PREFERENCE ELICITATION FROM PAIRWISE COMPARISONS IN MULTI-CRITERIA DECISION MAKING

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

February 2011

By
Sajid Siraj
School of Computer Science

Contents

Word count = 47873

List of Tables

List of Figures

University Name: The University of Manchester
Candidate's Name: Sajid Siraj
Degree Title: Doctor of Philosophy
Thesis Title: Preference Elicitation from Pairwise Comparisons
Date: 07-December-2010

Abstract

Decision making is an essential activity for humans and often becomes complex in the presence of uncertainty or insufficient knowledge. This research aims at estimating preferences using pairwise comparisons.

A decision maker uses pairwise comparison when he/she is unable to directly assign criteria weights or scores to the available options. The judgments provided in pairwise comparisons may not always be consistent for several reasons. Experimentation has been used to obtain statistical evidence related to the widely-used consistency measures. The results highlight the need to propose new consistency measures. Two new consistency measures - termed *congruence* and *dissonance* - are proposed to aid the decision maker in the process of elicitation. Inconsistencies in pairwise comparisons are of two types i.e. cardinal and ordinal. It is shown that both cardinal and ordinal consistency can be improved with the help of these two measures. A heuristic method is then devised to detect and remove intransitive judgments. The results suggest that the devised method is feasible for improving ordinal consistency and is computationally more efficient than the optimization-based methods.

There exist situations when revision of judgments is not allowed and prioritization is required without attempting to remove inconsistency. A new prioritization method has been proposed using the graph-theoretic approach. Although the performance of the proposed prioritization method was found to be comparable to other approaches, it has practical limitation in terms of computation time. As a consequence, the problem of prioritization is explored as an optimization problem. A new method based on multi-objective optimization is formulated that offers multiple non-dominated solutions and outperforms all other relevant methods for inconsistent set of judgments.

A priority estimation tool (PriEsT) has been developed that implements the proposed consistency measures and prioritization methods. In order to show the benefits of PriEsT, a case study involving Telecom infrastructure selection is presented.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I thank God Almighty for giving me enough powers, physical and mental, to pursue this first research endeavor. My parents have been the single greatest investor in my career; I have kept their pictures in my heart throughout the course of study. I also had their pictures in my wallet that got stolen at Manchester Airport last Christmas. I am also grateful to Hina who made available her support in a number of ways.

I acknowledge the contributions of my supervisors (Ludmil Mikhailov and John Keane) as marvelous. They have helped me taking the first step towards self-actualization, paving my way to become a researcher who aims to benefit his surroundings.

Thanks to the teams developing open-source software that helped in different ways whilst performing this research. The list includes Netbeans IDE, Qt, Subversion, JAMA, jMetal, Lyx, MikTEX and Texlive.

Recollecting the moments I spent here in Manchester for last three years; I added new friends, colleagues and several new thoughts in my red book. I thank all the friends, university staff and other people who helped me towards getting my research done.

Last but not least, I want to thank COMSATS Institute of Information Technology, Pakistan. They have financially funded me throughout the course of study.

Abbreviations & Acronyms

AHP	Analytic Hierarchy Process
AN	Additive Normalization
CC	Cardinal Consistency
CM	Consistency Measure
CR	Consistency Ratio
DLS	Direct Least Squares
DM	Decision Maker
EAST	Enumerating All Spanning Trees
EMO	Evolutionary Multiobjective Optimization
EV	Eigenvector
GA	Genetic Algorithm
GCI	Geometric Consistency Index
GM	Geometric Mean
ILP	Integer Linear Programming
LLAV	Logarithmic Least Absolute Values
LLS	Logarithmic Least Squares
MAD	Mean Absolute Deviation
MCDM	Multi-criteria Decision Making
MNV	Minimum Number of Violations
NCS	Normalized Column Sum
NV	Number of Violations
OC	Ordinal Consistency
PC	Pairwise Comparison
PCM	Pairwise Comparison Matrix
PriEsT	Priority Estimation Tool
PrInT	Prioritization using Indirect Judgments
TD	Total Deviation
TOP	Two-Objective Prioritization
UFO	Unusual and False Observations
WLS	Weighted Least Squares

Chapter 1

Introduction

This chapter provides an introduction to the research presented in this thesis.

1.1 Motivation

Decision making is essential for humans and often becomes complex in the presence of uncertainty or insufficient knowledge. Multi-criteria Decision Making (MCDM) refers to making decisions in the presence of several (and often competing) attributes or objectives. MCDM is considered to be an important and active area of research. In MCDM, a decision maker (DM) defines the goals which he/she wishes to achieve; the problem is then formulated and alternatives are discovered. These alternatives are then evaluated against each criterion and eventually prioritized. This research is focused on the prioritization stage of the MCDM process.

There exist several MCDM techniques to assist DMs in prioritizing alternatives. The relative importance of alternatives is usually represented in the form of a vector, termed a *preference vector* or *priority vector*. Most of the MCDM techniques involve the use of pairwise comparison (PC) whilst estimating preferences and the use of PC is becoming increasingly popular. DMs use PCs when unable to directly assign criteria weights or scores to the available alternatives. Moreover, the criteria in MCDM are not always *tangible* and thus may lack any scale of measurement; the PC method is considered to be "central" for measuring such *intangible* criteria [1].

In PC, a decision maker (DM) is asked to compare only two *stimuli* (criteria or alternatives) at a time, and a prioritization method is used to estimate the priority vector from a given set of judgments. Whilst there exist several approaches to this, determining the most suitable prioritization method remains an open problem. Different methods perform differently and no method outperforms other methods in all situations. Most prioritization methods are based on single-objective optimization; in contrast, there is a recent trend to use multi-objective optimization. Mikhailov [2] proposed the simultaneous optimization of two objectives for prioritization (TOP). The benefit of using TOP is that it can generate multiple non-dominated solutions for DMs to interactively select the most appropriate.

The PC judgments are provided with respect to some predetermined preference scale and are structured in a PC matrix (PCM). Each judgment contains two types of information about preference i.e. ordinal and cardinal. Ordinal information provides the direction of preference while the cardinal information provides the strength of preference. A priority vector gives an *ideal ranking* when it satisfies all the preference directions expressed in the given judgments. When a complete set of PC judgments is acquired, there exists the possibility of inconsistency among these judgments. Inconsistencies in PCM can be either cardinal or ordinal.

A PCM is considered to be *acceptable* when the value of its Consistency Ratio (CR) - the most widely-used measure for consistency - remains below 0.1. A number of Monte-Carlo based experiments have led to the conclusion that ordinally inconsistent (*intransitive*) PCMs are not always rejected by the criterion of $CR \leqslant 0.1$. Moreover, the possibility of having an *acceptable* PCM is sensitive to the selected scale of measurement. There exist two other well-known measures of consistency: Geometric Consistency Index (GCI) and Consistency Measure (CM). GCI has been found to have a linear relationship with CR, therefore the above mentioned inference is equally valid for GCI. Considering CM, although this has been shown to be useful in locating the most inconsistent set of judgments, there exists no associated threshold that can be used as a criterion for acceptance or rejection of a PCM.

This research has focused firstly on measuring the level of inconsistency in a PCM and then has proceeded to the problem of prioritization in the presence of inconsistency.

1.2 Aims & Objectives

There exists a need to measure individual contributions of PC judgments towards the overall inconsistency of a PCM. We aim to propose such a measure that will provide a decision aid to DMs for revising their inconsistent judgments.

There exist several measures for cardinal inconsistency, in contrast, ordinal inconsistency has received relatively less attention in the literature. The issue of measuring ordinal inconsistency will be investigated in detail. This is important as a priority vector with ideal ranking is only possible when judgments are ordinally consistent.

When inconsistency is considered to be irreducible, a prioritization method must be applied to inconsistent set of judgments. The second aim of this research is to propose such a prioritization method that should outperform currently available methods.

A decision support system will be developed that will support the newly proposed measures and prioritization methods.

1.3 Contributions

1.3.1 Two New Measures: Congruence & Dissonance

To begin with, experimentation has been used to obtain statistical evidences related to the widely-used consistency measures. Two new consistency measures - termed *congruence* and *dissonance* - are proposed to aid a DM in the elicitation process. It is shown that both cardinal and ordinal consistency can be improved with the help of these two measures. However, further investigation is needed to devise a threshold of acceptance based on these two measures.

1.3.2 A Heuristic Method to Rectify Intransitive Judgments

The issue of ordinal inconsistency is investigated here using a graph theoretic approach. A heuristic method has been devised to detect and remove *intransitive* judgments. The results suggest that the devised method is feasible for improving ordinal consistency and is computationally more efficient than optimization based methods.

1.3.3 Prioritization using Graph-theoretic Approach

Further, using this graph-theoretic approach; a new prioritization method has been proposed based on *enumerating all spanning trees* (EAST). It has been shown that the EAST method is applicable to incomplete PCMs without modification, unlike other popular methods which require intermediate steps to estimate missing judgments. Although the performance of the EAST method is comparable to other approaches, it has practical limitation in terms of computation time. As a consequence, the problem of prioritization has been explored as an optimization problem.

1.3.4 Prioritization using Multi-objective Optimization

As mentioned earlier, Mikhailov [2] proposed the simultaneous optimization of two objectives. Here, we extend this approach to minimize three objectives. The concept of direct and indirect judgments has been investigated and a new method called *Prioritization using Indirect Judgments* (PrInT) is then formulated. PrInT, like TOP, offers multiple non-dominated solutions and outperforms all other relevant methods for an *intransitive* set of judgments. PrInT generates solutions with minimum ordinal violations that remain as close as possible to the direct and indirect judgments in terms of Euclidean distance.

There exist situations in practice where user interaction is not possible. PrInT, however, generates multiple solutions where no solution is inferior to others. Therefore, this issue requires further investigation in order to select the most appropriate solution from a set of non-dominated solutions.

1.3.5 Priority Estimation Tool (PriEsT)

Based on the proposed consistency measures and the proposed prioritization methods, a priority estimation tool (PriEsT) has been developed as a decision support system. PriEsT offers a wide range of non-dominated solutions using the PrInT method. DMs have the flexibility to select any of these non-dominated solutions according to their requirements. Whilst other software tools extract a single solution from given judgments, PriEsT offers multiple equally good solutions. Users of PriEsT can select different prioritization methods to estimate preferences from the same PCM. This makes PriEsT an appropriate research tool to evaluate such methods.

PriEsT also has the ability to assist DMs in revising their judgments based on the *congruence* and *dissonance* measures. Existing tools do not offer visualization of a PCM in the form of a graph. PriEsT has the ability to view a PCM as a graph and highlights the presence of ordinal inconsistency. This graph view is shown to be helpful in rectifying an *intransitive* set of judgments.

PriEsT is demonstrated and evaluated using practical data acquired from a recent case-study. The presence of *intransitive* judgments has been highlighted in some of the acquired PCMs. It has been found that correcting these judgments leads to a different ranking of available alternatives. As mentioned earlier, there exist situations where prioritization is required without changing initial judgments. Considering this, PriEsT has been used to generate a set of non-dominated solutions using PrInT. Choosing a different solution from the set of non-dominated ones obviously ends with different weights. Although different, no solution is declared to be inferior.

1.4 Structure

This thesis contains eight chapters. The structure of the thesis is shown in Fig. 1.1. This chapter has introduced the research of the thesis. Chapter 2 overviews decision making techniques and highlights the importance of the PC method. It reviews related literature and identifies a need for additional consistency measures.

Chapter 3 begins with experimentation performed to obtain statistical evidences related to the most widely-used consistency measures. Two new consistency measures - congruence and dissonance - are then proposed and their use is illustrated with the help of examples. Chapter 4 proposes methods to improve both cardinal and ordinal consistency with the help of the two measures proposed in Chapter 3. The chapter then proposes a heuristic method to detect and remove *intransitive* judgments.

Chapter 5 introduces a new prioritization method called EAST and discusses the benefits and the limitations of this method. Due to the practical limitations of EAST that are identified, the issue of prioritization is re-addressed in Chapter 6 using a multi-objective optimization approach. This chapter formulates the PrInT method that offers DMs a wide range of non-dominated solutions.

Chapter 7 presents the PriEsT software as a decision support tool offering several new features. This includes the use of congruence and dissonance

Figure 1.1: Overview of the document

as discussed in Chapters 3 and 4, and the prioritization methods proposed in Chapters 5 and 6. A case study is presented to demonstrate the use of PriEsT.

Chapter 8 concludes with a summary of the research work and associated conclusions. The chapter ends with several suggestions for future work.

Chapter 2

Background

This chapter introduces decision making methods and highlights the importance of pairwise comparisons. It reviews the literature related to the problem of eliciting preferences from pairwise comparisons and several associated methods. The chapter ends with a detailed discussion on inconsistency that generates a need to propose new consistency measures.

2.1 Decision Making

Making decisions is a cognitive process of ranking available options in order to choose the most desirable. A goal of making decisions is to achieve the most desired objectives with the least expected penalties. Decision making often becomes complex in the presence of uncertainty or insufficient knowledge. Consider a dress selection problem for example, one has to consider climate, price, fashion and prestige in order to buy the most appropriate dress. Similarly, a corporate decision may depend on profit, reputation, employee satisfaction and long-term benefits etc.

A *criterion* is a characteristic property by which something can be judged. Different criteria usually uncover different dimensions of the available choices and may be declared as a *benefit* or *cost*. These criteria are either qualitative or quantitative in nature and the precedence of these criteria may vary dramatically under different circumstances. A rational *decision maker* (DM) is an individual who maximizes profits and/or minimizes losses [3]. The *alternatives* represent different choices available to a DM. The presence of several criteria may itself negatively impact the rational comparison of alternatives by a DM.

This possible confusion or uncertainty may lead to a naïve approach of simply adding up pluses and minuses. Whilst estimating preferences, such uncertainty may also introduce cognitive dissonance i.e. the holding of two contradictory beliefs simultaneously.

Multi-criteria Decision-Making (MCDM) refers to making decisions in the presence of several attributes or objectives. Zimmermann [4] categorized MCDM into two main types based on continuous and discrete decision space i.e. Multi-objective Decision Making and Multi-attribute Decision Making, respectively.

In MCDM, all available choices are measured against each criterion, and these measurements then facilitate the selection of one satisfactory choice. A DM is usually assisted in highlighting conflicts among the criteria and then a feasible compromise may be suggested, which is the closest to the ideal solution. Although most problem holders are interested in seeking the best alternative, some problems may require the relative importance of all available alternatives.

The use of MCDM techniques should enable several possible benefits including better use of available information, clearer justification to stake-holders, cost and time savings (profits), conflict resolution in group decision making and better response to unforeseen circumstances.

2.1.1 MCDM Process

MCDM is a multistage process that starts from defining a goal (or multiple goals) and usually terminates with execution of the selected choice. At a meta-level, the termination itself can be seen as intermediate stage of another (parent) MCDM process. The following are the major steps involved in MCDM:

1. **Conceptualize:** To begin with, the DM clearly defines the goals which he/she wishes to achieve. It is important at this initial stage to uncover all possible effects of the defined goals.

2. **Formulate:** Goal setting then follows planning and questions. Analysts use their specialized knowledge to assist DMs in defining a list of measurable criteria that should be considered. Certain measurements are identified as markers for achieved objectives. These measures usually assist all stake-holders to understand whether an objective is being met.

3. **Discover:** The availability of alternatives is usually time-related, hence

some may become unavailable before the decision is finalized. It is therefore important to *discover* as many alternatives as possible in a given time span. A limited number of alternatives are usually presented to DMs following the rule of 7 ± 2 [5]. The list of alternatives usually includes 'no action' or 'status quo' wherever possible, and delaying a decision may end up in this option category.

4. **Evaluate:** The next step is to evaluate the performance values for available alternatives against the considered criteria. These values are usually given by expert(s) related to each criterion under consideration. The performance values are usually obtained through a judgment acquisition process. The judgments can be derived from other data provided by non-experts, for example, opinions taken from potential customers or stakeholders. Experts usually may refer to other experts during the process.

5. **Prioritize:** The next process involves ranking the alternatives based on the judgments acquired in the previous step. There exists several mathematical techniques to derive preference weights from given judgments. If the judgments can be acquired interactively, this step is executed iteratively with the previous step (step 4). This allows judgments to be changed/corrected in order to improve quality of information. The elicited preference weights for given alternatives are suggested to the DM.

6. **Execute:** Finally, a DM selects the most feasible alternative suggested in the previous step (step 5). The outcome of this step remains useful, as the post-decision effects are usually fed into another MCDM problem.

These six steps are visualized in Figure 2.1 with a hypothetical problem of choosing the most appropriate food to eat. The DM is supposed to be careful about his/her health, finance and environment. Therefore, in step 2, (s)he considers the criteria of price, the amount of calories and the CO_2 emissions caused by the possible foods. Next, available options are explored and compared in step 3 and 4, respectively. In the prioritization phase (step 5), the most appropriate food is found to be fresh salad. Purchasing the fresh salad terminates this process, however, this may affect another MCDM problem of resort selection for holidays.

Duckstein and Opricovic [6] suggested MCDM problems can be seen at two different levels i.e. *managerial* and *engineering*. The managerial level defines

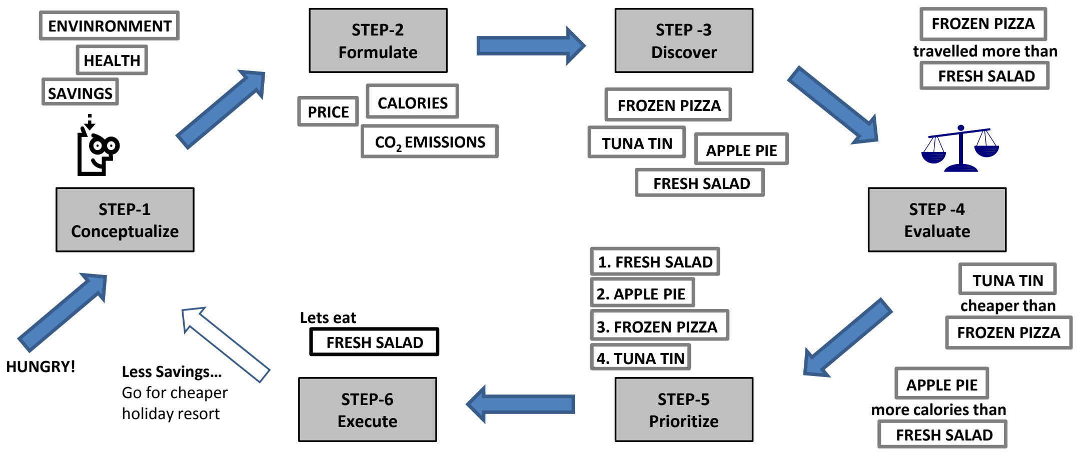


Figure 2.1: The general structure of the MCDM process

the goals and chooses the final alternative, while the engineering level proposes possible solutions (usually non-inferior ones) after a comprehensive analysis based on technical criteria and optimization procedures.

The managerial level generally is involved directly in the steps 1, 2 and 6. Other stages are mostly delegated to the engineering level [7]. Although each stage relies on its predecessor, some level of pipe-lining may be achieved to reduce overall duration.

The research in this thesis begins with step 4 and mainly focuses on step 5 i.e. the acquisition of judgment data and its use in prioritization of the available alternatives.

2.1.2 Prioritization in MCDM

Consider a problem defined on n alternatives $A_1, A_2, ..., A_n$ in the presence of t decision criteria $C_1, C_2, ..., C_t$. Each criterion, C_k, has an associated relative weight of importance, ϖ_k. These weights are usually normalized to have $\sum_k \varpi_k = 1$, where $k = 1, 2, ...t$. This problem can be expressed in an $n \times t$ matrix, termed a *decision matrix* [8]. In a decision matrix, each row indicates an alternative, A_i, and each column represents a criterion, C_k. A general decision matrix has the following form:

$$D = \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1t} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2t} \\ \dots & \dots & \dots & \dots \\ \delta_{n1} & \delta_{n2} & \dots & \delta_{nt} \end{bmatrix} \tag{2.1}$$

or $D = [\delta_{ik}]_{n \times t}$, where δ_{ik} is the performance value of A_i for C_k.

The performance values are usually derived from the judgments acquired from a DM or a consulted expert. There are also situations where δ_{ik} represents the aggregated performance value of a group e.g. voting or survey polls. When multiple DMs are involved in the decision making process, the *decision matrix* explodes into a three-dimensional structure where each plane represents a decision matrix for each DM. This three-dimensional structure can be termed a *decision cube*.

The MCDM prioritization problems can also be classified according to the type of available information i.e. deterministic, stochastic, or fuzzy information [9]. A combination of types is also possible.

Wallenius et al. [10] has recently shown the growing importance of MCDM methods and their applications in various real-world problems. The widely-used MCDM techniques are discussed in the next section.

2.1.3 MCDM Prioritization Methods

There exist several MCDM techniques to assist a DM in prioritizing alternatives, A_1, A_2, ..., A_n. As one would expect, different methods may yield different results to the same problem. The prioritization methods in MCDM are usually of two types i.e. *compensation* and *outranking* methods. The compensation methods calculate aggregated performance value (or utility) for each alternative. In contrast, outranking methods are based on multiple comparisons between alternatives that usually lead to an ordinal ranking instead of finding a global utility.

The widely used MCDM techniques are briefly discussed below.

2.1.3.1 WSM

The weighted sum model (WSM) is a commonly used compensation approach where the overall score, Δ_i, for an alternative, A_i, is calculated as:-

$$\Delta_i = \sum_{k=1}^{t} \varpi_k \delta_{ik} \tag{2.2}$$

The alternative, A_i^* with the maximum score is considered to be the most appropriate (or *winner*). WSM models all the criteria as *benefit criteria* implying that the feasibility of each option increases with its value.

The WSM technique is also known as the simplified multi-attribute rating approach (SMART) [9]. It is useful only for simple problems involving the same unit of measurement (e.g. feet, stones, euros etc.), as it is not justifiable to simply add different units together.

2.1.3.2 WPM

The weighted product model (WPM) is similar to WSM, where multiplication is used to aggregate results. WPM also models all criteria as *benefit criteria*. The overall score, Δ_i, for alternative, A_i, is calculated as:-

$$\Delta_i = \prod_{k=1}^{t} (\delta_{ik})^{\varpi_k} \tag{2.3}$$

The alternative, A_i^*, with maximum score is considered to be the most appropriate (or *winner*). WPM, unlike WSM, supports the use of ratio measurements in order to aggregate criteria with different units of measurement [9].

2.1.3.3 ELECTRE

ELECTRE is a family of outranking methods originated to avoid problems using WSM [11]. ELECTRE is an outranking approach and therefore suitable for situations where criteria are not measurable by a common standard.

This method uses normalized version of all the performance values. The performance values for each criterion are normalized to sum to one, as follows:-

$$\bar{\delta}_{ik} = \frac{\delta_{ik}}{\sum_j \delta_{jk}} \tag{2.4}$$

For each pair of alternatives, DM categorized the difference as dominating, indifferent or incomparable. Two indices, *concordance* and *discordance*, are then calculated based on these pairwise comparisons of alternatives. The concordance index , $\tilde{c}_{ij}$, is the sum of weights for all those criteria where A_i wins over A_j i.e.

$$\tilde{c}_{ij} = \sum_{k:\bar{\delta}_{ik} > \bar{\delta}_{jk}} \varpi_k \tag{2.5}$$

where $i \neq j$. The discordance index, $\tilde{d}_{ij}$, can be calculated using the following formula:-

$$\tilde{d}_{ij} = \begin{cases} \max_k \left(\frac{\bar{\delta}_{jk} - \bar{\delta}_{ik}}{\max_r \bar{\delta}_{rk} - \min_r \bar{\delta}_{rk}} \right) & if\, \bar{\delta}_{ik} > \bar{\delta}_{jk} \\ 0 & otherwise \end{cases} \tag{2.6}$$

where $i \neq j$.

A set of acceptable alternatives is obtained with the help of threshold values associated with these indices. Changing the threshold values governs the number of alternatives assessed to be feasible. The most apparent weakness of this method is the use of arbitrary threshold values [12, 13], as choosing different threshold values may generate different results. There has been several improvements since ELECTRE was proposed, see [14] for further information.

2.1.3.4 TOPSIS

TOPSIS was proposed by Hwang and Yoon [12] to address identified shortcomings in ELECTRE. This method searches for the alternative, A_i^*, which is considered closest to the *ideal solution* and furthest from the *negative-ideal solution*.

The *ideal solution* is derived as a composite of the best performance values found in D for each criterion, C_k. Similarly, the *negative-ideal solution* is the composite of the worst performance values.

Unlike ELECTRE, TOPSIS is a compensation method and is able to generate global weights. This method starts with the weighted normalization of all performance values against each criterion i.e.

$$x_{ik} = \frac{\varpi_k \delta_{ik}}{\sqrt{\sum_j \delta_{jk}^2}} \tag{2.7}$$

The ideal (x^*) and negative ideal (x^-) values are calculated for each criterion as follows:-

$$x_k^* = \max_i (x_{ik}) \tag{2.8}$$

$$x_k^- = \min_i (x_{ik}) \tag{2.9}$$

where all the criteria are considered benefit criteria. The ideal and negative ideal values can be swapped for cost criteria.

The distances from ideal and negative-ideal solutions are calculated in a

Euclidean sense i.e.

$$\Delta_i^- = \sqrt{\sum_k \left(x_{ik} - x_k^-\right)^2}$$
$$\Delta_i^* = \sqrt{\sum_k \left(x_{ik} - x_k^*\right)^2}$$

The overall weight, Δ_i, is calculated as:-

$$\Delta_i = \frac{\Delta_i^-}{\Delta_i^- + \Delta_i^*} \tag{2.10}$$

The alternative, A_i^*, with highest overall weights is considered the most feasible. A complete ranking can be achieved using this approach.

2.1.3.5 PROMETHEE

PROMETHEE is a family of algorithms based on the outranking approach, see [15] for details and history of these methods. Behzadian et al. [16] comprehensively summarize PROMETHEE-based methodologies and applications.

The method starts with the pairwise comparisons of available alternatives. A DM may declare the difference between any two alternatives to be *dominating* or *indifferent*. Alternatively, a DM may also suggest the two alternatives are *incomparable*. A preference relation between two alternatives is measured in the form of an index, ς, calculated as:-

$$\varsigma_{ij} = \mathcal{F}(A_i, A_j)$$

where $i, j = 1, 2, ..., n$ and $\mathcal{F}$ is a piece-wise continuous relation producing a value between zero and one i.e. $0 \leqslant \varsigma_{ij} \leqslant 1$. The value should be greater than zero when A_i is preferred over A_j. *Indifference* implies that both ς_{ij} and ς_{ji} are equal to zero. The value of one ($\varsigma_{ij} = 1$) implies complete dominance of A_i over A_j for a given criterion. Brans and Mareschal [15] proposed six types of these functions and considered them to be applicable to most practical MCDM problems.

For each pair of alternatives, all the preference indices measured for different criteria are aggregated using the weighted sum technique i.e.

$$\wp_{ij} = \sum_{k=1}^{t} \varpi_k \varsigma_{ij}$$

The overall strength of an alternative, A_i, is calculated as $\Delta_i^+ = \frac{1}{n-1} \sum_j \wp_{ij}$.

Brans and Mareschal [15] explained this strength using a graph-theoretic approach where Δ_i^+ is visualized as a *positive outranking flow*. Similarly, the overall weakness of the alternative A_i is calculated as $\Delta_i^- = \frac{1}{n-1}\sum_j \wp_{ji}$.

The strengths and weaknesses are used in PROMETHEE-I to obtain partial ranking. PROMETHEE-II calculates a net outranking flow, Δ_i, as the difference between the calculated strength and weakness i.e.

$$\Delta_i = \Delta_i^+ - \Delta_i^-$$

The alternative showing maximum net outflow is considered to be the most feasible. A complete ranking can be achieved using PROMETHEE-II.

2.1.3.6 AHP

The Analytic Hierarchy Process (AHP), proposed by Saaty [17], is based on the method of pairwise comparison (PC) to assess relative importance of *criteria* and *alternatives*. The main benefit of using this approach is to convert both objective and subjective judgments into relative weights of importance. The applications of AHP are numerous and it has been used worldwide [18, 19]. It was recently considered to be the most active area of research in MCDM [10].

AHP allows a DM to model a given problem using a hierarchical structure where the top of the hierarchy defines the ultimate goal (root). The criteria (nodes) are placed below this goal in a hierarchical way. All the alternatives form the lowest level of this hierarchy (leaves of the tree). This process of AHP is easy to relate with the first four MCDM stages described in section 2.1.1. A typical AHP hierarchy is shown in Figure 2.2 along with the involved MCDM stages.

The alternatives are assessed for the criteria lying at the lowest level. These criteria having no further sub-criteria are the *atomic* characteristics of the alternatives to be judged. In AHP, all the weights are normalized to sum up to one, and the preference values are aggregated for parent criterion using the weighted sum technique i.e.

$$\Delta_i = \sum_{k=1}^{t} \varpi_k \bar{\delta}_{ik} \tag{2.11}$$

The weights are aggregated for each parent criterion and eventually end up giving global weights for the root node (goal). The use of ratio-scale justifies

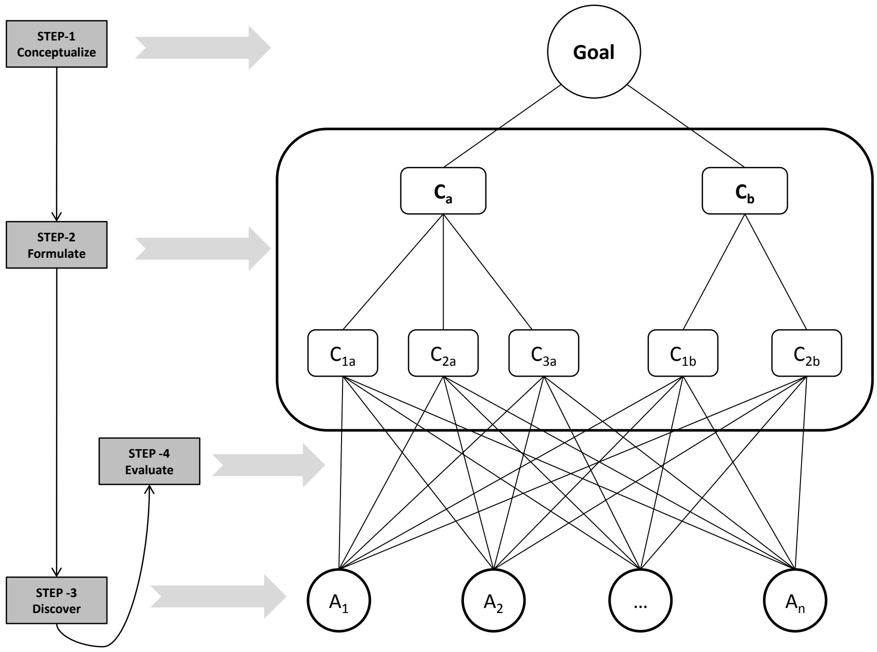


Figure 2.2: Overview of AHP Hierarchy

aggregation of *criteria* involving different units of measurement [1, 20]. AHP also models all the criteria as *benefit criteria*, however, *cost criteria* can also be modeled using a separate hierarchy in parallel [21]. The alternative, A_i^*, having maximum score is considered to be the most appropriate (or the *winner*).

In AHP, the ranking order of n alternatives may change due to the addition of an alternative or removal of an existing one [22]. This phenomenon is termed *rank reversal* and has been widely discussed in the AHP literature [14, 23–26].

The traditional AHP approach requires a complete set of PC judgments to be provided. However, the probability of acquiring incomplete information increases as the number of alternatives increases [27]. In such cases, the traditional approach cannot estimate preferences without estimating the missing information.

Beynon et al. [28] proposed the use of Dempster-Shafer theory (DST) in order to improve the AHP method. The use of DST has the potential of handling incomplete information. An advantage of using this approach is to lower the number of contending alternatives.

AHP has also been generalized into Analytic Network Process (ANP), where hierarchies are replaced with networks and the concept of feedback among criteria is introduced [29].

A recent trend is where Evolutionary Multi-objective Optimization (EMO) techniques are applied in AHP for optimal decision making [30, 31].

2.1.4 Summary

Triantaphyllou [9], in a comparison of techniques widely used in MCDM, concluded that finding the best method may be impossible. Wallenius et al. [10] compared MCDM methods using bibliometric analysis of citations from the year 1979 to 2004. The results showed AHP as the most active and growing area of research, followed by EMO. The use of EMO will be discussed in Chapter 6. Macharis et al. [32] analyzed AHP and PROMETHEE from various perspectives and proposed the use of some AHP features in PROMETHEE to possibly elicit better results.

The intention in the research in this thesis is not to compare these methods, but rather to highlight the use of the PC method. AHP uses PCs to assess both performance values and criteria weights. Roy [33] suggested the PC method for synthesizing preference relational system in ELECTRE. PROMETHEE methods use PCs in order to rank alternatives.

Whilst using DST to compare alternatives, Beynon et al. [28] suggested the use of PCs for assessing criteria weights. Macharis et al. [32] also proposed PCs for assessing criteria in PROMETHEE.

The criteria in MCDM are not always supposed to be *tangible*, and are not measured in well-defined units like dollars, euros or grams. Saaty [1] highlighted this need to measure relative importance of given options for *intangible* criteria and declared the use of PCs as being "central" for this purpose. MCDM methods that do not directly use the PC method may use it as an intermediate step to estimate values for the *intangible* factors.

In the PC method, a DM is asked to compare only two stimuli at a time, and a prioritization technique is used to extract the preference weights from a given set of judgments. Koczkodaj [34] considered PCs to be as important in decision making as derivatives are in calculus or the Eigenvalue is in linear algebra. PCs have been extensively used in the perceptual domain, eliciting preferences concerning feelings and intuition [19].

Next, we discuss the PC method in detail.

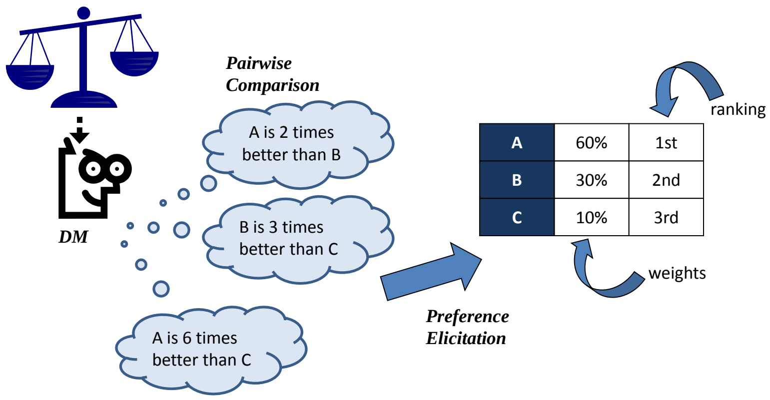


Figure 2.3: Preference Elicitation from Pairwise Comparison

2.2 Pairwise Comparisons

The use of PCs is a *divide-and-conquer* technique suggesting that a respondent is to analyze two items at a time. The respondents are advised to ignore all other items as *noise* during that comparison. Thurstone [35] introduced this scientific approach as the *law of comparative judgment*. As mentioned earlier, the PC method is often used as an intermediate step in MCDM. A DM uses PCs when unable to directly assign criteria weights or scores to the available alternatives. Additionally, the PC method is used in voting systems [36] and multi-agent AI systems [37]. Ranking players (or teams) can also be done with the help of PCs for the tournaments involving head-to-head matches [38, 39].

Figure 2.3 presents an overview of the elicitation process, showing how judgments are acquired and used to calculate preference weights and ranking. A DM can be an individual or organization, or even an agent within a multi-agent AI system.

The use of PCs is a vital part of the prioritization procedure in AHP, which provides a comprehensive and rational framework for structuring a decision problem. In AHP, the pairwise judgments are structured in a PC matrix and a prioritization procedure is applied to derive a corresponding priority vector. If the judgments are consistent then all prioritization methods give the same result. However, in the case of inconsistent judgments, different prioritization methods derive different priority vectors.

When DMs are presented with a number of items to be ranked, it is assumed that they can compare each pair of items and provide an ordinal preference judgment whether an item is preferred to another one (*preference dominance*), or both are equally preferred (*preference equivalence*). It is also assumed that DMs are able to express the strength of their preferences by providing additional cardinal information. However, some level of inconsistency in their preference judgments may exist.

2.2.1 Mathematical Formulation

Consider a prioritization of n elements $E_1, E_2, ..., E_n$. In MCDM, those elements could be either criteria $C_1, C_2, ..., C_n$ or alternatives $A_1, A_2, ..., A_n$. In the PC method, a DM assesses the relative importance of any two elements E_i and E_j by providing a ratio judgment a_{ij}, specifying by how much E_i is preferred to E_j. If the element E_i is preferred to E_j then $a_{ij} > 1$, if the elements are equally preferred then $a_{ij} = 1$ and if E_j is preferred to E_i then $a_{ij} < 1$.

Each set of pairwise ratio judgments, J, with n elements consists of m judgments. Although, m has an upper bound of n^2, however, $n(n-1)$ judgments are sufficient when self-comparison is considered unnecessary. This assumption is applicable to most practical cases, however, self-comparison may contain useful information in applications such as blind testing [40]. The number of judgments, m, is further reduced to $\frac{n(n-1)}{2}$, when the reciprocal property is strictly applicable i.e. $a_{ij} = {}^1/_{a_{ji}}$. The number of required judgments increases significantly as n increases, and becomes very large for $n > 9$.

The n^2 judgments in J can be used to construct a matrix $A = [a_{ij}]$ of the order $n \times n$. The PC matrix (PCM) includes all the self-comparison and reciprocal judgments. In practice, the PCM is constructed from m judgments, making it a positive reciprocal matrix with $a_{ii} = 1$ and $a_{ij} = {}^1/_{a_{ji}}$, as shown below:-

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} & ... & a_{1n} \\ {}^1/_{a_{12}} & 1 & a_{23} & ... & a_{2n} \\ {}^1/_{a_{13}} & {}^1/_{a_{23}} & 1 & ... & ... \\ ... & ... & ... & ... & ... \\ {}^1/_{a_{1n}} & {}^1/_{a_{2n}} & ... & ... & 1 \end{bmatrix} \quad (2.12)$$

The use of matrix notation forces DMs to provide complete set of judgments i.e. $m = \frac{n(n-1)}{2}$. In the case where incomplete judgments have been provided, a method has to be applied firstly to fill in the gaps. The AHP method, for example, structures the comparison judgments into this type of matrix [17].

The relationships between the elements of PCM can be depicted by means of

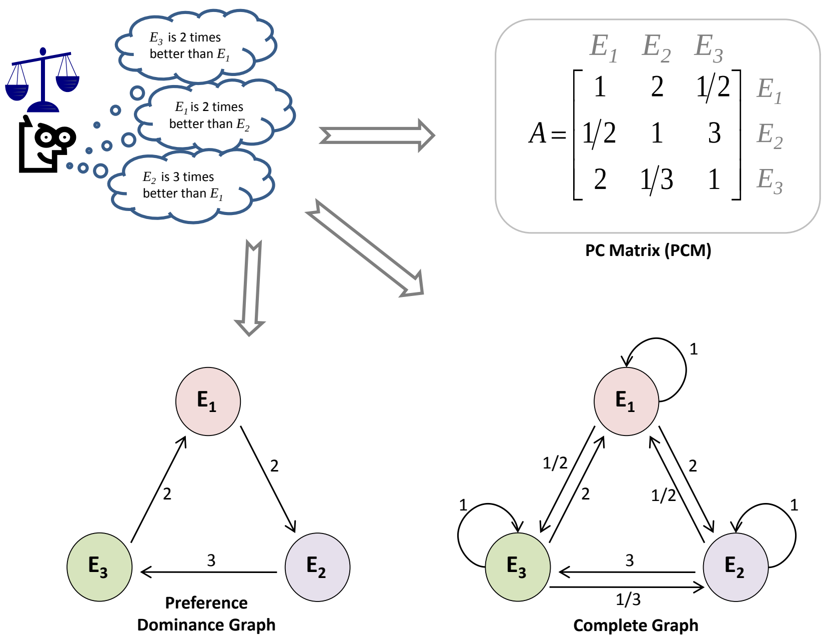


Figure 2.4: Various ways to represent a set of PC judgments

a directed graph (digraph), $G = (E, J)$, where E is the set of nodes representing n elements $E_1, E_2, ..., E_n$, and J represents the set of all ratio judgments $\{a_{ij}\}$ as weighted edges [38, 41]. When a complete set of judgments is provided, the digraph G becomes fully connected.

Figure 2.4 shows different ways to represent a set of PC judgments, J. The judgments are used to construct the PCM given on the top-right of Figure 2.4. The graph on the bottom-left shows the preference dominance where each link is directed from a dominating element to the dominated one. The other graph, on the bottom-right, is the fully-connected digraph including self-comparisons and reciprocal judgments.

2.2.2 Measurement Scales

The cardinal information, a_{ij}, is provided with respect to some predetermined preference scale. In the case of *tangible* criteria, this can be derived from the directly measured information as, for example, weights (in kgs) or price (in euros).

In the case of *intangibles*, Saaty [42] proposed to use a set of verbal judgments that correspond to the ratio-scale of 1 to 9. Pairwise judgments can be provided using an additive-scale, however, the use of ratio-scale is considered

Scale	*Transformation*
Linear	$a_{ij} = cx + b$
Balanced	$a_{ij} = \frac{x}{1+x}$
Geometric	$a_{ij} = bx^c$
Inverse	$a_{ij} = \frac{9}{10-x}$
Logarithmic	$a_{ij} = \log_c (x+1)^b$
Power	$a_{ij} = bc^{x-1}$
Root	$a_{ij} = \sqrt[c]{bx}$

Table 2.1: Measurement scales for ratio comparisons

to be more appropriate for measuring the relative intensities [41, 43, 44].

Despite its criticism and shortcomings [22, 45], the 1 to 9 scale has been used in various applications [18, 19, 23].

Harker and Vargas [41] compared the Saaty's 1 to 9 scale with other proposed scales and considered it to be more appropriate than others. In their comparison, the proposed scales included two linear scales, 1-5 and 1-15, and two non-linear ones, x^2 and $\sqrt{x}$. Their claim was, however, backed up by only one example, leaving the argument unresolved.

Lootsma [46] preferred the use of a geometric scale over the linear one, while Salo and Hämäläinen [24] proposed a balanced scale claiming that the weights generated by their proposed scales are more evenly dispersed as compared to Saaty's 1 to 9 scale. Finan and Hurley [47] proposed to re-calibrate the verbal scale, transforming it into a geometric scale and claimed to achieve better performance over linear scales.

Ishizaka et al. [48] summarized a comprehensive list of transformations applied to Saaty's 1-9 scale with a conclusion that there exists no single scale appropriate for all situations. The list of transformations is given in Table 2.1 where x is a variable judgment ranging between 1 to 9, and b and c are some constant values. The values of b and c are chosen such that $a_{ij} > 1$ when E_i is preferred over E_j and $a_{ij} = 1$ when both E_i and E_j are equally preferred.

Saaty [49] also proposed a finer-grained version of the 1 to 9 scale, where he suggested the use of 1.1 to 1.9 for comparing stimuli with *close resemblance*.

Once the judgments are given against some scale, the next step is to elicit preference weights from the acquired judgments (a_{ij}).

2.3 Prioritization Methods

Suppose that there exists a preference vector $r = (r_1, r_2, ..., r_n)^T$ such that r_i represents the preference intensity of E_i where $i = 1, 2, ..., n$. However, the preference vector r is unknown to a DM and should be estimated. The prioritization problem is to determine a priority vector $w = (w_1, w_2, ..., w_n)^T$ from J, which estimates the unknown preference vector r. The priority weights in ratio-comparisons are considered to have non-zero positive values ($w_i > 0$) and usually calculated with additional constraint of normalization i.e. $\sum w_i = 1$.

There are many prioritization methods that can be applied to derive a priority vector from a set of PC judgments. Choo and Wedley [50] analyzed and numerically compared 18 prioritization methods. It was shown that in the case of error-free (consistent) judgments, all prioritization methods give equal results, however, the results are different when matrix A is inconsistent.

The history of proposed prioritization methods is shown in Figure 2.5 with a timeline. The related methods are grouped chronologically in order to highlight different areas of research, applied to this problem. These methods are briefly discussed below.

2.3.1 Matrix-based Methods

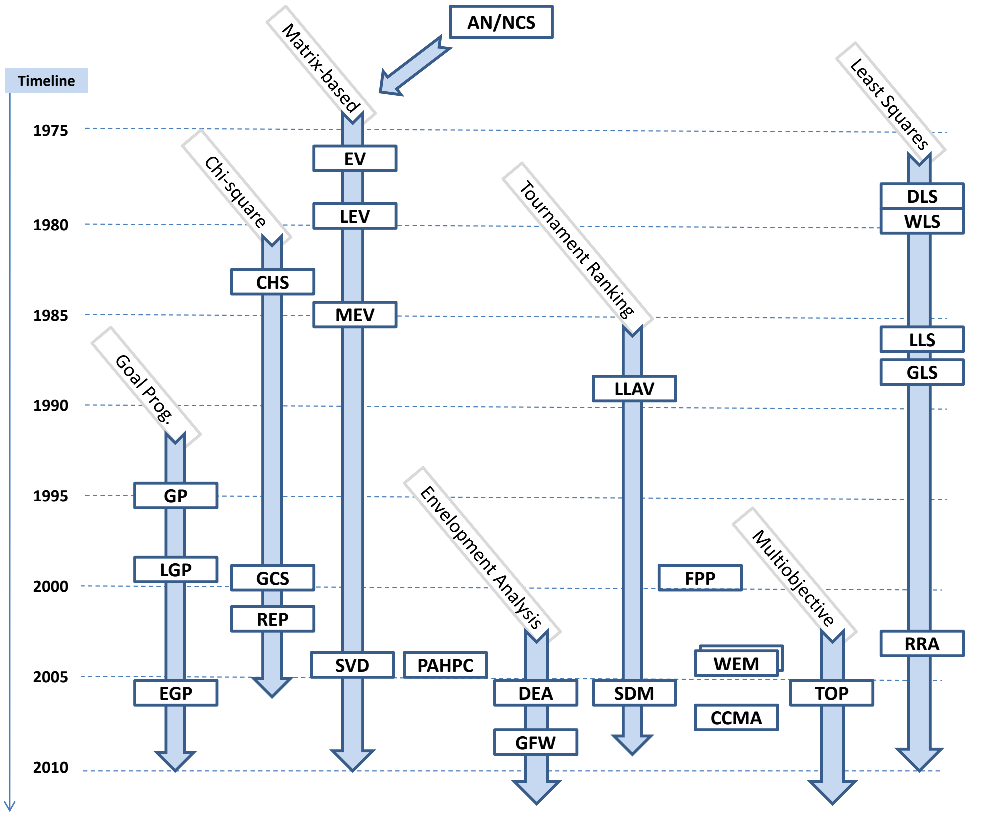
This category of methods depend on the generation of PCM from J and therefore requires either a complete set of PC judgments (i.e. $m = \frac{n(n-1)}{2}$) or the estimation of missing judgments prior to its execution.

2.3.1.1 Normalization Methods

The priority vector, w, can be calculated by normalizing the rows or columns of PCM in different possible ways. The method of *additive normalization* (AN) is obtained by adding all elements in each row and then normalizing the results i.e.

$$\tilde{w}_i = \sum_j a_{ij}$$
$$w_i = \frac{\tilde{w}_i}{\sum_j \tilde{w}_j} \tag{2.13}$$

Another possible way to obtain weights is using *normalized column sum*



AN	Additive normalization	**GFW**	Generating the most favorable weights	**PAHPC**	Possibilistic AHP for crisp data
CCMA	Correlation coefficient maximization	**GLS**	Geometric least squares	**REP**	Robust estimation of priorities
CHS	Chi-square minimization	**GP**	Goal programming	**RRA**	Robust regression analysis
DEA	Data envelopment analysis	**LEV**	Left Eigenvector	**SDM**	Strength-direction method
DLS	Direct least squares	**LGP**	Logarithmic goal programming	**SVD**	Singular value decomposition
EGP	Enhanced goal programming	**LLAV**	Logarithmic least absolute values	**TOP**	Two-objective prioritization
EV	Right Eigenvector	**LLS**	Logarithmic least squares	**WEM**	Worst error minimization
FPP	Fuzzy preference programming	**MEV**	Modified Eigenvector	**WLS**	Weighted least squares
GCS	Generalized chi-square minimization	**NCS**	Normalized column sum		

Figure 2.5: History of the proposed prioritization methods

(NCS), which is mathematically formulated as:-

$$\bar{a}_{ij} = \frac{a_{ij}}{\sum_i a_{ij}}$$
$$w_i = \frac{1}{n}\sum_j \bar{a}_{ij} \tag{2.14}$$

These methods are considered inappropriate, despite being in common use due to their ease of calculation [51–53].

Weights can also be obtained by taking the *geometric mean* (GM) of elements in each row of a PCM [54]. The GM technique can be formulated as:-

$$\bar{a}_{ij} = \frac{a_{ij}}{\sum_i a_{ij}}$$
$$w_i = \left(\prod_{j=1}^{n} \bar{a}_{ij}\right)^{\frac{1}{n}} \tag{2.15}$$

The *logarithmic least squares* method (to be discussed shortly) and GM produce the same results when a set of judgments is complete i.e. $m = \frac{n(n-1)}{2}$.

2.3.1.2 Eigenvector Methods

Saaty [42] proposed the *principal Eigenvector* (EV) of a given PCM to be used as the priority vector w. The computation of *Eigenvector* involves high-order algebraic equations depending on the value of n.

As the high-order algebraic equations are not easy to compute, the solution is usually estimated using the power method i.e. raising A to the power k where the NCS of A^k becomes non-differentiable from the NCS of $A^{(k-1)}$ [55].

The EV method is primarily used in AHP to calculate weights from the PCM. The consistency of the PCM is measured in the form of the Consistency Ratio (CR) [17]. CR is a measure of consistency based on the properties of positive reciprocal matrices (see section 2.6.2.1). The EV solutions are unsatisfactory when inconsistencies are larger ($CR > 0.1$), and has been investigated extensively from both the prioritization and consistency perspective.

Johnson et al. [56] pointed out that the *left Eigenvector* of A can also be used as the weight vector, w. It was emphasized that LEV may produce different results even for "nearly consistent" PCMs.

Cogger and Yu [57] suggested a *modified Eigenvector* approach using only the upper triangle of the PCM. They considered the other half to be redundant information. However, Golany and Kress [58] reported this to be ineffective, highlighting it as an 'outlier' in all the tested methods.

2.3.1.3 Other matrix-based approaches

Recently proposed matrix-based methods include *singular value decomposition* (SVD) and *correlation coefficient maximization approach* (CCMA). The SVD technique is based on the matrix theory of *low rank approximation* [59], and the CCMA estimates weights that are considered best correlated with the columns of a given PCM [60].

2.3.2 Optimization Methods

Except for EV, all the widely-used methods are based on optimization. In these methods, an objective function, $d(w)$, is formulated that needs to be minimized.

2.3.2.1 Least Distance Optimization

Least-distance optimization techniques are based on minimizing the deviation of estimated weights from the set of provided judgments. Chu et al. [61] proposed to minimize the total squared error (or *residuals*) between the given judgments and the estimated weights. The distance function (*minimand*) for the *direct least squares* (DLS) can be formulated as:-

$$d_{DLS}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_{ij} - \frac{w_i}{w_j} \right)^2 \tag{2.16}$$

where $\sum w_i = 1$.

DLS is known to be a non-linear optimization problem having no closed-form solution [50, 58]. Chu et al. [61] gave examples of the discrepancies of DLS and proposed the *weighted least squares* (WLS) method as a modification that minimizes the following distance function:-

$$d_{WLS}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_j a_{ij} - w_i)^2 \tag{2.17}$$

where $\sum w_i = 1$. Unlike DLS, WLS reduces the problem to a system of linear equations that can easily be solved, and provides a unique solution.

Crawford [54] proposed the *logarithmic least squares* (LLS) method, which makes use of the multiplicative properties of the PC judgments. LLS minimization assumes that the best-fit curve of a given type is the curve that has the minimal sum of the logarithmic squared deviations from a given set of data i.e.

$$d_{LLS}(w) = \sum_{i=1}^{n}\sum_{j=1}^{n}(\log a_{ij} - \log w_i + \log w_j)^2 \tag{2.18}$$

where $\sum w_i = 1$. The solution for LLS is always unique and can be found simply as the GM of the rows of PCM, as given in (2.15) [54].

Cook and Kress [62] related the prioritization problem to the tournament ranking problem, and proposed the *logarithmic least absolute value* (LLAV) method, introducing the concept of "*independence of irrelevant objects*". The minimand function for LLAV can be defined as:-

$$d_{LLAV}(w) = \sum_{i=1}^{n}\sum_{j=1}^{n}|\log a_{ij} - \log w_i + \log w_j| \tag{2.19}$$

where $\sum w_i = 1$.

To summarize, all the *minimands* used in the proposed optimization methods are the aggregated sum of the individual deviations, δ, between the given judgment and the estimated one i.e.

$$d(w) = \sum_{i=1}^{n}\sum_{j=1}^{n}\delta\left(a_{ij}, w\right) \tag{2.20}$$

where $\sum w_i = 1$.

2.3.2.2 Chi-Square Minimization

Jensen [63] proposed to use *chi-square* (χ^2) distance as an objective function to estimate weights. χ^2 is a quantitative measure used in statistics usually to determine the relationship between the *observed* data and the *expected* (theoretical) data. Similarly to WLS, this method also avoids the possibility of having multiple solutions.

Zu [64] proposed a *generalized chi-square* method as a generalized form of *chi-square minimization* and demonstrated its desirable properties such as rank preservation and invariance under transpose.

All the minimands used in this category are also the aggregated sum of the individual deviations, as formulated in (2.20).

2.3.2.3 Goal Programming

Bryson [65] proposed a *goal programming* (GP) approach to estimate a vector, w, that is considered to be the closest one to the provided judgments. The main benefit of using GP was referred to as "*single outlier neutralization*" implying that the weights are correctly estimated in the presence of a single outlier. Bryson and Joseph [66] then extended this approach by proposing the *logarithmic goal programming* (LGP) approach, which does not require PCM to be a reciprocal matrix.

Lin [67] introduced the *enhanced goal programming* approach by combining the benefits of GP and LLS in order to obtain unique and outlier-insensitive weights. However, Yuen [53] considers this to be a computationally less efficient method than both GP and LLS.

2.3.2.4 Fuzzy Preference Programming

Mikhailov [68] proposed a *fuzzy preference programming* (FPP) approach, formulating each judgment in J as a fuzzy hyper-line. The problem is defined as finding weights corresponding to the highest intersection of these hyper-lines.

This can be solved as a standard linear program. The problem is to estimate a priority vector that maximizes the satisfaction index (μ), see [68] for further details.

2.3.2.5 Worst Error Minimization

Whilst formulating the distance-based optimization methods, Choo and Wedley [50] introduced the possibility of minimizing the worst deviation among all the calculated individual deviations (δ) i.e.

$$d(w) = \max \delta \left(a_{ij}, w\right) \tag{2.21}$$

where $i, j \in \{1, 2, ...n\}$ and $\sum_i w_i = 1$.

This can be called a family of *worst error minimization* methods. Several such objective functions were proposed by applying different combinations of the weighted, logarithmic, absolute and squared errors. The FPP method was considered similar to the minimization of *weighted worst absolute error* in [50], however, the concept of satisfaction index is quite different from the worst absolute error. Therefore, the suggested similarity would appear to be a misjudgment.

2.3.3 Other Methods

There have been several other methods proposed in literature; for completeness we list them below.

Laininen and Hämäläinen [69] analyzed this problem using the theory of *linear regression*. A *robust regression analysis* was proposed that differs from the LLS method only in the presence of an outlying judgment (*outlier*).

Sugihara et al. [70] proposed to use an interval approach for prioritization, even if the judgments are given as crisp values. For this purpose, an approach was proposed to obtain interval priorities from a PCM, termed as *possibilistic AHP for crisp data*.

Hartvigsen [71] demonstrated that the well-known methods may violate the *order of preference* (to be discussed in section 2.4.3). He then proposed an LLAV-based method, termed the *strength-direction* method (SDM), in order to address the highlighted discrepancy in other methods.

Srdjevic [52] suggested a combination of different methods in specific relation to AHP. He considered AN, EV, WLS, LLS, LGP and FPP to be the most popular methods. One of these methods should be chosen for each PCM, depending on the various suggested criteria.

Ramanathan [72] derived weights using the *data envelopment analysis* and proved its correctness in case of having consistent PCM. It was further suggested that verification of inconsistent matrices is impossible, as there is no consensus as to a unique correct answer. Wang et al. [73] pointed out drawbacks in this approach, including the rank reversal issue and it being irrational for highly inconsistent PCM. They further suggested an improvement to this approach that they called "*generating the most favorable weights*".

A recent trend has been to use multi-objective optimization. Mikhailov [2] proposed the simultaneous optimization of *two objectives for prioritization* (TOP). An EMO technique has recently been proposed for this prioritization method [31]. The two selected objectives, *total deviation* and *priority violations,* are the well-known evaluation criteria used to compare various methods [52, 58]. We introduce these evaluation criteria before discussing the TOP method further in section 2.5.

2.4 Evaluation Criteria for Prioritization Methods

Prioritization methods have been extensively discussed and compared in the last couple of decades. However, there appears to be no consensus on the evaluation criteria used for such comparisons. The most widely-used criteria are mentioned below.

2.4.1 Quadratic Deviation

The quadratic or total deviation (TD) can be defined as the sum of all quadratic errors (residuals) between the estimated priority vector and the set of given judgments. The calculation for TD is shown mathematically in (2.22).

$$TD(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_{ij} - \frac{w_i}{w_j} \right)^2 \tag{2.22}$$

This criterion has been widely used to compare prioritization methods [51, 52, 58, 74]. To remove its dependence on n, TD can also be normalized as Euclidean distance (ED) (as used in [75]).

2.4.2 Mean Absolute Deviation

The *mean absolute deviation* (MAD) can be defined as the average of all absolute errors (residuals) between the estimated priority vector w and the ideal preference vector r, given as:-

$$MAD(w) = \frac{1}{n} \sum_{j} |w_j - r_j| \tag{2.23}$$

MAD is not applicable to real-world situations, as the preference vector r is unknown to a DM. This criterion has been used in simulation-based experiments [50], where the PCM is generated from an ideal vector, r, and then adding perturbation to it.

2.4.3 Priority Violations

The concept of priority violation was introduced by Ali et al. [38], formulating it for the tournament ranking problem. Golany and Kress [58] used this criterion to compare prioritization methods.

When E_i is preferred to E_j, it is assumed that the priorities of these two elements should preserve the preference direction i.e. $w_i > w_j$. However, while eliciting preferences, if E_j receives a larger priority weight i.e. $w_i < w_j$, then a *priority violation* occurs. Considering the ratio judgments, it can be formulated as a logarithmic test. $v_{ij} = step\left(\log a_{ij} \log \frac{w_j}{w_i}\right)$, where the $step$ function returns 1 for positive values and 0 otherwise.

To incorporate a *preference equivalence*, the violation v_{ij} can be re-defined as (2.24) by introducing the concept of half violation [58] i.e.

$$v_{ij} = \begin{cases} 1 & if\,(w_i < w_j)\,and\,(a_{ij} > 1) \\ \frac{1}{2} & if\,(w_i \neq w_j)\,and\,(a_{ij} = 1) \\ & or\,(w_i = w_j)\,and\,(a_{ij} \neq 1) \\ 0, & otherwise \end{cases} \tag{2.24}$$

The total number of violations, $NV(w)$, is a simple aggregation of all v_{ij}:-

$$NV(w) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} v_{ij} \tag{2.25}$$

A priority vector, w^*, gives an *ideal ranking* when it satisfies all preference directions expressed by the ordinal comparisons, giving no priority violation. In order to illustrate this, consider the following PCM:-

$$A = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & 8 \\ 2 & 1 & 3 & 1\frac{1}{2} \\ \frac{2}{5} & \frac{1}{3} & 1 & 2\frac{1}{2} \\ \frac{1}{8} & \frac{2}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

The priority vectors, w, estimated for A by different methods are listed in Table 2.2, along with the values of $TD(w)$ and $NV(w)$. The EV, GM(LLS) and DLS methods have produced solutions with $NV = 1$ whilst the LLAV and WLS methods have produced *ideal ranking* (i.e. $NV = 0$).

Method	w_1	w_2	w_3	w_4	TD	NV
EV	**0.3777**	0.3759	0.1468	0.0996	1.4473	1
DLS	**0.5620**	0.1845	0.1802	0.0732	0.8281	1
LLS	**0.3786**	0.3687	0.1618	0.0910	1.3955	1
LLAV	0.3623	**0.4348**	0.1449	0.0580	1.8187	**0**
WLS	0.3247	**0.4655**	0.1563	0.0536	2.1623	**0**
GM	**0.3786**	0.3687	0.1618	0.0910	1.3955	1

Table 2.2: An example of different rankings obtained using different methods

The concept of priority violations has been considered important in PC literature. Chandran et al. [76] considered priority violations to be important whilst defining the two desirable properties i.e. *element* and *row dominance*. D'Apuzzo et al. [77] also discussed priority violations and the need to obtain a *"coherent priority vector"* i.e. a vector giving *ideal ranking* (w^*).

2.4.4 Other Criteria

There are several other criteria that have been used to compare prioritization methods e.g. conformity, computation complexity, uniqueness of solutions and sensitivity to changes in the original judgments [51, 52, 58]. All these criteria have been considered to be of less significance than the criteria discussed above.

2.5 Two-Objective Prioritization

It should be noted that most prioritization methods are based on single-objective optimization. If TD is the only consideration for ranking prioritization methods, it is obvious that the DLS method will outperform all other methods. However, in practice, minimizing TD produces a solution with a greater number of priority violations (NV).

Mikhailov [2] highlighted this issue and introduced Two-Objective Prioritization (TOP) to optimize both TD and NV, and then an EMO technique (called PESA-II) was used for this purpose [31]. The main benefit of TOP is that it can generate multiple non-dominated (Pareto-optimal) solutions for DMs to interactively select the most appropriate according to their requirements.

Consider an example of school selection given in [42]. In this example, six criteria were selected to compare schools i.e. learning, friends, school life,

Method	$w_{learning}$	$w_{friends}$	w_{life}	$w_{training}$	w_{prep}	w_{music}	***TD***	***NV***
EV	.3207	.1395	.0347	.1285	.2374	.1391	**1.7083**	**0**
GM	.3159	.1391	.0360	.1251	.2360	.1477	**1.6630**	**0**
DLS	.1643	.2363	.0369	.1461	.2088	.2075	**1.3624**	**4**
WLS	.4387	.0903	.0329	.0928	.2214	.1238	**2.3698**	**1**
LLAV	.2380	.1985	.0332	.1658	.1658	.1986	**1.4757**	**2**
NCS	.1221	.0851	.1402	.4434	.1419	.0674	**2.1920**	**6**
FPP	.2767	.1165	.0692	.1614	.2149	.1612	**1.7011**	**2**
TOP	.1643	.2363	.0369	.1461	.2088	.2075	**1.3624**	**4**
TOP	.1652	.2220	.0370	.1466	.2220	.2070	**1.3652**	**3**
TOP	.1836	.2228	.0372	.1499	.2228	.1836	**1.3721**	**2**
TOP	.1987	.1987	.0374	.1519	.2144	.1987	**1.3827**	**1**
TOP	.2027	.2027	.0374	.1518	.2027	.2027	**1.3841**	**0**

Table 2.3: Preference weights estimated from A_{school} by different methods

vocational training, college preparation and music classes. In order to give preference weights to these criteria, the following judgments were acquired (as a PCM):-

$$A_{school} = \begin{bmatrix} 1 & 4 & 3 & 1 & 3 & 4 \\ \frac{1}{4} & 1 & 7 & 3 & \frac{1}{5} & 1 \\ \frac{1}{3} & \frac{1}{7} & 1 & \frac{1}{5} & \frac{1}{5} & \frac{1}{6} \\ 1 & \frac{1}{3} & 5 & 1 & 1 & \frac{1}{3} \\ \frac{1}{3} & 5 & 5 & 1 & 1 & 3 \\ \frac{1}{4} & 1 & 6 & 3 & \frac{1}{3} & 1 \end{bmatrix}$$

The priority weight vector, w, can be estimated using different methods as discussed above. The TOP method can also be applied in this example to generate multiple non-dominated solutions. The weights generated by different methods are listed in Table 2.3. Considering TD, DLS has outperformed other methods with TD=1.3624. However, it has produced four priority violations (NV=4). TOP suggests a minimum value of TD=1.3841 which is possible without generating any violation (see the last row of the results).

The results are also plotted in Figure 2.6 to highlight the Pareto-optimal front obtained using TOP, considering the two criteria for comparing prioritization methods i.e. $TD(w)$ and $NV(w)$. The plot confirms that the other methods lie away from the optimal front.

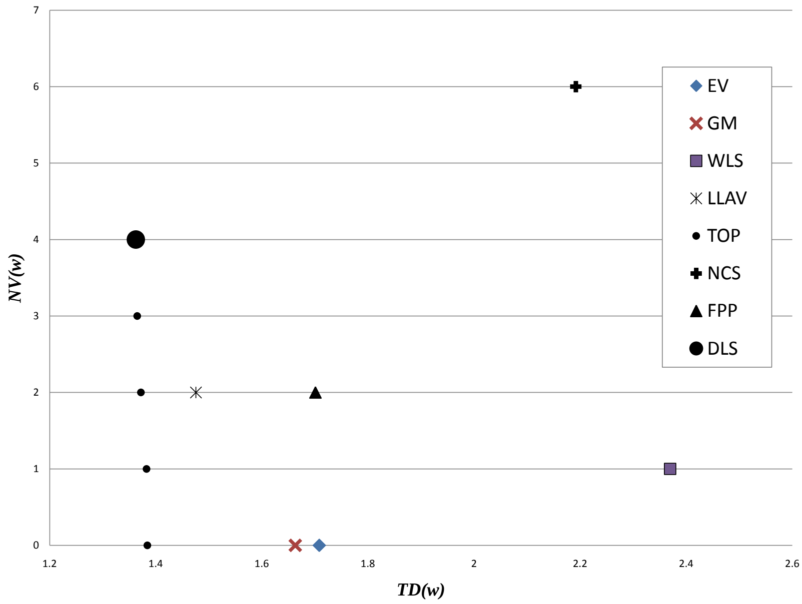


Figure 2.6: Weights shown in objective space (TD vs NV)

2.6 Inconsistency in Judgments

A complete set of judgments in the PC method creates an opportunity to have inconsistent information, primarily due to the redundancy inherent in its structure. One may argue to try to avoid redundancies present in data, however, redundancy proves quite useful in validating information acquired from not-so-perfect human minds.

There are several causes of inconsistency including psychological reasons, clerical errors, and an insufficient model structure [78]. The psychological reasons further break down into factors such as incomplete information, uncertainty and/or lack of concentration during the judgment process.

The judgment scale of 1 to 9 [17] is an appropriate example of an insufficient model structure. Consider that if A is 5 times better than B and B is 3 times better than C, then the consistent value of 15 for comparing A to C is not allowed when using the 1 to 9 scale.

There are situations where inconsistencies are unavoidable e.g. a tournament ranking problem involving upsets, where A beats B and B beats C, but then A loses to C (*intransitivity*). In this situation, ranking is required without amending any data, as teams cannot be asked to play again. Sugden [78] suggested such inconsistencies be considered as rational.

Another possible cause of inconsistency is incorrect data entry e.g. entering a reciprocal value into a PCM. This happens when a ratio for E_i to E_j is mistakenly provided at a transpose location in a PCM i.e. a_{ji} instead of a_{ij}. Lipovetsky and Conklin [79] termed such errors as *Unusual and False Observations* (UFO).

The issue of inconsistency in PCs has been discussed by many authors e.g. [1, 69, 80–83].

2.6.1 Types of Consistency

Consistency in PCs is generally of two types i.e. *cardinal consistency* (CC) and *ordinal consistency* (OC). The two types are discussed below.

2.6.1.1 Cardinal Consistency

The judgments of DMs are *cardinally consistent*, if the following conditions are met [17]:-

- $a_{ij} = \frac{1}{a_{ji}}$ for all i and j;
- $a_{ij} = a_{ik}a_{kj}$ for all i, j and k

When a DM is perfectly consistent in his/her judgments, then the priority vector w is exactly the same as the preference vector r, and the judgments a_{ij} have perfect values $a_{ij} = \frac{r_i}{r_j} = \frac{w_i}{w_j}$. In such a case, A is said to be (perfectly) *consistent* and can be represented as $A = \left[\frac{w_i}{w_j}\right]$. If the DM's judgments are *cardinally inconsistent* (i.e. $a_{ij} \neq a_{ik}a_{kj}$ for some i, j, k) then the corresponding comparison matrix A is said to be inconsistent. In this case, $a_{ij} \approx \frac{w_i}{w_j}$ and the estimated priority vector w approximates the unknown preference vector r.

2.6.1.2 Ordinal Consistency (Transitivity)

OC, which is also known as the *transitivity condition* between 3 elements, states that if E_i is preferred to E_j and E_j is preferred to E_k, then E_i should be preferred to E_k. Using the preference symbol $\rightarrow$, OC is represented as:-

- If $E_i \rightarrow E_j \rightarrow E_k$ then $E_i \rightarrow E_k$

The preference judgments are ordinally inconsistent (or *intransitive*) if $E_k \rightarrow E_i$ when $E_i \rightarrow E_j \rightarrow E_k$. Therefore, ordinal inconsistency can be defined as $E_i \rightarrow E_j \rightarrow E_k \rightarrow E_i$, which represents a *circular triad* of preferences [84], or a *three-way cycle*.

The preference relation $E_i \rightarrow E_j$ means that comparison judgment between those two elements satisfies the inequality $a_{ij} > 1$. Therefore, the ordinal inconsistency $E_i \rightarrow E_j \rightarrow E_k \rightarrow E_i$ means that the corresponding judgments are $a_{ij} > 1$, $a_{jk} > 1$ and $a_{ki} > 1$. The preference dominance is easy to depict by means of a *directed graph*, where the intransitive relationship $E_i \rightarrow E_j \rightarrow E_k \rightarrow E_i$ is represented as a cycle between those three elements [85].

When two comparison elements E_i and E_j are equally preferred, $E_i \sim E_j$ (*preference equivalence* or a *tie*), then the OC requires one of the following three conditions to be satisfied: $E_i \rightarrow E_k$ and $E_j \rightarrow E_k$, or $E_i \leftarrow E_k$ and $E_j \leftarrow E_k$, or $E_i \sim E_k$ and $E_j \sim E_k$. For example, if $a_{ij} = 1$ and $a_{ki} > 1$, then the OC demands that $a_{kj} > 1$. This implies that a judgment $a_{kj} < 1$ will introduce ordinal inconsistency. The ties present in PCMs can be shown as undirected edges between corresponding elements [81].

If the PC judgments are ordinally inconsistent, it is not possible to find a priority vector, w, that satisfies all preference directions expressed by the ordinal comparisons, and therefore, there will always be priority violations ($NV > 0$).

2.6.2 Cardinal Consistency Measures

There exist several measures proposed to assist a DM in accepting and/or updating the acquired judgments. Widely used measures discussed in PC literature are given below.

2.6.2.1 Eigenvalue-based Measures

Whilst proposing EV, Saaty [17] proposed a consistency measure based on the largest Eigenvalue for a given PCM. Using the fact that for a consistent positive reciprocal matrix, the largest Eigenvalue λ_{max} is equal to n, Saaty defined a measure of consistency, termed a Consistency Index (CI):-

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{2.26}$$

where perfect consistency implies that $CI = 0$ and $CI > 0$ for inconsistent matrices.

To define a unique consistency measure which does not depend on the dimension of the PCM, Saaty [17] further introduced the Consistency Ratio (CR)

n	[17]	[86]
3	0.58	0.5245
4	0.90	0.8815
5	1.12	1.1086
6	1.24	1.2479
7	1.32	1.3417
8	1.41	1.4056
9	1.45	1.4499

Table 2.4: Random Consistency Indices

criterion. This is the ratio between the Consistency Index and a Random Consistency Index (RI):-

$$CR = \frac{CI}{RI} \tag{2.27}$$

In the above equation, RI represents the average CI of a randomly generated PCM of the same dimension, n. The values of RI are statistically calculated from thousands (or millions) of randomly generated PCMs.

Table 2.4 shows the two sets of CR values given in [17] and [86]. Aguaron and Moreno-Jimenez [86] calculated the values of CR by increasing the number of iterations for their experiments, and therefore claiming higher precision.

Saaty [17] claimed that if the value of CR is smaller than or equal to 0.1, the estimated priority vector w can adequately approximate the unknown preference vector r. Therefore, the PCM is considered to be *acceptable* when the threshold of $CR \leqslant 0.1$ is met. However, if $CR > 0.1$, the estimated priorities could be erroneous, and DMs should be asked to improve the consistency by revising their subjective judgments. This threshold is used in various applications to accept or reject a PCM [19].

Similarly to CR, Alonso and Lamata [87] presented a statistical criterion for accepting/rejecting a PCM based on the threshold values for λ_{max} against each value of n.

2.6.2.2 Distance-based Measures

Chu et al. [61] used the mean square error (residual) as a measure of inconsistency whilst proposing WLS. Similarly, Crawford and Williams [88] proposed the *logarithmic residual mean square* (LRMS) as a natural measure of consistency, whilst proposing the geometric mean (GM) method. Aguaron and Moreno-Jimenez [86] formalized LRMS and termed it the *Geometric Consistency*

Index (GCI). This can be formulated as:-

$$GCI = \frac{2}{n\left(n-1\right)}\sum_{i=1}^{n-1}\sum_{j>i}^{n}\left(\log\left(a_{ij}\right)-\log\left(\frac{w_i}{w_j}\right)\right)^2 \tag{2.28}$$

where w is a priority vector estimated using the GM method.

The relationship between GCI and CR was found almost linear for $CR \leqslant 0.1$. Aguaron and Moreno-Jimenez [86] compared GCI to CR and proposed threshold values for GCI equivalent to CR. These values are $GCI = 0.31$ for $n = 3$, $GCI = 0.35$ for $n = 4$ and $GCI = 0.37$ for $n > 4$.

2.6.2.3 Measuring the Worst Error

Koczkodaj [34] described CR as a global measure that lacks a proper justification for its use. He proposed a measure, termed *Consistency Measure* (CM), in part to eliminate the drawbacks of CR, such as the inability to identify inconsistent judgments and its arbitrary threshold value of $CR \leqslant 0.1$.

CM is calculated by considering a set of three judgments (*triple*) from J at a time. The value of CM is chosen from the most inconsistent *triple* amongst all possible combinations i.e.

$$CM = \max_{i\neq j\neq k\neq i}\left(\overline{CM}_{i,j,k}\right) \tag{2.29}$$

where $\overline{CM}_{i,j,k}$ is the inconsistency for the triple $\{a_{ij}, a_{jk}, a_{ik}\}$ and is calculated using the following formula:-

$$\overline{CM}_{i,j,k} = \min\left(\frac{|a_{ij}-a_{ik}a_{kj}|}{a_{ij}}, \frac{|a_{ij}-a_{ik}a_{kj}|}{a_{ik}a_{kj}}\right) \tag{2.30}$$

In order to explain this, consider the following PCM with $n = 4$:-

$$A = \begin{bmatrix} 1 & 2 & 5 & 7 \\ \frac{1}{2} & 1 & 3 & 6 \\ \frac{1}{5} & \frac{1}{3} & 1 & 4 \\ \frac{1}{7} & \frac{1}{6} & \frac{1}{4} & 1 \end{bmatrix}$$

There exists four unique triples in A as given below:-

$$\begin{aligned}
\{a_{12}, a_{23}, a_{13}\} &\Rightarrow \{2, 3, 5\} \\
\{a_{12}, a_{24}, a_{14}\} &\Rightarrow \{2, 6, 7\} \\
\{a_{13}, a_{34}, a_{14}\} &\Rightarrow \{5, 4, 7\} \\
\{a_{23}, a_{34}, a_{24}\} &\Rightarrow \{3, 4, 6\}
\end{aligned}$$

The values of $\overline{CM}$ for each triple can be calculated using (2.30) as follows:-

$$\begin{aligned}
\overline{CM}_{1,2,3} &\Rightarrow \min(0.2, 0.167) = 0.167 \\
\overline{CM}_{1,2,4} &\Rightarrow \min(0.714, 0.417) = 0.417 \\
\overline{CM}_{1,3,4} &\Rightarrow \min(0.65, 1.857) = 0.65 \\
\overline{CM}_{2,3,4} &\Rightarrow \min(0.5, 1.0) = 0.5
\end{aligned}$$

The maximum value in these four triples is equal to 0.65, therefore, $CM = \max\left(\overline{CM}\right) = 0.65$. The most inconsistent set of judgments in this PCM is found to be $\{a_{13} = 5,\ a_{34} = 4,\ a_{14} = 7\}$.

2.6.2.4 Other measures

Although not widely used, there exist other measures for CC in PC judgments. For example, Dodd et al. [89] criticized the threshold of 10% ($CR \leqslant 0.1$) and proposed an alternative criterion based on *confidence level* that depends on the criticality of a decision. They claimed this new measure to be more flexible for DMs. Whilst proposing FPP, Mikhailov [68] also suggested using the satisfaction index, μ, as the natural consistency indicator.

2.6.3 Critique

Although widely used, CR has been much debated for its two major drawbacks i.e. sensitivity to scale [90, 91] and the threshold value of $CR \leqslant 0.1$ [82, 92].

CM has been shown to be useful for improving consistency by revising judgments [93]. However, similarly to the CR index, this measure cannot capture the ordinal inconsistency of the comparison judgments.

Bozoki and Rapcsak [92] compared the CR and CM with a conclusion that no single measure is adequate in all situations. They also questioned CR for using randomly generated PCMs as the reference point, and considered this reference inappropriate for analyzing human-based decisions.

From their definition, it can be seen that these measures of consistency (CR and CM) do not take into account the ordinal inconsistency (or intransitivity) of the PCM. This is also true for GCI, having a linear relationship with CR [86].

Generally, if the comparison matrices are ordinally consistent, most prioritization methods derive priorities with the same ranking, only with different intensities. If, however, the matrices are ordinally inconsistent (*intransitive*), there exists no priority vector satisfying all contradictory preferences. Therefore, different prioritization methods provide different ordinal rankings that partially correspond to the ordinal comparison judgments.

2.6.4 Measuring Ordinal Consistency

Ordinal inconsistency always implies cardinal inconsistency, however, the converse does not hold. Saaty's CR measures the cardinal inconsistency of the judgments, but does not capture their ordinal inconsistency (this is also true for other cardinal indexes, such as GCI and CM).

Generally, if a PCM is ordinally inconsistent, the value of its CR remains above 0.1; therefore satisfying the CR test may significantly reduce the chances of ordinal inconsistency. However, there are examples in the literature where matrices that satisfy the CR criterion can also be ordinarily inconsistent [81, 94].

2.6.4.1 Kendall's ζ

In order to measure ordinal inconsistency, Kendall [85] introduced an ordinal Coefficient of Consistence, ζ, for comparisons between elements with no preference equivalencies. Kendall's ζ is calculated using the following equation:-

$$\zeta = 1 - \frac{L}{L_{max}} \tag{2.31}$$

where L is the number of three-way cycles present in given PCM, and L_{max} is the maximum number of three-way cycles possible. The value of L_{max} is $^{(n^3-n)}/_{24}$ for odd values of n, and $^{(n^3-4n)}/_{24}$ when n is even, formulated in [85].

Algorithm 1 Detecting a three-way cycle in PCM

```
FOR ALL (i, j, k) from 1 to n WHERE (i ≠ j ≠ k ≠ i)
  IF log(a_ij)log(a_ik) ≤ 0 AND log(a_ik)log(a_jk) < 0 THEN
    STOP
  ELSE IF log(a_ij) = 0 AND log(a_ik) = 0 AND log(a_jk) ≠ 0 THEN
    STOP
  ELSE
    OK
END-FOR
```

Gass [95] formulated the prioritization problem for PCs as a tournament ranking problem and calculated the total number of cyclic judgments as:-

$$L = \frac{n(n-1)(2n-1)}{12} - \frac{1}{2}\sum_{i} s_i^2 \tag{2.32}$$

where s_i is the total number of wins claimed by E_i (also termed the *out-degree*). It should be noted that this definition of L does not support *preference equivalence*.

In order to include *preference equivalence*, Jensen and Hicks [81] extended Kendall's work and proposed the generalized coefficient of consistence. They suggested it be used as a supplement to CR, in the presence of intransitive judgments. The number of three-way cycles considering *preference equivalence* are always greater than L for obvious reasons.

Kwiesielewicz and van Uden [94] presented a simple yet effective algorithm to test the presence of any contradictory set of judgments, which is sensitive to the presence of *preference equivalence* as well. This is reproduced here as Algorithm 1. This algorithm can be modified to recalculate L with *preference equivalence* by counting all the intransitive sets of judgments. This proposed modification is given as Algorithm 2 here.

Iida [96] proposed an OC test based on Kendall's ζ to apportion the DM's inconsistency i.e. whether a DM is sufficiently capable of making judgments.

2.6.5 Proposed Improvements for Consistency

Different techniques have been introduced to detect, improve and/or adjust the inconsistency, either by using a local operator or a matrix-based global operation.

Algorithm 2 Calculating L using enumeration

SET $L = 0$
FOR EACH UNIQUE COMBINATION OF i, j, k WHERE $i, j, k \in \{1, 2, ..., n\}$
  IF $log(a_{ij})log(a_{ik}) \leq 0$ AND $log(a_{ik})log(a_{jk}) < 0$ THEN
    REGISTER the combination of i, j, k as a three-way cycle
    INCREMENT L by 1
  ELSE IF $log(a_{ij}) = 0$ AND $log(a_{ik}) = 0$ AND $log(a_{jk}) \neq 0$ THEN
    REGISTER the combination of i, j, k as a three-way cycle
    INCREMENT L by 1
END-FOR

Zeshui and Cuiping [97] proposed to reduce the CR of a given PCM using an iterative algorithm based on matrix algebra. However, the original information is not retained in the derived PCM when applying this method. Cao et al. [98] insisted on retaining the originally provided judgments as much as possible, thereby proposing a heuristic approach to improve CR. As both approaches involve matrix-based global operations, the later approach also tends to change all the elements of a given PCM.

Holsztynski and Koczkodaj [99] introduced a method to detect the most inconsistent judgment in a given PCM, and proposed to change only the blamed judgment. This iterative algorithm was proved to be convergent [93], however, this method does not consider OC to be improved. Consider the following PCM, for example:-

$$A = \begin{bmatrix} 1 & 2 & \frac{1}{2} & 9 \\ \frac{1}{2} & 1 & 2 & 7 \\ 2 & \frac{1}{2} & 1 & 2 \\ \frac{1}{9} & \frac{1}{7} & \frac{1}{2} & 1 \end{bmatrix} \tag{2.33}$$

with CM=0.889 and CR=0.242. The PCM is ordinally inconsistent due to $E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_1$. The method proposed by Holsztynski and Koczkodaj [99] suggests a_{14} is to be revised for inconsistency reduction. It is evident that ordinal inconsistency is caused by a_{13} and not a_{14}.

From the discussion so far, it is obvious that cardinal inconsistency does not imply ordinal inconsistency; and there exists no fine-grained measure for OC. Moreover, the existing methods for improving consistency of PCMs do not take into account ordinal inconsistency.

2.7 Summary

MCDM is a useful process in complex decision making problems and is currently considered an important and active area of research. The use of PC is becoming increasingly popular and is an integral part of AHP. There exist several prioritization methods for PCs, however, there is no method that outperforms all others in all situations.

Before prioritization, the acquired judgments are usually analyzed to check whether the inconsistency is *acceptable* or requires revision. CR, GCI and CM are the best known measures for CC, while Kendall's ζ is the best known measure to test for OC.

There exists no fine-grained measure for OC. The literature survey has identified a need in this area of research to obtain a consistency measure with an appropriate threshold that is able to capture both OC and CC and to highlight the most inconsistent judgment for possible revision. Considering this need, two new consistency measures, *congruence* and *dissonance*, are proposed in the next chapter.

Chapter 3

New Measures for Consistency

The chapter begins by describing several simulation-based statistical experiments whose purpose is to investigate the effects of intransitive judgments on the consistency of pairwise comparison (PC) matrices. The results highlight the need for new consistency measures. Two new consistency measures - termed *congruence* and *dissonance* - are then proposed to aid a decision maker (DM) in the process of elicitation.

3.1 Inconsistency and Measurement Scales

Experiments have been performed to investigate how well inconsistency in PC matrix (PCM) is measured when different scales of measurement are used. Monte-Carlo simulation has been used for this purpose.

Monte-Carlo experiments are based on repeated random sampling to compute results that are usually not achieved through a deterministic algorithm. Such methods are useful for modeling phenomena with significant uncertainty in inputs. Monte-Carlo methods have often been used previously for analyzing PCMs [75, 100–102].

PCMs were randomly generated using the sets of values given in Table 3.1. Each scale consists of equivalence (1), dominance (>1) and their reciprocal values. For each n ranging from 3 to 6, 50000 PCMs were generated by randomly picking the values from the selected scale. The cardinal consistency (CC) of each PCM was measured using Consistency Ratio (CR) while the ordinal consistency (OC) was measured with the number of three-way cycles, L.

The value of CR was calculated using λ_{max} as formulated in [42]. Each

Linear	1	2	3	4	5	6	7	8	9
Balanced	1	1.111	1.125	1.143	1.167	1.20	1.25	1.333	1.5
Geometric	1	2	4	8	16	32	64	128	256
Inverse	1	1.125	1.286	1.5	1.8	2.25	3.0	4.5	9.0
Logarithmic	1	1.585	2	2.322	2.585	2.807	3	3.17	3.322
Power	1	4	9	16	25	36	49	64	81
Root	1	1.414	1.732	2	2.236	2.449	2.646	2.828	3

* *Reciprocals for these values are not shown*

Table 3.1: The sets of values used for each scale of measurement

generated PCM was tested against the acceptability criterion of $CR \leqslant 0.1$. A PCM was declared *acceptable* when this criterion was met. To investigate ordinal properties, L was also calculated for each PCM with the help of Algorithm 2.

The results from these experiments are discussed below.

3.1.1 CR and Measurement Scales

The percentage of the *acceptable* PCMs found for each scale is shown in Table 3.2. Considering the *linear* scale of 1 to 9, the percentage of *acceptable* PCMs decreases with the increase in n i.e. 22.21% for $n = 3$, 3.36% for $n = 4$ and only 0.26% for $n = 5$.

The *geometric* and *power* scales are least likely to produce *acceptable* PCMs. In the performed experiments, no PCM was found *acceptable* for $n = 5$ when using the *power* scale.

For the *inverse* and *root* scales, the percentage decreases monotonically as the value of n is increased. The possibility of having an acceptable PCM remains around 60% when using the *logarithmic* scale and is invariant to n. The chances of obtaining an acceptable PCM with the *balanced* scale are higher than all the other tested scales and goes even higher for greater values of n (see Table 3.2). When PCMs were randomly generated for $n = 5$ using the *balanced* scale, all had CR below 0.1 (100%).

It is inferred from these results that a possibility of having *acceptable* PCM is highly sensitive to the selected scale. In other words, the value of CR is sensitive to the measurement scale chosen for generating PCMs.

n	*Linear*	*Balanced*	*Geometric*	*Inverse*	*Logarithmic*	*Power*	*Root*
3	22.21%	94.59%	12.97%	41.78%	67.61%	10.10%	51.72%
4	3.36%	98.88%	0.47%	18.37%	58.71%	0.29%	29.91%
5	0.26%	99.93%	0.01%	7.34%	58.15%	0.00%	14.64%

Table 3.2: Percentage of *acceptable* PCMs in randomly generated matrices

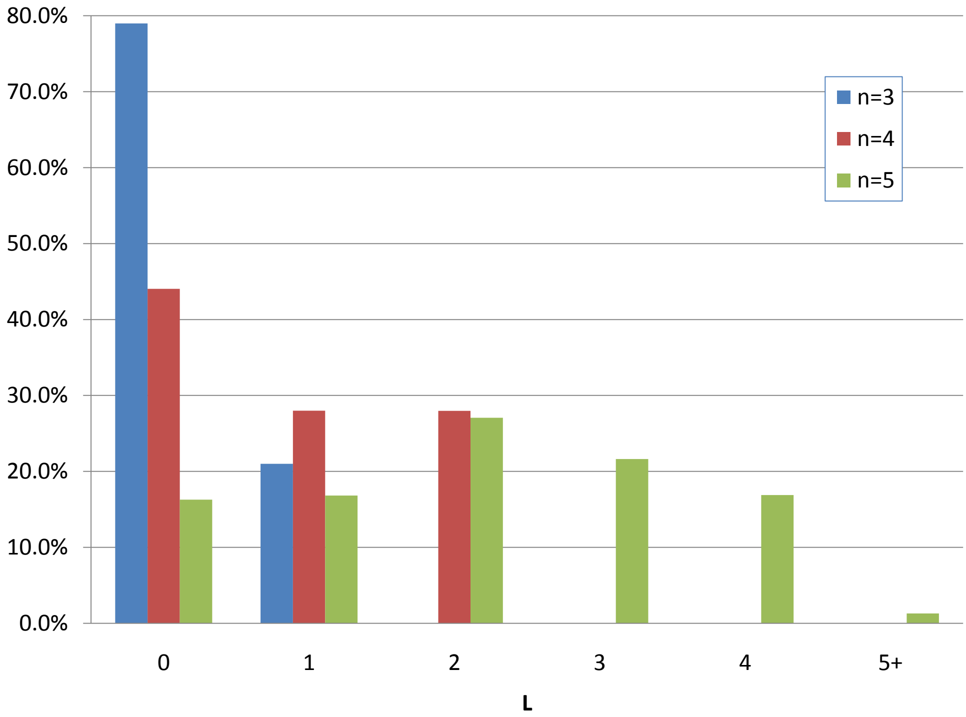


Figure 3.1: Distribution of generated PCMs against L

3.1.2 L and Measurement Scales

Fig. 3.1 shows the distribution of the PCMs generated with different number of three-way cycles.

A PCM is called *transitive* when $L = 0$ and *intransitive* when $L \neq 0$. The percentage of transitive PCMs generated for $n = 3$ was found higher (i.e. 79%) than the transitive PCMs generated for $n = 4$ (44%) and $n = 5$ (16.3%).

As only one three-way cycle is possible for $n = 3$, all the *intransitive* PCMs lie on $L = 1$. However, for $n > 3$, the *intransitive* PCMs get distributed over different values of L. Fig. 3.1 suggests that the *intransitive* PCMs with $L = 1$ and $L = 2$ are equally likely for $n = 4$. In the case of $n = 5$, it is more likely to obtain *intransitive* PCMs with $L = 2$.

It is highlighted here that all the scales generated similar results for this

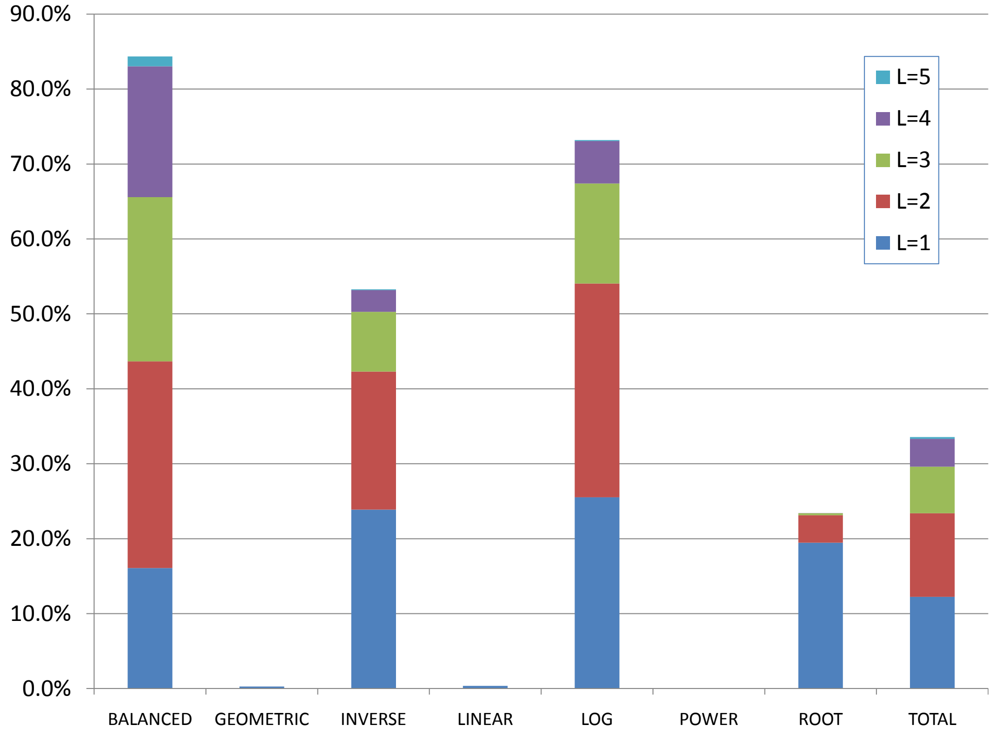


Figure 3.2: Percentage of IAPs found in *acceptable* PCMs ($n = 5$)

analysis. This is true for all the ratio-based scales, due to the following reasons:-

1. All the tests for L are based on the threshold of unity i.e. testing $a_{ij} \geqslant 1$ or $a_{ij} \leqslant 1$.
2. Each scale has an equal number of values above and below the threshold of unity (see Table 3.1).
3. When using uniform distribution, the probability of obtaining $a_{ij} > 1$ remains equal to the probability of obtaining $a_{ij} < 1$, regardless of the scale used.

3.2 Cardinal Measures for Ordinal Inconsistency

In order to investigate the use of CR for measuring OC, 15000 *acceptable* PCMs were generated for each scale. These matrices were then tested whether *transitive* ($L = 0$) or *intransitive* ($L \neq 0$). The *intransitive acceptable* PCMs (IAPs) were then counted for each scale. The results for $n = 5$ are shown in Fig. 3.2. Similar results were found for other values of n.

The percentage of IAPs was found very low for the *linear* and *geometric* scales (below 1%). No IAP was generated for the *power* scale. The *balanced* scale produced a very high percentage of IAPs, followed by the *logarithmic* scale. When the *inverse* scale was used, more than half (53.3%) of the generated PCMs were found to be *intransitive*. Similarly, 23.4% of *acceptable* PCMs were found to be *intransitive* when using the *root* scale.

From the results given above, it is concluded that *intransitive* PCMs do not always get rejected by Saaty's criterion of $CR \leqslant 0.1$. The CR criterion is found suitable for the *linear* scale of 1-to-9, however, this criterion is not equally appropriate for other measurement scales used to acquire judgments.

3.2.1 Analysis using Perturbed Judgments

The experiments discussed above have shown useful relations, however, a rational DM does not pick judgments randomly from the given pool of values. In other words, the judgment variables are inter-dependent. In order to simulate a set of human judgments, it is arguably more appropriate to construct a consistent PCM and then perturb it with different types of errors. For example, the presence of one or two outliers is usually considered to be a clerical error and can be simulated by inverting some of the judgments.

In order to simulate more human-like judgments, another set of experiments was performed starting from the generation of *ideal vectors*. Similar approaches have been used for such experiments [50, 103]. For each generated vector, r, a consistent PCM $A = \lfloor a_{ij} \rfloor$ was constructed, such that $a_{ij} = {}^{r_i}/_{r_j}$ for all $i, j = 1, 2, ..., n$. In order to introduce inconsistency, an error with uniform distribution was superimposed on each judgment in the upper triangular part of A. A distribution is called *uniform* when the probabilities of all possible outcomes are equal e.g. throwing a dice. The elements in the lower triangular part were recalculated accordingly, so that $a_{ji} = {}^{1}/_{a_{ij}}$. To increase the ordinal inconsistency (e.g. introducing clerical errors), some a_{ij} were also randomly swapped with their reciprocal elements a_{ji}. 50,000 such PCMs were generated for each n, where n was varied from 3 to 9.

The numerical results for these experiments are shown in Table 3.3. The second column of the table shows the total number of *acceptable* PCMs, k_A, found in the 50,000 randomly generated PCMs. It can be seen that k_A decreases as the matrix dimension, n, increases. On the other hand, the total number of

n	k_A	k_N	$k_{N \cap A}$
3	32937	4871	212
4	26751	13537	2098
5	23535	22582	5873
6	20351	30552	9145
7	16817	36978	10829
8	14794	41279	11647
9	12388	44299	10974

Table 3.3: The numbers for generated *acceptable* and *intransitive* PCMs

intransitive PCMs, k_N, increases considerably for higher values of n. The total number of IAPs found is denoted by $k_{N \cap A}$, and shown in the rightmost column of Table 3.3. The probability of having cycles for $n = 3$ is very small, but the results show a considerable increase in $k_{N \cap A}$ as n increases.

These results also conclude that *intransitive* PCMs are not always rejected by Saaty's criterion of $CR \leqslant 0.1$.

3.2.2 Other Cardinal Measures

The other two CC measures, Geometric Consistency Index (GCI) and Consistency Measure (CM), were also calculated against each generated *acceptable* PCM. The values of GCI were calculated using (2.28) and the values of CM were calculated using (2.29) and (2.30). The results were grouped against the values of CR from 0 to 0.1 with an increment of 0.01.

The average values of CM and GCI for $n = 5$ are shown in Fig. 3.3. The relation between CR and GCI is linear for $0 < CR < 0.1$, confirming the findings of Aguaron and Moreno-Jimenez [86]. A threshold of $GCI \simeq 0.35$ was found equivalent to $CR = 0.1$. As GCI is linearly related to CR, it is fair to comment that GCI also does not capture ordinal inconsistency in PCMs.

Fig. 3.3 shows that CM does not have a linear relationship to CR and appears to be more sensitive between $0 < CR < 0.05$. The results were found to be similar using different scales, hence, the average values in Fig. 3.3 represent all scales.

CM has the capability to locate the major cause of inconsistency by detecting the most inconsistent *triple.* However, as discussed earlier, this method does not consider how ordinal inconsistency is to be removed.

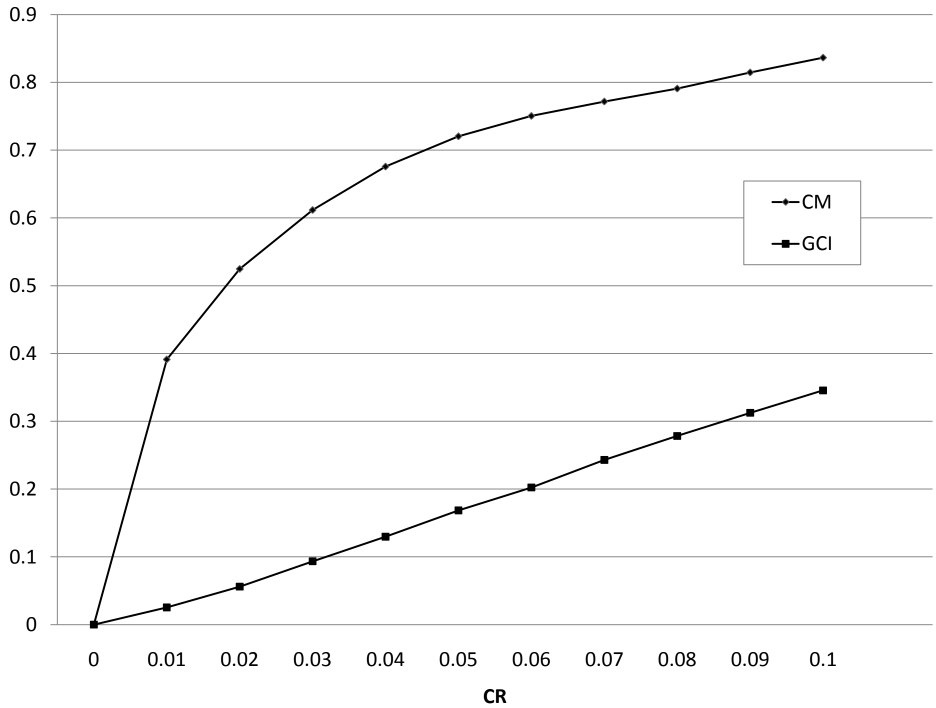


Figure 3.3: Relating CM and GCI with CR ($0 < CR < 0.1$)

3.3 Importance of Measuring Ordinal Inconsistency

So far, evidence has been collected to show that CR (and other consistency measures) does not capture ordinal inconsistency. An argument exists whether ordinal inconsistency is important enough to be measured and addressed when found. Recall: it is not possible to find a priority vector with ideal ranking when judgments are ordinally inconsistent. This is illustrated below with the help of an example.

3.3.1 Intransitive APCM: An Example

Consider a prioritization problem with five elements where a DM has provided the following PCM:-

$$A_1 = \begin{bmatrix} 1 & \frac{7}{4} & \frac{3}{4} & \frac{5}{2} & \frac{7}{4} \\ \frac{4}{7} & 1 & \frac{3}{4} & \frac{9}{4} & \frac{9}{4} \\ \frac{4}{3} & \frac{4}{3} & 1 & \frac{3}{4} & \frac{3}{4} \\ \frac{2}{5} & \frac{4}{9} & \frac{4}{3} & 1 & \frac{5}{8} \\ \frac{4}{7} & \frac{4}{9} & \frac{4}{3} & \frac{8}{5} & 1 \end{bmatrix} \tag{3.1}$$

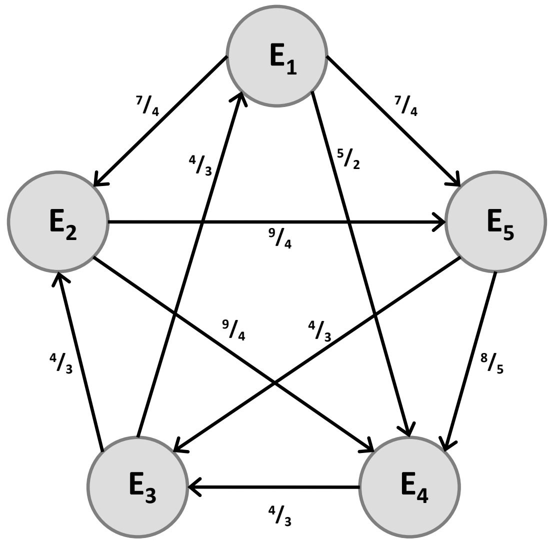


Figure 3.4: A preference graph with cyclic judgments

The CR of this PCM is equal to 0.083, which is below the threshold of 0.1, therefore the matrix can be called *acceptable* PCM. The digraph of this PCM is shown in Fig. 3.4. Each comparison element E_i is represented as a node and the judgments represent edges between the nodes. For example, the directed edge from E_1 to E_2 shows that E_1 is preferred to E_2, whereas the weight of the edge represents the intensity of the preference, which is equal to the value of the pairwise comparison judgment between those two elements, $a_{12} = \frac{7}{4}$.

It can be seen that the digraph contains four three-way cycles: $E_1 \rightarrow E_4 \rightarrow E_3 \rightarrow E_1$, $E_1 \rightarrow E_5 \rightarrow E_3 \rightarrow E_1$, $E_2 \rightarrow E_4 \rightarrow E_3 \rightarrow E_2$ and $E_2 \rightarrow E_5 \rightarrow E_3 \rightarrow E_2$. As the judgments are intransitive and there are cycles in its digraph, the matrix A_1 is ordinally inconsistent.

In matrices of dimension $n > 3$, cycles with more than three ways may exist. For example, see the graph shown in Fig. 3.4, there is a four-way cycle $E_1 \rightarrow E_5 \rightarrow E_4 \rightarrow E_3 \rightarrow E_1$. However, it is important to note that if there are no three-way cycles in the digraph, then there are no cycles involving more than three edges [95, 104].

The priority vectors w for A_1, obtained by different prioritization methods, are shown in Table 3.4. The values of $NV(w)$ for generated priority vectors are given in the last column of the table.

From the results it can be seen that the Eigenvector (EV) method, the Direct Least Squares (DLS) method and the Logarithmic Least Squares (LLS) methods

Method	w_1	w_2	w_3	w_4	w_5	$NV(w)$
EV	.269	.226	.203	.134	.169	4
DLS	.282	.239	.191	.124	.164	4
LLS	.275	.227	.194	.132	.172	4
LLAV	.303	.253	.139	.120	.186	3
MNV	.350	.286	.086	.136	.143	2

Table 3.4: Preference weights estimated from A_1 by different methods

derive priority vectors with the same ranking order $w_1 > w_2 > w_3 > w_5 > w_4$ (but different intensities). This ranking has four violations, i.e. $w_1 > w_3$, $w_2 > w_3$, $w_3 > w_4$ and $w_3 > w_5$ (as $a_{13} < 1$, $a_{23} < 1$, $a_{34} < 1$, $a_{35} < 1$). The Logarithmic Least Absolute Value (LLAV) method produces ranking with only three violations ($w_1 > w_3$, $w_2 > w_3$ and $w_3 > w_4$).

To obtain a priority vector with Minimum $NV(w)$ (MNV), an evolutionary computing optimization algorithm was applied. The results of the MNV method are shown in the last row of Table 3.4. The MNV ranking $w_1 > w_2 > w_5 > w_4 > w_3$ has only two violations - $w_1 > w_3$ and $w_2 > w_3$. Ranking with less than two violations cannot be achieved for this PCM.

For *intransitive* PCM, the MNV method can derive multiple priority vectors with different ranking, but the same value of $NV(w)$. As the focus here is to compare various prioritization methods with respect to violations, the properties of the MNV method and the problem of possible multiple solutions are not discussed further.

All possible solutions significantly violate the ordinal preferences of the DM. The PCM used in this example is of *acceptable* inconsistency as its $CR = 0.083$. Therefore by applying the CR test only, the PCM will be classified as acceptable, and the DM will not be asked to reconsider the judgments.

From those observations we may conclude that there is a need to investigate how $NV(w)$ is affected by the *intransitive* set of judgments.

3.3.2 Consistency Measures and $NV(w)$

Although there exist measures for consistency, no experiments or formulation has shown the relation between consistency measures and $NV(w)$. Next, we discuss the simulation experiments performed with the purpose of answering these questions. We present here a statistical relation between $NV(w)$ and the discussed consistency measures i.e. CR, CM, GCI, and Kendall's ζ.

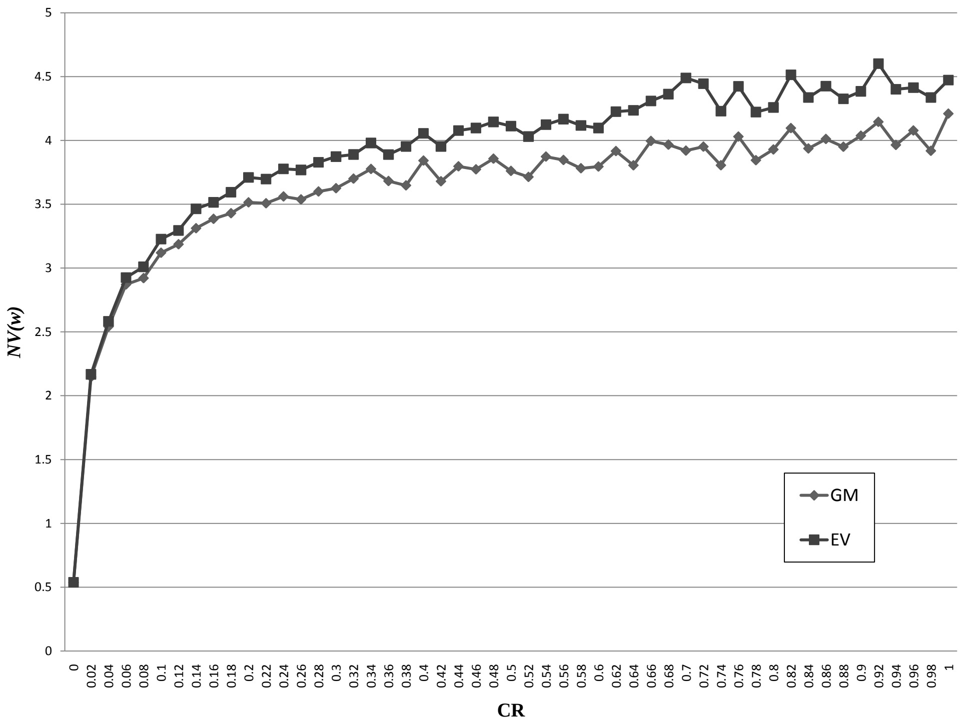


Figure 3.5: $NV(w)$ vs CR

3.3.2.1 Experiments

In order to investigate the relation between consistency measures and $NV(w)$, 50000 PCMs were generated by randomly picking the values from the scale of 1 to 9. To investigate ordinal properties, L was also calculated for each PCM with the help of Algorithm 2. The value of ζ was derived from L using (2.31). The consistency of each PCM was measured using CR, GCI, CM and ζ.

The experiments were performed for larger matrices ($n = 6$) in order to obtain a greater range of values for ζ. It should be highlighted here that the greater the value of ζ, the lower is the intransitivity in PCM. It is obvious from (2.31) that the granularity of ζ depends on L_{max}, incremented by $\frac{1}{L_{max}}$.

In order to calculate $NV(w)$, weights were estimated using both the EV and the GM methods. The values of CR, GCI, and CM are not discrete therefore the scale of values is divided into equally-spaced ranges. An average $NV(w)$ was calculated for each range of values. Figs. 3.5 to 3.8 give these results, calculated by EV and GM.

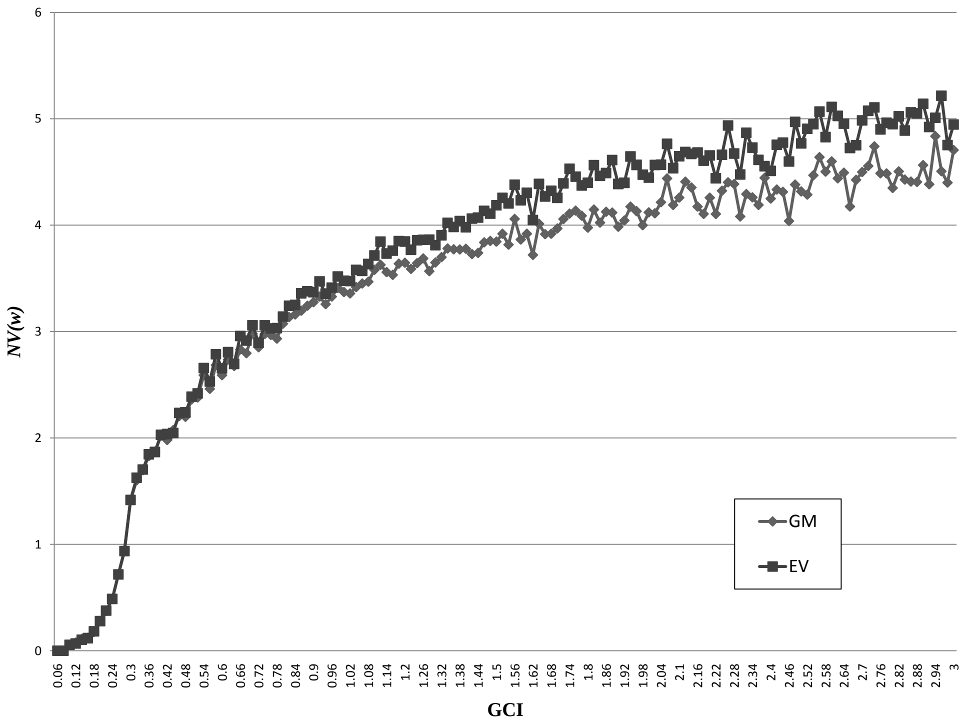


Figure 3.6: $NV(w)$ vs GCI

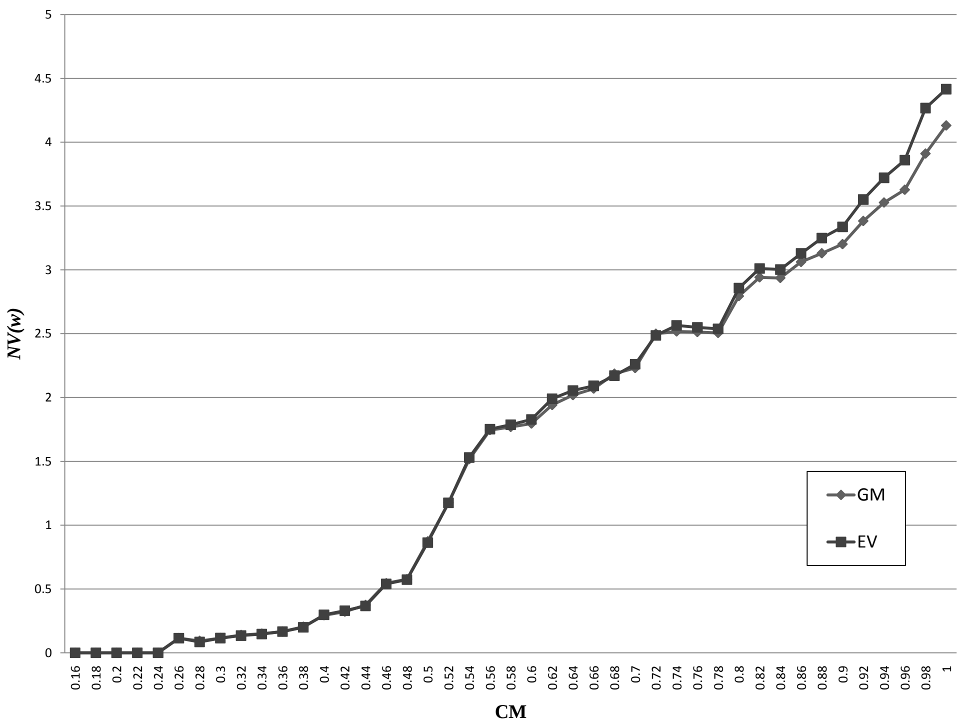


Figure 3.7: $NV(w)$ vs CM

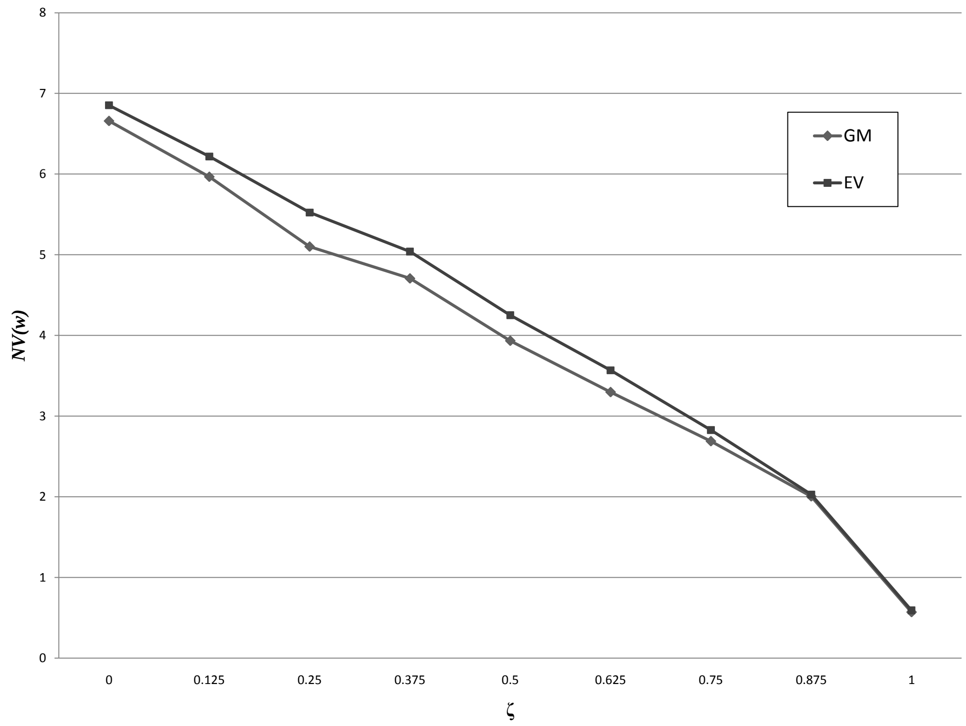


Figure 3.8: $NV(w)$ vs ζ

3.3.2.2 Results

It is evident in Fig. 3.5 that CR is sensitive in the range $CR < 0.1$, and does not easily capture inconsistencies above this value. The scale is truncated to 1.0 in order to highlight its sensitivity to $NV(w)$ around $CR = 0.1$. Fig. 3.6 shows the behavior of GCI similar to CR. The relation between CR and GCI is linear for $0 < CR < 0.2$, confirming the findings of Aguaron and Moreno-Jimenez [86]. The results for CM provide evidence that it is more effective than both CR and GCI at capturing ordinal inconsistency for $CM \gtrsim 0.5$ (see Fig. 3.7). However, CM is insensitive to $NV(w)$ for $CM \lesssim 0.5$.

The results relating ζ and $NV(w)$ are shown in Fig. 3.8. The graph in Fig. 3.8 shows that there is a linear relation between Kendall's ζ and $NV(w)$. It is concluded that the more the number of cycles (intransitivity) present, the more violations are likely to occur while estimating preferences. These results are in agreement with the findings of Jensen and Hicks [81].

This investigation clearly shows that ordinal inconsistency is the main reason for obtaining priorities that do not correspond to a DM's preferences. As the CC criteria cannot adequately identify such cases, the measurement of both

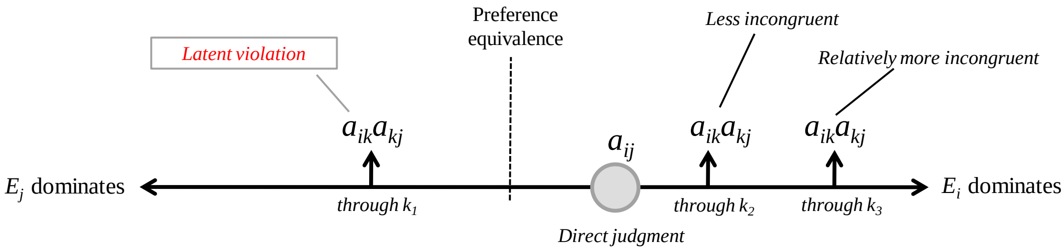


Figure 3.9: The indirect judgments ($a_{ik}a_{kj}$) on a judgment scale

cardinal and *ordinal* inconsistency may be regarded as the most important way to increase the accuracy of the decision-making process.

3.4 Congruence

Considering the CC test between E_i and E_j i.e. $a_{ij} = a_{ik}a_{kj}$ (for all i, j, k), if the DM's judgments are cardinally inconsistent, then we obtain at least one indirect judgment $a_{ik}a_{kj} \neq a_{ij}$ for some i, j and k where $i \neq k \neq j$. Fig. 3.9 illustrates this concept by locating direct and indirect judgments together on a judgment scale. Two of the indirect judgments in Fig. 3.9 suggest preference ratios higher than the direct judgment, while the third one (on the left) suggests a preference reversal (to be discussed in section 3.5). When a DM provides complete set of judgments in J, a total of $\frac{n(n-1)(n-2)}{2}$ indirect judgments can be inferred.

Considering the dispersion of indirect judgments on measurement scale, an individual deviation measure can be calculated for each judgment. These individual deviations are formulated as:-

$$\theta_{ij} = \frac{1}{n-2} \sum_{k=1}^{n} \delta(a_{ij}, a_{ik}a_{kj}) \tag{3.2}$$

where $i \neq k \neq j$ and δ is a function calculating the deviation between a direct and indirect judgment.

The set of values θ_{ij} can be used to construct a *congruence matrix* and the average of all these individual errors defines overall *congruence*:-

$$\Theta = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=1+1}^{n} \theta_{ij} \tag{3.3}$$

It is fair to comment that a single global value measuring inconsistency, whether cardinal or ordinal, is insufficient to locate all the inconsistent judgments present in a PCM. Although Koczkodaj's CM has the property of detecting the most inconsistent judgment, it fails to describe the contribution of other judgments to the overall inconsistency. It is, hence, proposed to measure inconsistency with the help of a *congruence matrix*.

A *congruence matrix* can be obtained using (3.2) with the following definition of δ:-

$$\delta\left(a_{ij}, a_{ik}a_{kj}\right) = \left|\log\left(a_{ij}\right) - \log\left(a_{ik}a_{kj}\right)\right| \tag{3.4}$$

It should be noted that the proposed definition consists of logarithmic errors generating a symmetric matrix. This removes the possibility of having different values in the upper and the lower triangles of the matrix.

Consider A_1, given in (3.1), to analyze inconsistency present among its constituent judgments. A *congruence matrix* for A_1 is calculated using (3.2) and (3.4) as follows:-

$$\theta = \begin{bmatrix} - & 0.61 & 1.06 & 0.69 & 0.69 \\ & - & 1.11 & 0.77 & 0.89 \\ & & - & 1.12 & 1.00 \\ & & & - & 0.35 \\ & & & & - \end{bmatrix}$$

where the values for self-comparison and reciprocal judgments are not shown, being redundant. The values in θ can be sorted in descending order as follows:-

θ_{34}	θ_{23}	θ_{13}	θ_{35}	θ_{25}	θ_{24}	θ_{14}	θ_{15}	θ_{12}	θ_{45}
1.12	1.11	1.06	1.00	0.89	0.77	0.69	0.69	0.61	0.35

The judgment, a_{34} is found to be most inconsistent, while a_{45} is considered the most consistent one. The two judgments a_{14} and a_{15} are found to be equally inconsistent ($\theta = 0.69$).

3.4.1 Outlier Detection

The set of *congruence* values, θ_{ij}, can be used to possibly spot the outlier with the highest deviation value. In order to identify an outlier with the help of *congruence*, consider the following example:-

$$A_2 = \begin{bmatrix} 1 & 2 & 4 & 8 & \frac{1}{3} \\ \frac{1}{2} & 1 & 2 & 4 & 8 \\ \frac{1}{4} & \frac{1}{2} & 1 & 2 & 4 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 & 2 \\ 3 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

The *congruence matrix* for A_2 can be calculated using (3.2) and (3.4) as follows:-

$$\theta = \begin{bmatrix} - & 1.29 & 1.29 & 1.29 & 3.87 \\ & - & 0 & 0 & 1.29 \\ & & - & 0 & 1.29 \\ & & & - & 1.29 \\ & & & & - \end{bmatrix}$$

The value of $\theta_{15} = 3.87$ shows that a_{15} is the least congruent with other judgments. The three judgments of a_{23}, a_{34} and a_{24} are found to be perfectly consistent. All the other judgments make an equal contribution to the overall inconsistency. As there exists a single judgment that shows the highest deviation, this suggests the presence of an outlier or UFO i.e. a_{15}.

3.4.2 Consistency Deadlock

Consider another example of inconsistent PCM as follows:-

$$A_3 = \begin{bmatrix} 1 & 2 & 2 & 2 & 2 \\ \frac{1}{2} & 1 & 2 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & 1 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$

The *congruence matrix* for A_3 is as follows:-

$$\theta = \begin{bmatrix} - & .69 & .69 & .69 & .69 \\ & - & .69 & .69 & .69 \\ & & - & .69 & .69 \\ & & & - & .69 \\ & & & & - \end{bmatrix}$$

This tells a completely different story: all given judgments are found to be equally inconsistent. This type of situation can be termed "*consistency deadlock*" as no improvement can be suggested. A single global measure of consistency cannot not describe the presence of consistency deadlock. In such cases, any of the provided judgments can be selected for revision.

3.5 Dissonance

The concept of indirect judgments can also be used to calculate OC, based on the ordinal violation between a given judgment and the corresponding indirect judgments. The preference relation $C_i \rightarrow C_j$ (implying $a_{ij} > 1$) should enforce $a_{ik}a_{kj} > 1$ for all $k = 1, 2, ..., n$. Therefore, whenever an indirect judgment contradicts this preference relation, a *latent violation* occurs. The concept of latent violation can be visualized in Fig. 3.9, showing a violation of a_{ij} by an indirect judgment through k_1. This is mathematically formulated as:-

$$\psi_{ij} = \frac{1}{(n-2)} \sum_{k} step\left(-\log a_{ij} \log a_{ik}a_{kj}\right) \tag{3.5}$$

where $i \neq k \neq j$ and the *step* function is defined as:-

$$step(x) = \begin{cases} 1 & if\, x > 0 \\ 0 & otherwise \end{cases}$$

The set of values ψ_{ij} construct a *dissonance matrix* that can supplement the *congruence matrix* in order to spot the most inconsistent judgments. The term proposed here relates to the notion of *cognitive dissonance* i.e. the holding of two contradictory beliefs simultaneously.

It is proposed to calculate OC as the average of all the elements in $[\psi_{ij}]$ as *overall dissonance* (Ψ):-

$$\Psi = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \psi_{ij} \tag{3.6}$$

The average is proposed instead of the aggregation to remove its dependence on n.

Although the values of CR, CM and Θ measure the level of inconsistency in judgments, Ψ uniquely measures ordinal inconsistency (as with Kendall's ζ), suggesting a number of possible preference reversals. This *dissonance* can be used to detect priority violations, as illustrated through examples in the next section.

The calculation of *congruence matrix*, $[\theta_{ij}]$, has been useful to analyze *cardinal* inconsistency. The use of *dissonance matrix*, $[\psi_{ij}]$, can further supplement the visualization of *ordinal* inconsistency. This is explained here with the help of examples already discussed above.

Considering the intransitive matrix A_1 again, the *dissonance matrix* is obtained using (3.5) as follows:-

$$\psi = \begin{bmatrix} - & 0.33 & 1.00 & 0.33 & 0.33 \\ & - & 0.67 & 0.33 & 0.33 \\ & & - & 1.00 & 0.67 \\ & & & - & 0.35 \\ & & & & - \end{bmatrix}$$

where transpose elements are not shown, being redundant.

The value of $\psi_{34} = 1$ suggests a_{34} to be completely dissonant with all the related judgments in A_1. The judgment of a_{34}, which was declared to be the most inconsistent, has also been found to be the most dissonant.

Consider another interesting example where the most inconsistent and the most dissonant judgments are different, as follows:-

$$A_4 = \begin{bmatrix} 1 & 9 & \frac{3}{4} & \frac{3}{4} & 9 \\ \frac{1}{9} & 1 & \frac{5}{4} & \frac{3}{4} & 4 \\ \frac{4}{3} & \frac{4}{5} & 1 & 2 & \frac{4}{3} \\ \frac{4}{3} & \frac{4}{3} & \frac{1}{2} & 1 & 3 \\ \frac{1}{9} & \frac{1}{4} & \frac{3}{4} & \frac{1}{3} & 1 \end{bmatrix}$$

The two consistency measures, calculated for each judgment in A_4, are given below:-

	a_{12}	a_{13}	a_{15}	a_{23}	a_{35}	a_{14}	a_{24}	a_{45}	a_{34}	a_{25}
$\theta \Longrightarrow$	2.10	1.87	1.66	1.60	1.53	1.43	1.33	1.16	1.13	0.95
$\psi \Longrightarrow$	0.33	0.67	0.33	0.67	0	1	0.67	0.33	0.67	0

The judgments are arranged with the values of θ_{ij} in descending order. The

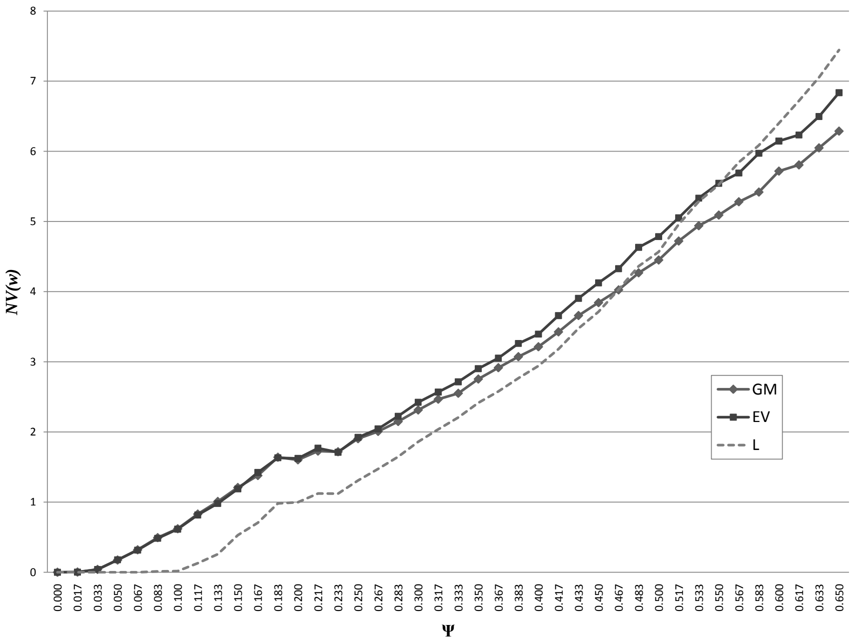


Figure 3.10: Average $NV(w)$ for various values of Ψ

value $\psi_{14} = 1$ for a_{14} implies that no judgment in A_4 supports the order of preference suggested by a_{14}. Whilst the most dissonant judgment is a_{14}, the most inconsistent is found to be a_{12} according to the CC measure. This paradox has generated a question of whether cardinal inconsistency should be given precedence over ordinal inconsistency.

3.5.1 Regression Analysis

In order to find a possible relation between Ψ and $NV(w)$, Monte-Carlo simulations were used as for CR, CM, GCI and ζ. Several consistent PCMs were constructed from randomly generated preference vectors. The ordinal inconsistency was induced by inverting some of the randomly selected elements i.e. assigning a reciprocal value. PCMs were generated for n varying from 4 to 9. The value of *dissonance*, Ψ, was calculated using (3.5) and (3.6). For every n, 60,000 PCMs were generated with different values for Ψ. All PCMs having the same value of Ψ were grouped together to calculate average values for the results. The granularity of Ψ can be derived from (3.5) and (3.6), equal to $\frac{2}{n(n-1)(n-2)}$.

In order to calculate $NV(w)$, weights were estimated using both the EV and the GM methods. Fig. 3.10 gives results calculated using EV and GM for $n = 6$, confirming a relation between *dissonance* (Ψ) and $NV(w)$. The more latent violations present in J, the more actual violations are likely to occur. The results appear similar to the one obtained for ζ, however, we observe two additional benefits of using Ψ compared to ζ. The average value of L is plotted in Fig. 3.10 to highlight its insensitivity at region $\Psi < 0.1$ for $NV(w)$. This region is significant, as it shows the possibility of having $NV(w) > 0$ for transitive PCM. Ψ can predict a possible violation in a transitive PCM, unlike ζ. In addition to this, the granularity used for the prediction is higher than ζ. From Fig. 3.10, a slight variation in monotonic increase at $\Psi \approx 0.20$ can be noticed. This phenomenon has not been explored and requires future investigation.

3.6 Illustrative Examples

To illustrate the concept of indirect judgments and the use of *congruence* and *dissonance*, four different types of PCMs are discussed below.

3.6.1 Consistent PCM

Consider an example of a consistent PCM given below:-

$$A_{ex1} = \begin{bmatrix} 1 & 2 & 4 & 12 \\ \frac{1}{2} & 1 & 2 & 6 \\ \frac{1}{4} & \frac{1}{2} & 1 & 3 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{3} & 1 \end{bmatrix}$$

The set of all indirect judgments calculated for A_{ex1} are identical to the direct judgments, therefore $\Theta = 0$ and $\Psi = 0$. The other consistency measures, CR and CM, also produce the same value, as expected. Hence, A_{ex1} contains a set of *congruent* judgments.

This can be visualized in Fig. 3.11 where each judgment lying in the upper triangular part of A_{ex1} is shown on a logarithmic scale. A circle in the figure depicts a direct judgment while indirect judgments are shown as an upwards arrow. All the indirect judgments in this case appear aligned with their direct judgment, following the consistency rule of $a_{ij} = a_{ik}a_{kj}$. Hence, all the prioritization methods generate the same weights i.e. $w = [\, 0.545 \; 0.273 \; 0.136 \; 0.045 \,]$.

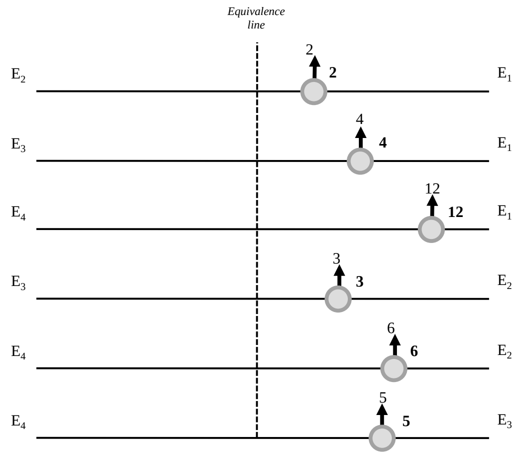


Figure 3.11: A_{ex1} with consistent judgments

3.6.2 Transitive PCM

Consider the following inconsistent PCM with $CR < 0.1$ and $L = 0$:-

$$A_{ex2} = \begin{bmatrix} 1 & 2\frac{1}{2} & 4 & 9\frac{1}{2} \\ \frac{2}{5} & 1 & 3 & 6\frac{1}{2} \\ \frac{1}{4} & \frac{1}{3} & 1 & 5 \\ \frac{2}{19} & \frac{2}{13} & \frac{1}{5} & 1 \end{bmatrix}$$

Unlike A_{ex1}, the indirect judgments in A_{ex2} are not equal to the direct ones. The indirect judgments may have different values than the direct ones, yet the order of *preference dominance* remains. As visible in Fig. 3.12, no indirect judgment has caused latent violations implying that the dissonance value is 0. Therefore, a preference vector with $NV = 0$ is possible for A_{ex2}. The values of CR, CM and Θ all suggest cardinal inconsistency among judgments, while Ψ confirms that A_{ex2} is ordinally consistent.

3.6.3 Transitive PCM with Latent Violations

Another PCM with $CR < 0.1$ and $L = 0$ is considered below:-

$$A_{ex3} = \begin{bmatrix} 1 & 3 & 2 & 6 \\ \frac{1}{3} & 1 & 1\frac{1}{5} & 2 \\ \frac{1}{2} & \frac{5}{6} & 1 & 3 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 1 \end{bmatrix}$$

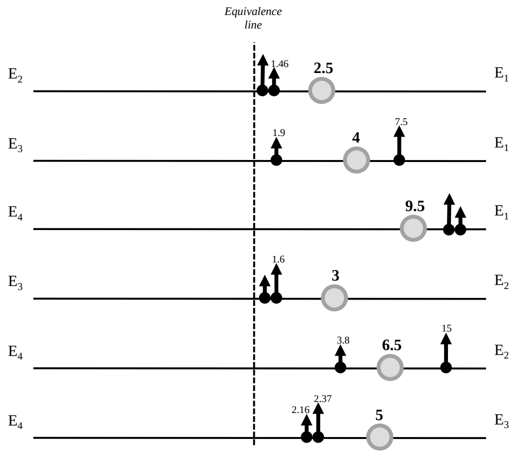


Figure 3.12: A_{ex2} having inconsistent judgments

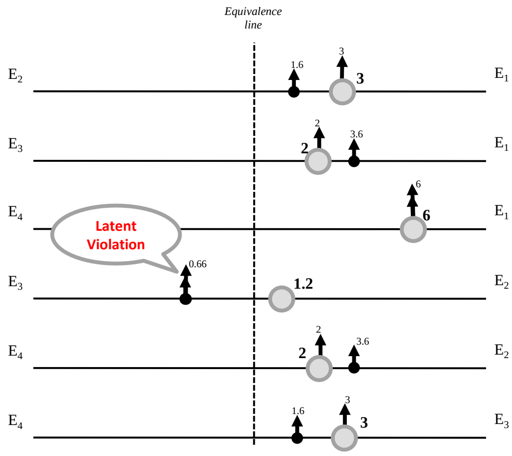


Figure 3.13: A_{ex3} having transitive but dissonant judgments

A_{ex3} is included here to investigate second-order intransitivity. Hartvigsen [71] pointed out that this comparison is transitive yet no method produces the *ideal ranking*. Considering the indirect judgments shown in Fig. 3.13, we can easily identify latent violations between E_2 and E_3. The direct judgment for a_{23} is 1.2, however, the two indirect judgments suggest 0.66, causing a priority violation whilst estimating weights. Although the values $CR = 0.031$, $CM = 0.555$ and $\Theta = 0.405$ highlight judgment inconsistency, the only measure suggesting a possibility of a priority violation is $\Psi = 0.167$.

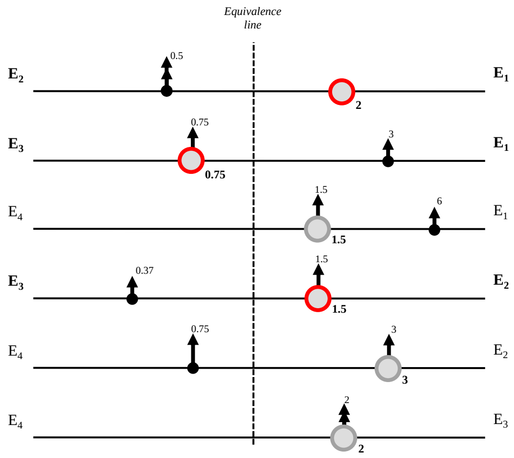


Figure 3.14: A_{ex4} having *intransitive* set of judgments

3.6.4 Intransitive PCM

An *intransitive* PCM ($L \neq 0$) always contain dissonant judgments. This is illustrated below with the following *intransitive* PCM:-

$$A_{ex4} = \begin{bmatrix} 1 & 2 & \frac{3}{4} & 1\frac{1}{2} \\ \frac{1}{2} & 1 & 1\frac{1}{2} & 3 \\ \frac{4}{3} & \frac{2}{3} & 1 & 2 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix}$$

There is one three-way cycle present in A_{ex4} i.e. $E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_1$. We discuss this example in detail with the help of Fig. 3.14. The two indirect judgments for a_{34} are perfectly consistent, while the judgment a_{14} has one consistent and one inconsistent judgment. The three judgments a_{13}, a_{23} and a_{24} show one latent violation each. Finally looking at a_{12}, we see that both indirect judgments are in conflict with the direct judgment. Thus the dissonance value ϕ_{12} is maximum (equals to 1), suggesting the judgment a_{12} as the outlier in this PCM.

The given examples have suggested measuring consistency as a pair of values i.e. (Θ, Ψ). Further investigation is required to explore the properties of these two consistency measures.

3.7 Conclusions

This chapter has investigated various consistency measures with the help of Monte-Carlo simulations. Useful findings regarding widely-used measurements scales are shown and discussed. The results confirm a relationship between cyclic judgments and the average number of violations.

Two new measures have been proposed: *congruence* for CC and *dissonance* for OC. These measures are illustrated through examples, and it is concluded that *congruence* is suitable for measuring both the CC of the individual judgments in a PC matrix (PCM) and the overall CC of the PCM. The use of *overall dissonance* (Ψ) is helpful in identifying priority violations. Regression analysis has been performed to formulate a relationship between the $NV(w)$ and Ψ. Further investigation is required to explain the non-monotonic increase of $NV(w)$ around $\Psi \approx 0.20$.

Furthermore, the two matrices showing individual *congruence* and *dissonance* can be used to detect and correct *inconsistent* (and/or *intransitive*) judgments. It is recommended to use these matrices as a useful addition to PC-based decision support tools. The discussion of these two measures for improving consistency is continued in the next chapter.

Chapter 4

New Methods for Improving Consistency

This chapter develops methods to improve the consistency in pairwise comparison (PC) matrices. Chapter 2 concluded with the discussion that existing methods for improving consistency do not take into account ordinal information. We continue this discussion with two new methods to improving consistency.

It is shown that both *cardinal consistency* (CC) and *ordinal consistency* (OC) can be improved with the help of the two measures proposed in previous chapter i.e. *congruence* and *dissonance*. The issue of ordinal inconsistency is then investigated further and a heuristic method is proposed to detect and rectify intransitive judgments. In order to compare the proposed method with existing ones, experimental results are analyzed.

4.1 Approaches for Improving Consistency

When both the *ordinal* and the *cardinal* inconsistency in PC judgments are found, the next obvious step is to improve the consistency prior to the process of prioritization. Different techniques have been introduced to detect and rectify inconsistency, either by using a local operator or a matrix-based global operation [97–99]. The improvements can be done manually by asking the decision maker (DM) to revise the judgments, or alternatively, this can be done automatically using an improvement technique.

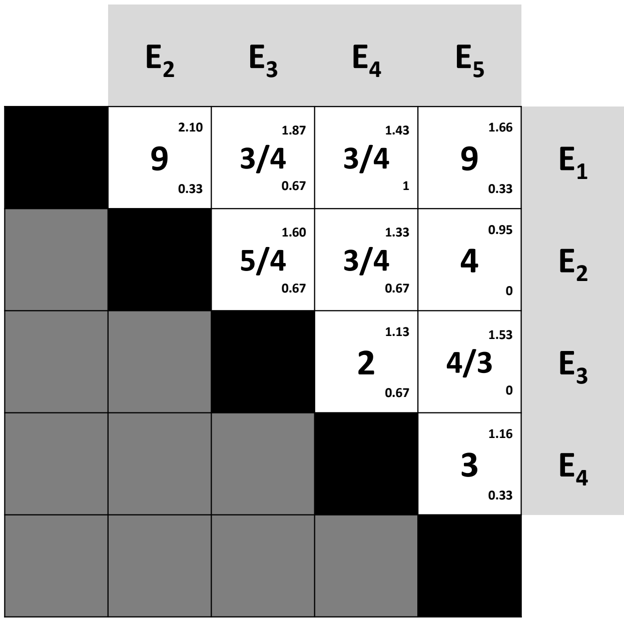


Figure 4.1: Offering θ and ψ for A_1 as numeric values

4.1.1 Manual Revision of Judgments

In situations where the provided judgments are allowed to be revised, it is suggested to offer the DM both θ_{ij} and ψ_{ij}. Consider the following PC matrix (PCM), for example:-

$$A_1 = \begin{bmatrix} 1 & 9 & \frac{3}{4} & \frac{3}{4} & 9 \\ \frac{1}{9} & 1 & \frac{5}{4} & \frac{3}{4} & 4 \\ \frac{4}{3} & \frac{4}{5} & 1 & 2 & \frac{4}{3} \\ \frac{4}{3} & \frac{4}{3} & \frac{1}{2} & 1 & 3 \\ \frac{1}{9} & \frac{1}{4} & \frac{3}{4} & \frac{1}{3} & 1 \end{bmatrix}$$

The PCM is presented to the DM in the form of a table, shown in 4.1, and is supposed to be manually revised by changing the values of a_{ij}. The two consistency measures, θ and ψ, are shown along with the original judgments in Fig 4.1. The values of θ_{ij} are shown above a_{ij} while ψ_{ij} is shown below a_{ij} in each cell. In practical situations, the DM may not make reasonably correct assessments as several numeric values on the table may cause information overload.

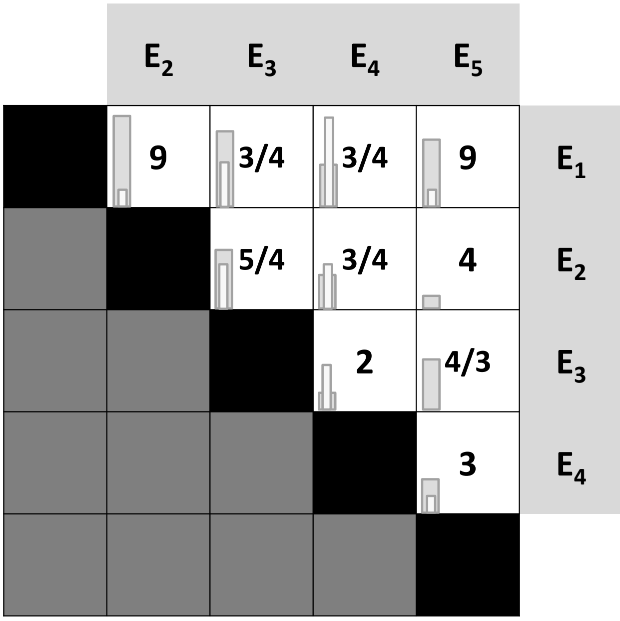


Figure 4.2: Offering θ and ψ for A_1 as a graphical aid

4.1.1.1 θ and ψ as a Graphical Aid

A better approach is to visualize the consistency of each judgment graphically as a bar graph. Fig. 4.2 shows this approach for A_1, where *congruence* is shown as a grey bar with each judgment and *dissonance* is shown as a white bar. A DM can easily spot the most inconsistent judgment ($a_{12} = 9$) to manually improve its value. When ordinal consistency has precedence, the value of a_{14} should be rectified first as it is ordinally the most inconsistent.

This graphical approach is useful to spot both *outliers* and the phenomenon of *consistency deadlock*. Consider the following two examples, already discussed in the previous chapter:-

$$A_2 = \begin{bmatrix} 1 & 2 & 4 & 8 & \frac{1}{3} \\ \frac{1}{2} & 1 & 2 & 4 & 8 \\ \frac{1}{4} & \frac{1}{2} & 1 & 2 & 4 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 & 2 \\ 3 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \quad , \quad A_3 = \begin{bmatrix} 1 & 2 & 2 & 2 & 2 \\ \frac{1}{2} & 1 & 2 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & 1 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$

Fig. 4.3 highlights the presence of an outlying judgment in A_2 where a_{15} is shown to be the most inconsistent, both cardinally and ordinally. Similarly, the phenomenon of *consistency deadlock* for A_3 is visually represented in Fig. 4.4.

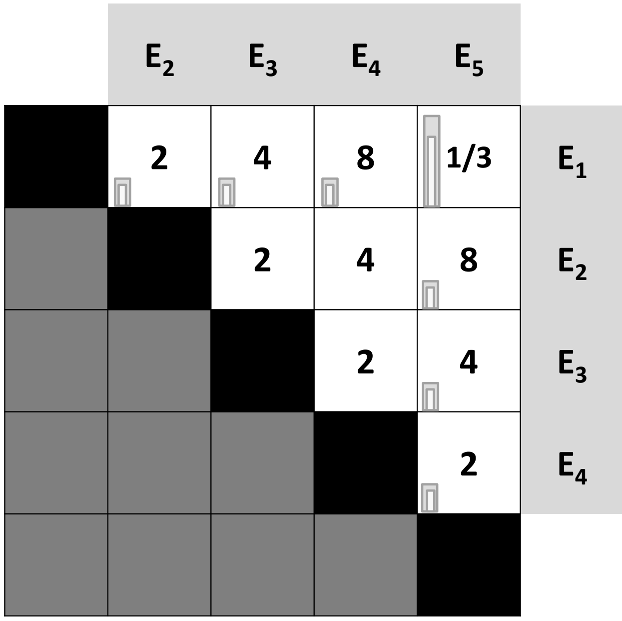


Figure 4.3: Judgment a_{15} as an outlier in A_2

All the judgments in A_3 are shown here to be equally responsible for inconsistency. There is no ordinal inconsistency present in A_3.

4.1.2 Reducing Inconsistency using Automated Approach

There exist situations in practice, where manual revision of judgments is not possible. One way to address this is to automatically remove inconsistent judgments with minimal deviation from the initial judgments. This can be achieved using the *congruence matrix*; choosing the judgment with the highest value of θ. When a set of judgments is ordinally inconsistent, the use of *dissonance matrix* proves supplemental, suggesting a two-step process i.e. firstly, choose the set of judgments with the highest value of ψ and then, from these short-listed ones, select the judgment with the highest value of θ.

In the presence of an *outlier*, a revised value for the blamed judgment can be calculated as:-

$$\tilde{a}_{ij} = \left(\prod_k a_{ik} a_{kj} \right)^{\frac{1}{n-2}} \tag{4.1}$$

where $i \neq k \neq j$. Another way to calculate a consistent value is to use the arithmetic mean i.e. $a_{ij} = \frac{1}{n-2} \sum_k a_{ik} a_{kj}$ where $i \neq k \neq j$. The use of the arithmetic or geometric mean may be an area for further investigation.

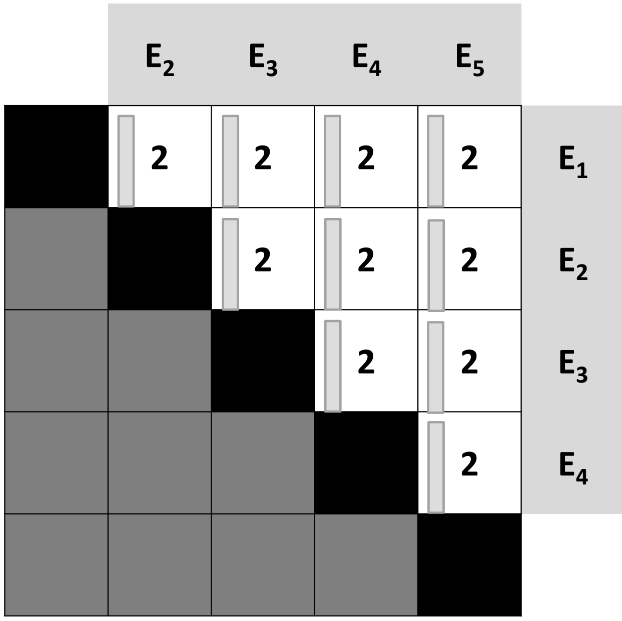


Figure 4.4: Consistency deadlock in A_3

In the case of ordinal inconsistency, the preference order of the most inconsistent judgment needs to be reversed. The suggested averaging process may not work in the presence of ordinal inconsistency. Considering the minimum deviation from the original value, a revised value can be calculated as:-

$$\tilde{a}_{ij} = \begin{cases} 0.99 & if\, a_{ij} > 1 \\ 1.01 & if\, a_{ij} < 1 \end{cases} \tag{4.2}$$

Consider A_2 for example; the value for a_{15} can be changed from $\frac{1}{3}$ to 16 using (4.1). If minimal change is to be expected, then the value of a_{15} should be changed to 1.01 using (4.2). Considering the phenomenon of *consistency deadlock*, as all the judgments in A_3 are equally inconsistent; the only possible improvement is to randomly select a judgment and replace its value using (4.1) which is equal to 4.

In the case of A_1, a_{12} is cardinally the most inconsistent and the suggested correction is $(a_{13}a_{32}a_{14}a_{42}a_{15}a_{52})^{\frac{1}{3}} = 1.1$. If ordinal inconsistency has priority, then a_{14} should instead be corrected and the suggested value for a_{14} is 1.01 using (4.2).

Holsztynski and Koczkodaj [99] proved that removing local inconsistencies incrementally will improve the overall consistency of PCMs. The same proof justifies convergence of the improvement process using θ_{ij} and ψ_{ij}.

4.2 Rectifying Intransitive Judgments

Correction of intransitive judgments can be achieved through a process of identifying all three-way cycles, and using this information to locate the most inconsistent judgment causing three-way cycles. As already discussed, the elimination of all three-way cycles will also remove all cycles of higher order.

Elimination of intransitive judgments can be stated as a graph theoretic problem to remove three-way cycles to achieve a Directed Acyclic Graph (DAG) - a problem - which is well known to be NP-Complete [105]. Slater [106] formulated this problem for tournament ranking. Several solutions have been proposed that consider ordinal information from pairwise judgments [62, 84, 95]. The focus here is to make use of cardinal information, in addition to ordinal, in order to derive adequate ranking from the PCs.

4.2.1 Optimization Approach

Elimination of intransitive judgments can be formulated as an optimization problem to remove cycles by changing a minimal number of elements in an intransitive PCM. In graph theory, this can be considered as a Minimum Feedback Arc Set (MFAS) problem [107]. The relationships between the elements of PCM can be depicted as a directed graph (digraph), $G = (E, Q)$, where E is the set of nodes representing n elements $E_1, E_2, ..., E_n$, and Q represents the set of all edges, $\{q_{ij}\}$. Each edge, q_{ij}, has an associated weight equal to the provided judgment, a_{ij}.

The MFAS problem consists of finding a minimum set of arcs, whose removal or reversal makes the graph acyclic ($L = 0$). This can be formulated as an integer linear programming (ILP) problem of the type [107]:-

$$\begin{aligned} minimize \quad & \sum_{q \in Q} x_q \\ s.t. \quad & \sum_{q \in \Gamma} x_q \geq 1 \, \forall \Gamma \in C_L \\ & x_q \in \{0, 1\} \; \forall q \in Q \end{aligned} \tag{4.3}$$

where C_L is the set of all cycles in G, and x_q are binary variables, equal to 1 if

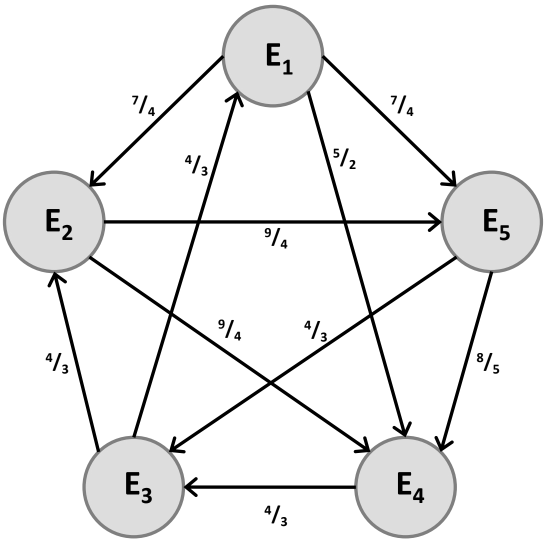


Figure 4.5: Preference graph for A_1

the edge q is in the feedback arc set and 0 otherwise.

Although various numerical optimization techniques, as well as evolutionary methods, are capable of solving the above problem, here we use an enumeration technique to find the optimum solution. Initially, all cycles are identified using Algorithm 2 and all the edges participating in at least one of the three-way cycles are determined. For each edge participating in three-way cycles (a blamed edge), a binary variable x_q is defined. By generating all different combinations of these binary variables, an optimal solution with minimum edge reversals is obtained.

It should be noted that the number of blamed edges cannot be equal to 4 (i.e. $\text{count}(x_q) \neq 4$). In the case of one three-way cycle, the number of blamed edges is always equal to 3, whilst the number of blamed edges is 5 for two overlapping cycles and 6 for two disjoint cycles. Therefore, the number 4 is impossible.

Consider again the intransitive PCM, A_1, which contains four three-way cycles, as already mentioned in Section 3.3. The graph for A_1 is shown in Fig. 4.5. The elements involved in those cycles are listed in the first row of Table 4.1.

As there are 8 edges in the three-way cycles ($\text{count}(x_q) = 8$), the overall number of patterns to test for this example is 255. The two solutions with minimum edge reversals obtained by the Enumeration algorithm are given in

	q_{14}	q_{15}	q_{24}	q_{25}	q_{31}	q_{32}	q_{43}	q_{53}
x_q (Solution 1)	0	0	0	0	0	0	1	1
x_q (Solution 2)	0	0	0	0	1	1	0	0

Table 4.1: Optimal solutions for A_1, obtained by the Enumeration algorithm

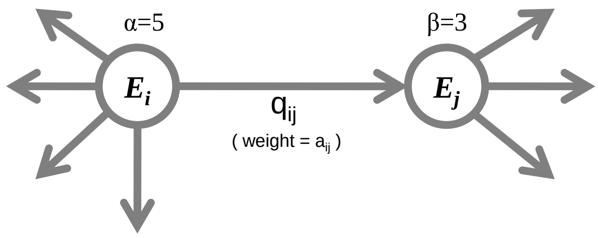


Figure 4.6: Outflows for source and destination elements

Table 4.1. Solution 1 shows that the two edges, q_{43} and q_{53}, need to be reversed. The corresponding judgments, a_{43} and a_{53}, need to be corrected by inverting their original values from $\frac{4}{3}$ to $\frac{3}{4}$. Similarly, solution 2 suggests the reversal of q_{31} and q_{32}.

Instead of reversing the judgments of both optimal solutions from $\frac{4}{3}$ to $\frac{3}{4}$, we may alternatively change the judgments to 0.99 using (4.2). This will reverse the edge and eliminate the cycles, however, it will also minimize overall changes. The value of the overall change may be used as an additional criterion for selection of the best MFAS solution.

In this example, both optimal solutions are equally good with respect to the overall changes, as all elements to be changed have equal values. Both solutions also improve the value of CR from 0.083 to 0.055.

4.2.2 Heuristic Approach

The correction of intransitive judgments can be achieved heuristically by using a greedy approach of locally removing one cycle and gradually converging towards a final transitive digraph. Such an approach is discussed for tournament ranking problems in [38]. However, in the case of PCs, the available cardinal information can be additionally used to further sift the possible edges to be reversed. The objective is to minimize the number of changes to be done to the original judgments provided.

Consider a partial preference digraph between any two elements E_i and E_j. Let α be the number of times that E_i is preferred over other elements in

Algorithm 3 A heuristic approach to rectify intransitive judgments

1. ENUMERATE TOTAL NUMBER OF THREE-WAY CYCLES
2. IF NO THREE-WAY CYCLE EXISTS, THEN GO TO STEP 7
3. CALCULATE $(\beta_j - \alpha_i)$ FOR EACH JUDGMENT a_{ij} WHERE $(i \neq j)$
4. SELECT THE JUDGMENT WITH $\max(\beta_j - \alpha_i)$
5. IF MULTIPLE JUDGMENTS FOUND WITH $\max(\beta_j - \alpha_i)$, THEN SELECT ONE WITH THE HIGHEST VALUE OF θ_{ij}
6. REVERSE THE SELECTED JUDGMENT AND GO TO STEP 1
7. TERMINATE

the overall digraph, similarly let β be the number of times E_j is preferred over other elements. Fig. 4.6 shows a digraph, where E_i is preferred to E_j as the edge q_{ij} is directed towards E_j. The element E_i is also preferred to four other elements, therefore, its number of outflows is $\alpha = 5$. Similarly, E_j is preferred to three other elements, so it has three outflows ($\beta = 3$).

Kendall and Smith [84] showed that when $\alpha > \beta$ and the edge q_{ij} is reversed, the number of three-way cycles (L) in the overall preference digraph is increased. Moreover, when $\alpha < \beta$, reversing the direction of the edge q_{ij} reduces the value of L. This observation can be used to develop an iterative heuristic algorithm, which can reduce both the intransitive elements and the number of three-way cycles.

At each iteration of the proposed Heuristic algorithm, the difference between the outflows of any two elements E_i and E_j is calculated and the edge, q_{ij} with the highest value of $(\beta_j - \alpha_i)$ is reversed. In the case of multiple edges meeting this condition, the most inconsistent edge is determined using the *congruence* measure, θ_{ij}. Algorithm 3 explains the steps that are required to rectify intransitive judgments using the proposed Heuristic approach.

Consider the digraph for A_1 shown in Fig. 4.5. The values of α and β for all its edges are listed in Table 4.2a. According to the algorithm, the edges with the highest value of $\beta - \alpha$ should be identified. In this case, q_{31} and q_{43} are the edges with the highest value of $(\beta - \alpha) = 1$. Therefore, the values of θ_{31} and θ_{43}, shown in Table 4.2a, are used to select the most inconsistent edge. As the level of inconsistency θ_{43} is greater than θ_{31}, the edge q_{43} should be reversed.

The edge can be reversed by interchanging the values of a_{34} and a_{43}. So the updated values are $a_{43} = \frac{3}{4}$ and $a_{34} = \frac{4}{3}$, respectively.

After the first iteration, two of the four cycles are removed from the original

	q_{12}	q_{14}	q_{15}	q_{24}	q_{25}	q_{31}	q_{32}	q_{43}	q_{53}	q_{54}
α_i	3	3	3	2	2	2	2	**1**	2	2
β_j	2	1	2	1	2	3	2	**2**	2	1
$(\beta_j - \alpha_i)$	-1	-2	-1	-1	0	1	0	**1**	0	-1
θ_{ij}	0.61	0.69	0.69	0.77	0.89	1.06	1.11	**1.12**	1.00	0.35

(a) 1st iteration

	q_{12}	q_{14}	q_{15}	q_{24}	q_{25}	q_{31}	q_{32}	q_{43}	q_{53}	q_{54}
α_i	3	3	3	2	2	3	3	3	**2**	2
β_j	2	0	2	0	2	3	2	0	**3**	0
$(\beta_j - \alpha_i)$	-1	-3	-1	-2	0	0	-1	-3	**1**	-2
θ_{ij}	0.61	0.59	0.69	0.67	0.89	0.96	1.01	0.82	**0.90**	0.25

(b) 2nd iteration

Table 4.2: The values of α, β and θ for A_1

PCM. The two cycles remaining in the updated PCM are:-

1. $E_1 \to E_5 \to E_3 \to E_1$
2. $E_2 \to E_5 \to E_3 \to E_2$

In the next iteration, the edges with the highest value of $\beta - \alpha$ are identified for the updated PCM. The edge q_{53} has the highest value of $(\beta - \alpha)$, as shown in Table 4.2b. The value of θ_{53} is irrelevant in this iteration, as there are no other edges with the same $(\beta - \alpha)$ value. Therefore, the comparison elements a_{53} and a_{35} are swapped and their new values become $a_{53} = \frac{3}{4}$ and $a_{35} = \frac{4}{3}$ respectively.

The updated form of A_1 after two iterations is shown below:-

$$\hat{A}_1 = \begin{bmatrix} 1 & \frac{7}{4} & \frac{3}{4} & \frac{5}{2} & \frac{7}{4} \\ \frac{4}{7} & 1 & \frac{3}{4} & \frac{9}{4} & \frac{9}{4} \\ \frac{4}{3} & \frac{4}{3} & 1 & \mathbf{\frac{4}{3}} & \mathbf{\frac{4}{3}} \\ \frac{2}{5} & \frac{4}{9} & \mathbf{\frac{3}{4}} & 1 & \frac{5}{8} \\ \frac{4}{7} & \frac{4}{9} & \mathbf{\frac{3}{4}} & \frac{8}{5} & 1 \end{bmatrix}$$

$\hat{A}_1$ has no three-way cycles and is transitive. Its CR is improved to 0.055 from the original value of 0.083. Similarly, the value of CM is improved from 0.775 to 0.697.

It can be seen that the obtained heuristic solution is equivalent to Solution 1 of the optimal enumeration algorithm, however, it is both much faster and

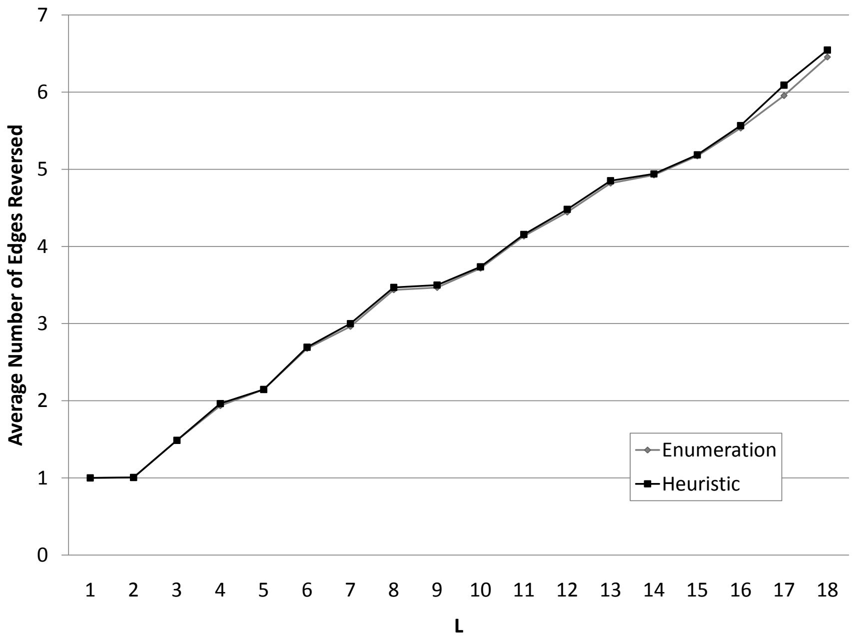


Figure 4.7: Average number of edge-reversals required w.r.t. L

simpler from a computation point of view.

We now compare the proposed heuristic algorithm to the optimal enumeration algorithm, using Monte-Carlo simulation. 2000 PCMs were randomly generated with L ranging from 1 to 18. The results in Fig. 4.7 show that the Heuristic algorithm achieves almost identical results to the Enumeration algorithm, with an occasional extra edge-reversal. The number of blamed edges, $\text{count}(x_q)$ varies for each case, however, there is a nearly linear increase in $\text{count}(x_q)$ when L is increased, as seen in Fig. 4.8.

Along with the enumeration approach, the proposed heuristic algorithm was also compared with ILP and a genetic algorithm (GA) approach [108]. The ILP approach was executed with the Revised Simplex method and the number of iterations set to 1000. The GA approach was executed with a population size of 10, maximum evaluations limited to 5000, crossover probability set to 0.5 and mutation probability also set to 0.5. The heuristic algorithm performed significantly faster compared to the other three approaches, as can be seen in Fig. 4.9. The computation time for the heuristic algorithm is less sensitive to the number of blamed edges, when compared to the enumeration algorithm. The performance comparison was obtained using an Intel-based computer with a Core2Duo T5500 CPU running at 1.66GHz and 2GB of physical memory. The

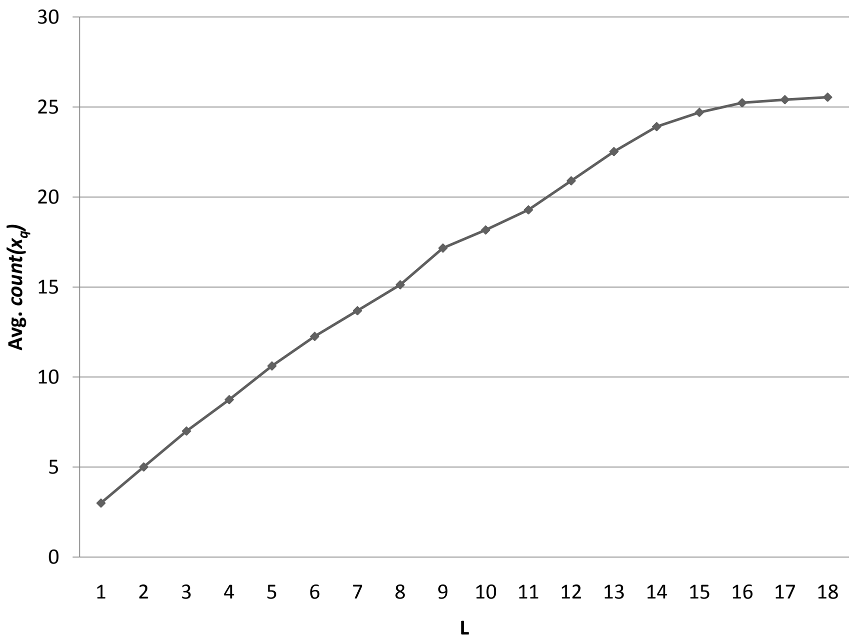


Figure 4.8: Increase in $\mathrm{count}(x_q)$ with L

tests were executed on Windows 7 with Java NetBeans IDE running in parallel.

The enumeration algorithm is particularly slow at finding solutions when the number of blamed edges exceeds 18. For example, ordinally inconsistent graphs with 26 blamed edges require an average processing time of 116.8 seconds to reach an optimum solution. In contrast, for the same set of problems, the heuristic algorithm finds a near-optimum solution in an average time of 0.858 milliseconds (approximately 5 orders of magnitude difference). Moreover, the increase in processing time for the enumeration algorithm is exponential, as seen in Fig. 4.9. The computation times for the ILP and the GA approach were more sensitive to the configured number of iterations and, therefore, performed better than enumeration for higher number of blamed edges i.e. higher than 13 for ILP and higher than 20 for GA. ILP performed much faster than GA (for well-known reasons). The results shown in Fig. 4.9 confirm the impossibility of having 4 blamed edges.

4.3 Summary

The *congruence* and *dissonance* matrix prove useful in visualizing the inconsistency of each judgment individually. It is shown that both CC and OC can be

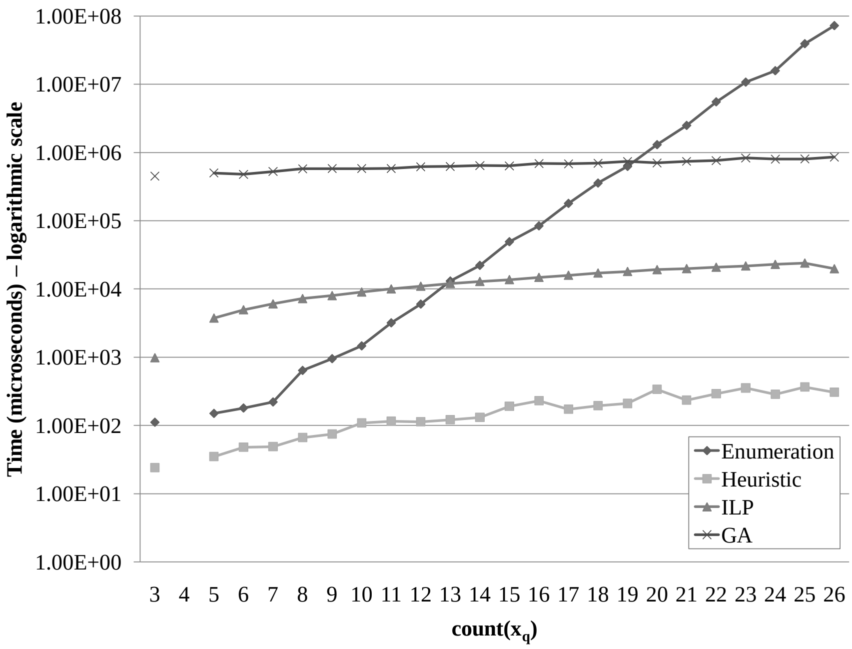


Figure 4.9: Average computation time w.r.t. number of blamed edges

improved with the help of these two measures.

The issue of *ordinal inconsistency* is thoroughly investigated and a heuristic method is then proposed to detect and rectify intransitive judgments. The use of the proposed method is explained through illustrative examples. The results suggest that the proposed method is feasible for improving OC and is computationally more efficient than the other evaluated approaches.

In this chapter, the graph-theoretic approach has been found useful for analyzing PCMs. The next chapter extends this discussion of using the graph-theoretic approach, and a new method has been proposed to elicit a preference vector from a given PCM.

Chapter 5

Prioritization using Graph Theory

This chapter proposes a new prioritization method using the graph-theoretic approach. The proposed method is formulated and illustrated with the help of examples. It is also shown that the method is applicable to incomplete pairwise comparison (PC) matrices without modification. Experimental results are shown to compare the performance of the new method with other methods.

5.1 Enumerating All Spanning Trees

It is possible to elicit preferences for n elements using a minimum number of $(n-1)$ independent judgments. The set of such independent judgments can be termed a "*pivotal combination*" of judgments. Mathematically, this represents a set of $(n-1)$ judgments, $\tau = \{a_{ij}\} \subseteq J$, such that there exists no k that allows the transitive relation i.e. $a_{ij} \Leftrightarrow a_{ik}a_{kj}$. When viewed as a graph, the set of judgments, τ, forms a *spanning tree* connecting all nodes together by a minimum number of edges.

When a complete set of judgments is provided ($m = \frac{n(n-1)}{2}$), several such trees can be extracted. The generation of all spanning trees has been widely used in computer science [109]. According to Cayley's theorem [110], the total number of spanning trees, η, is equal to $n^{(n-2)}$ for a fully-connected graph. A set of trees, $\Gamma = \{\tau_1, \tau_2, ..., \tau_\eta\}$ forms a *forest* that covers all the preferences provided by a *decision maker* (DM). A preference vector, $\tilde{w}$, can be generated from each pivotal combination, τ_s, where $s = 1, 2, ..., \eta$.

Harker [111] described the averaging process as a *natural* way to estimate

Algorithm 4 Prioritization using the enumeration of all spanning trees

FOR EACH COMBINATION OF $n-1$ JUDGMENTS, $\tau = \{a_{ij}\} \subseteq J$
  IF NO TRANSITIVE RELATION EXISTS i.e. $a_{ij} \not\Leftrightarrow a_{ik}a_{kj}$ FOR ALL $i \neq k \neq j$
    DECLARE THE COMBINATION AS PIVOTAL
    GENERATE A PRIORITY VECTOR, $\tilde{w}$ from τ
  END IF
END-FOR
CALCULATE AVERAGE OF ALL GENERATED VECTORS i.e. $w = \frac{1}{\eta}\sum \tilde{w}$
USE THE AVERAGE AS THE FINAL ESTIMATED VECTOR

preferences. Therefore, we use the *average* of all these vectors as a final estimated preference vector:-

$$w = \frac{1}{\eta}\sum_{s=1}^{\eta} \tilde{w}(\tau_s) \tag{5.1}$$

We call this method of obtaining a preference vector as *enumeration of all spanning trees* (EAST). The steps to calculate estimated preference vector are shown in Algorithm 4.

Consider a set of six judgments, J_1, to compare four stimuli, provided by a DM. The provided judgments are $a_{12} = \frac{1}{2}$, $a_{13} = 2\frac{1}{2}$, $a_{14} = 8$, $a_{23} = 3$, $a_{24} = 1\frac{1}{2}$ and $a_{34} = 2\frac{1}{2}$.

The value of η for J_1 is equal to 16, calculated using Cayley's theorem. The *forest* of trees, Γ_1 for J_1, can be visualized with the help of graph representation, as shown in Fig. 5.1. The figure displays every possible combination of minimum edges connecting all nodes together. Table 5.1 reveals all possible vectors, $\tilde{w}$, calculated from Γ_1. The final priority vector can be calculated as average of all these vectors using (5.1).

In graph theory, *arborescence* is a tree having one node considered as the root node, and all other nodes are directly connected to it by exactly one edge [112]. From Fig. 5.1 it can be seen that the forest Γ_1 contains four *arborescences* (visible on the extreme left of the forest). A PC matrix (PCM) can be constructed from J_1 as follows:-

$$A_1 = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & 8 \\ 2 & 1 & 3 & 1\frac{1}{2} \\ \frac{2}{5} & \frac{1}{3} & 1 & 2\frac{1}{2} \\ \frac{1}{8} & \frac{2}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

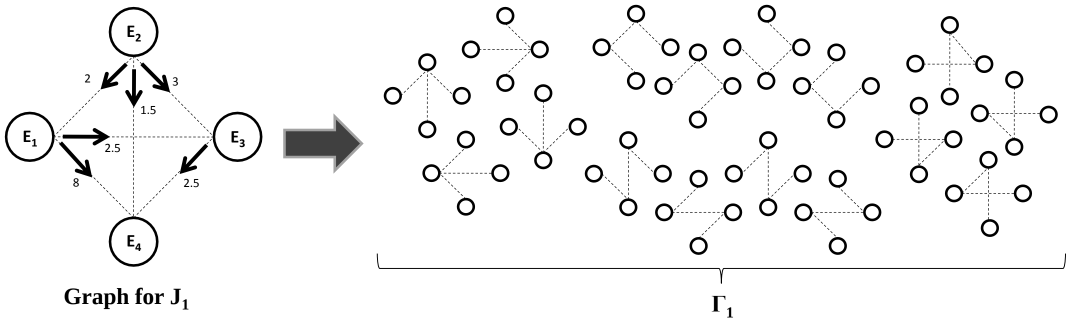


Figure 5.1: Graph representation for J_1, with a *forest* containing all spanning trees

The four *arborescences* represent the four columns of A_1 in normalized form. The average of these four columns gives the same weights produced by the additive normalization (AN) method, formulated as (2.13). We consider EAST to be a complete form of AN that takes into account all the possible combinations, including those overlooked by AN.

The set of $\tilde{w}$ is useful for analyzing consistency in judgments. The highest weight assigned by each τ_s is emphasized in Table 5.1 using bold, underlined font. Five out of sixteen vectors give the highest rank to w_1, whilst 10 of these vectors suggest w_2 as the most suitable; only one vector suggests w_3 as the best. This analysis suggests a democratic approach to accept ranking given by the majority of the generated trees, which is $w_2 - w_1 - w_3 - w_4$ for A_1.

We calculate $NV(w)$ for each $\tilde{w}$ to highlight the preference vectors responsible for the highest number of violations. The last two columns in Table 5.1 show the values of $TD(w)$ and $NV(w)$ calculated for each vector using (2.22) and (2.25), respectively. These vectors can be clustered based on their values for $TD(w)$ and $NV(w)$, and investigated further.

The weights generated by EAST along with other methods are shown in Table 5.2. The Eigenvector (EV), Geometric Mean (GM), and Direct Least Squares (DLS) methods generate solutions with one priority violation for A_1, and the Weighted Least Squared (WLS), Logarithmic Least Absolute Values (LLAV) and EAST methods give solutions with no priority violation. Of the methods that give no priority violations, EAST is closest to the initial judgments according to the Euclidean distance measure, $TD(w)$, as shown in Table 5.2.

τ_s	$\tilde{w}_1$	$\tilde{w}_2$	$\tilde{w}_3$	$\tilde{w}_4$	NV	TD
{ a_{12}, a_{14}, a_{23} }	0.2637	**0.5275**	0.1758	0.0330	0	4.2747
{ a_{12}, a_{24}, a_{34} }	0.1304	0.2609	**0.4348**	0.1739	1	2.0454
{ a_{13}, a_{23}, a_{24} }	0.2941	**0.3529**	0.1176	0.2353	0	1.8189
{ a_{13}, a_{23}, a_{34} }	0.3623	**0.4348**	0.1449	0.0580	0	2.3583
{ a_{13}, a_{14}, a_{23} }	0.3670	**0.4404**	0.1468	0.0459	1	1.7706
{ a_{14}, a_{23}, a_{34} }	**0.4211**	0.3947	0.1316	0.0526	2	2.1902
{ a_{14}, a_{23}, a_{24} }	**0.7273**	0.1364	0.0455	0.0909	0	2.1421
{ a_{14}, a_{24}, a_{34} }	**0.6154**	0.1154	0.1923	0.0769	0	4.2302
{ a_{13}, a_{14}, a_{24} }	**0.5839**	0.1095	0.2336	0.0730	0	4.3041
{ a_{12}, a_{23}, a_{24} }	0.2000	**0.4000**	0.1333	0.2667	2	2.2621
{ a_{12}, a_{23}, a_{34} }	0.2542	**0.5085**	0.1695	0.0678	0	3.2668
{ a_{13}, a_{24}, a_{34} }	**0.5556**	0.1333	0.2222	0.0889	2	3.9742
{ a_{12}, a_{13}, a_{14} }	0.2837	**0.5674**	0.1135	0.0355	2	0.8914
{ a_{12}, a_{14}, a_{34} }	0.2909	**0.5818**	0.0909	0.0364	2	0.9212
{ a_{12}, a_{13}, a_{24} }	0.2113	**0.4225**	0.0845	0.2817	3	2.2942
{ a_{12}, a_{13}, a_{34} }	0.2809	**0.5618**	0.1124	0.0449	2	0.9967

Table 5.1: Preference weight vectors extracted from J_1 using EAST

Method	w	NV	TD
EV	$(0.3777, 0.3759, 0.1468, 0.0996)^T$	1	25.66
GM	$(0.3786, 0.3687, 0.1618, 0.0910)^T$	1	23.88
LLS	$(0.3786, 0.3687, 0.1618, 0.0910)^T$	1	23.88
DLS	$(0.5167, 0.2457, 0.1653, 0.0723)^T$	1	12.22
WLS	$(0.3266, 0.4573, 0.1581, 0.0579)^T$	0	47.47
LLAV	$(0.3623, 0.4348, 0.1449, 0.0580)^T$	0	28.81
EAST	$(0.3651, 0.3717, 0.1593, 0.1038)^T$	0	27.29

Table 5.2: Preference weights estimated from A_1 (derived from J_1)

5.1.1 Handling Incomplete Sets of Judgments

Harker [113] investigated an incomplete set of judgments, where DMs are allowed to respond with *"don't know"* or *"not sure"* for some judgments. It has been highlighted that the probability of acquiring incomplete PCMs increases as n increases [27]. In such cases, the EV method cannot estimate preferences without applying an intermediate method to complete the missing judgments. The GM and *logarithmic least squares* (LLS) approaches are both equivalent for complete PCMs; however, only the LLS approach is applicable to incomplete PCMs.

EAST is applicable to incomplete PCMs without modification, generating a partial forest $\bar{\Gamma} \subset \Gamma$. In the case of an incomplete graph, η can be calculated with the help of a *Laplacian matrix* using Kirchhoff's matrix-tree theorem [114]. A Laplacian matrix, L, is the difference between the *degree matrix* and the *adjacency matrix* for a given graph, G. A *degree matrix*, D, is a diagonal matrix containing the degree of each vertex, E_i, as defined below:-

$$D_{ij} = \begin{cases} \deg(E_i) & if\ i = j \\ 0 & otherwise \end{cases}$$

The *adjacency matrix*, C, is a matrix with each element c_{ij} representing the number of edges from vertex E_i to vertex E_j. The determinant of the matrix, L^*, obtained by omitting any row and the corresponding column of L, gives the value of η [115].

Consider the following three incomplete forms of A_1:-

$$A_2 = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & - \\ 2 & 1 & 3 & 1\frac{1}{2} \\ \frac{2}{5} & \frac{1}{3} & 1 & 2\frac{1}{2} \\ - & \frac{2}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & - \\ 2 & 1 & - & 1\frac{1}{2} \\ \frac{2}{5} & - & 1 & 2\frac{1}{2} \\ - & \frac{2}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & - \\ 2 & 1 & - & 1\frac{1}{2} \\ \frac{2}{5} & - & 1 & - \\ - & \frac{2}{3} & - & 1 \end{bmatrix}$$

The PCM A_2 has one judgment missing for a_{14}. As stated, the number of trees in Γ_2 can be calculated using Kirchhoff's matrix-tree theorem. The *degree matrix*, D_2, and the *adjacency matrix*, C_2, are given below:-

$$D_2 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

The Laplacian matrix for A_2 can be calculated from D_2 and C_2 as below:-

$$LP_2 = D_2 - C_2 = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

A sub-matrix, LP_2^*, can be obtained by omitting the first row and the first column of LP_2. The determinant of LP_2^* gives the number of trees present in Γ_2 i.e.

$$\eta_2 = |LP_2^*| = \begin{vmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 2 \end{vmatrix} = 8$$

Fig. 5.2 shows the forests generated for A_2, A_3 and A_4. The forest, Γ_2, generated from A_2 contains eight trees from the total of sixteen trees in Γ_1. Similarly, the forest Γ_3 generated for A_3 contains four trees while Γ_4 contains only one. For A_4, only three judgments are provided by the DM, therefore, only one tree in Γ_4 has been generated.

The preference vectors for A_2, A_3 and A_4 are estimated using widely used

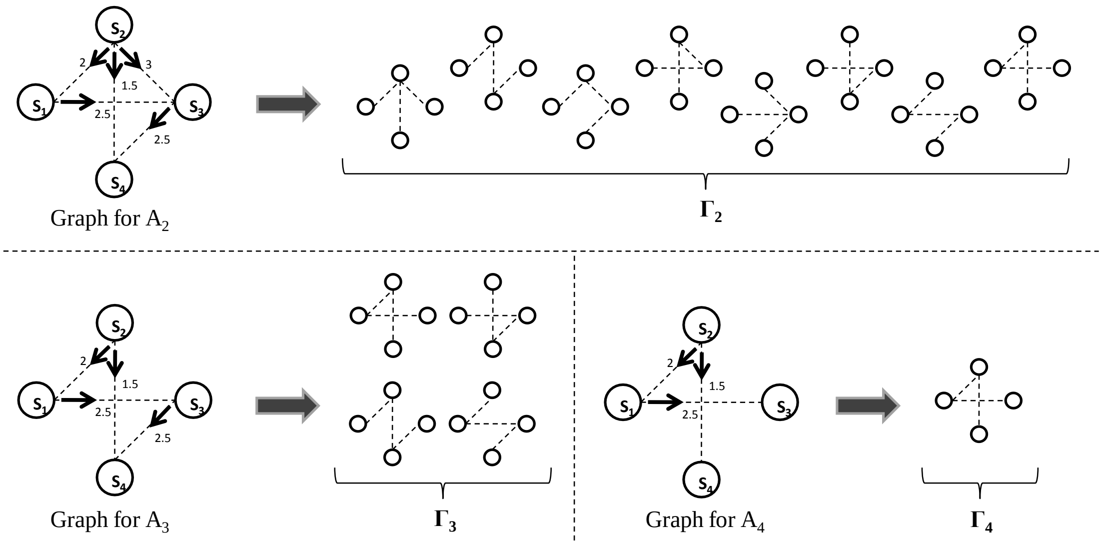


Figure 5.2: Spanning trees present in A_2, A_3, and A_4

optimization methods i.e. LLS, DLS, WLS, and LLAV. The results are shown in Table 5.3. The priority vector estimated using EAST is also given. EV and GM are not used due to their inability to estimate from an incomplete set of PC judgments.

The preference vectors generated for A_2 are shown first. Considering the two criteria of $NV(w)$ and $TD(w)$, EAST has produced the most suitable vector, giving no violation, as compared to DLS and LLAV. LLS and WLS has produced solutions with no violations ($NV = 0$) but the value of $TD(w)$ is higher than for EAST. Similar results are obtained for A_3, where two judgments are missing. Considering A_4, all the methods derive the same preference vector for obvious reasons.

5.2 Regression Analysis for Comparison

Regression analysis was carried out to compare EAST with different methods. To enable this analysis, Monte-Carlo simulations were used. For each iteration, a PCM, $A = \lfloor a_{ij} \rfloor$, was constructed by filling in random values from Saaty's 1-9 scale [42]. The elements in the lower triangular part of the generated PCM were recalculated accordingly, so that $a_{ji} = {}^{1}/_{a_{ij}}$. Such matrices were generated 3000 times for each value of n, ranging from 3 to 6. For each comparison, priority vectors were estimated using the EV, GM (LLS), DLS, WLS,

	Method	w_1	w_2	w_3	w_4	*NV*	*TD*
A_2:							
	LLS	0.2937	0.3966	0.174	0.1357	0	0.7148
	DLS	0.2865	0.3912	0.1375	0.1848	1	0.6375
	WLS	0.2741	0.4394	0.1511	0.1354	0	0.7484
	LLAV	0.2259	0.4308	0.1436	0.1997	1	0.6687
	EAST	0.2861	0.3843	0.1774	0.1521	0	0.6889
A_3:							
	LLS	0.3013	0.3547	0.2048	0.1392	0	0.6975
	DLS	0.4959	0.1822	0.2223	0.0997	1	0.6040
	WLS	0.2743	0.4328	0.157	0.1359	0	0.8210
	LLAV	0.2167	0.4025	0.1334	0.2473	1	0.7621
	EAST	0.2945	0.3446	0.2135	0.1474	0	0.6849
A_4:							
	LLS	0.2809	0.5618	0.1124	0.0449	0	0.0000
	DLS	0.2809	0.5618	0.1124	0.0449	0	0.0000
	WLS	0.2809	0.5618	0.1124	0.0449	0	0.0000
	LLAV	0.2809	0.5618	0.1124	0.0449	0	0.0000
	EAST	0.2809	0.5618	0.1124	0.0449	0	0.0000

Table 5.3: Preference weights estimated from A_2, A_3 and A_4

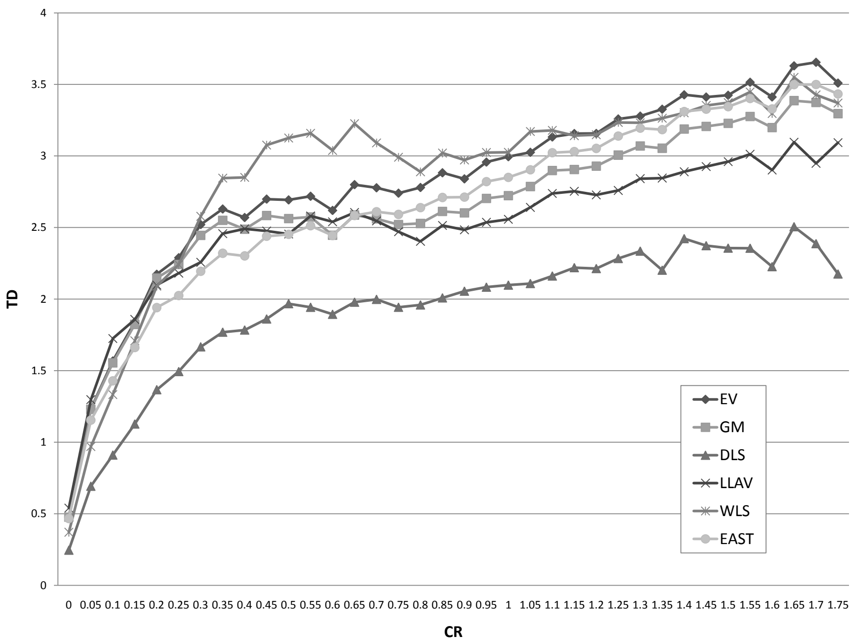
LLAV and EAST methods. All the optimization methods were implemented by a genetic algorithm written in Java with the help of the jMetal toolkit [116]. The algorithm was executed with a population size of 10, with maximum evaluations limited to 45000, crossover probability set to 0.5 and mutation probability also set to 0.5.

As mentioned earlier, the EV solutions are unsatisfactory when inconsistencies are larger ($CR > 0.1$). An equivalent threshold has been provided for GM [86]. Therefore, the above experiments were repeated to investigate how EAST behaves when $CR < 0.1$. 3000 PCMs were generated with $CR < 0.1$. The priority vectors were estimated using the EV, GM and EAST methods.

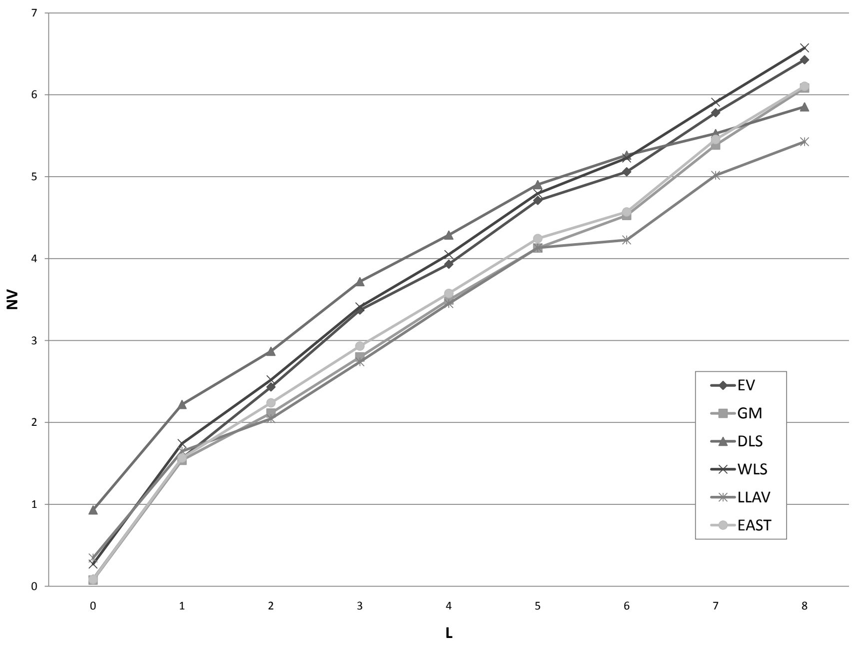
5.2.1 Results

The results were compared using two widely used criteria i.e. TD and NV. Fig. 5.3 gives results for the first stage of experiments with no constraints on the PCM's consistency, while Fig. 5.4 shows the results for PCMs generated with CR<0.1.

The values of TD generated by different methods are shown in Fig. 5.3a



(a) TD vs CR



(b) NV vs L

Figure 5.3: Regression results for EAST compared to various methods

against CR values ranging from 0 to 1.75. Fig. 5.3a shows that EAST performs better than EV, GM and LLAV when CR<0.1, and also outperforms WLS for 0.1<CR<0.65. LLAV and GM perform better than EAST for CR>0.65. The DLS proves to be the lower bound for all values of CR.

Fig. 5.3b shows the results for NV plotted against L. The plot for NV shows an almost linear relation for all methods. GM and EAST give the fewest violations for transitive PCMs, visible at $L = 0$. For intransitive PCMs, LLAV and GM outperform all other methods. However, EAST remains close to GM. LLAV produces the fewest violations in the case of highly intransitive PCMs.

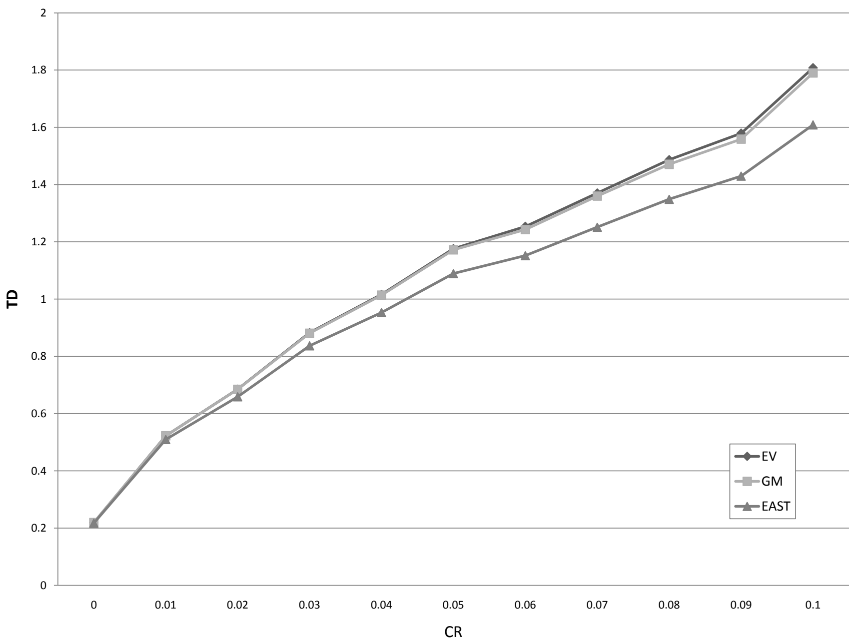
The second phase of the experiment was carried out to closely observe the EV, GM and EAST methods for acceptable PCMs (i.e. $CR \leqslant 0.1$). Fig. 5.4a shows that EAST has outperformed both EV and GM in the acceptable range of CR<0.1. All the methods perform similarly for the NV criterion, as shown in Fig. 5.4b. There is a strong similarity between the performance of EAST and GM for the NV criterion, as shown in Fig. 5.3b and Fig. 5.4b.

Considering the conformity criterion, EAST does not deviate from other methods and thus cannot be declared to be an *outlier*, as described for some other methods in [58].

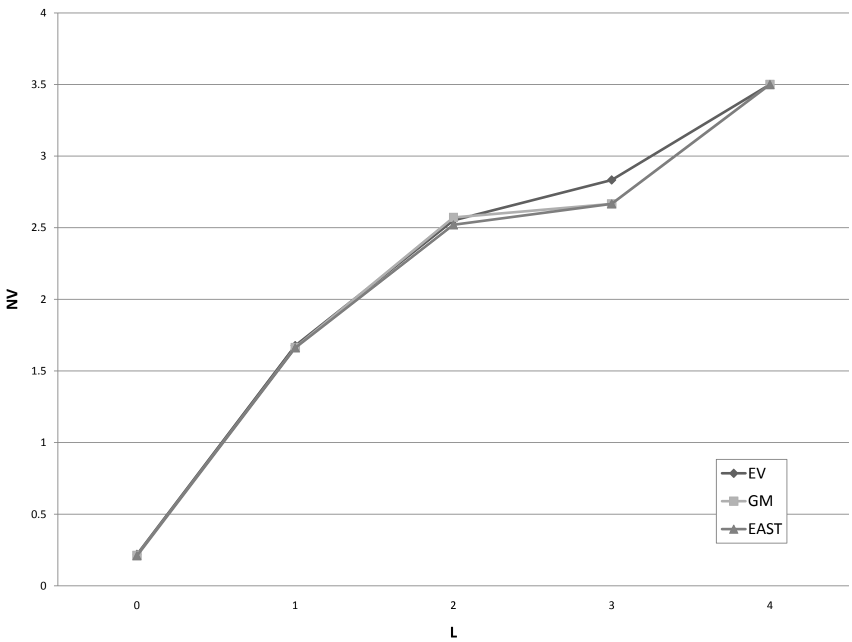
5.3 Limitation

As already discussed, $n^{(n-2)}$ spanning trees can be generated for a fully-connected graph. This means 16 trees for $n = 4$, 125 trees for $n = 5$, 1296 trees for $n = 6$, and a hundred million trees for $n = 10$. This imposes a current practical limitation on the generation of trees for $n > 9$. EAST has been tested with n ranging from 3 to 8 along with the other widely used methods. The machine used for this test was an Intel-based machine with Core2Duo T5500 CPU running at 1.66GHz. Figure 5.5 shows computation times on a logarithmic scale for the different methods.

GM and EV take the lowest amount of processing time and are insensitive to n. All the optimization-based algorithms (including LLS, DLS, WLS and LLAV) were computed with two different settings:- OPT1 shows the time taken with the maximum number of evaluations limited to 45000, whilst OPT2 shows the time taken with 25000 evaluations. The results for OPT1 are more accurate than for OPT2 for obvious reason: the optimization-based algorithms are more



(a) TD vs CR



(b) NV vs L

Figure 5.4: Regression results for EAST compared to EV and GM (CR<0.1)

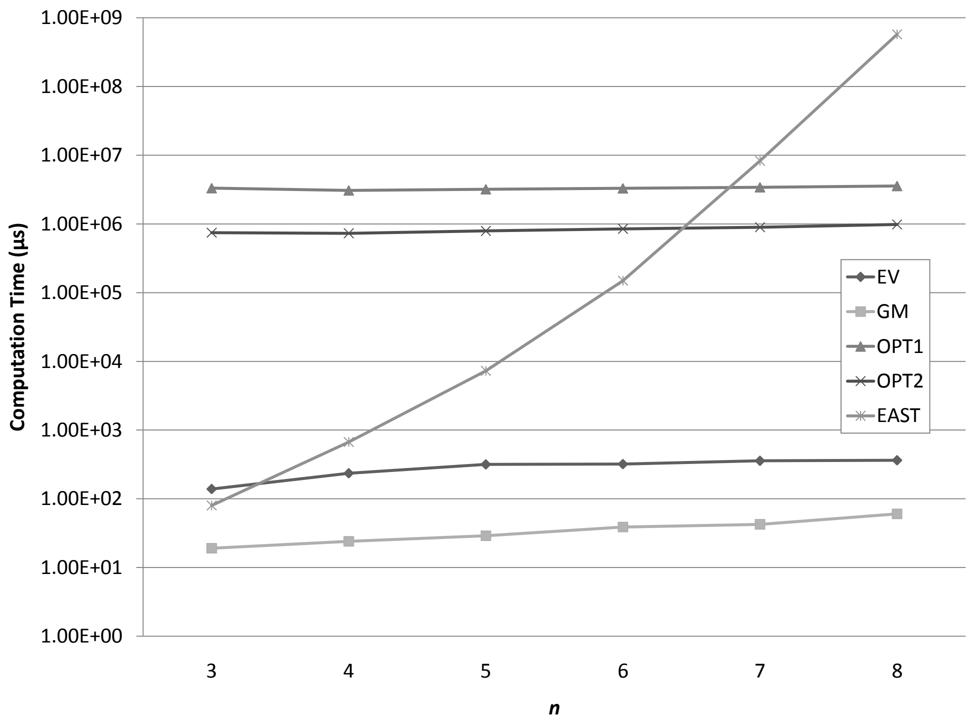


Figure 5.5: Comparing prioritization methods w.r.t. Computation time

accurate with an increased number of evaluations. EAST calculates preferences in less than a second for $3 \leqslant n \leqslant 6$. However, EAST does take more time than the other methods to calculate preferences for $n \geqslant 7$, evident in Figure 5.5.

Miller [5] established a rule of 7 ± 2 in cognitive psychology, explaining that human performance tends to deteriorate when handling more than seven stimuli. Based on this rule, Saaty [42] defined a maximum limit of $n = 9$ for PCMs. EAST therefore is applicable in all practical cases.

For a computer-aided automated process, the limit of $n \leqslant 9$ may not remain valid as greater number of stimuli can be analyzed in such cases. However, for large matrices, the full enumeration of all spanning trees may be unnecessary, for example, in the case where the evaluation of a partial forest may produce reliable enough results. The trees in a forest can be processed iteratively and w can be calculated using a running average approach. In this way, the iteration process can be stopped without need to search the whole forest, provided the running average is found to be convergent. The proposed iteration process and the concept of convergence are an area for further investigation.

5.4 Summary

This chapter has proposed a new prioritization method using the graph-theoretic approach to generate a forest containing all spanning trees. This reveals all the possible preferences present in the mind of a DM. The *mean* of these preferences has been suggested as the final priority vector, and the *variance* as the measure of inconsistency. EAST can estimate preferences from incomplete PCMs without modification, unlike other popular methods which require intermediate steps to estimate missing judgments.

Experimental results have shown that the performance of EAST is comparable to other approaches. Unfortunately, the new method becomes intractable when the number of elements to compare is greater than 9. Considering this limitation of EAST, the discussion of eliciting preferences is continued in the next section with a proposal to use multi-objective optimization.

Chapter 6

Multi-objective Prioritization

Due to the practical limitations of *enumerating all spanning trees* (EAST) highlighted in Chapter 6, the issue of prioritization is re-considered in this chapter using a different approach. This approach uses an optimization method to simultaneously minimize deviations from direct and indirect judgments. The new approach is applied to several examples and the generated priority vectors then compared to existing methods.

6.1 Prioritization Methods and NV

Most prioritization methods derive priorities with the same ranking when the set of pairwise comparison (PC) judgments is ordinally consistent (*transitive*), possibly with different intensities. However, different rankings are produced when the comparisons are ordinally inconsistent (*intransitive*). In order to analyze the effects of transitivity on the estimated priority vectors, Monte-Carlo simulations were used.

6.1.1 Monte-Carlo Experiments

Firstly, a number of preference vectors $r = (r_1, r_2, ..., r_n)^T$ were randomly generated and, for each vector a consistent PC matrix (PCM) was constructed, such that $a_{ij} = \frac{r_i}{r_j}$ for all $i, j = 1, 2, ..., n$.

In order to introduce inconsistency, uniformly distributed noise was superimposed on randomly selected elements in the upper triangular part of PCM and the elements in the lower triangular part were recalculated accordingly, so that $a_{ji} = {}^{1}\!/_{a_{ij}}$. In order to induce three-way cycles in these PCMs, some a_{ij}

were also randomly swapped with their reciprocal elements a_{ji}, thus increasing ordinal inconsistency.

Uniformly distributed random PCMs were generated for n varying from 3 to 9. The upper limit of $n = 9$ was selected according to the maximal dimension of the PCM, and because it is recommended by Saaty for the AHP method [17]. The number of three way cycles, L was calculated using Algorithm 2.

Two phases of experimentation have been carried out in order to explore the relationship between the *number of three-way cycles*, L and the *number of priority violations*, $NV(w)$. In phase one, PCMs were generated having uniform distribution with no constraints on their consistency. In the second phase, PCMs were generated with acceptable inconsistency i.e. $CR \leqslant 0.1$. For each comparison, priority vectors were estimated using the Eigenvector (EV), Geometric Mean (GM), Direct Least Squares (DLS), Weighted Least Squares (WLS) and Logarithmic Least Absolute Values (LLAV) methods. Along with these five methods, a new optimization method was applied to estimate the priority vector giving the minimum number of violations (MNV) for a given PCM.

All the optimization methods were implemented using a genetic algorithm written in Java, with the help of the jMetal toolkit [116]. The algorithm was executed with a population size of 20, maximum evaluations limited to 25000, crossover probability set to 0.5 and mutation probability also set to 0.5. 3,000 PCMs were generated for each value of n.

All PCMs with the same value of L were grouped together to calculate average values for the results. This grouping was done to enable an analysis of how different number of cycles present in a PCM may affect the number of violations generated by different methods. For each group, an average value of $NV(w)$ was calculated using different prioritization methods.

6.1.2 Results

The design of the Monte-Carlo simulations may influence the results and hence possibly change the rankings of prioritization methods. However, the primary target of this experiment was to indicate that different prioritization methods perform differently in the presence of cycles. The experiment does show a general trend for how different methods behave in the presence of intransitive judgments. The average $NV(w)$ generated by each prioritization method is sensitive to L. MNV gives the lower bound for the number of violations as would be expected and is confirmed in both Fig. 6.1a and Fig. 6.1b.

An important observation is that the EV and the GM method perform similarly when CR is bounded to remain less than 0.1. However, on average, GM gives a lower number of violations when compared to EV. WLS performs worse than both the GM and EV methods. In general, LLAV produces the lowest number of violations of all the methods; however, this does not apply for acceptable PCMs, where both EV and GM outperform LLAV. The DLS optimization performs worst giving the highest average value for $NV(w)$. It is important to highlight that no method is performing close to the lower bound (generated by MNV), indicating well the need for a new method to minimize violations.

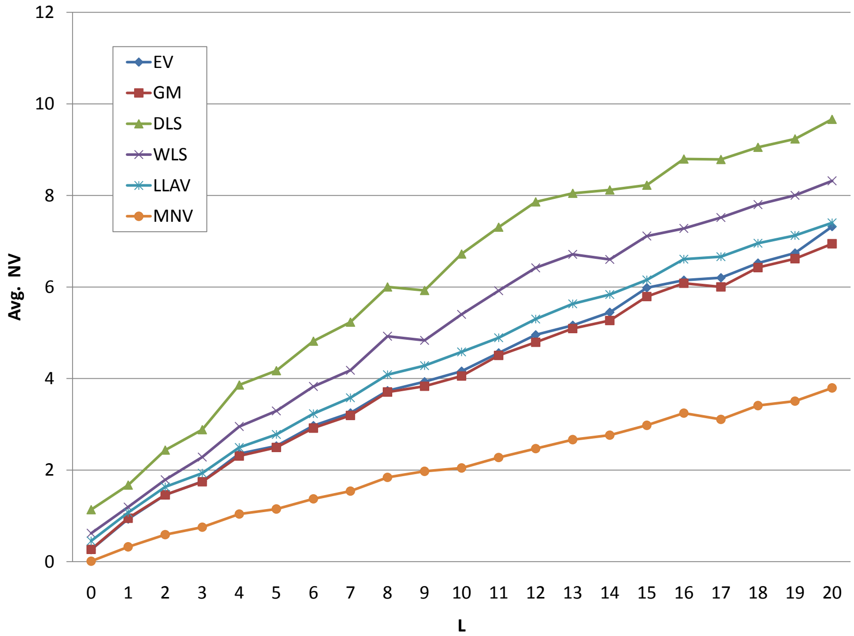
The results of the Monte-Carlo experiment confirms that the existing methods generate vectors with more violations than MNV. The idea of minimizing the number of violations has already been proposed in the Two-Objective Prioritization (TOP) method [2], where the objectives, TD and NV, are optimized using multi-objective optimization. The benefit of using TOP is that it can generate multiple Pareto-optimal solutions for a decision maker (DM) to interactively select the most appropriate. Here, we extend this approach to minimize three objectives.

6.2 Revealing the Indirect Judgments

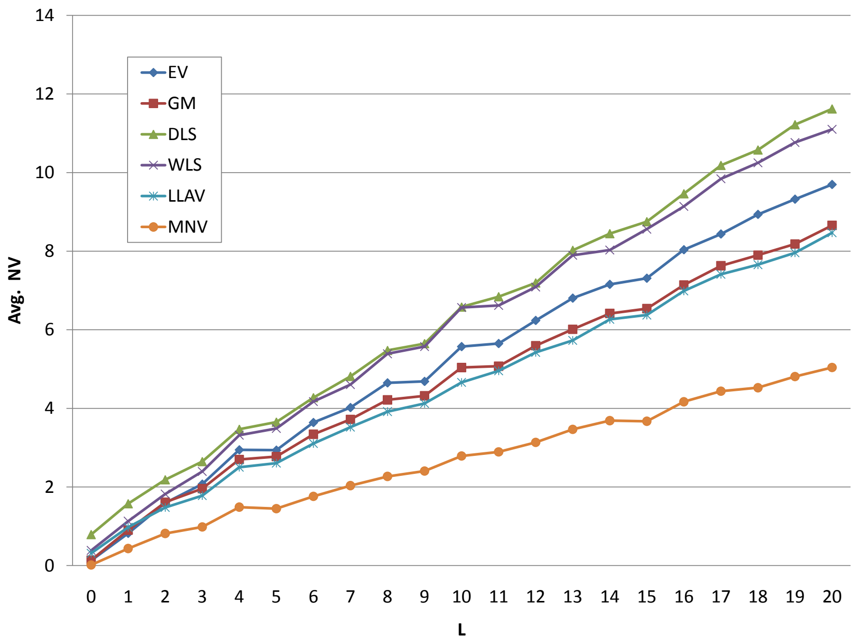
Considering the cardinal consistency test between E_i and E_j i.e. $a_{ij} = a_{ik}a_{kj}$ (for all i, j, and k), if the DM's judgments are cardinally inconsistent, then we obtain at least one indirect judgment $b_{ij} = a_{ik}a_{kj}$ incongruent with a_{ij} for some i, j and k where $i \neq j \neq k \neq i$. For each pair of elements in J, a maximum of $(n-2)$ indirect judgments can be obtained using different intermediary elements. Hence, a total number of indirect judgments, m_b, inferred from J has an upper bound of:-

$$m_b \leqslant \frac{n(n-1)(n-2)}{2} \tag{6.1}$$

Using (6.1) for $n = 3$, only three indirect judgments are possible ($m_b^{(n=3)} \leqslant 3$), one for each direct judgment. Similarly, for $n = 4$, each direct judgment can have two indirect judgments, suggesting a total number of 12 indirect judgments i.e. $m_b^{(n=4)} \leqslant \frac{4(4-1)(4-2)}{2} = 12$. The total number of direct and indirect judgments for $n = 3$ to 9 are shown in Table 6.1.



(a) For *acceptable* PCMs

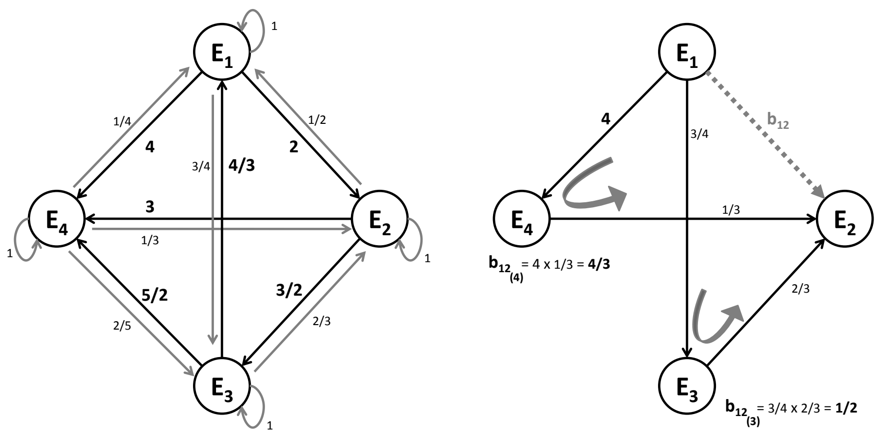


(b) For unconstrained PCMs

Figure 6.1: Average $NV(w)$ vs L

n	3	4	5	6	7	8	9
$\max(m)$	3	6	10	15	21	28	36
$\max(m_b)$	3	12	30	60	105	168	252

Table 6.1: Number of direct and indirect judgments possible in PCMs



(a) direct judgments (b) Indirect judgments $b_{12_{(3)}}$ and $b_{12_{(4)}}$

Figure 6.2: Graphs for A_{n4} showing direct and indirect judgments

Consider the following PCM with $n = 4$:

$$A_{n4} = \begin{bmatrix} 1 & 2 & \frac{3}{4} & 4 \\ \frac{1}{2} & 1 & \frac{3}{2} & 3 \\ \frac{4}{3} & \frac{2}{3} & 1 & \frac{5}{2} \\ \frac{1}{4} & \frac{1}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

The graph for A_{n4} is shown in Fig. 6.2a. An indirect judgment for a_{12} can be obtained as $b_{12_{(3)}} = a_{13}a_{32} = \frac{3}{4} \times \frac{2}{3} = \frac{1}{2}$. The subscript (3) for the index denotes the use of E_3 as an intermediate element. Similarly, the judgment using E_4 is calculated as $b_{12_{(4)}} = a_{14}a_{42} = \frac{4}{3}$. The use of the reciprocal property is obvious in these calculations. The indirect judgments $b_{12_{(3)}}$ and $b_{12_{(4)}}$ are highlighted in Fig. 6.2b. A complete set of indirect judgments for A_{n4} is provided below:-

$$\begin{aligned}
b_{12}: &\quad b_{12_{(3)}} = a_{13}a_{32} = 1/2, \quad b_{12_{(4)}} = a_{14}a_{42} = 4/3 \\
b_{13}: &\quad b_{13_{(2)}} = a_{12}a_{23} = 3, \quad b_{13_{(4)}} = a_{14}a_{43} = 8/5 \\
b_{14}: &\quad b_{14_{(2)}} = a_{12}a_{24} = 6, \quad b_{14_{(3)}} = a_{13}a_{34} = 15/8 \\
b_{23}: &\quad b_{23_{(1)}} = a_{21}a_{13} = 3/8, \quad b_{23_{(4)}} = a_{24}a_{43} = 6/5 \\
b_{24}: &\quad b_{24_{(1)}} = a_{21}a_{14} = 2, \quad b_{24_{(3)}} = a_{23}a_{34} = 15/4 \\
b_{34}: &\quad b_{34_{(1)}} = a_{31}a_{14} = 16/3, \quad b_{34_{(2)}} = a_{32}a_{24} = 2
\end{aligned}$$

The indirect judgments, involving two intermediary elements can also be calculated as $b_{ij}^{(2)} = a_{ik_1}a_{k_1k_2}a_{k_2j}$ where $i, j, k_1, k_2 \in \{1, 2, \ldots, n\}$. However, unique combinations of i, j, k_1 and k_2 are only possible for $n > 3$. The superscript (2) in $b_{ij}^{(2)}$ denotes the indirect judgment involving two intermediary elements, E_{k_1} and E_{k_2}). Therefore, $b_{ij}^{(2)}$ can be termed as second-order indirect judgments. When viewed as a graph, each indirect judgment is calculated with the help of three edges (judgments).

We can further generalize this by estimating a_{ij} through indirect assessments including up to $(n-2)$ intermediary elements i.e. b_{ij}, $b_{ij}^{(2)}$, $b_{ij}^{(3)}$, ..., $b_{ij}^{(n-2)}$. For example, indirect assessments including one, two and three intermediary elements are possible for $n = 5$.

The concept of indirect judgments has been analyzed for the two most widely used methods i.e. EV and GM.

6.2.1 EV and Indirect Judgments

In the EV method, preference weights can be estimated using the *power method* by raising the matrix A to the power k, and then normalizing the column sum of the raised matrix (A^k) [42]. Harker and Vargas [41] used a graph-theoretic approach to show that the EV method takes into consideration the indirect judgments. Each element, $\overline{a}_{ij}^{(k)}$ in A^k estimates the overall intensity of E_i over E_j along paths of length k.

Consider a generalized form of PCM with $n = 3$, as mentioned below:-

$$A_{n3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

This PCM is represented graphically in Fig. 6.3. Focusing on a_{12}, raising A_{n3}

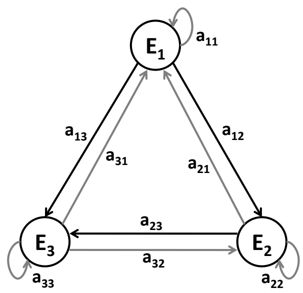


Figure 6.3: direct judgments

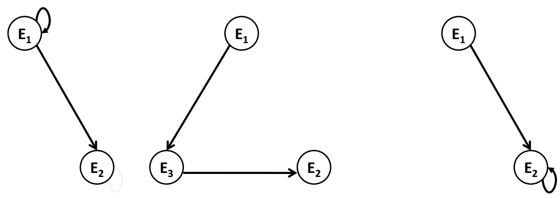


Figure 6.4: Three paths of length 2 from E_1 to E_2

to the power $k = 2$ will give the following result for $\overline{a}_{12}^{(2)}$:-

$$\overline{a}_{12}^{(2)} = a_{11}a_{12} + a_{12}a_{22} + a_{13}a_{32}$$

This shows the involvement of three indirect judgments aggregated together. When viewed on a graph, the three indirect judgments correspond to the three paths from E_1 to E_2, as shown in Fig 6.4. Two of the three paths carry self-comparisons i.e. a_{11} and a_{22}. Similarly for $k = 3$, $\overline{a}_{12}^{(3)}$ can be calculated as:-

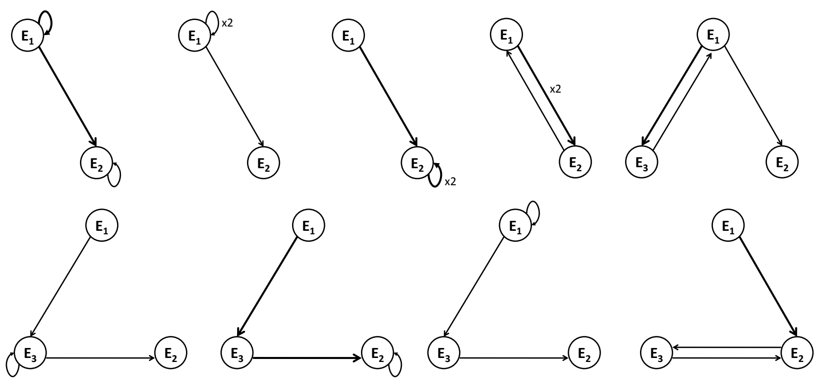


Figure 6.5: Nine paths of length 3 from E_1 to E_2

$$\begin{aligned}\overline{a}_{12}^{(3)} &= a_{12}a_{11}^2 + a_{12}^2a_{21} + a_{12}a_{13}a_{31} \\ &+a_{22}a_{11}a_{12} + a_{22}^2a_{12} + a_{22}a_{13}a_{32} \\ &+a_{32}a_{11}a_{13} + a_{32}a_{12}a_{23} + a_{32}a_{13}a_{33}\end{aligned}$$

Fig. 6.5 shows the paths of length $k = 3$ from E_1 to E_2. There exists nine such paths, however, most have redundant edges including either self-comparisons or reciprocal judgments, or both. As there are no indirect judgments of higher order possible for $n = 3$, raising the matrix to the power $k = 3$ provides no additional information. The paths generated for $k \geqslant n$ will always include redundant links that revisit already traversed nodes.

Harker and Vargas [41] suggested that increasing k reveals more and more of the interactions between alternatives. However, this intuitive justification ("paths of length k") includes self-comparisons. Considering the reciprocal property of judgments, self-comparisons appear redundant in estimating preferences and their inclusion in paths has not been justified.

6.2.2 GM and Indirect Judgments

A comprehensive work relating GM to indirect judgments is found in [117]. Brugha advocated that the GM approach synthesizes every direct judgment with the set of respective indirect judgments. He explained the aggregation of judgments using GM as:-

$$w_{ij} = \left\{ [a_{ij}]^2 \prod_k b_{ij_{(k)}} \right\}^{\frac{1}{n}}$$

where $i, j, k \in \{1, 2, \ldots, n\}$ and $i \neq j \neq k \neq i$. w_{ij} is the ratio between the estimated weights w_i and w_j, for E_i and E_j respectively.

Brugha further claimed that this result can be extended to two or more intermediary variables, apparently resulting in a more complicated formula.

In contrast, it is demonstrated here that indirect paths of higher order get suppressed when geometrically aggregated and are finally reduced to the aggregation of b_{ij}. We show this phenomenon of suppression for $n = 4$. Focusing on the judgment a_{12}, the two indirect judgments are $b_{12_{(4)}} = a_{14}a_{42}$ and

$b_{12_{(3)}} = a_{13}a_{32}$. The geometric aggregation of b_{12} is formulated as:-

$$\begin{aligned}\tilde{b}_{12} &= \left\{\prod_k b_{12_{(k)}}\right\}^{\frac{1}{n-2}} \\ &= \left\{b_{12_{(3)}} b_{12_{(4)}}\right\}^{\frac{1}{2}} \\ &= \{a_{13}a_{32}a_{14}a_{42}\}^{\frac{1}{2}}\end{aligned}$$

Next, the geometric aggregation of second-order indirect judgments can be formulated as:-

$$\tilde{b}_{12}^{(2)} = \left(\prod_{k_1,k_2 \in \{1,2,\ldots,n\}} a_{ik_1}a_{k_1k_2}a_{k_2j}\right)^{\frac{1}{q}} \tag{6.2}$$

where i, j, k_1, k_2 are all unique and q is the total number of second-order indirect paths available.

In the case of $n = 4$, the value of q is equal to 2 and the indirect judgments of second-order are $b^{(2)}_{12_{(3,4)}} = a_{13}a_{34}a_{42}$ and $b^{(2)}_{12_{(4,3)}} = a_{14}a_{43}a_{32}$. The geometric aggregation of these judgments is given below:-

$$\begin{aligned}\widetilde{b_{12}^{(2)}} &= \left\{\prod_{k_1,k_2} b^{(2)}_{12_{(k1,k2)}}\right\}^{\frac{1}{2}} \\ &= \{a_{13}a_{34}a_{42} * a_{14}a_{43}a_{32}\}^{\frac{1}{2}} \\ &= \left\{a_{13}a_{34}a_{42} * a_{14}(\frac{1}{a_{34}})a_{32}\right\}^{\frac{1}{2}} \\ &= \{a_{13}a_{42}a_{14}a_{32}\}^{\frac{1}{2}} = \tilde{b}_{12}\end{aligned}$$

The aggregation of second-order judgments has finally reduced to the aggregation of b_{12}. The same proof can be repeated for all the remaining elements in a PCM with $n = 4$. Provided that the reciprocal property is strictly adhered to, geometric aggregation reduces all the paths of higher orders to b_{ij} (i.e. first-order indirect judgments).

Wang et al. [118] also proposed the use of direct and indirect judgments, however, the formulation is no different to taking the geometric mean of all judgments. Thus, the weights obtained are identical to those generated by the GM (or LLS) method.

6.2.3 Critique

Considering optimization-based prioritization, indirect judgments are not explicitly considered by any existing method. Furthermore, no method exists that simultaneously minimizes deviations from both direct and indirect judgments. In order to estimate preferences, it is sensible to consider both the acquired judgments and the other latent judgments. We propose here a technique to minimize the deviations from both types of judgments.

6.3 Prioritization using Indirect Judgments

In order to estimate preferences, it is proposed to minimize the deviations from indirect judgments along with direct ones, simultaneously. The total deviation for indirect judgments can be formulated as a modification of $d(w)$, which aggregates all the individual deviations from indirect judgments, δ_b i.e.

$$d_b(w) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \delta_b\left(b_{ij_{(k)}}, w\right) \tag{6.3}$$

where $i \neq j \neq k \neq i$.

The individual deviations can be calculated as $\delta_b^{(DLS)}(w) = (b_{ij_{(k)}} - \frac{w_i}{w_j})^2$. δ_b can also be calculated using any other suitable distance function, for example $\delta_b^{(WLS)}(w) = (w_j a_{ik} a_{kj} - w_i)^2$ based on WLS or $\delta_b^{(LLAV)}(w) = |\log a_{ik}a_{kj} - \log w_i + \log w_j|$ based on LLAV.

The optimization problem of prioritization using indirect judgments can be formulated as:-

$$\begin{aligned} minimize \quad & d_{AOF}(w) = (1-\alpha)\, d(w) + \alpha d_b(w) \\ s.t. \quad & \sum_i w_i = 1, w_i > 0, i \in \{1, 2, ..., n\} \end{aligned}$$

where $0 \leqslant \alpha \leqslant 1$.

α gives the DM the flexibility to set the importance of his/her direct and indirect judgments. When $\alpha = 0$, the aggregate objective function $d_{AOF}(w)$ is reduced to the objective function $d(w)$.

The results of the Monte-Carlo experiments suggested including priority violations, $NV(w)$, as the third objective along with the direct and indirect deviations. In order to include $NV(w)$, the optimization problem can be redefined as:-

$$\begin{aligned} minimize \quad & d_{AOF2}(w) = (1-\beta)\left((1-\alpha)\, d(w) + \alpha d_b(w)\right) + \beta NV(w) \\ s.t. \quad & \sum_i w_i = 1, w_i > 0, i \in \{1, 2, ..., n\} \end{aligned}$$

where $0 \leqslant \alpha \leqslant 1$ and $0 \leqslant \beta \leqslant 1$.

The inclusion of β gives the DM flexibility to give different importance to TD and NV, as already suggested in [2].

The importance of the three objectives can be controlled using α and β. When $\alpha = 0$ and $\delta = \delta_{DLS}$, the proposed method becomes identical to TOP and, when both $\alpha = 0$ and $\beta = 0$, the objective gets reduced to $d(w)$ suggesting single-objective optimization.

6.3.1 Applying Multi-objective Optimization

When a set of provided judgments are inconsistent, there exists no single solution that simultaneously minimizes both $d(w)$ and $d_b(w)$. Another approach to solve this optimization problem is to use multi-objective optimization that generates all possible non-dominated solutions. A non-dominated solution is a solution that cannot be declared inferior to any other solution. In such case, aggregated objective functions like $d_{AOF}(w)$ and $d_{AOF2}(w)$ are no longer required. This technique of prioritization using indirect judgments (PrInT) can be formulated as:-

$$\begin{aligned} minimize \quad & [d(w),\, d_b(w),\, NV(w)]^T \\ s.t. \quad & \sum_i w_i = 1, w_i > 0, i \in \{1, 2, ..., n\} \end{aligned}$$

The generated priority vectors using PrInT are then provided to the DM to select one according to his/her requirements. Each non-dominated solution can also be achieved using the previously discussed technique using d_{AOF2}, as it will have related unique values of α and β.

This approach gives greater flexibility to the DM; however, it also demands more computation time considering the generation of all non-dominated solutions. Mikhailov and Knowles [31] proposed the use of evolutionary algorithms for the TOP method to generate all non-dominated solutions. Evolutionary methods are considered superior to other methods for non-convex objective functions as they do not get trapped in local optima. We also adopt an evolutionary approach to find all optimal solutions, considering the three objectives.

In the next section, we investigate possible benefits of the proposed technique and compare it to existing methods.

6.4 Illustrative Examples

To illustrate the use of the proposed technique involving indirect judgments, four different types of PCMs are discussed below. The weights are first estimated using the EV, GM, DLS, WLS, LLAV and TOP methods for each of the four PCMs. The Pareto-optimal front is then generated using PrInT with a fixed population of 100 non-dominated solutions. To compare the weights generated by the different methods, we calculate TD, NV and the deviation from second-order indirect judgments, termed TD2 hereafter.

6.4.1 Consistent PCM

Consider an example of a consistent PCM given below:-

$$A_1 = \begin{bmatrix} 1 & 2 & 4 & 12 \\ \frac{1}{2} & 1 & 2 & 6 \\ \frac{1}{4} & \frac{1}{2} & 1 & 3 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{3} & 1 \end{bmatrix}$$

A_1 is a set of fully consistent judgments and is included to demonstrate correctness of the proposed algorithm in the error-free case. The set of all indirect judgments calculated for A_1 are congruent (identical) to the direct judgments. Therefore, all the prioritization methods generate the same weights i.e. $w = \begin{bmatrix} 0.545 & 0.273 & 0.136 & 0.045 \end{bmatrix}$.

6.4.2 Transitive PCM

Consider the following inconsistent PCM with $CR < 0.1$ and $L = 0$:-

$$A_2 = \begin{bmatrix} 1 & 2\frac{1}{2} & 4 & 9\frac{1}{2} \\ \frac{2}{5} & 1 & 3 & 6\frac{1}{2} \\ \frac{1}{4} & \frac{1}{3} & 1 & 5 \\ \frac{2}{19} & \frac{2}{13} & \frac{1}{5} & 1 \end{bmatrix}$$

Method	w_1	w_2	w_3	w_4	*NV*
EV	.5328	.2872	.1390	.0410	0
GM	.5350	.2864	.1377	.0409	0
DLS	.4742	.3134	.1648	.0475	0
WLS	.5613	.2616	.1280	.0490	0
LLAV	.4966	.3287	.1241	.0505	0

Table 6.2: Preference weights estimated for A_2

The judgments in A_2 are inconsistent and therefore, the indirect judgments are incongruent to the direct ones. Although different, the order of *preference dominance* remains the same, as there is no judgment that suggests preference reversal. CR for A_2 is equal to 0.045, suggesting that the PCM is acceptable in AHP terms. The results given in Table 6.2 show $NV = 0$ for all the methods as expected.

We now compare the results in the objective space i.e. TD vs TD2. Fig. 6.6 shows a complete set of Pareto-optimal solutions produced by PrInT, where the DLS solution can be seen as one extreme of the optimal front. Interestingly, both the EV and GM solutions appear to be non-dominated. The LLAV and WLS methods appear to be dominated by other solutions. As there are no priority violations, TOP generates a single solution, which is the same as the DLS solution.

6.4.3 Transitive PCM with Latent Violations

As already discussed, there exists a possibility of $NV \neq 0$ for a transitive PCM. Such an example is considered below:-

$$A_3 = \begin{bmatrix} 1 & 3 & 2 & 6 \\ \frac{1}{3} & 1 & 1\frac{1}{5} & 2 \\ \frac{1}{2} & \frac{5}{6} & 1 & 3 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 1 \end{bmatrix}$$

The value of $CR = 0.0161$ confirms that A_3 is an *acceptable* set of judgments, however, considering the indirect judgments, we note a potential violation between E_2 and E_3. The direct judgment for a_{23} is 1.2, however, the two indirect judgments suggest 0.66, causing preference reversal while estimating weights.

The generated solutions are given in Table 6.3. The solutions for DLS, WLS,

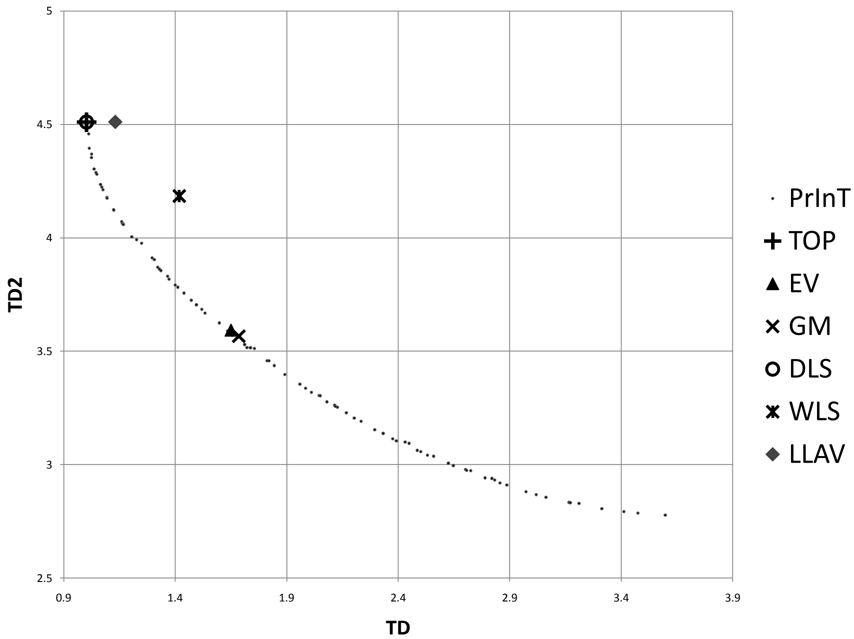


Figure 6.6: Preference weights for A_2 in objective space

EV and GM generate one priority violation despite lying on the Pareto-optimal front in the TD-TD2 objective plane (see Fig. 6.7). TOP produces one additional solution (apart from DLS) minimizing NV to zero. In contrast, the Pareto-optimal front contains 17 solutions with zero violations - all of which were generated by PrInT. These solutions are plotted in Fig. 6.7.

6.4.4 Intransitive PCM

Next , consider an *intransitive* set of PC judgments:-

$$A_4 = \begin{bmatrix} 1 & 2 & \frac{3}{4} & 1\frac{1}{2} \\ \frac{1}{2} & 1 & 1\frac{1}{2} & 3 \\ \frac{4}{3} & \frac{2}{3} & 1 & 2 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix}$$

A_4 has two three-way cycles present, $E_1 \rightarrow E_3 \rightarrow E_4 \rightarrow E_1$ and $E_2 \rightarrow E_3 \rightarrow E_4 \rightarrow E_2$. The results given in Table 6.4 show that the minimum possible value for NV is 1, obtained by TOP. WLS produces most violations (i.e. NV=3), whilst all other methods give two priority violations (i.e. NV=2).

An interesting outcome of having two Pareto-optimal curves is visible in Fig. 6.8, where each is related to a different value of NV. The optimal front with

Method	w_1	w_2	w_3	w_4	*NV*
EV	.5017	.1961	.2185	.0836	1
GM	.5039	.1945	.2175	.0839	1
DLS	.5015	.1716	.2435	.0833	1
WLS	.5054	.1795	.2314	.0836	1
LLAV	.4999	.1667	.2499	.0833	1
TOP	.5045	.2059	.2059	.0836	0
TOP	.5015	.1716	.2435	.0833	1
PrInT	.5038	.1798	.2331	.0833	0
PrInT	.5024	.1962	.2184	.0830	0
PrInT	.5055	.1876	.2233	.0836	0
PrInT	.5094	.2003	.2062	.0841	0

Table 6.3: Preference weights estimated for A_3

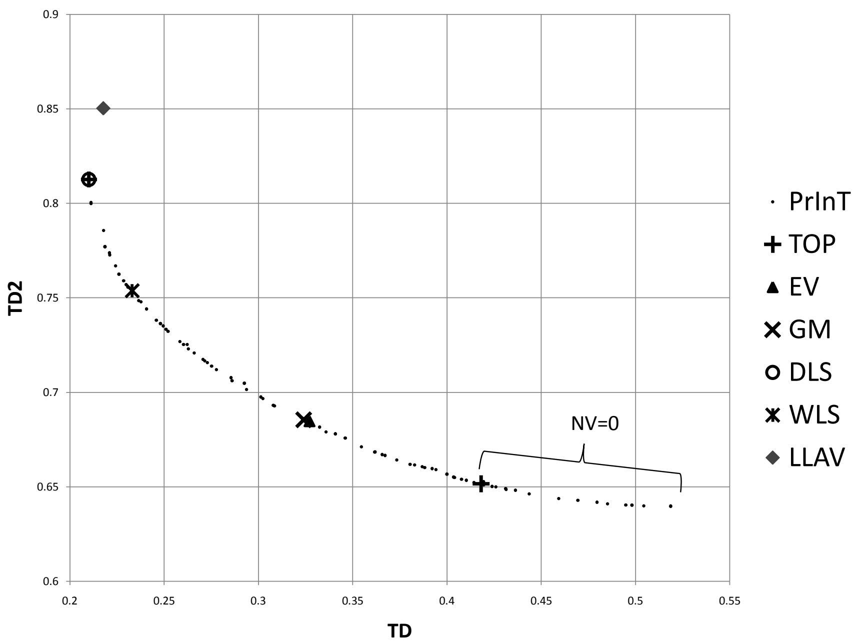


Figure 6.7: Preference weights for A_3 in objective space

Method	w_1	w_2	w_3	w_4	*NV*
EV	.3166	.2174	.2144	.2515	2
GM	.3193	.2294	.2038	.2475	2
DLS	.3631	.2283	.1623	.2463	2
WLS	.3346	.2424	.1834	.2395	3
LLAV	.3333	.2222	.1481	.2962	2
TOP	.3175	.2211	.1438	.3175	1
TOP	.3631	.2283	.1623	.2463	2

Table 6.4: Preference weights estimated for A_4

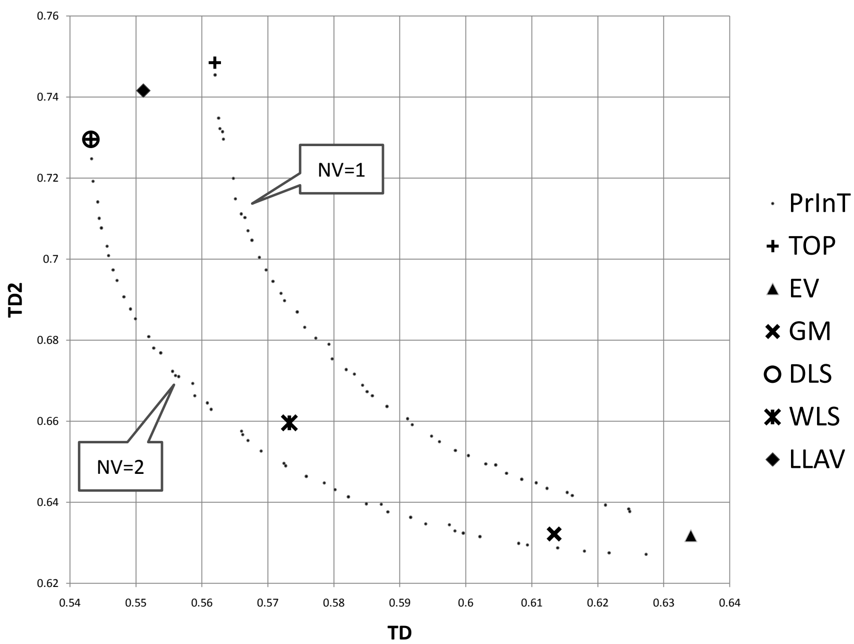


Figure 6.8: Preference weights for A_4 in objective space

NV=2 shows 51 non-dominated solutions. The other set related to NV=1 has 49 non-dominated solutions that also includes a TOP solution. In the presence of intransitivity, both EV and GM solutions are dominated by those obtained by TOP and PrInT.

From the results discussed above, it is evident that the new approach clearly offers the DM a wide range of non-dominated solutions, giving him/her the flexibility to select one according to his/her requirements. This approach outperforms all other methods for intransitive PCMs, giving solutions with minimum violations that are as close as possible to the direct and indirect judgments.

6.5 Summary

The problem of obtaining preferences from PCs has been explored by considering indirect judgments. The effect of intransitive judgments on the priority violations has been investigated. The results, calculated from thousands of random PCMs generated using Monte-Carlo simulation, confirm a relationship between the cyclic judgments and the average minimum number of violations.

A new method based on multi-objective optimization has been proposed to minimize deviation from both direct and indirect judgments, along with priority violations. The approach offers multiple non-dominated solutions from which the DM may select, according to his/her requirements. This approach outperforms all other methods for intransitive PCMs, giving solutions with minimum violations whilst remaining as close as possible to both the direct and indirect judgments.

Chapter 7

PriEsT - A Priority Estimation Tool

This chapter presents a priority estimation tool (PriEsT) as a decision support tool offering several new features. These features include the visualization of inconsistency discussed in chapter 4, and the prioritization methods proposed in chapters 5 and 6. In order to show the benefits of PriEsT, a case study involving Telecom infrastructure selection is presented.

7.1 Requirements

PriEsT has been developed as a decision support tool based on Analytic Hierarchy Process (AHP). There exist software tools based on pairwise comparison (PC), for example, ExpertChoice [119] and HIPRE [120]. However, as discussed in previous chapters, they lack the visualization of inconsistency among given judgments. This has been the first consideration whilst developing the new tool. PriEsT has the ability to assist decision makers (DMs) in revising their judgments based on the *congruence* and *dissonance* measures.

Secondly, PrInT offers multiple equally-good solutions, unlike other tools offering single solution. PriEsT implements the proposed technique of PrInT offering a wide range of Pareto-optimal solutions. The DM has the flexibility to select any of these non-dominated solutions according to his/her requirements.

So far, PriEsT has been developed as a prototype; the future aim is to develop PriEsT in accordance with ISO/IEC 9126. ISO/IEC 9126 is an international standard for the evaluation of software that classifies the quality of software in the set of six characteristics: *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* and *portability* [121].

The requirements for PriEsT are grouped into various categories. Each category (or group) is assigned a group-code and the requirements are sequentially numbered inside these categories. The codes for functional requirements are prefixed with FR and non-functional requirements with NR.

These requirements are listed below. The use of future tense implies that the requirements were written prior to the design and development.

7.1.1 Functional Requirements

7.1.1.1 Model

FR-01-001 Editable Criteria PriEsT will allow the creation of a list of measurable criteria, $\{C_1, C_2, ..., C_n\}$, that will be considered for comparing alternatives. This requirement is related to the second stage of the MCDM process (see Section 2.1.1). User will be allowed to add, change and remove criteria from a model.

FR-01-002 Editable Alternatives The addition and removal of alternatives, $\{A_1, A_2, ..., A_n\}$, will be supported. This creates the lowest level of hierarchy in AHP terms. Each alternative will have a unique name to avoid confusion when acquiring judgments.

FR-01-003 Editable Agents (Group Decision Support) The addition and removal of agents (DMs) will also be supported. This is required in order to support group decision making where judgments are acquired from several DMs.

FR-01-004 Hierarchical View As AHP is the main focus of this research, the DM will be allowed to edit and visualize criteria in a hierarchical structure.

FR-01-005 Preference Equivalence There exist two types of situations in practice; either ties are allowed or to be avoided. Hence, there will be a provision of enabling/disabling the use of preference equivalence.

7.1.1.2 Data Acquisition

FR-02-001 Table View In order to enter PC matrix (PCM), a table view will be offered to DMs where judgments will be entered, changed or removed. For each given judgment a_{ij}, the reciprocal judgment a_{ji} will automatically be updated.

FR-02-002 Graph View A graph view will be offered to DMs in order to highlight transitive relations among stimuli (*criteria* or *alternatives*). The judgments will be depicted as directed edges and preference equivalence will be shown as a dotted line.

FR-02-003 Judgment Scale Indirect judgments are best visualized when plotted on a judgment scale along with direct judgment. Therefore, this type of view will also be supported in PriEsT.

FR-02-004 Web Service In the case of group decision making, support will be added for remote entry of judgments. This will be provided in the form of a Web Service.

7.1.1.3 Visual Aids

FR-03-001 Individual Congruence The *congruence matrix* will be displayed in the form a bar graph, depicting the contribution of each judgment to overall inconsistency.

FR-03-002 Individual Dissonance Similar to congruence, the *dissonance matrix* will also be displayed to highlight the contribution of each judgment towards ordinal inconsistency.

FR-03-003 Most Inconsistent Triple It will be useful to highlight the most inconsistent triple of judgments in the given PCM. This will be calculated using (2.29) and (2.30).

FR-03-004 Three-way Cycles The presence of a three-way cycle will be highlighted in the Graph view (FR-02-002). In case of multiple three-way cycles, a list of selectable buttons will be offered to view each cycle individually.

FR-03-005 Consistency Measures The values for different consistency measures will be shown to the user. The list of measures includes *Consistency Ratio* (CR), *Consistency Measure* (CM), *number of three-way cycles* (L), *congruence* (Θ) and *dissonance* (Ψ).

7.1.1.4 Preference Elicitation

FR-04-001 Selecting Prioritization Method The DM will have the flexibility to select any available prioritization method. Along with the newly proposed ones, widely used methods will also be implemented for this purpose.

FR-04-002 Visualizing Priority Vectors The numeric values for each priority vector will be shown in the form of a table, along with the method used. In addition to this, the objectives of *total deviation from direct judgments* (TD), *indirect total deviation from indirect judgments* (TD2) and *number of priority* *violations* (NV) will also be given in numeric form.

FR-04-003 Objective Space TD-NV In order to compare the priority vectors generated by different methods, the vectors will be plotted in TD-NV space. A unique marker will be assigned to each method for evaluation purposes.

FR-04-004 Objective Space TD-TD2 The generated priority vectors will also be plotted in the TD-TD2 space. The DM will have the flexibility to choose any one of the generated solution as the final priority vector.

FR-04-005 Overall Ranking An overall ranking will be calculated and shown to the user. PriEsT will highlight missing judgments where the data is insufficient to obtain overall ranking.

7.1.1.5 Other Requirements

FR-05-001 XML Import/Export PriEsT data will be saved to a file in XML format. PriEsT will also be able to re-load the saved file. The saved file will contain the model, judgments and the estimated preferences.

FR-05-003 Web Application A web front-end for PriEsT will also be developed. This requirement suggests PriEsT to be developed using a three-tier architecture i.e. database server, application server and a front-end application.

7.1.2 Non-Functional Requirements

NR-01-001 Platform PriEsT will be developed using cross-platform tools in order to support as many operating systems as possible.

NR-01-002 Performance The basic functionality of PriEsT demands a current computer of average specification. However, the use of Evolutionary Algorithms for PrInT suggests more demanding hardware specifications. The initial target will be to run PriEsT on Intel-based computer with Core2Duo T5500 CPU running at 1.66GHz.

NR-01-003 Portability Based on the three-tier architecture (FR-05-003), the user front-end module will be designed for web-pages and also as a mobile phone application.

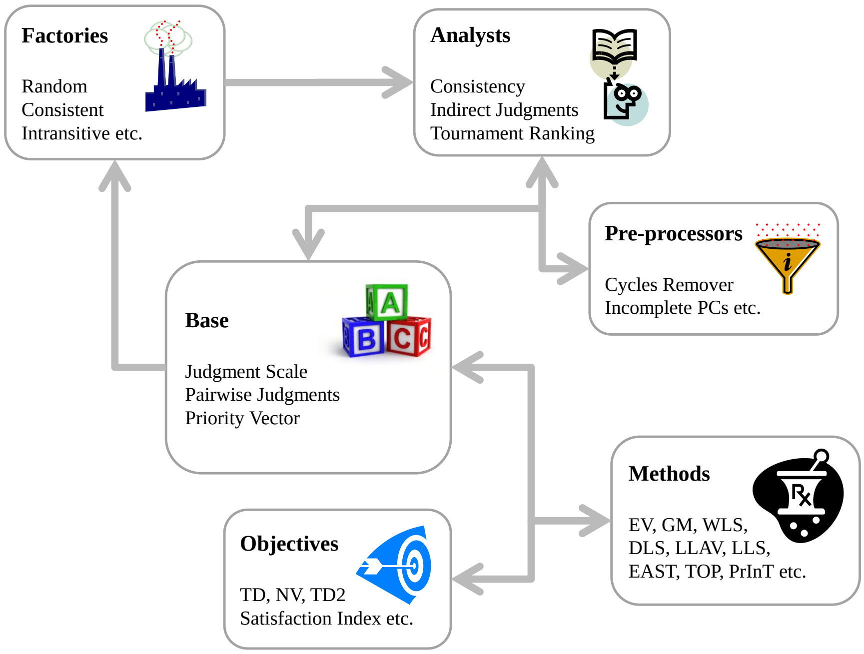


Figure 7.1: Overview of the PriEsT engine

7.2 Software Design & Development

The core of the PriEsT software, termed the *engine*, has been designed to be independent of user-interface libraries. The front-end has then been built on top of this engine.

7.2.1 PriEsT Engine

The engine consists of several building blocks, shown in Fig. 7.1. These blocks are briefly discussed below:-

7.2.1.1 Base

This block consists of the basic classes required to support pairwise comparisons i.e. *PC* (for Pairwise Judgments), *JudgmentScale* (for Measurement Scales) and *W* (for Priority Vectors).

7.2.1.2 Factories

A set of factory classes generate PCMs with different properties e.g. consistent, intransitive, acceptable etc. In addition to this, *PersistentFactory* allows save and/or load of PCMs from text files (serialization).

7.2.1.3 Analysts

During this research, several different properties of PCMs have been analyzed. This block contains the code written for analysis of PCMs: *ConsistencyAnalyzer* calculated Eigenvalues, CR and CM for a given PCM; *IndirectAnalyzer* is useful for calculating θ and ψ based on indirect judgments; the *TournamentAnalyzer* class calculates three-way cycles and Kendall's ζ.

7.2.1.4 Pre-processors

The pre-processors cover possible pre-processing of PCMs before prioritization. The *CyclesRemover* class suggests the removal of intransitive judgments by implementing the heuristic algorithm proposed in Chapter 4. The class to estimate missing judgments in an incomplete PCM has not been implemented yet. This class will be useful for the Eigenvector (EV) and Geometric Mean (GM) methods for incomplete set of judgments.

7.2.1.5 Methods

All prioritization methods are implemented in this block of code. EV, GM, *normalized column sum* (NCS) and EAST have been implemented in separate classes, whilst all the optimization-based algorithms use an *Optimization* class which is implemented using jMetal toolkit. Each optimization algorithm provides an objective function from the set of available objectives (to be discussed next). The TOP and PrInT methods use multiple objective functions simultaneously.

7.2.1.6 Objectives

This block implements all the objective functions studied in this research. This includes TD, NV, TD2, logarithmic deviations and absolute errors. Satisfaction index is also calculated to evaluate results for the *Fuzzy Preference Programming* (FPP) method.

7.2.2 Front-end Application

The user-interface of PriEsT Engine was developed using the Qt framework. Qt is an open-source cross-platform software development kit (SDK) for writing C++ applications. However, we have used Qt library in Java with the help of QtJambi (a bridge between Java and C++ for Qt SDK). The Qt SDK was chosen in reference to the requirement no. NR-01-001.

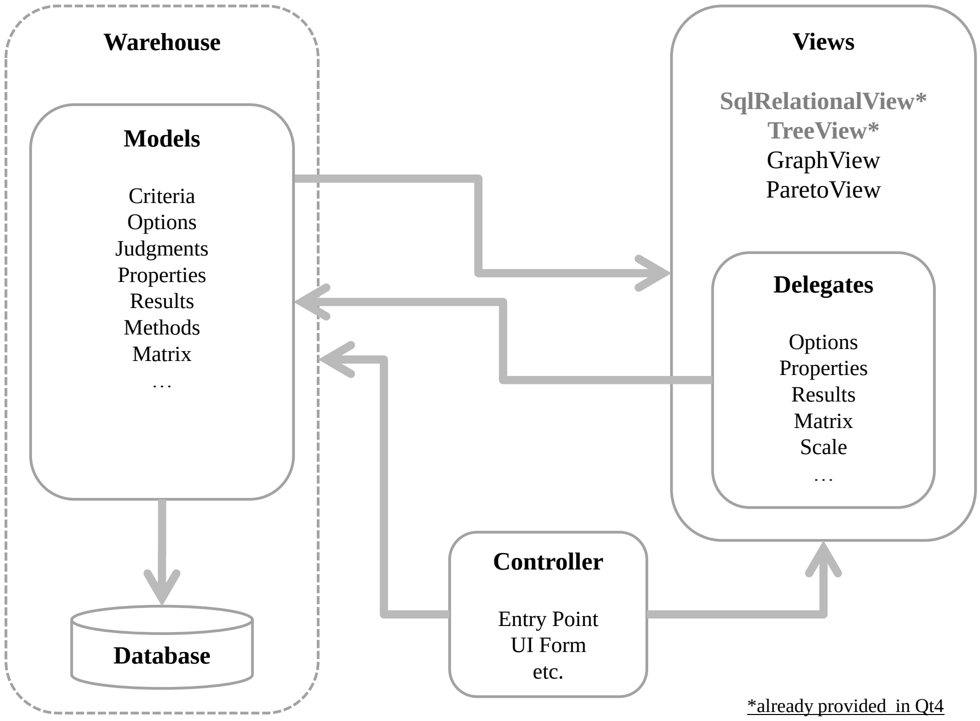


Figure 7.2: Model-View-Controller architecture for PriEsT Application

The application is based on the Model-View-Controller (MVC) architecture. Each "View" class has an associated "Delegate" class to communicate with its respective Model. All the data is ultimately preserved in a relational database. The structure of the MVC architecture of PriEsT is given in Fig. 7.2.

7.2.3 Development

The PriEsT engine has been developed in NetBeans IDE. A user interface designer for Qt is not available for NetBeans therefore Eclipse has been used for the user-interface development.

Each class was implemented according to the requirements and then unit-tested. Once all the major classes were implemented, system testing was performed with a written test-plan.

Table 7.1 shows the results of major tests performed against the requirements for PriEsT. Group decision making has not yet been implemented therefore the two tests T1-0003 and T1-0008 have not been passed. In addition to that, the two requirements, FR-05-002 and NR-01-003, related to front-end applications for web and phones have also not yet been developed. Considering

Test No.	*Requirements Covered*	*Test Details*	*Result*
T1-0001	FR-01-001 FR-01-004	Editable criteria and hierarchical view	ok
T1-0002	FR-01-002	Editable alternatives	ok
T1-0003	FR-01-003	Editable agents for group decisions	not supported
T1-0004	FR-01-005	Preference equivalence enable/disable	ok
T1-0005	FR-02-001	Table view for entry	ok
T1-0006	FR-02-002	Graph view for entry	read-only
T1-0007	FR-02-003	Entering judgments on measurement scale	read-only
T1-0008	FR-02-004	Entering judgments remotely using Web service	not supported
T1-0009	FR-03-001, FR-03-002	Show congruence & dissonance matrices	ok
T1-0010	FR-03-005	Display CR, CM, L, Θ and Ψ	ok
T1-0011	FR-03-003	Highlighting the most inconsistent triple	ok
T1-0012	FR-03-004	Highlighting all three-way cycles	ok
T1-0013	FR-04-001	Selecting different prioritization methods	ok
T1-0014	FR-04-002	Visualizing priority vectors	ok
T1-0015	FR-04-003	Plotting results in the TD vs. NV space	ok
T1-0016	FR-04-004	Plotting results in the TD vs. TD2 space	ok
T1-0017	FR-04-005	Finding overall ranking using Additive AHP	ok
T1-0018	FR-05-001	Exporting data to an XML file	ok
T1-0019	FR-05-001	Importing saved data from XML	ok
T1-0020	FR-05-002	Web-based application	not supported
T1-0021	NR-01-001	Cross-platform working Run on both Windows XP and Linux	ok
T1-0022	NR-01-002	Run PriEsT on Intel-based PC with Core2Duo T5500	acceptable
T1-0023	NR-01-003	Front-end: Web and Phone Applications	not supported

Table 7.1: System testing performed for PriEsT

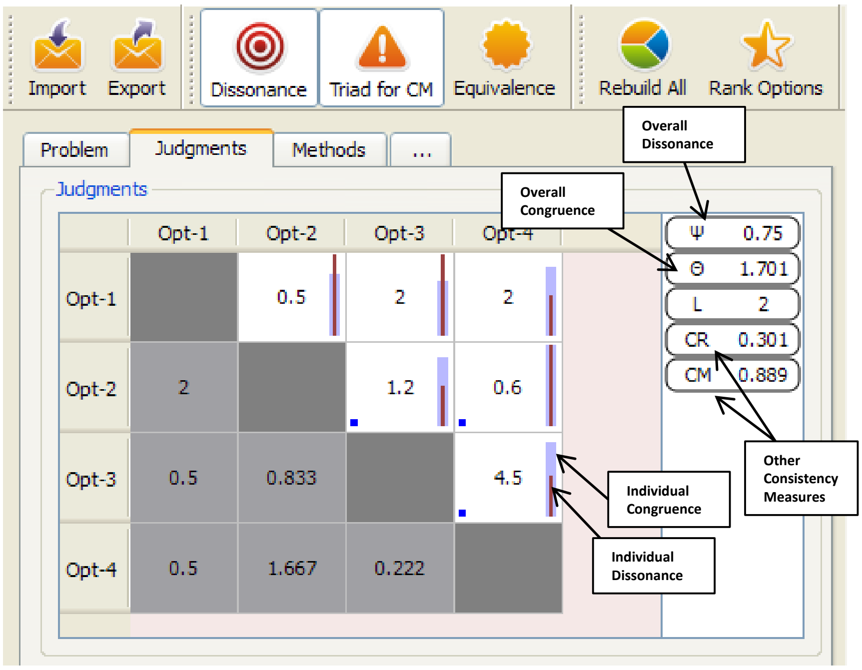


Figure 7.3: Visualizing Inconsistency in Table View

the topic of this research, these requirements are of secondary importance. The research-related requirements have been implemented and successfully tested.

7.3 PriEsT Features

PriEsT offers useful new features that have not been considered by other tools for pairwise comparisons. These salient features of PriEsT are discussed below.

7.3.1 Decision Aid

7.3.1.1 Visualizing Inconsistency in Table View

A way to visualize inconsistencies present in individual judgments was discussed in Chapter 3. This feature has been practically demonstrated in PriEsT where the DM is assisted in improving provided judgments. Fig. 7.3 shows the dissonance matrix plotted as bar graphs against their respective judgments. The most inconsistent triple (set of three judgments) is also shown with the help of blue dots on the blamed judgments.

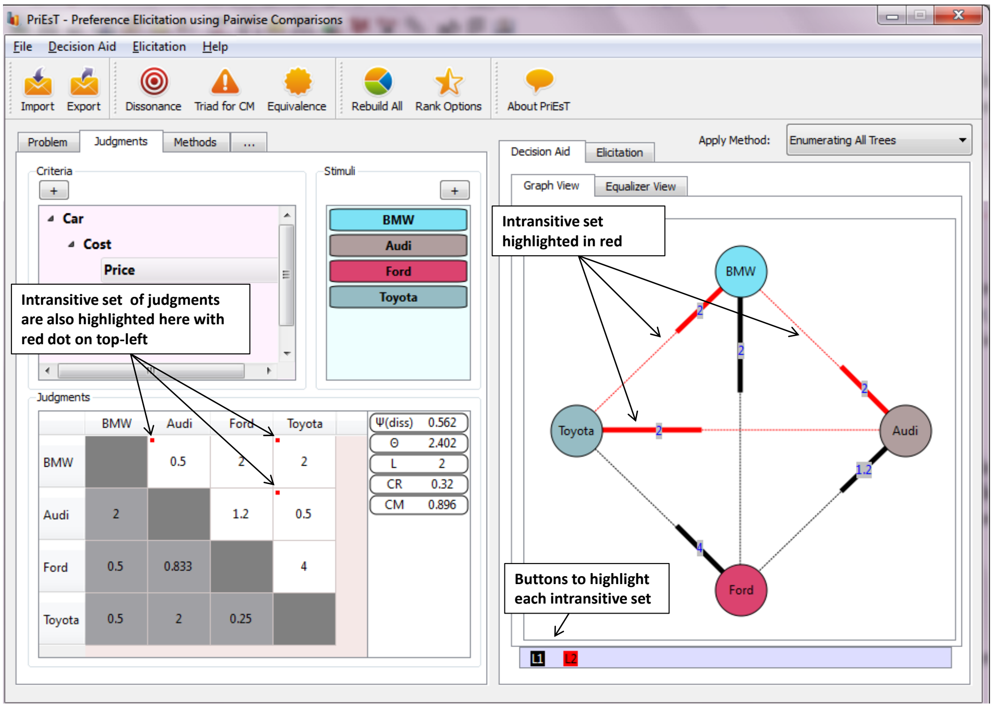


Figure 7.4: Graph View for Intransitive Set of Judgments

7.3.1.2 Graph View

The use of graph view has enormous potential in helping experts to analyze acquired judgments. The graph view proves helpful in visualizing intransitive judgments. In PriEsT, the presence of three-way cycles are highlighted in red, as shown in Fig 7.4. Along with three-way cycles, the most inconsistent triple can also be shown on the graph view.

7.3.1.3 Judgment Scale (Equalizer View)

Plotting all judgments on a measurement scale has been found useful to analyze inconsistency between direct and indirect judgments. As discussed in Chapter 3, this helps in visualizing cognitive dissonance present in the set of provided judgments. Fig. 7.5 shows how this aids the DM in finding the potential cause of priority violations.

7.3.2 Elicitation

Users of PriEsT are allowed to select different prioritization methods to estimate preferences from the same set of judgments. PriEsT therefore qualifies as an appropriate research tool to evaluate such methods.

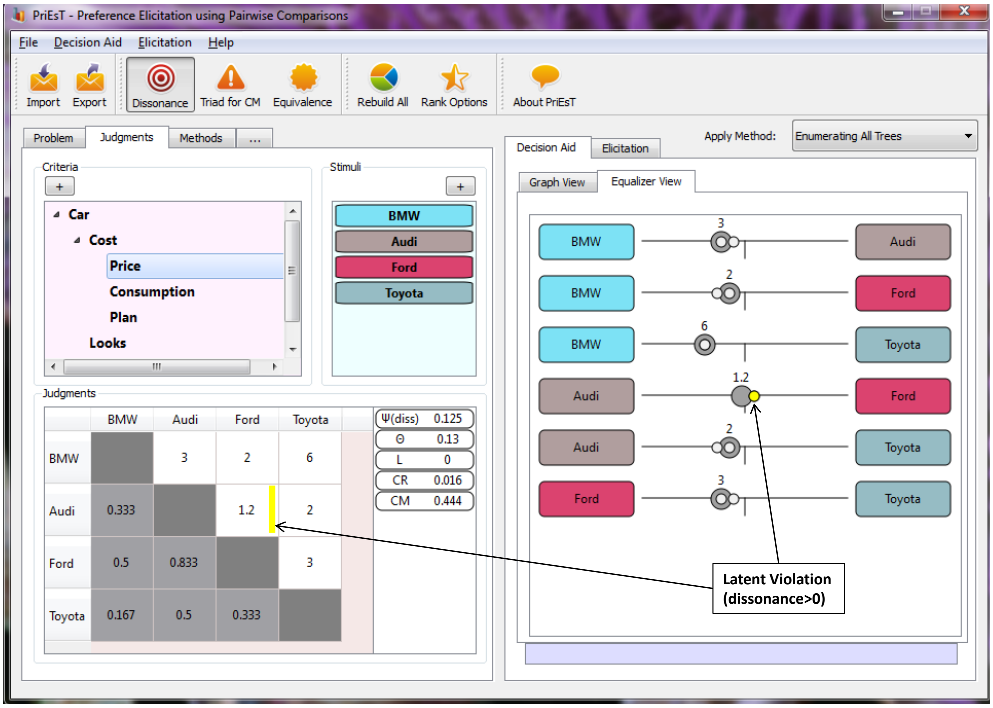


Figure 7.5: Dissonance Visible in Equalizer View

7.3.2.1 List of Solutions and Gantt View

The solutions generated by different methods are displayed as a list containing all numerical values of the generated weights. An alternative option is also provided for users to view the generated weights in the form of a Gantt chart. A method producing a different set of ranking can easily be spotted when viewed as Gantt chart. An example is shown in Fig. 7.6 where all the solutions have produced same ranking except the last solution generated by the TOP method. A unique color has been assigned to each element (or stimulus) for the purpose of identification. The last solution suggests $E_1 \to E_2 \to E_3 \to E_4$ while all the other solutions have suggested $E_1 \to E_3 \to E_2 \to E_4$.

7.3.2.2 Objective Space

Along with TD, the need to minimize NV and TD2 has been established in the previous chapter. PriEsT offers the DMs an interactive selection of any non-dominated solution by plotting them on two different objective spaces. The first is TD-NV space as shown in Fig. 7.7a, while the second is TD-TD2 space, proposed in chapter 6 (shown in Fig. 7.7b).

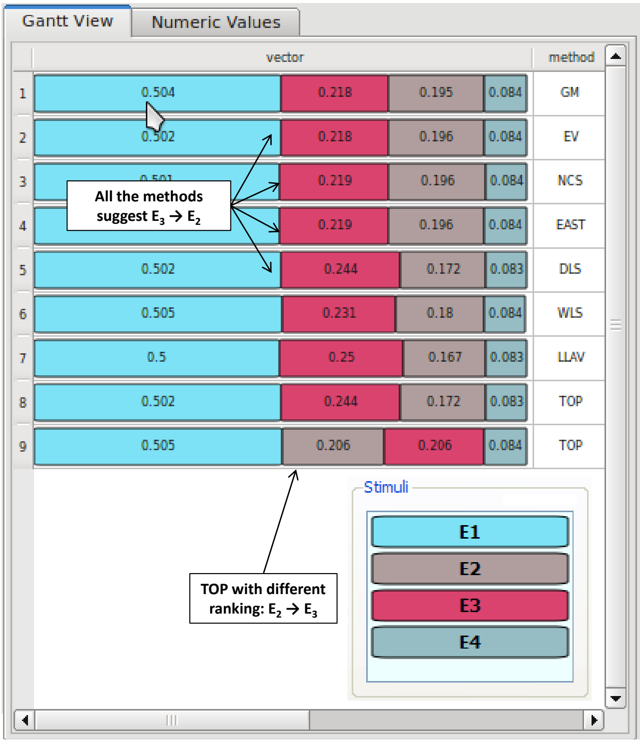
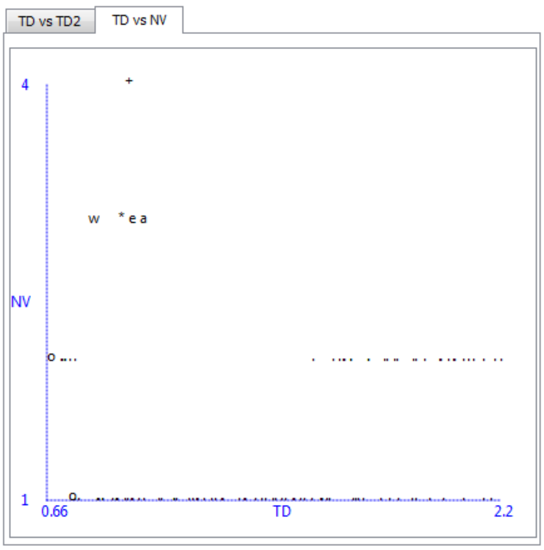
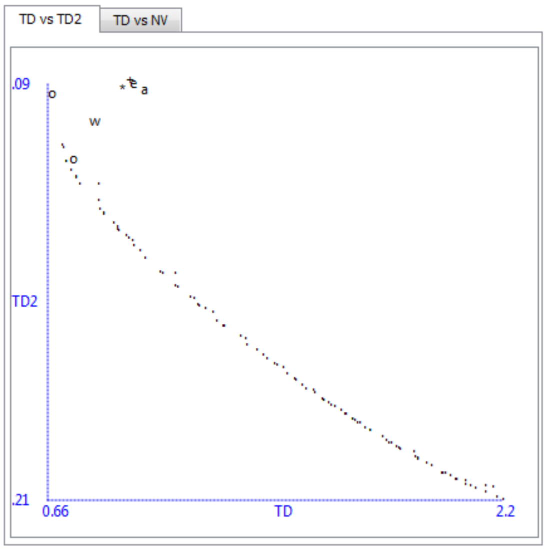


Figure 7.6: Gantt view for generated weights



(a) TD vs. NV



(b) TD vs. TD2

Figure 7.7: Visualizing Solutions in Objective Space

7.3.3 Other features

The use of XML format enables the integration of PriEsT with other tools and web technologies without doing major changes in its architecture. The use of XML also allows integration with spreadsheet applications (e.g. Microsoft Excel) and the importing of data from other software tools.

7.4 Case Study: Telecom Backbone Selection

In order to demonstrate the utility of the features of PriEsT, consider the practical data acquired in a recent study: selecting a backbone infrastructure for telecommunication in rural areas [122]. The research was primarily focused on the rural areas of developing countries, where the lack of adequate telecommunications infrastructure remains a major obstacle for providing affordable services in rural areas. The Analytic Hierarchy Process (AHP) and Analytic Network Process (ANP) have been used to model this complex problem.

The four alternatives discovered were Fiber-optic cable (G1), Power-line communication (G2), Microwave link (G3) and Satellite communication (G4). The problem was solved using AHP first and then by reformulating it for ANP.

The criteria used to compare these alternatives were grouped into six major categories including technical, infrastructural, economic, social, regulatory and environmental factors. These categories and their constituent criteria are presented in Fig. 7.8. The PCM acquired for prioritizing these six categories (top-level criteria) is given below:-

$$A_{top} = \begin{bmatrix} 1 & 3.08 & 1.32 & 3.08 & 4.53 & 2.94 \\ .325 & 1 & 4.09 & 2.94 & 3.16 & 3.56 \\ .758 & .244 & 1 & 4.09 & 7.94 & 8.49 \\ .325 & .34 & .244 & 1 & 1.19 & 1.19 \\ .221 & .316 & .126 & .84 & 1 & 1.19 \\ .34 & .281 & .118 & .84 & .84 & 1 \end{bmatrix}$$

Although A_{top} is a transitive PCM, the estimated vectors produce a priority violation (NV=1) when using EV or GM.

Most criteria lie under the *Technical* and *Infrastructure* categories. The *Technical* category includes nine criteria whilst the *Infrastructure* category has eight criteria used to compare the alternatives. The acquired PCMs for these two

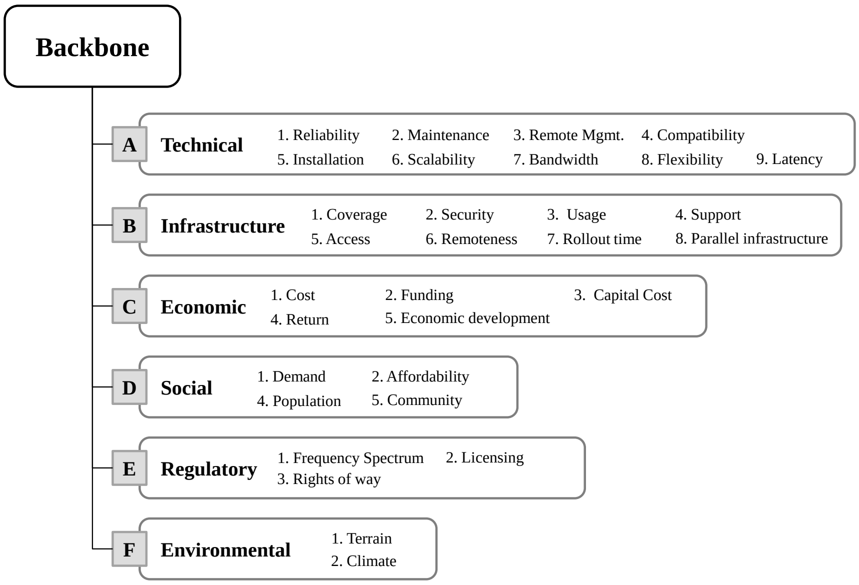
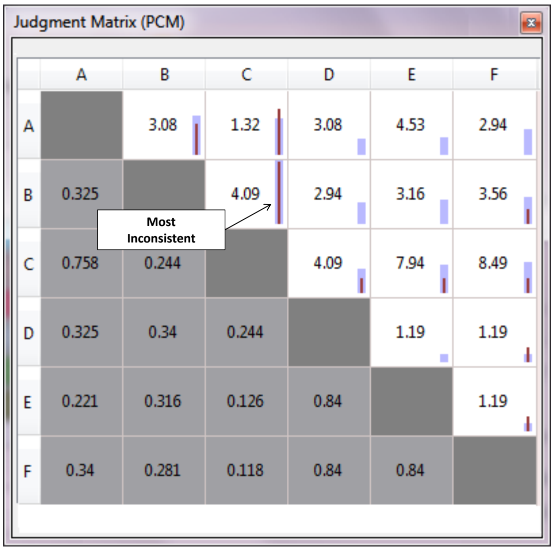


Figure 7.8: Criteria to compare the available backbone infrastructures

categories are given below:-

$$A_{tech} = \begin{bmatrix} 1 & 1.22 & 2.91 & 3.23 & 1.82 & 6.05 & 0.98 & 4.9 & 9.67 \\ .82 & 1 & 1.27 & 5.34 & 3.68 & 4.33 & 1 & 6.96 & 2.91 \\ .344 & .787 & 1 & 1.39 & 1.09 & 5.01 & 0.39 & 5 & 8 \\ .31 & .187 & .719 & 1 & 0.56 & 1.19 & 0.35 & 1.19 & 7 \\ .549 & .272 & .917 & 1.786 & 1 & 2.87 & 0.45 & 8.74 & 6.7 \\ .165 & .231 & .2 & .84 & .348 & 1 & 0.18 & 1.79 & 2.01 \\ 1.02 & 1 & 2.564 & 2.857 & 2.222 & 5.556 & 1 & 3.06 & 5.06 \\ .204 & .144 & .2 & .84 & .114 & .559 & .327 & 1 & 2.99 \\ .103 & .344 & .125 & .143 & .149 & .498 & .198 & .334 & 1 \end{bmatrix}$$

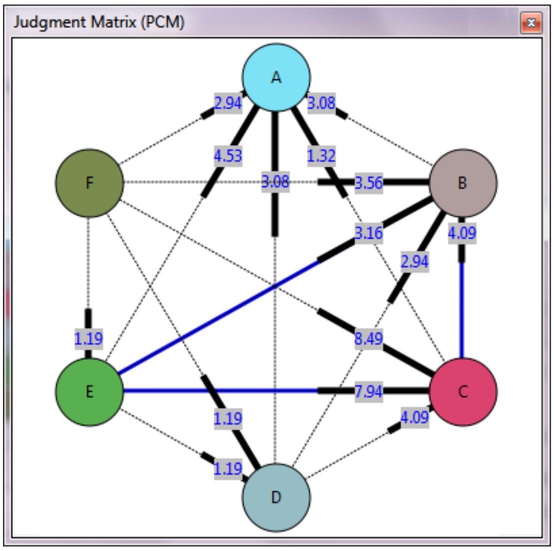
$$A_{infra} = \begin{bmatrix} 1 & 5.18 & 9 & 6.96 & 5.18 & 2.23 & 6.65 & 7.17 \\ .193 & 1 & 6.88 & 3.31 & .89 & .2 & 3.08 & .19 \\ .111 & .145 & 1 & .64 & .24 & .32 & .19 & .2 \\ .144 & .302 & 1.562 & 1 & .38 & 1.67 & 1 & .32 \\ .193 & 1.124 & 4.167 & 2.632 & 1 & .46 & 1.97 & .57 \\ .448 & 5 & 3.125 & .599 & 2.174 & 1 & 3 & 1.19 \\ .15 & .325 & 5.263 & 1 & .508 & .333 & 1 & .46 \\ .139 & 5.263 & 5 & 3.125 & 1.754 & .84 & 2.174 & 1 \end{bmatrix}$$



Judgment Matrix (PCM)

	A	B	C	D	E	F
A		3.08	1.32	3.08	4.53	2.94
B	0.325		4.09	2.94	3.16	3.56
C	0.758	0.244		4.09	7.94	8.49
D	0.325	0.34	0.244		1.19	1.19
E	0.221	0.316	0.126	0.84		1.19
F	0.34	0.281	0.118	0.84	0.84	

(a) Table-view



(b) Graph-view

Figure 7.9: Table-view and Graph-view for A_{top}

The final weights calculated using the EV and GM methods are found to be almost identical, as given below in normalized form:-

$$\textbf{EV:} \quad w_{G1} = 21.63\%, w_{G2} = 20.10, w_{G3} = 28.34\%, w_{G4} = 29.95\%$$
$$\textbf{GM:} \quad w_{G1} = 21.62\%, w_{G2} = 20.07, w_{G3} = 28.36\%, w_{G4} = 29.73\%$$

Satellite communication (G4) is considered the most preferred alternative with a weight of 29.95% (using EV), followed by Microwave (G3) with a weight around 28.34% (using EV).

A_{tech} and A_{infra} are intransitive PCMs that should be investigated along with A_{top} for their impact on the final result.

7.4.1 Investigation using PriEsT

The three matrices, A_{top}, A_{tech} and A_{infra} have been analyzed using PriEsT. The matrices were analyzed using both PriEsT's table-view and its graph-view.

7.4.1.1 A_{top}

The two views for A_{top} are shown in Fig. 7.9. Fig. 7.9a is a snapshot of the PCM when viewed as a table and the graph view is shown in Fig. 7.9b. The labels A to F in these figures correspond to the labels mentioned in Fig. 7.8.

The CR for this PCM is equal to 0.136 and therefore unacceptable in AHP terms. The contribution of each judgment towards overall inconsistency is visible in the table view. The most inconsistent judgment according to the *congruence* and *dissonance* measures is determined to be $a_{23} = 4.09$. The graph view helps to highlight the most inconsistent set of judgments. i.e. a_{23}, a_{25} and a_{35} (see Fig. 7.9b). This also suggests that the judgment a_{23} is amongst the most inconsistent.

The priority vectors obtained using EV and GM are as follows:-

$$\textbf{EV:} \quad \begin{bmatrix} .3046 & .2811 & .2444 & .0649 & .0521 & .0530 \end{bmatrix}^T$$
$$\textbf{GM:} \quad \begin{bmatrix} .3074 & .2461 & .2524 & .0760 & .0595 & .0585 \end{bmatrix}^T$$

The *ideal* ranking possible for this PCM is $E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4 \rightarrow E_5 \rightarrow E_6$, however, the ranking order suggested by EV is $E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4 \rightarrow E_6 \rightarrow E_5$. Although the judgments were found to be *transitive*, the EV method has violated order of preference for one judgment i.e. the judgment $a_{56} = 1.19$ suggests $E_5 \rightarrow E_6$ but the estimated value w_5 is less than w_6. GM produces a different ranking order: $A \rightarrow C \rightarrow B \rightarrow D \rightarrow E \rightarrow F$. This method has also generated a priority violation but at a different place i.e. $w_B < w_C$ when $a_{23} > 1$.

7.4.1.2 A_{tech}

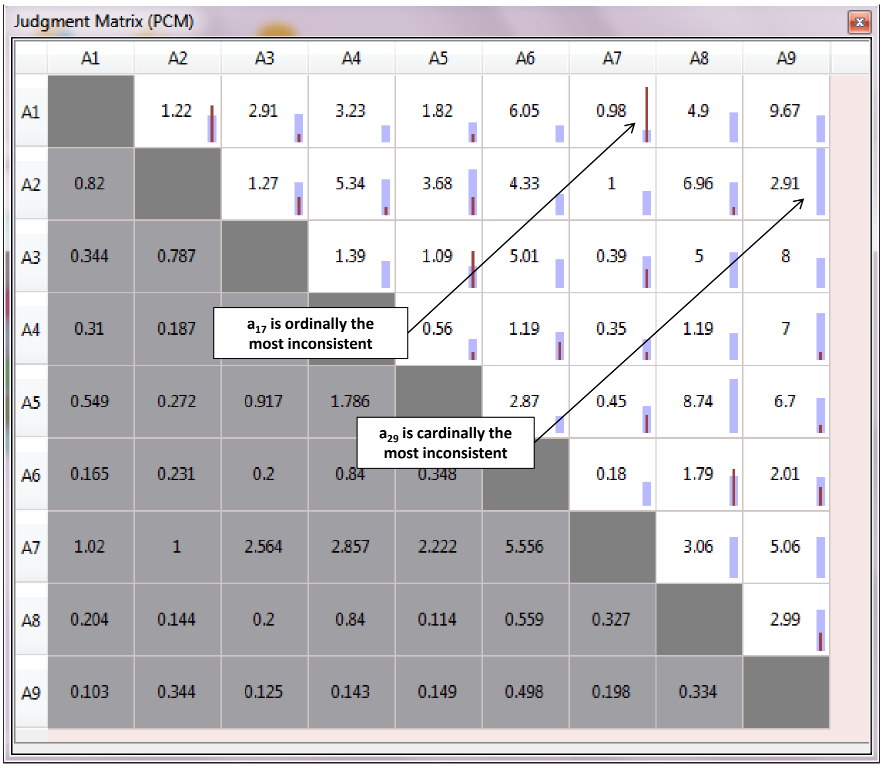
The table-view for A_{tech} is shown in Fig 7.10. The most inconsistent judgment according to the *congruence* measure is found to be a_{29}. However, the ordinal consistency measure, *dissonance*, suggests a_{17} as the most inconsistent. There exists a three-way cycle in this PCM i.e. $E_1 \rightarrow E_2 \sim E_7 \rightarrow E_1$. The judgment a_{29} does not contribute to this three-way cycle present in the PCM.

EV and GM solutions are given below:-

$$\textbf{EV:} \quad \begin{bmatrix} .2091 & .2021 & .1225 & .0648 & .1199 & .0391 & .1841 & .0357 & .0228 \end{bmatrix}^T$$
$$\textbf{GM:} \quad \begin{bmatrix} .2202 & .1931 & .1234 & .0635 & .1161 & .0406 & .1883 & .0342 & .0205 \end{bmatrix}^T$$

Both solutions give $NV = 1.5$. The judgment a_{17} suggests $E_7 \rightarrow E_1$ whilst w_1 is higher than w_7 using either EV or GM. The half-violation is added due to the presence of *preference equivalence*. The judgment a_{27} suggests $E_2 \sim E_7$ but w_2 is greater than w_7.

An intransitive PCM cannot produce a solution with $NV = 0$ therefore, the three-way cycle has to be removed. The *dissonance* measure suggests a_{17}



	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1		1.22	2.91	3.23	1.82	6.05	0.98	4.9	9.67
A2	0.82		1.27	5.34	3.68	4.33	1	6.96	2.91
A3	0.344	0.787		1.39	1.09	5.01	0.39	5	8
A4	0.31	0.187	[illegible]		0.56	1.19	0.35	1.19	7
A5	0.549	0.272	0.917	1.786		2.87	0.45	8.74	6.7
A6	0.165	0.231	0.2	0.84	0.348		0.18	1.79	2.01
A7	1.02	1	2.564	2.857	2.222	5.556		3.06	5.06
A8	0.204	0.144	0.2	0.84	0.114	0.559	0.327		2.99
A9	0.103	0.344	0.125	0.143	0.149	0.498	0.198	0.334	

Figure 7.10: Table-view for A_{tech}

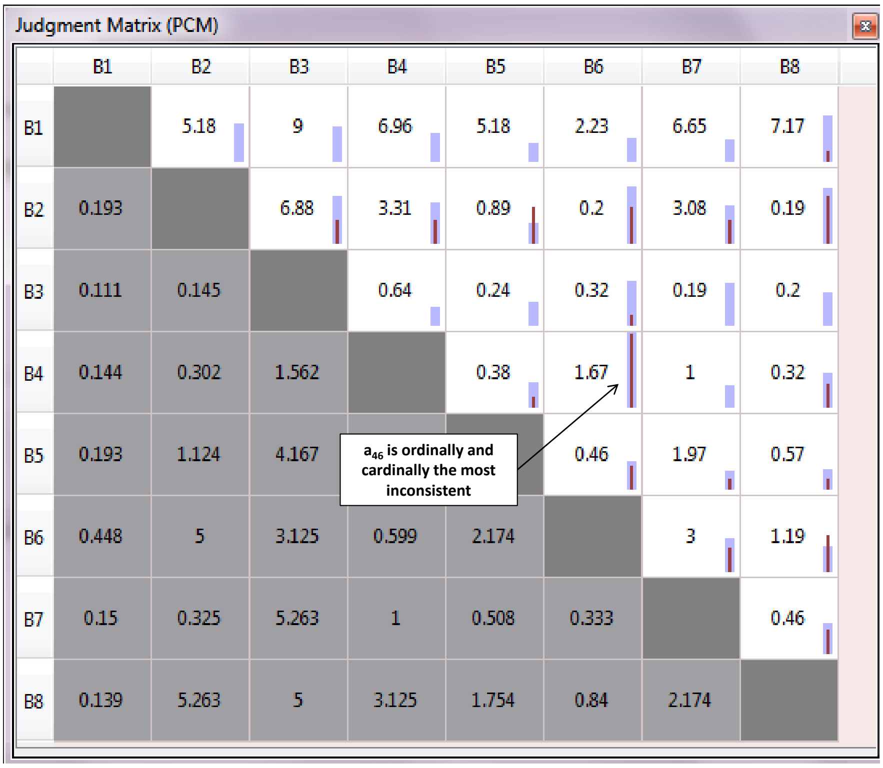
should be revised. Therefore, inverting the judgment of a_{17} will make the PCM transitive.

7.4.1.3 A_{infra}

The table-view of A_{infra} is given in Fig 7.10. The most inconsistent judgment according to the *congruence* measure is found to be a_{46}. The ordinal consistency measure, *dissonance*, also suggests a_{46} as the most inconsistent. There exists four three-way cycles in this PCM i.e.

$$\begin{aligned}
L_1 :& \quad E_2 \to E_4 \to E_6 \to E_2 \\
L_2 :& \quad E_6 \to E_5 \to E_4 \to E_6 \\
L_3 :& \quad E_4 \to E_6 \to E_8 \to E_4 \\
L_4 :& \quad E_4 \to E_6 \to E_7 \sim E_4
\end{aligned}$$

The judgment $a_{46} = 1.666$ has contributed the most to the three-way cycles present in the PCM. By inverting only the judgment a_{46}, all the three-way cycles can be rectified.



	B1	B2	B3	B4	B5	B6	B7	B8
B1		5.18	9	6.96	5.18	2.23	6.65	7.17
B2	0.193		6.88	3.31	0.89	0.2	3.08	0.19
B3	0.111	0.145		0.64	0.24	0.32	0.19	0.2
B4	0.144	0.302	1.562		0.38	1.67	1	0.32
B5	0.193	1.124	4.167			0.46	1.97	0.57
B6	0.448	5	3.125	0.599	2.174		3	1.19
B7	0.15	0.325	5.263	1	0.508	0.333		0.46
B8	0.139	5.263	5	3.125	1.754	0.84	2.174	

Figure 7.11: Table-view for A_{infra}

The EV and GM solutions for A_{infra} are:-

$$\textbf{EV:} \quad \begin{bmatrix} .3934 & .0888 & .0244 & .0606 & .0834 & .1500 & .0526 & .1468 \end{bmatrix}^T$$
$$\textbf{GM:} \quad \begin{bmatrix} .4102 & .0821 & .0246 & .0530 & .0928 & .1398 & .0555 & .1420 \end{bmatrix}^T$$

Both vectors generate two and a half violations i.e. $NV = 2.5$. The EV solution has violated a_{25} while the GM solution violated a_{68} instead. The judgments a_{46} and a_{47} have been violated by both the EV and GM solutions.

7.4.2 Improving Consistency

The investigations have highlighted the main sources of inconsistency i.e.

1. A_{top} is found to be unacceptable in AHP terms (CR=0.136). The major source of inconsistency is found to be $a_{23} = 4.09$, which is both ordinally and cardinally most inconsistent.

2. A_{tech} is found to be ordinally inconsistent (intransitive). Inverting the judgment of a_{17} can make the PCM transitive.

3. A_{infra} is also an intransitive PCM with $L = 4$. All the four three-way cycles can be removed by inverting a single judgment i.e. $a_{46} = 1.67$.

These judgments should be revised in order to improve the overall consistency of these matrices. As the judgments here cannot be revised manually, the suggested new values for the blamed judgments are:-

1. A_{top}: Change a_{23} from 4.09 to 0.99
2. A_{tech}: Change a_{17} from 0.98 to 1.01
3. A_{infra}: Change a_{46} from 1.67 to 0.99

The suggested values are calculated using (4.2). The final weights calculated after these improvements are given below in normalized form:-

EV: $w_{G1} = 20.70\%, w_{G2} = 21.20, w_{G3} = 29.35\%, w_{G4} = 28.74\%$
GM: $w_{G1} = 20.75\%, w_{G2} = 21.41, w_{G3} = 29.19\%, w_{G4} = 28.63\%$

Satellite communication (G4) is no longer the most preferred alternative, its weight has been reduced to 28.74% from 29.95%. The new results indicate that Microwave (G3) is the best alternative with a weight of 29.35% (using EV). The results for both EV and GM are almost in-differentiable.

7.4.3 Prioritization using PrInT

As mentioned earlier, there exist situations when revision of judgments is not allowed and prioritization is required without attempting to remove inconsistency. PriEsT has the ability to solve this problem using different prioritization methods. The solutions for the three matrices, A_{top}, A_{tech} and A_{infra} have been obtained in PriEsT using EV, GM and PrInT. The results are discussed below.

Table 7.2 lists the solutions for A_{top} generated by EV, GM and PrInT. When seen in the TD-TD2 plane, shown in Fig. 7.12, the EV and GM solutions are clearly dominated by the PrInT solutions.

PrInT has produced several solutions with $NV = 0$ and $NV = 1$. Fig. 7.12 shows all these solutions, however the solutions having $NV > 0$ are not listed in Table 7.2 being less relevant.

Similarly, the solutions for A_{tech} and A_{infra} are listed in Tables 7.3 and 7.4. In both cases, the EV and GM solutions generate more violations than the PrInT solutions. Moreover, the EV and GM solutions are again dominated by the PrInT solutions, as shown in Fig. 7.13a and Fig. 7.13b.

The solutions generated by PrInT are equally good and therefore any of them could be selected by the DM. Consider a situation where the solutions

Method	TD	TD2	NV	w_1	w_2	w_3	w_4	w_5	w_6
EV	1.4208	5.9541	1	0.3046	0.2811	0.2444	0.0649	0.0521	0.053
GM	1.329	6.1757	1	0.3074	0.2461	0.2524	0.076	0.0595	0.0585
PrInT-1	1.3065	5.9825	0	0.2342	0.1937	0.3887	0.0797	0.051	0.0528
PrInT-2	1.3073	5.9577	0	0.2138	0.2017	0.4027	0.0806	0.0521	0.0491
PrInT-3	1.3522	5.8889	0	0.2923	0.2746	0.2704	0.0676	0.0494	0.0457
PrInT-4	1.3562	5.8643	0	0.2859	0.2787	0.2744	0.0685	0.0479	0.0446
PrInT-5	1.383	5.8498	0	0.2832	0.282	0.2713	0.0737	0.046	0.0437
PrInT-6	1.4477	5.813	0	0.2933	0.2787	0.2744	0.0659	0.0456	0.0421
PrInT-7	1.5404	5.8032	0	0.306	0.2893	0.2457	0.0683	0.0473	0.0435

Table 7.2: Solutions for A_{top}

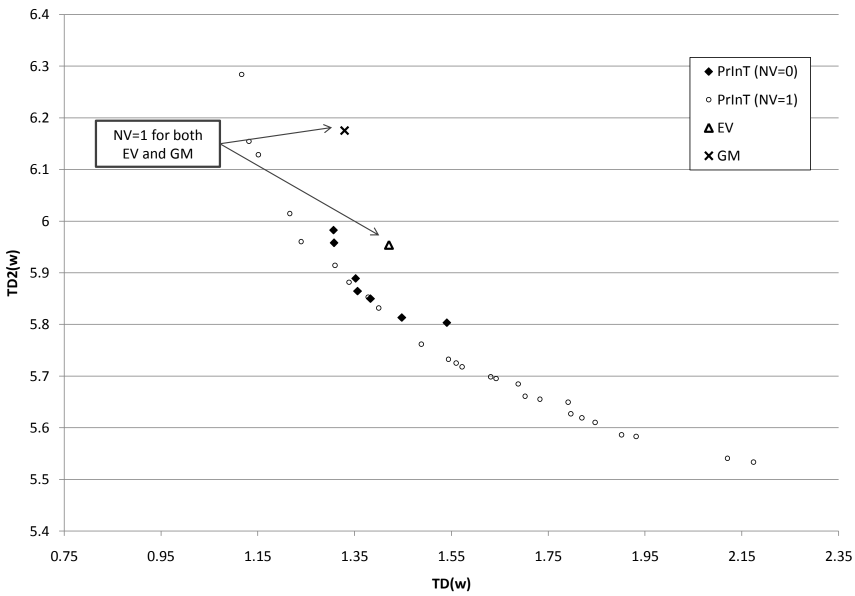


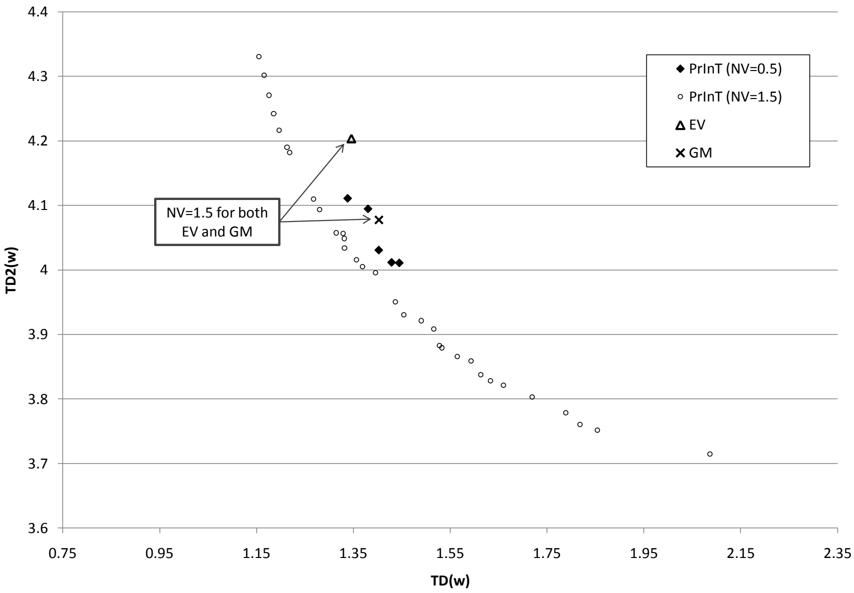
Figure 7.12: Solutions for A_{top} in TD-TD2 plane

Method	TD	TD2	NV	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
EV	1.3461	4.2035	1.5	0.209	0.2021	0.1225	0.0648	0.1199	0.0391	0.1841	0.0357	0.0228
GM	1.4026	4.0775	1.5	0.2202	0.1931	0.1234	0.0635	0.1161	0.0406	0.1883	0.0342	0.0205
PrInT-1	1.3383	4.1109	0.5	0.1973	0.1652	0.147	0.0613	0.1453	0.0366	0.1994	0.0274	0.0204
PrInT-2	1.3803	4.0947	0.5	0.1956	0.1691	0.1461	0.0636	0.1316	0.0357	0.209	0.0291	0.0202
PrInT-3	1.4025	4.0309	0.5	0.1962	0.1841	0.1333	0.0673	0.1331	0.0353	0.2016	0.0293	0.0198
PrInT-4	1.4289	4.0119	0.5	0.1945	0.1791	0.1392	0.0621	0.1353	0.0401	0.2014	0.0294	0.0191
PrInT-5	1.4448	4.0111	0.5	0.1973	0.1775	0.138	0.0615	0.1341	0.0439	0.1996	0.0291	0.0189

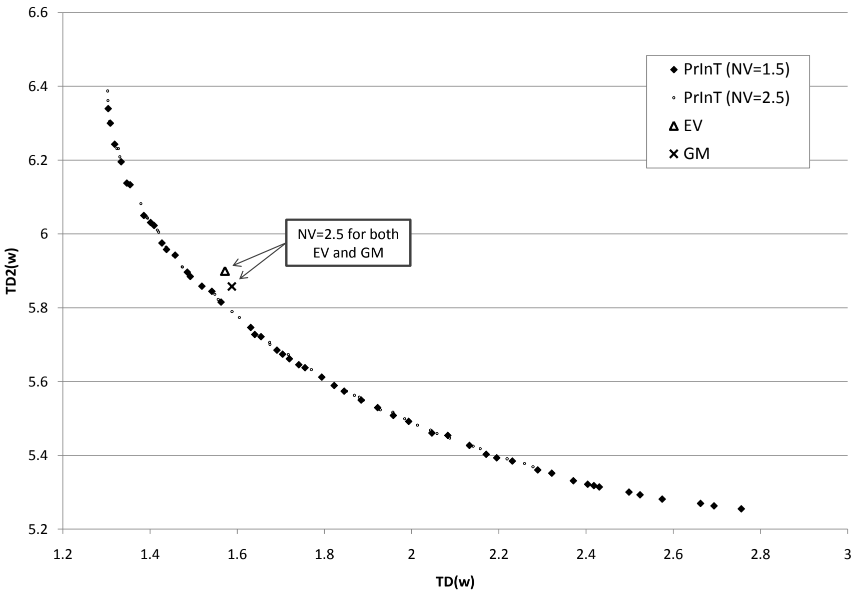
Table 7.3: Solutions for A_{tech}

Method	TD	TD2	NV	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
EV	1.5716	5.8996	2.5	0.3934	0.0888	0.0244	0.0606	0.0834	0.15	0.0526	0.1468
GM	1.5875	5.8581	2.5	0.4102	0.0821	0.0246	0.053	0.0928	0.1398	0.0555	0.142
PrInT-1	1.3041	6.3398	1.5	0.3768	0.0919	0.0339	0.051	0.0923	0.1639	0.0544	0.1358
PrInT-2	1.3089	6.2998	1.5	0.3884	0.0761	0.0339	0.0511	0.0876	0.1637	0.0613	0.1379
PrInT-3	1.3188	6.2426	1.5	0.3851	0.0822	0.0317	0.0509	0.0918	0.163	0.0612	0.134
PrInT-4	1.334	6.1952	1.5	0.3657	0.0855	0.0309	0.0474	0.0996	0.1712	0.0513	0.1484
PrInT-5	1.3468	6.1379	1.5	0.3712	0.0712	0.0306	0.0465	0.093	0.1663	0.0553	0.1659
PrInT-6	1.354	6.1332	1.5	0.3886	0.0766	0.0313	0.0482	0.0855	0.1609	0.052	0.157
PrInT-7	1.386	6.05	1.5	0.3739	0.0809	0.0285	0.0462	0.0893	0.1641	0.0542	0.163
PrInT-8	1.4015	6.0309	1.5	0.3627	0.0738	0.0279	0.0434	0.0946	0.1707	0.0568	0.1701
PrInT-9	1.4092	6.0227	1.5	0.3622	0.0732	0.0277	0.0421	0.1002	0.1706	0.0567	0.1673
PrInT-10	1.427	5.9753	1.5	0.3739	0.0717	0.0274	0.0449	0.0926	0.1672	0.0556	0.1668
PrInT-11	1.4376	5.958	1.5	0.3768	0.0714	0.0272	0.0447	0.0922	0.1665	0.0552	0.1659
PrInT-12	1.4574	5.942	1.5	0.394	0.076	0.027	0.0479	0.0847	0.1589	0.0557	0.1558
PrInT-13	1.4853	5.8961	1.5	0.3852	0.061	0.027	0.0484	0.0856	0.1716	0.0523	0.1689
PrInT-14	1.4925	5.8848	1.5	0.3814	0.0607	0.0268	0.044	0.0993	0.1713	0.052	0.1646
PrInT-15	1.5187	5.8578	1.5	0.3933	0.0602	0.0272	0.0429	0.0848	0.1728	0.0524	0.1665
PrInT-16	1.5421	5.8442	1.5	0.3651	0.0587	0.0262	0.0427	0.0824	0.1917	0.0507	0.1826
PrInT-17	1.563	5.8152	1.5	0.3884	0.0605	0.0267	0.0408	0.0881	0.1822	0.0478	0.1654
PrInT-18	1.6307	5.7461	1.5	0.3785	0.0595	0.024	0.0427	0.0874	0.183	0.0574	0.1676
PrInT-19	1.6404	5.7275	1.5	0.3803	0.0594	0.0242	0.0429	0.0881	0.1839	0.0526	0.1685
PrInT-20	1.6546	5.7216	1.5	0.3793	0.053	0.0244	0.0429	0.0881	0.1837	0.0523	0.1763
PrInT-21	1.6912	5.6845	1.5	0.3946	0.0615	0.0239	0.0431	0.0854	0.1776	0.0507	0.1632
PrInT-22	1.7043	5.6739	1.5	0.3894	0.0567	0.0239	0.042	0.0862	0.1782	0.0512	0.1725
PrInT-23	1.7194	5.6612	1.5	0.3842	0.064	0.0233	0.0427	0.0877	0.1824	0.0477	0.168
PrInT-24	1.7415	5.645	1.5	0.3825	0.0631	0.0229	0.0426	0.0864	0.1801	0.0497	0.1729
PrInT-25	1.7554	5.6372	1.5	0.3788	0.0632	0.0229	0.0421	0.0865	0.1804	0.0465	0.1794
PrInT-26	1.7938	5.6113	1.5	0.396	0.0575	0.0227	0.0444	0.0856	0.1793	0.0509	0.1636
PrInT-27	1.8223	5.5886	1.5	0.3919	0.0625	0.0225	0.0416	0.0855	0.1783	0.0462	0.1714
PrInT-28	1.8455	5.5736	1.5	0.3897	0.0547	0.0223	0.0437	0.0842	0.1826	0.0485	0.1745
PrInT-29	1.8842	5.5492	1.5	0.3868	0.0529	0.022	0.0428	0.0884	0.1843	0.0472	0.1757
PrInT-30	1.9217	5.5288	1.5	0.383	0.057	0.0212	0.0427	0.0868	0.1834	0.0484	0.1775
PrInT-31	1.9579	5.5079	1.5	0.3994	0.0579	0.0215	0.0416	0.0879	0.1783	0.0458	0.1675
PrInT-32	1.9931	5.4913	1.5	0.4028	0.0518	0.0215	0.042	0.0862	0.1807	0.0462	0.1689
PrInT-33	2.0467	5.4605	1.5	0.3867	0.0519	0.0209	0.0405	0.0833	0.2007	0.0449	0.1712
PrInT-34	2.0827	5.4536	1.5	0.376	0.0504	0.0202	0.0394	0.081	0.1951	0.049	0.1888
PrInT-35	2.1712	5.4026	1.5	0.3862	0.0528	0.0198	0.0392	0.0882	0.1935	0.043	0.1773
PrInT-36	2.2311	5.3844	1.5	0.3961	0.0525	0.0195	0.0423	0.0778	0.1922	0.0442	0.1754
PrInT-37	2.2893	5.3603	1.5	0.3822	0.0555	0.0189	0.0377	0.0847	0.2029	0.0434	0.1747
PrInT-38	2.3217	5.3511	1.5	0.3798	0.0545	0.0187	0.0374	0.0842	0.2086	0.0431	0.1736
PrInT-39	2.3708	5.331	1.5	0.3897	0.0506	0.0188	0.037	0.0842	0.2015	0.0422	0.176
PrInT-40	2.4036	5.3214	1.5	0.3919	0.0502	0.0187	0.0368	0.0839	0.2002	0.0419	0.1763
PrInT-41	2.418	5.3173	1.5	0.3922	0.0504	0.0186	0.0373	0.0839	0.2007	0.0419	0.175
PrInT-42	2.4985	5.2996	1.5	0.3869	0.0507	0.0179	0.0371	0.0847	0.1951	0.0422	0.1855
PrInT-43	2.524	5.2925	1.5	0.3958	0.0507	0.0179	0.0371	0.0847	0.1958	0.0423	0.1755
PrInT-44	2.5748	5.2811	1.5	0.3894	0.0499	0.0176	0.0365	0.0834	0.1989	0.0416	0.1827
PrInT-45	2.6626	5.2694	1.5	0.3842	0.0505	0.0171	0.0344	0.0845	0.2018	0.0421	0.1855
PrInT-46	2.6936	5.2629	1.5	0.3976	0.0503	0.0173	0.0349	0.0851	0.2033	0.0408	0.1706
PrInT-47	2.756	5.2547	1.5	0.392	0.0506	0.0169	0.034	0.0835	0.1996	0.0398	0.1835
PrInT-48	2.8598	5.246	1.5	0.3797	0.0481	0.0165	0.0335	0.0813	0.2235	0.039	0.1785

Table 7.4: Solutions for A_{infra}



(a) A_{tech}



(b) A_{infra}

Figure 7.13: Solutions for A_{tech} and A_{infra} in TD-TD2 plane

selected are PrInT-7 for A_{top}, PrInT-2 for A_{tech} and PrInT-32 for A_{infra} (see tables 7.2, 7.3 and 7.4). The overall generated weights will be:-

PrInT: $w_{G1} = 21.31\%$, $w_{G2} = 19.70\%$, $w_{G3} = 29.35\%$, $w_{G4} = 30.64\%$.

Choosing a different solution from the set of non-dominated ones will obviously end up with different weights. Although different, no solution can be declared to be inferior.

It can be argued that PrInT should produce a single solution from PrInT to support situations where user interaction is not possible. We consider this to be a future area of research: 'the selection of the most appropriate solution from within a set of Pareto-optimal solutions' in the context of pairwise comparisons.

7.5 Summary

This chapter has discussed the development and evaluation of a priority estimation tool (PriEsT) that offers several new features. In the case of inconsistent judgments, PriEsT offers a wide range of Pareto-optimal solutions based on multi-objective optimization. The DM has the flexibility to select any of these non-dominated solutions according to his/her requirements. PriEsT also assists the DM in revising his/her judgments and highlights any dissonance and/or intransitivity present in given judgments.

The features of PriEsT have been demonstrated and evaluated with practical data acquired for selecting the most appropriate Telecom infrastructure for rural areas. PriEsT has highlighted the presence of intransitive judgments in the acquired data. It has been found that correcting these judgments has lead to a different ranking of the available alternatives. When revision of judgments is not allowed, the EV and GM solutions are dominated by the PrInT solutions.

Chapter 8

Conclusions & Future Work

This chapter summarises the contributions of the research and suggests areas for further investigation.

8.1 Summary

Inconsistency in pairwise comparison (PC) judgments is of two types i.e. *cardinal* and *ordinal*. Existing consistency measures have been investigated and found to be insufficient for measuring inconsistency. The research has therefore focused firstly on obtaining measures to capture both cardinal consistency (CC) and ordinal consistency (OC).

Two new measures have been introduced: *congruence* to address CC and *dissonance* to address OC. It has been shown through examples that, unlike other CC measures, *congruence* is suitable for measuring both the CC of the individual judgments in a PC matrix (PCM) and the overall CC of the PCM. The introduction of the *dissonance* measure has been found to be supplemental to *congruence* and is useful in identifying priority violations. Because of their utility, the two matrices representing individual *congruence* and *dissonance* have been recommended as a useful addition to PC-based decision support tools.

The next obvious step is to attempt to reduce the identified inconsistency prior to the process of prioritization. A heuristic method has been proposed to detect and remove intransitive judgments and thus address OC. The new method both improves OC and is computationally more efficient than optimization based methods.

The research then proceeded to consider the problem of prioritization when the inconsistency is considered to be irreducible. A new method, termed EAST, has been proposed to estimate preferences using a graph-theoretic approach that reveals all possible preferences from a given PCM. The *mean* of these preferences is used as the final priority vector. EAST can estimate preferences from incomplete PCMs without modification, unlike other methods which require intermediate steps to estimate missing judgments. Although the performance of the proposed prioritization method was found to be comparable to other approaches, nonetheless, it has practical limitation in terms of computation time: EAST becomes intractable when the number of elements to compare is greater than 9.

As a direct consequence of this computational limitation, prioritization has then been explored as an optimization problem. A new method based on multi-objective optimization has been formulated for direct and indirect judgments. The method suggests minimizing the ordinal violations along with the deviations from direct and indirect judgments in terms of Euclidean distance. This method offers multiple non-dominated solutions and outperforms all other relevant methods in terms of the three defined objectives.

A priority estimation tool (PriEsT) has been developed that offers several features. This includes both new ways to measure inconsistency and an implementation of the proposed prioritization methods. In contrast to other relevant software tools, PriEsT offers multiple equally good solutions along with an interactive selection process for the user.

In order to show the benefits of PriEsT, a case study involving Telecom infrastructure selection is presented. The different solutions generated by PrInT obviously end up giving different weights, however, no one solution can be considered to be inferior.

8.2 Future Work

8.2.1 Threshold for Congruence

The *congruence* measure requires a threshold value as with, for example, $CR \leqslant 0.1$ that declares a PCM to be *acceptable*. Such a threshold would also enable individual judgments to be declared to be *acceptable* or *unacceptable*.

For purpose of demonstration, assume that the threshold value of 0.6 for *congruence* has been selected for the following two PCMs generated with the linear scale of 1 to 9:-

$$A_1 = \begin{bmatrix} 1 & 2 & 4 & 8 & \mathbf{\frac{1}{3}} \\ \frac{1}{2} & 1 & 2 & 4 & 8 \\ \frac{1}{4} & \frac{1}{2} & 1 & 2 & 4 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 & 2 \\ 3 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & 2 & 4 & 8 & \mathbf{9} \\ \frac{1}{2} & 1 & 2 & 4 & 8 \\ \frac{1}{4} & \frac{1}{2} & 1 & 2 & 4 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 & 2 \\ 3 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

The value of CR for A_1 is 0.55 suggesting that the PCM is *unacceptable* and the value of CR for A_2 is 0.009 that qualifies A_2 as an *acceptable* PCM. However, the use of CR does not reveal the sources of inconsistency in these PCMs.

The *congruence* matrices for A_1 and A_2 are shown below:-

$$A_1 : \theta = \begin{bmatrix} - & 1.29 & 1.29 & 1.29 & \underline{\mathbf{3.87}} \\ & - & 0 & 0 & 1.29 \\ & & - & 0 & 1.29 \\ & & & - & 1.29 \\ & & & & - \end{bmatrix}$$

$$A_2 : \theta = \begin{bmatrix} - & 0.19 & 0.19 & 0.19 & \underline{\mathbf{0.58}} \\ & - & 0 & 0 & 0.19 \\ & & - & 0 & 0.19 \\ & & & - & 0.19 \\ & & & & - \end{bmatrix}$$

The *congruence* measure detects a_{15} to be an outlier in both PCMs. The value of $\theta_{15} = 3.87$ for A_1 suggests that judgment a_{15} is highly inconsistent. The value of $\theta_{15} = 0.58$ for A_2 suggests that a_{15} is less inconsistent in A_2 than in A_1.

The overall *congruence*, Θ, for A_1 is equal to 1.16 using (3.4) (i.e. the average of the individual *congruence* values). The threshold of 0.6 suggests that the PCM is *unacceptable* as $\Theta > 0.6$. Based on this threshold, only three judgments in A_1 are found to be *acceptable* i.e. a_{23}, a_{34} and a_{24}.

A_2 has an overall *congruence* value of $\Theta = 0.17$. The threshold of 0.6 declares A_2 to be an *acceptable* PCM. All the judgments in A_2 are also considered to be *acceptable* when using this threshold value.

As demonstrated above, a properly justified threshold value for *congruence* will offer increased utility in comparison to the threshold available for CR. We consider this to be an important area for future work; finding an appropriate threshold value for the *congruence* measure to determine *acceptable* and *unacceptable* judgments in a PCM.

8.2.2 Investigating the Non-monotonic Increase in Dissonance

As discussed in Chapter 3, Monte-Carlo simulations were used in order to find a possible relation between overall *dissonance* (Ψ) and the number of violations (NV). The results have shown a slight variation in monotonic increase of NV at $\Psi \approx 0.20$ (seen in Fig. 3.10). This phenomenon has not been explored in depth here and requires future investigation.

It is contended that this variation is related to the linear scale of 1-to-9 used for measurements. However, empirical evidence is required to support this contention. A possible approach to this would be to use Monte-Carlo simulations with different measurement scales. If the variation in monotonic increase of NV is different for each measurement scale, the obvious next step would be to compare these values for each measurement scale.

Assume that the lowest values for NV are expected when using the *balanced* scale. This will justify the use of the *balanced* scale for judgments when a minimum number of violations is considered to be of high importance.

We consider this to be an important area to investigate that may provide better justification for the selection of a measurement scale.

8.2.3 Detecting Sources of Inconsistency

A single global measure (e.g. CR, CM or GCI) appears to be insufficient to locate all types of inconsistencies. This has led here to the development and suggested use of the *congruence* and *dissonance* matrices to represent individual inconsistencies. As discussed in chapter 2, there exist different sources of inconsistency. Sugden [78] categorized their cause into psychological reasons, clerical errors and insufficient model structures. It is suggested that the *congruence* and *dissonance* matrices may also help detect the source of inconsistency.

Consider the above mentioned PCMs (A_1 and A_2): the *congruence* matrices for A_1 and A_2 suggest a_{15} to be an outlier in both PCMs. The *dissonance* matrices for A_1 and A_2 are shown below:-

$$A_1 : \psi \;=\; \begin{bmatrix} - & 0.33 & 0.33 & 0.33 & \underline{\mathbf{1.00}} \\ & - & 0 & 0 & 0.33 \\ & & - & 0 & 0.33 \\ & & & - & 0.33 \\ & & & & - \end{bmatrix}$$

$$A_2 : \psi \;=\; \begin{bmatrix} - & 0 & 0 & 0 & \underline{\mathbf{0}} \\ & - & 0 & 0 & 0 \\ & & - & 0 & 0 \\ & & & - & 0 \\ & & & & - \end{bmatrix}$$

The *dissonance* matrix for A_1 reveals that judgment a_{15} is ordinally inconsistent and needs to be reversed. This provides a hint that there is a possible clerical error in A_1.

In contrast to A_1, the *dissonance* matrix for A_2 suggests that a_{15} is ordinally consistent. This suggests that it is unlikely there is a clerical error in A_2. The inconsistency may be present due to some other reason. In this case, the most likely cause appears to be the use of 1-9 scale.

In order to improve inconsistency in PC judgments, it is important to know the cause of inconsistency whatever it may be. Consider the Analytic Hierarchy Process (AHP) where several PCMs are generated and processed in order to estimate final preferences. The decision maker may rectify each PCM with a different approach provided that he/she knows the cause of inconsistency in these PCMs.

The above example shows the potential use of the two suggested measures for this purpose. This should be investigated further for all known sources of inconsistency in PC judgments. In turn, a software prototype may be appropriate in order to present this feature to potential users who can determine its utility.

8.2.4 Consistency Analysis using EAST

As mentioned in chapter 5, the generation of all spanning trees suggests all the preference vectors possible from a given PCM. This approach may be useful in analyzing inconsistency among judgments. Based on statistical theory, the *variance* among the vectors $\tilde{w}(\tau_s)$ (generated for each tree τ_s) can be used as a

τ_s	$\tilde{w}_1$	$\tilde{w}_2$	$\tilde{w}_3$	$\tilde{w}_4$	σ_s
τ_1	0.2637	0.5275_	0.1758	0.0330	0.010
τ_2	0.1304	0.2609	0.4348	0.1739	0.037
τ_3	0.2941	0.3529_	0.1176	0.2353	0.006
τ_4	**0.3623**	**0.4348**	**0.1449**	**0.0580**	**0.002**
τ_5	**0.3670**	**0.4404**	**0.1468**	**0.0459**	**0.002**
τ_6	**0.4211**	**0.3947**	**0.1316**	**0.0526**	**0.002**
τ_7	0.7273	0.1364	0.0455	0.0909	0.050
τ_8	0.6154	0.1154	0.1923	0.0769	0.033
τ_9	0.5839	0.1095	0.2336	0.0730	0.031
τ_{10}	0.2000	0.4000	0.1333	0.2667	0.014
τ_{11}	0.2542	0.5085	0.1695	0.0678	0.008
τ_{12}	0.5556	0.1333	0.2222	0.0889	0.024
τ_{13}	0.2837	0.5674	0.1135	0.0355	0.013
τ_{14}	0.2909	0.5818	0.0909	0.0364	0.015
τ_{15}	0.2113	0.4225	0.0845	0.2817	0.016
τ_{16}	0.2809	0.5618	0.1124	0.0449	0.012

Table 8.1: List of deviations calculated for each tree in Γ_1

measure of inconsistency, as defined below:-

$$\sigma = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\eta-1}\sum_{s=1}^{\eta}\left(w_i - \tilde{w}_i(\tau_s)\right)^2\right) \tag{8.1}$$

The value of σ is always equal to zero for a consistent set of judgments. Here, we demonstrate how variance can help in trimming the trees responsible for a large degree of deviation. Consider the following example from chapter 5:-

$$A_1 = \begin{bmatrix} 1 & \frac{1}{2} & 2\frac{1}{2} & 8 \\ 2 & 1 & 3 & 1\frac{1}{2} \\ \frac{2}{5} & \frac{1}{3} & 1 & 2\frac{1}{2} \\ \frac{1}{8} & \frac{2}{3} & \frac{2}{5} & 1 \end{bmatrix}$$

The list of all preference vectors for A_1 was given in 5.1. The value of variance for A_1 is calculated as $\sigma = 0.018$ using (8.1). The generated preference vectors can be clustered according to their deviation from the average preference vector.

Table 8.1 highlights three of the trees (τ_4, τ_5 and τ_6) that generate vectors that are closest to the average preference vector (i.e. $\sigma_s \approx 0$). Similarly, five of the trees have generated vectors that deviate more than the variance (i.e. $\sigma_s > \sigma$). The deviation values for these five trees are underlined in Table 8.1.

Therefore, the EAST-generated vectors can be grouped into three clusters based on variance, σ_s i.e. closest to average vector, farthest from average vector and the rest. The vectors are clustered in three groups for demonstration purposes only. The number of clusters to be created is itself an area for further investigation.

As mentioned earlier, the threshold of $CR \leqslant 0.1$ lacks a formal justification. In contrast, it may be possible to use EAST to justify the declaration of *acceptable* or *unacceptable* PCM based on statistical theory. For example, a PCM is *acceptable* when 75% of the generated vectors have variance, $\sigma_s < \sigma$. Alternatively, consistency can be analyzed with the cluster giving $\sigma_s \approx 0$. For example, a PCM can be declared to be *acceptable* when 50% of the EAST-generated vectors lie closest to the average preference vector (i.e. $\sigma_s \approx 0$). The ideas of acceptability and the threshold values need both further investigation and better justification based on empirical evidence.

8.2.5 Geometric Mean for Incomplete PCMs

It has been highlighted that the probability of acquiring incomplete PCMs increases as n increases [27]. In such cases, the EV and GM methods cannot estimate preferences without applying an intermediate method to complete the missing judgments.

It has been found that the geometric mean of EAST-generated vectors produce the same results as those produced by the geometric mean (GM) method. This is an interesting finding as the GM method is not applicable to incomplete PCMs while EAST can estimate preferences from incomplete PCMs without modification.

As discussed earlier, the GM method can be replaced with the LLS method for incomplete PCMs. Hence, further investigation is required to compare EAST with the LLS approach. It is evident in Fig. 5.5 that EAST outperforms LLS for complete PCMs with $n < 7$ in terms of computation time. However, EAST has been found to be computationally expensive for larger matrices ($n \geqslant 7$), therefore, experimentation is required to investigate the practical limitations of EAST for incomplete PCMs.

8.2.6 Generating a Single Solution from PrInT

The PrInT method generates several non-dominated solutions. This may lead to information overload causing a DM to neither wish nor be able to select from many solutions. Automated selection of a preferred solution from within a set of non-dominated solutions is an area for further work.

Consider the minimum number of violations: the TOP method produces a single solution. We emphasize here that although TOP gives a single solution with minimum number of violations, the weights produced to minimize violations are not well dispersed. This can be explained with the following example:-

$$A = \begin{bmatrix} 1 & 1.9 & 1.1 \\ 0.53 & 1 & 1.2 \\ 0.91 & 0.83 & 1 \end{bmatrix}$$

The PCM, A has been generated using the scale of 1.1 to 1.9 that is used to compare stimuli with close resemblance [49]. The preference vectors generated by EV, GM, TOP and PrInT are listed in Table 8.2. The DLS/TOP solution in the table shows that the minimum value of TD possible for this PCM is 0.211. Although the PCM is transitive, the solution for DLS/TOP generates a violation i.e. $w_2 < w_3$ whereas $a_{23} > 1$. The EV and GM solutions have also generated the same violation. Other solutions generated by TOP and PrInT confirm that an *ideal ranking* is possible giving zero violation.

Consider the second solution generated by the TOP method (which is also generated by PrInT). The minimum value of TD suggests $w_2 < w_3$, however, this violates the second objective of NV. TOP suggests minimizing TD whilst keeping $NV = 0$. Therefore, the value of w_2 needs to be increased, leading to a solution with $w_2 \gtrapprox w_3$. Further increasing the value of w_2 will obviously end up in higher values for TD. Hence, the solution with $w_2 \gtrapprox w_3$ gives an equilibrium state for minimizing TD with $NV = 0$.

PrInT, in comparison to TOP, has the further objective that TD2 be minimized simultaneously. Table 8.2 shows that PrInT has generated several solutions where w_2 and w_3 do not have similar values. PrInT-14 has been highlighted to show the dispersion of weights for this solution. The ratio between w_1 and w_2 is 1.86 which is close to the given judgment $a_{12} = 1.9$. Similarly, the value of $\frac{w_2}{w_3} = 1.21$ comfortably approximates the value of $a_{23} = 1.2$. The judgment $a_{13} = 1.1$ has not been satisfied ($\frac{w_1}{w_3} = 2.25$), however, the order of preference

Method	w_1	w_2	w_3	NV	TD	TD2	w_1/w_2	w_1/w_3	w_2/w_3
EV	0.4195	0.2815	0.2991	1	0.233	0.507	1.49	1.40	0.94
GM	0.4195	0.2815	0.2991	1	0.233	0.507	1.49	1.40	0.94
DLS/TOP	0.4292	0.2511	0.3197	1	0.211	0.541	1.71	1.34	0.79
PrInT/TOP	**0.4285**	**0.2858**	**0.2857**	**0**	**0.245**	**0.450**	**1.50**	**1.50**	**1.00**
PrInT-0	0.4322	0.2841	0.2838	0	0.246	0.437	1.52	1.52	1.00
PrInT-1	0.4355	0.2825	0.282	0	0.247	0.425	1.54	1.54	1.00
PrInT-2	0.4392	0.2805	0.2803	0	0.248	0.412	1.57	1.57	1.00
PrInT-3	0.4429	0.2788	0.2783	0	0.251	0.397	1.59	1.59	1.00
PrInT-4	0.4497	0.2759	0.2744	0	0.258	0.370	1.63	1.64	1.01
PrInT-5	0.4684	0.2685	0.2631	0	0.294	0.289	1.74	1.78	1.02
PrInT-6	0.4677	0.2698	0.2625	0	0.295	0.288	1.73	1.78	1.03
PrInT-7	0.4677	0.2712	0.261	0	0.299	0.282	1.72	1.79	1.04
PrInT-8	0.4737	0.2742	0.2521	0	0.329	0.231	1.73	1.88	1.09
PrInT-9	0.4721	0.2778	0.25	0	0.334	0.226	1.70	1.89	1.11
PrInT-10	0.4864	0.2712	0.2423	0	0.374	0.158	1.79	2.01	1.12
PrInT-11	0.4864	0.2746	0.239	0	0.386	0.141	1.77	2.04	1.15
PrInT-12	0.4953	0.2707	0.234	0	0.417	0.094	1.83	2.12	1.16
PrInT-13	0.4915	0.2789	0.2297	0	0.428	0.081	1.76	2.14	1.21
PrInT-14	**0.5048**	**0.2708**	**0.2245**	**0**	**0.469**	**0.018**	**1.86**	**2.25**	**1.21**
PrInT-15	0.5051	0.2713	0.2236	0	0.474	0.012	1.86	2.26	1.21
PrInT-16	0.4996	0.275	0.2254	0	0.457	0.036	1.82	2.22	1.22
PrInT-17	0.5028	0.2733	0.2239	0	0.469	0.019	1.84	2.25	1.22
PrInT-18	0.4951	0.2876	0.2173	0	0.489	0.001	1.72	2.28	1.32

Table 8.2: Comparing TOP and PrInT solutions

has been maintained. We suggest this solution as the most appropriate generated by PrInT based on the dispersion of generated weights.

As an example, Elsner and van den Driessche [123] demonstrated the use of PCM in music theory where frequencies can be attached to the keys of an instrument (e.g. piano) with the help of ratio comparisons. We consider this an interesting example to justify dispersion of generated weights, as assignment of the same frequency to two different keys is infeasible. A feasible approach would be to assign the frequencies to be as different as possible.

8.2.7 Enhancements to PriEsT

The requirements for the prototype of PriEsT have been collected by analysing existing tools that support the use of PCs. We aim to further develop PriEsT according to the guidelines provided by ISO/IEC 9126. The quality of PriEsT should be improved by interviewing potential users and updating the requirements accordingly. The following areas have already been suggested for improvement:

- Currently, judgments are entered in PriEsT as numeric values leading to the construction of a PCM. Verbal scales have been widely used for intangible criteria, therefore, we aim to support a verbal scale in future.
- Sensitivity analysis should be incorporated into PriEsT to see how results vary when judgments are slightly changed.
- A provision of web-based data-entry should make the software more useful.
- PriEsT saves all acquired data and results in a relational database system. Spreadsheet applications are often considered to be essential in the domain of decision making; hence, it is planned to implement an import/export facility for spreadsheet applications such as Microsoft Excel.
- An important feature to include is support for group decision making. This feature combined with web-based data-entry should significantly increase the use of PriEsT.
- Considering the PrInT method, a better way to visualize the three objectives is by using 3D plot. The TD-TD2 plot will be transformed into a 3D graph by having different values of NV on z-axis.

8.3 Conclusion

The research has addressed the two most significant problems related to pairwise comparisons. The first problem of consistency measurement has been addressed by proposing the two measures for CC and OC i.e. *congruence* and *dissonance*, respectively. The second problem is to derive preferences from inconsistent set of judgments; for this, the use of PrInT method gives non-inferior solutions with minimum ordinal violations. A priority estimation tool (PriEsT) has been developed that offers several new features. The benefits of using PriEsT have been demonstrated with the help of a case study. The research has concluded with the identification and overview of several areas of further investigation.

Bibliography

[1] T. L. Saaty, Relative Measurement and Its Generalization in Decision Making. Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process, Real Academia de Ciencias, España 102 (2) (2008) 251–318.

[2] L. Mikhailov, Multiobjective Prioritisation in the Analytic Hierarchy Process using Evolutionary Computing, in: A. Tiwari, J. Knowles, E. Avineri, K. Dahal, R. Roy (Eds.), Applications of Soft Computing: Recent Trends, Springer, 123–132, 2006.

[3] P. Gabor, Management Theory and Rational Decision Making, Management Decision 14 (5) (1993) 274–281.

[4] H. Zimmermann, MultiCriteria Decision Making in Ill-structured Situations, in: Fuzzy Sets, Decision Making and Expert Systems, Kluwer Academic Publishers, 125–151, 1987.

[5] G. A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, Psychological Review 63 (2) (1956) 81–97.

[6] L. Duckstein, S. Opricovic, Multiobjective optimization in river basin development, Water Resour. Res. 16 (1) (1980) 14–20.

[7] S. Opricovic, G.-H. Tzeng, Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS, European Journal of Operational Research 156 (2) (2004) 445–455.

[8] S. Pugh, Total Design: Integrated Methods for Successful Product Engineering, Addison-Wesley, 1991.

[9] E. Triantaphyllou, Multi-Criteria Decision Making Methods: A Comparative Study, Kluwer Academic Publishers, Dordrecht, 2000.

[10] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, K. Deb, Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead, Management Science 54 (7) (2008) 1336–1349.

[11] B. Roy, Classement et choix en presence de points de vue multiples (la methode Electre), la Revue d'Informatique et de Recherche Operationelle (RIRO) 2 (8) (1968) 57–75.

[12] C.-L. Hwang, K. P. Yoon, Multiple attribute decision making: Methods and applications: A state-of-the-art survey, Springer-Verlag, Berlin, 1981.

[13] K. P. Yoon, C.-L. Hwang, Multiple Attribute Decision Making, An Introduction, SAGE Publications, 1995.

[14] J. Figueira, S. Greco, M. Ehrogott, V. Mousseau, B. Roy, ELECTRE Methods, in:

F. Hillier (Ed.), Multiple Criteria Decision Analysis: State of the Art Surveys, vol. 78, Springer New York, 133–153, 2005.

[15] J. Brans, B. Mareschal, PROMETHEE Methods, in: J. Figueira, S. Greco, M. Ehrgott (Eds.), Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, 163–196, 2005.

[16] M. Behzadian, R. Kazemzadeh, A. Albadvi, M. Aghdasi, PROMETHEE: A comprehensive literature review on methodologies and applications, European Journal of Operational Research 200 (1) (2010) 198–215.

[17] T. L. Saaty, The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation, McGraw-Hill, New York, 1980.

[18] F. Zahedi, The Analytic Hierarchy Process: A Survey of the Method and Its Applications, Interfaces 16 (4) (1986) 96–108.

[19] O. Vaidya, S. Agarwal, Analytic hierarchy process: An overview of applications, European Journal of Operational Research 169 (1) (2006) 1–29.

[20] T. L. Saaty, Fundamentals of decision making and priority theory with the AHP, RWS Publications, Pittsburgh, PA, 1994.

[21] W. C. Wedley, E. U. Choo, B. Schoner, Magnitude adjustment for AHP benefit/cost ratios, European Journal of Operational Research 133 (2) (2001) 342–351.

[22] V. Belton, T. Gear, On a short-coming of Saaty's method of analytic hierarchies, Omega 11 (3) (1983) 228–230.

[23] L. Vargas, An overview of the analytic hierarchy process and its applications, European Journal of Operational Research 48 (1).

[24] A. A. Salo, R. P. Hämäläinen, On the measurement of preferences in the analytic hierarchy process, Journal of Multi-Criteria Decision Analysis 6 (6) (1997) 309–319.

[25] S. Bozoki, An Analysis of the Analytic Hierarchy Process, MSc, Eotvos University, Budapest, 2001.

[26] Y. Dong, Y. Xu, H. Li, M. Dai, A comparative study of the numerical scales and the prioritization methods in AHP, European Journal of Operational Research 186 (1) (2008) 229–242.

[27] M. Fedrizzi, S. Giove, Incomplete pairwise comparison and consistency optimization, European Journal of Operational Research 183 (1) (2007) 303–313.

[28] M. Beynon, B. Curry, P. Morgan, The Dempster-Shafer theory of evidence: an alternative approach to multicriteria decision modelling, Omega 28 (1) (2000) 37–50.

[29] T. L. Saaty, Fundamentals of the analytic network process - Dependence and feedback in decision-making with a single network, Journal of Systems Science and Systems Engineering 13 (2004) 129–157.

[30] N. F. Zakaria, H. M. Dahlan, A. R. C. Hussin, Deriving priority in AHP using evolutionary computing approach, WSEAS Trans. Info. Sci. and App. 7 (5) (2010) 714–724.

[31] L. Mikhailov, J. Knowles, Priority Elicitation in the AHP by a Pareto Envelope-Based Selection Algorithm, in: Multiple Criteria Decision Making for Sustainable

Energy and Transportation Systems, vol. 36, Springer, 249–257, 2010.

[32] C. Macharis, J. Springael, K. D. Brucker, A. Verbeke, PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP, European Journal of Operational Research 153 (2) (2004) 307–317.

[33] B. Roy, Paradigms and Challenges, in: J. Figueira, S. Greco, M. Ehrgott (Eds.), Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, 4–24, 2005.

[34] W. Koczkodaj, A new definition of consistency of pairwise comparisons, Mathematical and Computer Modelling 18 (7) (1993) 79–84.

[35] L. Thurstone, A Law of Comparative Judgment, Psychological review 34 (4) (1927) 273–273.

[36] S. Merrill, A Comparison of Efficiency of Multicandidate Electoral Systems, American Journal of Political Science 28 (1) (1984) 23–48.

[37] Q. Zhang, H. Xue, C. Chen, C. Wu, An adaptive planning model based on multiagent, in: Machine Learning and Cybernetics, 2008 International Conference on, vol. 2, 1121–1126, 2008.

[38] I. Ali, W. Cook, M. Kress, On the Minimum Violations Ranking of a Tournament, Management Science 32 (6) (1986) 660–672.

[39] W. Cook, M. Kress, An mth Generation Model for Weak Ranking of Players in a Tournament, The Journal of the Operational Research Society 41 (12) (1990) 1111–1119.

[40] W. Koczkodaj, M. Orlowski, Computing a consistent approximation to a generalized pairwise comparisons matrix, Computers & Mathematics with Applications 37 (3) (1999) 79–85.

[41] P. Harker, L. Vargas, The Theory of Ratio Scale Estimation: Saaty's Analytic Hierarchy Process, Management Science 33 (11) (1987) 1383–1403.

[42] T. L. Saaty, A scaling method for priorities in hierarchical structures, Journal of Mathematical Psychology 15 (3) (1977) 234–281.

[43] S. Stevens, Toward a resolution of the fechner-thurstone legacy, Psychometrika 26 (1) (1961) 35–47.

[44] D. H. Krantz, Measurement Structures and Psychological Laws, Science 175 (4029) (1972) 1427–1435.

[45] J. S. Dyer, Remarks on the Analytic Hierarchy Process, Management Science 36 (3) (1990) 249–258.

[46] F. A. Lootsma, Scale sensitivity in the multiplicative AHP and SMART, Journal of Multi-Criteria Decision Analysis 2 (2) (1993) 87–110.

[47] J. S. Finan, W. J. Hurley, Transitive calibration of the AHP verbal scale, European Journal of Operational Research 112 (2) (1999) 367–372.

[48] A. Ishizaka, D. Balkenborg, T. Kaplan, Influence of aggregation and measurement scale on ranking a compromise alternative in AHP, Journal of the Operational Research Society (2010) 1–11.

[49] T. L. Saaty, Highlights and critical points in the theory and application of the Analytic Hierarchy Process, European Journal of Operational Research 74 (3)

(1994) 426–447.

[50] E. Choo, W. C. Wedley, A common framework for deriving preference values from pairwise comparison matrices, Computers & Operations Research 31 (6) (2004) 893–908.

[51] L. Mikhailov, M. Singh, Comparison analysis of methods for deriving priorities in the analytic hierarchy process, in: Systems, Man, and Cybernetics, IEEE International Conference on, vol. 1, 1037–1042, 1999.

[52] B. Srdjevic, Combining different prioritization methods in the analytic hierarchy process synthesis, Computers & Operations Research 32 (7) (2005) 1897–1919.

[53] K. K. F. Yuen, Analytic hierarchy prioritization process in the AHP application development: A prioritization operator selection approach, Applied Soft Computing 10 (4) (2010) 975–989.

[54] G. B. Crawford, The geometric mean procedure for estimating the scale of a judgement matrix, Mathematical Modelling 9 (3-5) (1987) 327–334.

[55] T. L. Saaty, Some Mathematical Concepts of the Analytic Hierarchy Process, Behaviormetrika 18 (29) (1991) 1–9.

[56] C. R. Johnson, W. Beine, T. J. Wang, Right-left asymmetry in an eigenvector ranking procedure, Journal of Mathematical Psychology 19 (1) (1979) 61–64.

[57] K. Cogger, P. L. Yu, Eigenweight vectors and least-distance approximation for revealed preference in pairwise weight ratios, Journal of Optimization Theory and Applications 46 (4) (1985) 483–491.

[58] B. Golany, M. Kress, A multi-criteria evaluation of methods for obtaining weights from ratio-scale matrices, European Journal of Operational Research 69 (2) (1993) 210–220.

[59] S. I. Gass, T. Rapcsak, Singular value decomposition in AHP, European Journal of Operational Research 154 (3) (2004) 573–584.

[60] Y. Wang, C. Parkan, Y. Luo, Priority estimation in the AHP through maximization of correlation coefficient, Applied Mathematical Modelling 31 (12) (2007) 2711–2718.

[61] A. Chu, R. Kalaba, K. Spingarn, A comparison of two methods for determining the weights of belonging to fuzzy sets, Journal of Optimization Theory and Applications 27 (4) (1979) 531–538.

[62] W. Cook, M. Kress, Deriving weights from pairwise comparison ratio matrices: An axiomatic approach, European Journal of Operational Research 37 (3) (1988) 355–362.

[63] R. E. Jensen, Comparisons of eigenvector, least squares, chi square, and logarithmic least squares methods of scaling a reciprocal matrix, working paper 153, Trinity University, 1983.

[64] Z. S. Zu, Generalized Chi Square Method for the Estimation of Weights, Journal of Optimization Theory and Applications 107 (2000) 183–192.

[65] N. Bryson, A Goal Programming Method for Generating Priority Vectors, The Journal of the Operational Research Society 46 (5) (1995) 641–648.

[66] N. Bryson, A. Joseph, Generating consensus priority point vectors: a logarithmic goal programming approach, Computers & Operations Research 26 (6) (1999)

637–643.

[67] C. C. Lin, An Enhanced Goal Programming Method for Generating Priority Vectors, The Journal of the Operational Research Society 57 (12) (2006) 1491–1496.

[68] L. Mikhailov, A Fuzzy Programming Method for Deriving Priorities in the Analytic Hierarchy Process, The Journal of the Operational Research Society 51 (3) (2000) 341–349.

[69] P. Laininen, R. P. Hämäläinen, Analyzing AHP-matrices by regression, European Journal of Operational Research 148 (3) (2003) 514–524.

[70] K. Sugihara, H. Ishii, H. Tanaka, Interval priorities in AHP by interval regression analysis, European Journal of Operational Research 158 (3) (2004) 745–754.

[71] D. Hartvigsen, Representing the strengths and directions of pairwise comparisons, European Journal of Operational Research 163 (2) (2005) 357–369.

[72] R. Ramanathan, Data envelopment analysis for weight derivation and aggregation in the analytic hierarchy process, Computers & Operations Research 33 (5) (2006) 1289–1307.

[73] Y. Wang, C. Parkan, Y. Luo, A linear programming method for generating the most favorable weights from a pairwise comparison matrix, Computers & Operations Research 35 (12) (2008) 3918–3930.

[74] D. Budescu, R. Zwick, A. Rapoport, A Comparison of the Eigenvalue Method and The Geometric Mean Procedure for Ratio Scaling, Applied Psychological Measurement 10 (1) (1986) 69–78.

[75] M. Herman, W. Koczkodaj, A Monte Carlo study of pairwise comparison, Information Processing Letters 57 (1) (1996) 25–29.

[76] B. Chandran, B. Golden, E. Wasil, Linear programming models for estimating weights in the analytic hierarchy process, Computers & Operations Research 32 (9) (2005) 2235–2254.

[77] L. D'Apuzzo, G. Marcarelli, M. Squillante, Generalized consistency and intensity vectors for comparison matrices, International Journal of Intelligent Systems 22 (12) (2007) 1287–1300.

[78] R. Sugden, Why be Consistent? A Critical Analysis of Consistency Requirements in Choice Theory, Economica 52 (206) (1985) 167–183.

[79] S. Lipovetsky, W. M. Conklin, Robust estimation of priorities in the AHP, European Journal of Operational Research 137 (1) (2002) 110–122.

[80] B. Schoner, W. C. Wedley, Ambiguous Criteria Weights in AHP: Consequences and Solutions, Decision Sciences 20 (3) (1989) 462–475.

[81] R. E. Jensen, T. Hicks, Ordinal data AHP analysis: A proposed coefficient of consistency and a nonparametric test, Mathematical and Computer Modelling 17 (4-5) (1993) 135–150.

[82] H. Monsuur, An intrinsic consistency threshold for reciprocal matrices, European Journal of Operational Research 96 (2) (1997) 387–391.

[83] R. Ramanathan, U. Ramanathan, A qualitative perspective to deriving weights from pairwise comparison matrices, Omega 38 (3-4) (2009) 228–232.

[84] M. Kendall, B. Smith, On the Method of Paired Comparisons, Biometrika

31 (3/4) (1940) 324–345.

[85] M. Kendall, Further Contributions to the Theory of Paired Comparisons, Biometrics 11 (1) (1955) 43–62.

[86] J. Aguaron, J. Moreno-Jimenez, The geometric consistency index: Approximated thresholds, European Journal of Operational Research 147 (1) (2003) 137–145.

[87] J. Alonso, T. Lamata, Consistency in the Analytic Hierarchy Process: A New Approach, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14 (4) (2006) 445–459.

[88] G. B. Crawford, C. Williams, A note on the analysis of subjective judgment matrices, Journal of Mathematical Psychology 29 (4) (1985) 387–405.

[89] F. J. Dodd, H. A. Donegan, T. B. M. McMaster, A statistical approach to consistency in AHP, Mathematical and Computer Modelling 18 (6) (1993) 19–22.

[90] C. K. Murphy, Limits on the analytic hierarchy process from its consistency index, European Journal of Operational Research 65 (1) (1993) 138–139.

[91] W. E. Stein, P. J. Mizzi, The harmonic consistency index for the analytic hierarchy process, European Journal of Operational Research 177 (1) (2007) 488–497.

[92] S. Bozoki, T. Rapcsak, On Saaty's and Koczkodaj's inconsistencies of pairwise comparison matrices, Journal of Global Optimization 42 (2) (2008) 157–175.

[93] W. Koczkodaj, S. J. Szarek, On distance-based inconsistency reduction algorithms for pairwise comparisons, Logic Journal of the IGPL (2010) jzp062.

[94] M. Kwiesielewicz, E. van Uden, Inconsistent and contradictory judgements in pairwise comparison method in the AHP, Computers & Operations Research 31 (5) (2004) 713–719.

[95] S. Gass, Tournaments, Transitivity and Pairwise Comparison Matrices, The Journal of the Operational Research Society 49 (6) (1998) 616–624.

[96] Y. Iida, The Number of Circular Triads in a Pairwise Comparison Matrix and a Consistency Test in the AHP, Journal of the Operations Research, Society of Japan 52 (2) (2009) 174–185.

[97] X. Zeshui, W. Cuiping, A consistency improving method in the analytic hierarchy process, European Journal of Operational Research 116 (2) (1999) 443–449.

[98] D. Cao, L. Leung, J. Law, Modifying inconsistent comparison matrix in analytic hierarchy process: A heuristic approach, Decision Support Systems 44 (4) (2008) 944–953.

[99] W. Holsztynski, W. W. Koczkodaj, Convergence of inconsistency algorithms for the pairwise comparisons, Information Processing Letters 59 (4) (1996) 197–202.

[100] M. A. Seaman, J. R. Levin, R. C. Serlin, New Developments in Pairwise Multiple Comparisons: Some Powerful and Practicable Procedures, Psychological Bulletin 110 (3) (1991) 577 – 586.

[101] W. Koczkodaj, Testing the accuracy enhancement of pairwise comparisons by a Monte-Carlo experiment, Journal of Statistical Planning and Inference 69 (1) (1998) 21–31.

[102] A. Ishizaka, M. Lusti, How to derive priorities in AHP: a comparative study, Central European Journal of Operations Research 14 (4) (2006) 387–400.

[103] C. Lin, A revised framework for deriving preference values from pairwise comparison matrices, European Journal of Operational Research 176 (2) (2007) 1145–1150.

[104] F. Harary, L. Moser, The Theory of Round Robin Tournaments, The American Mathematical Monthly 73 (3) (1966) 231–246.

[105] G. Even, J. Naor, B. Schieber, M. Sudan, Approximating Minimum Feedback Sets and Multi-cuts in Directed Graphs, Algorithmica 20 (2) (1998) 151–174.

[106] P. Slater, Inconsistencies in a Schedule of Paired Comparisons, Biometrika 48 (3/4) (1961) 303–312.

[107] P. Pardalos, T. Qian, M. Resende, A Greedy Randomized Adaptive Search Procedure for the Feedback Vertex Set Problem, Journal of Combinatorial Optimization 2 (4) (1998) 399–412.

[108] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, J. Martin, PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001, Morgan Kaufmann Publishers, 283–290, 2001.

[109] H. N. Gabow, E. W. Myers, Finding All Spanning Trees of Directed and Undirected Graphs, SIAM Journal on Computing 7 (3) (1978) 280–287.

[110] A. Cayley, A theorem on trees, Quart. J. Math 23 (1889) 376–378.

[111] P. T. Harker, Incomplete pairwise comparisons in the analytic hierarchy process, Mathematical Modelling 9 (11) (1987) 837–848.

[112] W. Tutte, Graph Theory, Cambridge University Press, ISBN 978-0-521-79489-3, 2001.

[113] P. T. Harker, Alternative modes of questioning in the analytic hierarchy process, Mathematical Modelling 9 (3-5) (1987) 353–360.

[114] G. Kirchhoff, Ueber die Auflosung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Strome gefuhrt wird, Annalen der Physik und Chemie 148 (12) (1847) 497–508.

[115] S. Chaiken, D. J. Kleitman, Matrix Tree Theorems, Journal of Combinatorial Theory, Series A 24 (3) (1978) 377–381.

[116] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, E. Alba, jMetal: A Java Framework for Developing Multi-Objective Optimization Meta-heuristics, Tech. Rep. ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, 2006.

[117] C. Brugha, Relative measurement and the power function, European Journal of Operational Research 121 (3) (2000) 627–640.

[118] Y. Wang, K. Chin, Y. Luo, Aggregation of direct and indirect judgments in pairwise comparison matrices with a re-examination of the criticisms by Bana e Costa and Vansnick, Information Sciences 179 (3) (2009) 329–337.

[119] E. Forman, T. Saaty, M. Selly, R. Waldron, Expert Choice, Decision Support Software, McLean, VA, 1983.

[120] R. P. Hämäläinen, H. Lauri, HIPRE 3+ User's Guide, Systems Analysis Laboratory, Helsinki University of Technology, URL http://www.sal.hut.fi/Downloadables/hpdemo.html, 1995.

[121] ISOIEC JTC1 Directives, Procedures for the Technical Work of ISO/IEC JTC1, URL http://www.iso.org/iso/jtc1_home.html, 1999.

[122] Y. Gasiea, An Analytic Decision Approach to Rural Telecommunication Infrastructure Selection, Ph.D. thesis, Manchester School of Mechanical, Aerospace and Civil Engineering, University of Manchester, 2010.

[123] L. Elsner, P. van den Driessche, Max-algebra and pairwise comparison matrices, II, Linear Algebra and its Applications 432 (4) (2010) 927–935.