40, and fourth ≈ 0.20. But ranks really emphasize on rating data sequentially as: 1st/4, 2nd/4, and so forth. And it is credible from elementary statistics that rate data is best handled using harmonic techniques. For example the average of 30 mph and 60 mph over a fixed distance is not (30 + 60) / 2 = 45 mph, but rather 2 / (1/30 + 1/60) = 40 mph. Notice that calculations for ROCs are conceptualized on the similar pattern emphasized above. Crain [3] used ROCs for multi-attribute weight determination in his dissertation.

This method is a simple way of giving weight to a number of items ranked according to their importance. The decision-makers usually can rank items much more easily than give weight to them. This method takes those ranks as inputs and converts them to weights for each of the items. The conversion is based on the following formula:

$w_i = \frac{1}{m} \sum_{n=i}^{m} \frac{1}{n}$, where '*m*' denotes the number of items and 'w_i' is the weight for i[th] item.

For example, if there are 4 items, the item ranked first will be weighted (1 + 1/2 + 1/3 + 1/4) / 4 = 0.521, the second will be weighted (1/2 + 1/3 + 1/4) / 4 = 0.271, the third (1/3 + 1/4) / 4 = 0.146, and the last (1/4) / 4 = 0.062.

An outright code of the above methodology is given in APPENDIX (A).

2.2 Overview of Ratio Method

The Ratio Method is another simple way of calculating weights for a number of critical factors. A decision-maker should first rank all the items according to their importance in the preferred domain. The next step is giving weight to each item based on its rank in the interval [10, 90]. Here lowest ranked item will be given a weight of 10 and rests of the items are rated in multiples of 10 based on the preferences given by decision maker. For example if item 2 is five times more important to item 1 (the lowest ranked item), then item 2 is provided with a rating 50. The last step is normalizing these raw weights as proposed by Weber and Borcherding [14]. This process is shown in the example below. Note that the weights should not necessarily jump 10 points from one item to the next. Any increase in the weight is based on the subjective judgment of the decision-maker and reflects the difference between the importances of the items. Ranking the items in the first step helps in assigning more accurate weights. For example, if there are 4 items ranked successively with the priorities 50, 40, 20 and 10 respectively, then the normalized respective weighted score of each item will be 0.417, 0.333, 0.167, and 0.083 respectively. Normalized weights are simply calculated by dividing the raw weight of each item over the sum of the weights for all items. For example, normalized weight for the first item is calculated as 50/ (50 + 40 + 20 + 10) = 0. 417. The sum of normalized weights is equal to 1 (0.417 + 0.333 + 0.167 + 0.083 = 1).

Ratio method can be easily applied in single- and multi-dimensional MCDM problems. An advantage of this method is that instead of the actual values it can use relative ones.

It is conspicuous to note here that Ratio Method entrusts more power in the hands of decision maker to prioritize the attributes quantitatively. The integrated code of the methodology is given in APPENDIX (B).

3. PROBLEM STATEMENT

Checklists for software evaluation were crafted on the basis of the conducted survey results which are probably adequate and flexible enough to serve the different purposes and intentions of evaluation. The conventional approach to predictive evaluation is to use a checklist exploring technical and non-technical features of the requisite software. Technical features include:

- *Robustness (T_1)*: The software should be capable of dealing with vast quantities of data without crashing.
- *Support (T_2):* Imperative support should be provided not only initially as training, but also in future if things go wrong by ringing a help desk.
- *Performance (T_3):* The software package must be sufficiently contemporary to furnish prevailing requirements.
- *Portability (T_4):* The package should accede to export prerequisites for data transfer if needed.
- *Compatibility (T_5):* The new software should run on the existing hardware and under the existing operating system.
- *Cost/benefit (T_6):* The package should comply with value for money standards.

For Non–Technical aspects, the overall emphasis is on educational issues:

- *Feature Clarity (N_1)*: Features should be self explanatory and user friendly.
- *Creativity (N_2)*: Package should stimulate thinking ability of the children.
- *Interactive Learning (N_3):* The software package must be sufficiently equipped to propagate learning atmosphere.
- *Problem Solving (N_4)*: The package should encourage problem solving techniques in impressive ways.
- *Edutainment (N_5)*: It should be able to inculcate knowledge using play way methods.

The problem of the evaluation of alternatives submitted to a multi-criteria decision analysis is a contentious task. Usually there is no optimal solution as no alternative is the best one on each criterion. Problem is to evaluate the software according to their credibility on the set of weighted judgment criteria.
The most demanding phase in designing the evaluation model is to structure the decision problem. The goal is to recon the software according to a set of criteria for evaluation.
Figure 2 gives the hierarchical structure of the problem conceded to MCDM.

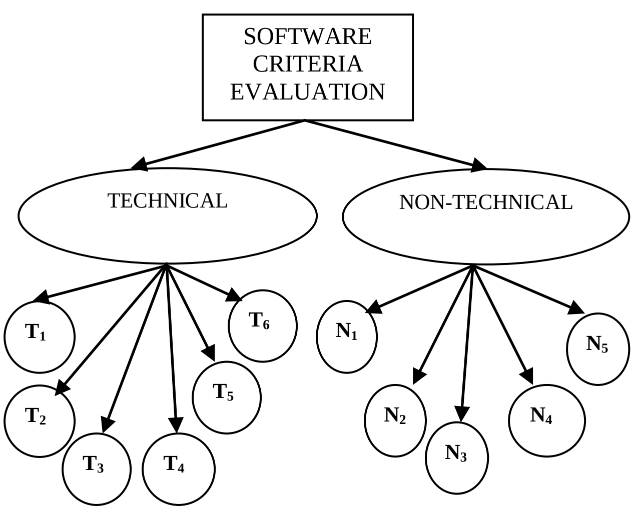


Figure 2: Hierarchical structure of the MCDM problem

4. SOLUTION PROCEDURE

4.1 Rank Order Centroid (ROC) Method

To evaluate the above hierarchy using ROC methodology (introduced in section 2.1), decision makers are asked to allot rankings to the leveled criteria according to their requisite priorities. Using ROC coding, these ranks (such as first, second, third) are converted into ratings or weights, which are numeric values (0.611, 0.278, 0.111).
For this example, we begin by looking at our top-level attributes: Technical and Non-Technical. We rank them from most important to least important. Then we compute the ROC for each of these two high-level attributes.
The process is repeated for sub-criteria at Technical and Non-Technical level exclusively. This exercise provides local priority weights to criteria at each level.
The nodes at each level are compared pairwise with respect to their contribution to the nodes above them to find their respective global weights. We rank each of the criterion in the final set by evaluating it with respect to upper level attributes separately. The evaluation process finally generates the global weights for each requisite criterion of interest.
The procedure can be extended to another level to assign prioritized weightings to the alternatives.

4.2 Ratio Method

Proposed methodology intends to authorize more power to decision maker while allotting priorities to decision criteria. For allotting weights using Ratio Method (acquainted in section 2.2), decision makers are asked to assign subjective judgments to the given set of criteria in multiples of 10 in the interval [10, 90].
For the given hierarchy (Figure 2), we begin by looking at our top-level attributes viz. Technical and Non-Technical. If decision maker says, Technical is 5 times more important to Non-Technical part, then we rate [Technical: Non-Technical] $\approx$ [50: 10] Now we compute the comparative ratings for each of these two high-level attributes as:

$$\left[\ \frac{50}{50+10}, \frac{10}{50+10}\ \right] \approx [\ .833, .167\ ]$$

The procedure is repeated for sub-criteria at Technical and Non-Technical level exclusively. This exercise provides local priority weights to criteria at each level. The process of finding global weights for each criterion is same as discussed for ROC Method in section 4.1.

5. ILLUSTRATION

We now illustrate the proposed methodologies via an example in which three IT companies say A_1, A_2 and A_3 are providing software befitting requirements of an educational institution. The important criteria at Technical as well as Non-Technical levels have already been discussed in section 3. We have identified eleven criteria in all .to be employed in software evaluation. Table 1 depicts the specifications of the three alternatives negotiated subjectively as Poor (P), Average (A), Good (G), Very Good (VG) and Excellent (E) with respect to the requisite criteria.

Table 1. Subjective data for software evaluation

Criteria ↓	Alternatives		
	A_1	A_2	A_3
Robustness	P	G	VG
Support	E	A	P
Performance	P	G	V
Portability	E	A	P
Compatibility	**G**	**VG**	**A**
Cost/benefit	G	P	E
Feature Clarity	G	A	G
Creativity	P	E	A
Interactive Learning	A	P	VG
Problem Solving	E	P	A
Edutainment	VG	VG	P

The scenarios for this survey analysis were crafted after a study of the responses from our previous surveys and of the predictions made to work out the hierarchy (Fig 2) using the given methodologies. For this purpose, subjective ratings of the alternatives corresponding to various criteria are needed to be converted to numeric values to facilitate the decision making.

5.1 Software Evaluation Using ROC Method

To evaluate the hierarchy (Figure 2), Decision Makers were asked to write each criterion on a sticky note and arrange them on a wall or desk and reorder with the most important on top. This is best done in a pairwise fashion by selecting the criteria two at a time and asking, “If an alternative could meet only one of these, which criteria would I choose?” Then, move the chosen one to the top and the other to the bottom of the arrangement.

Decision Makers prioritized level 1 criteria viz. Technical and Non-Technical as first and second respectively. Using ROC coding, assigned numeric weights are 0.75 and 0.25 respectively. Similarly level two criteria are rated subject to prioritized responses inquisition. The sequential order provided by the authorized persons was (Compatibility, Cost/benefit, Support, Performance, Robustness, and Portability) for Technical specifications and (Edutainment, Feature Clarity, Creativity, Problem Solving, Interactive Learning) for Non-Technical blueprint. Table 2 shows the local and global priority weights calculated using ROC code given in APPENDIX (A). Here global priority weights for each one are emphasized bold to differentiate them from local priority weights.

Table 2. Objective ratings for software evaluation

ROC Calculations For Criteria At Different levels					
Technical (0.75)			Non-Technical (0.25)		
Robustness	.061	**.046**	Feature Clarity	.256	**.064**
Support	.158	**.119**	Creativity	.157	**.039**
Performance	.103	**.077**	Interactive Learning	.040	**.010**
Portability	.028	**.021**	Problem Solving	.090	**.023**
Compatibility	.408	**.306**	Edutainment	.457	**.114**
Cost/benefit	.242	**.182**			

Now to rank the alternatives using ROC methodology, fuzzy measurement scale is developed to provide crisp scores to subjective ratings given in Table 1. It is noteworthy to mention here that any standard fuzzy scale may serve the purpose yet we are using ROC Methodology to develop the said scale.

Using ROC code, ratings provided to P, A, G, VG, & E are .040, .090, .157, .256 & .457 respectively. The global weight of each alternative subject to a given criterion is calculated by multiplying criterion’s global weight to its local weight. Table 3 elucidates local and global (final) numeric ratings of the subjective data given in Table 1. Again global priority weights for each alternative are emphasized bold for specification

Table 3. Alternatives’ priority ratings by ROC Method

Criteria ↓	**Alternatives**					
	A_1		**A_2**		**A_3**	
Robustness **(.046)**	.040	**.002**	.157	**.007**	.256	**.012**
Support **(.119)**	.457	**.054**	.090	**.011**	.040	**.005**
Performance **(.077)**	.040	**.003**	.157	**.012**	.256	**.020**
Portability **(.021)**	.457	**.010**	.090	**.002**	.040	**.001**
Compatibility **(.306)**	.157	**.048**	.256	**.078**	.090	**.028**
Cost/benefit **(.182)**	.157	**.029**	.040	**.007**	.457	**.083**
Feature Clarity **(.064)**	.157	**.010**	.090	**.006**	.157	**.010**
Creativity **(.039)**	.040	**.002**	.457	**.018**	.090	**.004**
Interactive Learning **(.010)**	.090	**.0009**	.040	**.0004**	.256	**.003**
Problem Solving **(.023)**	.457	**.011**	.040	**.001**	.090	**.002**
Edutainment **(.114)**	.256	**.029**	.256	**.029**	.040	**.005**
Score	0.199		0.171		0.173	
Normalized Score	0.367		0.315		0.319	

The decision maker can then obtain a total score for each alternative simply by summing these global weights over all the attributes. Scores are then normalized by dividing each individual score by total score. Normalization facilitates comparative evaluation on a common percentile scale.

5.2 Software Evaluation Using Ratio Method

To evaluate the same problem using Ratio Method, Decision Makers were asked to assign prioritized ratings to leveled attribute so that ratio scale measurements may be developed for criteria at different levels of hierarchy. Decision Makers characterized Technical aspects five times prior to Non-Technical ones. So respective weighing to the attributes was (0.833, 0.167) using Ratio Method coding given in APPENDIX (B). For level 2 Technical criteria viz. (Compatibility, Cost/benefit, Support, Performance, Robustness, Portability), the authorized response was designated as (90, 80, 70, 50, 20, 10). For Non-Technical specifications viz. (Edutainment, Feature Clarity, Creativity, Problem Solving, Interactive Learning), adjudged ratio scale values were (90, 70, 60, 40, 10).
Table 4 depicts the calculated local and global priority weights using Ratio Method.

Table 4. Objective weightings for software evaluation

Ratio Scale Calculations For Criteria At Different levels					
Technical (0.833)			Non-Technical (0.167)		
Robustness	.063	**.053**	Feature Clarity	.259	**.043**
Support	.219	**.182**	Creativity	.222	**.037**
Performance	.156	**.130**	Interactive Learning	.037	**.006**
Portability	.031	**.026**	Problem Solving	.148	**.025**
Compatibility	.282	**.235**	Edutainment	.333	**.056**
Cost/benefit	.250	**.208**			

A fuzzy scale (.037, .111, .222, .296, .333) was fostered for (P, A, G, VG, E) against authorized prioritized responses (10, 30, 60, 80, 90). Table 5 shows final scorings using Ratio Method.

Table 5. Alternatives' priority ratings via Ratio Method

Criteria ↓	**Alternatives**					
	A$_1$		**A$_2$**		**A$_3$**	
Robustness **(.053)**	.037	**.002**	.222	**.012**	.296	**.016**
Support **(.182)**	.333	**.061**	.111	**.020**	.037	**.007**
Performance **(.130)**	.037	**.005**	.222	**.029**	.296	**.039**
Portability **(.026)**	.333	**.009**	.111	**.003**	.037	**.001**
Compatibility **(.235)**	.222	**.052**	.296	**.070**	.111	**.026**
Cost/benefit **(.208)**	.222	**.046**	.037	**.008**	.333	**.069**
Feature Clarity **(.043)**	.222	**.010**	.111	**.005**	.222	**.010**
Creativity **(.037)**	.037	**.001**	.333	**.012**	.111	**.004**
Interactive Learning **(.006)**	.111	**.0007**	.037	**.0002**	.296	**.002**
Problem Solving **(.025)**	.333	**.008**	.037	**.001**	.111	**.003**
Edutainment **(.056)**	.296	**.017**	.296	**.017**	.037	**.002**
Score	0.212		0.177		0.179	
Normalized Score	0.373		0.312		0.315	

6. ROC METHOD VS RATIO METHOD

Rank order centroids decompose a problem very much like Ratio Method. The key advantage of ROC Methodology is its simplicity in surveying whereas Ratio method requires quantified rating of prioritized alternatives. Yet Ratio Method has its own advantages when the decision makers influence their prioritized specifications regarding significances. In the above illustration, it is unlocked that our ROC results correlated almost perfectly with the Ratio Method results. It was an obvious observation as the quantified ratings provided in Ratio Method for level 1 criteria viz. Technical and Non-Technical were almost commensurable to that in ROC methodology. A compelling difference may evolve in results shown by two methodologies if criteria ratings in Ratio Method are out rightly antithetic.

7. CONCLUDING REMARKS

Objectivity in educational research is not a new concept, nor are the basic ideas of validity and reliability in measurement. As we move into the 21st century, our children deserve rigorous, well-constructed evaluative methods applied to the practices that are subject to public criticism and evaluation. When computer-savvy educators have access to this kind of solid evaluative information, they can plan and teach more effectively and tap the power and excitement that computers can bring to the learning process. The implementation of new technologies is necessarily a good investment of resources in preparing children for life in a technologically saturated workforce. The methods described here are capable of helping educational institutions for gaining a deeper understanding of professional decisions they face and reduce their initial state of uncertainty about the best course of action. The key to making these methods useful in this competitive scenario is making sure they both meet the needs of the intended users and provide substantive decision making support.
The proposed methodologies could be applied to materialize the selection of effective alternative in sustainable development, structures, technologies, investments, etc. The future work will include incorporation of this approach into practical software applications i.e. case studies will be processed and evaluated. The detailed evaluation will be obtained through the results of

these case studies and comparison of this approach with other similar approaches.

8. REFERENCES

[1] Barron, F.H. and Barrett, B.E. 1996. Decision Quality Using Ranked Attribute Weights. Management Science, 42(11), 1515-1523.

[2] Buckleitner, W. 1985. A Survey of Early Childhood Software (1984 to 1993). High/Scope Educational Research Foundation, Ypsilanti, MI.

[3] Crain, W.F. 2003. Multi-attribute Weight Determination: Elicitation & Approximation. Dissertation, George Mason University, Spring 2003.

[4] Escobedo, Theresa H. and Evans, Sharon 1997. A Comparison of Child-Tested Early Childhood Education Software with Professional Ratings. Paper presented at the Annual Meeting of the American Education Research Association (Chicago, IL, March 28, 1997). ED411060.

[5] Haugland, Susan W. and Shade, Daniel D. 1988. Developmentally Appropriate Software for Young Children. Young Children; 43(4), 37-43.

[6] Ignizio, J.P. 1982. Linear Programming in Single and Multi-objective Systems, Prentice-Hall, Englewood Cliffs, New Jersy.

[7] Jones, Nancy Baker, Ed. and Vaughan, Larry, Ed. 1983. Evaluation of Educational Software: A Guide to Guides. Southwest Educational Development Lab., Austin, TX.; Northeast Regional Exchange, Inc., Chelmsford, MA ED237064.

[8] Lathrop, A. and Goodson, B. 1983. Courseware in the Classroom: Selecting, Organizing, and Using Educational Software. Menlo Park, CA: Addison-Wesley.

[9] Rucker, Chauncy N. and Others 1985. The Connecticut Special Education Network for Software Evaluation.

[10] Serafini, P.1985. Mathematics of Multi Objective Optimization, Springer-Verlag, NewYork.

[11] Serafim O. 2007. A fuzzy compromise solution for multicriteria problems, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 15(3), 363–380.

[12] Steuer, R.E. 1986, Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley and Sons, New York.

[13] Tammen, Jill; Brock, Laurie.1997. CD-ROM Multimedia: What Do Kids Really Like? MultiMedia Schools; 4(3), 54-56, 58-59.

[14] Weber, M., Borcherding, K. 1993. Behavioral influences on weight judgments in multiattribute decision making. European Journal of Operations Research , 67,

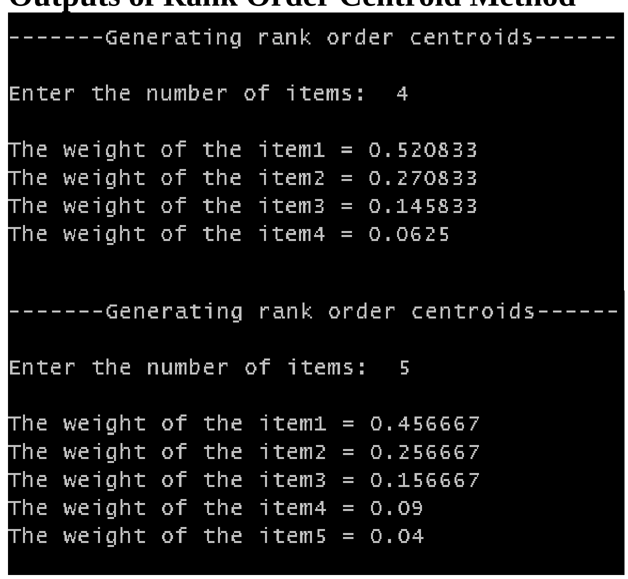
1-12.

APPENDIX (A)

Code for Rank Order Centroid Method

```
#include <iostream.h>
#include <conio.h>
double roc(int,int);
void main()
{
int n;
cout<<"-----Generating rank order centroids----";
cout<<"\nEnter the number of items: ";
cin>>n;
for(int i = 1; i<=n;i++)
cout<<"\nThe weight of the item"<<i<<" = "<<roc(n,i);
getch();
}
double roc(int n,int i)
{
double result = 0.0;
for(int j = i;j<=n;j++)
result = result + (1.0/j);
return result/n;
}
```

Outputs of Rank Order Centroid Method



```
-------Generating rank order centroids------

Enter the number of items:  4

The weight of the item1 = 0.520833
The weight of the item2 = 0.270833
The weight of the item3 = 0.145833
The weight of the item4 = 0.0625


-------Generating rank order centroids------

Enter the number of items:  5

The weight of the item1 = 0.456667
The weight of the item2 = 0.256667
The weight of the item3 = 0.156667
The weight of the item4 = 0.09
The weight of the item5 = 0.04
```

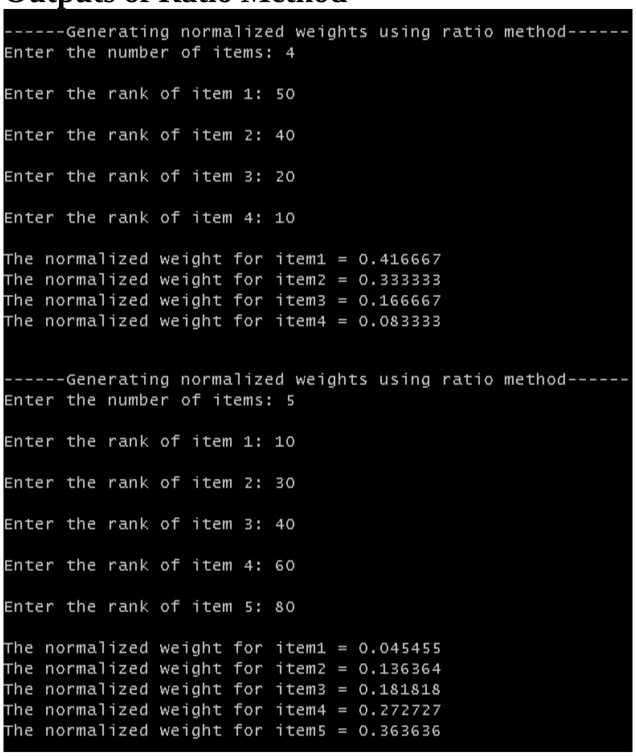
APPENDIX (B)

Code for Ratio Method

```
#include <iostream.h>
#include <conio.h>
void main()
{
clrscr();
float n, rank[100];
float n_weights[100];
float total=0;
cout<<"------Generating normalized weights using ratio method-
-----";
cout<<"\nEnter the number of items: ";
```

```
cin>>n;
for(int i = 1; i<=n;i++)
{
cout<<"\nEnter the rank of item"<<i<<":";
cin>>rank[i];
total = total + rank[i];
}
for(int j = 1;j<=n;j++)
{
n_weights[j] = (rank[j]/total);
cout<<"\nThe normalized weight for item"<<j<<" =
"<<n_weights[j];
}
getch();
}
```

Outputs of Ratio Method



```
------Generating normalized weights using ratio method------
Enter the number of items: 4

Enter the rank of item 1: 50

Enter the rank of item 2: 40

Enter the rank of item 3: 20

Enter the rank of item 4: 10

The normalized weight for item1 = 0.416667
The normalized weight for item2 = 0.333333
The normalized weight for item3 = 0.166667
The normalized weight for item4 = 0.083333


------Generating normalized weights using ratio method------
Enter the number of items: 5

Enter the rank of item 1: 10

Enter the rank of item 2: 30

Enter the rank of item 3: 40

Enter the rank of item 4: 60

Enter the rank of item 5: 80

The normalized weight for item1 = 0.045455
The normalized weight for item2 = 0.136364
The normalized weight for item3 = 0.181818
The normalized weight for item4 = 0.272727
The normalized weight for item5 = 0.363636
```