NEW WEIGHTING PROCEDURES MINIMIZING JUDGMENTAL ERROR AND REFINING INCONSISTENCY FOR MULTIPLE CRITERIA DECISION MAKING PROBLEMS

by

KOOK JIN NAM

Bachelor of Science
Korea Military Academy
Seoul, Korea
1975

Master of Science
Naval Postgraduate School
Monterey, California
1984

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1990

Thesis
1990D
N174n
cop. 2

Oklahoma State Univ. Library

NEW WEIGHTING PROCEDURES MINIMIZING JUDGMENTAL
ERROR AND REFINING INCONSISTENCY FOR
MULTIPLE CRITERIA DECISION
MAKING PROBLEMS

Thesis Approved:

M. Palmer Terrell
Thesis Adviser

Michael Branson

Kenneth E Case

Joe H. Mize

C M Wade

Dean of the Graduate College

1375595

PREFACE

The objective of this study is to develop new weighting methods for use in solving multiple criteria decision making problems.

I wish to express sincere appreciation to my major adviser, Dr. M. Palmer Terrell, for his guidance, assistance, and encouragement throughout this research and during my doctoral studies. Appreciations also to my committee members, Dr. Michael H. Branson, Dr. Kenneth E. Case, Dr. Joe H. Mize, and Dr. William D. Warde, for their interest and assistance.

I also wish to thank the School of Industrial Engineering and Management at Oklahoma State University for financial support.

Thanks are extended to Republic of Korean Army Headquarters for their financial and moral support, and for giving me the opportunity to fulfill this study.

Finally, I wish to dedicate this dissertation to my parents, Mr. and Mrs. Hyodeok Nam, my wife, Eunja Kim, and my children Bomi and Jaeho, for their prayers, sacrifice, understanding, encouragement, and love.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

Table | Page

Table | Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The General Problem

Weighting procedures have been used since the beginning of human life. Humans use some kind of weighting procedure, implicitly or explicitly, whenever they have need to allocate resources among a set of activities or to select the most important activity.

In recent history, many researchers have contributed their efforts for developing methods of weight determination. In general, weight determination methods are concerned with determining the preference of decision makers. Because of the nature of this problem and its breadth of application, an interdisciplinary interest has been developed in this area. In particular, the problem has been studied by economists, engineers, environmentalists, management scientists, mathematicians, operations researchers, statisticians, system analysts, urban planners, etc.

The importance of generating better weights for multiple criteria decision making (MCDM) problems continues to be of much interest to researchers and decision makers alike. The research interest in this area stems from both its simplicity of use in additive models and its applicability to problems in many diverse fields.

Statement of the Problem

Introduction

One of the purposes of deriving weights is for their use in additive models. Due to their simplicity, additive weight methods have great appeal in MCDM problems (Frazelle, 1985). It is important to study the weighting determination procedures closely and determine and understand the strengths and the weaknesses of the procedures. Research effort and direction can be motivated through such an analysis.

Theoretical Validation, Quality, and Simplicity

Many techniques for MCDM problems use weights to combine attributes into a single sum that indicates value or suitability. The most frequently applied multiple criteria decision rule is the weighting summation or linear model:

$$V_k = \sum_{i=1}^{n} W_i X_{ik} \qquad (1.1)$$

where V_k = value of the suitability of alternative k;

X_{ik} = the level of criterion i for alternative k;

W_i = the true weight of criterion i.

Many researchers have contributed their efforts to the development of better methods for determining the values of W_i. As a result of these researchers' effort, many methodologies have been developed from simple methodologies such as the ranking method, rating method, point allocation method, or unit weighting method to more sophisticated methodologies such as successive paired comparison method, indifference trade-off method, and eigen-vector method. Although the relatively easy models such as ranking method, rating method, point allocation method, and unit weight method are simple to use, they do lack formal theory. To be a theoretically valid model, the decision maker's trade-off should be reflected when comparing the criteria to each other (Fischer, 1977) (Hobbs, 1979). Theoretically the most defensible methods are those such as successive paired comparison methods and indifference trade-off methods, but they are the most complicated methods to use. Unfortunately, there is no guarantee that a theoretically valid method generates more superior weights than those generated using theoretically invalid methods (Einhorn and Hogarth, 1975). The purpose of the research to be presented in this paper is to contribute to the development of new methods which are theoretically valid, more superior in their use compared to other methods, and more easy to use.

Consistency Assumptions and Inconsistency

Since weights are difficult to estimate directly, researchers estimate these weights by using ratios of one criterion to another obtained through interaction with the decision maker. The comparisons used to construct the ratios may or may not be consistent. The necessary judgment used in making comparisons is dependent on many factors, such as personal experience, learning, situations, the state of mind, etc. The consistency assumption for comparisons is very critical. For instance, the main difference of various eigen-vector methods (more completely discussed in the literature review) developed by Saaty (1977), Cogger and Yu (1985), Takeda, Cogger, and Yu (1987) is the assumption of consistency. Saaty (1977) assumes that decision makers are consistent in their comparisons. Other researchers, however, do not agree with this consistency assumption because they believe most decision makers are going to be somewhat inconsistent, even after repeated attempts to alert them to their inconsistencies and attempts to refine the estimated reciprocal portion of the matrix. With this argument, they have devoted their research efforts to refining decision makers' inconsistencies in pairwise comparisons.

It does not really matter which eigen-vector method is used when the response of the decision maker is consistent in the pairwise comparisons, because they will give the same solution. This aspect demonstrates a need for developing methods which refine decision makers' inconsistencies in an appropriate and better way.

Minimization of Judgmental Error

Minimization of judgmental error is a new and important concept when estimating weights using subjective approaches. Due to a decision maker's inconsistency, knowledge, interest, state of mind, fatigue, and other factors, the weights will include possible error. However, none of the subjective approaches account for or consider this error (Schmitt and Levine, 1977). Minimizing this error term when estimating weights is very important.

The research to be presented will contribute to resolving these problem issues of the decision maker's inconsistency and judgmental error and thus lead to an improved model(s) for estimating weights. Now that the general problem area and issues have been discussed, Chapter II will summarize in additional detail the pertinent literature related to the topic.

Summary of Research Goal and Objectives

Based on the above discussion, the research goal is stated as follows:

Research Goal

To develop new weighting methods for use in solving MCDM problems based on the minimization of a decision maker's judgmental error and the refinement of a decision maker's inconsistency.

This research goal will be reached by achieving the following objectives:

Objectives

1. By developing three new analytical models based on minimizing the sum of a decision maker's judgmental error using all a_{ij} of a pairwise comparison matrix for refinement of a decision maker's inconsistency, utilizing linear programming as an optimization tool.
2. By testing the analytic models developed in this research against others reported in the literature using a simulation model to generate a decision maker's judgment of pairwise comparisons which includes simulated judgmental error. The testing phase will include setting up the hypotheses, computing the test statistic, drawing conclusions.
3. By comparing and analyzing the quality of the weights produced by the three proposed models with three models reported in the literature that use variations of eigen-vector methods: Saaty's Eigen-

vector Method, Cogger and Yu's Eigenweight Vector Method, and Takeda, et al.'s Graded Eigenvector Method. The testing criteria is to be based on the Euclidean distance measure and city block distance measure.

4. By developing a comprehensive and flexible interactive computer program to ease the task of data input, model optimization, statistical test.

Contribution

This research develops new weighting procedures employing the minimization of judgmental error and the refinement of decision maker's inconsistency using pairwise comparisons and linear programming, and compares the new procedures to other existing methodologies. This research contributes to minimizing judgmental error unlike other subjective methods. This research also contributes to refining the decision maker's inconsistency, unlike other weighting procedures, by using all a_{ij} in pairwise comparisons when estimating weights. This refining procedure is very simple when compared to the methods of enumerating all possible index orders or eliciting additional sets of weights from a decision maker. This research also provides an additional benefit by making available to both decision makers and researchers an interactive computer mode that facilitates easy and accurate input.

CHAPTER II

LITERATURE REVIEW

Introduction

This chapter reviews developments in the literature relevant to the research objective which was presented in Chapter I. The extensive literature on weight determination methods using subjective approaches has been reviewed. The subjective approaches use decision maker's decomposed judgments on criteria, rather than using the levels of criteria. The decision maker's judgments are often unrepresentative of true importance. Furthermore, judgmental error is seldom considered systematically. Various subjective methods exist. This chapter is divided into seven sections according to these methodologies which are: (1) Ranking, (2) Rating, (3) Point allocation, (4) Unit weighting, (5) Successive paired comparison, (6) Indifference trade-off, and (7) Eigen-vector methods. These methods are extracted from the surveys of Eckenrode (1965), Huber (1974), Cook and Stewart (1975), Hobbs (1980), and Takeda, et al. (1987).

Ranking

Using the ranking methodology (Eckenrode, 1965), decision makers order the criteria from the most important to the least important. Weights from these methods are on an ordinal scale of measurement, as ratios of weights are arbitrarily fixed. With ordinal scales only the ordering of phenomena is significant. The differences in numbers or their ratios are not considered important. Jopling (1974) and Watson (1974) make applications of the ranking method in a power plant siting study.

Rating

The rating method asks decision makers to rate on, say, a scale of 0 to 10, according to the importance of each criterion. Theoretically valid weights are not assured because a decision maker's definition of importance may have little to do with the relative value of the criteria. Eckenrode (1965) emphasizes the attractiveness of the ease of use of this method. Groups often apply this method assisted by Delphi technique (Delbecq et al, 1975) (Voelker, 1977).

Point Allocation

In the point allocation method, the decision maker is asked to distribute a fixed number of points among the various criteria so as to reflect their relative importance. This straightforward method was suggested as a good method by Hoffman (1960) and Schoemaker and Waid (1982), even though this method lacks formal theory. Similar point allocation methods have been advocated by Moore and Baker (1969) in various scoring models for evaluating engineering and R&D projects.

Unit Weighting

The unit weighting method standardizes the criteria in order to cause them to exhibit equal mean and variance, and then adds them together into a composite score.

Einhorn and Hogarth (1975) declare that the unit weighting method is a viable methodology for predictive purposes. They illustrate several reasons to support their declaration. The reasons are that unit weights are not estimated from the data and therefore do not consume degrees of freedom, and unit weights are free from judgmental error so that unit weights cannot reverse the true relative weights of the criteria. In addition to Einhorn and Hogarth's work, there have been a number of empirical studies by Trattner (1963), Lehman (1971), Fischer (1972), and Beckwith and Lehman (1973) that have shown that the unit weighting method is a good procedure for predictive purposes. Schmidt (1971, 1972) and Claudy (1972) have used simulation techniques in their works with the results generally showing that the unit weighting scheme performs quite well compared to regression. But Schoemaker and Waid (1982) do not agree with these results. They declare that the unit weighting method is clearly inferior relative to other methods such as linear regression, eigen-vector method and point allocation method after finishing their experiment on college admission. The use of the unit weighting method is desirable when the problem has many criteria and it is really difficult for the decision maker to figure out the relative importance of each criterion. Schoemaker and Waid's college admission problem has just four criteria. On the other hand, other researchers' problems have more than twelve criteria. This is the main reason of drawing different conclusions.

Successive Paired Comparison

This method proposed by Churchman and Ackoff (1954) uses two stages to determine the importance or weight of the criteria. First, the decision maker ranks criteria in order of importance as in the ranking method. The decision maker tentatively assigns the value 1 to the most important criterion and values between 0 and 1 to the other criteria in order of importance. The second stage systematically checks to see if those weights are consistent with trade-offs that the decision maker is willing to make. This is done via a number of questions and a question and answer scheme that asks the decision maker to decide whether the criterion with value of 1 is more important than all other criteria combined. If so, the decision maker may need to consider an increase in the value of the most important criterion, VC(1), so that VC(1) is greater than the sum of all other values of criteria. If not, the decision maker needs to adjust the value of the most important criterion, VC(1), so that VC(1) is less than the sum of all other criteria values. The decision maker then decides whether the second most important criterion is more important than the sum of all lower-valued criteria. The decision maker continues this process until n-1 criteria have been so evaluated. Any inconsistencies between a choice and the values assigned by the decision maker must be resolved by changing a choice, the values, or both. This can be very difficult and time consuming when there are many criteria. This method assures that the weights are valid because the decision maker checks the weights against acceptable trade-offs. Stimson (1969) applies this methodology for solving a public health problem and Davidson (1974) for solving a regional planning problem.

Indifference Trade-off

The indifference trade-off method (Huber, 1974), assures theoretically valid weights by determining if the decision maker will or will not trade-off one criterion value for another. Enough questions as to acceptable trade-offs are asked in order to solve for a unique set of weights. Consistency checks are especially important here as a decision maker will probably be very inconsistent on the first try because the decision makers usually will not think systematically about the trade-offs they are willing to make. In answering these questions, decision makers are forced to focus on their values of the criteria which is a desirable characteristic of this method. This technique has been applied in several site selection studies by Keeney and Nair (1977) and Keeney (1979).

Eigen-vector Methods

The eigen-vector method developed by Saaty (1977) requires pairwise comparisons of criteria in terms of relative importance. He explicitly assumes that the decision maker is consistent in the comparisons.

$$C' = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ . & . & & . \\ . & . & & . \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad (2.1)$$

The decision maker constructs the nxn pairwise comparison matrix of C' as can be seen in (2.1). In such a matrix, a_{ij} is the relative strength or importance of criterion i compared to criterion j. The decision maker's enforcement of $a_{ji}=1/a_{ij}$ due to the assumption of consistency makes mathematical analysis easier (Saaty, 1980) (Belton, 1986). However this is not, in general, congruent with human perception (Cogger and Yu, 1985). Even though Saaty's eigen-vector method has a rigid consistency assumption, hundreds of applications have been made to MCDM problems because its weights are reasonably good and easy to use (Schoemaker and Waid, 1982). Saaty's weights are determined by normalizing the eigen-vector associated with the maximum eigenvalue of the ratio matrix.

Cogger and Yu (1985) developed the New Eigenweight Vector Method. This method is based on Saaty's original eigen-vector method. These individuals recognized that stable and internally consistent estimates of weights may be difficult to obtain since humans have perceptions and judgments which are subject to change due to their psychological states and various information inputs. Based on this argument, they assume that the decision maker is not necessarily consistent in the comparisons. To reflect the inconsistency of comparisons they derive weights from all the index orders of the criteria. From the matrix of (2.1), the relation $a_{ij}=1/a_{ji}$ may not hold in this case. The weights are estimated in recursive fashion by

$$W_{n-1} = a_{n-1,n} W_n \tag{2.2}$$

$$W_{n-2} = (a_{n-2,n-1}W_{n-1} + a_{n-2,n}W_n) / 2 \tag{2.3}$$

$$\vdots$$

$$W_1 = (a_{12}W_2 + a_{13}W_3 + \dots + a_{1n}W_n) / (n-1) \tag{2.4}$$

From (2.2) through (2.4), W_k is obtained from the average of ($a_{k,k+1} W_{k+1}$, $a_{k,k+2} W_{k+2}$, ... , $a_{k,n} W_n$). Once W_n is estimated, W_{n-1} can be estimated in (2.2) with one step, then W_{n-2} can be estimated in (2.3) with two steps, etc. Thus, in estimating n element weight vector W, the ratio estimate $a_{n-1,n}$ is most important, $a_{n-2,n-1}$ and $a_{n-2,n}$ are second most important, etc. This indicates that the index order of the criteria can affect the estimate of W. Thus Cogger and Yu emphasize the need to enumerate all index orders of the criteria. Cogger and Yu's weights are the geometric mean of the weights from all possible index order combinations of the criteria.

Saaty's eigen-vector method explicitly requires consistency in the pairwise comparisons. This assumption makes mathematical analysis easier, but is not always congruent with human perception as mentioned earlier. Cogger and Yu (1985) refine this consistency assumption by allowing decision maker's inconsistency and obtaining weights for all the possible index orders. They also emphasize that this makes computation less difficult when compared to Saaty's method. However, enumerating all possible index orders is not an easy task. Cogger and Yu's method produces three different index orders for a problem having three criteria, twelve for a problem having four criteria, and n!/2 for a problem having n criteria. The number of different index orders increases dramatically as the number of criteria increases. One more very important flaw of the Cogger and Yu method to be pointed out is that their weight is the geometric mean of the weights from all possible index orders. An index order of 360 must be enumerated when the problem has six criteria. A severe underflow problem is encountered when multiplying the numbers which are less than 1.0 360 times. The mathematics prohibits the calculation of the geometric mean when the problem has more than five criteria.

Takeda, et al. (1987) developed the Graded Eigenvector Method which generalizes the methods of Saaty (1977), and Cogger and Yu (1985). It differs from that of Saaty by allowing the solution to reflect the decision maker's inconsistencies revealed by the estimates in the reciprocal portion of the matrix. It also differs from the Cogger and Yu procedure by choosing a specific index order rather than enumerating all possible index orders. The Graded Eigenvector Method is another version that attempts to refine Saaty's consistency assumption by allowing decision maker's inconsistency. To accomplish this refinement, the following form for a C' matrix is used instead of (2.1).

$$C' = \begin{bmatrix} a_{11} & \beta_{12}a_{12} & \beta_{13}a_{13} & \cdots & \beta_{1n}a_{1n} \\ & a_{22} & \beta_{23}a_{23} & \cdots & \beta_{2n}a_{2n} \\ & & \cdot & & \cdot \\ & & & \cdot & a_{n-1n} \\ & & & & a_{nn} \end{bmatrix} \tag{2.5}$$

where $\beta_{ij}>0$ and $\sum_{j=i+1}^{n} \beta_{ij}=1$ for each $i=1,2,\ldots,n-2$. After modifying equations (2.2) through (2.4), the weights can be estimated in recursive fashion by

$$
\begin{aligned}
W_{n-1} &= a_{n-1,n}\, W_n \\
W_{n-2} &= (\beta_{n-2,n-1}a_{n-2,n-1}W_{n-1} + \beta_{n-2,n}a_{n-2,n}W_n) \\
&\;\;\vdots \\
W_1 &= (\beta_{12}a_{12}W_2 + \beta_{13}a_{13}W_3 + \ldots + \beta_{1n}a_{1n}W_n)
\end{aligned}
$$

However, the tasks of providing a set of weights, β_{ij}, which is the normalized values of D(i,j) for i=1,2,...,n-2, and j=i+1,...,n, in addition to providing the values of pairwise comparisons, a_{ij}, are not easy from the decision maker's view point. D(i,j) represents the decision maker's confidence, or degree of knowledge when comparing criterion i with criterion j.

Cogger and Yu (1985) and Takeda, et al. (1987) have tried to refine the Saaty's consistency assumption by allowing decision maker's inconsistency in pairwise comparisons. Cogger and Yu resolve this problem by getting the geometric mean of weights from all possible index orders. In the case of Takeda, et al., they elicit an additional sets of weights, β_{ij}, from the decision maker to avoid enumerating all possible index orders. They refine and generalize some aspects of the problem, but add elements of complexity to their approaches.

Conclusion

This chapter presents a survey of the literature relative to the research objective detailed in Chapter I. As summarized in Table 2.1, this survey has concentrated on several features of the weighting methods such as theoretical validation, simplicity, allowance for decision maker's inconsistency and minimization of judgmental error. Comparing the methods to each other using several important features illustrated in Table 2.1, the first four methods share one good feature which is simplicity of use. The successive paired comparison method and the indifference trade-off method have a theoretical background but none of the other features. Saaty's eigen-vector method has two good features which are theoretical validation and simplicity of use. The methods of Cogger and Yu and Takeda, et al. have theoretical validation, simplicity of use, and allowance for decision maker's inconsistency. From this summary, eigen-vector methods have relatively better features compared to other methods. The development of the new weighting methods which have more than three good features can be considered at this point. Particularly, the feature of the minimization of the decision maker's judgmental error is a new concept for estimating weights using subjective approaches. Also, it is desirable for methods to be developed for reflecting decision maker's inconsistency more systematically than the

Cogger and Yu's method and the Takeda, et al.'s method. The research goal and objectives to be pursued was contributed to reflect the need of these new concepts.

A summary of weighting methods shown in this Chapter and a chronological summary for each method are provided in Table 2.2 and Table 2.3 respectively.

TABLE 2.1

SUMMARY OF FEATURES OF VARIOUS WEIGHTING METHODS

Method	TV[1]	SOU[2]	AOI[3]	MJE[4]
Ranking	No	Yes	No	No
Rating	No	Yes	No	No
Point Allocation	No	Yes	No	No
Unit Weighting	No	Yes	No	No
Successive paired Comparison	Yes	No	No	No
Indifference Trade-off	Yes	No	No	No
Eigen-vector				
Saaty	Yes	Yes	No	No
Cogger and Yu	Yes	Yes	Yes	No
Takeda et al.	Yes	Yes	Yes	No

1 Theoretical Validation
2 Simplicity of use
3 Allowance of Inconsistency
4 Minimization of Judgmental Error

TABLE 2.2

SUMMARY OF VARIOUS WEIGHTING METHODS

Methods	Authors
Ranking	Eckenrode (1965) Jopling (1974) Watson (1974)
Rating	Eckenrode (1965) Delbecq et al. (1975) Voelker (1977)
Point Allocation	Hoffman (1960) Moore and Baker (1969) Schoemaker and Waid (1982)
Unit Weighting	Trattner (1963) Lehman (1971) Schmidt (1971, 1972) Claudy (1972) Fischer (1972) Beckwith and Lehmann (1973) Einhorn and Hogarth (1975) Schoemaker and Waid (1982)
Successive Paired Comparison	Churchman and Ackoff (1954) Stimson (1969) Davidson (1974)
Indifference Trade-off	Huber (1974) Keeney and Nair (1977) Keeney (1979)
Eigen-vector	Saaty (1977, 1980)) Schoemaker and Waid (1982) Cogger and Yu (1985) Belton (1986) Takeda et al. (1987)

TABLE 2.3

CHRONOLOGICAL SUMMARY OF WEIGHTING METHODS

Method \ Year	54	60	63	65	69	71	72	73	74	75	77	79	82	85	87
Ranking				x					xx						
Rating				x						x	x				
Point Allocation		x			x								x		
Unit Weighting			x			xx	xxx	x		x	x				
Successive Paired Comparison	x				x				x						
Indifference Trade-off									x		x	x			
Eigen-vector											x		x	x	x

CHAPTER III

MODEL DEVELOPMENT

Introduction

From the literature review in Chapter II, seven different weighting methods have been reviewed. None of the methods meet all the desirable characteristics such as theoretical validation, refinement of decision maker's inconsistency, minimization of judgmental error, quality, and simplicity. In this chapter, three new weighting methods which appear to meet the desirable characteristics will be developed. Several assumptions and notations have been made for developing the weight determination models.

Assumptions

The basic assumptions which are utilized in developing the models are as follows:

1) The pairwise comparisons, with possible error between two criteria, are made by a single decision maker or by multiple decision makers on the basis of some global objective.

2) The methodology imposes no requirement that the paired comparisons satisfy the reciprocal property.

3) Measurements on each of the n criteria are ratio scaled.

4) Inconsistency in human judgment is uniformly distributed on the interval (.5, 1.5) for a simulation run used to analyze the results from a single decision maker. Inconsistency for a second decision maker's judgment is uniformly distributed on the interval (.3, 1.7) for analyzing the results from the two decision maker problem.

Notation

To facilitate the development of the mathematical models to be presented, the following notation is introduced and will be used throughout the research.

i = 1,2,...,n where n is the number of criteria.

r = 1,2,...,R where R is the number of replications for a simulation run.

V_k = a composite value of the suitability of alternative k.

X_{ik} = the level of criterion i for alternative k.

W_{ij} = the ratio of W_i and W_j which is W_i/W_j.

W_i = true weight of criterion i.

$W_i^{(r)}$ = an estimated weight of criterion i at the r^{th} replication.

W = true weight vector.

W' = estimated weight vector.

a_{ij} = decision maker's estimated value of W_{ij}.

a_{ijq} = a_{ij} values estimated by decision maker q.

ε_{ij} = possible judgmental error when W_{ij} is estimated. This is a uniform random variable on the interval (.5, 1.5) with mean of one.

e_i = aggregated judgmental error for criterion i.

C = matrix constructed from true weights.

C' = matrix consisting of pairwise comparisons of criteria obtained from a decision maker.

C'_q = a C' matrix constructed from decision maker q.

C'_{avg} = matrix of the averages of the C'_q.

C_i = represents the criterion i.

$C_i > C_j$ = represents that criterion i is more important than criterion j.

D(i,j) = represents decision maker's confidence, or degree of knowledge when comparing criterion i with criterion j.

β_{ij} = normalized values of D(i,j) for i=1,2,...,n-1 and j=i+1,...,n.

Ω_1 = denotes a set of relations (i,j) for all i,j=1,2,...,n except i=j such that criterion i is more important than criterion j in a pairwise comparison.

Ω_2 = denotes a set of relations (m,n) for all m,n=1,2,...,n except m=n such that criterion m is "how much" more (a_{mn}>1), or less (a_{mn}<1), or equally (a_{mn}=1) important than criterion n in a pairwise comparison.

γ_{ij} = represents the aggregated judgmental error for all (i,j) in Ω_1.

δ_{mn} = represents the aggregated judgmental error for all (m,n) in Ω_2.

I_{ij} = integer variables taking either 0 or 1.

J_{mn} = integer variables taking either 0 or 1. M = a large number greater than max(a_{ij}) for all i,j=1,2,...,n.

NC = number of criteria.

α = probability of Type I error.

β = probability of Type II error.

H_0 = null hypothesis.

H_1 = alternative hypothesis.

μ_k = population mean of the differences between the true weight vector and estimated weight vector from model k.

d_f = difference between the true weight vector and the estimated weight vector to be detected where f identifies the measure of goodness of fit used such as 1 for a Euclidean distance measure and 2 for a city block distance measure.

d_{fk} = d_f value calculated from model k.

d'_f = average of d_{fk}.

R_p = least significant ranges.

p = number of between models.

q_α = significant studentized ranges for Duncan's new multiple-range test.

fe = error degree of freedom.

S_d = standard error of a between models' mean.

The New Models for Estimating Weights Using Pairwise Comparison Matrix from a Single Decision Maker

The weighting methods to be developed are based on pairwise comparisons constructed from a single decision maker, and optimized via linear programming for the purpose of minimizing the judgmental error. Pairwise comparisons used in these models were developed by Hay (1958) and revised by Buel (1960). Pairwise comparison is the process of comparing one criterion against another, with never more than two criteria involved in each comparison. This simplifica-tion of comparisons usually promotes greater accuracy.

The models developed in this research are of a linear form which allows linear programming to be utilized as an optimization tool. In addition, linear programming has the capability of producing solutions in a reasonable amount of time with readily available software.

Model 1 Development

For constructing a pairwise comparison matrix, denote the criteria by $C_1, C_2, \ldots, C_n$ and their true weights by $W_1, W_2, \ldots, W_n$. In this ideal case, the relations between the weights W_i and the judgments a_{ij} are simply given by

$$\frac{W_i}{W_j} = a_{ij} \tag{3.1}$$

for all i,j=1,2,...,n. The results of pairwise comparisons may be represented by a matrix C as follows:

$$C = \begin{array}{c} \\ C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{bmatrix} W_1/W_1 & W_1/W_2 & \ldots & W_1/W_n \\ W_2/W_1 & W_2/W_2 & \ldots & W_2/W_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ W_n/W_1 & W_n/W_2 & \ldots & W_n/W_n \end{bmatrix} \end{array} \tag{3.2}$$

This matrix has positive entries everywhere, 1's on the main diagonal, and satisfies the reciprocal property. This matrix C satisfies the cardinal consistency property $a_{ij}*a_{jk} = a_{ik}$ and is called consistent. This property says that if any row of C is given, the rest of the entries can be determined from this relation. However, it would be unrealistic to require these relations to hold in the general case.

Now suppose that the scale is not known, and that the entries in the matrix are estimates of the ratios. In this case the cardinal consistency relation above may not hold, and an ordinal relation of the form $W_i>W_j$, $W_j>W_k$ implying $W_i>W_k$ may not hold. As a realistic representation of the situation in pairwise comparisons, it is likely that inconsistency in judgments may occur. Despite their best efforts, people's feelings and preferences are often inconsistent and intransitive (Takeda, et al., 1987).

The only parameters in this model are the W_i. These parameters are estimated from a decision maker's judgments, a_{ij}, which are equal to W_i/W_j when the true weights are known. When the judgments, a_{ij}, are obtained from a decision maker, they may not be equal to W_i/W_j because W_i is never known. To construct a pairwise comparisons matrix, a decision maker is asked to decide how much criterion i is more important than criterion j for all i,j=1,2,...,n except i=j. These questions are needed for assurance of theoretical validation. After making n(n-1) comparisons, the results may be represented by a matrix as shown by (3.3). This matrix has positive entries everywhere, 1's on the main diagonal but does not necessarily satisfy the reciprocal property. That is, a_{ij} is not necessarily equal to $1/a_{ji}$. In addition, the matrix C' does not necessarily satisfy the cardinal consistency property either.

$$C' = \begin{array}{c} \\ C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \end{array} \qquad (3.3)$$

As mentioned earlier, the relation in (3.1) holds when C is equal to C'. Using this relation, the W_i can be written as follows:

$$\begin{array}{rl} W_i &= a_{i1}W_1 \\ W_i &= a_{i2}W_2 \\ & \vdots \\ + \; W_i &= a_{in}W_n \\ \hline nW_i &= \sum_{j=1}^{n} a_{ij}W_j \end{array} \tag{3.4}$$

But in the general case, the relation (3.4) may not hold because a decision maker's judgmental error is included in the a_{ij}. This occurs due to factors such as lack of knowledge, personal experience, interest, fatigue, state of mind, etc.

Consequently, instead of the ideal case relations of (3.4), the more realistic realizations for the general case can be considered to take the form

$$nW_i \begin{array}{c} > \\ \approx \\ < \end{array} \sum_{j=1}^{n} a_{ij}W_j \tag{3.5}$$

for i=1,2,...,n. To make the relation (3.5) an equality, an unrestricted variable, e_i, is added to (3.5) as follows:

$$nW_i = \sum_{j=1}^{n} a_{ij}W_j + e_i \tag{3.6}$$

More explicitly,

$$
\begin{array}{llllll}
(a_{11}-n)W_1+ & a_{12}W_2+\ldots+ & a_{1n}W_n+e_1 & & =0 & \\
a_{21}W_1+ & (a_{22}-n)W_2+\ldots+ & a_{2n}W_n & +e_2 & =0 & \\
. & . & . & & . & (3.7) \\
. & . & . & & . & \\
a_{n1}W_1+ & a_{n2}W_2+\ldots+ & (a_{nn}-n)W_n & +e_n & =0 &
\end{array}
$$

As given, these simultaneous linear equations have the trivial solution of $W_i=0$ and $e_i=0$ for all i. For this trivial solution all the V_k, where k identifies the alternative, turn out to be zero (see equation (1.1)). The trivial solution does not convey any useful information so that the model should preclude its selection. To prevent triviality, an equation of the form $\Sigma W_i=h$ for all i where h is any positive number, preferably 1 for standardizing the unit of measurement, can be added to (3.7) without any loss of generality. Now the system can be written as follows:

$$
\begin{array}{llllll}
(a_{11}-n)W_1+ & a_{12}W_2+\ldots+ & a_{1n}W_n+e_1 & & =0 & \\
a_{21}W_1+ & (a_{22}-n)W_2+\ldots+ & a_{2n}W_n & +e_2 & =0 & \\
. & . & . & & . & (3.8) \\
. & . & . & & . & \\
a_{n1}W_1+ & a_{n2}W_2+\ldots+ & (a_{nn}-n)W_n & +e_n & =0 & \\
W_1+ & W_2+\ldots+ & W_n & & =1 &
\end{array}
$$

With the addition of the normalization constraint, the system (3.8) now assures the existence of the solution, and the weights can be calculated from (3.8) by minimizing the sum of judgmental error as shown below.

Mathematical Statement of Model 1

$$\text{Minimize } \sum_{i=1}^{n} e_i$$

Subject to

$$\sum_{j=1}^{n} a_{ij}W_j - nW_i + e_i = 0 \text{ for } i=1,2,\ldots,n$$

$$\sum_{j=1}^{n} W_j = 1$$

$W_j \geq 0$ for all j

e_i is unrestricted.

This mathematical model can be solved via linear programming.

Model 2 Development

The second model derived from relations (3.3) and (3.6) is to be considered. Additional information can be extracted from the C' matrix (3.3). The first type of information is "which criterion is more important than which criterion". At most n(n-1) relations of $C_i \geq C_j$ are available. One understands that $a_{ij} \geq 1$ directly implies that $C_i \geq C_j$. Let Ω_1 denote a set of relations (i,j) such that criterion i is more important than criterion j in a pairwise comparison. $C_i \geq C_j$ implies that $W_i \geq W_j$ because the decision maker determines that criterion i is more important than or equally important to criterion j. This relation, however, may not hold for some of the pairs because of the possibility of the various sources of error. γ_{ij} is introduced to identify and aggregate the various sources of error. Using (3.6),

$$nW_i \geq nW_j$$

$$\Longrightarrow \sum_{k=1}^{n} a_{ik}W_k + e_i \geq \sum_{k=1}^{n} a_{jk}W_k + e_j$$

$$\Longrightarrow \sum_{k=1}^{n} (a_{ik}-a_{jk})W_k + \gamma_{ij} \geq 0 \qquad (3.9)$$

for all (i,j) in Ω_1.

The second type of information extracted from (3.3) is which criterion is "how much" more, or less, or equally important than which criterion. This "how much" term is denoted by a_{mn} in a pairwise comparison. At most n(n-1) terms of $W_m \geq a_{mn}W_n$ or $W_m \leq a_{mn}W_n$ are available. If $a_{mn} = 1/a_{nm}$, then either a_{mn} or a_{nm} can be used. Let Ω_2 denote a set of relations (m,n) in a pairwise comparison. δ_{mn} is introduced to identify and aggregate the various sources of error. Then

$$W_m - a_{mn}W_n + \delta_{mn} \geq 0 \text{ when } a_{mn} \geq 1 \qquad (3.10)$$

and

$$a_{mn}W_n - W_m + \delta_{mn} \geq 0 \text{ when } a_{mn} \leq 1 \qquad (3.11)$$

for all (m,n) in Ω_2. Using (3.9), (3.10), and (3.11), the second model is completed as follows:

Mathematical Statement of Model 2

$$\text{Minimize} \sum_{\substack{\text{all } (i,j) \text{ in } \Omega_1 \\ \text{all } (m,n) \text{ in } \Omega_2}} (\gamma_{ij}+\delta_{mn})$$

Subject to

$$\sum_{k=1}^{n} (a_{ik}-a_{jk})W_k + \gamma_{ij} \geq 0 \text{ for all } (i,j) \text{ in } \Omega_1$$

$$W_m - a_{mn}W_n + \delta_{mn} \geq 0 \text{ if } a_{mn}\geq 1$$

$$a_{mn}W_n - W_m + \delta_{mn} \geq 0 \text{ if } a_{mn}\leq 1$$

$$\sum_{k=1}^{n} W_k = 1$$

where $W_k \geq 0$ for all k, and γ_{ij} and δ_{mn} are unrestricted for all (i,j) in Ω_1 and (m,n) in Ω_2. This mathematical model can be optimized via linear programming.

Model 3 Development

The third model to be considered is model 2 with an alternative objective function. Instead of minimizing the amount of possible error, minimizing the number of violations of equations for all (i,j) in Ω_1 and all (m,n) in Ω_2 is considered. This consideration is based on the reasoning that even though the sum of γ_{ij} and δ_{mn} might be minimized, the number of violations of equations for all (i,j) in Ω_1 and all (m,n) in Ω_2 might increase. This model can be formulated as follows:

Mathematical Statement of Model 3

$$\text{Minimize} \sum_{\substack{\text{all } (i,j) \text{ in } \Omega_1 \\ \text{all } (m,n) \text{ in } \Omega_2}} (I_{ij}+J_{mn})$$

Subject to

$$\sum_{k=1}^{n} (a_{ik}-a_{jk})W_k + MI_{ij} \geq 0 \text{ for all } (i,j) \text{ in } \Omega_1$$

$$W_m - a_{mn}W_n + MJ_{mn} \geq 0 \text{ if } a_{mn} \geq 1$$

$$a_{mn}W_n - W_m + MJ_{mn} \geq 0 \text{ if } a_{mn} \leq 1$$

$$\sum_{k=1}^{n} W_k = 1$$

where $W_k \geq 0$ for all k, M is a large number greater than $\max(a_{ij})$ for all $i,j=1,2,\ldots,n$, I and J are 0 or 1 integer variables. The above model can be solved by a mixed integer programming code.

Procedures for Estimating Weights Using Pairwise Comparison Matrices from Multiple Decision Makers

There are a number of circumstances in which it is desirable to reflect the judgment of several decision makers on a single analysis. It is a reasonable assumption that multiple decision makers work to accomplish some common objective even though they have different backgrounds.

The procedures for estimating weights from multiple decision makers consider the opinions of decision makers by utilizing pairwise comparison matrices constructed by the decision makers. The procedures are appropriate in situations where the decision makers cannot be presumed nearly identical in their pairwise comparison judgment. They are also appropriate when the purpose of analysis is the prediction of a composite which, in some sense, represents the aggregate behavior of the decision makers. Two procedures for estimating weights from multiple decision makers are suggested below. The results of the simulation run will be reported in Chapter V.

Estimating Weights after Averaging Pairwise Comparison Matrices

Each decision maker constructs a pairwise comparison matrix. The procedure of constructing a pairwise comparison matrix is exactly the same as explained in the previous section. The only difference is that the number of pairwise comparison matrices equals the number of decision makers. From each decision maker, pairwise comparison matrix (C'_q) is constructed by the decision maker q as shown in (3.12) where

$$C'_q = \begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{array}{c} C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_n \end{array} & \begin{bmatrix} a_{11q} & a_{12q} & \cdots & a_{1nq} \\ a_{21q} & a_{22q} & \cdots & a_{2nq} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1q} & a_{n2q} & \cdots & a_{nnq} \end{bmatrix} \end{array} \qquad (3.12)$$

q=1,2,...,N stands for the index of the decision maker.

After constructing N C'_q matrices, the averages of the C'_q which calculated by the formula $C'_{avg} = \frac{1}{N} \sum_{q=1}^{N} C'_q$ are obtained as shown in (3.13). The weights can be estimated using (3.13) as an input data to any models developed in previous section.

$$C'_{avg} = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{bmatrix} \sum_{q=1}^{N} a_{11q}/N & \sum_{q=1}^{N} a_{12q}/N & \cdots & \sum_{q=1}^{N} a_{1nq}/N \\ \sum_{q=1}^{N} a_{21q}/N & \sum_{q=1}^{N} a_{22q}/N & \cdots & \sum_{q=1}^{N} a_{2nq}/N \\ \vdots & \vdots & & \vdots \\ \sum_{q=1}^{N} a_{n1q}/N & \sum_{q=1}^{N} a_{n2q}/N & \cdots & \sum_{q=1}^{N} a_{nnq}/N \end{bmatrix} \end{array} \qquad (3.13)$$

Averaging Individual Weights of Decision Makers

The C'_q matrix shown in (3.12) is constructed by the decision maker q. The weights can be estimated using C'_q pairwise comparison matrix. N weight vectors, one for each decision maker, can be calculated. The weights for a given problem are then estimated by averaging the N individual weights.

CHAPTER IV

USING THE INTERACTIVE COMPUTER PROGRAM

Introduction

This chapter illustrates the use of an interactive computer program which permits easy utilization of the weighting methods presented in the previous chapter. The actual FORTRAN program is documented and appears in Appendix A. It has been implemented on an IBM 3081D.

The entire program is interactive, and the user is prompted for all necessary inputs by the computer. Many typical and/or often-used values of inputs are preprogrammed, but can be easily modified when necessary. Only when a set of inputs has been checked by the program and verified by the user does the program continue.

Integer values are usually entered without a decimal point; however, a decimal may be included. With the prompting and verification feature, the input mechanism is virtually self-explanatory. It does require that the user understand the terms being input and their mathematically feasible range.

In the remainder of this chapter, actual interactive output is interspersed with comments and explanations. All computer outputs shown are automatically generated by the computer except for the input values which follow a question mark (?). These question marks remind the user to enter the input values.

Interactive Program Development

An interactive routine is designed such that the decision maker and/or the researcher can iteratively provide information for constructing a pairwise comparison matrix which is used to achieve satisfactory weights. Figure 4.1 illustrates the components of the interactive computer model. The inputs to the computer model and the output expected from the computer model are given as follows:

INPUTS : 1. Number of decision makers,

2. Number of criteria, and

3. a_{ij} values of pairwise comparisons.

OUTPUT : Weights.

Since existing codes are not designed for interactive mode and simulation purposes, available linear programming and mixed integer programming codes (Kuester and Mize, 1973) are modified to meet the special purposes.

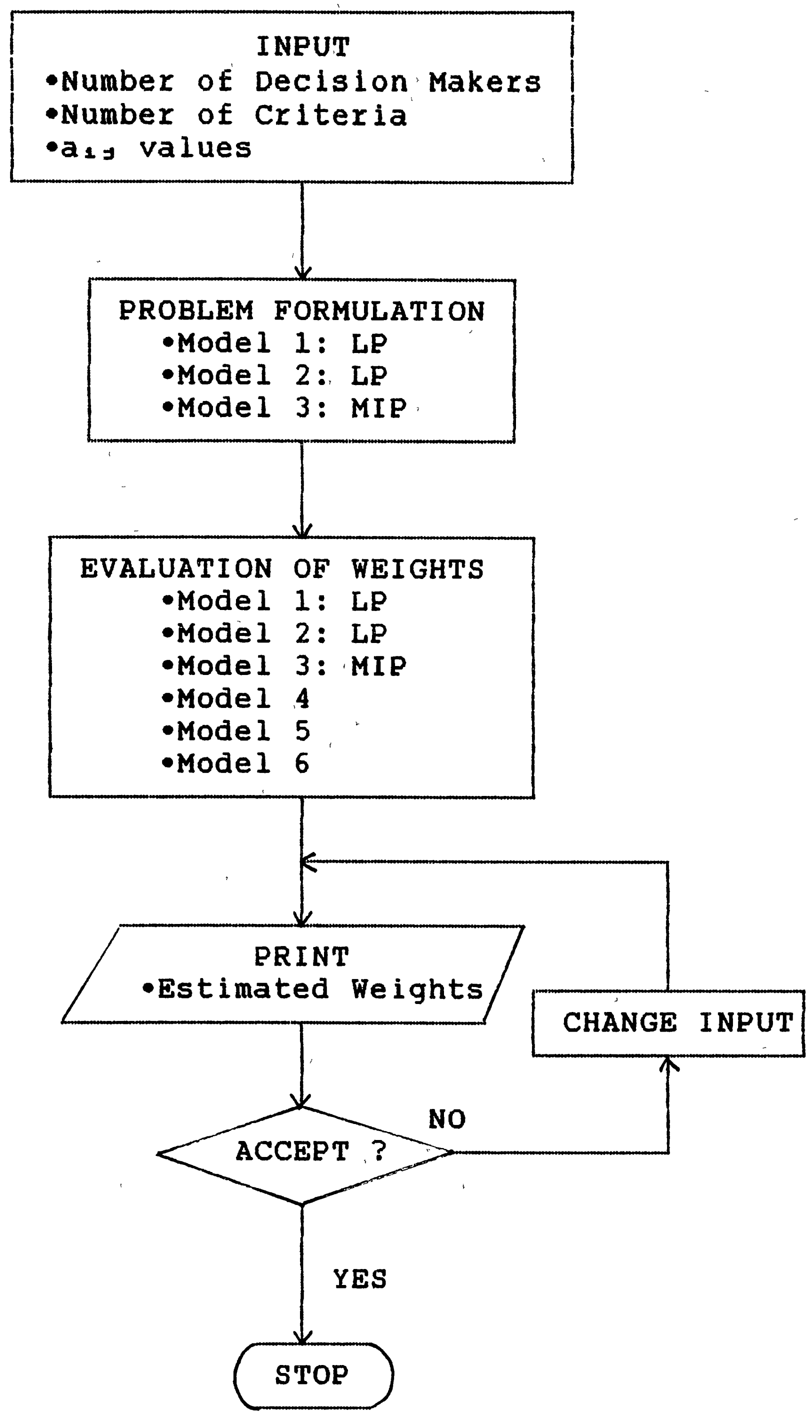

Figure 4.1. Flowchart for Interactive Model

Overview

The interactive computer program provides the capability of interactively entering pairwise comparisons data from a decision maker(s) for use in any of the models of this research. It also provides the capability of choosing any model of the three methods developed in Chapter III in addition to the three eigen-vector methods. The program begins by presenting the main option menu (M.1). The user has entered a "1", indicating a desire to enter the input data of pairwise comparisons matrix for estimating weight.

```
        ************************
        ***     MAIN   MENU    ***
        ************************

    1. INPUT PAIRWISE COMPARISONS,                    (M.1)
    2. EXIT THE PROGRAM.

    ==> ENTER THE OPTION NUMBER!
    ?
    1
```

Input Pairwise Comparisons

After option 1 (Input Pairwise Comparisons) is selected, the user is asked to enter the number of decision makers. Then the program prints the number of decision makers entered for verification by the user shown as follows.

```
==> ENTER THE NUMBER OF DECISION MAKERS!
?
1


YOU HAVE 1 DECISION MAKER(S).  IS THIS NUMBER CORRECT?

==> ENTER 1=YES, 2=NO. <<<
? 1
```

Only after the user confirms the validity of the input does the program continue. After this verification, the program prompts the user to enter the number of criteria. After the number of criteria is entered, the program prints the input data for verification by the user as follows.

```
==> ENTER THE NUMBER OF CRITERIA!
?
3

YOU HAVE 3 CRITERIA.  IS THIS NUMBER CORRECT?

==> ENTER 1=YES, 2=NO. <<<
?
1
```

After the number of decision makers and the number of criteria have been entered and confirmed, a value of relative importance between criterion i and criterion j is requested iteratively and is illustrated as follows.

```
*** THIS INPUT IS FOR DECISION MAKER 1! ***

==> BY HOW MUCH IS CRITERION 1 MORE IMPORTANT THAN
    CRITERION 2 ?
?
1.03
```

```
==> BY HOW MUCH IS CRITERION 1 MORE IMPORTANT THAN
    CRITERION 3 ?
?
3.67

==> BY HOW MUCH IS CRITERION 2 MORE IMPORTANT THAN
    CRITERION 1 ?
?
0.55

==> BY HOW MUCH IS CRITERION 2 MORE IMPORTANT THAN
    CRITERION 3 ?
?
2

==> BY HOW MUCH IS CRITERION 3 MORE IMPORTANT THAN
    CRITERION 1 ?
?
0.27

==> BY HOW MUCH IS CRITERION 3 MORE IMPORTANT THAN
    CRITERION 2 ?
?
0.5
```

Communication with the decision maker(s) is needed to provide input for this kind of pairwise comparisons. Upon completion of entering pairwise comparisons data, the program prints these input data for verification by the user shown below.

```
*********************************************
*** VALUES RECEIVED FROM DECISION MAKER 1 ***
*********************************************

1      1.03  3.67
0.55   1     2
0.27   0.5   1

*** ARE THESE DATA CORRECT ? ***

==> ENTER 1=YES, 2=NO. <<<
?
2
```

If the user desires to correct any input data, then a selection of "2" is entered and the program prompts the user for entering a row index number, a column index number, and a corrected value of relative importance. The prompts and responses to correct input data are illustrated in (M.2).

```
*** THIS INPUT IS FOR DECISION MAKER 1! ***

==> ENTER ROW INDEX NUMBER!
?
1

==> ENTER COLUMN INDEX NUMBER!                          (M.2)
?
2

==> ENTER CORRECTED VALUE OF RELATIVE IMPORTANCE!
?
1.83


*** DO YOU NEED TO CHANGE MORE? ***

==> ENTER 1=YES, 2=NO. <<<
?
2
```

The program then prompts "DO YOU NEED TO CHANGE MORE?". If the user needs to change more, a selection of "1" is entered and the procedure of (M.2) is repeated. If a selection of "2" is made, then the new pairwise comparisons matrix is displayed for user confirmation as shown below.

```
*********************************************
*** VALUES RECEIVED FROM DECISION MAKER 1 ***
*********************************************

1       1.83   3.67
0.55    1      2
0.27    0.5    1


*** ARE THESE DATA CORRECT ? ***

==> ENTER 1=YES, 2=NO. <<<
?
1
```

Upon completion of the entering of input data for the pairwise comparisons matrix, the program prompts for the model option. If the user desires to use model 1 to estimate weights of a given problem, then the user responds with a selection of "1".

```
**************************
*** MODEL AVAILABILITY ***
**************************

1. MODEL 1
2. MODEL 2
3. MODEL 3
4. MODEL 4
5. MODEL 5
6. MODEL 6

==> ENTER THE MODEL NUMBER!
?  1
```

The estimation of the weights for given pairwise comparisons matrix is performed after making the selection of model. Upon completion, the program prints the estimated weights as shown below.

```
*************************
*** ESTIMATED WEIGHTS ***
*************************

W(1) = 0.55
W(2) = 0.3
W(3) = 0.15


*** DO YOU WANT TO GO BACK TO THE MAIN MENU? ***

==> ENTER 1=YES, 2=NO. <<<
?
1
```

If the user wants to solve another problem, a selection of "1" is needed for the main menu. If the user needs to exit the program, a selection of "2" is needed. The user can repeat the procedure until he/she has no need of it.

Summary

The features of the interactive computer program of this research have been illustrated in this chapter. An example is given for describing the capability of the program. The interactive feature and its convenience make this computer program a useful tool for communicating with decision makers and for estimating the weights to a given problem.

CHAPTER V

RESULTS, COMPARISON, AND ANALYSIS

Introduction

This chapter reports the results of the testing of the models developed in this research. It includes comparing the results of the three models developed in this research with the three eigen-vector methods reviewed earlier; Saaty's eigen-vector method, Cogger and Yu's eigenweight vector method, and Takeda, et al.'s graded eigenvector method.

Simulation was used to compare the three models developed in this research with the three eigen-vector methods. These three eiegn-vector methods are utilized for comparisons because the weights of these three eigen-vector methods are estimated from a pairwise comparison matrix as is done for the three models developed in this research.

Takeda, et al. (1987) also used simulation in their comparative study of their method with Saaty's method and Cogger and Yu's method using eight decision making settings involving up to five criteria shown in Table 5.1. The resulting choices in the order of generating better solutions were Takeda, et al.'s method, Cogger and Yu's method, and Saaty's method.

TABLE 5.1

SUMMARY OF EIGHT DECISION MAKING SETTING PROBLEMS USED BY TAKEDA, ET AL.

Problem	W	D(i,j)*
1	(0.15,0.55,0.3)	D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3
2	(0.3,0.15,0.55)	D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3
3	(0.55,0.3,0.15)	D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3
4	(0.55,0.3,0.15)	D(1,2)=0.9, D(1,3)=0.8, D(2,3)=0.6
5	(0.2,0.4,0.1,0.3)	D(1,2)=0.7, D(1,3)=0.9, D(1,4)=0.8, D(2,3)=0.7, D(2,4)=0.6, D(3,4)=0.4
6	(0.2,0.4,0.1,0.3)	D(1,2)=0.8, D(1,3)=0.7, D(1,4)=0.9, D(2,3)=0.4, D(2,4)=0.6, D(3,4)=0.5
7	(0.2,0.4,0.1,0.3)	D(1,2)=0.7, D(1,3)=0.6, D(1,4)=0.8, D(2,3)=0.5, D(2,4)=0.6, D(3,4)=0.3
8	(0.25,0.3,0.15,0.1,0.2)	D(1,2)=0.6, D(1,3)=0.7, D(1,4)=0.8, D(1,5)=0.6, D(2,3)=0.7, D(2,4)=0.6, D(2,5)=0.6, D(3,4)=0.5, D(3,5)=0.8, D(4,5)=0.3

*decision maker's confidence when comparing criterion i with criterion j.

A critical choice in Takeda, et al.'s simulation study was the modeling of inconsistency of human judgment which was treated as random variation. The statistical model that they selected for simulating of human judgment was

$$a_{ij} = w_{ij}\varepsilon_{ij} \tag{5.1}$$

where W_{ij} was assumed to have a true value and ε_{ij} was assumed to be a uniformly distributed random variable on the interval (.5, 1.5) with a mean of one. The pairwise comparison matrix, C', for estimating weights using the six methods mentioned above is generated using (5.1).

Measurement of Goodness of Fit

In order to quantify the desirability of various methods under the same conditions, two different measures of 'goodness of fit' will be used. The first measure is essentially an error term based on an Euclidean distance measure, dik, between the parameter values and the estimated values while the second measure is an error term based on a city block distance measure, d_{2k}. The Euclidean distance measure implies the shortest distance between two points and the city block distance measure implies a longer distance between two points in a geometric sense (Zeleny, 1982). These are given by;

$$d_{1k} = \frac{1}{R} \sum_{r=1}^{R} \left[\sum_{i=1}^{n} (W_i - W_i^{(r)})^2 \right]^{1/2} \qquad (5.2)$$

and

$$d_{2k} = \frac{1}{R} \sum_{r=1}^{R} \sum_{i=1}^{n} \left| W_i - W_i^{(r)} \right| \qquad (5.3)$$

where k represents the weighting method such as 1 for the Model 1, 2 for the Model 2, 3 for the Model 3, 4 for the

Saaty's method, 5 for the Cogger and Yu's method, and 6 for the Takeda, et al.'s method,

r is the replication number, $r=1,2,\ldots,R$,

i is the criterion number, $i=1,2,\ldots,n$,

W_i is the true weight of criterion i, and

$W_i^{(r)}$ is the estimated weight of criterion i at the r^{th} replication.

Deciding the Number of Replications

In order to determine the significance between the true weights and the estimated weights from model k based on Euclidean distance measure of goodness of fit (the same procedure can be applied to city block distance measure of error), it is necessary to show that a distance between the true weight vector and the estimated weight vector is significant when Type I error is α and Type II error is β. α refers the probability of falsely rejecting the null hypothesis rather than accepting it and β refers the probability of falsely accepting the null hypothesis rather than rejecting it. The appropriate formula (Steel and Torrie, 1980) for determining R when the hypothesis alternatives are one sided, is given by (5.4)

$$R = \frac{(Z_\alpha + Z_\beta)^2\ \sigma^2}{d_s^2} \tag{5.4}$$

where d_f is a difference between the true weight vector and the estimated weight vector, f identifies the measure of goodness of fit used such as 1 for a Euclidean distance measure and 2 for a city block distance measure, and σ^2 is the variance of these differences. Since R is likely to be a fractional value, the next higher integer value will be used for R. This formula has obvious difficulty. σ^2 is rarely known and so it must be estimated. If σ^2 is underestimated, the number of replications, R, is too small; if σ^2 is overestimated, then R is too large. In this research, to overcome this problem, a pilot study was used to estimate σ^2. The calculated variances of the differences between the true weight vector and the estimated weight vector, using a sample size of 30, for the six models are shown in Table 5.2. NC represents the number of criteria. The decision making settings used for obtaining the results of Table 5.2 are W=(.55,.3,.15) for NC=3, W=(.2,.4,.1,.3) for NC=4, and W=(.25,.3,.15,.1,.2) for NC=5. When the number of criteria is three (NC=3) and the Euclidean distance measure is used, the maximum estimated variance of differences is .00221. This maximum value was used for conservative purposes as an estimated variance in order to determine the appropriate number of replications for the simulation run of NC=3. The number of replications for the simulation runs was determined by (5.4) and reported in Table 5.3.

TABLE 5.2

ESTIMATED VALUES OF σ^2 WHEN N=30

Model	NC=3		NC=4		NC=5	
(k)	Var(d_{1k})	Var(d_{2k})	Var(d_{1k})	Var(d_{2k})	Var(d_{1k})	Var(d_{2k})
1	.00072	.00170	.00094	.00271	.00044	.00135
2	.00221	.00485	.00162	.00405	.00229	.00707
3	.00147	.00325	.00140	.00355	.00203	.00659
4	.00064	.00213	.00048	.00156	.00038	.00134
5	.00097	.00227	.00052	.00118	.00038	.00118
6	.00199	.00546	.00079	.00216	.00060	.00224
MAX	.00221	.00546	.00162	.00405	.00229	.00707

TABLE 5.3

NUMBER OF REPLICATIONS FOR SIMULATION RUN
WHEN $\alpha=\beta=0.025$

Number of criteria	NC=3		NC=4		NC=5	
goodness of fit*	E	C	E	C	E	C
d_z values used	0.05	0.08	0.045	0.07	0.04	0.07
Number of replication	16	16	15	15	24	25

*E stands for Euclidean distance measure and C stands for city block distance measure.

Experimental Design

The experimental design for the simulation is summarized in Figure 5.1. This experiment will be repeated for each of eight decision making settings introduced by Takeda, et al. (1987) which were shown in Table 5.1. At each replication, the C' matrix is generated from equation (5.1) and the six methods are applied in order to estimate

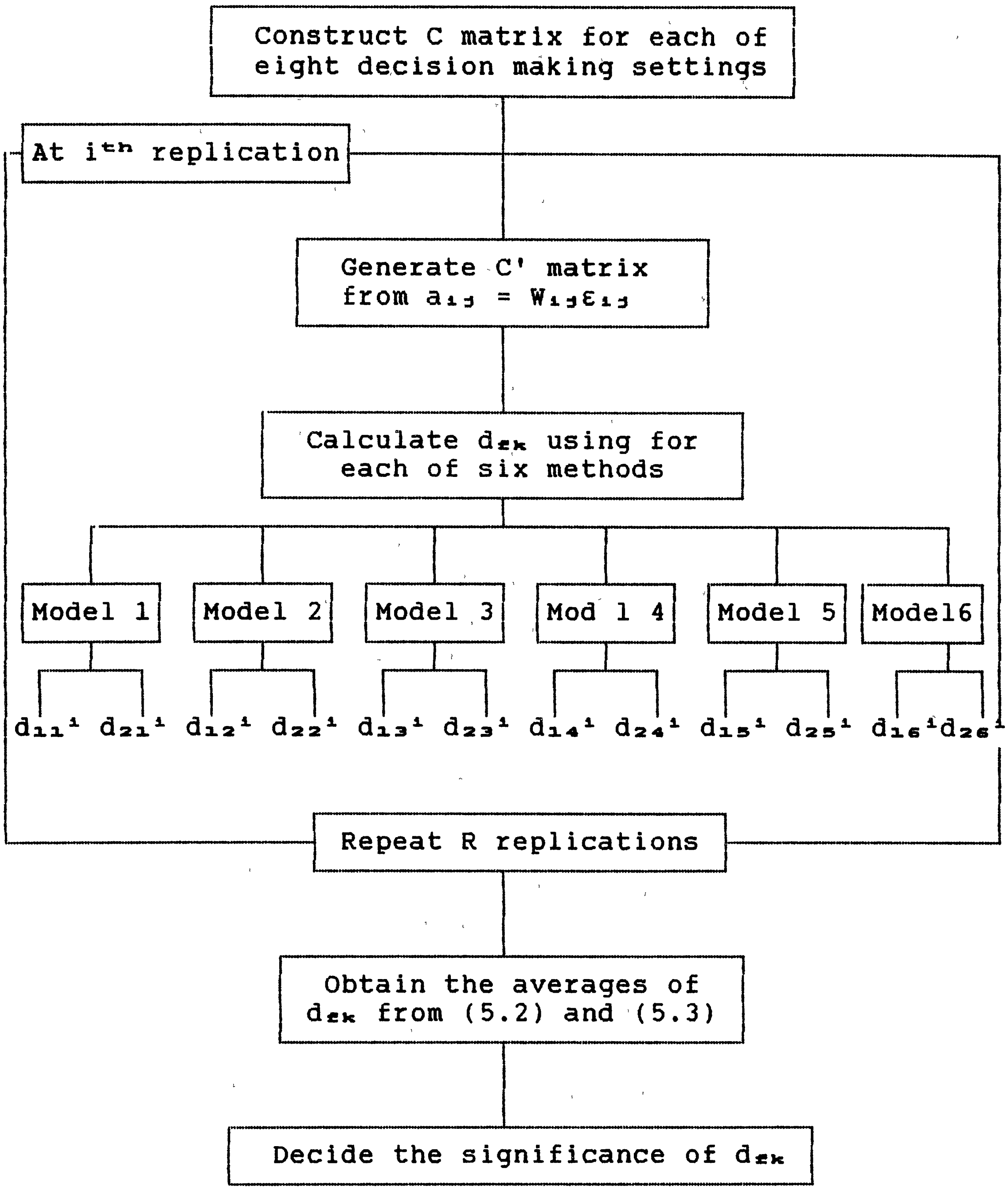

Figure 5.1. Summary of Experimental Design for Simulation

their own weight vector. Then, the distance measures are calculated by using the two different measures of goodness of fit. Repeating R times, the averages of distance measures are obtained using (5.2) and (5.3). The statistical test for determining which method is superior can be carried out. For the statistical test to determine the significance of the difference between models, the hypotheses are set up as follows:

Null Hypothesis (H_0): $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$

Alternative Hypothesis (H_1): At least one is different

Test Statistic : $F = \dfrac{\sum_{k=1}^{6} (d_{zk} - d'_z)^2 / 5}{S_p^2}$

Critical Region : Reject H_0 if

$F > F(dfn, dfd, \alpha)$

where μ_k, k=1,2,...,6 is the population mean of the differences between the true weight vector and the estimated weight vector from model k. d_{zk} is calculated from equation (5.2) and (5.3). d'_z is the average of d_{zk} for k=1,2,...,6. S_p^2 is pooled sample variance.

Duncan's new multiple-range test (1955) is used to find out which model is different from which model when null hypothesis, H_0, is rejected.

Results, Comparison, and Analysis

In this section, the results of the simulations are presented, compared, and analyzed in order to decide if one or more methods are better than the others. Eight decision setting problems introduced by Takeda, et al. (1987) and shown in Table 5.1 were used for the simulation run.

The structure of the tables (see Table 5.4 for example) reporting the simulation results is as follows. In the table heading, the true weight vector W is given first. Second, the decision maker's confidence, or degree of knowledge when comparing criterion i with criterion j represented by D(i,j) for Takeda, et al.'s method is given. Third, the number of replications, R, for detecting a particular difference is reported. Fourth, the seed number used for generating uniform random numbers is given. The uniform random numbers were generated from the RANF introduced by Chandler (1970).

The average of weights, averages of differences between true weight vector and estimated weight vector, and the variation of those differences are then reported for the three models, developed in Chapter III which are represented by Model 1, Model 2, and Model 3 respectively. The solution given by Saaty's approach is represented by Model 4, the solution obtained from Cogger and Yu's method is represented by Model 5, and the Graded Eigenvector Method developed by Takeda, et al. is represented by Model 6.

Table 5.4, based on R=16, indicates that the estimated weight vectors from Model 1 and Model 5 are preferred over the others based on the calculated d'_1 and d'_2.

In Table 5.5 and Table 5.6, the F value is obtained in order to determine the existence of a statistical significance between models by dividing the between models' mean square by the within models' mean square. The calculated F value is compared with the tabular F value for 5 and 90 degrees of freedom to decide whether to accept the null hypothesis of no difference between population means or the alternative hypothesis of a difference. The tabular F value for 5 and 90 degrees of freedom is 2.33 at the 5 percent of significance level. Since calculated F does not exceed 5 percent tabular F, the experiment provides no evidence of real differences between models for both measures.

TABLE 5.4

SIMULATION RESULTS BASED ON W=(0.15,0.55,0.3), D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3 R=16, NC=3, SEED=0

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.1627,.5256,.3117)	.0435	.0007	.0684	.0017
2	(.1532,.5851,.2617)	.0672	.0015	.1029	.0033
3	(.1567,.5791,.2642)	.0685	.0014	.1049	.0030
4	(.2512,.4316,.3172)	.1607	.0015	.2464	.0036
5	(.1470,.5568,.2962)	.0323	.0002	.0499	.0006
6	(.1469,.5552,.2979)	.0780	.0016	.1216	.0040

TABLE 5.5

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.4

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0103	.0021	1.7826
Within Models	90	.1035	.0012	
Total	95	.1138		

TABLE 5.6

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.4

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0240	.0048	1.7752
Within Models	90	.2430	.0027	
Total	95	.2670		

Table 5.7, differs from Table 5.4 only in the true weight vector, and also indicates that Model 1 and Model 5 are the preferred solution methods based on d'_1 and d'_2. If one had to rank the models in the order of generating a better weight vector to come behind Model 1 and Model 5 based on calculated d'_1 and d'_2, it would be Model 2, Model 3, Model 6, and Model 4 respectively.

TABLE 5.7

SIMULATION RESULTS BASED ON W=(0.3,0.15,0.55),
D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3
R=16, NC=3, SEED=0

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.3087,.1679,.5234)	.0462	.0007	.0702	.0019
2	(.2418,.1513,.6069)	.0849	.0018	.1305	.0038
3	(.2428,.1559,.6013)	.0872	.0018	.1333	.0039
4	(.3921,.1189,.4890)	.1199	.0016	.1901	.0042
5	(.2968,.1539,.5493)	.0449	.0007	.0686	.0016
6	(.2980,.1552,.5468)	.0874	.0016	.1373	.0042

In Table 5.8 and Table 5.9, the F value is obtained in order to determine the existence of a statistical significance between models by dividing the between models' mean square by the within models' mean square. Since calculated F does not exceed 5 percent tabular F, the experiment provides no evidence of real differences between models for both measures.

TABLE 5.8

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE
MEASURE DATA OF TABLE 5.7

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0041	.0008	0.5985
Within Models	90	.1230	.0014	
Total	95	.1271		

TABLE 5.9

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.7

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0106	.0021	0.6490
Within Models	90	.2940	.0033	
Total	95	.3046		

Table 5.10 and Table 5.13 yield Model 1 and Model 5 again as best models based on the calculated d'_1 and d'_2, but a somewhat different result on the other models. The reasons are most likely due to the different values of D(i,j) used in Model 6 and different seed number used in all models. In this case, Model 6, Model 3, Model 2, and Model 4 is the order of generating better weight vectors behind model 1 and Model 5. Again the comparison is based on the calculated d'_1 and d'_2. But, from statistical point of view, there is no evidence of any differences between models as can be seen in Table 5.11, Table 5.12, Table 5.14, and Table 5.15.

TABLE 5.10

SIMULATION RESULTS BASED ON W=(0.55,0.3,0.15), D(1,2)=0.9, D(1,3)=0.6, D(2,3)=0.3 R=16, NC=3, SEED=472

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.5555,.2825,.1620)	.0380	.0003	.0599	.0008
2	(.5890,.2589,.1521)	.0786	.0015	.1208	.0035
3	(.5846,.2607,.1547)	.0742	.0013	.1146	.0033
4	(.4377,.3770,.1853)	.1430	.0006	.2253	.0018
5	(.5481,.3015,.1504)	.0431	.0007	.0686	.0019
6	(.5480,.3014,.1506)	.0594	.0016	.0928	.0038

TABLE 5.11

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.10

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0072	.0015	1.5000
Within Models	90	.0900	.0010	
Total	95	.0972		

TABLE 5.12

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.10

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0179	.0036	1.4185
Within Models	90	.2265	.0025	
Total	95	.2444		

TABLE 5.13

SIMULATION RESULTS BASED ON W=(0.55,0.3,0.15),
D(1,2)=0.9, D(1,3)=0.8, D(2,3)=0.3
R=16, NC=3, SEED=0

Model	W'	d'_1	$\sigma_{d'_1}^2$	d'_2	$\sigma_{d'_2}^2$
1	(.5536,.2831,.1633)	.0450	.0007	.0699	.0018
2	(.5833,.2654,.1513)	.0745	.0015	.1141	.0032
3	(.5826,.2608,.1566)	.0736	.0013	.1118	.0028
4	(.4324,.3795,.1881)	.1498	.0008	.2365	.0028
5	(.5412,.3083,.1505)	.0449	.0010	.0709	.0024
6	(.5406,.3069,.1525)	.0675	.0023	.1039	.0053

TABLE 5.14

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE
MEASURE DATA OF TABLE 5.13

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0075	.0015	1.1538
Within Models	90	.1140	.0013	
Total	95	.1215		

TABLE 5.15

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE
MEASURE DATA OF TABLE 5.13

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0188	.0038	1.2361
Within Models	90	.2745	.0031	
Total	95	.2933		

Table 5.16, Table 5.19, and Table 5.22 present simulation results for the case of NC=4 criteria weights. Utilizing the same true weight vector, differing values of D(i,j) and seed number are used for generating a pairwise comparison matrix and a simulation run.

Table 5.16, again, indicates that model 1 and Model 5 are superior to the other models having smaller values of d'_1 and d'_2. Model 4, which generated the worst weight vector in case of NC=3, becomes fourth when NC=4. There is no differences between models from statistical view point as shown in Table 5.17 and Table 5.18 since calculated F values do not exceed the tabular F value, 2.33 for 5 and 84 degrees of freedom.

TABLE 5.16
SIMULATION RESULTS BASED ON W=(0.2,0.4,0.1,0.3),
D(1,2)=0.7,D(1,3)=0.9, D(1,4)=0.8,D(2,3)=0.7,
D(2,4)=0.6,D(3,4)=0.4, R=15, NC=4, SEED=0

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2088,.3888,.1038,.2986)	.0410	.0007	.0681	.0019
2	(.1515,.4585,.1017,.2883)	.1178	.0016	.1916	.0042
3	(.1580,.4532,.1029,.2859)	.1173	.0011	.1903	.0032
4	(.2006,.3845,.1055,.3094)	.0673	.0008	.1165	.0024
5	(.2020,.4127,.0940,.2913)	.0421	.0007	.0677	.0014
6	(.2059,.4106,.0880,.2955)	.0622	.0006	.1038	.0017

TABLE 5.17

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.16

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0061	.0012	1.3269
Within Models	84	.0769	.0009	
Total	89	.0830		

TABLE 5.18

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.16

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0157	.0031	1.2745
Within Models	84	.2071	.0025	
Total	89	.2444		

Table 5.19 uses the same true weight vector but different D(i,j) and seed number used from those in Table 5.16. On the average, the models can be ranked from better to worse solutions as Model 1, Model 5, Model 6, Model 4, Model 3, and Model 2. No statistical differences are indicated between models as shown in Table 5.20 and Table 5.21.

TABLE 5.19

SIMULATION RESULTS BASED ON W=(0.2,0.4,0.1,0.3), D(1,2)=0.8,D(1,3)=0.7, D(1,4)=0.9,D(2,3)=0.4, D(2,4)=0.6,D(3,4)=0.5, R=15, NC=4, SEED=40

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2029,.3806,.1009,.3156)	.0460	.0003	.0808	.0008
2	(.1532,.4400,.1006,.3062)	.0992	.0013	.1645	.0038
3	(.1574,.4538,.1039,.2849)	.0974	.0012	.1601	.0030
4	(.2190,.3522,.1172,.3116)	.0743	.0007	.1320	.0026
5	(.2062,.3869,.1064,.3005)	.0488	.0003	.0817	.0009
6	(.2182,.3854,.1122,.2842)	.0677	.0006	.1148	.0016

TABLE 5.20

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.19

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0026	.0005	0.7153
Within Models	84	.0616	.0007	
Total	89	.0642		

TABLE 5.21

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.19

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0067	.0013	0.6360
Within Models	84	.1777	.0021	
Total	89	.1845		

Table 5.22, the same true weight vector but different D(i,j) and seed number used from Table 5.16 and Table 5.19, indicates that Model 1 is ranked first based on the calculated d'_1 and d'_2. If one had to pick a method to come in second place behind Model 1 based on smaller values of d'_1 and d'_2, it would be Model 5. Model 6 would be picked third, Model 4 fourth, Model 2 fifth, and Model 3 would be sixth.

Table 5.23 and Table 5.24 indicate no statistical significance between models since the calculated F values do not exceed 5 percent tabular F value for 5 and 84 degrees of freedom.

TABLE 5.22

SIMULATION RESULTS BASED ON W=(0.2,0.4,0.1,0.3), D(1,2)=0.7,D(1,3)=0.6,D(1,4)=0.8, D(2,3)=0.5, D(2,4)=0.6,D(3,4)=0.3, R=15, NC=4, SEED=921

Model	W'	d'_1	$\sigma_{d'_1}^2$	d'_2	$\sigma_{d'_2}^2$
1	(.2033,.3828,.1092,.3047)	.0419	.0008	.0700	.0021
2	(.1449,.4532,.1040,.2979)	.1034	.0033	.1721	.0079
3	(.1479,.4583,.1040,.2898)	.1115	.0067	.1812	.0142
4	(.1914,.3790,.1246,.3050)	.0669	.0004	.1171	.0011
5	(.1952,.4065,.1046,.2937)	.0454	.0004	.0769	.0012
6	(.2010,.4045,.1048,.2897)	.0614	.0005	.1035	.0013

TABLE 5.23

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE
MEASURE DATA OF TABLE 5.22

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0043	.0009	0.4262
Within Models	84	.1693	.0020	
Total	89	.1736		

TABLE 5.24

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE
MEASURE DATA OF TABLE 5.22

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0111	.0022	0.4791
Within Models	84	.3892	.0046	
Total	89	.4003		

Table 5.25, based on R=25 and NC=5, yields quite similar results to those in Table 5.22 except Model 4 is now in third place and Model 3 is in fifth place. No statistical significance between models is detected. As shown in Table 5.26 and Table 5.27, the calculated F values do not exceed 5 percent tabular F value, 2.29, for 5 and 144 degrees of freedom.

TABLE 5.25

SIMULATION RESULTS BASED ON W=(.25,.3,.15,.1,.2), D(1,2)=0.6, D(1,3)=0.7, D(1,4)=0.8, D(1,5)=0.6, D(2,3)=0.7,D(2,4)=0.6,D(2,5)=0.6,D(3,4)=0.5, D(3,5)=0.8,D(4,5)=0.3, R=25, NC=5, SEED=0

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2520,.2878,.1507,.1038,.2057)	.0381	.0005	.0671	.0015
2	(.2560,.3705,.0927,.1003,.1805)	.1211	.0021	.2173	.0059
3	(.2541,.3714,.0939,.0999,.1807)	.1198	.0019	.2157	.0053
4	(.2429,.3036,.1481,.1041,.2014)	.0519	.0004	.0955	.0014
5	(.2461,.3058,.1473,.0981,.2027)	.0401	.0004	.0725	.0011
6	(.2498,.3114,.1424,.0979,.1985)	.0578	.0006	.1074	.0023

TABLE 5.26

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.25

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0075	.0015	1.5186
Within Models	144	.1416	.0010	
Total	149	.1491		

TABLE 5.27

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.25

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	5	.0239	.0048	1.6406
Within Models	144	.4199	.0029	
Total	149	.4438		

Even though there were no statistical evidences of significance between models indicated, Model 1 has been ranked the first, based on smallest values of the calculated d'_1 and d'_2, for all the decision making setting problems except problems 1 and 2 as summarized in Table 5.28. The largest problem used by Takeda, et al. (1987) has five criteria. What if the problem size is larger than five-criteria problem? Additional simulation runs were made for the problems of NC=7 and NC=9 shown in Table 5.29 after eliminating two worst models based on largest values of d'_1 and d'_2 which were Model 2 and Model 3.

TABLE 5.28

SUMMARY OF THE SIMULATION RESULTS FOR EIGHT DECISION MAKING SETTING PROBLEMS

Decision Making Setting Problem	Number of Criteria	First Ranked Model
1	3	Model 5
2	3	Model 5
3	3	Model 1
4	3	Model 1
5	4	Model 1
6	4	Model 1
7	4	Model 1
8	5	Model 1

The true weight vectors, W, are provided by this author. The decision maker's confidence, D(i,j), when comparing criterion i with criterion j for the Model 6 is generated by (0,1) uniform random numbers since it is not available from previous work.

TABLE 5.29

PROBLEM DESCRIPTIONS FOR ADDITIONAL SIMULATION RUN

i	W	D(i,j)
7	(.2,.12,.15,.1,.2,.05,.18)	(0,1) Uniform
9	(.2,.12,.08,.1,.17,.05,.15,.1,.03)	Random Numbers

Table 5.30, Table 5.33, and Table 5.36 indicate that the weights from Model 1 are the best ones based on the calculated values of d'_1 and d'_2. Model 6 would be picked second, Model 4 third. No weights can be calculated from Model 5. As explained in Chapter II, a weight from Model 5, due to Cogger and Yu (1985), is the geometric mean of all the weights generated from the possible index orders. An index order of 2520 must be enumerated when NC=7. A severe underflow problem is encountered when multiplying the numbers which are less than 1.0 2520 times. At this point, mathematics of this technique prohibits the calculation of the geometric mean when the problem has more than five criteria.

Table 5.31, Table 5.32, Table 5.34, Table 5.35, Table 5.37, and Table 5.38 indicate that statistical significance between models exists since all calculated F values exceed 5 percent tabular F value, 2.39, for 2 and 74 degrees of freedom.

TABLE 5.30

SIMULATION RESULTS BASED ON W=(.2,.12,.15,.1,.2,.05,.18), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=7, SEED=3211

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2024,.1212,.1502,.1016,.1891, .0538,.1817)	.0293	.0002	.0601	.0007
4	(.0874,.1298,.0831,.2303,.1435, .1534,.1725)	.2310	.0007	.5149	.0037
5	No Weights Estimated				
6	(.2012,.1227,.1441,.0983,.1964, .0460,.1913)	.0590	.0003	.1232	.0010

*D(1,2)=.68,D(1,3)=.62,D(1,4)=.97,D(1,5)=.82,D(1,6)=.81, D(1,7)=.53,D(2,3)=.67,D(2,4)=.95,D(2,5)=.40,D(2,6)=.73, D(2,7)=.64,D(3,4)=.65,D(3,5)=.40,D(3,6)=.83,D(3,7)=.85, D(4,5)=.96,D(4,6)=.92,D(4,7)=.81,D(5,6)=.93,D(5,7)=.05, D(6,7)=.54,

TABLE 5.31

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.30

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.0237	.0119	29.6454
Within Models	72	.0288	.0004	
Total	74	.0525		

TABLE 5.32

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.30

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.1214	.0607	33.7272
Within Models	72	.1296	.0018	
Total	74	.2510		

Duncan's (1955) new multiple-range test is used to see which model is different from which model since H_o is rejected. This test consists of computing the least significant ranges, R_p, by Eq. 5.5 and applying it to differences between all pairs of means.

$$R_p = q_\alpha(p,fe)\ S_d \tag{5.5}$$

where q_α is obtained from significant studentized ranges for new multiple-range test (Steel and Torrie, 1980), p is the number of between models, fe is error df, and S_d is the standard error of a between models' mean.

For the Euclidean distance measure data of Table 5.30, the values for Duncan's test are summarized as follows:

p	2	3
$q_\alpha(p,72)$	2.83	2.98
R_p	0.0113	0.0119

(5.6)

A summary of the test results, using d'_{ik} for k=1,4,6, follows.

Model 1	Model 6	Model 4
.0293	.0590	.2310

Duncan's test indicates that the average distance between the true weight vector and estimated weight vector from Model 1 is different from those from the other two models. The same test result occurs for the city block distance measure.

TABLE 5.33

SIMULATION RESULTS BASED ON W=(.2,.12,.15,.1,.2,.05,.18), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=7, SEED=4444

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2034,.1239,.1513,.1013,.1911, .0522,.1768)	.0303	.0002	.0606	.0007
4	(.0930,.1162,.0791,.2390,.1373, .1592,.1762)	.2337	.0006	.5227	.0037
5	No Weights Estimated				
6	(.1980,.1185,.1449,.1008,.2112, .0485,.1781)	.0518	.0004	.1080	.0013

*D(1,2)=.68,D(1,3)=.62,D(1,4)=.97,D(1,5)=.82,D(1,6)=.81, D(1,7)=.53,D(2,3)=.67,D(2,4)=.95,D(2,5)=.40,D(2,6)=.73, D(2,7)=.64,D(3,4)=.65,D(3,5)=.40,D(3,6)=.83,D(3,7)=.85, D(4,5)=.96,D(4,6)=.92,D(4,7)=.81,D(5,6)=.93,D(5,7)=.05, D(6,7)=.54,

TABLE 5.34

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.33

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.0250	.0125	31.2173
Within Models	72	.0288	.0004	
Total	74	.0538		

TABLE 5.35

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.33

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.1293	.0646	34.0140
Within Models	72	.1368	.0019	
Total	74	.2661		

Duncan's new multiple-range test is applied to see which model is different from which model since H_0 is rejected. A summary of the test results, using d'_{1k} for k=1,4,6 in Table 5.33 and (5.6), follows.

Model 1	Model 6	Model 4
.0303	.0518	.2337

Duncan's test indicates that the average distance between the true weight vector and estimated weight vector from Model 1 is different from those from the other two models. The same test result is made for the city block distance measure.

TABLE 5.36

SIMULATION RESULTS BASED ON W=(.2,.12,.15,.1,.2,.05,.18), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=7, SEED=5678

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2047,.1208,.1517,.0999,.1881, .0528,.1820)	.0288	.0002	.0584	.0006
4	(.0904,.1207,.0831,.2274,.1452, .1595,.1737)	.2281	.0005	.5123	.0035
5	No Weights Estimated				
6	(.1924,.1210,.1531,.1037,.1956, .0499,.1843)	.0532	.0003	.1095	.0011

*D(1,2)=.68,D(1,3)=.62,D(1,4)=.97,D(1,5)=.82,D(1,6)=.81, D(1,7)=.53,D(2,3)=.67,D(2,4)=.95,D(2,5)=.40,D(2,6)=.73, D(2,7)=.64,D(3,4)=.65,D(3,5)=.40,D(3,6)=.83,D(3,7)=.85, D(4,5)=.96,D(4,6)=.92,D(4,7)=.81,D(5,6)=.93,D(5,7)=.05, D(6,7)=.54,

TABLE 5.37

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.36

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.0236	.0118	35.4529
Within Models	72	.0240	.0003	
Total	74	.0476		

TABLE 5.38

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.36

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	2	.1236	.0618	35.6619
Within Models	72	.1248	.0017	
Total	74	.2484		

Duncan's new multiple-range test is applied to see which model is different from which model since H_0 is rejected. A summary of the test results, using d'_{ik} for k=1,4,6 in Table 5.36 and (5.6), follows.

Model 1	Model 6	Model 4
.0288	.0532	.2281

Duncan's test indicates that the average distance between the true weight vector and estimated weight vector from Model 1 is different from those from the other two models. The same test result occurs for the city block distance measure.

Additional simulation runs were made using NC=9 problem after eliminating Model 4 and Model 5 from further considerations since Model 4 was determined as worst model by Duncan's new multiple-range test and as mentioned before, no weights can be estimated from Model 5 when the number of criteria is more than five.

Table 5.39, Table 5.42, and Table 5.45 indicate that the weights from Model 1 are better than the weights from Model 6 based on smaller values of the calculated d'_1 and d'_2. The same F test was applied in order to determine the existence of a statistical significance between two models.

As indicated in Table 5.40 and Table 5.41, no statistical differences between two models are detected since calculated F values do not exceed the 5 percent tabular F value, 2.84, for 1 and 48 degrees of freedom.

TABLE 5.39

SIMULATION RESULTS BASED ON W=(.2,.12,.08,.1,.17,.05,.15, .1,.03), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=9, SEED=6156

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2020,.1209,.0809,.1014,.1566, .0514,.1513,.1033,.0322)	.0252	.0003	.0535	.0009
6	(.2006,.1218,.0796,.1003,.1669, .0481,.1542,.0990,.0295)	.0464	.0003	.1088	.0011

*D(1,2)=.47,D(1,3)=.17,D(1,4)=.64,D(1,5)=.50,D(1,6)=.89, D(1,7)=.72,D(1,8)=.92,D(1,9)=.55,D(2,3)=.75,D(2,4)=.69, D(2,5)=.37,D(2,6)=.28,D(2,7)=.71,D(2,8)=.17,D(2,9)=.20, D(3,4)=.09,D(3,5)=.03,D(3,6)=.49,D(3,7)=.77,D(3,8)=.63, D(3,9)=.98,D(4,5)=.71,D(4,6)=.77,D(4,7)=.23,D(4,8)=.88, D(4,9)=.74,D(5,6)=.09,D(5,7)=.96,D(5,8)=.31,D(5,9)=.20, D(6,7)=.80,D(6,8)=.33,D(6,9)=.58,D(7,8)=.13,D(7,9)=.25, D(8,9)=.14,

TABLE 5.40

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.39

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0002	.0002	0.7491
Within Models	48	.0144	.0003	
Total	49	.0146		

TABLE 5.41

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.39

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0015	.0015	1.5290
Within Models	48	.0480	.0010	
Total	49	.0495		

TABLE 5.42

SIMULATION RESULTS BASED ON W=(.2,.12,.08,.1,.17,.05,.15, .1,.03), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=9, SEED=7312

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.2013,.1206,.0804,.1012,.1641, .0511,.1499,.0994,.0320)	.0202	.0000	.0467	.0002
6	(.2051,.1227,.0792,.0994,.1626, .0482,.1484,.1043,.0301)	.0507	.0003	.1163	.0013

*D(1,2)=.47,D(1,3)=.17,D(1,4)=.64,D(1,5)=.50,D(1,6)=.89, D(1,7)=.72,D(1,8)=.92,D(1,9)=.55,D(2,3)=.75,D(2,4)=.69, D(2,5)=.37,D(2,6)=.28,D(2,7)=.71,D(2,8)=.17,D(2,9)=.20, D(3,4)=.09,D(3,5)=.03,D(3,6)=.49,D(3,7)=.77,D(3,8)=.63, D(3,9)=.98,D(4,5)=.71,D(4,6)=.77,D(4,7)=.23,D(4,8)=.88, D(4,9)=.74,D(5,6)=.09,D(5,7)=.96,D(5,8)=.31,D(5,9)=.20, D(6,7)=.80,D(6,8)=.33,D(6,9)=.58,D(7,8)=.13,D(7,9)=.25, D(8,9)=.14,

Table 5.43 and Table 5.44 indicate that there is some statistical differences between two models since calculated F values exceed the 5 percent tabular F value, 2.84, for 1 and 48 degrees of freedom.

TABLE 5.43

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.42

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0005	.00047	3.1008
Within Models	48	.0072	.00015	
Total	49	.0077		

TABLE 5.44

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.42

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0024	.00240	3.2294
Within Models	48	.0360	.00075	
Total	49	.0384		

TABLE 5.45

SIMULATION RESULTS BASED ON W=(.2,.12,.08,.1,.17,.05,.15, .1,.03), D(i,j)* IS GENERATED FROM (0,1) UNIFORM RANDOM NUMBERS, R=25, NC=9, SEED=8866

Model	W'	d'_1	$\sigma_{d'1}^2$	d'_2	$\sigma_{d'2}^2$
1	(.1997,.1222,.0804,.1031,.1630, .0502,.1491,.1009,.0314)	.0240	.0001	.0513	.0004
6	(.2034,.1151,.0778,.1014,.1707, .0546,.1412,.1046,.0312)	.0496	.0002	.1144	.0009

*D(1,2)=.47,D(1,3)=.17,D(1,4)=.64,D(1,5)=.50,D(1,6)=.89, D(1,7)=.72,D(1,8)=.92,D(1,9)=.55,D(2,3)=.75,D(2,4)=.69, D(2,5)=.37,D(2,6)=.28,D(2,7)=.71,D(2,8)=.17,D(2,9)=.20, D(3,4)=.09,D(3,5)=.03,D(3,6)=.49,D(3,7)=.77,D(3,8)=.63, D(3,9)=.98,D(4,5)=.71,D(4,6)=.77,D(4,7)=.23,D(4,8)=.88, D(4,9)=.74,D(5,6)=.09,D(5,7)=.96,D(5,8)=.31,D(5,9)=.20, D(6,7)=.80,D(6,8)=.33,D(6,9)=.58,D(7,8)=.13,D(7,9)=.25, D(8,9)=.14,

TABLE 5.46

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.45

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0003	.00033	2.1845
Within Models	48	.0072	.00015	
Total	49	.0075		

Table 5.47 indicates that there is some statistical differences between the two models since calculated F values exceed the 5 percent tabular F value, 2.84, for 1 and 48 degrees of freedom.

TABLE 5.47

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.45

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.0020	.00200	3.0628
Within Models	48	.0312	.00065	
Total	49	.0332		

Finally, Model 1 should be selected if one had to pick a best methodology to estimate weight. There were no statistical significance indicated when the number of criteria is less than or equal to five, but practically speaking, Model 1 was always ranked number one except in decision making setting problem 1 and 2 as shown in Table 5.28. When the number of criteria is more than five, the differences calculated from the true weight and the estimated weight from Model 1 are significantly different from those of the other methods. This significance implies that the weight estimated from Model 1 is better than the others based on F test and Duncan's new multiple-range test. The second best methodology would be Model 6 if the number of criteria is six or more. For the small size problems which have less than six criteria, then Model 5 is the recommended second choice. There were no statistical significances indicated between Model 5 and Model 6 when the number of criteria is less than or equal to five, but the weights from Model 5 were always better than those from Model 6 based on smaller values of d'_1 and d'_2.

Discussions on Multiple Decision Makers

In order to compare the two procedures introduced in Chapter III for estimating weights under the situation of having multiple decision makers, three of the decision making setting problems and measurements of goodness of fit are used. Estimating weights using C'_{avg} matrix (3.13) obtained by averaging N pairwise comparison matrices was the first procedure. Estimating weights by averaging the N individual weights calculated from N C'_q matrix (3.12) was the second procedure.

For this study, Model 1, which is determined as a best model in this research, is used for calculating the average and the variance of the differences between the true weight vector and the estimated weight vectors from the two procedures. It is assumed that two decision makers are involved in this problem. It is also assumed that the variation of the decision makers' judgment follows a uniform distribution (0.5, 1.5) and (0.3, 1.7) respectively. The decision making settings used in this comparison are W=(.55,.3,.15) for NC=3, W=(.2,.4,.1,.3) for NC=4, and W=(.25,.3,.15,.1,.2) for NC=5. Table 5.48, based on N=30 replications, indicates that procedure 1 generates better weights all the time, regardless of the decision making setting problems, based on smaller values of the calculated d'_1 and d'_2.

TABLE 5.48

SIMULATION RESULTS FOR TWO DECISION MAKERS WHEN N=30

P*	NC=3				NC=4				NC=5			
	d'_1	$\sigma^2_{d'_1}$	d'_2	$\sigma^2_{d'_2}$	d'_1	$\sigma^2_{d'_1}$	d'_2	$\sigma^2_{d'_2}$	d'_1	$\sigma^2_{d'_1}$	d'_2	$\sigma^2_{d'_2}$
1	.065	.002	.100	.004	.069	.003	.117	.007	.070	.002	.124	.005
2	.076	.002	.117	.003	.077	.003	.130	.007	.082	.002	.145	.005

*P stands for procedures, 1 for procedure 1 and 2 for procedure 2.

The F test is applied in order to determine the existence of a statistical difference between the two procedures. As can be seen in the following Tables, no statistical differences are indicated since calculated F values do not exceed 5 percent tabular F value, 2.79, for 1 and 58 degrees of freedom.

TABLE 5.49

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=3

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00006	.00006	0.0303
Within Models	58	.11600	.00200	
Total	59	.11606		

TABLE 5.50

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=3

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00003	.00003	0.0107
Within Models	58	.17400	.00300	
Total	59	.17403		

TABLE 5.51

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=4

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00007	.00007	0.0360
Within Models	58	.11600	.00200	
Total	59	.11607		

TABLE 5.52

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=4

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00014	.00014	0.0413
Within Models	58	.20300	.00350	
Total	59	.20314		

TABLE 5.53

ANALYSIS OF VARIANCE FOR EUCLIDEAN DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=5

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00008	.00008	0.0121
Within Models	58	.40600	.00700	
Total	59	.40608		

TABLE 5.54

ANALYSIS OF VARIANCE FOR CITY BLOCK DISTANCE MEASURE DATA OF TABLE 5.48 WHEN NC=5

Source of Variation	df	Sum of Square	Mean Square	F
Between Models	1	.00022	.00022	0.0441
Within Models	58	.29000	.00500	
Total	59	.29022		

CHAPTER VI

SUMMARY, CONCLUSIONS, AND RECOMMEMDATIONS

This chapter summarizes all the steps carried out in order to fulfill the goal and objectives of this research. Conclusions from this research are then provided. Finally, recommendations for future work and possible extensions of this research are outlined.

Summary

Chapter I of this research provides the problem statement. Introduction of the background of various weighting methods is given. The research goal which involves several objectives is then identified. An extensive literature survey of various weighting methods is given in Chapter II. Chapter III develops the new weighting methods employing the minimization of judgmental error and the refinement of decision maker's inconsistency using pairwise comparisons and linear programming. This research contributes the idea of considering the minimization of a decision maker's judgmental error unlike other subjective approaches. This research also contributes to refining a decision maker's inconsistency by using all a_{ij} in pairwise comparisons when estimating weights. A comprehensive, interactive computer program has been developed and described in Chapter IV. This aspect provides benefits to both decision makers and researchers. This interactive feature of the program will be a great asset in communicating with decision makers. The results of simulation for the purpose of comparison and analysis are provided in Chapter V.

In order to fulfill the research goal and objectives, the following accomplishments have been achieved:

1. Three analytical models based on the minimization of a decision maker's judgmental error and refinement of a decision maker's inconsistency have been developed. These three models use the same pairwise comparison matrix as used in various eigen-vector methods.
2. Two procedures of estimating weights under the situation of having multiple decision makers have been illustrated. These procedures use the same pairwise comparison matrices as mentioned before.
3. An interactive and comprehensive computer program has been developed and designed. This program implements six weight estimation methods of the (1) Proposed Model 1, (2) Proposed Model 2, (3) Proposed Model 3, (4) Saaty's Method, (5) Cogger and Yu's Method, and (6) Takeda, et al.'s Method.

Conclusions and Recommendations

Based on the results obtained in this research, the best model of estimating weights by using a pairwise comparison matrix is the Model 1 developed in the research.

The results of this research are interesting and encouraging. The Model 1 developed in this research estimates weights for MCDM settings more accurately based on the Euclidean distance measure and the city block distance measure than those obtained by the three eigen-vector methods. This is directly due to the effects of the minimization of a decision maker's judgmental error and the refinement of a decision maker's inconsistency.

Possible further work with respect to weight estimating methods using a pairwise comparison matrix is as follows:

1. The intention of adding more constraints to Model 2 and Model 3 was to improve the quality of the weights. But, adding these constraints made the results worse. Finding a better constraining method can be an extension of this research.
2. Two averaging procedures have been used to estimate weights for multiple decision makers. Another method, for instance, $a_{ijmin} \le W_i/W_j \le a_{ijmax}$, where a_{ijmin} is the minimum value of a_{ijq}, and a_{ijmax} is the maximum value of a_{ijq}, $q=1,2,\ldots,N$, may be considered in an extension to this research.

BIBLIOGRAPHY

Beckwith, N. E. and Lehmann, D. R. "The Importance of Differential Weights in Multiple Attribute Models of Consumer Attitude", Journal of Marketing Research, 10, (1973), 141-145.

Belton, V. "A Comparison of the Analytic Hierarchy Process and a Simple Multi-attribute Value Function", European Journal of Operational Research, 26, (1986), 7-21.

Buel, W. D. "A Simplification of Hay's Method of Recording Paired Comparison", Journal of Applied Psychology, 44, 5, (1960), 347-348.

Chandler, J. P. Technical Writing: RANF--Uniformly Distributed Psuedo-Random Numbers, Stillwater, Oklahoma State University, 1970.

Churchman, C. W. and Ackoff, R. L. "An Approximate Measure of Value", Operations Research, 2, (1954), 172-191.

Claudy, J. G. "A Comparison of Five Variable Weighting Procedures", Educational and Psychological Measurement, 32, (1972), 311-322.

Cogger, K. O. and Yu, P. L. "Eigenweight Vector and Least-Distance Approximation for Revealed Preference in Pairwise Weights Ratios", Journal of Optimization Theory and Applications, 46, 4, (August, 1985), 483-491.

Cook, R. L. and Stewart, T. R. "A Comparison of Seven Methods for Obtaining Subjective Descriptions of Judgmental Policy", Organizational Behavior and Human Performance, 13, (1975), 31-45.

Davidson, F. "Dimensions of Utility in a Regional Planning Context", Decision Science, 5, 1, (1974), 91-101.

Delbecq, A. L., Van de Ven, A. H. and Gustafson, D. H. "Group Techniques; Nominal Group and Delphi Processes", Glenview, III.: Scott, Foresman and Co., 1975.

Eckenrode, R. T. "Weighting Multiple Criteria", Management Science, 12, 3, (1965), 180-192.

Einhorn, H. J. and Hogarth, R. M. "Unit Weighting for Decision Making", Organizational Behavior and Human Performance, 13, 2, (1975), 171-192.

Fischer, G. W. "Four Methods for Assessing Multi-attribute Utilities: An Experimental Validation", Engineering Psychology Laboratory, University of Michigan, 1972.

Fischer, G. W. "Convergent Validation of Decomposed Multi-attribute Utility Assessment Procedures for Risky and riskless Decisions", Organizational Behavior and Human Performance, 18, (1977), 295-315.

Frazelle, E. "Suggested Techniques Enable Multi-Criteria Evaluation of Material Handling Alternatives", Industrial Engineering, (February, 1985), 42-48.

Hay, E. N. "A Simple Method of Recording Paired Comparisons", Journal of Applied Psychology, 42, 2, (1958), 139-140.

Hobbs, B. F. "Analytical Multiobjective Decision Methods for Power Plant Siting", Unpublished manuscript, Brookhaven National Laboratory, Division of Regional Studies, Upton, N.Y., September, 1979.

Hobbs, B. F. "A Comparison of Weighting Methods in Power Plant Siting", Decision Science, 11, (1980), 725-737.

Hoffman, P. J. "The Paramorphic Representation of Clinical Judgement", Psychological Bulletin, 57, (1960), 116-131.

Huber, G. P. "Methods for Quantifying Subjective Probabilities and Multi-attribute Utilities", Decision Science, 5, 3, (1974), 430-458.

Jopling, D. "Plant Site Evaluation using Numerical Rankings", Power Engineering, (March, 1974), 56-59.

Keeney, R. L. "Evaluation of Proposed Storage Sites", Operations Research, 27, 1, (1979), 48-64.

Keeney, R. L. and Nair, K. "Nuclear Siting using Decision Analysis", Energy Policy, 5, 3, (1977), 223-231.

Kuester, J. L. and Mize, J. H. Optimization Techniques with FORTRAN, New York: McGraw-Hill, 1973.

Lehman, D. R. "Television Show Preference: Application of a Choice Model", Journal of Marketing Research, 8, (1971), 47-55.

Saaty, T. L. "A Scaling Method for Priorities in Hierarchical Structures", Journal of Mathematical Psychology, 15, 3, (1977), 234-281.

Saaty, T. L. The Analytic Hierarchy Process, New York: McGraw-Hill, 1980.

Schmidt, F. L. "The Relative Efficiency of Regression and Simple Unit Predictor Weights in Applied Differential Psychology", Educational and Psychological Measurement, 31, (1971), 699-714.

Schmidt, F. L. "The Reliability of Differences Between Linear Regression Weights in Applied Differential Psychology", Educational and Psychological Measurement, 32, (1972), 879-886.

Schmitt, N. and Levine, R. L. "Statistical and Subjective Weights: Some Problems and Proposals", Organizational Behavior and Human Performance, 20, (1977), 15-30.

Schoemaker, P. J. H. and Waid, C. C. "An Experimental Comparison of Different Approaches to Determining Weights in Additive Utility Models", Management Science, 28, 2, (1982), 182-196.

Steel, R. G. D. and Torrie, J. H. Principles and Procedures of Statistics: A Biometrical Approach, New York: McGraw-Hill, 1980.

Stimson, D. H. "Utility Measurement in Public Health Decision Making", Management Science, 16, 2, (1969), 17-30.

Takeda, E., Cogger, K. O., and Yu, P. L. "Estimating Criterion Weights using Eigenvectors: A Comparative Study", European Journal of Operations Research, 29, (1987), 360-369.

Trattner, M. H. "Comparison of Three Methods for Assembling Aptitude Test Batteries", Personnel Psychology, 16, (1963), 221-232.

Voelker, A. H. "Facility Siting: An Application of the Nominal Group process(ORNL/NUREG/TM-81)", Oak Ridge, Tenn.: Oak Ridge National Laboratory, 1977.

Watson, J. "Proposed Methodology of Siting Nuclear Power Plants in France (MTR-6772)", McLean, Va.: MITRE Corporation, 1974.

Zeleny, M. Multiple Criteria Decision Making, New York: McGraw-Hill, 1982.

```
C***********************************************************
C                                                          *
C   PURPOSE:                                               *
C        THIS COMPUTER PROGRAM IS DESIGNED FOR ESTIMATING *
C   WEIGHTS USING THREE PROPOSED MODELS INTRODUCED IN      *
C   CHAPTER III UNDER THE INTERACTIVE MODE BETWEEN         *
C   DECISION MAKERS AND RESEARCHERS.                       *
C                                                          *
C   AUTHOR: KOOK JIN NAM                                   *
C           SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT
C           OKLAHOMA STATE UNIVERSITY                      *
C                                                          *
C   DISSERTATION ADVISER: DR. M. PALMER TERRELL            *
C                                                          *
C***********************************************************
C                                                          *
C DEFINITION OF VARIABLES:                                 *
C                                                          *
C      IC     -- NUMBER OF CRITERIA                        *
C                                                          *
C      IMODEL -- MODEL INDICATOR; 1 FOR MODEL1, 2 FOR      *
C                MODEL 2, ETC                              *
C                                                          *
C      ND     -- NUMBER OF DECISION MAKERS                 *
C                                                          *
C      CW     -- CALCULATED WEIGHT VECTOR                  *
C                                                          *
C      ISIZE  -- INTERMEDIATE STORAGE AREA WHICH IS        *
C                NZR1VR*(2*N-NZR1VR+1)/2 OR AS LARGE AS    *
C                POSSIBLE                                  *
C                                                          *
C      NZR1VR -- NUMBER OF INTEGER VARIABLES               *
C                                                          *
C      SOLMIN -- ESTIMATE OF OBJECTIVE FUNCTION IF KNOWN, *
C                ZERO OTHERWISE                            *
C                                                          *
C      PCTTOL -- TOLERANCE AS FRACTION OF OBJECTIVE        *
C                FUNCTION FOR CONTINUOUS SOLUTION (MAY BE *
C                 LEFT AT ZERO)                            *
C                                                          *
C      M      -- TOTAL NUMBER OF ROWS                      *
C                                                          *
C      N      -- TOTAL NUMBER OF COLUMNS WHICH IS EQUAL TO*
C                THE SUM OF X AND Y VARIABLES PLUS 1 FOR  *
C                RIGHT HANDSIDE                            *
C                                                          *
C      NM1    -- DO LOOP PARAMETERS: NM1 = N - 1           *
C                                                          *
C      UPBND  -- VECTOR OF INTEGER VARIABLE'S UPPER        *
C                BOUNDS; SIZE = N - 1                      *
C                                                          *
```

```
C       IROW    -- VECTOR OF CONSTRAINT TYPES; SIZE = M - 1:*
C                  +1>=BI, 0=BI, -1<=BI                      *
C                                                            *
C       ITEMP   -- COLUMN OF COEFFICIENTS  BEING READ IN     *
C                  ROW I INCLUDING OBJECTIVE ROW             *
C                                                            *
C       VAL     -- COEFFICIENT VALUE OF COLUMNS SPECIFIED BY*
C                  ITEMP FOR ROW I                           *
C                                                            *
C       ATAB    -- INITIAL WORKING TABLEAU, N BY M ARRAY     *
C                                                            *
C*************************************************************
C
C*** THIS COMPUTER PROGRAM DESIGNED FOR RUNNING
C    INTERACTIVELY
C
      DOUBLE PRECISION UPBND(37), TPVAL(31), BTMVL(31),
     *ATAB(34,36), VAL(31), TBSAV(33,36), SAVTAB(34,645),
     *T(36), CC(10,10,10), CW(10 ), ZC(5,5)
      DOUBLE PRECISION SOLMIN, PCTTOL, TLRNCE, YVECT,
     *ATAB11, AMAX, RTIO, ALFA, ARTIO, ADELT, ZOPT, ATAB12,
     *X1, AMAX2, AMAX3, ALW, AUP, RTIO2, DIFF1, DIFF2,
     *DIFF, SVALW, ANDCT4, DABS
      COMMON IROW (33), ITBROW (33), ISVROW (33,30), KSVN
     *(31), ICOL (36), ITBCOL (36), IVAR (36)
      COMMON ISVRCL (30), ICORR (30), ISVN (30)
      NI = 5
      NO = 6
C
C*** PROMPT THE MAIN MENU
C
  731 WRITE(NO,10)
   10 FORMAT(1H1,12X,24(1H*),/,13X,'***    MAIN  MENU
     ****',/,13X,
     *24(1H*),/,/,5X,'1. INPUT PAIRWISE COMPARISONS,',/,
     *             5X,'2. EXIT THE PROGRAM.',/,/,
     *             5X,'==> ENTER THE OPTION NUMBER|')
C
      READ(NI,*) MENU
      GO TO (60,730) MENU
      WRITE(NO,11)
   11 FORMAT(/,5X,'??? ENTERED NUMBER ERROR ???  TRY IT
     *AGAIN|')
      GO TO 731
   60 WRITE(NO,12)
   12 FORMAT(1H1,/,5X,'==> ENTER THE NUMBER OF DECISION
     *MAKERS|')
      READ(NI,*) ND
      WRITE(NO,13) ND 13 FORMAT(/,/,5X,'YOU HAVE ',I2,'
     *DECISION MAKER(S). IS THIS NUMBER CORRECT?',/,
     *            5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
```

```
      IF(INQUR.EQ.2) GO TO 60
  732 WRITE(NO,14)
   14 FORMAT(1H1,/,5X,'==> ENTER THE NUMBER OF CRITERIA|')
      READ(NI,*) IC
      WRITE(NO,15) IC 15 FORMAT(/,/,5X,'YOU HAVE ',I2,'
     *CRITERIA. IS THIS NUMBER CORRECT?'
     *         ,/,5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.2) GO TO 732
      DO 733 K = 1, ND
      WRITE(NO,50) K
   50 FORMAT(1H1,/,5X,'*** THIS IS FOR DECISION MAKER',I2,'
     *!')
      DO 16 I = 1, IC
      DO 17 J = 1, IC
      IF(I.EQ.J) GO TO 18
      WRITE(NO,735) I, J
  735 FORMAT(1H1,/,5X,'==> BY HOW MUCH IS
     *CRITERIA',I2,'MORE IMPORTANT THAN CRITERIA',I2,' ?')
      READ(NI,*) AMOUNT
      CC(K,I,J) = AMOUNT
      GO TO 17
   18 CC(K,I,J) = 1.0
   17 CONTINUE
   16 CONTINUE
C
C*** ECHO PRINT OUT INPUT DATA
C
   28 WRITE(NO,20) K
   20 FORMAT(1H1,5X,39(1H*),/,5X,'*** VALUES RECEIVED FROM
     *DECISION MAKER',I2,' ***',/,5X,39(1H*),/,/)
      DO 21 I = 1, IC
   21 WRITE(NO,*) (CC(K,I,J),J=1,IC)
      WRITE(NO,22)
   22 FORMAT(/,/,5X,'*** ARE THESE DATA CORRECT? ***',
     *          /,5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.1) GO TO 733
   27 WRITE(NO,55)
   55 FORMAT(/,5X,'==> ENTER DECISION MAKER INDEX|')
      READ(NI,*) K1
      WRITE(NO,23)
   23 FORMAT(/,5X,'==> ENTER ROW INDEX NUMBER|')
      READ(NI,*) I
      WRITE(NO,24)
   24 FORMAT(/,5X,'==> ENTER COLUMN INDEX NUMBER|')
      READ(NI,*) J
      WRITE(NO,25) 25 FORMAT(/,5X,'==> ENTER CORRECTED
     *VALUE OF RELATIVE IMPORTANCE|')
      READ(NI,*)CC(K1,I,J)
      WRITE(NO,26)
   26 FORMAT(/,5X,'*** DO YOU NEED TO CHANGE MORE? ***',
```

```
     *        /,5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.1) GO TO 27
      GO TO 28
  733 CONTINUE
      WRITE(NO,30)
   30 FORMAT(1H1,5X,26(1H*),5X,'*** MODEL AVAILABILITY
     ***',/,5X,26(1H*),/,/,5X,'1. MODEL 1',
     *            /,5X,'2. MODEL 2',
     *            /,5X,'3. MODEL 3',
     *            /,5X,'4. MODEL 4',
     *            /,5X,'5. MODEL 5',
     *            /,5X,'6. MODEL 6',
     *            /,5X,'==> ENTER THE MODEL NUMBER|')
      READ(NI,*) IMODEL
   40 X1 = 1.0
      GO TO (701,702,702,750,751,752), IMODEL
  701 IF(ND.GE.2) GO TO 703
      DO 704 I = 1, IC
      DO 705 J = 1, IC
      IF(I.EQ.J) GO TO 706
      C(I,J) = CC(1,I,J)
      GO TO 705
  706 C(I,J) = 1.0
  705 CONTINUE
  704 CONTINUE
      GO TO 707
  703 DO 708 I = 1, IC
      DO 709 J = 1, IC
      IF(I.EQ.J) GO TO 710
      S1 = 0.0
      DO 711 K = 1, ND
      S1 = S1 + CC(K,I,J)
  711 CONTINUE
      C(I,J) = S1 / FLOAT(ND)
      GO TO 709
  710 C(I,J) = 1.0
  709 CONTINUE
  708 CONTINUE
C***
C     INPUT PARAMETERS  M = TOTAL NO. OF ROWS, N = TOTAL
C     NO. OF COLS.  NZR1VR = NO. OF INTEGER VARIABLES
  707 M = IC+2
      N = 2*IC+1
      NZR1VR = IC
C***
C     READ MATRIX ELEMENTS
      DO 8903 I = 1, M
      DO 8903 J = 1, N
 8903 ATAB(I,J) = 0.0
      DO 8010 J = 2, IC+1
 8010 ATAB(1,J) = 1.0
```

```
      J = 2
      DO 8011 I = 2, IC+1
      ATAB(I,J) = 1.0
 8011 J = J + 1
      DO 8020 I = 2, IC+1
      DO 8021 J = NZR1VR+2, N
      IF((I-1).EQ.(J-IC-1)) GO TO 8022
      ATAB(I,J) = C(I-1,J-IC-1)
      GO TO 8021
 8022 ATAB(I,J) = 1.0-FLOAT(IC)
 8021 CONTINUE
 8020 CONTINUE
      DO 8030 J = NZR1VR+2, N
 8030 ATAB(M,J) = 1.0
      ATAB(M,1) = 1.0
      GO TO 712
  702 IN=0
      IF(ND.GE.2) GO TO 713
      DO 714 I = 1, IC
      DO 715 J = 1, IC
      IF(I.EQ.J) GO TO 716
      C(I,J) = CC(1,I,J)
      GO TO 715
  716 C(I,J) = 1.0
  715 CONTINUE
  714 CONTINUE
      GO TO 717
  713 DO 718 I = 1, IC
      DO 719 J = 1, IC
      IF(I.EQ.J) GO TO 720
      S1 = 0.0
      DO 721 K = 1, ND
      S1 = S1 + CC(K,I,J)
  721 CONTINUE
      C(I,J) = S1 / FLOAT(ND)
      GO TO 719
  720 C(I,J) = 1.0
  719 CONTINUE
  718 CONTINUE
C***
C     INPUT PARAMETERS  M = TOTAL NO. OF ROWS, N = TOTAL
C     NO. OF COLS.  NZR1VR = NO. OF INTEGER VARIABLES
  717 DO 8044 I = 1, IC-1
 8044 IN = IN + I
      M = 3*IN+3
      N = 3*IN+IC+1
      NZR1VR = 3*IN
C***
C     READ MATRIX ELEMENTS
      DO 722 I = 1, M
      DO 722 J = 1, N
  722 ATAB(I,J) = 0.0
```

```
      I =1
      I2 = 1
      DO 8444 J = 2, NZR1VR+1
 8444 ATAB(I,J) = 1.0
      I = I + 1
      DO 723 II = 1, IC-1
      DO 724 J = II+1, IC
      IF(C(II,J).GE.1.0) GO TO 8023
      DO 8024 K = NZR1VR+2, N
      ATAB(I,K) = C(J,I2) - C(II,I2)
      I2 = I2 + 1
 8024 CONTINUE
      I = I + 1
      I2 = 1
      IF(I.GT.(IN+1)) GO TO 8027
      GO TO 8022
 8023 DO 8026 K = NZR1VR+2, N
      ATAB(I,K) = C(II,I2) - C(J,I2)
      I2 = I2 + 1
 8026 CONTINUE
      I = I + 1
      I2 = 1
      IF(I.GT.(IN+1)) GO TO 8027
  724 CONTINUE
  723 CONTINUE
 8027 DO 8028 II = NZR1VR+1, NZR1VR+IC-1
      DO 8029 J = II+1, NZR1VR+IC
      IF(C(II-NZR1VR,J-NZR1VR).GE.1.0) GO TO 725
      ATAB(I,II+I2) = -1.0
      ATAB(I,I2+J) = C(II-NZR1VR,J-NZR1VR)
      GO TO 726
  725 ATAB(I,II+I2) = 1.0
      ATAB(I,I2+J) = -C(II-NZR1VR,J-NZR1VR)
  726 IF(C(J-NZR1VR,II-NZR1VR).GE.1.0) GO TO 727
      ATAB(I+IN,II+I2) = C(J-NZR1VR,II-NZR1VR)
      ATAB(I+IN,I2+J) = -1.0
      GO TO 728
  727 ATAB(I+IN,II+I2) = -C(J-NZR1VR,II-NZR1VR)
      ATAB(I+IN,I2+J) = 1.0
  728 I = I + 1
 8029 CONTINUE
 8028 CONTINUE
      DO 729 J = NZR1VR+2, N
  729 ATAB(M,J) = 1.0
      I2 = 2
      DO 8031 J = 2, NZR1VR+1
      ATAB(I2,J) = 1.0
 8031 I2 = I2 + 1
      ATAB(M,1) = 1.0
C     INITIALIZATION
  712 ISIZE = 645
      INDCT7=1
```

```
      KSVN(1)=1
      INDCTR=1
      ICNTR=0
      IOUT1 = 0
      I1ROW=1000
      ADELT = 5.0E-7
C***
C      READ AND WRITE PROBLEM IDENTIFICATION: PUT 1 IN COL.
C      1
C***
C     IOUT2 = INITIAL WORKING TABLEAU
C     IOUT3=CONTINUOUS SOLUTION TABLEAU
      IOUT2 = 1
      IOUT3 = 1
      IPACK = 0
C***
C     SOLMIN=UPPER BOUND ON OBJ. FUNCTION FOR INTEGER
C            SOLUTION
C     PCTTOL=INPUT TOLERANCE AS FRACTION OF OBJECTIVE
C            FUNCT.  FOR CONT. SOLUTION SET EACH ZERO FOR
C            UNKNOWN PROBLEM.
      SOLMIN = 0.0
      PCTTOL = 0.0
   73 DO 72 I=1,N
   72 T(I)=0.
      NM1=N-1
   74 IF(SOLMIN)786,787,786
C***
C     INPUT UPPER BOUND ON OBJECTIVE FUNCTION
  786 TLRNCE=SOLMIN
      PCTTOL=-1.
      GO TO 90
  787 ITOL=1
      SOLMIN = 1E35
      IF(PCTTOL)90,788,90
  788 PCTTOL=.1
C***
C     INPUT UPPER BOUNDS ON VARIABLES (ZERO MEANS NO UPPER
C     BOUND)
   90 IF(IMODEL.EQ.1) GO TO 901
      DO 8015 I = 1, NZR1VR
      UPBND(I) = 1.0
 8015 CONTINUE
      DO 8016 I = NZR1VR+1, NM1
      UPBND(I) = 0.0
 8016 CONTINUE
      GO TO 1
  901 DO 903 I = 1, NM1
  903 UPBND(I) = 0.0
    1 IROW(1)=0
      IROW(M)=0
C**    CONSTRAINT TYPES: ( +1, = 0, ' -1 )
```

```
      DO 8017 I = 2, M-1
 8017 IROW(I) = +1
C**    MATRIX FORMAT:  PACKED = 1,  UNPACKED = 0  .
      IF ( M .LT. 2) GO TO 450
C***
C     PRINT INPUT TABLEAU FOR ERROR CHECK
 9520 DO 954 I=2,M
      IF(IROW(I))953,9521,9521
 9521 DO 9523 J=2,N
 9523 ATAB(I,J)=-ATAB(I,J)
      GO TO 954
  953 ATAB(I,1)=-ATAB(I,1)
  954 CONTINUE
  450 CONTINUE
  955 DO 98 I=2,N
      IF(UPBND(I-1))96,96,98
   96 UPBND (I-1) = 1E3
   98 CONTINUE
C***
C     COMPUTE NO. OF Y VECTORS
  981 YVECT=UPBND(1)+1.
      IF ( NZR1VR .LT. 2) GO TO 322
      DO 982 I=2,NZR1VR
  982 YVECT=YVECT*(UPBND(I)+1.)
  322 CONTINUE
C***
C     SET SOLUTION VECTOR OF VARIABLES EQUAL TO ZERO
C     AND SAVE ORIGINAL UPPER BOUNDS
  985 DO 99 I=2,N
   99 IVAR(I-1)=0
C***
C     INITIALIZE ROW AND COLUMN IDENTIFIERS,+K=VARIABLE NO.
C     K, ZERO = ZERO SLACK, -K = POSITIVE SLACK
      IF ( M .LT. 2) GO TO 451
      DO 102 I=2,M
      IF(IROW(I))100,102,100
  100 IROW(I)=1-I
  102 CONTINUE
  451 CONTINUE
      ATAB11=ATAB(1,1)
      ICOL(1) = 0
      DO 103 J=2,N
      IF(ATAB(1,J))1022,1025,1025
 1022 DO 1023 I=1,M
      ATAB(I,1)=ATAB(I,1)+ATAB(I,J)*UPBND(J-1)
 1023 ATAB(I,J)=-ATAB(I,J)
      ICOL(J)=1000+J-1
      GO TO 103
 1025 ICOL(J)=J-1
  103 CONTINUE
      GO TO 254
C***
```

```
C      START DUAL LP
C      CHOOSE PIVOT ROW, MAXIMUM POSITIVE VALUE IN CONSTANT
C      COLUMN
  112 AMAX = 0.0
      IF ( M .LT. 2) GO TO 452
      DO 120 I=2,M
      IF(ATAB(I,1))120,120,115
  115 IF(ATAB(I,1)-AMAX)120,120,117
  117 AMAX=ATAB(I,1)
      IPVR=I
  120 CONTINUE
  452 CONTINUE
C***
C      IF NO POSITIVE VALUE, LP FINISHED (PRIMAL FEASIBLE)
      IF(AMAX)265,265,130
C      CHOOSE PIVOT COLUMN, ALGEBRAICALLY MAXIMUM RATIO
C      A(1,J)/A(PIVOTROW FOR A (PIVOTROW,J) NEGATIVE. IF NO
C      NEGATIVE A(PIVOTROW,J) PROBLEM INFEASIBLE
  130 AMAX = -1E35
      IF(N-2)143,132,132
  132 IPVC=0
      DO 140 J=2,N
      IF(ATAB(IPVR,J))133,140,140
  133 RTIO=ATAB(1,J)/ATAB(IPVR,J)
      IF(RTIO-AMAX)140,137,135
  135 AMAX=RTIO
  136 IPVC=J
      GO TO 140
  137 IF(ATAB(IPVR,J)-ATAB(IPVR,IPVC))136,140,140
  140 CONTINUE
      IF(IPVC)150,143,150
  143 GO TO (145,435,542,610,665),INDCTR
  145 GO TO 999
C***
C      CARRY OUT PIVOT STEP
  150 ALFA=ATAB(IPVR,IPVC)
C**     UPDATE TABLEAU
      DO 180 J=1,N
      IF(ATAB(IPVR,J))152,180,152
  152 IF(J-IPVC)153,180,153
  153 ARTIO=ATAB(IPVR,J)/ALFA
      DO 175 I=1,M
      IF(ATAB(I,IPVC))157,175,157
  157 IF(I-IPVR)160,175,160
  160 ATAB(I,J)=ATAB(I,J)-ARTIO*ATAB(I,IPVC)
      IF(DABS(ATAB(I,J))-ADELT) 165, 165, 175
  165 ATAB(I,J) = 0.0
  175 CONTINUE
  180 CONTINUE
      DO 190 J=1,N
  190 ATAB(IPVR,J)=ATAB(IPVR,J)/ALFA
C***
```

```
C     EXCHANGE ROW AND COLUMN IDENTIFIERS
      ISV=IROW(IPVR)
      IROW(IPVR)=ICOL(IPVC)
C***
C     IF PIVOT ROW WAS ZERO SLACK, SET MODIFIED PIVOT
C     COLUMN ZERO.
  195 DO 196 I=1,M
  196 ATAB(I,IPVC)=ATAB(I,N)
      ICOL(IPVC)=ICOL(N)
      N=N-1
      GO TO 200
  197 DO 198 I=1,M
  198 ATAB(I,IPVC)=-ATAB(I,IPVC)/ALFA
      ICOL(IPVC)=ISV
      ATAB(IPVR,IPVC)=1./ALFA
C***
C     COUNT PIVOTS
  200 ICNTR=ICNTR+1
      IF(IROW(IPVR)+1000)210,205,210
  205 DO 207 J=1,N
  207 ATAB(IPVR,J)=ATAB(M,J)
      IROW(IPVR)=IROW(M)
      M=M-1
  210 IF(IOUT1)240,2505,240
  240 CONTINUE
 2505 GO TO (254,251,252,253,2535),INDCTR
C***
C     IF SEEKING INTEGER SOLUTION, TEST OBJECTIVE FUNCTION
C     AGAINST CURRENT SOLUTION
  251 IF(ATAB(1,1)-TLRNCE)254,435,435
  252 IF(ATAB(1,1)-TLRNCE)254,542,542
  253 IF(ATAB(1,1)-TLRNCE)254,610,610
 2535 IF(ATAB(1,1)-TLRNCE)254,665,665
C***
C     IF CONSTANT COLUMN OF ZERO SLACK ROW IS NEG., REVERSE
C     SIGNS OF ENT
  254 IF ( M .LT. 2) GO TO 453
      DO 260 K = 2, M
      IF(IROW(K))260,255,260
  255 IF(ATAB(K,1))256,260,260
  256 DO 258 L=1,N
  258 ATAB(K,L)=-ATAB(K,L)
  260 CONTINUE
  453 CONTINUE
C     GO TO NEXT PIVOT STEP
      GO TO 112
  265 CONTINUE
C***
C     IF ANY BASIS VARIABLE EXCEEDS ITS UPPER BOUND,
C     COMPLEMENT IT, AND PIVOT ON CORRESPONDING ROW
      IF ( M .LT. 2) GO TO 454
      DO 275 I=2,M
```

```
      IF(IROW(I))275,275,266
  266 J=IROW(I)
      IF(J-1000)268,268,267
  267 J=J-1000
  268 IF(UPBND(J)+ATAB(I,1))269,275,275
  269 IF(ADELT+UPBND(J)+ATAB(I,1))270,274,274
  270 ATAB(I,1)=-ATAB(I,1)-UPBND(J)
      DO 271 K=2,N
  271 ATAB(I,K)=-ATAB(I,K)
      IPVR=I
      IF(J-IROW(I))272,273,272
  272 IROW(I)=J
      GO TO 130
  273 IROW(I)=IROW(I)+1000
      GO TO 130
  274 ATAB(I,1)=-UPBND(J)
  275 CONTINUE
  454 CONTINUE
C***
C     TRUE END OF LINEAR PROGRAMMING
C     SET SOLUTION VECTOR VALUES FOR BASIC VARIABLES
      IF ( M .LT. 2) GO TO 455
      DO 280 I=2,M
      IF(IROW(I))280,280,277
  277 IF(IROW(I)-1000)279,279,278
  278 J=IROW(I)-1000
      T(J)=UPBND(J)+ATAB(I,1)
      GO TO 280
  279 J=IROW(I)
      T(J)=-ATAB(I,1)
  280 CONTINUE
  455 CONTINUE
C***
C     SET SOLUTION VECTOR VALUES FOR NON-BASIC VARIABLES IN
C     COMPLEMENTED
      DO 285 I=2,N
      IF(ICOL(I))285,285,282
  282 IF(ICOL(I)-1000)284,284,283
  283 J=ICOL(I)-1000
      T(J)=UPBND(J)
      GO TO 285
  284 J=ICOL(I)
      T(J)=0.
  285 CONTINUE
      GO TO (286,437,548,615,670),INDCTR
C***
C     FIRST TIME,WRITE CONTINUOUS SOLUTION TABLEAU
C     IF REQUESTED
  286 ZOPT =DABS( ATAB(1,1))
      IF(IMODEL.EQ.3) GO TO 290
      GO TO 999
C***
```

```
C      COMPUTE ABSOLUTE TOLERANCE
  290 ATAB12=ATAB(1,1)
      ATAB11 =DABS (ATAB11 - ATAB(1,1))
      IF(PCTTOL)294,293,292
  292 TLRNCE=PCTTOL*ATAB11+ATAB12
      GO TO 294
  293 TLRNCE = 1E35
  294 CONTINUE
C***
C      DETERMINE WHETHER CONTINUOUS SOLUTION IS MIXED
C      INTEGER SOLUTION
      IF ( M .LT. 2) GO TO 456
  301 DO 310 I=2,M
      IF(IROW(I))310,310,302
  302 IF(IROW(I)-1000)303,303,304
  303 IF(IROW(I)-NZR1VR)305,305,310
  304 IF(IROW(I)-1000-NZR1VR)305,305,310
  305 AJO1 = ATAB(I,1)
      AJO2 = ADELT
      AJO3 = X1
      IF(AMOD(-AJO1,AJO3)-AJO2) 310,310,306
  306 IF(1.0-AMOD(-AJO1,AJO3)-AJO2) 310,310,295
  310 CONTINUE
  456 CONTINUE
      GO TO 999
C***
C DETERMINE WHETHER PROBLEM FITS IN MEMORY , AND IF SO
C WHETHER TO SAVE ALL INTERMEDIATE TABLEAUS OR ONLY SOME
  295 IF(N-NZR1VR)297,297,298
  297 ISVLOC=(N*(N+1))/2
      GO TO 299
  298 ISVLOC=(NZR1VR*(2*N-NZR1VR+1))/2
  299 IF(ISIZE-ISVLOC)3001,3001,300
  300 I1ROW=0
      GO TO 315
 3001 NONBSC=0
      DO 3006 J=2,N
      IF(ICOL(J))3006,3006,3002
 3002 IF(ICOL(J)-1000)3003,3004,3004
 3003 IF(ICOL(J)-NZR1VR)3005,3005,3006
 3004 IF(ICOL(J)-1000-NZR1VR)3005,3005,3006
 3005 NONBSC=NONBSC+1
 3006 CONTINUE
      IF(N-NZR1VR)3007,3007,3008
 3007 ISVLOC=N+((N-NONBSC)*(N-NONBSC+1))/2
      GO TO 3009
 3008 ISVLOC=N+((NZR1VR-NONBSC)*(N-NONBSC+N-NZR1VR+1))/2
 3009 IF(ISIZE-ISVLOC)3010,3010,315
 3010 GO TO 999
  315 CONTINUE
C***
C      BEGIN INTEGER PROGRAMMING
```

```
  400 I1=1
  402 AMAX = -X1
      KSVN(I1+1)=KSVN(I1)
C***
C     CHOOSE NEXT INTEGER VARIABLE TO BE CONSTRAINED
C     TRY NONBASIC VARIABLES FIRST, CHOOSING ONE WITH
C     LARGEST SHAD PRICE
      DO 4085 I=2,N
      IF(ICOL(I))4085,4085,405
  405 IF(ICOL(I)-1000)406,407,407
  406 IF(ICOL(I)-NZR1VR)408,408,4085
  407 IF(ICOL(I)-1000-NZR1VR)408,408,4085
  408 IF(AMAX-ATAB(1,I))4082,4085,4085
 4082 ISVI=I
      AMAX=ATAB(1,I)
 4085 CONTINUE
C***
C     IF NONE LEFT, TRY BASIC VARIABLES
      IF ( AMAX + X1) 4087, 420, 4087
C***
C     VARIABLE CHOSEN
 4087 IVAR(I1)=ICOL(ISVI)
      BTMVL(I1)=-1.
      ISVRCL(I1)=ISVI
      ICORR(I1)=0
      VAL (I1) = 0.0
C***
C     IF OBJECTIVE FUNCTION VALUE + SHADOW PRICE EXCEEDS
C     TOLERANCE, INDICATE UPWARD DIRECTION INFEASIBLE
      IF(ATAB(1,1)+ATAB(1,ISVI)-TLRNCE)410,409,409
  409 TPVAL(I1)=1000.
      IF(I1-1)4101,4101,4095
 4095 ISVN(I1)=0
      GO TO 4132
  410 TPVAL(I1)=1.
C***
      IF(I1-1)4100,4101,4100
C     SAVE ENTIRE TABLEAU OR ONLY COLUMN CORRESPONDING TO
C     CURRENT NONBASIC VARIABLE, DEPENDING ON SIZE OF PROB
C     AND 2ND DIM OF SAVTAB
 4100 IF(I1-I1ROW)4132,4101,4101
 4101 L=KSVN(I1)
      DO 412 J=1,M
      ISVROW(J,I1)=IROW(J)
      DO 411 K=1,N
      I=L+K-1
      IF(J-1)4105,4105,411
 4105 SAVTAB(M+1,I)=ICOL(K)
  411 SAVTAB(J,I)=ATAB(J,K)
  412 CONTINUE
      ISVN(I1)=N
      KSVN(I1+1)=L+N
```

```
 4132 ICOL(ISVI)=ICOL(N)
      DO 4135 J=1,M
 4135 ATAB(J,ISVI)=ATAB(J,N)
      N=N-1
      GO TO 5000
C     CHOOSE NEXT INTEGER VARIABLE TO BE CONSTRAINED FROM
C     AMONG BASIC VARIABLES IN CURRENT TABLEAU
  420 CONTINUE
      IF(I1-I1ROW)4204,600,4205
 4204 I1ROW=I1
 4205 INDCT7=1
  421 AMAX = -X1
      IF ( M .LT. 2) GO TO 457
      DO 425 I2=2,M
      IF(IROW(I2))425,425,422
  422 IF(IROW(I2)-1000)423,424,424
  423 IF(IROW(I2)-NZR1VR)4241,4241,425
  424 IF(IROW(I2)-1000-NZR1VR)4241,4241,425
 4241 AMAX2 = 1.0E35
      AMAX3 = -1.0E35
      AJO = -ATAB(I2,1) + ADELT
      ALW = AINT(AJO)
      AUP=ALW+1.
      IF(N-1)426,426,4240
 4240 DO 4246 I3=2,N
      IF(ATAB(I2,I3))4244,4246,4242 4242
      RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)4243,4246,4246
 4243 AMAX2=RTIO
      GO TO 4246
 4244 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)4246,4246,4245
 4245 AMAX3=RTIO2
 4246 CONTINUE
      IF ( AMAX3 + 1E34) 430, 430, 4247
 4247 IF (AMAX2 - 1E34) 4248, 429, 429
 4248 DIFF1 =DABS (AMAX2 * (ATAB(I2,1) + ALW))
      DIFF2 =DABS (AMAX3 * (ATAB(I2,1) + AUP))
      DIFF =DABS (DIFF1 - DIFF2)
      IF(DIFF-AMAX)425,425,4249
 4249 AMAX=DIFF
      SVALW=ALW
      ISVI2=I2
      IF(DIFF1-DIFF2)4251,4251,4252
 4251 ANDCT4=0.
      GO TO 425
 4252 ANDCT4=1.
  425 CONTINUE
  457 CONTINUE
      ALW=SVALW
      I2=ISVI2
      VAL(I1)=ALW+ANDCT4
```

```
      BTMVL(I1)=VAL(I1)-1.
 4255 TPVAL(I1)=VAL(I1)+1.
      GO TO 432
C***
C     IF NO. OF COLS=1 AND RIGHT HAND SIDE=0, DONT GO TO LP
  426 IF (DABS( ATAB(I2,1) + ALW) - ADELT) 427, 427, 5100
  427 BTMVL(I1)=-1.
      TPVAL(I1)=1000.
      VAL(I1)=ALW
      IVAR(I1)=IROW(I2)
      IROW(I2)=0
      GO TO 5000
C***
C     CONSTRAINING VARIABLE IN LOWER DIRECTION INFEASIBLE
  429 BTMVL(I1)=-1.
      IF (DABS ( ATAB(I2,1) + ALW) - ADELT ) 4295, 4295,
     *                                       4296
 4295 ANDCT4=0.
      VAL(I1)=ALW+ANDCT4
      GO TO 4255
 4296 TPVAL(I1)=ALW+2.
      ANDCT4=1.
      GO TO 431
C***
C     CONSTRAINING VARIABLE IN UPPER DIRECTION INFEASIBLE
  430 TPVAL(I1)=1000.
      BTMVL(I1)=ALW-1.
      ANDCT4=0.
  431 VAL(I1)=ALW+ANDCT4
C***
C     SAVE ENTIRE TABLEAU
  432 JSVN=N
      L=KSVN(I1)
  438 DO 439 I3=1,M
      ISVROW(I3,I1)=IROW(I3)
      DO 439 I4=1,N
      I6=L+I4-1
      IF(I3-1)4385,4385,439
 4385 SAVTAB(M+1,I6)=ICOL(I4)
  439 SAVTAB(I3,I6)=ATAB(I3,I4)
      ISVN(I1)=N
      KSVN(I1+1)=L+N
      ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      ISVRCL(I1)=I2
      IVAR(I1)=IROW(I2)
      ICORR(I1)=1
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 433, 433, 434
  433 ATAB (I2,1) = 0.0
  434 INDCTR=2
C***
C     RETURN TO CARRY OUT LP
```

```
      IF(IOUT1)240,254,240
C     INFINITE RETURN
  435 IF(ANDCT4)4355,4352,4355
 4352 BTMVL(I1)=-1.
      GO TO 5120
 4355 TPVAL(I1)=1000.
      GO TO 5120
C***
C     FINITE RETURN
  437 GO TO 5000
C     TEST FOR ANY INTEGER VARIABLES LEFT TO BE CONSTRAINED
 5000 IF(I1-NZR1VR)5050,550,550
C     INCREMENT POINTER AND RETURN TO CONSTRAIN NEXT
C     INTEGER VARIABLE
 5050 I1=I1+1
      IF(IOUT1)5051,402,5051
 5051 GO TO 402
C***
C     DECREMENT POINTER AND CONSTRAIN CURRENT VARIABLE TO
C     CURRENT VALUE + OR - 1
 5100 I1=I1-1
 5115 IF(I1)995,995,5120
 5120 IF(IVAR(I1)-1000)5151,5151,5152
 5151 K=IVAR(I1)
      GO TO 5153
 5152 K=IVAR(I1)-1000
 5153 I2=ISVRCL(I1) 5155 IF(BTMVL(I1))516,517,517
  516 IF(TPVAL(I1)-UPBND(K))518,518,5100
  517 IF(TPVAL(I1)-UPBND(K))530,530,525
C***
C     TOP END FEASIBLE
  518 INDCT5=1
 5181 IF(ICORR(I1))5198,5182,5198
 5182 IF(I1-I1ROW)5183,5198,5198
 5183 INDCT8=1
      IF(I1-1)5185,5198,5185
 5185 INDCT5=4
      ISVI1=I1-1
      I1=1
      GO TO 5198
 5190 DO 5194 I3=1,ISVI1
      I4=ISVRCL(I3)
      ICOL(I4)=ICOL(N)
      DO 5193 J=1,M
      IF(VAL(I3)-1.)5193,5191,5192
 5191 ATAB(J,1)=ATAB(J,1)+ATAB(J,I4)
      GO TO 5196
 5192 ATAB(J,1)=ATAB(J,1)+VAL(I3)*ATAB(J,I4)
 5196 INDCT8=2
 5193 ATAB(J,I4)=ATAB(J,N)
      N=N-1
 5194 CONTINUE
```

```
 5195 I1=ISVI1+1
      INDCT5=1
      GO TO 521
C***
C     RETRIEVE SAVED TABLEAU
 5198 N=ISVN(I1)
      L=KSVN(I1)
      DO 5199 I3=1,M
      IROW(I3)=ISVROW(I3,I1)
      DO 5199 I4=1,N
      I6=L+I4-1
      IF(I3-1)5197,5197,5199
 5197 ICOL(I4)=SAVTAB(M+1,I6)
 5199 ATAB(I3,I4)=SAVTAB(I3,I6)
 5205 GO TO (521,526,531,5190),INDCT5
  521 VAL(I1)=TPVAL(I1)
      TPVAL(I1)=TPVAL(I1)+1.
      IF(ICORR(I1))541,522,541
  522 DO 523 I3=1,M
      ATAB(I3,1)=ATAB(I3,1)+(VAL(I1)*ATAB(I3,I2))
      IF (DABS ( ATAB(I3,1)) - ADELT) 5225, 5225, 523
 5225 ATAB(I3,1)=0.
  523 ATAB(I3,I2)=ATAB(I3,N)
      ICOL(I2)=ICOL(N)
      N=N-1
      IF(ATAB(1,1)-TLRNCE)5235,5100,5100
 5235 IF(I1-I1ROW)650,5415,5415
C***
C     BOTTOM END FEASIBLE
  525 INDCT5=2
      GO TO 5198
  526 VAL(I1)=BTMVL(I1)
      BTMVL(I1)=BTMVL(I1)-1.
      GO TO 541
C***
C     BOTH ENDS FEASIBLE
  530 INDCT5=3
      GO TO 5198
  531 AMAX2 = 1.0E35
      AMAX3 = -1.0E35
      DO 536 I3=2,N
      IF(ATAB(I2,I3))534,536,532
  532 RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)533,536,536
  533 AMAX2=RTIO
      GO TO 536
  534 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)536,536,535
  535 AMAX3=RTIO2
  536 CONTINUE
      IF(AMAX2-1.E35)538,537,537
C***
```

```
C      BOTTOM END INFEASIBLE
  537 BTMVL(I1)=-1.
      GO TO 521
  538 IF(AMAX3+1.E35)539,539,540
C***
C      TOP END INFEASIBLE
  539 TPVAL(I1)=1000.
      GO TO 526
  540 DIFF1 =DABS ( AMAX2 * (ATAB(I2,1) + BTMVL (I1)))
      DIFF2 =DABS ( AMAX3 * (ATAB(I2,1) + TPVAL (I1)))
      IF(DIFF1-DIFF2)526,526,521
  541 ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 5412, 5412, 5415
 5412 ATAB(I2,1)=0.
 5415 INDCTR=3
      IF(IOUT1)240,2505,240
C***
C      INFINITE RETURN
  542 GO TO (544,547,543),INDCT5
  543 IF(TPVAL(I1)-VAL(I1)-1.)545,544,545
  544 TPVAL(I1)=1000.
      GO TO 5120
  545 IF(VAL(I1)-BTMVL(I1)-1.)546,547,546
C***
  546 CONTINUE
  547 BTMVL(I1)=-1.
      GO TO 5120
C***
C      FINITE RETURN
  548 GO TO 5000
C      FEASIBLE INTEGER SOLUTION OBTAINED
  550 TLRNCE=ATAB(1,1)
      SOLMIN=1.
C***
C      WRITE CURRENT BEST MIXED INTEGER SOLUTION
      ZOPT =DABS( ATAB( 1,1))
      DO 560 I = 1, NZR1VR
      IF(IVAR(I))554,560,554
  554 IF(IVAR(I)-1000)555,555,557
  555 J=IVAR(I)
      T(J)=VAL(I)
      GO TO 560
  557 J=IVAR(I)-1000
      T(J)=UPBND(J)-VAL(I)
  560 CONTINUE
      GO TO 5115
  600 GO TO (605,4205),INDCT7
  605 INDCTR=4
      IF(IOUT1)240,254,240
C***
C      INFINITE RETURN
```

```
  610 GO TO 5100
C***
C     FINITE RETURN
  615 INDCT7=2
C***
C     IF USING SECOND SOLUTION METHOD, SAVE TABLEAU
C     MODIFIED FOR NONZERO VALUE OF NONBASIC VARIABLE IN
C     TBSAV
  650 DO 655 I=1,M
      ITBROW(I)=IROW(I)
      DO 655 J=1,N
  655 TBSAV(I,J)=ATAB(I,J)
      DO 660 J=1,N
  660 ITBCOL(J)=ICOL(J)
      JSVN=N
      INDCTR=5
      IF(IOUT1)240,254,240
C***
C     INFINITE RETURN
  665 GO TO (544,5120),INDCT8
C     FINITE RETURN
C***
C     IF USING SECOND SOLUTION METHOD, RETRIEVE MODIFIED
C     TABLEAU FROM TBSAV, AS THIS CORRESPONDS TO SAVED
C     COLUMNS FOR I1 LESS THAN I1ROW
  670 N=JSVN
      DO 675 I=1,M
      IROW(I)=ITBROW(I)
      DO 675 J=1,N
  675 ATAB(I,J)=TBSAV(I,J)
      DO 680 J=1,N
  680 ICOL(J)=ITBCOL(J)
      GO TO 5000
C***
C     OUTPUT FINAL SOLUTION.
  995 IF(ITOL)996,999,996
  996 IF(SOLMIN-1.E35)999,997,997
  997 ITOL=ITOL+1
      TLRNCE=FLOAT(ITOL)*PCTTOL*ATAB11+ATAB12
      N=ISVN(1)
      DO 9972 I=1,M
      IROW(I)=ISVROW(I,1)
      DO 9972 J=1,N
 9972 ATAB(I,J)=SAVTAB(I,J)
      DO 9973 K=1,N
 9973 ICOL(K)=SAVTAB(M+1,K)
      GO TO 400
  999 DO 19 I = 1, IC
   19 CW(I) = T(NM1-IC+I)
      GO TO 9999
  750 CALL EIGENP(N,NM,A,T,EVR,EVI,VECR,VECI,INDIC,IMAX)
      GO TO 9999
```

```
  751 CALL MODEL5(IC,TW,NMRUNS,C)
      GO TO 9999
  752 CALL MODEL6(IC,TW,NMRUNS,R,C)
C
 9999 WRITE(NO,31)
   31 FORMAT(1H1,5X,25(1H*),/,5X,'*** ESTIMATED WEIGHTS
     ***', */,5X,25(1H*),/,/,/)
      WRITE(NO,736) (CW(I),I=1,IC)
  736 FORMAT(2X,5F12.6)
      WRITE(NO,32)
   32 FORMAT(/,5X,'*** DO YOU WANT TO GO BACK TO MAIN MENU?
     ****',/,5X,'==> ENTER 1=YES, 2=NO <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.2) GO TO 730
      WRITE(NO,33)
   33 FORMAT(1H1,/,5X,'*** VALUES USED ARE AS FOLLOWS:',/)
      DO 734 K = 1, ND
      WRITE(NO,51) K
   51 FORMAT(/,5X,'FOR ',I2,'TH DECISION MAKER',/)
      DO 34 I = 1, IC
   34 WRITE(NO,*) (CC(K,I,J),J=1,IC)
  734 CONTINUE
      WRITE(NO,35)
   35 FORMAT(/,/,5X,'*** FOR SENSITIVITY ANALYSIS OR
     *RELECTING THE CHANGES OF MIND OF DECISION MAKER ***')
   41 WRITE(NO,52)
   52 FORMAT(/,5X,'==> ENTER DECISION MAKER INDEX|')
      READ(NI,*) K1
      WRITE(NO,36)
   36 FORMAT(/,5X,'==> ENTER ROW INDEX NUMBER|')
      READ(NI,*) I
      WRITE(NO,37)
   37 FORMAT(/,5X,'==> ENTER COLUMN INDEX NUMBER|')
      READ(NI,*) J
      WRITE(NO,38) 38 FORMAT(/,5X,'==> ENTER CORRECTED
     *VALUE OF RELATIVE IMPORTANCE|')
      READ(NI,*)CC(K1,I,J)
      WRITE(NO,39)
   39 FORMAT(/,5X,'*** DO YOU NEED TO CHANGE MORE? ***',
     *        /,5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.1) GO TO 41
      WRITE(NO,42) K1
   42 FORMAT(1H1,/,5X,'*** VALUES CHANGED FROM ',I2,'TH
     *DECISION MAKER ARE AS FOLLOWS:',/)
      DO 43 I = 1, IC
   43 WRITE(NO,*) (CC(K1,I,J),J=1,IC)
      WRITE(NO,44)
   44 FORMAT(/,/,5X,'*** ARE THESE DATA CORRECT? ***',
     *          /,5X,'==> ENTER 1=YES, 2=NO. <<<')
      READ(NI,*) INQUR
      IF(INQUR.EQ.1) GO TO 40
```

```
      GO TO 41
  730 STOP
      END
C
C**********************************************************
C
      SUBROUTINE EIGENP(N, NM, A, T, EVR, EVI, VECR, VECI,
     *                  INDIC, IMAX)
C
C**********************************************************
C
      DOUBLE PRECISION D1,D2,D3,PRFACT
      INTEGER I,IVEC,J,K,K1,KON,L,L1,M,N,NM,IMAX
      REAL ENORM,EPS,EX,R,R1,T DIMENSION A(NM,1),
     *VECR(NM,1),VECI(NM,1), EVR(NM),EVI(NM), INDIC(NM)
      DIMENSION IWORK(100), LOCAL(100), PRFACT(100),
     *SUBDIA(100), WORK1(100), WORK2(100),WORK(100)
      IF(N.NE.1)GO TO 1
      EVR(1) = A(1,1)
      EVI(1) = 0.0
      VECR(1,1) = 1.0
      VECI(1,1) = 0.0
      INDIC(1) = 2
      GO TO 25
C
    1 CALL SCALE(N,NM,A,VECI,PRFACT,ENORM)
C
C THE COMPUTATION OF THE EIGENVALUES OF THE NORMALIZED
C MATRIX
      EX = EXP(-T*ALOG(2.0))
C
      CALL HESQR(N, NM, A, VECI, EVR, EVI, SUBDIA, INDIC,
     *           EPS, EX, IMAX)
C
      J = N
      I = 1
      LOCAL(1) = 1
      IF(J.EQ.1)GO TO 4
    2 IF(ABS(SUBDIA(J-1)).GT.EPS)GO TO 3
      I = I + 1
      LOCAL(I) = 0
    3 J = J - 1
      LOCAL(I) = LOCAL(I) + 1
      IF(J.NE.1)GO TO 2
C
C THE EIGENVECTOR PROBLEM
    4 K = 1
      KON = 0
      L = LOCAL(1)
      M = N
      DO 10 I = 1, N
      IVEC = N-I+1
```

```
      IF(I.LE.L)GO TO 5
      K = K+1
      M = N-L
      L = L+LOCAL(K)
    5 IF(INDIC(IVEC).EQ.0)GO TO 10
      IF(EVI(IVEC).NE.0.0)GO TO 8
C
C TRANSFER OF AN UPPER HESSENBERG MATRIX OF THE ORDER M
C FROM THE ARRAYS VECI AND SUBDIA INTO THE ARRAY A.
      DO 7 K1 = 1,M
      DO 6 L1 = K1,M
    6 A(K1,L1) = VECI(K1,L1)
      IF(K1.EQ.1)GO TO 7
      A(K1,K1-1) = SUBDIA(K1-1)
    7 CONTINUE
C
C THE COMPUTATION OF THE REAL EIGENVECTOR IVEC OF THE
C UPPER-HESSENBERG MATRIX CORRESPONDING TO THE REAL
C EIGENVALUE EVR(IVEC)
C
      CALL REALVE(N, NM, M, IVEC, A, VECR, EVR, EVI, IWORK,
     *            WORK, INDIC, EPS, EX)
C
      GO TO 10
C
C THE COMPUTATION OF THE COMPLEX EIGENVECTOR IVEC OF THE
C UPPER HESSENBERG MATRIX CORRESPONDING TO THE COMPLEX
C EIGENVALUE EVR(IVEC)+I*EVI(IVEC).  IF THE VALUE OF KON IS
C NOT EQUAL TO ZERO THEN THIS COMPLEX EIGENVECTOR HAS
C ALREADY BEEN FOUND FROM ITS CONJUGATE.
    8 IF(KON.NE.0)GO TO 9
      KON = 1
C
      CALL COMPVE(N, NM, M, IVEC, A, VECR, VECI, EVR, EVI,
     *            INDIC, IWORK, SUBDIA, WORK1, WORK2, WORK,
     *            EPS, EX)
C
      GO TO 10
    9 KON = 0
   10 CONTINUE
C
      DO 12 I = 1,N
      DO 11 J = I,N
      A(I,J) = 0.0
   11 A(J,I) = 0.0
   12 A(I,I) = 1.0
      IF(N.LE.2)GO TO 15
      M = N-2
      DO 14 K = 1,M
      L = K+1
      DO 14 J= 2,N
      D1 = 0.0
```

```
      DO 13 I = L,N
      D2 = VECI(I,K)
   13 D1 = D1 + D2*A(J,I)
      DO 14 I = L,N
   14 A(J,I) = A(J,I)-VECI(I,K)*D1
C
   15 KON = 1
      DO 24 I = 1,N
      L = 0
      IF(EVI(I).EQ.0.0)GO TO 16
      L = 1
      IF(KON.EQ.0)GO TO 16
      KON = 0
      GO TO 24
   16 DO 18 J  = 1,N
      D1 = 0.0
      D2 = 0.0
      DO 17 K = 1,N
      D3 = A(J,K)
      D1 = D1+D3*VECR(K,I)
      IF(L.EQ.0)GO TO 17
      D2 = D2+D3*VECR(K,I-1)
   17 CONTINUE
      WORK(J) = D1/PRFACT(J)
      IF(L.EQ.0)GO TO 18
      SUBDIA(J) =D2/PRFACT(J)
   18 CONTINUE
C
      IF(L.EQ.1)GO TO 21
      D1 = 0.0
      DO 19 M = 1,N
   19 D1 = D1+WORK(M)**2
      D1 = DSQRT(D1)
      DO 20 M = 1,N
      VECI(M,I) = 0.0
   20 VECR(M,I) = WORK(M)/D1
      EVR(I) = EVR(I)*ENORM
      GO TO 24
C
   21 KON = 1
      EVR(I) = EVR(I)*ENORM
      EVR(I-1) = EVR(I)
      EVI(I) = EVI(I)*ENORM
      EVI(I-1) = -EVI(I)
      R = 0.0
      DO 22 J = 1,N
      R1 = WORK(J)**2 + SUBDIA(J)**2
      IF(R.GE.R1)GO TO 22
      R = R1
      L = J
   22 CONTINUE
      D3 = WORK(L)
```

```
      R1 = SUBDIA(L)
      DO 23 J = 1,N
      D1 = WORK(J)
      D2 = SUBDIA(J)
      VECR(J,I) = (D1*D3+D2*R1)/R
      VECI(J,I) = (D2*D3-D1*R1)/R
      VECR(J,I-1) = VECR(J,I)
   23 VECI(J,I-1) = -VECI(J,I)
   24 CONTINUE
C
   25 RETURN
      END
C
C**************************************************************
C
      SUBROUTINE SCALE(N,NM,A,H,PRFACT,ENORM)
C
C**************************************************************
C
      DOUBLE PRECISION COLUMN,FACTOR,FNORM,PRFACT,Q,ROW
      INTEGER I,J,ITER,N,NCOUNT,NM
      REAL BOUND1,BOUND2,ENORM
      DIMENSION A(NM,1),H(NM,1),PRFACT(NM)
C
      DO 2 I = 1,N
      DO 1 J = 1,N
    1 H(I,J) = A(I,J)
    2 PRFACT(I) = 1.0
      BOUND1 = .75
      BOUND2 = 1.33
      ITER = 0
    3 NCOUNT = 0
      DO 8 I = 1,N
      COLUMN = 0.0
      ROW = 0.0
      DO 4 J = 1,N
      IF(I.EQ.J)GO TO 4
      COLUMN = COLUMN+ABS(A(J,I))
      ROW = ROW+ABS(A(I,J))
    4 CONTINUE
      IF(COLUMN.EQ.0.0)GO TO 5
      IF(ROW.EQ.0.0)GO TO 5
      Q = COLUMN/ROW
      IF(Q.LT.BOUND1)GO TO 6
      IF(Q.GT.BOUND2)GO TO 6
    5 NCOUNT = NCOUNT+1
      GO TO 8
    6 FACTOR = DSQRT(Q)
      DO 7 J = 1,N
      IF(I.EQ.J)GO TO 7
      A(I,J) = A(I,J)*FACTOR
      A(J,I) = A(J,I)/FACTOR
```

```
    7 CONTINUE
      PRFACT(I) = PRFACT(I)*FACTOR
    8 CONTINUE
      ITER = ITER+1
      IF(ITER.GT.30)GO TO 11
      IF(NCOUNT.LT.N)GO TO 3
C
      FNORM = 0.0
      DO 9 I = 1,N
      DO 9 J = 1,N
      Q = A(I,J)
    9 FNORM = FNORM+Q*Q
      FNORM = DSQRT(FNORM)
      DO 10 I = 1,N
      DO 10 J = 1,N
   10 A(I,J) = A(I,J)/FNORM
      ENORM = FNORM
      GO TO 13
C
   11 DO 12 I = 1,N
      PRFACT(I) = 1.0
      DO 12 J = 1,N
   12 A(I,J) = H(I,J)
      ENORM = 1.0
C
   13 RETURN
      END
C
C**********************************************************
C
      SUBROUTINE HESQR(N, NM, A, H, EVR, EVI, SUBDIA,
     *                 INDIC, EPS, EX, IMAX)
C
C**********************************************************
C
      DOUBLE PRECISION S,SR,SR2,X,Y,Z
      INTEGER I,J,K,L,M,MAXST,M1,N,NM,NS,IMAX
      REAL EPS,EX,R,SHIFT,T
      DIMENSION A(NM,1), H(NM,1), EVR(NM), EVI(NM),
     1          SUBDIA(NM), INDIC(NM)
C
      IF(N-2)14,1,2
    1 SUBDIA(1) = A(2,1)
      GO TO 14
    2 M = N-2
      DO 12 K = 1,M
      L = K+1
      S = 0.0
      DO 3 I = L,N
      H(I,K) = w(I,K)
    3 S = S+ABS(A(I,K))
      IF(S.NE.ABS(A(K+1,K)))GO TO 4
```

```
      SUBDIA(K) = A(K+1,K)
      H(K+1,K) = 0.0
      GO TO 12
    4 SR2 = 0.0
      DO 5 I = L,N
      SR = A(I,K)
      SR = SR/S
      A(I,K) = SR
    5 SR2 = SR2+SR*SR
      SR = DSQRT(SR2)
      IF(A(L,K).LT.0.0)GO TO 6
      SR = -SR
    6 SR2 = SR2-SR*A(L,K)
      A(L,K) = A(L,K)-SR
      H(L,K) = H(L,K)-SR*S
      SUBDIA(K) = SR*S
      X = S*DSQRT(SR2)
      DO 7 I = L,N
      H(I,K) = H(I,K)/X
    7 SUBDIA(I) = A(I,K)/SR2
C PREMULTIPLICATION BY THE MATRIX PR.
      DO 9 J = L,N
      SR = 0.0
      DO 8 I = L,N
    8 SR = SR+A(I,K)*A(I,J)
      DO 9 I = L,N
    9 A(I,J) = A(I,J)-SUBDIA(I)*SR
C POSTMULTIPLICATION BY THE MATRIX PR.
      DO 11 J = 1,N
      SR = 0.0
      DO 10 I = L,N
   10 SR = SR+A(J,I)*A(I,K)
      DO 11 I = L,N
   11 A(J,I) = A(J,I)-SUBDIA(I)*SR
   12 CONTINUE
      DO 13 K = 1,M
   13 A(K+1,K) = SUBDIA(K)
C
      SUBDIA(N-1) = A(N,N-1)
   14 EPS = 0.0
      DO 15 K = 1,N
      INDIC(K) = 0
      IF(K.NE.N)EPS = EPS+SUBDIA(K)**2
      DO 15 I = K,N
      H(K,I) = A(K,I)
   15 EPS = EPS+A(K,I)**2
      EPS = EX*SQRT(EPS)
C
      SHIFT = A(N,N-1)
      IF(N.LE.2)SHIFT = 0.0
      IF(A(N,N).NE.0.0)SHIFT = 0.0
      IF(A(N-1,N).NE.0.0)SHIFT = 0.0
```

```
      IF(A(N-1,N-1).NE.0.0)SHIFT = 0.0
      M = N
      NS = 0
      MAXST = N*10
C
      DO 16 I = 2,N
      DO 16 K = I,N
      IF(A(I-1,K).NE.0.0)GO TO 18
   16 CONTINUE
      DO 17 I = 1,N
      INDIC(I) = 1
      EVR(I) = A(I,I)
   17 EVI(I) = 0.0
      GO TO 37
C
   18 K = M-1
      M1 = K
      I = K
C
      IF(K)37,34,19
   19 IF(ABS(A(M,K)).LE.EPS)GO TO 34
      IF(M-2.EQ.0)GO TO 35
   20 I = I-1
      IF(ABS(A(K,I)).LE.EPS)GO TO 21
      K = I
      IF(K.GT.1)GO TO 20
   21 IF(K.EQ.M1)GO TO 35
C
      S = A(M,M)+A(M1,M1)+SHIFT
      SR = A(M,M)*A(M1,M1)-A(M,M1)*A(M1,M)+0.25*SHIFT**2
      A(K+2,K) = 0.0
C
      X = A(K,K)*(A(K,K)-S)+A(K,K+1)*A(K+1,K)+SR
      Y = A(K+1,K)*(A(K,K)+A(K+1,K+1)-S)
      R = DABS(X)+DABS(Y)
      IF(R.EQ.0.0)SHIFT=A(M,M-1)
      IF(R.EQ.0.0)GO TO 21
      Z = A(K+2,K+1)*A(K+1,K)
      SHIFT = 0.0
      NS = NS + 1
C
      DO 33 I = K,M1
      IF(I.EQ.K)GO TO 22
C
      X = A(I,I-1)
      Y = A(I+1,I-1)
      Z = 0.0
      IF(I+2.GT.M)GO TO 22
      Z = A(I+2,I-1)
   22 SR2 = DABS(X)+DABS(Y)+DABS(Z)
      IF(SR2.EQ.0.0)GO TO 23
      X = X/SR2
```

```
      Y = Y/SR2
      Z = Z/SR2
   23 S = DSQRT(X*X+Y*Y+Z*Z)
      IF(X.LT.0.0)GO TO 24
      S = -S
   24 IF(I.EQ.K)GO TO 25
      A(I,I-1) = S*SR2
   25 IF(SR2.NE.0.0)GO TO 26
      IF(I+3.GT.M)GO TO 33
      GO TO 32
   26 SR = 1.0-X/S
      S = X-S
      X = Y/S
      Y = Z/S
C
      DO 28 J = I,M
      S = A(I,J)+A(I+1,J)*X
      IF(I+2.GT.M)GO TO 27
      S = S+A(I+2,J)*Y
   27 S = S*SR
      A(I,J) = A(I,J)-S
      A(I+1,J) = A(I+1,J)-S*X
      IF(I+2.GT.M)GO TO 28
      A(I+2,J) = A(I+2,J)-S*Y
   28 CONTINUE
C
      L = I+2
      IF(I.LT.M1)GO TO 29
      L = M
   29 DO 31 J = K,L
      S = A(J,I)+A(J,I+1)*X
      IF(I+2.GT.M)GO TO 30
      S = S+A(J,I+2)*Y
   30 S = S*SR
      A(J,I) = A(J,I)-S
      A(J,I+1) = A(J,I+1)-S*X
      IF(I+2.GT.M)GO TO 31
      A(J,I+2) = A(J,I+2)-S*Y
   31 CONTINUE
      IF(I+3.GT.M)GO TO 33
      S = -A(I+3,I+2)*Y*SR
   32 A(I+3,I) = S
      A(I+3,I+1) = S*X
      A(I+3,I+2) = S*Y+A(I+3,I+2)
   33 CONTINUE
C
      IF(NS.GT.MAXST)GO TO 37
      GO TO 18
C
   34 EVR(M) = A(M,M)
      EVI(M) = 0.0
      INDIC(M) = 1
```

```
      M = K
      GO TO 18
C
   35 R = 0.5*(A(K,K)+A(M,M))
      S = 0.5*(A(M,M)-A(K,K))
      S = S*S+A(K,M)*A(M,K)
      INDIC(K) = 1
      INDIC(M) = 1
      IF(S.LT.0.0)GO TO 36
      T = DSQRT(S)
      EVR(K) = R-T
      EVR(M) = R+T
      EVI(K) = 0.0
      EVI(M) = 0.0
      M = M-2
      GO TO 18
   36 T = DSQRT(-S)
      EVR(K) = R
      EVI(K) = T
      EVR(M) = R
      EVI(M) = -T
      M = M-2
      GO TO 18
C
   37 TMAX = 0.0
      DO 38 I = 1, N
      IF(EVR(I).LT.TMAX) GO TO 38
      TMAX = EVR(I)
      IMAX = I
   38 CONTINUE
      RETURN
      END
C
C**********************************************************
C
      SUBROUTINE REALVE(N, NM, M, IVEC, A, VECR, EVR, EVI,
     *                  IWORK, WORK, INDIC, EPS, EX)
C
C**********************************************************
C
      DOUBLE PRECISION S,SR
      INTEGER I,IVEC,ITER,J,K,L,M,N,NM,NS
      REAL BOUND,EPS,EVALUE,EX,PREVIS,R,R1,T
      DIMENSION A(NM,1), VECR(NM,1), EVR(NM), EVI(NM),
     1          IWORK(NM), WORK(NM), INDIC(NM)
C
      VECR(1,IVEC) = 1.0
      IF(M.EQ.1)GO TO 24
C
      EVALUE = EVR(IVEC)
      IF(IVEC.EQ.M)GO TO 2
      K = IVEC+1
```

```
      R = 0.0
      DO 1 I = K,M
      IF(EVALUE.NE.EVR(I))GO TO 1
      IF(EVI(I).NE.0.0)GO TO 1
      R = R+3.0
    1 CONTINUE
      EVALUE = EVALUE+R*EX
    2 DO 3 K = 1,M
    3 A(K,K) = A(K,K)-EVALUE
C
      K = M-1
      DO 8 I = 1,K
      L = I+1
      IWORK(I) = 0
      IF(A(I+1,I).NE.0.0)GO TO 4
      IF(A(I,I).NE.0.0)GO TO 8
      A(I,I) = EPS
      GO TO 8
    4 IF(ABS(A(I,I)).GE.ABS(A(I+1,I)))GO TO 6
      IWORK(I) = 1
      DO 5 J = I,M
      R = A(I,J)
      A(I,J) = A(I+1,J)
    5 A(I+1,J)= R
    6 R = -A(I+1,I)/A(I,I)
      A(I+1,I) = R
      DO 7 J = L,M
    7 A(I+1,J)= A(I+1,J)+R*A(I,J)
    8 CONTINUE
      IF(A(M,M).NE.0.0)GO TO 9
      A(M,M) = EPS
C
    9 DO 11 I = 1,N
      IF(I.GT.M)GO TO 10
      WORK(I)= 1.0
      GO TO 11
   10 WORK(I) = 0.0
   11 CONTINUE
C
      BOUND = 0.01/(EX*FLOAT(N))
      NS = 0
      ITER = 1
C
   12 R = 0.0
      DO 15 I = 1,M
      J = M-I+1
      S = WORK(J)
      IF(J.EQ.M)GO TO 14
      L = J+1
      DO 13 K = L,M
      SR = WORK(K)
   13 S= S-SR*A(J,K)
```

```
   14 WORK(J) = S/A(J,J)
      T = ABS(WORK(J))
      IF(R.GE.T)GO TO 15
      R = T
   15 CONTINUE
C
      DO 16 I = 1,M
   16 WORK(I) = WORK(I)/R
C
      R1 = 0.0
      DO 18 I = 1,M
      T = 0.0
C
      DO 17 J = I,M
   17 T = T+A(I,J)*WORK(J)
      T = ABS(T)
      IF(R1.GE.T)GO TO 18
      R1 = T
   18 CONTINUE
      IF(ITER.EQ.1)GO TO 19
      IF(PREVIS.LE.R1)GO TO 24
   19 DO 20 I = 1,M
   20 VECR(I,IVEC) = WORK(I)
      PREVIS = R1
      IF(NS.EQ.1)GO TO 24
      IF(ITER.GT.6)GO TO 25
      ITER = ITER+1
      IF(R.LT.BOUND)GO TO 21
      NS = 1
C
   21 K = M-1
      DO 23 I = 1,K
      R = WORK(I+1)
      IF(IWORK(I).EQ.0)GO TO 22
      WORK(I+1) = WORK(I)+WORK(I+1)*A(I+1,I)
      WORK(I) = R
      GO TO 23
   22 WORK(I+1) = WORK(I+1)+WORK(I)*A(I+1,I)
   23 CONTINUE
      GO TO 12
C
   24 INDIC(IVEC) = 2
   25 IF(M.EQ.N)GO TO 27
      J = M+1
      DO 26 I = J,N
   26 VECR(I,IVEC) = 0.0
   27 RETURN
      END
```

```
C
C***********************************************************
C
      SUBROUTINE COMPVE(N, NM, M, IVEC, A, VECR, H, EVR,
     1                  EVI, INDIC, IWORK, SUBDIA, WORK1,
     2                  WORK2, WORK, EPS, EX)
C
C***********************************************************
C
      DOUBLE PRECISION D,D1
      INTEGER I,I1,I2,ITER,IVEC,J,K,L,M,N,NM,NS
      REAL B,BOUND,EPS,ETA,EX,FKSI,PREVIS,R,S,U,V
      DIMENSION A(NM,1), VECR(NM,1), H(NM,1), EVR(NM),
     1          EVI(NM), INDIC(NM), IWORK(NM), SUBDIA(NM),
     2          WORK1(NM), WORK2(NM), WORK(NM)
C
      FKSI = EVR(IVEC)
      ETA = EVI(IVEC)
C
      IF(IVEC.EQ.M)GO TO 2
      K = IVEC+1
      R = 0.0
      DO 1 I = K,M
      IF(FKSI.NE.EVR(I))GO TO 1
      IF(ABS(ETA).NE.ABS(EVI(I)))GO TO 1
      R = R+3.0
    1 CONTINUE
      R = R*EX
      FKSI = FKSI+R
      ETA = ETA+R
C
    2 R = FKSI*FKSI+ETA*ETA
      S = 2.0*FKSI
      L = M-1
      DO 5 I = 1,M
      DO 4 J = I,M
      D = 0.0
      A(J,I) = 0.0
      DO 3 K = I,J
    3 D = D+H(I,K)*H(K,J)
    4 A(I,J) = D-S*H(I,J)
    5 A(I,I) = A(I,I)+R
      DO 9 I = 1,L
      R = SUBDIA(I)
      A(I+1,I) = -S*R
      I1 = I+1
      DO 6 J = 1,I1
    6 A(J,I) = A(J,I)+R*H(J,I+1)
      IF(I.EQ.1)GO TO 7
      A(I+1,I-1) = R*SUBDIA(I-1)
    7 DO 8 J = I,M
    8 A(I+1,J) = A(I+1,J)+R*H(I,J)
```

```
    9 CONTINUE
C
      K = M-1
      DO 18 I = 1,K
      I1 = I+1
      I2 = I+2
      IWORK(I) = 0
      IF(I.EQ.K)GO TO 10
      IF(A(I+2,I).NE.0.0)GO TO 11
   10 IF(A(I+1,I).NE.0.0)GO TO 11
      IF(A(I,I).NE.0.0)GO TO 18
      A(I,I) = EPS
      GO TO 18
C
   11 IF(I.EQ.K)GO TO 12
      IF(ABS(A(I+1,I)).GE.ABS(A(I+2,I)))GO TO 12
      IF(ABS(A(I,I)).GE.ABS(A(I+2,I)))GO TO 16
      L = I+2
      IWORK(I) = 2
      GO TO 13
   12 IF(ABS(A(I,I)).GE.ABS(A(I+1,I)))GO TO 15
      L = I+1
      IWORK(I) = 1
C
   13 DO 14 J = I,M
      R = A(I,J)
      A(I,J) = A(L,J)
   14 A(L,J) = R
   15 IF(I.NE.K)GO TO 16
      I2 = I1
   16 DO 17 L = I1,I2
      R = -A(L,I)/A(I,I)
      A(L,I) = R
      DO 17 J = I1,M
   17 A(L,J) = A(L,J)+R*A(I,J)
   18 CONTINUE
      IF(A(M,M).NE.0.0)GO TO 19
      A(M,M) = EPS
C
   19 DO 21 I = 1,N
      IF(I.GT.M)GO TO 20
      VECR(I,IVEC) = 1.0
      VECR(I,IVEC-1) = 1.0
      GO TO 21
   20 VECR(I,IVEC) = 0.0
      VECR(I,IVEC-1) = 0.0
   21 CONTINUE
C
      BOUND = 0.01/(EX*FLOAT(N))
      NS = 0
      ITER = 1
      DO 22 I = 1,M
```

```
   22 WORK(I) = H(I,I)-FKSI
C
   23 DO 27 I = 1,M
      D = WORK(I)*VECR(I,IVEC)
      IF(I.EQ.1)GO TO 24
      D = D+SUBDIA(I-1)*VECR(I-1,IVEC)
   24 L = I+1
      IF(L.GT.M)GO TO 26
      DO 25 K = L,M
   25 D = D+H(I,K)*VECR(K,IVEC)
   26 VECR(I,IVEC-1) = D-ETA*VECR(I,IVEC-1)
   27 CONTINUE
C
      K = M-1
      DO 28 I = 1,K
      L = I+IWORK(I)
      R = VECR(L,IVEC-1)
      VECR(L,IVEC-1) = VECR(I,IVEC-1)
      VECR(I,IVEC-1) = R
      VECR(I+1,IVEC-1) = VECR(I+1,IVEC-1)+A(I+1,I)*R
      IF(I.EQ.K)GO TO 28
      VECR(I+2,IVEC-1) = VECR(I+2,IVEC-1)+A(I+2,I)*R
   28 CONTINUE
C
      DO 31 I = 1,M
      J = M-I+1
      D = VECR(J,IVEC-1)
      IF(J.EQ.M)GO TO 30
      L = J+1
      DO 29 K = L,M
      D1 = A(J,K)
   29 D = D-D1*VECR(K,IVEC-1)
   30 VECR(J,IVEC-1) = D/A(J,J)
   31 CONTINUE
C
      DO 35 I = 1,M
      D = WORK(I)*VECR(I,IVEC-1)
      IF(I.EQ.1)GO TO 32
      D = D+SUBDIA(I-1)*VECR(I-1,IVEC-1)
   32 L = I+1
      IF(L.GT.M)GO TO 34
      DO 33 K = L,M
   33 D = D+H(I,K)*VECR(K,IVEC-1)
   34 VECR(I,IVEC) = (VECR(I,IVEC)-D)/ETA
   35 CONTINUE
C
      L = 1
      S = 0.0
      DO 36 I = 1,M
      R = VECR(I,IVEC)**2+VECR(I,IVEC-1)**2
      IF(R.LE.S)GO TO 36
      S = R
```

```
      L = I
   36 CONTINUE
C
      U = VECR(L,IVEC-1)
      V = VECR(L,IVEC)
      DO 37 I = 1,M
      B = VECR(I,IVEC)
      R = VECR(I,IVEC-1)
      VECR(I,IVEC) = (R*U+B*V)/S
   37 VECR(I,IVEC-1) = (B*U-R*V)/S
C
      B = 0.0
      DO 41 I = 1,M
      R = WORK(I)*VECR(I,IVEC-1)-ETA*VECR(I,IVEC)
      U = WORK(I)*VECR(I,IVEC)+ETA*VECR(I,IVEC-1)
      IF(I.EQ.1)GO TO 38
      R = R+SUBDIA(I-1)*VECR(I-1,IVEC-1)
      U = U+SUBDIA(I-1)*VECR(I-1,IVEC)
   38 L = I+1
      IF(L.GT.M)GO TO 40
      DO 39 J = L,M
      R = R+H(I,J)*VECR(J,IVEC-1)
   39 U = U+H(I,J)*VECR(J,IVEC)
   40 U = R*R+U*U
      IF(B.GE.U)GO TO 41
      B = U
   41 CONTINUE
      IF(ITER.EQ.1)GO TO 42
      IF(PREVIS.LE.B)GO TO 44
   42 DO 43 I = 1,N
      WORK1(I) = VECR(I,IVEC)
   43 WORK2(I) = VECR(I,IVEC-1)
      PREVIS = B
      IF(NS.EQ.1)GO TO 46
      IF(ITER.GT.6)GO TO 47
      ITER = ITER+1
      IF(BOUND.GT.SQRT(S))GO TO 23
      NS = 1
      GO TO 23
C
   44 DO 45 I = 1,N
      VECR(I,IVEC) = WORK1(I)
   45 VECR(I,IVEC-1) = WORK2(I)
   46 INDIC(IVEC-1) = 2
      INDIC(IVEC) = 2
   47 RETURN
      END
```

```
C
C**********************************************************
C
      SUBROUTINE MODEL5(IC, TW, NMRUNS, C)
C
C**********************************************************
C
      INTEGER ITEMP, ITEMP1, ITEMP2, IC, II(5), IT, NMRUNS,
      REAL TN, TN2, TW(5), W(5), W1(5), W2(5), W3(5),
     1     CYW(5), C(5,5), TT
      DO 100 I = 1, IC
      II(I) = I
      W(I) = 0.0
      W1(I) = 0.0
      W2(I) = 1.0
      W3(I) = 0.0
  100 CONTINUE
C
C     CALCULATE THE WEIGHT VECTORS AFTER GENERATING ALL
C     POSSIBLE INDEX ORDERS
C
      IF(IC.EQ.5) GO TO 300
      IF(IC.EQ.4) GO TO 200
      DO 103 J = 1, 3
      IF(J.EQ.1) GO TO 104
      ITEMP = II(1)
      II(1) = II(2)
      II(2) = II(3)
      II(3) = ITEMP
  104 W(II(3)) = 1.0
      W(II(2)) = C(II(2),II(3)) * W(II(3))
      W(II(1)) =
     1(C(II(1),II(2))*W(II(2))+C(II(1),II(3))*W(II(3)))/2.0
      TT = TT+1.0
      TN = 0.0
      DO 109 I = 1, IC
  109 TN = TN + W(II(I))
      DO 105 I = 1, IC
  105 W1(I) = W(I) / TN
      DO 106 I = 1, IC
  106 W2(I) = W2(I)*W1(I)
  103 CONTINUE
  400 TN2 = 0.0
C
C     CALCULATE GEOMETRIC MEAN OF ALL WEIGHT VECTORS
C
      DO 107 I = 1,IC
      W3(I) = W2(I) ** (1.0/TT)
  107 TN2 = TN2 + W3(I)
      DO 108 I = 1, IC
  108 CYW(I) = W3(I) / TN2
      WRITE(6,*) (CYW(I),I=1,IC)
```

```
      GO TO 500
  200 DO 201 I = 1, IC
      IF(I.EQ.1) GO TO 202
      ITEMP = II(1)
      II(1) = II(2)
      II(2) = II(3)
      II(3) = II(4)
      II(4) = ITEMP
  202 DO 203 J = 1, IC
      IF(J.EQ.1) GO TO 204
      ITEMP = II(2)
      II(2) = II(3)
      II(3) = II(4)
      II(4) = ITEMP
      IF(J.EQ.4) GO TO 203
  204 W(II(4)) = 1.0
      W(II(3)) = C(II(3),II(4)) * W(II(4))
      W(II(2)) = (C(II(2),II(3))*W(II(3))+C(II(2),II(4)) *
     1           W(II(4)))/2.0
      W(II(1)) = (C(II(1),II(2))*W(II(2))+C(II(1),II(3)) *
     1           W(II(3)) + C(II(1),II(4))*W(II(4)))/3.0
      TT = TT+1.0
      TN = 0.0
      DO 209 K = 1, IC
  209 TN = TN + W(II(K))
      DO 207 K = 1, IC
  207 W1(K) = W(K) / TN
      DO 208 K = 1, IC
  208 W2(K) = W2(K)*W1(K)
  203 CONTINUE
  201 CONTINUE
      GO TO 400
  300 DO 301 I = 1, IC
      IF(I.EQ.1) GO TO 302
      ITEMP1 = II(1)
      II(1) = II(2)
      II(2) = II(3)
      II(3) = II(4)
      II(4) = II(5)
      II(5) = ITEMP1
  302 DO 303 J = 1, IC
      IF(J.EQ.1) GO TO 304
      ITEMP = II(2)
      II(2) = II(3)
      II(3) = II(4)
      II(4) = II(5)
      II(5) = ITEMP
      IF(J.EQ.IC) GO TO 303
  304 DO 305 K = 1,4
      IF(K.EQ.1) GO TO 306
      ITEMP = II(3)
      II(3) = II(4)
```

```
      II(4) = II(5)
      II(5) = ITEMP
      IF(K.EQ.4) GO TO 305
  306 W(II(5)) = 1.0
      W(II(4)) = C(II(4),II(5)) * W(II(5))
      W(II(3)) = (C(II(3),II(4))*W(II(4))+C(II(3),II(5)) *
     1           W(II(5)))/2.0
      W(II(2)) = (C(II(2),II(3))*W(II(3))+C(II(2),II(4)) *
     1           W(II(4)) + C(II(2),II(5))*W(II(5)))/3.0
      W(II(1)) = (C(II(1),II(2))*W(II(2))+C(II(1),II(3)) *
     1           W(II(3)) + C(II(1),II(4))*W(II(4)) + 2
     2           C(II(1),II(5)) * W(II(5)))/4.0
      TT = TT+1.0
      TN = 0.0
      DO 309 L = 1, IC
  309 TN = TN + W(II(L))
      DO 307 L = 1, IC
  307 W1(L) = W(L) / TN
      DO 308 L = 1, IC
  308 W2(L) = W2(L)*W1(L)
  305 CONTINUE
  303 CONTINUE
  301 CONTINUE
      GO TO 400
  500 RETURN
      END
C
C***********************************************************
C
      SUBROUTINE MODEL6(IC, TW, NMRUNS, C)
C
C***********************************************************
C
      INTEGER  IC,IT,NMRUNS
      REAL TN1, TN2, TW(5), R(5,5), C(5,5), W(5), TAKW(5)
C     CALCULATE THE WEIGHTS
      TN2 = 0.0
      DO 106 I = IC, 1, -1
      W(I) = 0.0
      IF(I.EQ.IC) GO TO 108
      DO 107 J = I+1, IC
      W(I) = W(I) + C(I,J) * W(J)
  107 CONTINUE
      GO TO 110
  108 W(I) = 1.0
  110 TN2 = TN2 + W(I)
  106 CONTINUE
      DO 109 I = 1, IC
      TAKW(I) = W(I) / TN2
  109 CONTINUE
      RETURN
      END
```

2

VITA

Kook Jin Nam

Candidate for the Degree of

Doctor of Philosophy

Thesis: NEW WEIGHTING PROCEDURES MINIMIZING JUDGMENTAL ERROR AND REFINING INCONSISTENCY FOR MULTIPLE CRITERIA DECISION MAKING PROBLEMS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Taegu City, Republic of Korea, October 25, 1951, the son of Mr. and Mrs. Hyodeok Nam.

Education: Graduate from Kyesung High School, Taegu, Republic of Korea, in January, 1971; received Bachelor of Science degree in Ordnance Engineering from Korea Military Academy, Seoul, Republic of Korea, in March, 1975; received Master of science degree in Operations Research from U.S. Naval Postgraduate School, Monterey, California, in September, 1984; completed requirements for the Doctor of Philosophy degree in Industrial Engineering at Oklahoma State University, Stillwater, Oklahoma, in May, 1990.

Professional Experience: Platoon leader in 21st Infantry Division, Republic of Korea Army(ROKA), August, 1975 to June, 1977; Platoon leader in Seoul Security Command, June, 1977 to May, 1978; Company commander, in 9th Infantry Division, ROKA, June, 1978 to February, 1981; Company commander, in Joint Security Area, Panmunjum, Korea, March, 1981 to February, 1982; Graduate student in U.S. Naval Postgraduate School, Monterey, California, September, 1982 to September, 1984; Staff officer in ROKA Headquarters, Seoul, Korea, October, 1984 to July, 1986; Teaching Assistant, School of Industrial Engineering and Management, Oklahoma State University, August, 1987 to December, 1989.