

# Preference Elicitation from Pairwise Comparisons for Traceable Multi- Criteria Decision Making

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the  
faculty of Engineering and Physical Sciences

2015

By  
Edward Abel  
School of Computer Science

# Contents

<b>DECELERATION .....</b>	<b>14</b>
<b>COPYRIGHT .....</b>	<b>15</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>16</b>
<b>ABBREVIATIONS AND ACRONYMS.....</b>	<b>17</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>18</b>
<b>1.1 Motivation.....</b>	<b>18</b>
<b>1.2 Aims and Objectives .....</b>	<b>19</b>
<b>1.3 Contributions.....</b>	<b>21</b>
1.3.1 Measures of Compromise .....	21
1.3.2 Approach to inconsistency reduction .....	21
1.3.3 Approach to group aggregation.....	21
1.3.4 Approach to large group aggregation.....	22
1.3.5 MOODS decision support tool.....	22
1.3.6 Publications .....	23
<b>1.4 Structure .....</b>	<b>23</b>
<b>CHAPTER 2 BACKGROUND .....</b>	<b>25</b>
<b>2.1 Multi-Criteria Decision Analysis .....</b>	<b>25</b>
2.1.1 Decision Problem Dimensions.....	25
2.1.2 Common MCDA problem elements, stages and outcomes .....	27
2.1.3 MCDA methods .....	29
2.1.4 MCDA Conclusions .....	34
<b>2.2 Pairwise Comparison .....</b>	<b>34</b>
2.2.1 PC Notation, Properties and Graph Visualisation.....	35

2.2.2	Judgment Scale Functions.....	37
2.2.3	Prioritization Methods.....	39
2.2.4	Pairwise Comparison Conclusions.....	43
<b>2.3</b>	<b>Inconsistency.....</b>	<b>43</b>
2.3.1	Ordinal and Cardinal Inconsistency.....	44
2.3.2	Ordinal Inconsistency Measures.....	46
2.3.3	Cardinal Inconsistency Measures.....	48
2.3.4	Reducing Inconsistency.....	50
2.3.5	Previous approaches to automated inconsistency reduction.....	51
2.3.6	Inconsistency Conclusions.....	54
<b>2.4</b>	<b>Group Decision Making.....</b>	<b>55</b>
2.4.1	Group MCDA.....	55
2.4.2	Aggregation of Individual Priorities.....	59
2.4.3	Aggregation of Individual Judgments.....	61
2.4.4	Other aggregation Approaches.....	64
2.4.5	Aggregation of PCs Discussions.....	65
<b>2.5</b>	<b>Summary.....</b>	<b>67</b>
<b>CHAPTER 3 MEASURES OF COMPROMISE.....</b>		<b>68</b>
<b>3.1</b>	<b>Introduction.....</b>	<b>68</b>
<b>3.2</b>	<b>Judgment Set Representation.....</b>	<b>69</b>
<b>3.3</b>	<b>Measures of compromise.....</b>	<b>71</b>
3.3.1	Number of Judgment Violations.....	72
3.3.2	Total Judgment Deviation.....	73
3.3.3	Number of Judgment Reversals.....	74
<b>3.4</b>	<b>Measures of Compromise Example.....</b>	<b>76</b>
<b>3.5</b>	<b>Conclusions.....</b>	<b>77</b>

**CHAPTER 4 REDUCING INCONSISTENCY IN PAIRWISE  
COMPARISONS USING MULTI-OBJECTIVE EVOLUTIONARY  
COMPUTING 79**

<b>4.1</b>	<b>Automatically Reducing Inconsistency in Pairwise Comparison .....</b>	<b>79</b>
<b>4.2</b>	<b>Approach to reducing inconsistency in a set of PCs .....</b>	<b>80</b>
4.2.1	Stages of the Approach to reducing inconsistency in PCs.....	80
4.2.2	Multi-Objective Optimisation (MOO).....	82
4.2.3	Objectives usable within the Approach.....	85
4.2.4	Implementation Analysis .....	87
4.2.5	Constraints .....	99
4.2.6	Approach to the reduction of the inconsistency of a set of PCs Discussions	101
<b>4.3</b>	<b>Experimentation Examples .....</b>	<b>101</b>
4.3.1	Example 4.1: Comparison against other approaches .....	102
4.3.2	Example 4.2: Inconsistency Measures .....	105
4.3.3	Example 4.3: Interactively adding constraints .....	108
4.3.4	Example 4.4: Larger objective sets .....	110
<b>4.4</b>	<b>Conclusions .....</b>	<b>112</b>

**CHAPTER 5 GROUP AGGREGATION OF PAIRWISE COMPARISONS  
USING MOO 113**

<b>5.1</b>	<b>Aggregation of PC judgments of a group of DMs.....</b>	<b>113</b>
<b>5.2</b>	<b>MOO Approach to PC aggregation.....</b>	<b>114</b>
5.2.1	Stages of Approach to Aggregation of PC judgments of a group of DMs ...	116
5.2.2	Multi Objective Optimisation .....	118
5.2.3	Objectives usable within the Approach.....	121
5.2.4	Analysis of solutions to aid towards selection of a single solution .....	122
5.2.5	MOO Approach to PC aggregation Discussion .....	131
<b>5.3</b>	<b>Examples .....</b>	<b>132</b>
5.3.1	Example 5.1: Global solution analysis and comparison to GMM .....	132
5.3.2	Example 5.2: DM weights of importance analysis .....	135
5.3.3	Example 5.3: Interactively adding constraints .....	140

5.3.4	Example 5.4: Incorporating inconsistency reduction during aggregation ....	144
<b>5.4</b>	<b>MOO Approach to PC aggregation within AHP Decision .....</b>	<b>147</b>
5.4.1	Formulation of the decision problem .....	147
5.4.2	Elicitation, aggregation and analysis of criteria priorities .....	149
5.4.3	Elicitation, aggregation and analysis of alternatives priorities with respect to each criterion .....	150
5.4.4	Synthesis to a final ranking .....	152
5.4.5	AHP Example conclusions .....	153
<b>5.5</b>	<b>Conclusions .....</b>	<b>154</b>
<b>CHAPTER 6 CLUSTERING DECISION MAKERS WITH RESPECT TO SIMILARITY OF VIEWS .....</b>		<b>156</b>
<b>6.1</b>	<b>Introduction .....</b>	<b>156</b>
<b>6.2</b>	<b>Aggregation of Many Decision Makers .....</b>	<b>156</b>
6.2.1	Multi-Objective Optimisation with large objective sets .....	156
6.2.2	Analysis of MOO Aggregation Approach for large group of DMs' PCs .....	157
6.2.3	Multi-objective optimisation with large numbers of objectives .....	162
<b>6.3</b>	<b>Approach for aggregation of a large group of DMs' PCs .....</b>	<b>163</b>
6.3.1	Stages of Approach for aggregation of a large group of DMs' PCs .....	164
6.3.2	Stage 1: Clustering the group of DMs .....	165
6.3.3	Stage 2: Deriving a single representation for each cluster .....	167
6.3.4	Clustering evaluation experimentation .....	169
6.3.5	Stage 3: Multi-Objective Optimisation .....	176
6.3.6	Approach for aggregation of a large group of DMs' PCs Discussions .....	180
<b>6.4</b>	<b>Examples .....</b>	<b>180</b>
6.4.1	Example 6.1: Small DM group .....	181
6.4.2	Example 6.2: Large number of DMs .....	184
<b>6.5</b>	<b>Conclusions .....</b>	<b>190</b>
<b>CHAPTER 7 MOODS: A WEB BASED DECISION SUPPORT TOOL .....</b>		<b>191</b>

<b>7.1</b>	<b>Introduction .....</b>	<b>191</b>
<b>7.2</b>	<b>Architecture Design .....</b>	<b>191</b>
<b>7.3</b>	<b>Interface Implementation .....</b>	<b>194</b>
7.3.1	Problem Setup .....	194
7.3.2	DM Elicitation.....	196
7.3.3	Objective Space.....	197
7.3.4	Table Results.....	199
7.3.5	Clustering Setup .....	200
7.3.6	Clustering Results .....	201
7.3.7	Clustering Evaluation.....	202
<b>7.4</b>	<b>Usage Scenarios .....</b>	<b>203</b>
7.4.1	Reducing inconsistency for single DM .....	203
7.4.2	Group Aggregation.....	204
7.4.3	Large Group Aggregation .....	204
<b>7.5</b>	<b>Conclusions .....</b>	<b>205</b>
<b>CHAPTER 8 CONCLUSIONS AND FUTURE WORK.....</b>		<b>209</b>
<b>8.1</b>	<b>Summary.....</b>	<b>209</b>
<b>8.2</b>	<b>Future Work.....</b>	<b>210</b>
8.2.1	Modified Measures of Compromise.....	210
8.2.2	Additional Clustering Work.....	215
8.2.3	Further investigations of large objective set optimisation .....	217
8.2.4	MOODS tool extensions and enhancements.....	218
<b>8.3</b>	<b>Conclusion.....</b>	<b>219</b>
<b>BIBLIOGRAPHY .....</b>		<b>220</b>

Word Count: 64306

# List of Tables

Table 2.1: Decision problem dimensions.....	26
Table 2.2: Saaty 1-9 scale mapping from verbal scale.....	37
Table 2.3: Inconsistency Measures Example.....	46
Table 2.4: Inconsistency Example Measures values.....	50
Table 2.5: Example Measures values [81].....	52
Table 2.6: Inconsistency Reduction Approaches.....	54
Table 2.7: 4DM Group Example Data.....	60
Table 2.8: 4DM Group Example Separate Preference Vectors.....	60
Table 2.9: 4DM Group AMM aggregation of priorities Preference vector.....	61
Table 2.10: 4DM GMM Aggregated PCM.....	63
Table 2.11: 4DM Group GMM aggregation of priorities Preference vector.....	63
Table 3.1: Example measures calculation.....	76
Table 4.1: Exhaustive search sequence complexities.....	90
Table 4.2 : MOGA Setup parameters.....	94
Table 4.3: N=3 Example: MOGA Evaluation.....	95
Table 4.4: N=4 Example: MOCeIl Evaluation.....	97
Table 4.5: Example 4.1: [CR: 0.17].....	102
Table 4.6: Example 4.1 possible solution [CR: 0.098 TJD: 7].....	104
Table 4.7: Example 4.1 solution from Xu and Wei [81] [CR: 0.097 TJD: 14.219].....	104
Table 4.8: Example 4.1 solution from Cao et al [82] [CR: 0.099 TJD:15.63].....	105
Table 4.9: Example 4.2 Judgments.....	106
Table 4.10: Example 4.3 Judgments [CR: 1.08].....	108
Table 4.11: Example 4.3 Constrained objective space solutions.....	110
Table 5.1: Example 5.1: D: 2 N: 5.....	133
Table 5.2: Example 5.1: STJD Measures.....	134
Table 5.3: Example 5.1: GMM PCM.....	134
Table 5.4: Example 5.1: MOO Global Solution PCM.....	134
Table 5.5: Example 5.1: Preference vector from global solution of MOO approach ...	135
Table 5.6: Example 5.2: Preference vectors from weights analysis.....	136
Table 5.7: Example 5.2: D: 4 N: 6.....	136
Table 5.8: Example 5.2: Agreement values between each pair of DMs.....	137
Table 5.9: Example 5.2: Average Agreement values of each DM.....	137

Table 5.10: Example 5.2: Weighted solutions compromise values .....	138
Table 5.11: Example 5.2: Weighted solutions preference vectors.....	139
Table 5.12: Example 5.3: 3 DMs' Judgments.....	140
Table 5.13: Example 5.3: Initial significant solutions data.....	141
Table 5.14: Example 5.3: Initial judgment sets compromise.....	141
Table 5.15: Example 5.3: chosen solution and preference vector.....	144
Table 5.16: Example 5.4 Judgments .....	145
Table 5.17: PCM relating to criteria preferences of the three DMs.....	149
Table 5.18: Aggregated PCM from GMM of the criteria preferences.....	150
Table 5.19: Aggregated criteria weights from MOO and GMM .....	150
Table 5.20: MOO and GMM aggregation of alternatives with respect to Expansion Capacity.....	151
Table 5.21: Alternatives preference vectors with respect to Expansion Capacity .....	152
Table 5.22: Aggregated preference vectors for alternatives with respect to criterion ..	152
Table 5.23: Final alternatives rankings for MOO and GMM approaches .....	153
Table 6.1: Cluster analysis example data.....	171
Table 6.2: Example 6.1: 6 DMs' Judgments.....	181
Table 6.3: Example 6.1: k=2 Aggregated PCMs .....	183
Table 6.4: Example 6.1: Preference vector derived from weighted global solution.....	184
Table 6.5: Example 6.2: k=3 aggregated PCMs .....	186
Table 6.6: Example 6.2: k=3 weighted global solution preference vector .....	187
Table 6.7: Example 6.2: cluster measures analysis.....	188
Table 6.8: Example 6.2: k=2 aggregated PCMs .....	188
Table 6.9: Example 6.2: k=2 weighted global solution preference vector .....	190
Table 8.1: Correlation between measures of compromise, N=5 .....	211

# List of Figures

Figure 1.1: Overview of thesis chapters.....	24
Figure 2.1: AHP hierarchical example structure.....	32
Figure 2.2: Wealth of nations DAG.....	36
Figure 2.3: Verbal scale function mapping onto numerical scales.....	39
Figure 2.4: Left: Ordinal inconsistency. Right: Cardinal consistency.....	45
Figure 2.5: Inconsistency Measures Example DAG.....	47
Figure 2.6: 4DM aggregation of individual judgments.....	58
Figure 2.7: 4DM aggregation of individual priorities.....	59
Figure 2.8: Ordinal inconsistency increasing during GMM aggregation.....	64
Figure 2.9: CR value greater than 0.1 during GMM aggregation.....	65
Figure 3.1: Example encoding of set of judgments for an N=4 PCM.....	70
Figure 3.2: Judgment Encoding to ensure equal alteration across the scale.....	71
Figure 3.3: Measures of compromise.....	72
Figure 3.4: 1-9 scale value Key.....	72
Figure 3.5: Total Judgment Deviation (TJD).....	74
Figure 3.6: Squared Total Judgment Deviation (STJD).....	74
Figure 3.7: Number of Judgment Reversals (NJR).....	75
Figure 3.8 Left: Initial Judgment set. Right: Judgment set with inconsistency removed.....	76
Figure 3.9: Measures of Compromise of difference between DM views.....	78
Figure 4.1: Flowchart of Approach to reducing inconsistency in a set of PCs.....	81
Figure 4.2 Left: Single objective problem. Right: Illustrative Pareto dominance definitions.....	83
Figure 4.3: Illustration of MOO approach.....	85
Figure 4.4: MOO objectives.....	87
Figure 4.5: GA general stages.....	88
Figure 4.6 Left: Example true front Right: Example found front.....	91
Figure 4.7 Left: GD, Centre: IGD, Right: Spread.....	92
Figure 4.8 : Example hypervolume calculation.....	93
Figure 4.9: N=3 exhaustive search example.....	93
Figure 4.10: N=3 exhaustive search example true front.....	94
Figure 4.11: N=3 Example: Average Hypervolume % and stability.....	95
Figure 4.12: N=4 Exhaustive Search Example.....	96

Figure 4.13: N=4 Example: Average Hypervolume % and stability .....	98
Figure 4.14 Left: N=4 Solution set found via MOCeII. Right: N=4 true Front.....	98
Figure 4.15: Constrained Pareto Dominance during 2-Objective minimisation.....	100
Figure 4.16: Example 4.1 Solution Space.....	103
Figure 4.17: Example deviation calculation outside whole scale step.....	105
Figure 4.18: Example 4.2 Left: CR and STJD objectives. Right: L and STJD objectives .....	106
Figure 4.19 Example 4.2 Left: CM and STJD objectives. Right: GCI and STJD objectives .....	107
Figure 4.20: Example 4.3 Solution Space. CR and TJD objectives.....	108
Figure 4.21 Left: Example 4.3 Objective Space with CR constraint. Right: Example 4.3 objective space with CR and TJD constraints.....	109
Figure 4.22: Example 4.4 Additional Objectives Solution Space analysis.....	111
Figure 5.1: Flowchart of the stages of the approach of aggregation of PC judgments of a group of DMs .....	117
Figure 5.2: Illustration of MOO approach of aggregation of 3 DMs' preferences .....	120
Figure 5.3: Global solution calculation illustration.....	125
Figure 5.4 Left: Weights DM1:2/3 DM2:1/3. Right: Weights DM1: 0.75 DM2:0.25 .	127
Figure 5.5: Analysis of weighted values .....	128
Figure 5.6: Equal compromise range calculation illustration .....	130
Figure 5.7: TSTJD and range values from the equal compromise illustrative example .....	131
Figure 5.8: Example 5.1 objective space .....	133
Figure 5.9 Left Weights {DM1:0.65, DM2:0.35}. Right Weights {DM1:0.65, DM2:0.35} .....	135
Figure 5.10: Example 5.2 Dynamic Weights Analysis DM1 & DM4 Views.....	138
Figure 5.11: Example 5.2 Dynamic Weights Analysis DM1 & DM3 Views.....	139
Figure 5.12: Example 5.3: Initial objective space DM1 and DM2 view .....	141
Figure 5.13 Example 5.3: Left: DM2 constraints. Right: DM2 and DM1 constraints	142
Figure 5.14 Example 5.3: Left: DM2 and DM1 constraints. Right: DM2, DM1 and DM3 constraints .....	143
Figure 5.15: Example 5.3: Final Constrained objective space DM1 and DM2 view ...	143
Figure 5.16 Left: Example 5.4 from DM1 and DM2 view (with CR as 3 <sup>rd</sup> objective). Right: Example 5.4 From CR and Total compromise view.....	145

Figure 5.17 Left: Example 5.4 from DM1 and DM2 view (with L as 3 <sup>rd</sup> Objective). Right: Example 5.4 from L and Total compromise view.....	146
Figure 5.18: Hierarchy representation of the AHP example decision .....	148
Figure 5.19: Criteria with respect to the problem .....	149
Figure 5.20: Alternatives compared with respect to Expansion Capacity criterion.....	151
Figure 6.1: Average size of true fronts and average size of fronts from MOCeCell.....	159
Figure 6.2: 20 DM objective space from DM1 and DM2 view .....	160
Figure 6.3: Average size of fronts found from MOCeCell as % of the size of true fronts	161
Figure 6.4 : Average GD and IGD of MOCeCell MOGA over range of DM values. ....	162
Figure 6.5: Flowchart of the approach to aggregation of a large group of DMs .....	165
Figure 6.6: Illustration of clustering input for 3 DMs' judgments.....	167
Figure 6.7: Example of 3 cluster member's judgment's as DAGs .....	168
Figure 6.8: Examples of single representation judgment's as DAGs .....	168
Figure 6.9: Evaluation of clustering values.....	170
Figure 6.10: Analysis of objective reduction vs. information loss .....	171
Figure 6.11: DM50 example analysis of objective reduction vs. information loss .....	172
Figure 6.12: Monte Carlo experimentation process .....	173
Figure 6.13: Average Redundancy N=3 D=20 .....	174
Figure 6.14: Rate of Average Redundancy N=3 D=20.....	175
Figure 6.15: Average Redundancy over range of N values when D=20 .....	175
Figure 6.16: Average Redundancy N=5 D=50 .....	176
Figure 6.17: Illustration of MOO stage finding global compromise .....	178
Figure 6.18: Illustration of MOO stage with cluster weights based on cluster size.....	179
Figure 6.19: Example 6.1: k-value analysis .....	182
Figure 6.20: Example 6.1: k=2 objective space .....	183
Figure 6.21: Example 6.2 recycling plant alternative locations .....	184
Figure 6.22: Example 6.2: Analysis over range of k values .....	185
Figure 6.23: Example 6.2: k=3 objective space .....	187
Figure 6.24: Example 6.2: k=2 objective space .....	189
Figure 7.1: Overview of MOODS architecture.....	192
Figure 7.2: Overview of Interface .....	195
Figure 7.3: Problem Setup panel .....	196
Figure 7.4: DM Elicitation panel .....	197
Figure 7.5: Objective Space panel .....	198
Figure 7.6: Table Results panel.....	199

Figure 7.7: Table Results panel with expanded Judgments panel .....	200
Figure 7.8 Clustering Setup panel .....	201
Figure 7.9: Clustering Results panel .....	202
Figure 7.10: Clustering Evaluation panel .....	203
Figure 7.11: MOODS use during single DM inconsistency reduction .....	206
Figure 7.12: MOODS use during group aggregation.....	207
Figure 7.13: MOODS use during large group aggregation.....	208
Figure 8.1: N=5. STJD and NJR correlation.....	212
Figure 8.2 : Deviation and reversal measures comparison .....	213
Figure 8.3: Total judgment deviation with reversals modification measure.....	213
Figure 8.4: Modified TJD (MTJD). .....	214
Figure 8.5 : Left: Hierarchal clustering. Right: Fuzzy clustering .....	217

**University Name:** The University of Manchester

**Candidate Name:** Edward Abel

**Degree Title:** Doctor of Philosophy

**Thesis Title:** Preference Elicitation from Pairwise Comparisons for Traceable Multi-Criteria Decision Making

**Date:** 30<sup>th</sup> September 2015

## **Abstract**

For many decisions validation of their outcomes is invariably problematic to objectively assess. Therefore to aid analysis and validation of decision outcomes, approaches which provide improved traceability and more semantically meaningful measurements of the decision process are required. Hence, this research investigates traceability, transparency, interactivity and auditability to improve the decision making process. Approaches and evaluation measures are proposed to facilitate a richer decision making experience.

Multi-Criteria Decision Analysis (MCDA) seeks to determine the suitability of alternatives of a goal with respect to multiple criteria. A key component of prominent MCDA methods is the concept of pairwise comparison. For a set of elements, pairwise comparison enables an accurate and transparent extraction and codification of a decision maker's preferences, though facilitating a separation of concerns. From a set of pairwise comparisons, a ranking of the elements under consideration can be calculated.

There are scenarios when a set of pairwise comparisons undergo alteration, both for individual and multiple decision makers. A set of measures of compromise are proposed to quantify the alteration that a set of pairwise comparisons undergo in such scenarios. The measures seek to provide a decision maker with meaningful knowledge regarding how their views have altered.

A set of pairwise comparisons may be inconsistent. When inconsistency is present it adversely affects a ranking of the elements derived from the comparisons. Moreover inconsistency within pairwise comparisons used for consideration of more than a handful of elements is almost inevitable. Existing approaches that seek to alter a set of comparisons to reduce inconsistency lack traceability, flexibility, and specific consideration of alteration to the judgments in a way that is meaningful to a decision maker. An approach to inconsistency reduction is proposed that seeks to address these issues.

For many decisions the opinions of multiple decision makers are utilized, either to avail of their combined expertise or to incorporate conflicting views. Aggregation of multiple decision makers' pairwise comparisons seek to combine the views of the group into a single representation of views. An approach to group aggregation of pairwise comparisons is proposed that models compromise between the decision makers, facilitates decision maker constraints, considers inconsistency reduction during aggregation and dynamically incorporates decision maker weights of importance.

With internet access becoming widespread being able to garner the views of a large group of decision makers' views has become feasible. An approach to the aggregation of a large group of decision makers' preferences is proposed. The approach facilitates understanding regarding both the agreement and conflict within the group during calculation of an overall group consensus.

A Multi-Objective Optimisation Decision Software (MOODS) prototype tool has been developed that implements both the new measures of compromise and the proposed approaches to inconsistency reduction and group aggregation.

# **Deceleration**

No portion of the work referred to in this thesis 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

- i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s policy on Presentation of Theses

# **Acknowledgements**

I wish to thank my supervisors John Keane and Ludmil Mikhailov for all their support, and guidance during my research studies. Additionally I wish to acknowledge the support of the Engineering and Physical Science Research Council (EPSRC) for funding my research via their doctoral training grant. I wish to also thank the researchers and institutions whose open source software and tools aided my research studies.

# Abbreviations and Acronyms

<b>AHP</b>	Analytic Hierarchy Process
<b>CM</b>	Consistency Measure
<b>CR</b>	Consistency Ratio
<b>DM</b>	Decision Maker
<b>DSS</b>	Decision Support System
<b>EV</b>	Eigenvector prioritization method
<b>GCI</b>	Geometric Consistency Index
<b>GD</b>	Generational Distance
<b>GM</b>	Geometric Mean prioritization method
<b>HV</b>	HyperVolume
<b>IGD</b>	Inverted Generation Distance
<b>L</b>	Number of 3 way cycles
<b>MCDA</b>	Multiple Criteria Decision Analysis
<b>MOCeII</b>	Multi-Objective Cellular Algorithm
<b>MOGA</b>	Multi-Objective-Genetic Algorithm
<b>MOO</b>	Multi-Objective Optimisation
<b>MOODS</b>	Multi-Objective Optimisation Decision Software
<b>NJR</b>	Number of Judgment Reversals
<b>NJV</b>	Number of Judgment Violations
<b>NSGAI</b>	Non-dominated Sorting Genetic Algorithm II
<b>NV</b>	Number of Violations
<b>PC</b>	Pairwise Comparison
<b>PCM</b>	Pairwise Comparison Matrix
<b>SPEA2</b>	Strength Pareto Evolutionary Algorithm 2
<b>STJD</b>	Squared Total Judgment Deviation
<b>TD</b>	Total Deviation
<b>TID</b>	Total Indirect Deviation
<b>TJD</b>	Total Judgment Deviation

# Chapter 1 Introduction

This chapter provides an introduction to the research presented in this thesis. After a discussion of the motivations and aims of the research, the contributions of the thesis are outlined. Finally an overview of the structure of the thesis is presented.

## 1.1 Motivation

For many decisions validation of their outcomes regarding their correctness and their acceptance is invariably problematic to objectively assess. This research investigates traceability, transparency, interactivity and auditability within decision making procedures to seek a richer decision making experience – we will discuss later what we mean by these terms. As Decision Makers (DM)s we are subject to fragilities such as, biases, inconsistencies and irrationalities [1], and are often confronted with decisions where multiple minds are tasked to work together to reach, where differing opinions may exist, a compromise consensus. The work investigates how we can identify and tackle the impacts of these fragilities in a more interactive and traceable way, and how we can enhance the transparency of group decisions to more clearly reveal compromise, to facilitate a more traceable and auditable approach to interactively reach consensus.

A Decision Support System (DSS), can be defined as *“a system that couples the intellectual resources of an individual with the capabilities of the computer to ultimately improve the quality of decisions”* [2]. There are various ways a DSS can enhance decision making, such as providing structure to overcome short term memory limitations [3] and performing complex numerical calculations to a high degree of accuracy.

Multi-Criteria Decision Analysis (MCDA) seeks to determine the suitability of alternatives of a goal with respect to multiple criteria. Various MCDA methods have been proposed to determine the suitability of a decision’s alternatives and to derive various granularities of decision outcomes.

A key component of prominent MCDA methods is the concept of pairwise comparison (PC). PC enables the decomposition of a larger decision problem into more manageable smaller chunks, facilitating a separation of concerns that enables an accurate extraction of a DM’s preferences. For a set of elements under consideration a PC judgment can be made for each pair of elements and from this set of comparison

judgments a one-dimensional ranking of the elements, a Preference Vector, can be derived. A preference vector is derived through the use of a Prioritization Method and many methods have been proposed for this task.

The unification of the smaller chunks of each PC judgment may result in inconsistency being present in the set of judgments as a whole. When inconsistency is present in a set of judgments any preference vector derived will only be an estimate of the judgments' information. Consequently, different prioritization methods may derive different preference vector estimates. Moreover inconsistency within PC used for consideration of more than a handful of elements is almost inevitable [4]. As inconsistency within a set of DM judgments can adversely affect the accuracy of a resulting preference vector consideration of its reduction is important. Current approaches to reducing inconsistency within a set of PCs offer little traceability, flexibility or specific consideration of alteration to the judgments in a way that is semantically meaningful to a DM.

For many real-world decisions the opinions of multiple DMs is utilised, either to avail of their combined expertise or to incorporate conflicting views and experiences, and therefore group aggregation is an important consideration. Furthermore with ubiquitous access to the internet via a multitude of devices becoming widespread being able to garner and aggregate the views of a large group of DMs' views has become feasible. Group aggregation of PCs seek to aggregate the views of multiple DMs to reach a single consensus preference vector. Current approaches for PC aggregation lack facilities to, model in semantically meaningful ways the compromise that each DM's views undergo during aggregation, incorporate DM constraints, consider inconsistency during aggregation and dynamically incorporate DM's weights of importance.

## **1.2 Aims and Objectives**

As validation of decisions outcomes is invariably problematic to objectively assess, approaches with improved traceability and more semantically meaningful measurements would aid a DM through providing more evidence of the decision process. The traceability of a decision process is the extent to which a trail of documentation and measurements are revealed during the decision from its inputs to its outcomes. Such a trail can reveal to a DM and others, quantitative measures, trade-offs and choices during the decision to reach the outcomes, aiding both transparency and auditability. Transparency is the extent to which the process of decision making to reach an outcome is exposed and can be observed for scrutiny, explanation and understanding. The

auditability of a decision process is the extent to which it can be inspected and examined, for evaluation and verification purposes. Traceability trails additionally make it easier to perform sensitivity analysis to investigate how variations in the decision's inputs or processes might affect the outcomes, helping facilitate more interactive decision making. Such traceability and sensitivity analysis can help decision outcomes be more objectively assessed and can increase a DM's acceptance of the outcomes.

The traceability of a decision process can be enhanced through looking to reveal measurements that are semantically meaningful - that is, measurements that a DM can more easily comprehend and relate to, that will enhance the knowledge and understanding they can glean from the decision. More meaningful measures can additionally facilitate greater DM interactivity and sensitivity analysis during a decision, further helping validation and acceptance of decision outcomes.

This work seeks to propose more traceable, transparent, and auditable approaches which utilise more semantically meaningful measures, thus facilitating richer decision making through approaches that are both more systematic and dynamic.

We seek firstly to facilitate richer decision making by proposing measures to reveal semantically meaningful knowledge to a DM. This way a DM can more easily comprehend outcomes and the stages to reach them. Semantically meaningful measures can additionally enable greater interaction from a DM both when defining inputs and parameters, and for exploration tasks supporting a DM towards outcomes.

We seek to additionally facilitate richer decision making through proposing approaches which enhance traceability, flexibility and auditability. We propose a flexible approach to the reduction of inconsistency within a set of PCs that seeks to reveal the trade-offs involved when seeking inconsistency reduction. Additionally we propose an approach to the aggregation of the PCs of a group of DMs' that seeks to reveal the amount of compromise each DM undergoes to reach consensus. The approach aids a group in interactively and traceably reaching a consensus so as to aid transparency and auditability of the process to reach a consensus. Furthermore we propose an approach to the aggregation of the PCs of a large group of DMs', which seeks to reveal views of similarity and conflict within the group during the pursuit of a group consensus.

As a demonstration and proof-of-concept of these proposed approaches to inconsistency reduction and group aggregation we have developed a web-based decision support tool, within a design that seeks to foster the interactivity of the approaches.

## **1.3 Contributions**

### **1.3.1 Measures of Compromise**

A DM's judgments will undergo alteration during scenarios, such as when looking to reduce the amount of inconsistency within their judgments, or when looking to reach a group aggregation between multiple DMs. A range of metrics are defined to measure the amount of alteration a DM's judgments undergo in such scenarios. These *Measures of Compromise* seek to give a DM semantically meaningful knowledge of the amount of alteration their judgments have undergone. Meaningful measures should aid a DM to more easily comprehend and calibrate the amount of alteration their judgments have undergone and enhance the traceability and validity of such scenarios. Additionally measures that are more meaningful to a DM should enable easier interaction within such scenarios to, for example, set constraints to define thresholds of alteration.

### **1.3.2 Approach to inconsistency reduction**

Inconsistency within a set of judgments can adversely affect the accuracy of a resulting preference vector, hence consideration of its reduction is important. Existing approaches to reduction of inconsistency within a set of judgments are restrictive in terms of the type of inconsistency reduction sought as well as offering little traceability. Therefore a new approach to the reduction of inconsistency within a set of judgments is proposed that offers a more traceable process to inconsistency reduction. In the approach inconsistency and alteration to a DM's judgments are modelled as separate objectives. The type of inconsistency reduction sought is flexible to a DM's preferences. The measures of compromise are used to model alteration to the DM's judgments and reveal to the DM the nature of the trade-offs involved between reducing inconsistency and alteration to their views. To aid a DM to discern the alteration to their judgments the approach seeks solutions that maintain the original judgment representation scheme employed by the DM to input their judgments helping traceability through the process. The approach facilitates setting of constraints upon the objectives for a DM to define thresholds of inconsistency and of alteration.

### **1.3.3 Approach to group aggregation**

Within group decision making the aggregation of the opinions of multiple DMs is an important consideration. The modelling of conflict between objectives is utilised within a proposed approach to the aggregation of a group of DMs' views. Within this approach the alteration to each DM's views is modelled as a separate objective. The approach looks

to create a richer process to group aggregation through the use of the measures of compromise to measure alteration to each DM's judgments. This enhances the traceability and validity of the aggregation process as the amount of alteration each DM undergoes to reach aggregation is revealed through meaningful measures. The approach facilitates additional analysis of global and fairest levels of compromise within the group to further aid the group towards reaching a consensus. Furthermore the approach enables constraints to be set by DMs regarding the amount of compromise they are willing to undergo in the pursuit of reaching a consensus. Moreover the approach allows for DM weights of importance to be incorporated dynamically into the process. The approach can additionally seek to reduce inconsistency during the aggregation process.

### **1.3.4 Approach to large group aggregation**

The proposed approach for aggregating a group of DMs' views is aimed at modest sized groups of less than half a dozen DMs. Scaling issues regarding the approach's performance are identified through investigating the use of the approach for group aggregation of increasingly larger groups of DMs,. Consequently ways to address such issues are explored and an approach for aggregation of a large group of DMs is proposed. The approach facilitates a traceable procedure from the DMs' judgments to a final group aggregation. The approach first utilises clustering to group the DMs into sub-groups based upon the similarity, or agreeability, of their views. The approach additionally enables sensitively analysis to be performed to aid the selection of an appropriate number of sub-groups. Next a single representation of the views of each sub-group's members is derived. As the approach seeks to group similar DMs together, creating a single representation of each sub-group facilitates reduction in the complexity of the problem by looking to identify the redundancy within the views of the DMs. Through the use of the measures of compromise to calculate the single representations the amount of similarity within each sub-group is revealed. The approach then seeks to reach group aggregation with each sub-group modelled as a separate objective.

### **1.3.5 MOODS decision support tool**

Derived from supporting the proposed approaches a Multi-Objective Optimisation Decision Software (MOODS) tool has been developed that can be employed within multiple scenarios. MOODS is an interactive web-based tool that runs in all major browsers utilizing native HTML code with no plugins or downloads required.

MOODS can be utilised by a single DM looking to reduce and understand their inconsistency implementing the proposed approach to inconsistency reduction. MOODS

can additionally be utilised within group decision making via the proposed approach to group aggregation. Furthermore MOODS can be utilised for the aggregation of a large group of DMs and implements the proposed approach to the aggregation of a large number of DMs. The tool's extensible design facilitates additional development and future work to be easily implemented into its framework.

### **1.3.6 Publications**

Publications published during the work include:

E. Abel, L. Mikhailov, and J. Keane, "Reducing Inconsistency in Pairwise Comparisons Using Multi-objective Evolutionary Computing," *IEEE Int. Conf. Syst. Man, Cybern.*, pp. 80–85, 2013.

E. Abel, L. Mikhailov, and J. Keane, "Group aggregation of pairwise comparisons using multi-objective optimization," *Inf. Sci. (Ny)*, vol. 322, pp. 257–275, Nov. 2015.

E. Abel, L. Mikhailov, and J. Keane, "Clustering Decision Makers with respect to similarity of views," in *2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, 2014, pp. 40–47.

## **1.4 Structure**

The structure of the chapters within this thesis are shown in Figure 1.1. This chapter has motivated the work and itemized the contributions. In Chapter 2 decision making procedures and methodologies are outlined followed by discussions of the pairwise comparison technique. Issues of inconsistency and group decision making are then considered.

Chapter 3 proposes measures of compromise that can be used, in semantically meaningful ways, to calibrate and assess the amount of alteration of a set of views and alteration between multiple sets of views.

In Chapter 4 the measures of compromise (from Chapter 3) are used to propose an approach to reduction of inconsistency within a DM's set of judgments. The approach looks to reduce inconsistency for the minimal amount of alteration to a DM's views.

Chapter 5 proposes an approach to the aggregation of a group of DMs' views. The approach uses the measures of compromise (from Chapter 3) to seek aggregation between a group of DMs that minimises the amount of alteration to each DM's views. The

approach reveals the trade-offs involved and facilitates interactive analysis between the DMs to help towards reaching a group aggregation.

The approach to aggregating a group of DMs' views (from Chapter 5) is aimed at modest sized groups and when applied to larger groups of DMs can suffer scaling issues, as identified in Chapter 6. Following this the chapter proposes an approach to the aggregation of a large group of DMs. The approach seeks to cluster DMs into sub-groups based upon the similarity of their views in order to reduce the complexity of the problem.

Chapter 7 presents the MOODS decision support tool. The tool implements the proposed measures of compromise presented in Chapter 3, and the proposed approaches to reducing inconsistency and group aggregation presented in Chapters 4-6.

Finally, in Chapter 8, conclusions of the research are presented along with avenues for future investigations.

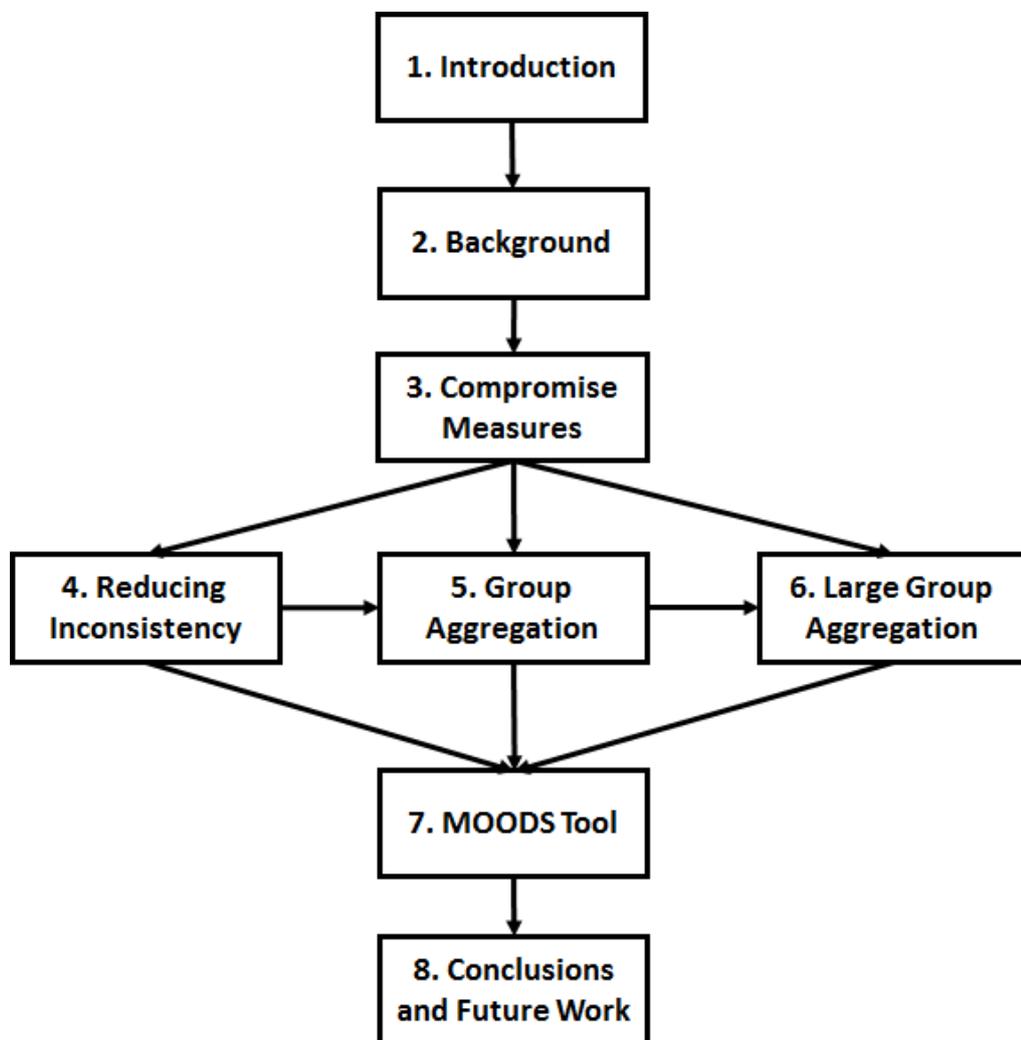


Figure 1.1: Overview of thesis chapters

## Chapter 2 Background

This chapter introduces Multi-Criteria Decision Analysis (MCDA), the significance of pairwise comparison within decision making, discussions of inconsistency and group decision making. It provide background to the technical developments dealt with in the thesis identifying limitations to investigate.

First an overview of MCDA is presented along with discussions of prominent MCDA methods. Next the problem of eliciting preferences using pairwise comparison is discussed and an overview of procedures for deriving a ranking of elements from a set of pairwise comparisons are presented. This is followed by an examination of issues surrounding inconsistency within pairwise comparison. The problem of aggregation of multiple DMs' views from pairwise comparison is then considered.

### 2.1 Multi-Criteria Decision Analysis

In this section after a brief discussion contextualising MCDA within the wider landscape of decision problems, MCDA problem elements, stages and outcomes are discussed. An overview of prominent MCDA methods is then presented.

#### 2.1.1 Decision Problem Dimensions

This work investigates single and group decision making within the field of MCDA. To contextualise MCDA within the wider landscape of decision problems we can consider decision problems with respect to the 3 dimensions: *Criteria*, *DMs* and *Uncertainty* [5].

1. **Criteria:** does the problem under consideration involve a single criterion ( $I$ ) or multiple criteria ( $N$ )? (see Section 2.1.2)
2. **DMs:** is there a single DM ( $I$ ) or are multiple DMs ( $M$ ) involved?
3. **Uncertainty:** is the decision being considered explicitly incorporating uncertainty or not?

Various combinations of these dimensions leads to various Decision Making approaches [5]; these combinations and their commonly associated names are shown in Table 2.1.

A problem with a single DM, a single criterion and no uncertainty, can be considered a *1D (one-dimensional) Optimisation* problem. For example, a single DM looking only to find the cheapest cost of a product.

A problem with many DMs considering a single criterion with no uncertainty, is generally termed *Social Choice*. For example, during election voting a large number of voters are giving their judgment upon who to elect.

A problem with a single DM considering multiple criteria, is termed an *MCDA* problem. For example, a single DM buying a new car considering purchase costs, interior style and safety features as criteria.

A problem with multiple DMs considering multiple criteria is termed a *Group MCDA* problem. For example, a group of managers all considering the choice of a product supplier considering, cost, reputation and delivery time as criteria.

Table 2.1: Decision problem dimensions

	1D Optimisation	Social Choice	MCDA	Group MCDA	Decision Making under uncertainty	Social choice under uncertainty	MCDA under uncertainty	Reality
<b>DMs</b>	1	M	1	M	1	M	1	M
<b>Criteria</b>	1	1	N	N	1	1	N	N
<b>Uncertainty</b>	No	No	No	No	Yes	Yes	Yes	Yes

There are also combinations that consider uncertainty within the decision process. *Decision making under uncertainty* occurs when, for example, a single DM looks to find the cheapest loft conversion considering the uncertainty of estimations of the cost of loft conversion quotes. When many DMs are considering a single criterion under uncertainty we term this *Social choice under uncertainty*. When a single DM is considering multiple criteria and considering uncertainty we term this *MCDA under uncertainty*.

In *Reality* some conjecture that virtually all decisions are in some way the most complex combination of these dimensions [5] – that is, they are group decisions considering multiple uncertain criteria. For example, when looking simply for the cheapest car, we subconsciously consider other criteria, we subconsciously incorporate opinions from friends, mechanics and the media into our decision, and uncertainty is

attached to our cost valuations with respect to the wider world and economy. However, in modelling decision problems, we look to model the complexities of the world as a simplification of reality such that the model is useful [6]. A perfect prediction of tomorrow's weather that will take a week to calculate has little use. Therefore for a decision problem we have to consider the trade-off between model accuracy and complexity. Regarding these dimensions this work explores MCDA and Group MCDA problems.

### 2.1.2 Common MCDA problem elements, stages and outcomes

We can define a general MCDA problem as “*Seeking to determine the suitability of alternatives of a goal with respect to criteria*”.

These are a shared set of elements and notation [7] common to most MCDA methodologies:

1. A single or multiple set of objectives which are considered the **Goal** of the decision:  $G$ .
2. A set of  $m$  **alternatives** that represent the set of possible outcomes to the defined goal  $A_1 \dots A_m$ . As well as actionable outcomes, no action may constitute an alternative outcome [8].
3. A set of  $n$  **criteria** for which the alternatives to the goal are to be evaluated with respect to  $C_1 \dots C_n$ . Criteria may be termed *benefit* criteria or *cost* criteria: for benefit criteria the higher their value the better, for cost criteria the lower their value the better. Additionally, the importance of each criterion with respect to the goal will undoubtedly be different and numerical **criteria weights** can be used to define the significance of each criteria.

There are a set of stages common to many MCDA methodologies [7]. The stages are presented in a chronological order although in reality the decision making process is a dynamic procedure and a DSS should incorporate flexibility accordingly.

1. **Problem Formulation**: this stage involves the conceptualisation and formulation of the overall goal of the decision problem well as the criteria and alternatives.

2. **Data Elicitation:** this stage involves eliciting from the DM (or group of DMs) their qualitative opinions, as well as collating any quantitative data, relating to the various elements of the decision.
3. **Data Evaluation:** the data evaluation stage may involve conversion of some of the data to common scales or evaluating the data for the presence of anomalies and contradictions.
4. **Data Aggregation:** with the data elicited and evaluated it can then be aggregated. In group decision making extra considerations include how differing weighting of importance of DMs will be handled.
5. **Outcomes:** from the aggregation, depending upon the DM's needs, outcome of various granularities can be derived.
6. **Evaluation of results:** the final stage of the process is to determine what course of action is to be taken based upon evaluation of the findings of the decision process. This stage can also involve analysis processes such as sensitivity analysis.

From the outcomes stage, various granularities of outcomes of the alternatives can be attained. What outcome is required needs to be considered, in the context of the problem and the DMs. Roy [9] defined four decision problem formulations within the MCDA context:

1. **Description decisions:** describe a decision's elements and relationships, to extract and present to a DM descriptive information about the decision.
2. **Choice decisions:** concern only selecting a single alternative from the group of alternatives.
3. **Sorting decisions:** (sometimes also referred to as classification decisions), look to sort and categorise the group of alternatives into sub-groups. This may be into unordered sub-groups, such as sorting a group of country economies into agricultural, industrial or maritime, or into preference ordered sub-groups such as sorting a group of country economies into strong, fair and weak.

4. **Ranking decisions:** (sometimes referred to as ordering decisions), look to derive a ranking either full or partial of a decision's alternatives. A full ranking may be ordinal or cardinal:
  - a. **Partial Ranking:** looks to discern some preference ranking between alternatives with respect to other alternatives. Some conclusions may be drawn on preference between alternatives but there will also be alternatives for which preference between is indeterminate.
  - b. **Full Ordinal Ranking:** looks to derive a ranking of the alternatives from best to worst without consideration of the extent of differential between each place in the ranking.
  - c. **Full Cardinal Ranking:** looks to derive a ranking of the alternatives from best to worst with the amount of differential between each ranking position calculated.

There are two additional problem types commonly proposed within the literature. An **Elimination Problem** [10] is a binary variant of a sorting problem with only two classes defined (accepted and eliminated) from which the alternatives are sorted. A **Design Problem** [11] looks to create or identify a goal or action that will satisfy the aspirations of a DM. The aim is to aid a DM with procedures for creating better alternatives.

This work investigates problems in which we look to derive a full cardinal ranking of the alternatives.

### 2.1.3 MCDA methods

Various MCDA methods exist to aid a DM in the evaluation of alternatives with respect to multiple criteria to derive an outcome. Overviews of prominent methods are presented next.

#### 2.1.3.1 *Simple Weighted Methods*

The Weighted Sum Model (WSM) is a simple and commonly used approach. Using each criteria weight and alternative score a cardinal ranking of the alternatives can be derived. WSM model assumes all criteria as benefit criteria and that each criterion's utility increases with its value. The WSM is appropriate only for simple problems involving the same units of measurements for all the criteria [7].

The Weighted Product Model (WPM) is similar to the WSM however multiplication is utilised instead of addition. Like the WSM the WPM assumes all criteria as benefit criteria. The WPM can be altered to support aggregation of criteria of different measurement units, see [7].

#### **2.1.3.2 *Technique for Order Preference by Similarity to an Ideal Solution***

In the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method [12] a complete ranking of a set of alternatives is obtained based upon their distances from the ideal solution and the negative ideal solution. The most preferred alternative (with the largest relative closeness value) is the one that is the closest distance from the ideal solution whilst being the furthest away from the negative ideal solution. First, the data relating to the different alternative outcomes with respect to the criteria is normalised, allowing for comparisons between data from different scales. Next the data is weighted via the criteria weights to calculate criteria weighted alternatives data. TOPSIS does not consider elicitation of criteria weights from a DM. Next, from the weighted normalised dataset, the ideal solution and negative ideal solution are determined. These represent the theoretical best and worst alternatives from all the alternatives data combined (assuming each criterion is a benefit criterion). Next the distance that each alternative is away from both these solutions is determined with respect to their Euclidian distance and the relative closeness value of each alternative is then calculated. From this a cardinal ranking of the alternatives can be created.

#### **2.1.3.3 *Multi-Attribute Utility Theory***

Multi-Attribute Utility Theory (MAUT) looks to model explicitly the utility function that a DM consciously or subconsciously utilises over a range of values of a criterion [13]. MAUT can help to model, with utility functions, how the amount of utility for a criterion changes over the range of values of the criterion. For example, when considering buying a new laptop a DM may consider criteria that include screen size and hard disk space. A utility function's simplest form is linear in nature - that is, the amount of utility increase of a criterion over its range of values will be steady. Such a function would be appropriate if the DM's view with respect to hard disk space were that each increase of space from the lowest to the highest resulted in the same level of increase in utility. A DM's preference of utility increase will invariably not be constant over the range of values. For example, the DM may attach more utility to an increase in screen size between 10in and 12in than the utility increase between 16in and 18in, despite both representing an increase of 2ins. Here higher utility increases may occur between lower values of the criterion. A

utility function can look to model such non-linear relationships. The utility functions of the criteria along with weights of importance of the criteria are then utilised to assess the alternatives and calculate a cardinal ranking of the alternatives. Defining utility functions to capture accurately a DM's views can be challenging and various methods of elicitation have been defined such as UTilities Additives method (UTA) [14], UTA<sup>GMS</sup> [15] and The Generalized Regression with Intensities of Preference (GRIP) [16]. See [17] for a full discussion of MAUT function elicitation methods.

#### **2.1.3.4 Analytic Hierarchy Process**

The Analytic Hierarchy Process (AHP) developed by Saaty [18], makes extensive use of the concept of pairwise comparison (see Section 2.2 ) utilised within a hierarchical framework defining the decision goal, its alternatives and the set of criteria against which the alternatives are to be compared. AHP can be utilised for both single and group decision making problems. AHP is a popular method and has created a large body of literature and applications in highly diverse areas [19]. The AHP procedure can be broken down into 5 broad stages.

1. **Problem definition and Hierarchy construction.** First the goal of the decision problem is defined along with the alternative outcomes and the criteria relating to the decision problem. These elements are then represented within a hierarchical structure where each layer is dependent on the layer above. Criteria may themselves be made up of multiple sub-criteria that are represented 1 layer below them in the hierarchy. For example, the criteria of cost of a car may be composed of 2 sub-criteria of purchase cost and fuel costs. Figure 2.1 shows a visual representation of a decision of selecting a renewable energy source, from which 5 criteria and 3 alternatives have been identified.
2. **DM's preference elicitation.** The elements on a single layer are then compared with respect to the dependencies they share with the layer above them in the hierarchy through pairwise comparisons of elements on the same layer. AHP allows the combination of both tangible and intangible data to be considered simultaneously within the same decision problem.

3. **Aggregation.** Once all information has been elicited, aggregation is performed to derive a cardinal ranking (weights), for the elements at each layer for each dependent element in the higher layer.
4. **Synthesis.** Synthesis of the rankings at each layer of the hierarchy is then performed to calculate a ratio ranking of the decision alternatives of the bottom layer of the hierarchy.
5. **Sensitivity analysis.** Analysis of the final ranking and of the stages to derive the ranking can be performed. Such analysis may include sensitivity analysis to observe how changes to elicited judgments affect the final rankings.

Extensions to AHP have been proposed. For problems considering sets of elements larger than 9, Ishizaka proposed the cluster and pivot method, see [20]. Additionally AHPSort [21] has been proposed as an extension to allow AHP to be used for sorting decision problems.

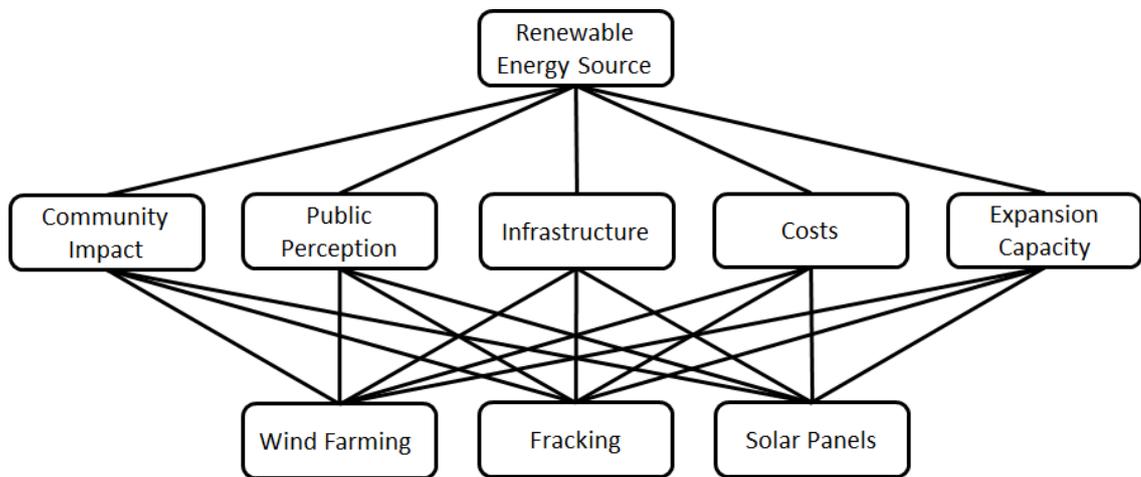


Figure 2.1: AHP hierarchical example structure

To aid the DM in the use of AHP and to counter issues such as rank reversals, see [22], Saaty defined a set of 4 axioms to be followed, see [8]. The fourth axiom states that any layer in the decision problem's hierarchy is independent on the lower layers. Consequently, when interdependencies exist within the layers of the system more complex modelling is required. The Analytic Network Process (ANP) [23] can aid in this more complex dependency modelling. Given a problem to determine the choice of airline for a flight with criteria of 'price' and 'leg room', there is a dependency between these

two criteria which we may wish to consider. ANP can be utilised to model these more complex relationships [23]. Although facilitating richer modelling ANP also introduces issues of increased complexity. The questions that are now asked of the DM to give context to their comparisons can be complex and the number of comparisons required can become unwieldy [24]. Despite its richer modelling prowess ANP is less utilised than AHP. Here we appreciate the trade-off between model complexity and accuracy of the representation of reality.

#### **2.1.3.5 Outranking Methods**

The Preference Ranking Organisation METHod for Enrichment Evaluations (PROMETHEE) [25] is a term used for a family of MCDA methods used to calculate a partial or full ranking of a set of decision alternatives. The approach has no direct consideration of the determination of the importance of each criterion, rather they are assumed to be directly given by the DM. The approach requires the DM to consider two alternatives with respect to the criteria and determine if one alternative outranks the other, is indifferent to the other or is incomparable to the other. This evaluation process, formulated for each criterion, is determined by a preference function. The outranking requirements define the extent to which an alternative should dominate another to be considered to outrank it, see [26] for a comprehensive list of preference functions. From this analysis we can define from each alternative both positive outranking flow and negative outranking flow values [26]. From these values a ranking or partial ranking of the alternatives is achieved depending upon the flavour of PROMETHEE used: PROMETHEE I [25] creates a partial ranking of the alternatives; PROMETHEE II [25] creates a complete ranking of the alternatives; PROMETHEE III [27] ranks alternatives based on intervals; PROMETHEE IV [27] deals with continuous data; and PROMETHEE V [28] identifies a subset of alternatives based upon a set of constraints.

For the ELimination and Choice Expressing Reality (ELECTRE) method [29], as with PROMETHEE there are various flavours which result in either a partial or full ranking of the alternatives [30]. Firstly the data relating to the different alternatives is processed so it can be represented on a common measurement scale. The process then involves comparisons between a pair of alternatives  $a$  and  $b$  with respect to each criterion resulting in one of four outcomes:  $a$  is strictly preferred to  $b$ ;  $b$  is strictly preferred to  $a$ ;  $a$  is indifferent to  $b$ ;  $a$  is incompatible to  $b$ . From this, concordance and discordance values between  $a$  and  $b$  are determined. For an outranking to occur between 2 alternatives  $a$  and  $b$ , a sufficient majority of criteria should affirm the outranking. The concordance

value represents the weighted total of all the criteria of  $a$  that outrank  $b$ . The discordance between a pair of alternatives represents the maximum value from the largest disagreement between the criteria of the pair of alternatives. Once the relations between the alternatives are established then the exploration phase is used to arrive at recommendations and outcomes depending upon the flavour used.

#### **2.1.4 MCDA Conclusions**

MCDA decision problems seek to determine the suitability of the alternatives of the decision goal with respect to the criteria. A key part of prominent MCDA methods such as AHP and ANP is the concept of pairwise comparison. They can also be utilised to enhance stages of MCDA methods such as TOPSIS or PROMETHEE to aid in the elicitation of criteria weights. The next section looks in detail at pairwise comparison; its properties and its abilities to represent a set of DM judgments.

### **2.2 Pairwise Comparison**

Pairwise Comparison (PC) enables the breaking down of a larger decision problem into more manageable smaller chunks. This segmentation of a larger decision problem can be achieved through the use of the Law of Comparative judgment [31]. A PC allows a DM to consider only a pair of elements and to determine their preference, and strength of preference, between the pair, with respect to an intangible factor. Given a set of elements to rank, PC can be used to elicit from a DM their preference and strength of preference for each pair. From this set of PCs a ranking of the elements can then be derived. This ability to take only a pair of elements of a decision at a time and consider just these 2 elements helps to achieve a separation of concerns for the DM and assists them in achieving a more accurate reflection of their judgments [8], [32]. The strength of PC has been shown thorough experimentation with a deterministic example. DMs were asked to use PC to determine the different ratio sizes between a set of 2-dimensional shapes. It was shown that PC aided DMs in making highly accurate estimates of the reality, in this case the true ratio size differences between the shapes, see [8].

Next PC notation is outlined, followed by discussions of various scales that can be utilised to represent the strength of preference of a DM's views. This is followed by an overview of prominent methods to derive a ranking of elements from a set of PCs along with measures for evaluating the rankings produced from these methods.

### 2.2.1 PC Notation, Properties and Graph Visualisation

Given a set of  $n$  elements  $e_1$  to  $e_n$ . The set of PCs, one for each pairing combination of elements in the set, can be collated into a two-dimensional Pairwise Comparison Matrix (PCM) inside which every element is compared with each other along both axis of the matrix as shown in matrix  $M$ , where  $M_{12}$  represents the DM preference between elements  $e_1$ , and  $e_2$ .

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{1n} \\ M_{21} & M_{22} & M_{2n} \\ M_{n1} & M_{n2} & M_{nn} \end{pmatrix} \quad (2.1)$$

Given a comparison  $M_{xy}$ , between elements  $x$  and  $y$  we can denote that a DM prefers element  $x$  to element  $y$  with the notation  $x \succ y$ . Various numerical scales may be utilised to represent the strength of preference and are discussed in Section 2.2.2; the most widely utilised being the Saaty 1-9 Scale [33]. When, for example, element  $x$  is preferred 3 times more than element  $y$ , this can be denoted as  $x \succ y$  with a preference strength of 3. Conversely the reciprocal comparison  $M_{yx}$ , that element  $y$  is 3 times less preferred than element  $x$ , may be denoted as  $y \succ x$  with a preference strength of  $1/3$ . If neither element is preferred over the other then the elements are said to be equally preferred, usually denoted by a 1. An element compared with itself is also said to have equal preference, and again denoted with a 1. Matrix  $M$  requires  $n^2$  comparison judgments to be completed. However the trace of  $M$  will represent the self-comparisons of elements and therefore can be set to equal preference and represented as a 1. Additionally,  $M$  contains redundant information within the reciprocal judgments. The judgments  $M_{12}$  and  $M_{21}$  are multiplicative inversely related, such that if  $e_1$  is preferred to  $e_2$  twice as much then we can deduce that  $e_2$  is preferred half as much as  $e_1$ . Thus given  $M_{12} = x$ , we can infer that  $M_{21} = 1/x$ . Utilising this reciprocal property along with the self-comparison property reduces the number of comparisons needed as well as reducing potential contradictions from occurring. By using these two properties  $M$  can be rewritten as:

$$M = \begin{pmatrix} 1 & M_{12} & M_{1n} \\ 1/M_{12} & 1 & M_{2n} \\ 1/M_{1n} & 1/M_{2n} & 1 \end{pmatrix} \quad (2.2)$$

Employing these properties the number of judgments  $j$  needed to complete a PCM is reduced to:

$$j = \frac{n(n-1)}{2} \quad (2.3)$$

A PCM and its properties can be visualised succinctly via a Directed Acyclic Graph (DAG). Each element is represented as a graph node and each judgment is depicted by a directed arc from the preferred element to the other. Equally preferred elements can be shown via an undirected arc. Each arc is then labelled with the judgment preference strength value. The DAG may contain every PCM cell element value, with self-comparisons depicted as arcs to themselves and with reciprocal judgments. However to aid graph clarity these are usually omitted. A DAG representation of a PCM can help a DM assimilate the overall inclinations of the nature of the PCM more quickly and allow them to more easily identify patterns, such as cycles within the PCM, which might be hidden within the matrix view. Figure 2.2 shows a DAG of a set of elements created from a PCM in [33]. Here the DM is asked to define their preference strength of the wealth of 7 different countries: United States (US); Soviet Union (USSR); China (C); France (FR); United Kingdom (UK); Japan (JP) and Germany (GER). From the DAG representation the US's dominance (in the DM's eyes) is easily visible.

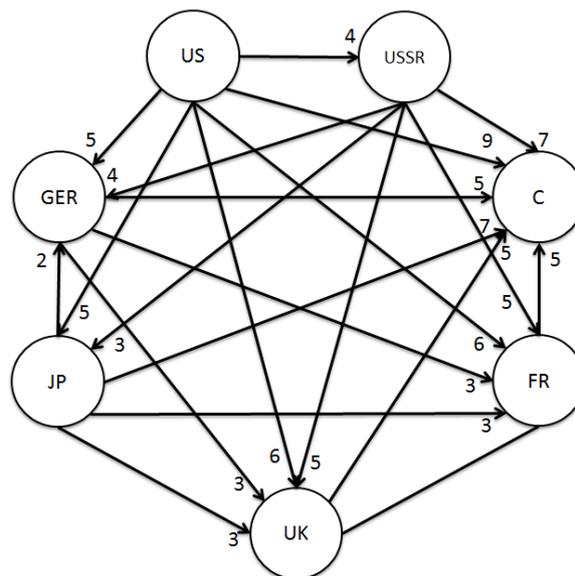


Figure 2.2: Wealth of nations DAG

From a PCM a one-dimensional representation of a DM's judgments – a *Preference Vector* – can then be derived through the use of a *Prioritization Method*. There are many

prioritization methods for this process of reducing a two-dimensional matrix into a one-dimensional ranking, see Section 2.2.3. When a PCM is perfectly consistent any prioritization method can be utilised to derive the true preference vector of the set of judgments. However when inconsistency is present within the PCM any preference vector derived will only be an estimate of the information of the set of judgments. Inconsistency within a PCM, its identification and affects are discussed in Section 2.3 . In this section the 1-9 scale has been utilised, however many different judgment scale functions exist and they are discussed next.

### 2.2.2 Judgment Scale Functions

A key consideration of PC is the scale to use to denote the strength of preference of each of the DM’s judgments. The scale that is utilised can have implications upon both the PCM’s ability to accurately represent the DM’s judgments as well as implications regarding the consistency within the set of judgments. This work focuses on ratio-based scales. The strengths of preference of a DM’s judgments can be formulated by the use of a verbal scale - whose points are expressed as words helping to give meaningful context to the DM’s qualitative judgments. The question can be posed to the DM, given a pair of elements, compared with respect to some higher element, which element has preference, and through the verbal scale by how much. Using a verbal scale is “intuitively appealing, user friendly and more common in our everyday lives than numbers” [34]. These verbal judgments can then be mapped onto a numerical scale whose numeric values become the judgment strength values within a PCM. Table 2.2 shows an example of this mapping and definitions of both the verbal scale and 1-9 numerical scale we saw earlier.

Table 2.2: Saaty 1-9 scale mapping from verbal scale

<b>Verbal Preference Strength</b>	<b>Numerical</b>
Equal importance	1
Weak or slight	2
Moderate Importance	3
Moderate plus	4
Strong Importance	5
String plus	6
Very strong or demonstrated importance	7
Very very strong	8
Extreme importance	9

Although the most popular scale, the 1-9 scale, has been questioned by some of its ability to accurately represent a DM's preferences and its ability to create PCMs containing a set of consistent judgments [35]. Consequently many other scales have been proposed that can be mapped from the same 9 point verbal scale. In the Power scale [36], each verbal scale point is raised to a common power  $a$ ; when  $a=2$  this function results in a range of numerical values between 1 and 81. The Geometric scale [37], uses geometric powers for a mapping of the verbal scale with respect to a common numeric base  $a$ ; when  $a=2$  the scale's mapping range from 1 to 512 with the interval between the verbal points rapidly increasing. The Ma-Zheng scale [38], (sometimes referred to as the Inverse scale [34]) results in a set of numeric scale values from 1-9 like the linear scale. However there is a smaller spread between the values in the first two thirds of the verbal points and a larger spread of the numeric values between the last third of the verbal point's values. The Salo Hamalainen scale [39], (sometimes referred to as the balanced scale [34]) multiplies the verbal points by a small constant  $e$ , where they suggest a value of  $e$  as  $1/20$  or  $1/17$ ; when  $e=1/20$ , the mapping function creates a set of numeric values that range between 1 and 9. The spread of values for the lower verbal values are more closely clustered than the 1-9 scale but larger than for the Ma-Zeheng scale. The Logarithmic scale [34] creates numerical values dictated by the base value  $a$  of the log function; when  $a=2$  the scale's mapping values range from 1 to 3 and  $1/3$ ; when  $a=10$  the mapping results in a range of values from 0.3 to 1. The Root scale [36], takes the root to a parameter value  $a$  of the verbal scale values; when  $a=2$  a set of numeric values from 1 to 3 is created from the mapping function. Figure 2.3 shows graphically the mapping functions of the 9 point verbal scale onto the scale functions<sup>1</sup>.

The different judgment scales seek to accurately represent a DM's views whilst considering the trade-off being complexity and ease of use. A DM could make a choice of scale most suited to their disposition and the decision problem at hand. In this work the 1-9 scale is utilised within the proposed approaches due to its overwhelming prominence. However the approaches proposed are independent of a specific scale and could be extended to use any of the above scales.

---

<sup>1</sup> Without the power, geometric and Salo Hamalainen  $e=1/17$  mappings, whose final values rise so steeply they hinder clarity of the other mappings.

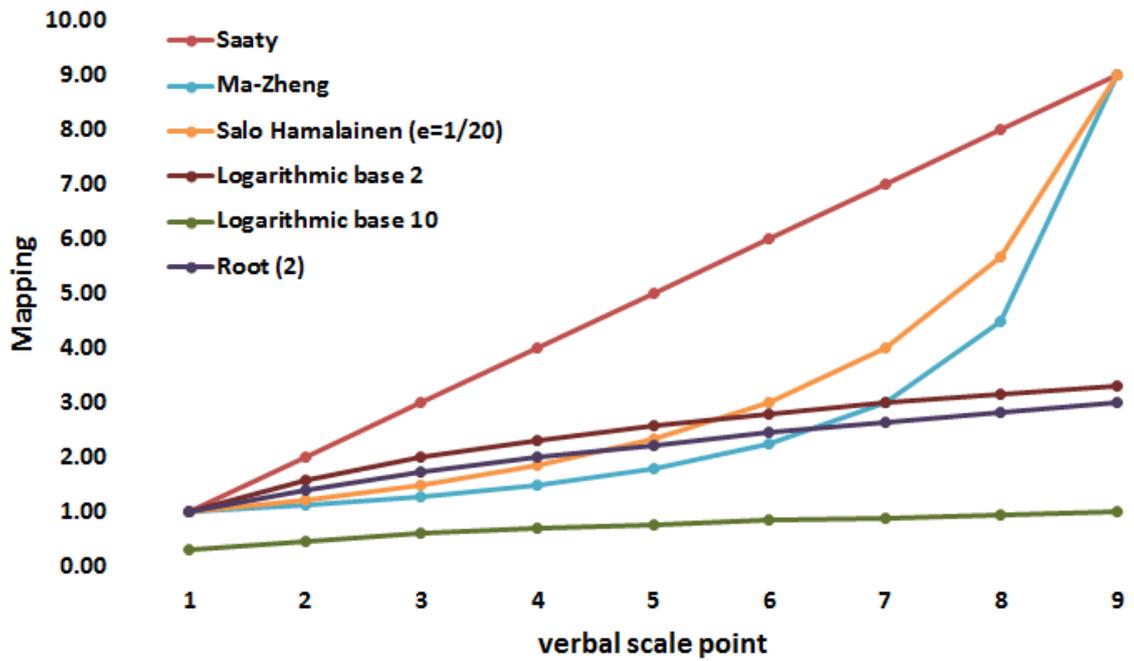


Figure 2.3: Verbal scale function mapping onto numerical scales

### 2.2.3 Prioritization Methods

When a PCM is perfectly consistent any prioritization method will derive the true preference vector of the set of judgments [4]. However when inconsistency is present within the PCM any preference vector derived will only be an estimate of the information of the judgments. Inconsistency within a PCM, its identification and affects are discussed in Section 2.3 . When inconsistency is present different prioritization methods may then derive different estimates. The larger the amount of inconsistency, the greater the possible discrepancy between prioritization methods. Inconsistency within a PCM of more than a handful of elements is almost inevitable [4] therefore various prioritization methods and means of evaluation between their resulting preference vector is an important area of consideration. A brief overview of prominent prioritization methods is now given, see [4] for fuller discussions and comparisons, and see [40] for further comparisons of methods and investigations of similarities between methods.

For a completed set of  $n$  elements utilised to create a PCM  $M$  there exists a preference vector  $w = \{w_1, w_2, \dots, w_n\}^T$ , where  $w_i$  represents the ranking weighting of the element  $i$  within the matrix for  $i = 1$  to  $n$ . If the DM is perfectly consistent then the relationship between each element in  $M$  and pair of corresponding weights is such that  $M_{ij} = w_i/w_j$ . We can use this property to express  $M$  as:

$$M = \begin{pmatrix} w_1/w_1 & w_1/w_2 & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_n \\ w_n/w_1 & w_n/w_2 & w_n/w_n \end{pmatrix} \quad (2.4)$$

When a PCM is perfectly consistent it will be of rank one and a true preference vector can be derived by taking the average of the elements in any column of the matrix. However, when inconsistency is present in a PCM then any prioritization method will only attain an estimate of the preference vector  $w$ . Therefore different prioritization methods may produce different preference vector results.

### 2.2.3.1 *Prioritization Methods Overview*

The most straightforward prioritization method is the Additive Normalisation (AN) method, where first vertical normalisation of the matrix is performed by dividing each element in a column by the sum of the column; then the mean of each row is taken to calculate the weight of each element [4]. Although widely used due to its ease of application it has been considered inferior due to its simplicity [41]. Slight variants of AN exist, see [41].

The Geometric Mean (GM) method finds the preference vector weights via the product of each row raised to the inverse power of  $n$  [42]. These weights are then usually normalized to sum to 1. The geometric mean is sometimes considered more appropriate than the arithmetic mean as outliers have less effect upon the resulting preference vector [42].

The Enumerating All Spanning Trees (EAST) method [43] utilises indirect judgments to derive a preference vector from either a complete or incomplete PCM. EAST is considered a more complete form of the AN method taking into account more information to calculate its preference vector [43]. The method centres on finding all the spanning trees [44] from the DAG representation of a PCM. From the spanning trees a set of preference vectors can be derived and a final preference vector derived from their average. As the size of a PCM increases the number of spanning trees present quickly increases, consequently as  $n$  gets large the processing time of EAST becomes a practical limitation.

The principal Eigenvector method (EV) was proposed by Saaty [33], for use as the prioritization method within AHP. The EV method essentially looks to represent the two-

dimensional PCM data set as a one-dimensional vector. The preference vector is derived from the eigenvector of the largest eigenvalue of the PCM. This is then usually normalised to sum to 1.

There are a group of prioritization methods referred to as optimisation methods. They look to optimise the value of an objective function to determine a preference vector under a set of constraints. The Direct Least Squares (DLS) method [45] looks to minimise the sum of the differences between the matrix and the derived values. It utilises the  $A_{ij} = w_i/w_j$  relationship between the matrix and the possible values of the preference vector to find a set of weights which matches this relationship the closest. DLS is a hard objective to solve due to not having a closed form and it may not have a unique solution [46]. Hence further to this a closed form of the objective function titled the Weighted Least Squares (WLS) method was proposed [45], which has been shown to have a unique solution, see [47].

The Logarithm Least Squares (LLS) [42] method is a logarithmic variation upon the WLS approach. This again has been shown to have a unique solution [42]. It was shown that this solution is the equivalent of the GM method solution.

The Logarithmic Least Absolute Value (LLAV) method, proposed in [48], is a variation of the LLS method where the absolute values of each comparison are considered.

The Fuzzy Programming (FP) method proposed by [49] utilises fuzzy logic to solve the prioritization preference vector problem. In the FP approach, each judgment is represented as a fuzzy hyper-line and the method looks to find an approximate point of intersection of these fuzzy lines, see [49].

The Two-objective Optimisation Prioritisation (TOP) method proposed in [50] seeks to simultaneously minimise two objectives, one of the total deviation and one of the number of violations. For definitions of these objectives see Section 2.2.3.2. Due to the conflicting nature of these objectives, a set of trade-off preference vectors will be derived. From this set of solutions no solution is considered superior and the DM can choose one based upon their preferences as a compromise between the objectives.

The Prioritization with Indirect judgments (PrInT) method [51], seeks to simultaneously optimise three objective functions. As well as objectives looking to minimise total deviation and the number of violations a third objective looks to minimise a measure based on the indirect judgment information within a PCM. The Total Indirect Deviation objective looks to minimise the total distance between the PCM's indirect

judgments and the preference vector's weight ratios, see [51]. Like TOP this method results in a set of trade-off preference vectors being derived, the DM can then choose one from this (large) set based upon their preferences.

Various other methods have been proposed. Bryson proposed a method using goal programming (GP) to find a preference vector [52]. This approach has the benefit of nullification of a single outlier. Lin proposed the Enhanced goal programming approach [53] to find a preference vector, that looks to combine the benefits of the GP and LLS methods. A method proposed using chi-square distance as an objective function to find the preference vector was proposed by Jensen [54], using the chi-distance measure to minimise distance between judgments and a preference vector. Further to this Zu proposed a generalized chi-square approach, a generalized form of chi-square minimization [55]. An approach based upon linear regression proposed by Laininen and Hämäläinen [56] performs similarly to the LLS method. A method proposed by Ramanathan derives a preference vector through Data Envelopment Analysis (DEA) [57].

### ***2.2.3.2 Evaluation of Preference Vectors***

Given that inconsistency within a PCM is almost inevitable [4] and consequently that different prioritization methods may produce different preference vectors, evaluation measures have been proposed to appraise and compare different preference vectors.

Total Deviation (TD) is a measure of the total distance between the original PCM and a derived preference vector's weight ratios. Additionally dividing by the number of elements in the matrix, allows TD measurements between matrixes of different dimensions to be compared [58]. Another variant uses only the judgments from the top triangle of the PCM and without a final square root operation [50], [59].

Number of Violations (NV), proposed in [60] and utilised for PC in [46], is a measure of the amount of ordinal rank preservation of the judgments in a PCM that is captured within a preference vector. Given a PCM judgment between elements  $x$  and  $y$  where  $x \succ y$ : if in the resulting preference vector it is the case that the weight of  $x$  is less than the weight of  $y$  then a violation has occurred. This was extended in [46] to include the concept of half violations to consider cases of preference equivalence.

Mean Absolute Deviation (MAD) is a measure of how close a derived preference vector matches the reality of the elements under consideration. The MAD is defined as the average distance between each element of the preference vector and the element in reality. This evaluation measure can only be employed if the true preference vector of the elements is known.

Conformity (C) is a measure of how close a preference vector conforms to the average of a set of preference vectors derived from other prioritization methods. It can be useful to evaluate a new method and was utilised in [49] to show a new method's conformity with existing methods.

### **2.2.3.3 Discussion of Prioritization Methods**

There are a multiple of methods to derive a preference vector from a set of judgments. Although certain methods may be more suited to certain scenarios there is no consensus upon the most suitable method to use. Various claims about methods have been proposed, for example, Saaty suggests [61] that that EV is more appropriate than the LLS method. Conversely it has been shown that the EV method suffers from right-left inconsistency, which leads to rank reversals after an inversion of the scale, originally discovered in [62], see [17] for a worked example. It has been shown that this issue does not occur when using the geometric mean [42]. Therefore approaches that are independent of a specific prioritization method would be more flexible to different scenarios and DM preferences.

### **2.2.4 Pairwise Comparison Conclusions**

Pairwise comparison is an effective method for eliciting views from a DM through facilitating a separation of concerns. Various scales can be utilised in defining the strength of preference of each judgment from a DM. There are many prioritization methods that can be utilised to derive a preference vector ranking from a set of judgments. Inconsistency within a set of judgments affects the accuracy of any preference vector derived, therefore if we can look to identify and reduce inconsistency we can look to derive more accurate preference vectors. Identification and measurement of inconsistency within PCMs and approaches to tackle it are discussed next.

## **2.3 Inconsistency**

Within PC the consistency of a PCM is the extent to which its set of judgments are coherent. PC facilitates a separation of concerns to aid in breaking down a complex problem into a set of smaller chunks and from a set of PC judgments the extra redundant information present makes for a richer level of information. However the amalgamation of the smaller chunks into a PCM may result in inconsistency being present in the set of judgments as a whole. When inconsistency is present in a PCM any preference vector derived will only be an estimate of the judgment's information. Consequently, different prioritization methods may derive different preference vector estimates. Inconsistency within a PCM of more than a handful of elements is almost inevitable [4] and therefore

needs to be considered. The greater the amount of inconsistency present, the more a derived preference vector only represents an estimate of the PCM's judgment information. Approximations of highly inconsistent PCMs produce large errors, hence "approximations from such matrixes make little practical sense" [63] therefore inconsistency within a set of judgments is an issue that needs to be tackled.

There are several reasons why inconsistency may occur within a set of judgments. Firstly if the reciprocal property within a PCM is not upheld then inconsistency may occur. For example, if a DM defines a judgment that  $x \succ y$  yet also defines a judgment that  $y \succ x$ , then the reciprocal property has been breached. Such occurrences have been referred to as Unusual and False Observations (UFO) [64]. Such inconsistency can be avoided by eliciting only one of such pairs of judgment from the DM and inferring the second judgment. Inconsistency may also be present due to insufficient complexity of the modelling process to represent a DM's views [65]. The 1-9 scale facilitates streamline elicitation of a DM's judgments however it may be insufficient to represent extreme views. For example, if a DM defines that  $y \succ z$  with a strength of 9 and that  $x \succ y$  twice as much, then it follows that  $x$  should be preferred over  $z$  by a value greater than 9, which cannot be represented by the 1-9 scale. Inconsistency may also be present due to physiological reasons, such as incomplete information. For example, given a set of elements for comparison, over the course of the comparisons the DM's knowledge of the elements might evolve resulting in their opinion towards elements subtly changing.

Inconsistency of a PCM can be categorised as ordinal or cardinal in nature, these are discussed next.

### **2.3.1 Ordinal and Cardinal Inconsistency**

Inconsistency within a set of PC judgments may be categorized as either ordinal or cardinal, both are important considerations for a DM. Ordinal inconsistency identifies inconsistent information without the strengths of preference of the DM's judgments being considered. For example, given a set of 3 elements,  $x$ ,  $y$  and  $z$ : if  $x \succ y$ ,  $y \succ z$  and  $z \succ x$ , then the judgments are intransitive and contradictory, and ordinal inconsistency is present. The DAG of these judgments is shown in Figure 2.4: Left. From this we see that a cycle is present between these judgments, in this case a 3-way cycle (a cycle between 4 elements could be termed a 4-way cycle and so on). In this example each judgment had a preferred element, ordinal inconsistency can also be present within a set of judgments containing equal preference judgments. For example, if  $x$  and  $y$  are equally preferred ( $x \sim y$ ) then for the set of judgments to be ordinally consistent the remaining judgments must

be:  $x \succ z$ , and  $y \succ z$ , or  $x \prec z, y \prec z$ , or  $x \sim z, y \sim z$ . When ordinal inconsistency is present in a set of judgments then any preference vector derived from them will always contain ranking violations (NV), see Section 2.2.3.2.

Cardinal inconsistency identifies inconsistency between a set of judgments taking into account the strength of preference of each judgment. For a set of judgments to be cardinally consistent then each judgment  $j$  should maintain transitivity - that is, the relation between a first element and a second and between a second element and a third should hold between the first and third, therefore  $j_{xz} = j_{xy} * j_{yz}$  for all  $x, y, z$ . For example, considering a set of 3 elements  $x, y$  and  $z$ : if  $x \succ y$  with a preference strength of  $a$  and  $y \succ z$  with a preference strength of  $b$ , then, for the judgment set to be cardinally consistent, the final judgment between elements  $x$  and  $z$  would need to be such that  $x \succ z$  with a preference strength of  $a*b$ . The DAG of this judgment set is shown in Figure 2.4: Right. When ordinal inconsistency is present in a set of judgments then cardinal inconsistency will also be present, but not vice versa; furthermore a cardinally consistent set of judgments will also be ordinally consistent.

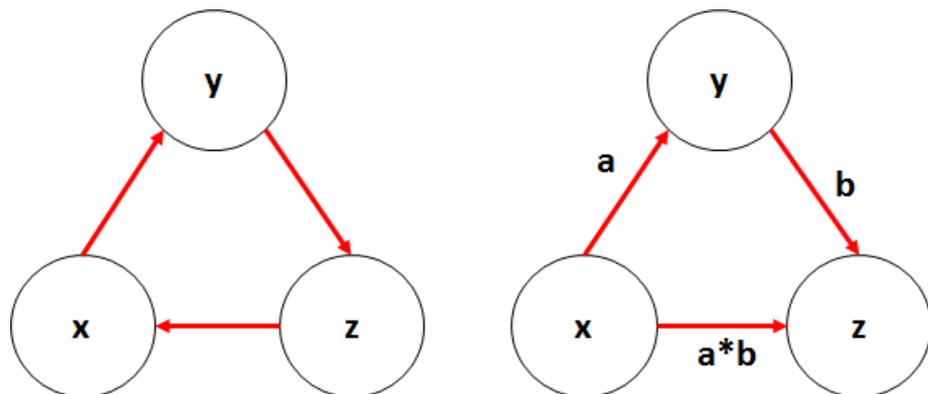


Figure 2.4: Left: Ordinal inconsistency. Right: Cardinal consistency

Next we discuss various measures that have been defined to quantify the level of inconsistency within a set of judgments, first ordinal measures, then cardinal measures. To aid these discussion we define an example set of judgments of 5 elements shown in Table 2.3, and shown as a DAG in Figure 2.5.

Table 2.3: Inconsistency Measures Example

	A	B	C	D	E
A	1	2	3	1/3	4
B	1/2	1	1	1	3
C	1/3	1	1	4	2
D	3	1	1/4	1	3
E	1/4	1/3	1/2	1/3	1

### 2.3.2 Ordinal Inconsistency Measures

A measure of ordinal inconsistency was proposed by Gass [66], in which the problem is formulated as a tournament ranking (with 0 and 1 utilised to represent judgments as losses and wins respectively). For an  $n$  element problem the approach determines the total number of three way cycles ( $c$ ) utilizing the number of wins of each element  $s_i$ .

$$c = \frac{n(n-1)(n-2)}{6} - \frac{\sum_{i=1}^n s_i(s_i-1)}{2} \quad (2.5)$$

From this we can determine the cycles present in the judgments however, the approach does not consider preference equivalence. Alternatively ordinal inconsistency can be measured via Kendall's Coefficient of Consistence ( $\zeta$ ) [67]. This measure looks to determine the number of 3-way cycles present ( $KL$ ) within a set of judgments in relation to the maximum number possible of 3-way-cycles ( $KL_{max}$ ). Given a set of  $n$  elements  $L_{max} = \binom{n^3 - 4n}{24}$  when  $n$  is even, and  $L_{max} = \binom{n^3 - n}{24}$  when  $n$  is odd [67].

$$\zeta = 1 - \frac{KL}{KL_{max}} \quad (2.6)$$

Consideration of the maximum possible number of cycles makes the measure independent of the size of the PCM, and allows comparison between sets of judgments of different number of elements. When  $\zeta = 1$  the judgments contain no 3-way cycles.

Kendall's measure was utilised by Iida [68] to determine if a DM is sufficiently ordinally consistent enough in their judgments, see [68]. Kendall's Coefficient of Consistence is calculated under the assumption that the PCM contains no preference

equivalence judgments so again has no consideration of ordinal inconsistency present through equal preference judgments. Looking to overcome this limitation, the Kendall measure been extended in [69] to look to include preference equivalence within the calculation.

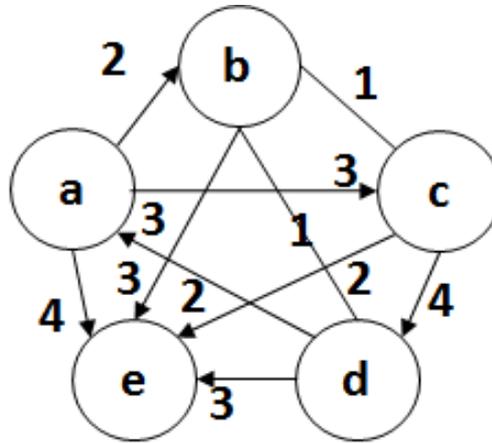


Figure 2.5: Inconsistency Measures Example DAG

An algorithm to determine if a set of judgments contains any 3-way cycles, including consideration of equal preference judgments was proposed in [70]. Their algorithm to determine the presence of any 3-way cycles can be determined via [70], is shown in Algorithm 2.1.

---

**Algorithm 2.1: determining 3-way cycles including equal preference consideration**

---

```

FOR all (i,j,k) from 1 to n WHERE (i ≠ j ≠ k ≠ i )
  IF log(aij) log(aij) ≤ 0 AND log(aik) log(ajk) ≤ 0 THEN
    Cycle Present!
  ELSE IF log(aij) = 0 AND log(aik) = 0 AND log(ajk) ≠ 0 THEN
    Cycle Present!
  ELSE
    No cycle present
  END IF
END FOR

```

---

This allows calculation of whether a DM's judgments contain cycles. It can be extended to determine the total number of 3-way cycles present, through a counter that is incremented with each cycle found. With this we have a measure of ordinal inconsistency that also considers equal preference judgments. Additionally we can record the elements involved in each ordinal cycle. Using this as an example, we can identify that there are two cycles, between:

1. {a, c, d}  $a \rightarrow c \rightarrow d \rightarrow a$
2. {b, c, d} equal preference between b and c, and between b and d, yet  $c \rightarrow d$ .

We refer hereafter to this measure that considers the number of cycles including consideration of equal preference cycles as **L**. When considering cycles within a set of judgments we only need to consider cycles of 3 elements as it has been shown that eliminating all 3-way cycles ensures elimination of cycles of higher orders [71].

### 2.3.3 Cardinal Inconsistency Measures

Various measures have been proposed to measure the amount of cardinal inconsistency present within a set of judgments. By far the most prominent is Consistency Ratio (CR) proposed by Saaty [33]. First the eigenvalue of the largest eigenvector of the PCM ( $\lambda_{max}$ ) is calculated. When an order  $n$  PCM is perfectly consistent then  $\lambda_{max} = n$ . Next, the Inconsistency Index (CI) of the PCM is determined.

$$CI = (\lambda_{max} - n) / (n - 1) \quad (2.7)$$

The CR is then found by dividing the CI by the Random Consistency Index (RI) for the order of the PCM. The RI values represent the average inconsistency found over 50,000 trials of randomly generated matrixes for each PCM order, see [18]. It has been argued that the simulation of 50,000 tests to determine RI values for each value of  $n$  was insufficient to obtain a fair reflection of average values. Other researchers have done similar simulations with much higher numbers of trials [72], although the results were more accurate they were similar to Saaty's original findings.

$$CR = CI / RI \quad (2.8)$$

The lower the CR value, the lower the amount of cardinal inconsistency present in the PCM. Saaty further proposed an acceptability threshold value of a PCM's CR value [18]. The threshold is designed to be an indicator as to whether a PCM is consistent enough for a satisfactory preference vector estimate to be derived. Using this threshold, when a PCM has a CR value of 0.1 or less, it is considered to be acceptable. For our example from Table 2.3,  $CR=0.27$  thus this is considered to have unacceptable levels of inconsistency

present. It has been argued that the choice of the 0.1 threshold to determine an acceptable level of inconsistency is arbitrary and not based upon solid foundations [73]. Therefore giving a DM control over such a threshold is likely to be beneficial.

Other measures of cardinal inconsistency have been proposed based upon the transitive properties of a set of judgments. Consistency Measure (CM) proposed in [73], is a more fine-grained alternative of the CR measure that considers the inconsistency between each triple of judgments. Considering each possible sets of 3 judgments at a time CM determines the inconsistency of a triple via:

$$CM_{ijk} = \min\left(\frac{|a_{ij} - a_{ik}a_{kj}|}{a_{ij}}, \frac{|a_{ij} - a_{ik}a_{kj}|}{a_{ik}a_{kj}}\right) \quad (2.9)$$

From our example the judgments between elements  $a, c, d$   $CM_{ace} = \min(0.97, 35) = 0.97$ , are the most inconsistent triple. CM gives not just a measure of inconsistency but also identification of where the highest levels of inconsistency within the set of judgments occurs.

Measures of cardinal inconsistency have been proposed based upon calculations of the distances between a set of judgments and a derived preference vector from the judgments. For example, Aguaron & Moreno-Jimenez proposed Geometric consistency Index (GCI) [72] an inconsistency measure based upon the distance measurements between the preference vector derived using the GM prioritization method and the original judgments. GCI is calculated via:

$$GCI = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j>i}^n (\log a_{ij} - \log(\frac{w_i}{w_j}))^2 \quad (2.10)$$

Where  $w_i$  is the ranking value for element  $i$  in the preference vector. Comparison of GCI and CR was shown to have an almost linear relationship. A threshold of acceptability of GCI has been proposed [72], when  $n=3$   $GCI \leq 0.31$ ,  $n=4$   $GCI \leq 0.35$ , when  $n>4$   $GCI \leq 0.37$ . Other distance -based measures making use of a derived preference vector have been proposed, Chu et al. [45], whilst proposing the WLS, proposed a consistency measure using the mean square error. Crawford & Williams [74] proposed a distance measure when proposing the GM prioritization method. A downside of such distance based measures is that they require a preference vector to be derived.

Peláez and Lamata proposed a cardinal inconsistency measure based on determinant of the matrix termed the Consistency Index (CI) [75]<sup>2</sup>. For a three-element matrix the determinant of a judgment triple will be 0 when perfectly consistent:

$$\det(ijk) = \frac{a_{ik}}{a_{ij}a_{jk}} + \frac{a_{ij}a_{jk}}{a_{ik}} - 2 \quad (2.11)$$

The CI of an  $n \times n$  PCM is then calculated by the average of the CI of the matrix of each possible sets of three judgments. Furthermore Ji and Jian also proposed a consistency measure based on the transitivity rule, see [76].

We see there are various measures both of ordinal and cardinal inconsistency that seek to give a quantifiable measure of inconsistency. For our example we calculate values from some of these measures as shown in Table 2.4. Next we discuss how we can utilise such measures to tackle inconsistency within a set of judgments.

Table 2.4: Inconsistency Example Measures values

Measure	Value
L*	1
L	2
CR	0.27
CM	0.97
GCI	4.75

\*number of 3-way cycles with no consideration of equal preference judgments

### 2.3.4 Reducing Inconsistency

Inconsistency present within a set of judgments has adverse effects upon any preference vector derived from them. Therefore if we can look to reduce the amount of inconsistency before deriving a preference vector then we can look to diminish its adverse effects. Once inconsistency has been identified, there are various ways it may be tackled:

1. Getting the DM to review their judgments;
2. Automatically altering the judgments in some way;
3. Proceeding but attempting to take the inconsistency knowledge into consideration

---

<sup>2</sup> Not to be confused with the Random Consistency Index (also CI) from CR calculation.

In Chapter 4 of this work we propose a new approach based upon the second of these, therefore after a brief overview of the other strategies a discussion of previous approaches to automatically altering judgments to reduce inconsistency is presented.

Getting a DM to review and alter their judgments seeks to reduce the amount of inconsistency manually. Knowledge of the inconsistency present can be utilised to help guide a DM in altering their judgments. Harker proposed an approach to identify the judgment whose adjustment results in the largest reduction in a PCM's inconsistency [77]. Satty proposed approaches in [78] to aid a DM in selecting a single judgment to alter, to facilitate the most reduction in inconsistency with a single judgment change. Another approach to detect the single most inconsistent judgment in a PCM has been proposed in [79]. Here the most inconsistent judgment is determined with respect to cardinal inconsistency. Through utilizing the redundant information present through indirect judgments the measures of Congruence and Dissonance proposed in [80] help to identify to a DM the judgment that is the most inconsistent, both ordinally and cardinally respectively.

When proceeding without alteration to the judgments, analysis of inconsistency within the set of judgments can aid selection of the most appropriate prioritization method. Different prioritization methods have different procedures and as such are affected differently by inconsistency. For example, the WLS method may be a less appropriate method for a PCM with a single large outlying inconsistent value, and the LLAV method may be an appropriate method when a large variety in the range of inconsistent deviations is present within a PCM.

### **2.3.5 Previous approaches to automated inconsistency reduction**

Using measures of inconsistency we can look to quantify the amount of inconsistency present in a set of judgments. Then through alteration of the judgments we can look to reduce the amount of inconsistency, and determine updated amounts of inconsistency now present. Generally previous approaches to automatically alter the judgments in a PCM seeking to reduce inconsistency focus upon either ordinal or cardinal inconsistency not both. Additionally when alteration to judgments is considered, it is only considered through constraints or as part of a combined single objective. Moreover little attempt is made to make the alteration semantically meaningful to the DM. Furthermore, when seeking to reduce inconsistency to a threshold value, they offer no control for the DM to define the threshold value.

A convergence algorithm approach has been proposed in [81] which looks to find an altered PCM that has a cardinal inconsistency (CR) measure below a threshold ( $CR < 0.1$ ). The approach can alternatively be applied iteratively to reduce CR to 0. The algorithm looks to find a cardinally-consistent altered PCM as a single objective whilst seeking to ensure the amount of departure from the original judgments is below given ranges (via hard constraints). An example of an original PCM and altered PCM taken from [81] are shown in Table 2.5. The values of the altered PCM are composed of judgment values that fall outside of the original judgment scale (here 1-9), therefore are difficult for a DM to comprehend how their judgments have changed. Additionally the constraints used to measure departure from the original judgments are difficult for a DM to semantically comprehend and relate to how their judgments have changed, which hinders auditability of the process. Furthermore as alteration is used only as a constraint there is no explicit consideration of looking to minimum the amount of alteration.

Table 2.5: Example Measures values [81]

Original Judgments				Altered Judgments			
<b>CR = 0.213</b>				<b>CR=0.098</b>			
	E1	E2	E3		E1	E2	E3
E1	1	7	1/5	E1	1	5.984	0.234
E2	1/7	1	1/8	E2	0.167	1	0.107
E3	5	8	1	E3	4.274	9.358	1

A similar convergence algorithm approach was proposed in [82]. Again only cardinal inconsistency (CR) is considered with the aim to find a solution below a threshold ( $CR < 0.1$ ). The values of altered PCMs found are composed of judgment values that fall outside of the original scale utilised so again hinder comprehension. Alteration is considered (through similar calculations as defined in [81]) again as hard constraints by which to determine if the found altered PCMs are feasible. Alteration constraints are considered with little attempt to be meaningful for a DM to relate to and to comprehend with respect to how their judgments have changed.

An approach that focuses on reducing ordinal inconsistency is proposed in [83]. This approach seeks to reduce the number of 3-way cycles within a PCM via an iterative process of judgment reversals. At each iteration it seeks to reverse a judgment that will result in the maximum reduction of 3-way cycles to converge to a solution PCM without

any 3-way cycles. On each iteration the approach focuses on identifying the judgment which will have the most impact upon ordinal inconsistency. Through seeking to remove ordinal inconsistency optimally fewer iterations will be required and thus fewer reversals required to reach a set of fully consistent judgments. If multiple judgments represent the maximum reduction of 3-way cycles then cardinal inconsistency is considered as a tie-breaker to determine which judgment is reversed. Here the cardinal inconsistency of a judgment is measured via the amount of discrepancy between the judgments strength and the measurement of indirect judgment strength, see. [83].

Inconsistency reduction has also been addressed via the approach in [84] implemented utilizing genetic algorithms. Only cardinal inconsistency (CR) is considered as a single objective to look to find solutions for which cardinal inconsistency is below a threshold, again  $CR < 0.1$ . The amount of alteration between found solutions and the original judgments is not explicitly considered. Additionally solutions are modelled in such a way that the reciprocal property of the PCM is not always maintained, therefore the condition that  $a_{ij} = 1/a_{ji}$  is not always maintained in found solutions, which may introduce additional inconsistency into the judgments. The amount the reciprocal property is violated by is defined via a user-settable tolerance parameter.

Similarly a genetic algorithm is utilised in [85] to reduce the inconsistency of a PCM; here the PCM and the altered PCM are represented as fuzzy numbers. This approach only considers cardinal inconsistency looking to find a solution with a lower CR value. During evaluation of solutions during optimisation, the CR of individuals and the alteration of the amount of change are considered as a single objective. Individuals with feasible CR are assigned a high evaluation value. Alteration is considered to then rank the remaining solutions with CR 0.1 or higher.

Similarly an approach defined in [86] looks to find an altered solution with reduced inconsistency considering only cardinal inconsistency. The approach looks to find altered solutions with the lowest value of CR measure. The approach models the problem as a non-linear programming model and a genetic algorithm is utilised to solve it. During the operation of the genetic algorithms individuals are evaluated via the level of cardinal inconsistency and consideration of alteration to the PCM considered via their combination into a single objective function. Where the level of CR is used to measure cardinal inconsistency and similarity between the initial PCM and the solution set is measured as a log-based calculation being the amount of distance change between the

two judgments sets [86]. The measure of alteration utilised is difficult for a DM to semantically interpret with respect to the alteration of their judgments.

A summary of the functionally and considerations of these approaches is shown in Table 2.6. We summarize that no approach appears to consider both ordinal and cardinal inconsistency (allowing choice by a DM), or explicitly considers alteration separately, or facilitates DM control over thresholds.

Table 2.6: Inconsistency Reduction Approaches

Approach	Ordinal	Cardinal	Reciprocal properly Maintained	Scale maintained	Alteration consideration	DM set threshold
Xu and Wei [81]		✓	✓		*	
Cao et. al [82]		✓	✓		*	
Siraj et. al [83]	✓	***	✓	✓		****
Costa [84]		✓				
Wang et. al [85]		✓	✓	*****	**	
Sun et. al [86]		✓	✓		**	

\* Only as a constraint

\*\* Only as part of a single objective

\*\*\* Only as a tie breaker

\*\*\*\* Has trail of each reversal to 0 cycles

\*\*\*\*\* Original judgments and altered judgments represented as fuzzy numbers

### 2.3.6 Inconsistency Conclusions

Types of inconsistency have been discussed, along with measures that have been proposed to quantify inconsistency within a set of judgments. Proposed approaches to alteration of judgments looking to reduce inconsistency within a judgment set show little emphasis upon flexibility, with no facilities for a DM to choose how inconsistency is to be measured, ordinal and or cardinal. Furthermore some approaches do not always maintain the reciprocal properly of the original PCM, and some approaches find solutions with values outside of the original judgment scale. When approaches consider alteration to the judgments in the pursuit of inconsistency reduction, it is not explicitly considered, instead it is only considered as part of a single objective or as a constraint. Additionally when alteration is considered little effort made to provide semantic meaning for a DM regarding how their judgments have altered. Furthermore when seeking to reduce

inconsistency to a threshold value they do not provide control to the DM to define the threshold value.

## **2.4 Group Decision Making**

For many real-world decisions the opinion of multiple DMs is utilised, either to avail of their combined expertise or to incorporate conflicting views and experiences. It is generally considered that a group of DMs can make superior decisions than a single DM [1]. Many instances of belief in such a view include for example, in law counts where guilt is determined through a group of jurists, or where a cabinet of ministers oversees national policy decisions, or when a large company's cooperate strategies as decided upon by its board of directors. Reasoning as to why a group can make better decisions include greater overall knowledge within the group than with an individual, as well as a potential increased watchfulness of errors and ambiguities [1]. Additionally through the combination of multiple views, bias that might be present with just a single DM will be eliminated [17] (or at least diluted). However equally important is the consideration that a group of DMs face issues relating to the incorporation of conflicting views. Such cases will make synthesising the views of the group of DMs less than straightforward. Next discussions of the additional issues of Group decision making within MCDA are presented.

### **2.4.1 Group MCDA**

Within MCDA, a group of DMs using a method such as AHP need to consider how derived preference vectors calculated at each level reflect the combined views of the group. Additionally in methods such as TOPSIS a group needs to consider how the weights of importance of the criteria can reflect the combined views of the group. For Group MCDA, there are a number of additional considerations, these concern the formulation of the problem, the weights of importance of each DM and the aggregation of the group's views into a decision outcome. These additional considerations are discussed next.

#### **2.4.1.1 Formulation of Problem**

The first concern is how the stage of formulation of the decision problem and its elements will be defined. Formulation may be defined by a single overseeing DM after which the other DMs then give their views upon the problem [1]. For example, for a government decision a short list of alternatives along with criteria to assess them against may already have been formulated by the government who then wish to get the opinions of experts in

the field with regard to the alternatives and criteria. Alternatively formulation of the decision may occur via a more interactive approach between the DMs involved [18]. Such interaction may take place through for example an “awareness session” as proposed in [87], in which through discussions the group of DMs fashion the elements for a decision problem.

#### **2.4.1.2 *DM Weights of importance***

Due to differences in rank or expertise within a group of DMs the importance of each DM’s views will invariably not always carry equal weight. The weights of importance of a group of DMs may be such that they reflect the expertise of the DMs [88]. Therefore the weights of importance of the DMs involved need to be considered as well as how they will be calculated.

Weights may be determined by a decision overseer, who decides how much weight each DM is assigned. Alternatively weights may be based upon representation within the decision where each weight is determined by the amount of effect the decision outcome will have upon each DM. Alternatively a more participatory approach to the calculation of DM weights can be utilised. Ramnathan and Ganesh proposed a method utilizing PC in which each DM weights each of the other DMs through PC [89], from which a vector ranking of DM weights is derived. Such an approach may be open to abuse from DMs deliberately skewing results to seek a higher weight. Additionally such an approach also requires DMs to have accurate knowledge of the other DMs involved. Such uncertainties could be modelled through modelling the comparisons through fuzzy numbers as an approach in [90]. A similar participatory approach to weights calculation is proposed in [87], here DMs use PC to assess expertise only towards the other DMs, and the approach additionally facilities vetoes during the comparisons. Such participatory approaches to weights calculation add additional overhead to the amount of elicitation required from each DM for the decision.

A different type of approach was proposed by Cho and Cho to utilise the DMs’ inconsistency measures in determining their weights [91]. Here the DMs’ inconsistency levels are utilised (based around the CR value of each DM) and more weight is given to more consistent DMs. As we have seen inconsistency affects the accuracy of preference vectors derived, therefore giving less weight to inconsistent DMs will reduce its adverse effects in a final ranking (and potentially make DMs strive to be more consistent so their opinions carry more weight). However an inconsistency measure is certainly not

synonymous with (domain) expertise of a DM and inconsistency may occur for a variety of reasons as we saw earlier.

Deriving weights of importance for a group of DMs is usually not straightforward, and hard to evaluate a priori. Therefore being able to alter weights dynamically and analysis their impact upon group aggregation would be useful.

#### **2.4.1.3 *Synthesis of DMs' views***

When PCs are utilised within a group environment, the process of deriving a preference vector needs to incorporate the synthesis of the group of DMs' views into the formulation of a single preference vector for the group. Generally four approaches can be taken [92]:

1. Consensus
2. Consensus Vote
3. Aggregation of individual judgments
4. Aggregation of individual priorities

The first of these involves discussions between the DMs about each judgment to arrive at a single entry for each judgment. Such an approach will only be effective when the group of DMs are a synergistic group and not a collection of individuals which may result in much discussion with no agreement reached [93]. Additionally any differences in opinion and compromises required to reach agreement will be lost hindering traceability. Furthermore such an approach makes little practical sense when the number of DMs is large or in different locations and time zones. When there is much disagreement between the DMs then the second approach of Consensus Voting could be used to try to achieve a single entry for each judgment through voting. Such approaches retain only the decided or voted for judgment hence reveal no information about the separate views between the group hindering traceability. Conversely the third and fourth approaches look to elicit separate judgments from each DM for each judgment. The problem then becomes how to aggregate these individual judgments together, and how to explicitly incorporate DM weights of importance.

Through eliciting separate judgments from each DM a PCM for each DM can be created. From these we then seek to derive a single preference vector, representing the combined preferences of all the DMs. Given a group of DMs providing their PC preferences for a set of elements, the problem is to aggregate the PCM of each DM into a single preference vector with additional consideration of the weights of importance of

the DMs. A further consideration for such scenarios is how constraints might be incorporated into the aggregation process and how they could then aid in a negotiation process [94]. When individual judgments are elicited from each DM, aggregation can then take the form of either Aggregation of Individual Judgments or Aggregation of Individual Priorities [34]. In the former the judgments from each DM are aggregated into a single set of judgments from which a single group preference vector is derived, see Figure 2.6. In the latter a separate preference vector is derived from each DM's judgments, then the set of preference vectors are aggregated into a single preference vector, see Figure 2.7.

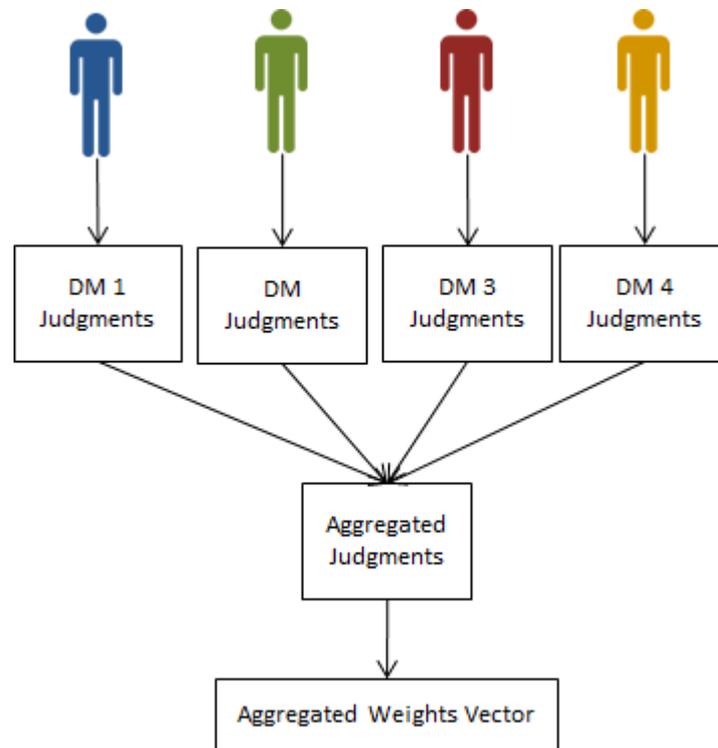


Figure 2.6: 4DM aggregation of individual judgments

Two prominent methods to these aggregation approaches are the Geometric Mean Method (GMM) for aggregation of individual judgments and the Weighted Arithmetic Mean Method (WAMM) for aggregation of individual priorities. When all the DMs involved are perfectly consistent it has been shown that both these mathematical approaches will produce the same final group preference vector [95]. However as we have seen inconsistency for a single DM is almost inevitable for more than a handful of elements so with more DMs involved the more likely inconsistency will be present. We now look in detail at these two approaches of aggregation of individual priorities and

aggregation of individual judgments. An example of judgments from 4DMs (of equal weights) for 4 elements using the 1-9 scale is shown in Table 2.7, which will be used to discuss these approaches.

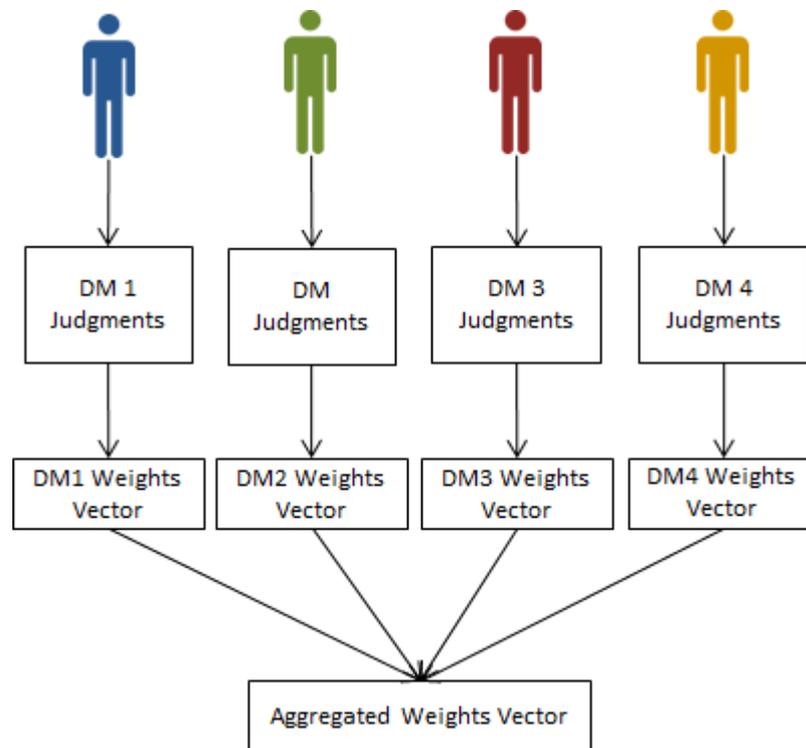


Figure 2.7: 4DM aggregation of individual priorities

### 2.4.2 Aggregation of Individual Priorities

Aggregation of Individual Priorities involves the calculation of a preference vector for each DM from their judgments. Then a single preference vector can be calculated through aggregation of the set of these preference vectors as shown in Figure 2.7. The Weighted Arithmetic Mean Method (WAMM) [89] is utilised within aggregation of individual priorities to aggregate each separate DM preference vector into a single aggregated preference vector. This is generally done using the arithmetic mean<sup>3</sup> and the weights of the DMs, generally normalised to sum to 1. Given a problem with  $n$  elements and a group of  $D$  decision makers from which a separate preference vector  $x_1$  to  $x_D$  has been derived we can calculate the WAMM from:

<sup>3</sup> Alternatively the geometric mean could be utilised for aggregation of individual priorities calculation, see [98]; however [89] conjecture that the arithmetic mean should be used.

$$WAMM = \sum_{j=1}^N \sum_{i=1}^D x_{ji} w_i \quad (2.12)$$

Where  $x_{ji}$  is the priority of element  $j$  for DM  $i$  and  $w_i$  is the weight of decision maker  $i$ .

Table 2.7: 4DM Group Example Data

<b>DM1</b>					<b>DM2</b>				
CR: 0.29 L: 0					CR: 0.35 L: 1				
	E1	E2	E3	E4		E1	E2	E3	E4
E1	1	4	3	5	E1	1	1/6	1/5	1/4
E2	1/4	1	6	8	E2	6	1	1	1/3
E3	1/3	1/6	1	6	E3	5	1	1	6
E4	1/5	1/8	1/6	1	E4	4	3	1/6	1

<b>DM3</b>					<b>DM4</b>				
CR: 1.6 L: 2					CR: 0.31 L: 0				
	E1	E2	E3	E4		E1	E2	E3	E4
E1	1	5	5	1/9	E1	1	1	1/6	1/7
E2	1/5	1	1/5	2	E2	1	1	1/9	1/3
E3	1/5	5	1	4	E3	6	9	1	1/6
E4	9	2	1/4	1	E4	7	3	6	1

Using WAMM to perform aggregation for our example we first derive individual preference vectors for each DM as shown in Table 2.8 (here using the GM prioritization method).

Table 2.8: 4DM Group Example Separate Preference Vectors

	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>
<b>DM1</b>	0.49	0.33	0.13	0.04
<b>DM2</b>	0.06	0.24	0.47	0.24
<b>DM3</b>	0.30	0.12	0.33	0.24
<b>DM4</b>	0.07	0.07	0.29	0.57

From these four preference vectors the WAMM preference vector is shown in Table 2.9:

Table 2.9: 4DM Group AMM aggregation of priorities Preference vector

	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>
<b>Aggregated Preference Vector</b>	0.23	0.19	0.31	0.27

The WAMM has no consideration of the level of alteration that each DM's views undergo in reaching consensus, as the resulting final preference vector gives no indication of each DM's compromise, which hinders the traceability of the aggregation process. Furthermore the WAMM has no capability to facilitate constraints of tolerance to be defined by the DMs to control the amount of compromise their views may undergo in attempting to reach consensus. Additionally the WAMM has no consideration of inconsistency during aggregation which, if high in a DM's judgments, will adversely affect the accuracy of the individual preference vectors derived for each DM. In our example, DM3 is highly inconsistent, and all 4 DMs have initial CR values greater than 0.1 which will affect the accuracy of the aggregation.

### 2.4.3 Aggregation of Individual Judgments

During aggregation of Individual Judgments each judgment for each DM is aggregated one by one into the creation of a single aggregated set of judgments. From this aggregated set of judgments a single preference vector can then be derived as shown in Figure 2.6. The Geometric Mean Method (GMM) [93] can be used to aggregate the PCM's of multiple DMs into a single aggregated PCM. Originally proposed under the assumption of equal weights of importance of each DM, a weighted GMM approach (sometimes referred to as the WGMM) can calculate a weighted aggregated PCM, incorporating unequal DM weights of importance. A single group preference vector is then derived from this aggregated weighted PCM. The geometric mean should be utilised for aggregation of individual judgments as opposed to the arithmetic mean to ensure the reciprocal property of the judgments is preserved [96]. For example, given judgments from 2 DMs for 2 elements where DM1 prefers Element1 9 times more than Element2 and DM2 prefers Element2 9 times over Element1 - thus prefers Element1 1/9 over Element 2. From these 2 judgments we see that the 2 DMs' strength of preference regarding these 2 elements are opposing and the aggregation of these equally extreme

views should undergo equal compromise (assuming equal DM weights of importance). Using the geometric mean as aggregation this is the case:

$$\sqrt{1/9 \cdot 9} = 1. \quad (2.13)$$

However use of the arithmetic mean results in unequal compromise during aggregation (here favouring DM1):

$$\frac{1/9 + 9}{2} = 4.56 \quad (2.14)$$

During such aggregation it has been shown that Pareto optimality may not be maintained in such aggregation [89]. If all DMs judge that they prefer element A over B yet within the aggregation this is reversed then Pareto optimality is not maintained. However Van den Honert and Lootsma contend that such cases may occur and are expected as the aggregation is attempting to calculate a compromise of DM views and is not representative of any one opinion of the group of DMs [97]. However knowledge of such compromise is lost during the GMM aggregation process therefore hindering tracability.

Given a group of  $D$  decision makers a set of judgments is elicited. We calculate each aggregated judgment  $A_j$  from:

$$A_{j_i} = \sqrt[1/D]{\prod_{i=1}^D j_i} \quad (2.15)$$

where the  $i$ th DM's judgment is  $j_i$ . Here assuming equal weights for each DM, given the weights of DMs, each aggregated judgment can be calculated via [98]:

$$A_{j_i} = \sqrt[1/D]{\prod_{i=1}^D j_i^{w_i}} \quad (2.16)$$

Subject to  $\sum_{i=1}^D w_i = 1$

where  $w_i$  is the weight of the  $i$ th DM. Using the GMM to perform aggregation for our example first an aggregated PCM is derived, shown in Table 2.10. From this aggregated PCM a single preference vector is derived, shown in Table 2.11 (here using the GM prioritization method).

Table 2.10: 4DM GMM Aggregated PCM

	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>
E1	1	1.35	0.84	0.37
E2	0.74	1	0.60	1.15
E3	1.19	1.65	1	2.21
E4	2.66	0.87	0.45	1

Table 2.11: 4DM Group GMM aggregation of priorities Preference vector

	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>
<b>Aggregated Preference Vector</b>	0.19	0.20	0.35	0.25

We observe that a different final preference vector has been derived for our example from the GMM and WAMM approaches, due to inconsistency being present within the DM judgments. Like WAMM, the GMM has no consideration of the levels of compromise that each DM's judgments undergo during aggregation to reach a consensus. This hinders understanding and traceability of the aggregation process. Furthermore the calculated aggregated PCM does not maintain the judgment scale utilised to elicit the DM judgments, making it harder for the DMs to discern how their judgments have altered to reach this consensus. Similarly the GMM has no capability to facilitate constraints of tolerance to be defined by the DMs. Additionally, the GMM does not consider inconsistency during the aggregation process and it has been shown that the level of ordinal inconsistency may actually increase during the aggregation process [83]. An example taken from [83], modified so all the initial judgments adhere to the 1-9 scale, is shown in Figure 2.8. From this example we can see that there is a single 3-way cycle in each DM's views whereas there are 2 cycles within the aggregated PCM. Regarding cardinal inconsistency [99] and [100] showed that when using the GMM the level of cardinal inconsistency in the aggregated PCM will be at most that of the most inconsistent DM, with CI utilised as the inconsistency measure in [99] and CR as the inconsistency measure in [100]. However as the GMM does not consider how to tackle inconsistency

the aggregation may result in an aggregated PCM with unacceptable levels of cardinal inconsistency, when adhering to the CR threshold measure of 0.1. For example, Figure 2.9 shows the judgments of 2 DMs, one whose judgments has CR below 0.1 and one whose judgments do not. The aggregated PCM derived with GMM results in a PCM with CR: 0.26, so greater than 0.1. As we have seen high inconsistency within a PCM adversely affects the accuracy of a preference vector derived from it. However aggregation at the judgment level would allow the opportunity to seek to reduce inconsistency during the aggregation into a single set of judgments, which is not possible with aggregation of individual priorities.

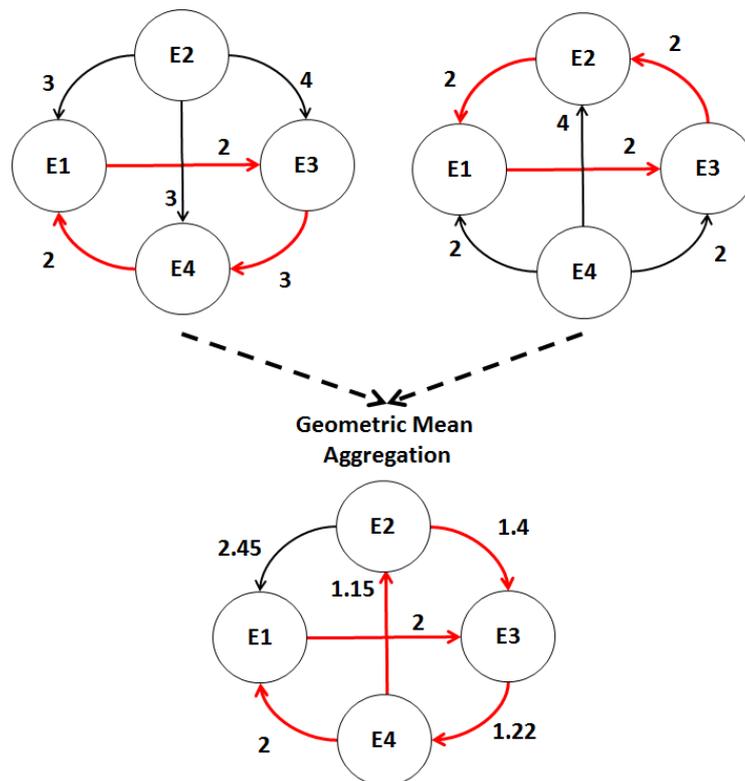


Figure 2.8: Ordinal inconsistency increasing during GMM aggregation

#### 2.4.4 Other aggregation Approaches

Aggregation approaches have been proposed to consider group decision making scenarios with incomplete sets of judgments, through linear programming and fuzzy programming in [101] and through a bayesian based approach in [102]. Such scenarios may occur when a DM has insufficient expertise to judge every comparison. Jaganathan et al. proposed a fuzzy AHP approach to aggregation in [90] which makes use of fuzzy numbers within the AHP method for group decision support. Bryson and Joseph proposed an aggregation

approach utilizing logarithmic goal programming techniques for reaching consensus where preference vectors are represented as numeric intervals [103].

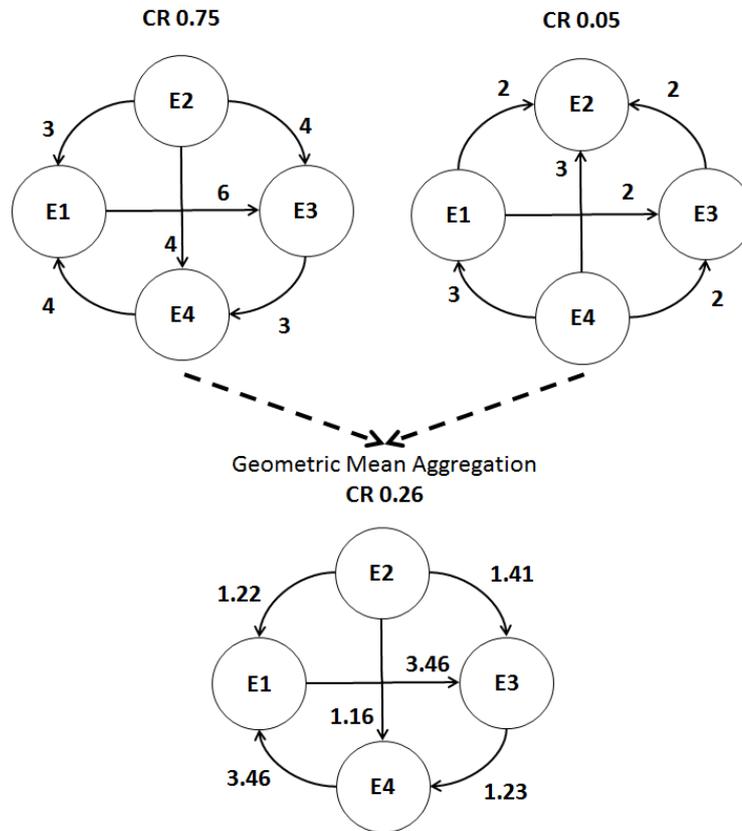


Figure 2.9: CR value greater than 0.1 during GMM aggregation

#### 2.4.5 Aggregation of PCs Discussions

Regarding the WAMM and GMM approaches for aggregation of a group of DMs, there has been discussion of which is most appropriate [89]. Forman, Peniwati recommend that when considering both the GMM and the WAMM, the most suitable mathematical aggregation generally depends on largely unknown information, such as, for example, if the group is a synergistic unit or a collection of individuals [98]. However such information invariably is unavailable at the start of a decision process.

Both WAMM and GMM approaches have no explicit consideration of the amount of alteration that each DM's views undergoes to reach consensus. In such aggregation approaches individual identities are lost within the aggregation [17]. This hinders traceability of the aggregation process as well as validity of the decision outcomes. This additionally hinders extraction of knowledge from the group of DMs, such as which DMs are most similar and which are most in conflict. Descriptive approaches have been proposed that look to analyse a group of DMs. Multiple authors have proposed measures

of dispersion that seek to determine when the group is not homogenous enough to recommend that further discussions are required before aggregation, see [104], [105]. Here confidence intervals and geometric dispersion values are utilised to measure if the group is harmonious, see [104], [105]. However such measures are hard for a DM to semantically interpret with respect to their judgments. The GAIA (Geometrical Analysis for Interactive Aid) plane [106] can be utilised as a descriptive extension of the PROMETHEE MCDA method [25] for a single DM. GAIA uses the PCA dimensionally reduction technique [107] to visualise in 2-dimensions the alternatives with respect to each other and the criteria. The GAIA plane is also usable within a group setting [108], where the group is modelled through an additional dimension of data to be reduced into 2-dimensions. This can then show in 2 dimensions DMs in relation to each other and the alternatives, see [108]. An approach was proposed to help identify outliers within a group of DMs within group AHP problems in [92]. Using individual preference vectors from each DM the approach looks to visualise the group through the use of Sammon map dimensionally reduction technique [109]. Exposure of outlying DMs may make such DM's judgments more objective [92]. The measure is the distance that each DM is from the preference vector generated from the GMM aggregation approach, therefore the knowledge found, as with the GAIA plane may be difficult for a DM to semantically interpret with respect to their original judgments.

Additionally both the WAMM and GMM approaches lack provision for DMs to define thresholds upon their judgments to be modelled as constraints in the aggregation process. Such thresholds should allow DMs to define the amount of alteration to their judgments they are willing to concede in looking to reach a consensus and hence increase their control over the process. The lack of semantic meaning of alteration is a hindering factor to DMs setting meaningful constraints.

Furthermore neither the WAMM nor GMM consider inconsistency reduction during the aggregation process which may even increase during the aggregation process. As we have seen earlier, inconsistency affects the accuracy of any preference vector derived.

As we have discussed, determining precise DM weights is not straight-forward. Both the WAMM and GMM incorporate weights into the aggregation process as fixed values and any alteration in DM weights requires re-running of the aggregation process. A more flexible handling of DM weights during aggregation would be beneficial, to facilitate sensitivity analysis.

In conclusion we have identified four areas of limitations of current aggregation approaches, lack of:

- Modelling compromise in semantically meaningful ways
- Facilitating DM constraints upon alteration
- Considering inconsistency during aggregation
- Allowing dynamic DM weights of importance

## **2.5 Summary**

MCDA and its prominent methods are an active area of research. Though facilitating a separation of concerns, PC enables an accurate extraction of a DM's preferences and are a primary part of MCDA methods such as AHP and ANP and can be used as an extension for methods such as TOPSIS to derive criteria weights. Inconsistency within PC is almost inevitable and it affects the accuracy of derived preference vectors from a set of judgments. Approaches that look to reduce inconsistency lack flexibility, interactivity and specific consideration of the alteration to the judgments in a way that is semantically meaningful to a DM. Group aggregation of PCs seek to aggregate the views of multiple DMs to reach a single consensus preference vector. Current aggregation approaches lack provision to model the compromise that each DM's views undergo during aggregation, to facilitate DM constraints, to consider inconsistency during aggregation and to dynamically incorporate DM weights of importance.

## Chapter 3 Measures of Compromise

### 3.1 Introduction

In this chapter measures of compromise for determining alteration to judgments are proposed that seek to be semantically meaningful to DMs. The proposed measures seek to be relatable to by a DM, so that they can enhance the knowledge and understanding a DM can glean from a decision. When using PC for eliciting views from a DM there are multiple scenarios where being able to measure the difference between sets of judgments should be beneficial. Such as:

- During inconsistency reduction when a DM's judgments are altered in some way to look to reduce the amount of inconsistency present;
- During group aggregation when a set of aggregated judgments is sought, during which it should be useful to measure the amount of alteration each DM's judgments undergo to reach consensus;
- When dealing with a large group of DMs being able to measure the difference between DM views would aid tasks such as identifying and grouping together those DMs with similar views;

By looking to define measures that are semantically meaningful to a DM we seek to aid a DM in a richer understanding of these scenarios. Measures that are meaningful to a DM should enrich the decision process by making such scenarios more auditable and traceable and allow richer sensitivity analysis to be performed. Having measures that are meaningful should also aid a DM in being able to set informed constraints and vetoes within these scenarios; for example, to aid a DM in defining the amount of alteration they are willing to tolerate when looking to reach a group consensus.

In this chapter judgment set representations and encodings are presented. Next, an analysis of how alteration to judgments can be measured is discussed and a set of measures of compromise are proposed. Finally a brief outline is presented of how these measures will be utilised within subsequent chapters of this thesis.

## 3.2 Judgment Set Representation

Given a problem with  $N$  elements we elicit a PC from a DM, using a preference scale such as the 1-9 scale, of each pair of elements within the  $N$  set of elements to construct a PCM. From the PCM a preference vector ranking of the elements under consideration is derived. Due to the reciprocal property and self-comparisons of a set of judgments, as we saw in Chapter 2, the minimum number of judgments  $J$  required to construct a complete PCM is:

$$J = \frac{N(N - 1)}{2} \quad (3.1)$$

From a completed PCM we can extract a set of  $J$  judgments that will contain all the information to reconstruct the PCM. The top triangle of judgments from a PCM can be extracted to define  $J$ . We can represent this set of  $J$  judgments as a one-dimensional array. For the range of values of a bounded judgment scale we can then convert the judgments into integer values of their position along the scale that they represent. When using the 1-9 scale there are 17 possible values that each judgment can take so we can convert each judgment into an integer value between 1 and 17. For example, from a completed PCM when  $N=4$  we extract 6 judgments to define  $J$  as a vector, which we can represent as an encoded integer vector, as shown in Figure 3.1. Then for a scenario such as looking to reduce inconsistency within a set of judgments we can measure the amount of alteration between the original judgments represented in this way and a set of judgments altered to reduce inconsistency also represented in this way.

Such a representation will ensure that each step along a scale is considered as equal alteration. For example, the step between 1/8 and 1/6 should be the same amount of alteration as between 6 and 8 – both 2 steps along the scale, as shown in Figure 3.2. This is important to ensure no disparity between how judgments are elicited (if a question is phrased “which is preferred A or B” or “which is preferred B or A”). Conversely if each judgment is represented as a decimal placed value alteration between fractions will be treated as smaller compared to their whole number equivalent values. Additionally such a representation will ensure that the reciprocal property of a PCM is maintained. Constructing a PCM from a set of judgments represented in this way will maintain the reciprocal property of the judgments. Conversely, as discussed in Chapter 2, some previous approaches to reducing inconsistency represent the whole of a PCM in its encoding and the found solutions do not always maintain this reciprocal property.

Furthermore representation of a PCM in this way when utilised to alter the judgments in some way will ensure that the altered judgments maintain the original scale of judgments. This will ensure that an altered set of judgments will be more comprehensible to a DM, as they will be able to more easily interpret any alterations that their judgments have undergone, and will also maintain the model of representation of the original judgments. Conversely as discussed in Chapter 2 previous approaches to reducing inconsistency invariably look to find solutions with judgments outside of the original scale values. Additionally, during group aggregation, aggregation of individual judgment approaches such as the GMM invariably find solutions with judgments outside of the original scale values.

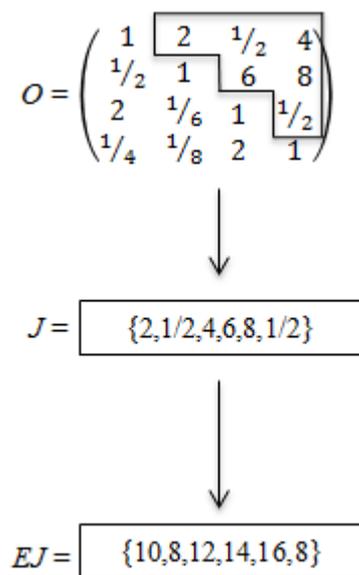


Figure 3.1: Example encoding of set of judgments for an N=4 PCM

Such encoding has been implemented for representation of judgment sets elicited using the 1-9 scale, due to its overwhelming prominence in theoretical and practical studies. Therefore the approaches and their examples presented in this thesis utilise this encoding and the 1-9 to elicit judgments from DMs. However our approaches are independent of a specific scale and could be extended to implement other bounded scales such as those discussed in Chapter 2. Next we will discuss how we can measure alteration in various ways through measures of compromise.

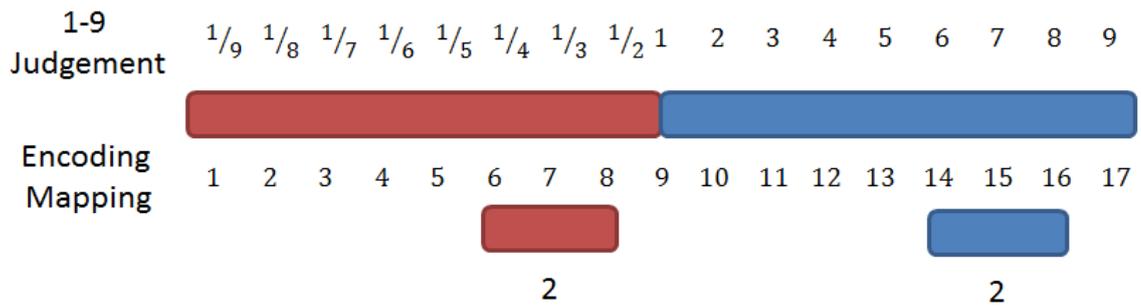


Figure 3.2: Judgment Encoding to ensure equal alteration across the scale

### 3.3 Measures of compromise

To measure the alteration between an original judgment and an altered judgment, or more generally the distance between 2 judgments, we consider how to quantify the distance between a pair of judgments. Then across a set of judgments we can calculate a measure of total distance between two judgment sets. Given a judgment set ( $O$ ) represented as a set of judgments  $\{o_1, o_2, \dots, o_j\}$  of cardinality  $J$ . We can look to measure the amount of alteration between  $O$  and a second *Altered* judgment set ( $A$ ) of judgments  $\{a_1, a_2, \dots, a_j\}$ . We measure alteration to a judgment in a number of ways:

1. Considering alteration as a binary value of whether the judgment has changed, we term this a **Judgment Violation**;
2. Considering the amount of alteration the judgment has changed by, we term this **Judgment Deviation**;
3. Gauging if the judgment preference has changed from one element being preferred to the other, we term this a **Judgment Reversal**.

Visually we can represent the measures of Violation, Deviation and reversal, as shown in Figure 3.3. For a whole set of judgments we can use these measures of alteration to a single judgment to define measures of compromise as.

1. **Number of Judgment Violations (NJV)**: a measure of the number of judgments that have changed;
2. **Total Judgment Deviation (TJD)**: a measure of the total amount of change between each judgment;

3. **Squared Total Judgment Deviation (STJD)**: a variant of TJD which gives more emphasis to larger amounts of deviation to a judgment;
4. **Number of Judgment Reversals (NJR)**: a measure of the number of judgments that have been reversed.

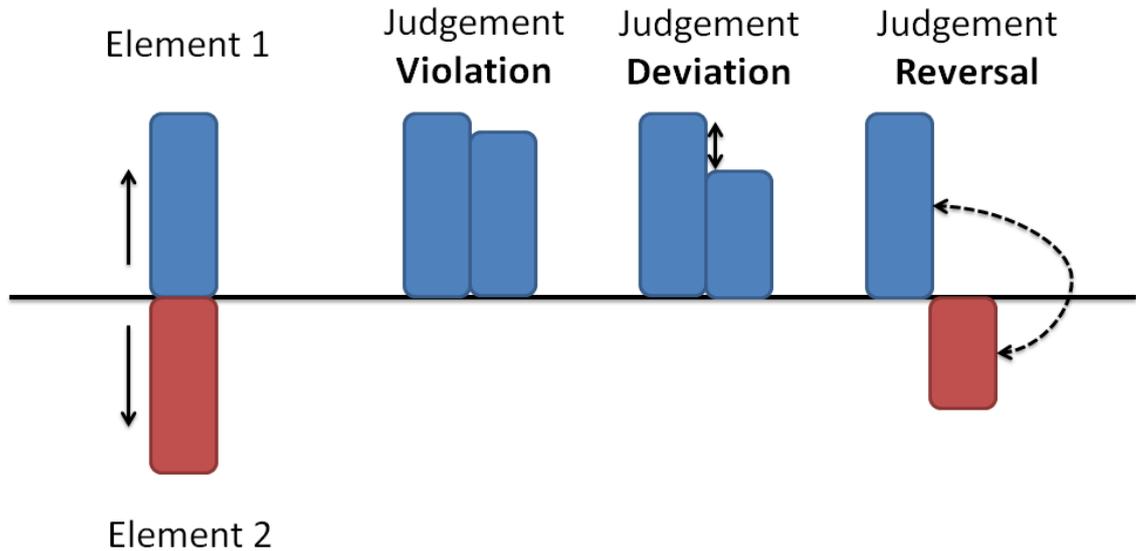


Figure 3.3: Measures of compromise

We now define each measure. To aid analysis within each definition we visualise the set of possible values of each measure for a single judgment across the range of values of the scale. When using the 1-9 scale, and using the key in Figure 3.4, we show how the value of a measure changes over the range of values of the scale for a single judgment.

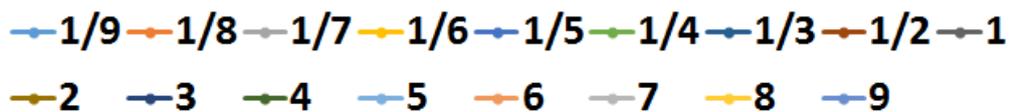


Figure 3.4: 1-9 scale value Key

### 3.3.1 Number of Judgment Violations

The Number of Judgment Violations (NJV) is a measure of the number of the original set of judgments that have changed, where  $\delta$  evaluates to 0 or 1 for each Boolean evaluation.

$$NJV = \sum_{j=1}^J \delta(o_j \neq a_j) \quad (3.2)$$

The range of values of NJV for a single judgments will be either 0 or 1. From NJV a DM can see how many of their judgments have altered, without consideration of the amount of change to each judgment. NJV could be useful when a DM is seeking solely to look to minimum the number of their judgments that change in the pursuit of a goal.

### 3.3.2 Total Judgment Deviation

Total Judgment Deviation (**TJD**) is a measure of the total amount of change between each judgment from the original judgments and an altered judgment set. It takes into consideration the amount of preference change between each judgment comparison.

$$TJD = \sum_{j=1}^J |o_j - a_j| \quad (3.3)$$

We can visually see the range of the measure for a judgment over the values of the 1-9 scale in Figure 3.5. From TJD a DM can get a sense of the total amount of alteration their judgments have endured. TJD could be useful when a DM is seeking to minimize the total amount of steps along the scale their judgments undergo in pursuit of a goal.

A modified version of the TJD measure is the Squared Total Judgment Deviation (**STJD**):

$$STJD = \sum_{j=1}^J (o_j - a_j)^2 \quad (3.4)$$

Here the deviations between the corresponding judgments in both sets are squared; consequently altered judgments with a large alteration in steps along the scale will have a greater impact upon the measure's total. The range of the measure for a single judgment is shown in Figure 3.6. We can observe how STJD gives greater emphasis to larger deviation changes, therefore STJD would be a useful measure when a DM is seeking to avoid large changes to their judgments in the pursuit of a goal.

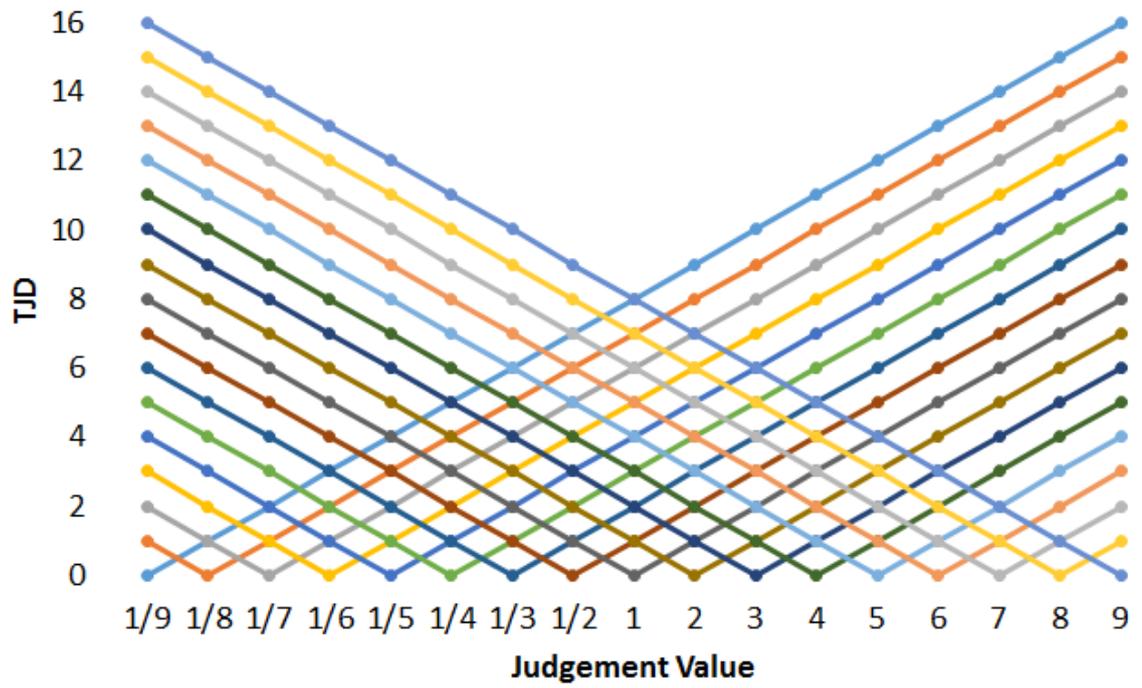


Figure 3.5: Total Judgment Deviation (TJD)

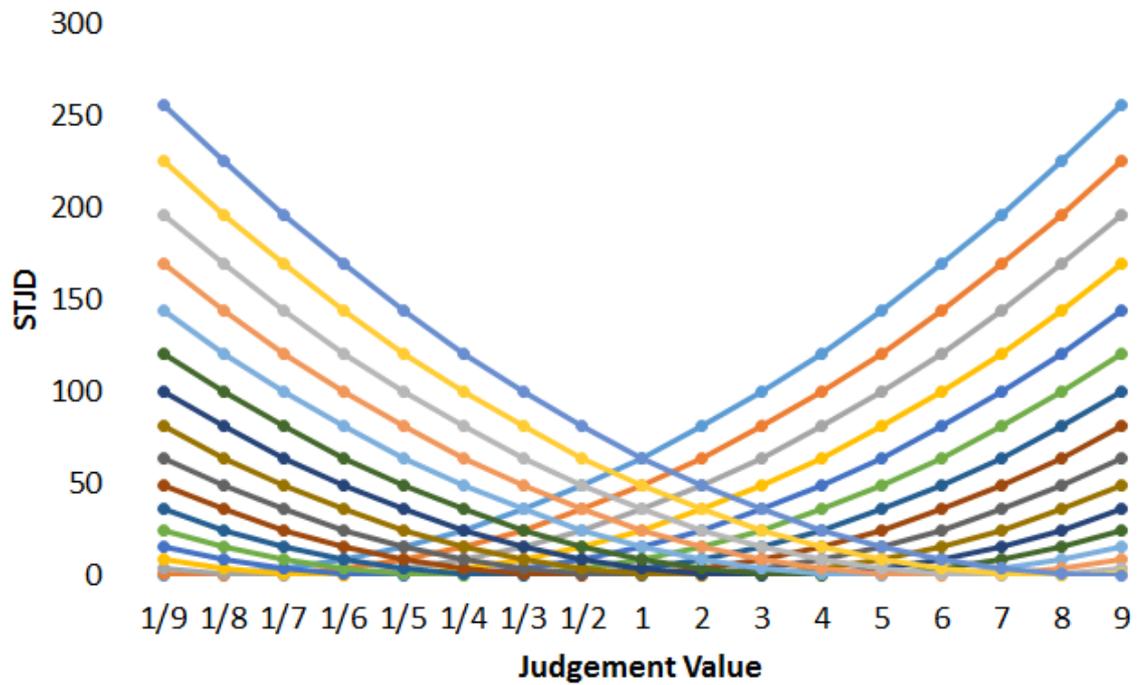


Figure 3.6: Squared Total Judgment Deviation (STJD)

### 3.3.3 Number of Judgment Reversals

The Number of Judgment Reversals (NJR) is a measure of the number of judgments from the original set whose preference has been inverted in an altered judgment set. For example, given an original judgment between elements  $x$  and  $y$  where  $x \succ y$ : if in an altered

judgment set it is the case that  $x < y$  then a judgment reversal has occurred. This measure of compromise also considers half reversals. Half reversals are defined as occurring when a judgment of equal preference is altered to be a judgment of not equal preference or a judgment not of equal preference is altered to be a judgment of equal preference. When using the 1-9 scale we can specify equal preference, greater than equal preference and less than equal preference, as 1, greater than 1 and less than 1 respectively.

$$NJR = \sum_{j=1}^J R_j \quad (3.5)$$

where

$$R_j \begin{cases} 1: o_i > 1 \text{ and } a_i > 1 \\ 1: o_i < 1 \text{ and } a_i < 1 \\ 0.5: o_i = 1 \text{ and } a_i \neq 1 \\ 0.5: o_i \neq 1 \text{ and } a_i = 1 \\ 0: \text{otherwise} \end{cases}$$

For a DM, NJR give a clear indication of the number of their judgments that have been inverted regardless of their strength, which we can see visually in Figure 3.7. The NJR could be a useful measure for a DM seeking solely to minimise the number of preference changes to their judgments in the pursuit of a goal.

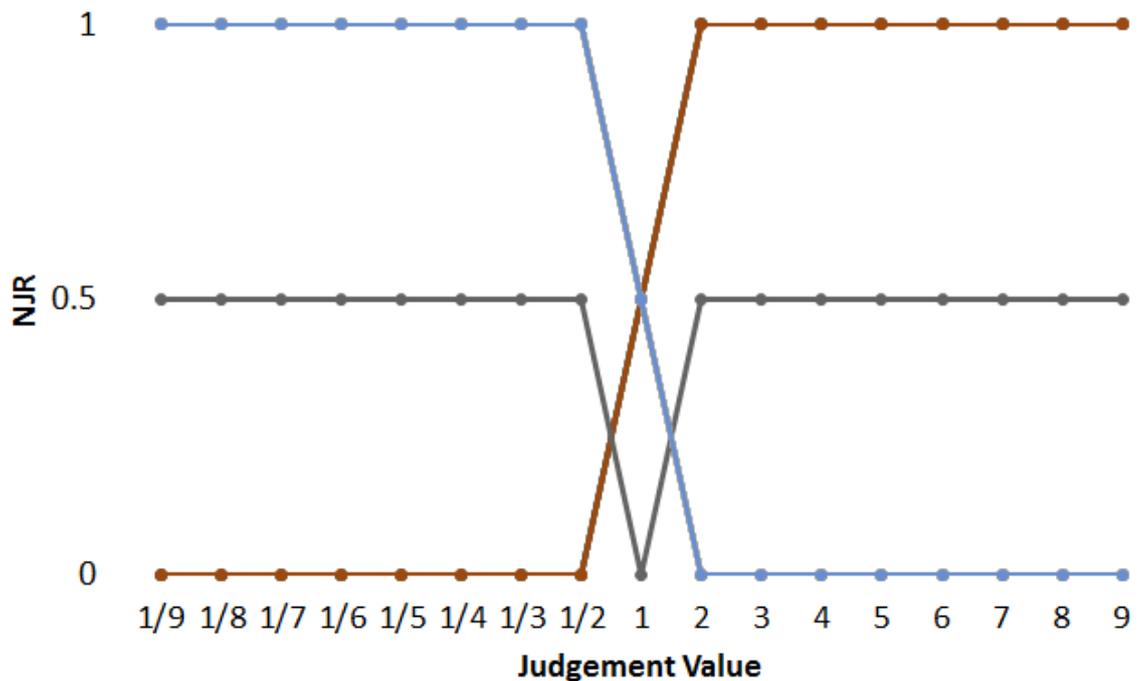


Figure 3.7: Number of Judgment Reversals (NJR)

### 3.4 Measures of Compromise Example

With these defined measures of change we show briefly how they can be utilised within the scenarios discussed in the chapter's introduction. The measures can be employed to aid understanding during a scenario of reducing inconsistency in a DM's judgments. For example in Figure 3.8: Left we have the DAG of the set of judgments we introduced earlier in Figure 3.1. Initial inconsistency measures of these judgments are CR: 0.55 and L: 2.0. In Figure 3.8: Right a DAG is shown of an altered version of the judgments with the inconsistency removed, so here CR: 0 and L: 0.

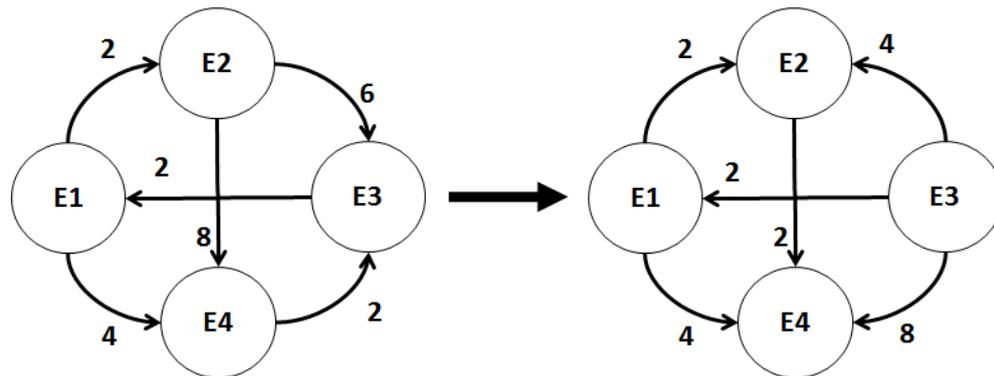


Figure 3.8 Left: Initial Judgment set. Right: Judgment set with inconsistency removed

We can encode these sets of judgments and measure the distance between them with the measure of compromise, as shown in Table 3.1. From this a DM can see that to facilitate the removal of inconsistency, just 3 of their judgments have changed, and that 2 of their judgments have been reversed. Additionally they see that a total amount of 22 scale steps occurred. Furthermore the large STJD value in comparison to the TJD value reveals to the DM that large deviation steps have occurred. From this a DM can discern descriptively how their judgments have been altered to, in this example, eliminate the inconsistency from their judgments.

Table 3.1: Example measures calculation

	Value
<b>NJV</b>	3
<b>TJD</b>	22
<b>STJD</b>	164
<b>NJR</b>	2

Additionally we can utilise the measures to help discern the difference in views between DMs, for scenarios of group aggregation. For example, Figure 3.9 shows as DAGs the judgments from 3 DMs, along with the measure of compromise of TJD and NJR values

between each pair of DMs. From this we observe that DM1 and DM3 appear closer in their views with both fewer reversals and less deviation between their judgments than between DM2 and either DM1 or DM3. Such descriptive analysis of the measures of compromise during the scenario of group aggregation could facilitate traceability and transparency of how each DM's judgments are effected to reach aggregation.

In these examples the measures of compromise have been used for descriptive purposes. Subsequent chapters will investigate how these measures can be utilised as objectives, such as using the measures of compromise to seek to reduce inconsistency for the minimum amount of alteration, and using the measures to seek to reach a group aggregation for the minimum amount of alteration.

### **3.5 Conclusions**

In this chapter we have defined a range of measures of compromise for measuring the alteration of a set of judgments or the distance between sets of judgments. The measures aim to be meaningful to a DM to allow them to more easily relate to and extract knowledge from them. Through defining measures that are meaningful to a DM we seek to aid richer approaches to decision making within scenarios such as inconsistency reduction and group aggregation. Measures that are meaningful to a DM should enrich the decision making process through making such scenarios more traceable and auditable. Additionally using meaningful measures should allow for richer interactivity and sensitivity analysis to be performed. A DM can more easily interpret the values of such measures and more easily interact within the scenarios for example through setting meaningful constraints. Various scenarios for which these measures can be utilised within are explored over the rest of the thesis. Chapter 4 proposes an approach to inconsistency reduction within a set of judgments using the measures to look to reduce inconsistency for the minimum amount of alteration. Chapter 5 proposes an approach to the aggregation of a group of DMs' judgments utilizing the measures to aid analysis of the amount of compromise that each DM undergoes during aggregation. In chapter 6 an approach is proposed for grouping a large group of DMs into groups utilizing the measures of compromise to help discern similarity between DMs and create a single views from a group of similar DMs to help reduce the complexity of large group aggregation.

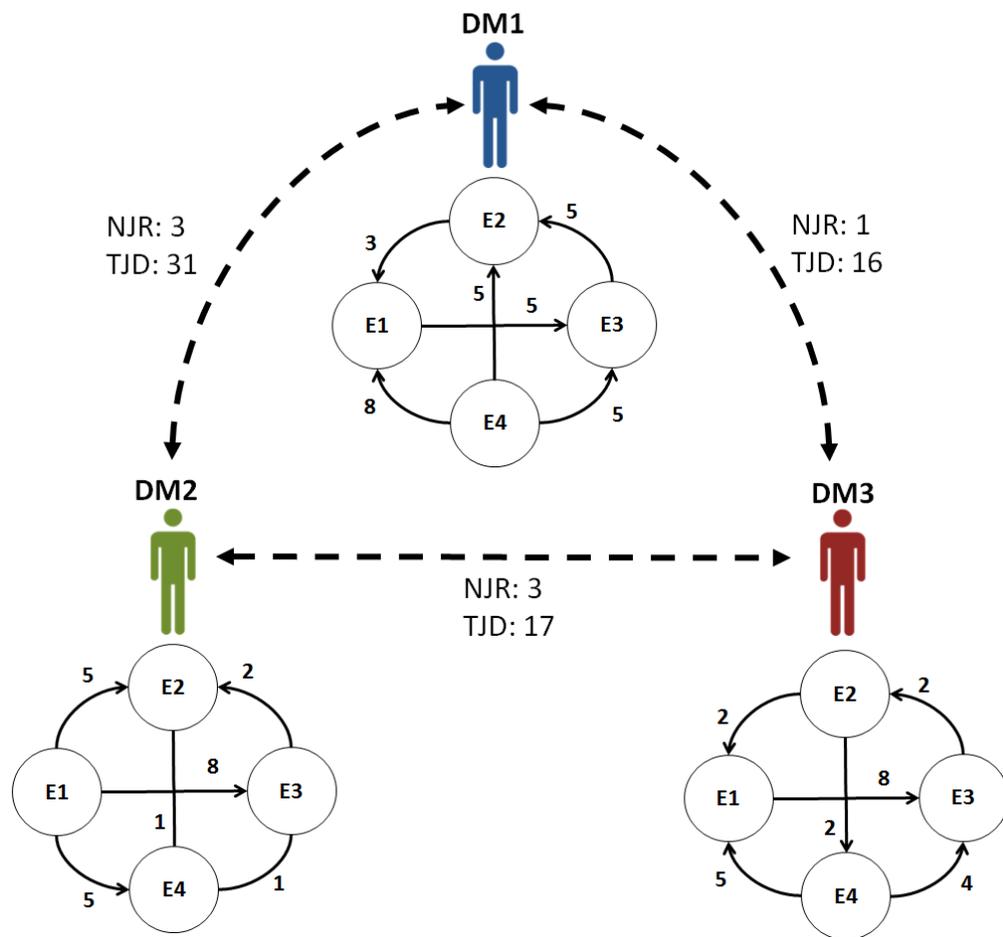


Figure 3.9: Measures of Compromise of difference between DM views

# **Chapter 4 Reducing Inconsistency in Pairwise Comparisons Using Multi-objective Evolutionary Computing**

In this chapter we present a new approach to reducing inconsistency within a set of PC judgments via Multi-Objective Optimisation (MOO). After the problem definition an outline of the stages of the MOO approach to the problem is presented along with the objectives usable within the approach. Next analysis of an exhaustive search implementation to solve the problem is discussed with comparisons to Multi-Objective Genetic Algorithms (MOGA) to assess their suitability for the problem. Following this how constraints are implemented within the approach are briefly discussed. This is followed by a number of examples of the approach. Finally conclusions are presented.

## **4.1 Automatically Reducing Inconsistency in Pairwise Comparison**

As outlined in Chapter 2, PCs can be utilised to elicit an accurate representation of a DM's views through facilitating a separation of concerns. However inconsistency may be present in a set of judgments elicited via PCs and such inconsistency is almost inevitable for more than a handful of elements. Inconsistency is an important consideration as its presence in a set of judgments will have adverse effects upon a derived preference vector ranking of the elements under consideration. Inconsistency may be tackled through looking to automatically alter the judgments to seek reduction. Multiple approaches to automatically reducing inconsistency in a set of judgments were analysed in Chapter 2, where their limitations were identified and discussed. Current approaches only consider either ordinal or cardinal inconsistency and also look to determine altered solutions with judgments that fall outside of the original judgment scale utilised. Furthermore alteration to judgments is only considered as constraints or as part of a combined single objective. Moreover little attempt is made to make the alteration required to facilitate the inconsistency reduction semantically meaningful to a DM. Additionally the approaches offer little or no capacity to allow a DM to define their own constraints and thresholds upon either inconsistency measure levels or levels of alteration. Therefore in this chapter

we present an approach to the reduction of inconsistency in a set of judgments that looks to overcome these limitations.

## **4.2 Approach to reducing inconsistency in a set of PCs**

The approach looks to optimally reduce inconsistency within a set of DM judgments through modelling inconsistency measures and measures of alteration to the DM's judgments as separate objectives via MOO. The approach takes in a set of judgments from a DM and looks to find *Altered Solutions*, which are new judgments sets that will be derived from the MOO process. Both cardinal and/or ordinal inconsistency can be considered giving a DM control over the type of inconsistency reduction to seek. Alteration to judgments is explicitly considered in the approach making use of the measures of compromise proposed in Chapter 3. The approach facilitates inconsistency reduction whilst also looking to minimise the amount of alteration to achieve the reduction. The use of the measures of compromise give a DM considerable control over how alteration is measured to meet their needs. Moreover they help a DM glean understanding of the process and knowledge of the trade-offs involved, helping both traceability and sensitivity analysis. Additionally the approach allows a DM to set constraints relating to the amount of inconsistency reduction they are seeking to achieve as well as the amount of alteration they are willing to tolerate. Furthermore the approach seeks to alter judgments in such a way that the judgments maintain the original scale utilised by the DM, hence allowing a DM to more easily discern how their judgments have altered and ensuring maintenance of the initial judgment scale used by the DM during judgment elicitation. The approach is independent of a specific prioritization method, so any method can be utilised to derive a ranking from the Altered Solutions found, enabling the approach to be flexible to different scenarios and DM preferences. The approach implements the 1-9 scale to elicit judgments and this scale is utilised within examples however, the approach could be extended to be used with any bounded scale, again enabling the approach to be flexible to different scenarios and DM preferences. Next the stages of the approach are outlined.

### **4.2.1 Stages of the Approach to reducing inconsistency in PCs**

The stages of the approach, shown in Figure 4.1, can be summarized as follows:

1. The number of elements of the problem is defined;

2. Judgments are elicited from the DM pertaining to their preferences between the elements;
3. The objectives for the MOO process are selected by the DM consisting of one or more measures of compromise objectives and one or more inconsistency measure objectives, see Section 4.2.3;
4. The set of objectives are then utilised within a MOO framework to find the set of trade-off Altered Solutions between the objectives, see Section 4.2.2;

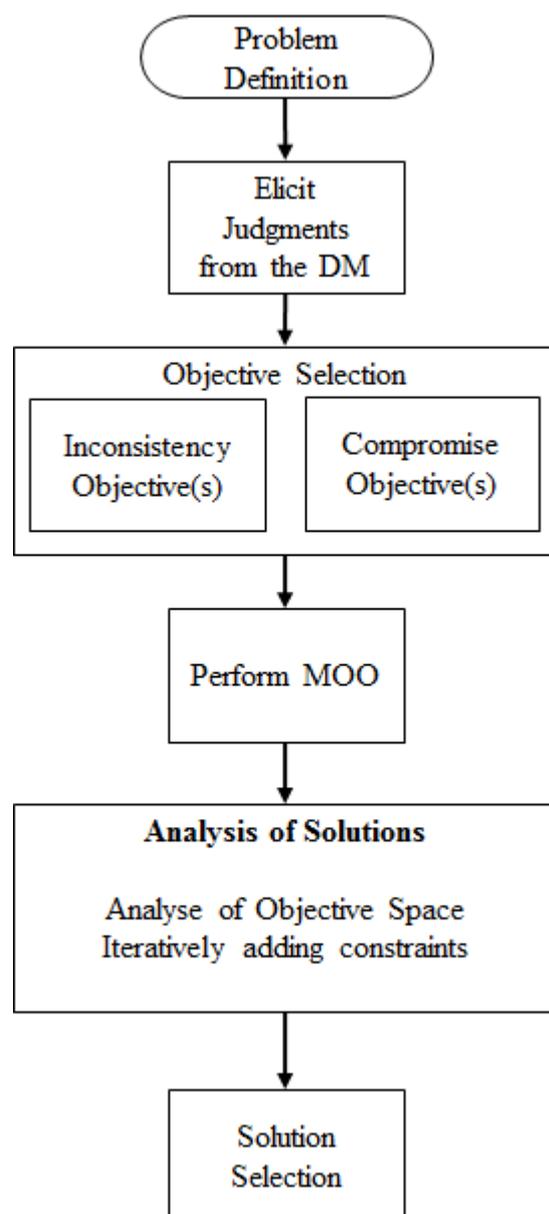


Figure 4.1: Flowchart of Approach to reducing inconsistency in a set of PCs

5. Analysis of the set of Altered Solutions can then be performed to aid the DM towards the selection of a single solution from the set of solutions. Such analysis can be performed via:
  - i. Gleaning knowledge of the trade-off front of solutions in the objective space, and of the nature of the inconsistency reduction and the compromise to facilitate it. Inspection of, and comparisons between, the solutions found can be performed to aid a DM in the selection of a final solution. Additionally such analysis of the nature of the trade-off front for the problem may aid a DM in recalibrating their goals as to what are achievable levels of reduction for various amounts of alteration;
  - ii. Additionally through analysis of the set of found Altered Solutions a DM can iteratively add feasible constraints to gradually drill down to a sub-region of the objective space to help select a final solution, see Section 4.2.5;

#### **4.2.2 Multi-Objective Optimisation (MOO)**

Many real-world problems consist of multiple, frequently conflicting, objectives. One approach to solving MOO problems is to weight each objective function and then combine them together to create a combined single objective that is solved via single objective optimisation. However such an approach dictates that the weights of each objective are defined by the DM prior to the optimisation which may be difficult for a DM when they are unsure of their preferences regarding the objectives. Furthermore such an approach reveals no knowledge regarding the relationship and nature of conflict between the objectives. Single objective optimisation looks to find a global optimal solution and thus returns a single solution to the DM. A single objective (minimisation) problem's cost surface is illustrated in Figure 4.2: Left. Alternatively the objectives can be optimised simultaneously. In such an approach there will not be a single solution - due to the conflicting nature of the objectives - instead a range of possible trade-off solutions will exist. Without additional information all these solutions are equally preferred [110]. When evaluating solutions with respect to multiple objectives we can distinguish between them via the notion of Pareto dominance. The set of trade-off solutions to a multiple objective problem are termed *Pareto optimal solutions*. For each solution any improvement in one of the objectives will result in a decrease within one or more of the

other objectives of the problem. This set of solutions map out the trade-off front of the problem termed the *Pareto front*. Solutions can be compared based upon their dominance with respect to the set of objectives of the problem. Given 2 individuals  $I_1$  and  $I_2$ :  $I_1$  is said to dominate  $I_2$  if for the set of the objectives  $O$  it has a greater objective value for at least one objective and no worse objective values for any of the other objectives:

$$O(I_1) \geq O(I_2) \quad (4.1)$$

$I_1$  is said to strongly dominate solution  $I_2$  if for each objective it has a greater objective value:

$$O(I_1) > O(I_2) \quad (4.2)$$

Solutions which are not dominated by any other solutions are termed non-dominated solutions. The set of non-dominated solutions of a problem represent the Pareto front of the problem. Figure 4.2: Right illustrates these concepts for a two-objective minimisation problem. MOO looks to determine a set of non-dominated solutions to best approximate the Pareto front of a problem.

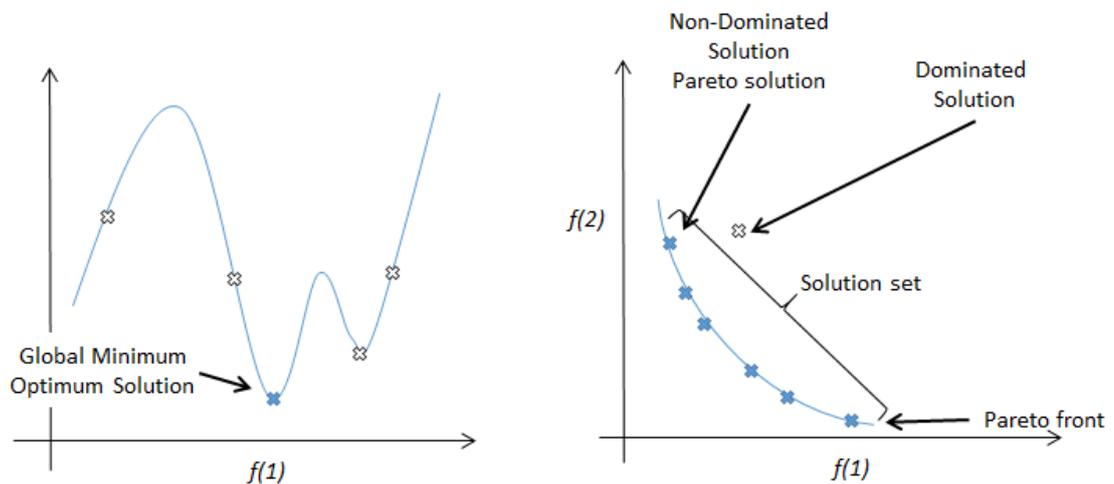


Figure 4.2 Left: Single objective problem. Right: Illustrative Pareto dominance definitions

The approach seeks to reduce inconsistency within a set of judgments through modelling inconsistency measures and alteration to the judgments as separate objectives (chosen by the DM) via MOO. Due to the conflicting nature between objectives of inconsistency measures and objectives of compromise measures, there will not be a single solution that

optimizes all the objectives; rather a range of non-dominated solutions will exist. Given a problem with  $n$  elements and a complete  $n$  by  $n$  PCM of judgments from a DM, a Judgment Set of Original judgments  $O$  of cardinality  $J$  can be selected, containing enough information to reconstruct the whole of the PCM.  $O$  can be represented as the upper triangle of a PCM and then encoded as defined in Chapter 3. We seek the set of non-dominated Altered Solutions for the chosen objectives. We represent each Altered solution as a judgment set of cardinality  $J$ , denoted as  $A = \{a_1, a_2, \dots, a_J\}$  obtained by minimising the set of objectives  $\Lambda$ . The MOO problem can be formulated as:

$$\text{Minimize } [\Lambda] \quad (4.3)$$

where

$$\Lambda = \{E, B\}$$

The set of objectives  $\Lambda$  consists of two subsets. The first subset  $E$  represents one or more measures of compromise objectives chosen by the DM; the second subset  $B$  represents one or more measures of inconsistency chosen by the DM. The approach additionally allows a DM to set constraints both upon the amount of inconsistency reduction they are seeking and upon the amount of compromise they are willing to tolerate in the pursuit of reducing inconsistency. Setting constraints upon inconsistency objectives allows a DM to set bounds upon the amount of inconsistency permitted within altered solutions. Thus a constraint  $f_j$  upon an inconsistency objective  $\beta_j$  from objective subset  $B$  is defined as:

$$\beta_j(A) \leq f_j \quad (4.4)$$

For example when the Consistency Ratio (CR) measure is chosen as an objective by a DM they can additionally choose to define a constraint upon the upper value of the objective such as 0.1 (thus adhering to Saaty's recommendation that acceptable PCMs should have a CR value no greater than 0.1) or any other threshold value of the DM's choosing. Setting constraints upon measures of compromise objectives allow a DM to set bounds upon the amount of compromise they are willing to accept to reduce inconsistency. Given a constraint of  $c_i$  upon measure of compromise objective  $\varepsilon_j$  from objective subset  $E$  the following constraint could be defined:

$$\varepsilon_i(A) \leq c_i \quad (4.5)$$

For example, when the measure of compromise Number of Judgment Reversals (NJR) is chosen as an objective by a DM they could additionally define a constraint upon the objective of 3, this way seeking only to find Altered Solutions with 3 reversals or less to their original judgments. So, the constrained MOO problem can be formulated as:

$$\text{Minimize } \{E, I\} \quad (4.6)$$

subject to

$$\begin{aligned} \varepsilon_i(A) &\leq c_i \\ \beta_j(A) &\leq f_j \end{aligned}$$

for  $i=1,2,\dots,p$ , and  $j=1,2,\dots,q$

where  $E$  is of size  $p$  and  $B$  is of size  $q$ .

An illustration with one measure of compromise and one inconsistency measure is shown in Figure 4.3. The illustration shows how from an initial Original PCM, a set of  $J$  judgments is extracted as the input to the MOO and an objective space of an inconsistency measure and a compromise measure with non-dominated solutions is shown. A DM may then select any of these non-dominated solutions from which a preference vector ranking can be derived. The objectives that are usable within the approach are outlined next.

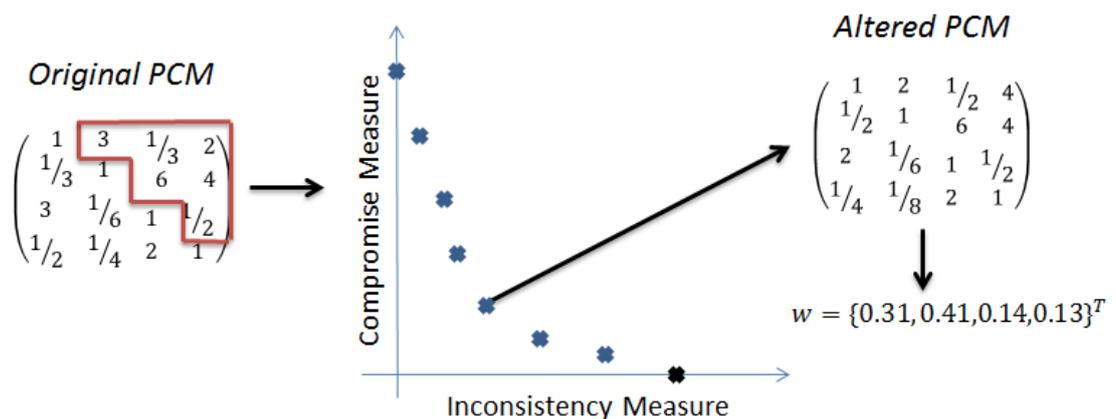


Figure 4.3: Illustration of MOO approach

### 4.2.3 Objectives usable within the Approach

The MOO approach utilises one or more measure of inconsistency and one or more measures of compromise as objectives. The objectives that are usable by a DM within the approach are shown in Figure 4.4.

The Consistency Measures Objectives are (for discussion of these see Chapter 2):

1. **CR**: Consistency Ratio [33]: a measure of the level of cardinal inconsistency present, proposed by Saaty along with an additional acceptability threshold value of 0.1;
2. **L**: An ordinal measure of the number of 3-way cycles present within a set of judgments that additionally considers equal preference;
3. **CM**: Consistency measure [73]: a cardinal measure of inconsistency that considers the most inconsistent triplet of judgments within the whole set;
4. **GCI**: Geometric consistency Index measure [72]: an inconsistency measure based upon the distance measurements between a preference vector derived using the GM prioritization method and the original judgments.

The Measures of Compromise Objectives are (for discussion of these see Chapter 3):

1. **NJV**: Number of Judgment Violations: a measure of the number of judgments that have changed;
2. **TJD**: Total Judgment Deviation: a measure of the total amount of change between each judgment;
3. **STJD**: Squared Total Judgment Deviation: a variant of TJD which gives more emphasis to larger amounts of deviation to a judgment;
4. **NJR**: Number of Judgment Reversals: a measure of the number of judgments that have been reversed.

Furthermore tackling the problem via a MOO framework allows for additional measures of inconsistency, say if a new measure is devised, to be implemented into the approach. Similarly any further measures of compromise that are defined could also be implemented into the approach. Additional objective measures would only need to define an evaluation function to be implementable into the approach.

We have defined how we will tackle the problem of reducing inconsistency via MOO. Next we investigate the implementation of the MOO problem.

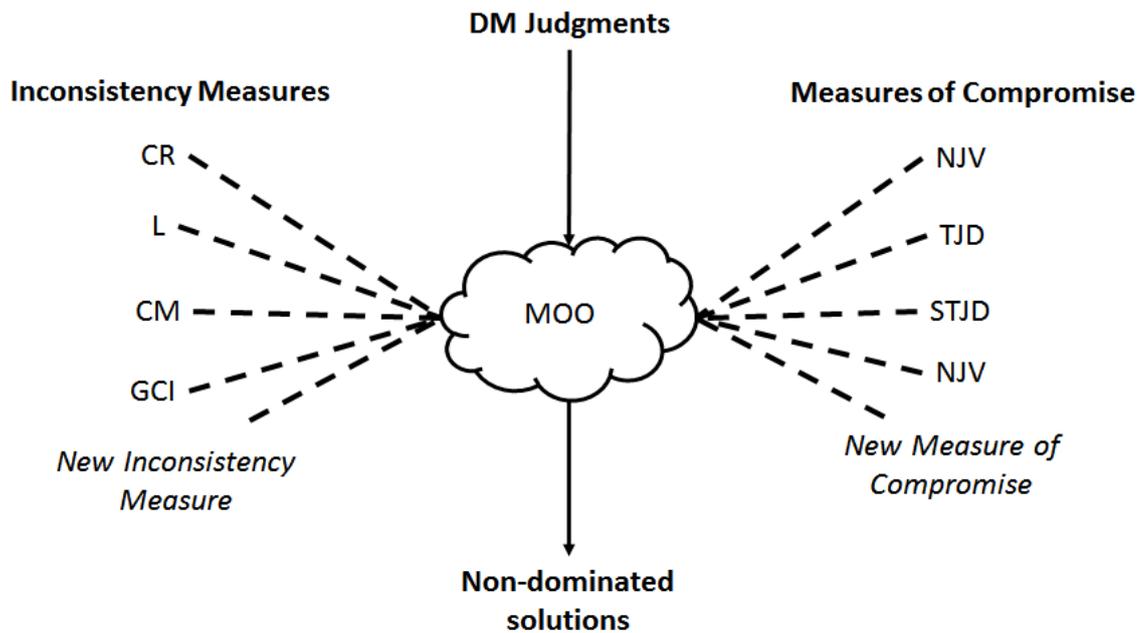


Figure 4.4: MOO objectives

#### 4.2.4 Implementation Analysis

This section evaluates the use of Multi-Objective Genetic Algorithm (MOGA)s to implement the MOO approach to inconsistency reduction. After a brief overview of genetic algorithms, we discuss finding the true solution set to multi-objective problems. This is followed by experiments to evaluate the viability of MOGAs as an implementation strategy.

##### 4.2.4.1 Multi-Objective Genetic Algorithms (MOGA)

For many real-world operational research problems Evolutionary Computing (EC) approaches can be used to swiftly arrive at a high quality approximation of the solution [111]. EC approaches make use of meta-heuristics to dynamically evolve a solution to a problem. Genetic Algorithms (GA) take inspiration from the natural biological world utilizing the notions of natural selection and survival of the fittest as a means of solving optimisation problems. Computational techniques to exploit these concepts were first proposed in 1975 [112] and then first realised by Goldberg [113] within the framework of a GA. GAs facilitate simultaneous searching of a wide area of a cost surface and their population approach helps deal with complex cost surfaces. Additionally a GA can handle with ease both continuous and discrete objective variables.

The common stages and operations of a (single objective) GA are shown in Figure 4.5. An initial population of individuals is created which is then evolved over many

generations. The Cost function determines how individuals will be evaluated and distinguished with regards to their utility. Each individual's fitness is used to help determine the likelihood individuals will be selected to create new offspring for the next generation. Various methods of selection may be utilised, see [114] for discussion and analysis of selection schemes. With the individuals for mating selected, the mating (crossover) process then takes place for the creation of new offspring individuals; see [115] for more details on crossover schemes. Additionally Mutation, adding random mutations to individuals, is performed to stimulate diversity within the population. Over many generations these individuals evolve towards a solution to the problem.

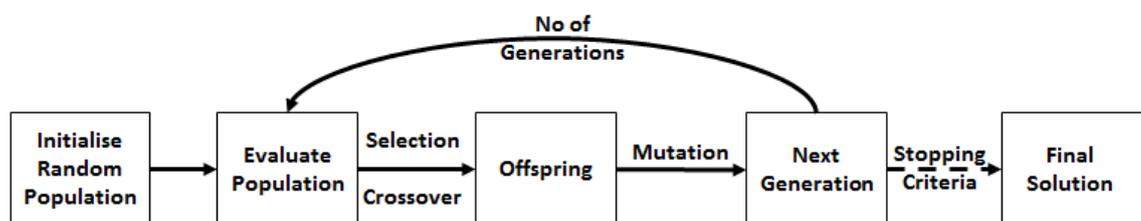


Figure 4.5: GA general stages

For multiple objective problems a MOGA can be utilised. A MOGA seeks to find a set of solutions that are both as close to the true Pareto front of the problem as possible, as well as being as evenly spread out along the front as possible.

Early MOGAs include the Vector Evaluated Genetic Algorithm (VEGA) [116], an approach termed Multi Objective Genetic Algorithm (MOGA) [117], and the Niche Pareto Genetic Algorithm (NPGA) [118]. The performance of these early naïve approaches has been superseded by more sophisticated recent MOGAs.

More recent MOGAs look to both effectively explore the solution space but also ensure retention of the best found solutions so far utilizing the concept of Elitism. This can be achieved via carrying over superior parents into the next generation as utilised in The Non-dominated Sorting Genetic Algorithm II (NSGAI) proposed in [119]. NSGAI seeks to foster the evolution of a population that gravitates towards the Pareto front of the problem using Pareto Front Ranking, see [120]. Superior members of the previous generation are carried over into the next generation ensuring the best individuals are retained. The final solution set of (only) non-dominated solutions is presented to the DM. However there is no way to control the number of final solutions that are presented.

Alternatively elitism can be achieved via the use of an external population in the form of an Archive [121] that can retain the best solutions found so far as well as allow the

main population to concentrate upon exploring the solution space. Furthermore an archive can provide additional flexibility by allowing a DM control over the size of solution set presented to the DM. The Strength Pareto Evolutionary Algorithm 2 (SPEA2) proposed in [122] utilises an external archive whose solutions can be presented to the DM as the final set of solutions. At each generation the population's individuals are evaluated with respect to Pareto dominance by calculating the raw fitness value of each individual, see [122]. The higher the fitness value, the higher the probability that an individual will be selected as a parent of the next generation's offspring. At each generation the archive is filled with the best raw fitness individuals, so may contain non-dominated and dominated solutions (when less non-dominated solutions than the archives size are found). This gives control of the exact number of solutions that will be returned, however the final archive may contain both non-dominated and dominated solutions.

Therefore alternatively a MOGA with an archive which only contains non-dominated solutions such as the Multi-Objective Cellular Algorithm (MOCeII) [123] would allow only non-dominated solutions to be presented to a DM as well as give the DM some control over the number that are returned. In MOCeII the population is structured into a two-dimensional grid and individuals are only permitted to mate with those individuals close to them in the grid, this way imposing restrictive mating. Created offspring that dominate a parent replace the parent in its position in the grid [123]. Furthermore after each generation a defined number of solutions from the archive are added to random positions in the population replacing the individuals in those locations. MOCeII gives a DM control of the maximum number of solutions that may be returned, all of which will be non-dominated

Other recent MOGAs include the Archive-based hYbrid Scatter Search Algorithm (AbYSS) [124] that utilises principles from both NSGAII SPEA2, and scatter search. The large number of parameters of AbYSS can make fine-tuning the algorithm to the problem at hand a complex task. Pareto Envelope Selection Algorithm 2 (PESA2) [125] utilises region-based selection and employs adaptive grid crowding procedure to ensure a spread of solutions, see [126]. An archive is additionally employed and at each generation the population is replaced by individuals from the archive which dictates that the archive needs to be a significant size in comparison to the population [59], giving a DM less control over the size of the archive. The Indicator-based Evolutionary Algorithm (IBEA) [127] looks to give a DM more control over its operation through using DM-specified indicators to determine selection precedence between solutions, integrating DM

preference information into the operation. However, without making use of an archive the number of solutions presented to the DM is not controllable.

To highlight some points from this discussion within the implementation experimentation that follows, the NSGAI, SPEA2 and MOCell algorithms will be evaluated.

#### 4.2.4.2 True Solution Sets

For multi-objective problems with discrete input variable ranges there exists a true Pareto front of non-dominated solutions. This true front can be found via an exhaustive (Brute Force) search of every possible sequence of input variables to the problem. Finding a true front of a problem can help to facilitate evaluation of the quality of solution sets found from heuristic optimisation approaches such as MOGAs. For the problem of altering a set of judgments looking to reduce the inconsistency within the judgments the order of the judgments is significant, each judgment can take any value from the scale and multiple judgments can have the same value. Therefore we think of the problem as a permutation with replacement problem. If we consider the 1-9 scale (which has 17 possible values for each judgment) then the total number of possible solution sequences for a problem with  $J$  judgments is.

$$S = 17^J \quad (4.7)$$

Therefore for such an NP-hard problem it will quickly become unwieldy and computationally expensive as the number of elements increase, as shown in Table 4.1.

Table 4.1: Exhaustive search sequence complexities

<b>N</b>	<b>J</b>	<b>Sequence Count</b>	<b>Standard Form</b>
3	3	4913	$4913 \times 10^3$
4	6	24137569	$2.41 \times 10^7$
5	10	2015993900449	$2.01 \times 10^{12}$
6	15	2862423051509815793	$2.86 \times 10^{18}$
7	21	69091933913008732880827217	$6.91 \times 10^{25}$

However we can calculate the true fronts of problems with smaller values of N to analyse how successful MOGAs might be at finding an approximation of the true fronts. To help

to explore the suitability of MOGAs to implement the approach to inconsistency reduction via MOO we have implemented an exhaustive search which we can compare with solution sets found via MOGAs. To evaluate solution sets found via MOGAs various evaluation measures have been proposed and are briefly outlined next.

#### 4.2.4.3 MOGA evaluation measures

Unlike a single objective optimisation problem, where different algorithm solutions can be compared with respect to how close to the global solution each has reached, the evaluation of a set of non-dominated solutions found from a MOGA is less straightforward. Various measures have been proposed to evaluate the quality of solution sets found.

The true Pareto front of non-dominated solutions can be utilised to see how well a stochastic approximation of the front created via a MOGA has performed. To help illustrate the evaluation measures let us take a two-objective minimisation problem which has a true Pareto Front of six non-dominated solutions. As an example, we compare four solutions that have been found from an approximation method against the six solutions of the true front. This true front and the found front are shown in Figure 4.6.

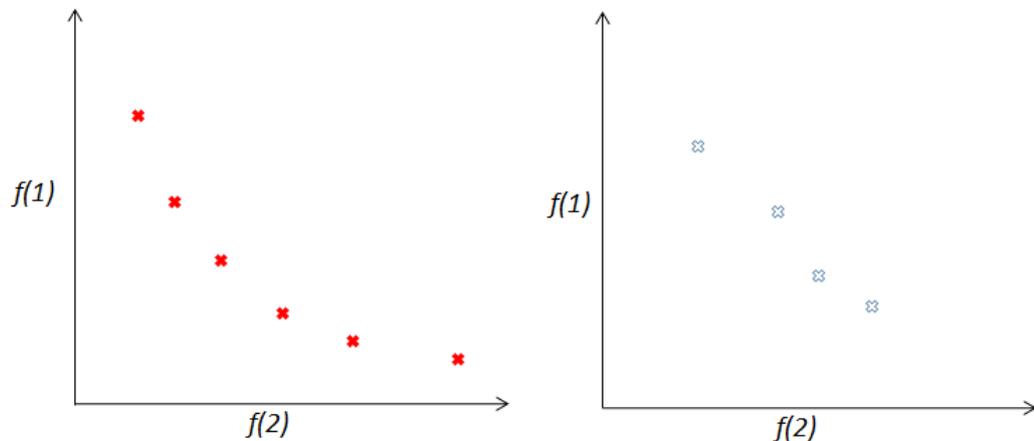


Figure 4.6 Left: Example true front Right: Example found front

We can measure the Generational Distance (GD) [128] of the found solution set – which is a measure of the distance between each found solution and the nearest true solution. Alternatively we can utilise the Inverted Generation Distance (IGD) [128] which is a measure of the distance between each true solution and the nearest found solution. Evidently the IGD will penalise more a solution set that fails to locate enough of the true front of solutions so is considered a measure of both Pareto dominance with

some consideration of spread of solutions. GD is considered a measure of Pareto dominance only, as a solution set that only finds a single solution that lies on the true front of solutions will have a GD of 0. Both measures are computationally efficient even for a large number of objective problems. The Spread metric [119], is a diversity measure of how well a problem's front has been mapped out via a measure of how evenly spread an approximation set is with additional consideration of how close to the edges of the front a solution set has discovered, see [119]. These three measures for the synthetic true front and solution set are shown in Figure 4.7.

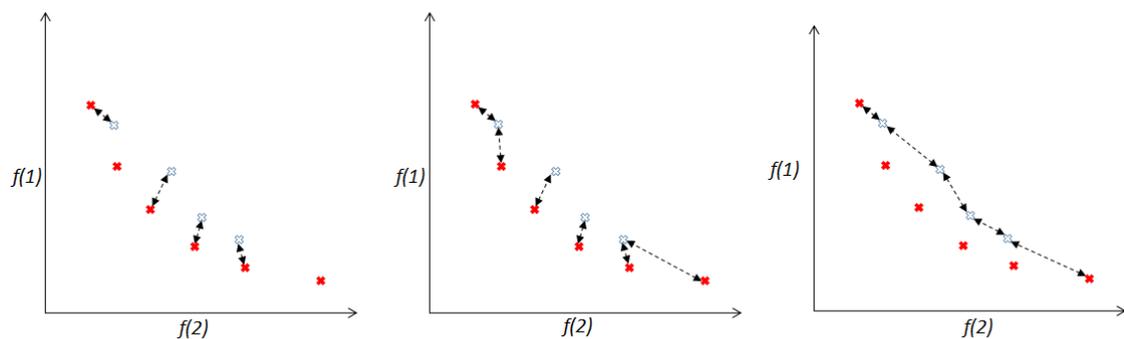


Figure 4.7 Left: GD, Centre: IGD, Right: Spread

An alternative measure is the HyperVolume (HV) measure [121], a more complete measure of both Pareto dominance and spread. Using the edges of the true front of solutions a perpendicular reference point is determined, then the unions of the hypercube areas that each solution covers is calculated as a volume. This way we measure the percentage of the volume of the true front volume that a solution set has found<sup>4</sup>. Hypervolume is a useful measure of both pareto dominance and spread however as its calculation is NP-complete [129] it becomes a time-consuming evaluation measure for problems with large numbers of objectives. Hypervolume can also be used as a measure of convergence speed. Given that a true front represents 100% HV coverage, we can assess the percentage of HV at regular intervals of a MOGA's operation. Then we can determine how many generations or evaluations it takes the MOGA to reach close to optimal solutions, for example 98% hypervolume. MOGA can then be evaluated with respect to how quickly they can converge to a close to optimal convergence of the front

<sup>4</sup> Within the experimentation as the problems are dealing with minimisation objectives inversion of the fronts is performed. Additionally as objectives of different measurement scales are utilised normalisation of fronts is performed to standardise the objectives ranges.

of the problem [130]. The visualisation of the hypervolume measure for the synthetic true front and solution set are shown in Figure 4.8.

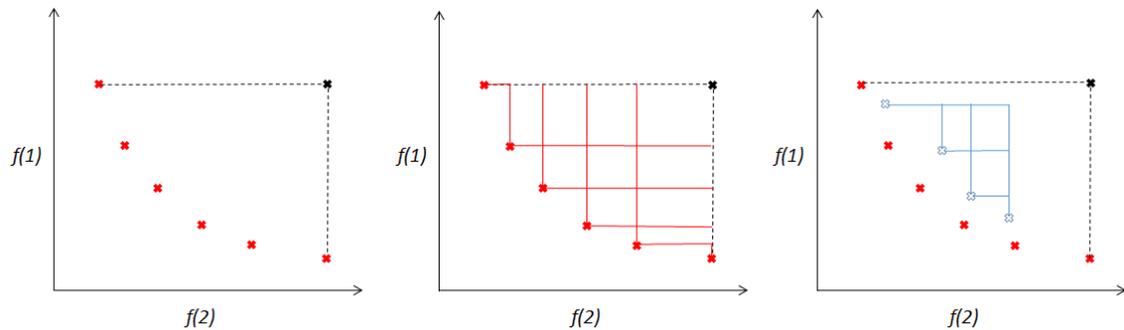


Figure 4.8 : Example hypervolume calculation

We will utilise these measures to evaluate solution sets found via MOGAs against true fronts for a problem. These evaluation measures will be used to look next at an  $N=3$  problem.

#### 4.2.4.4 $N=3$ Exhaustive Search Evaluation

Given an  $N=3$  problem we can elicit judgments from a DM as shown as a DAG in Figure 4.9.

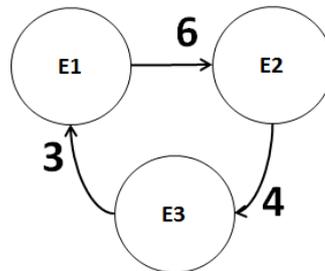


Figure 4.9:  $N=3$  exhaustive search example

With objectives of STJD and CR, we can utilise the exhaustive search implementation to find the true front of solutions for this 2-objective set. From this we find there are 24 solutions that make up the true front, the objective space of which is shown in Figure 4.10, (with Saaty's CR 0.1 threshold shown via a dotted vertical line). We can then evaluate the performance of MOGAs through measuring the Hyper Volume Percentage (HV%), Generational Distance (GD) and Inverted Generational Distance (IGD) of the found solution sets. The evaluation measures averaged over 10 runs for NSAGII, SPEA2 and MOCcell are shown in Table 4.3. The full parameters used for each MOGA are shown in Table 4.2.

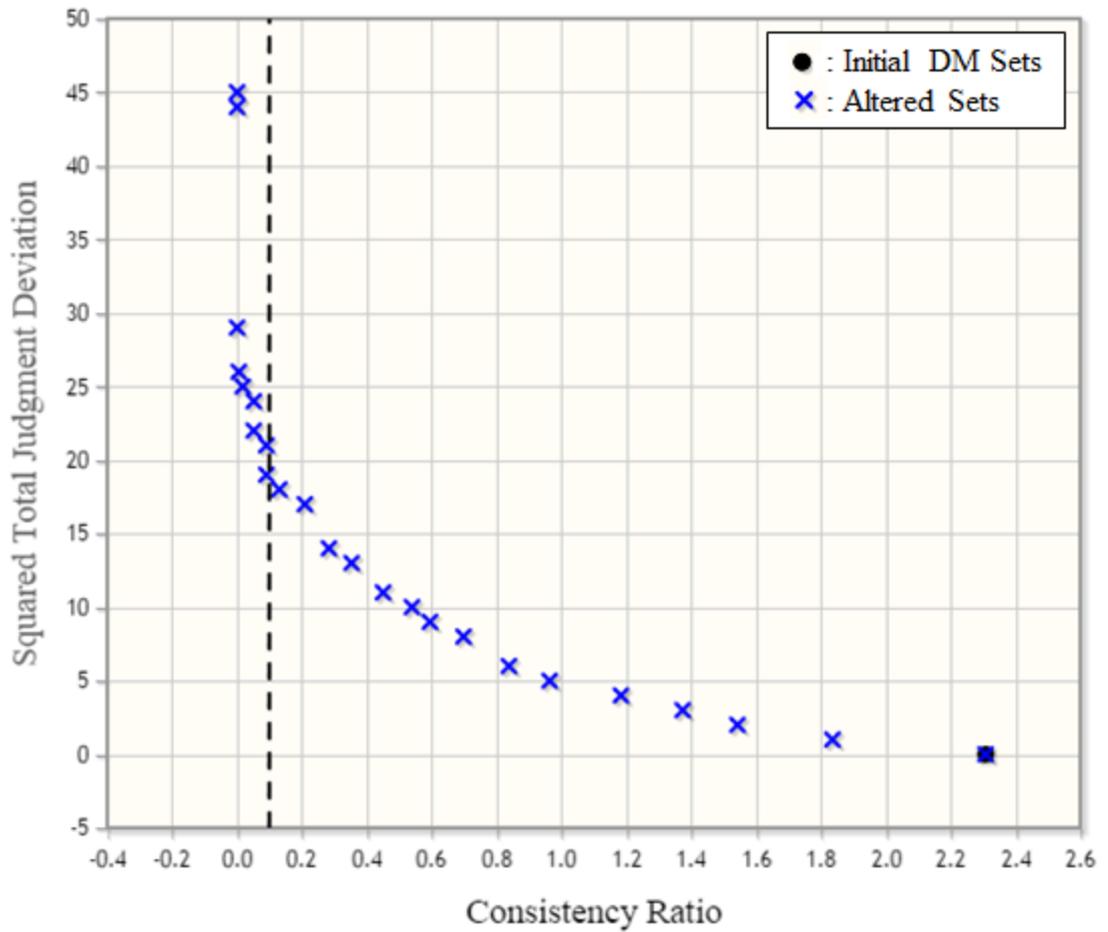


Figure 4.10: N=3 exhaustive search example true front

Table 4.2 : MOGA Setup parameters

	<b>NSGAI</b>	<b>SPEA2</b>	<b>MOCe</b>
<b>Population size</b>	100	100	100
<b>No. of evaluations</b>	5000	5000	5000
<b>Selection</b>	Binary-Tournament	Binary-Tournament	Binary-Tournament
<b>Crossover</b>	single point	single point	single point
<b>Crossover probability</b>	0.9	0.9	0.9
<b>Mutation</b>	Bit flip	Bit flip	Bit flip
<b>Mutation probability</b>	0.01	0.01	0.01
<b>Archive size</b>	-	25	25
<b>Neighbourhood size</b>	-	-	8
<b>Feedback</b>	-	-	10

We see that on average all 3 of the MOGAs are able to find solution sets that are very close to the true front in terms of the HV% as well as having low GD and IGD values.

Table 4.3: N=3 Example: MOGA Evaluation

MOGA	Evaluations	HV %	GD	IGD
NSGAI	5'000	99%	0.25	0.32
SPEA2	5'000	99%	0.28	0.43
MOCeII	5'000	99%	0.27	0.40

Here 5k evaluations were performed for each MOGA; we can additionally analyse performance and stability over a range of evaluations sizes - thus obtaining an idea of the average rate of increase of quality and stability of a MOGA. Figure 4.11 shows analysis of the MOCeII MOGA across a range of evaluation values, from 100 up to 25k, showing the average HV% and the standard deviation average over 10 runs for each evaluation value.

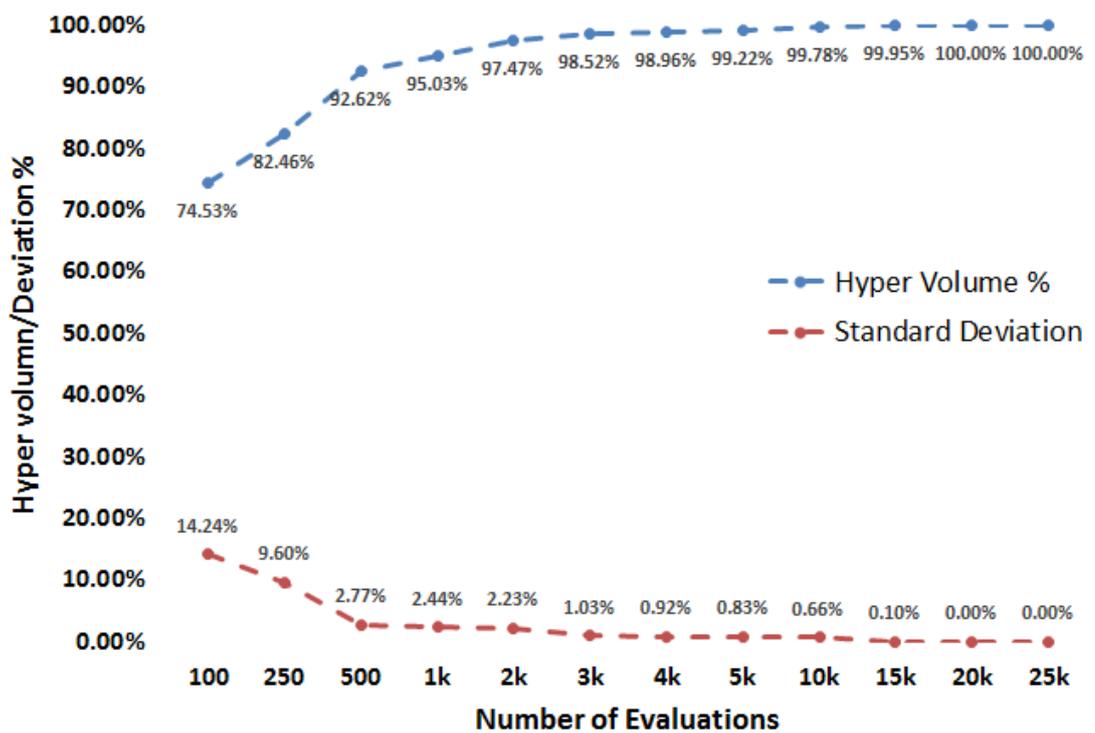


Figure 4.11: N=3 Example: Average Hypervolume % and stability

We see that MOCeII achieves a close to optimal solution (and one that is very stable) with a relatively small number of evaluations (around 3000). Furthermore we observe that for evaluations greater than 5k we see less improvement in performance and that it takes 20k evaluations before we see the true front perfectly found with 0 deviation over

10 runs. This highlights the power of MOGAs to rapidly find a close to optimal solution. This efficiency gets more pronounced with increasing complexity, we look next at an N=4 problem.

#### 4.2.4.5 N=4 Exhaustive Search Evaluation

Given an N=4 problem we can elicit judgments from a DM, shown as a DAG in Figure 4.12.

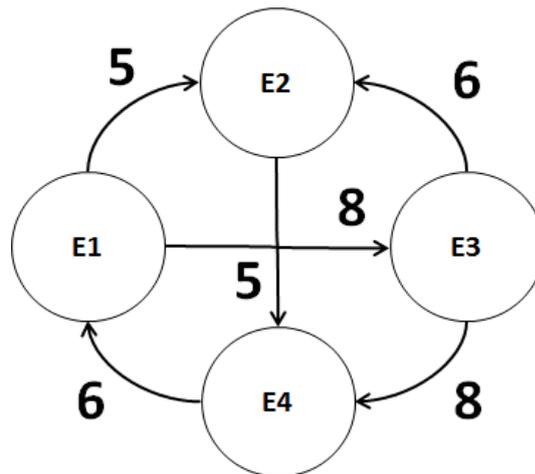


Figure 4.12: N=4 Exhaustive Search Example

With objectives of STJD and CR, we determine the true front of solutions to the problem via the exhaustive searching implementation. When N=4 over 24 million evaluations are required to determine the true front through exhaustive searching. From this we find that there are 123 solutions that make up the true front, the objective space of which is shown in Figure 4.14: Right. Visually analysing true fronts allows interesting observations regarding inconsistency reduction to be made. For example, here as with the last example, we see that for edge of the front with CR measure close to 0, inconsistency reduction requires a large increase in the compromise objective, here STJD. Analysis of the front by a DM in this way is useful to reveal, in this case, that seeking to completely remove inconsistency will be much more costly than seeking a near to 0 inconsistency solution.

We analyse solution sets found from MOGAs against the true front of solutions for the problem. Figure 4.9 shows the evaluation average measures over 10 runs for the NSGAI, SPEA2 and MOCell MOGAs, (using the parameters from Table 4.2 with the exceptions of an archive size of 125 set for SPEA2 and MOCell, and a population size of

125 for NSGAI). As with the previous example we see that all the algorithms have been able to find close to optimal approximations of the true front with each achieving an average hypervolume of over 98%.

Table 4.4: N=4 Example: MOCcell Evaluation

<b>MOGA</b>	<b>Evaluations</b>	<b>HV %</b>	<b>GD</b>	<b>IGD</b>
<b>NSGAI</b>	5'000	98.6%	0.113	0.280
<b>SPEA2</b>	5'000	98.8%	0.148	0.299
<b>MOCcell</b>	5'000	98.5%	0.124	0.313

When N=4 we have a much larger front of non-dominated solutions, 123, which will only increase for larger sizes of N. This has some practical implications upon the choice of MOGA. MOGAs without an archive, such as NSGAI, are restricted by their population size as to how many non-dominated solutions they can find, as well as being unable to control the number of solutions that are presented to a DM. Increasing the population size to try to alleviate this is impractical for higher values of N as it will increase the number of evaluations per generation. MOGAs that implement an archive can define a large archive size and still retain a smaller population size. However MOGAs such as SPEA2, whose archive strategy is to fill the archive after each generation (first non-dominated solutions found are added then the remaining space is filled by the least dominated solutions found), might result in non-dominated solutions being presented to a DM. Conversely using a MOGA such as MOCcell which employs an archive that only stores the non-dominated solutions found so far gives a DM control over the maximum number of solutions returned and ensures only non-dominated ones are returned. Therefore using MOCcell to further analyse this problem we again analyse performance over a range of evaluation values from 100 to 25k, the averages over 10 runs of which are shown in Figure 4.13. Again we see that after only a few thousand evaluations we achieve high performance and stable results, with performance increases tailing off with higher evaluation values. So we see that we can achieve an average of over 98% performance with only 5,000 evaluations, which is 0.02% of the number of the evaluations of an exhaustive search.

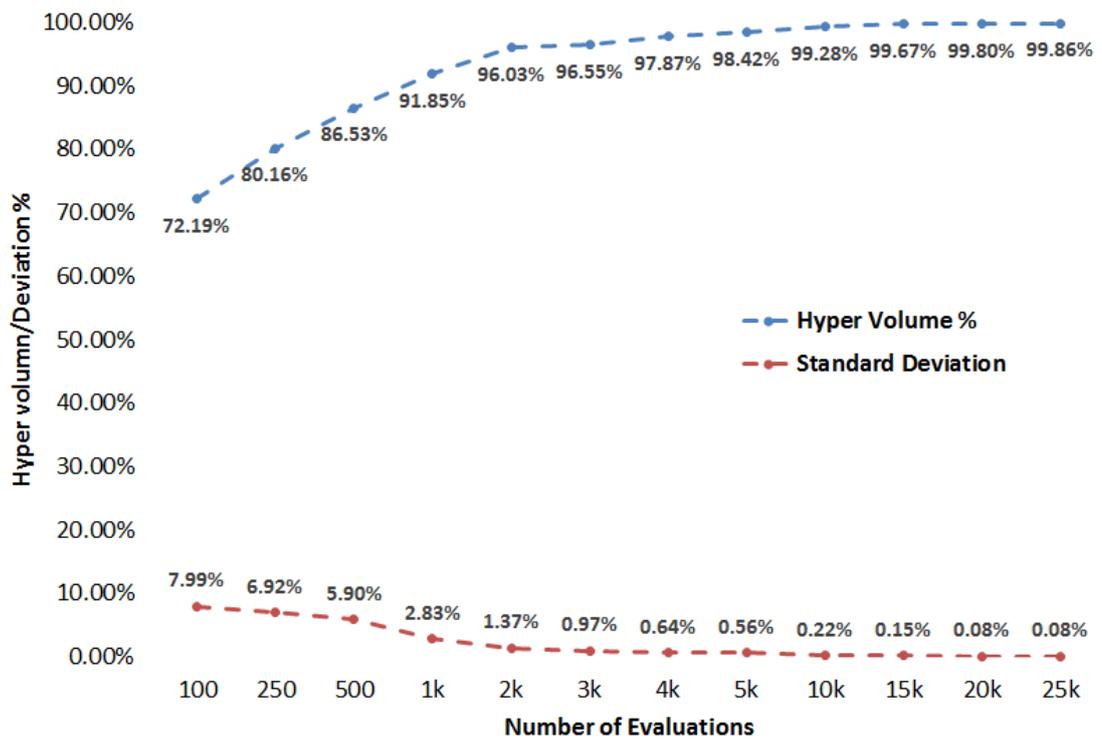


Figure 4.13: N=4 Example: Average Hypervolume % and stability

The results from a MOCeCell run for this problem are shown in of Figure 4.14: Left (with the large archive defined highlighting the front to show visually that the MOCeCell has achieved a close approximation of the true front).

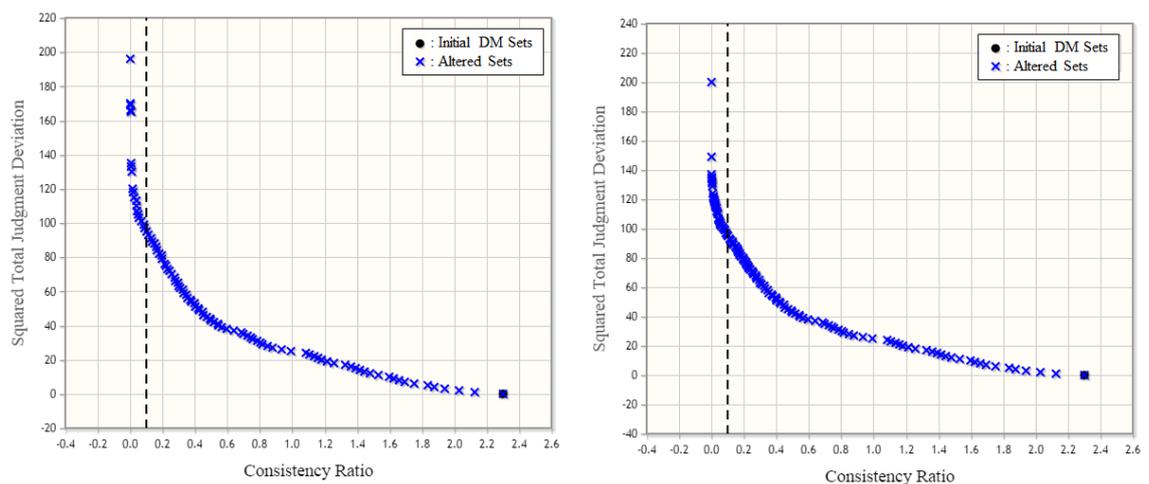


Figure 4.14 Left: N=4 Solution set found via MOCeCell. Right: N=4 true Front

#### 4.2.4.6 Implementation Analysis Conclusions

We have utilised exhaustive searching to analyse problems where the number of elements is 3 or 4. However when N=5 (and J is 10) the exhaustive search complexity jumps to

17<sup>10</sup>=2015993900449 - which would result in taking 83521 times longer than an N=4 problem (if an N=4 problem took 1 hour to run then an N=5 problem would take over 9 years to complete). However from the experimentations we have seen that MOGAs are a viable implementation strategy that will allow us to find a close to optimal solution in a fraction of the time of exhaustive searching. For an N=5 problem using a MOGA with 25k evaluations would require only 0.00000124% of the evaluations of an exhaustive search and this efficiency becomes even more pronounced for larger values of N. For the choice of MOGA the experimentation showed all the MOGAs tested achieve high performance and all are usable within the approach. However as discussed, MOCeLL has advantages for a DM to control the maximum number of solutions returned via its archive and ensuring only non-dominated solutions are presented to a DM. Therefore MOCeLL will be utilised within the examples presented later. Next constraints and how they are implemented within the approach are discussed.

#### **4.2.5 Constraints**

For problems such as reducing inconsistency within a set of judgments as well as seeking to find the set of trade-off solutions, additional consideration needs to be given to ultimately help towards selecting a single solution from the set of non-dominated solutions [131]. Support towards aiding the selection of a single solution may be incorporated before the search. Such approaches generally attach weights to the objectives before the search process, see [132]. These strategies assume that such preferences are known and clear from the start of the problem, which is rarely the case. Through implementation of the problem via MOO the approach makes no assumptions about weights of objectives before the process.

Alternatively support towards the selection of a single solution may be incorporated in a more interactive way through utilizing constraints expressing a DM's tolerance levels to reduce the size of the objective space towards areas of interest. This could help facilitate the selection of a single solution through iteratively reducing the objective space size. Within MOGAs constraints can be tackled through various strategies such as discarding infeasible solutions, reducing the fitness of infeasible solutions or repairing infeasible solutions to be feasible, see [133] for a review. Discarding infeasible solutions could result in solutions of high quality Pareto dominance that are only just infeasible being lost. Strategies to repair infeasible solutions introduce added complexity regarding defining repair functions. Therefore in the approach we seek to implement constraints via reducing the fitness of infeasible solutions to push the population towards the feasible

region of the objective space, and in addition by ensuring that only feasible solutions are added to the archive. We consider any constraints for a problem first in the evaluation process to favour feasible individuals over infeasible solutions and penalise constraint violating solutions. The approach implements the Constrained Pareto Dominance [119] as defined for constraint handling in NSGAI. Here each constraint is considered during the selection stage utilizing the extra constraint information in the binary tournament selection operation. In binary tournament selection a pair of solutions are randomly selected and the fittest of the pair is selected to be a parent for the next generation's individuals. With additional constraints the pair of solutions are additionally evaluated to determine if they are feasible or not. If both individuals are feasible then their Pareto dominance is compared as normal, Figure 4.15: Left. If one solution is feasible and one infeasible then the feasible one is chosen, see Figure 4.15: Centre. If both solutions are infeasible then the solution that is the least infeasible is chosen, see Figure 4.15: Right. Thus feasible solutions are favoured over infeasible solutions pushing the population's individuals towards the feasible area of the Pareto front. The approach additionally implements a hard constraint upon the archive to only allow feasible solutions to be added to the archive, this way ensuring that only feasible solutions will be presented to the DM. This additionally enhances the feedback operation of MOCeII as only feasible solutions will be fed back from the archive into the population helping to further steer the population towards the feasible region of the objective space.

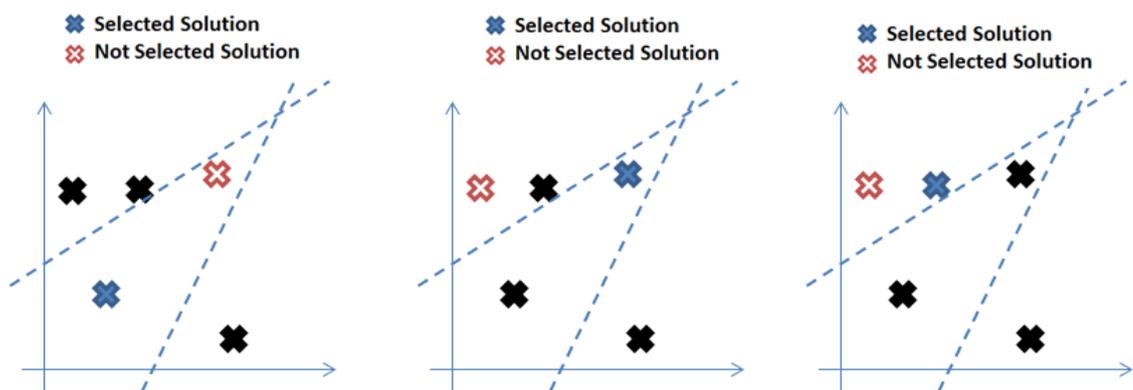


Figure 4.15: Constrained Pareto Dominance during 2-Objective minimisation

In the approach constraints can be utilised in an interactive iterative manner. An initial search of the objective space can reveal to the DM the nature of the objective space regarding inconsistency reduction for their chosen objectives. Informed by the knowledge of the objective space a DM can then set feasible (and achievable) constraints to the

problem. Constraints can then be iteratively added to drill down into the objective space<sup>5</sup> to aid the DM in the selection of a single solution.

#### **4.2.6 Approach to the reduction of the inconsistency of a set of PCs Discussions**

We have outlined the rationale for the approach to the reduction of the inconsistency of a set of judgments and outlined the stages of the approach. The MOO definition of the problem has been presented along with the objectives usable in the approach. Through comparisons against true fronts of problems the suitability of MOGAs has been shown along with discussions of different MOGA's merits. Finally how constraints are implemented within the approach has been discussed. In the next section the approach and its benefits are explored through examples and comparisons to other approaches.

### **4.3 Experimentation Examples**

In this section, step-by-step examples of the approach are presented.

1. Example 4.1 explores a PCM taken from [81] and compares the approach to that of other approaches for inconsistency reduction;
2. Example 4.2 takes a PCM with high levels of both cardinal and ordinal inconsistency and explores how the approach is flexible to allow a DM to use different inconsistency measures to suit their preferences;
3. Example 4.3 illustrates how a DM can iteratively add constraints to aid in selection of a single solution;
4. Example 4.4 explores using multiple inconsistency objectives simultaneously and using multiple measures of compromise simultaneously.

For these examples the MOCcell algorithm was utilised with the following parameter settings: population size of 100 (10 x 10 grid); maximum evaluations count of 25,000; Selection is performed via binary tournament with single point crossover (with crossover probability 0.9) and bit flip mutation (with probability 0.01) employed. The size of the

---

<sup>5</sup> Constraints can be added as part of a new search utilising the constraints with the operation of the MOO search however they can also be added to a set of found solutions to simply slice up the objective space of the solutions without re-searching if that meets the DM's preferences.

archive is definable by the DM and stated in each example (with the feedback value set to 25% of the size of the archive).

#### 4.3.1 Example 4.1: Comparison against other approaches

Example 4.1 uses an 8 element PCM, shown in Table 4.5, taken from [81] and used during previous approaches proposed by Xu and Wei [81] and by Cao et al. [82]. The initial CR is 0.17, thus greater than Saaty's 0.1 threshold of acceptance. Both approaches then look to derive an altered PCM solution which has CR value less than 0.1. We utilise their solutions for comparison against the MOO approach to inconsistency reduction for this problem.

Table 4.5: Example 4.1: [CR: 0.17]

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>1</b>	1	5	3	7	6	6	1/3	1/4
<b>2</b>	1/5	1	1/3	5	3	3	1/5	1/7
<b>3</b>	1/3	3	1	6	3	4	6	1/5
<b>4</b>	1/7	1/5	1/6	1	1/3	1/4	1/7	1/8
<b>5</b>	1/6	1/3	1/3	3	1	1/2	1/5	1/6
<b>6</b>	1/6	1/3	1/4	4	2	1	1/5	1/6
<b>7</b>	3	5	1/6	7	5	5	1	1/2
<b>8</b>	4	7	5	8	6	6	2	1

Tackling this problem with the MOO approach given DM-chosen objectives of CR and TJD and an archive size of 10, Figure 4.16 shows the objective space of solutions found. The CR threshold of 0.1 is shown via a dashed vertical line.

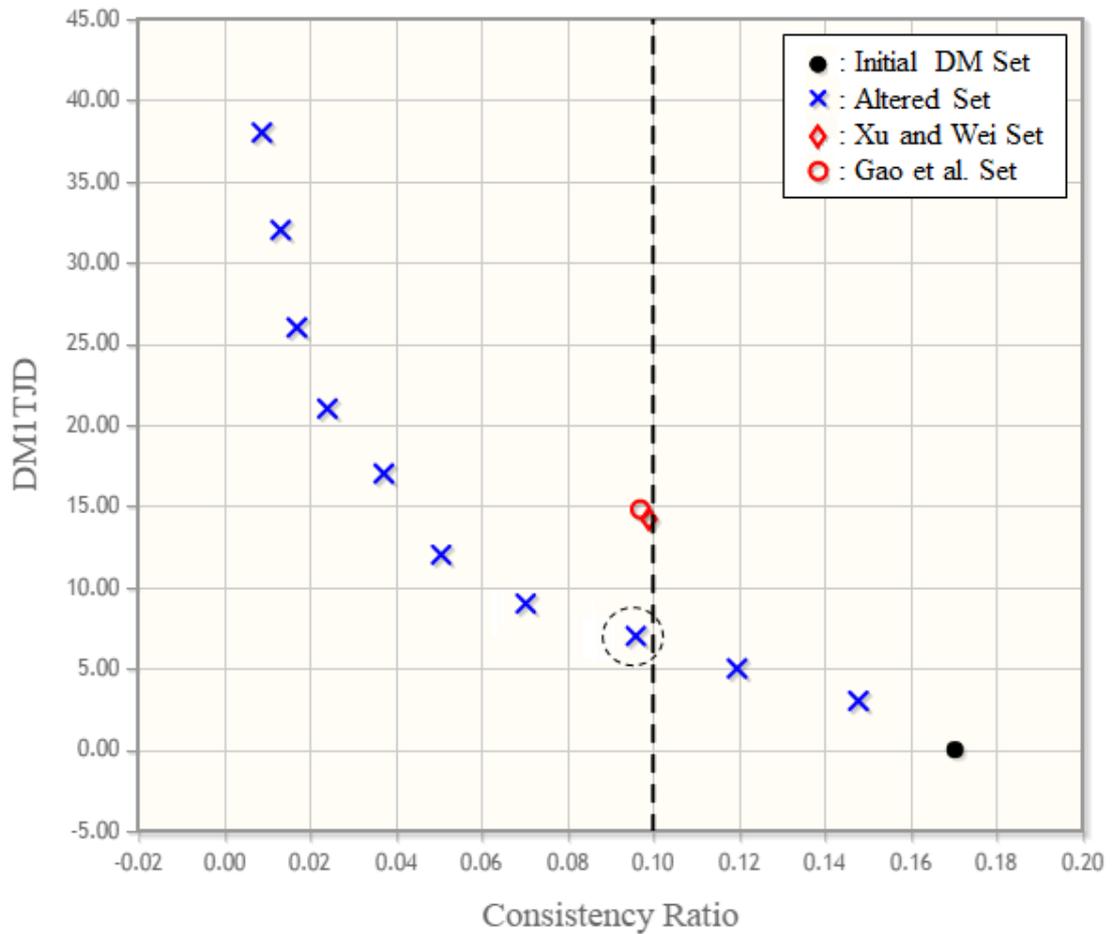


Figure 4.16: Example 4.1 Solution Space

The DM is then free to review and select any of the 10 solutions found. For instance, the DM could select the first solution along the Pareto front with a CR value less than 0.1, identified via a dotted circle in Figure 4.16, the PCM of which is shown in Table 4.6. From this a DM has a solution with CR less than 0.1 and a meaningful measure of the amount of alteration to reach this – from the TJD value of 7, the DM sees that 7 scale steps of compromise occurred to find this solution. Additionally in Figure 4.16 we have plotted the solutions found for this problem from the Xu and Wei, and Cao et al. approaches, the PCMs of which are shown in Table 4.7 and Table 4.8 respectively. We see in Figure 4.16 that both of these solutions are dominated by solutions found via the MOO approach with respect to the amount of deviation the original judgments have undergone. Additionally from these solutions we see it will be more difficult for a DM to discern how their judgments have changed as the solutions contain values outside of the originally used scale. We calculate a deviation measure for these solutions based upon the fractional amount of scale steps that each modified judgment has undergone, the sum of which is used to plot these solutions within the objective space in Figure 4.16. For example, taking the judgment between elements 3 and 7, for the Xu and Wei solution in

Table 4.5 and Table 4.7, we see the judgment of 6 has changed to a judgment of 4.155, which in terms of scale steps we can calculate as 1.845 steps. Similarly for fractional judgments we can determine the deviation again as the amount of scale steps that occur. For example, taking the judgment between elements 3 and 8, for the Xu and Wei solution from Table 4.5 and Table 4.7 we see the judgment of 1/5 (0.2) has altered to 0.249 which in terms of scale steps represents 0.976 steps. These example deviation calculations are shown visually in Figure 4.17. We can calculate the total deviation of these 2 approach's solutions as 14.219 for Xu and Wei and 15.63 for Cao et al, both of which are greater amounts of deviation than the solution selected from the MOO approach in Table 4.6.

Table 4.6: Example 4.1 possible solution [CR: 0.098 TJD: 7]

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>1</b>	1	5	3	7	6	6	1/2	1/2
<b>2</b>	1/5	1	1/3	5	2	3	1/5	1/7
<b>3</b>	1/3	3	1	6	3	4	3	1/5
<b>4</b>	1/7	1/5	1/6	1	1/3	1/4	1/7	1/8
<b>5</b>	1/6	1/2	1/3	3	1	1/2	1/5	1/6
<b>6</b>	1/6	1/3	1/4	4	2	1	1/5	1/6
<b>7</b>	2	5	1/3	7	5	5	1	1/2
<b>8</b>	2	7	5	8	6	6	2	1

Table 4.7: Example 4.1 solution from Xu and Wei [81] [CR: 0.097 TJD: 14.219]

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>1</b>	1	4.524	2.339	7.523	5.888	5.686	0.425	0.292
<b>2</b>	0.221	1	0.326	4.516	2.671	2.580	0.222	0.147
<b>3</b>	0.427	3.067	1	6.749	3.460	4.188	4.155	0.249
<b>4</b>	0.133	0.221	0.148	1	0.373	0.287	0.134	0.104
<b>5</b>	0.170	0.374	0.289	2.681	1	0.561	0.197	0.147
<b>6</b>	0.176	0.388	0.234	3.479	1.784	1	0.204	0.153
<b>7</b>	2.354	4.497	0.241	7.479	5.073	4.899	1	0.501
<b>8</b>	3.419	6.786	4.024	9.624	6.783	6.551	1.996	1

Table 4.8: Example 4.1 solution from Cao et al [82] [CR: 0.099 TJD:15.63]

	1	2	3	4	5	6	7	8
1	1	4.4412	2.3682	7.6743	5.8559	5.6079	0.4201	0.2968
2	0.2252	1	0.3210	4.4224	2.6175	2.5392	0.2268	0.1486
3	0.4223	3.1151	1	6.9149	3.5351	4.2774	4.5105	0.2487
4	0.1303	0.2261	0.1446	1	0.3805	0.2927	0.1313	0.1030
5	0.1708	0.3820	0.2829	2.6278	1	0.5722	0.1960	0.1444
6	0.1783	0.3938	0.2338	3.4166	1.7478	1	0.2055	0.1494
7	2.3804	4.4099	0.2217	7.6136	5.1012	4.8661	1	0.5004
8	3.3689	6.7293	4.0215	9.7130	6.9235	6.6949	1.9984	1

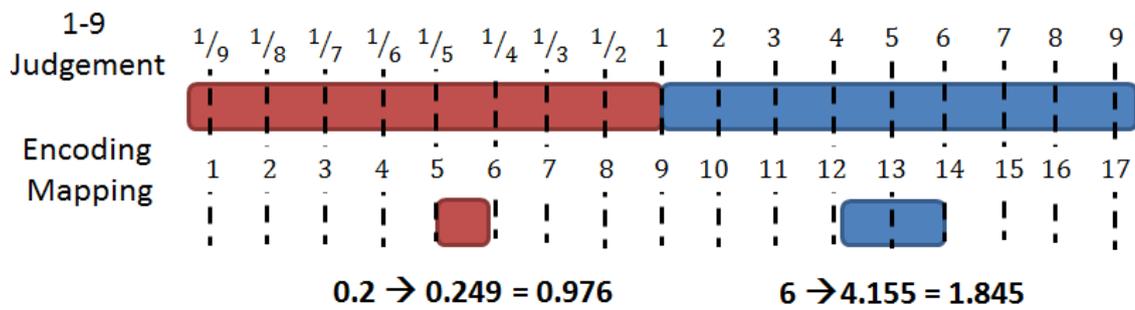


Figure 4.17: Example deviation calculation outside whole scale step

### 4.3.2 Example 4.2: Inconsistency Measures

Next we present an example with high initial levels of cardinal and ordinal inconsistency and illustrate how different inconsistency measures of a DM's choosing can be utilised within the approach. A PCM for a 9 element problem is shown in Table 4.9, the initial inconsistency measure are CR: 0.76, L: 9, CM: 0.99 and GCI: 50.8.

To seek inconsistency reduction a DM can choose an inconsistency measure of their preference. For example, if the DM seeks to reduce cardinal inconsistency and wishes to utilise the CR measure they can select CR as an objective. Given further that the DM chooses STJD as a measure of compromise objective and an archive of size 10, the solution space is shown in Figure 4.18: Left. From this we see that a large amount of alteration is required to find a solution which is below the CR threshold of 0.1

Table 4.9: Example 4.2 Judgments

	1	2	3	4	5	6	7	8	9
1	1	1/8	1/3	1/7	1/3	1	8	1/9	1/4
2	8	1	5	1/2	1/3	4	3	7	5
3	3	1/5	1	2	1/2	1/6	7	7	1/9
4	7	2	1/2	1	1	5	2	2	1/9
5	3	3	2	1	1	7	6	5	6
6	1	1/4	6	1/5	1/7	1	2	1/6	1
7	1/8	1/3	1/7	1/2	1/6	1/2	1	1	8
8	9	1/7	1/7	1/2	1/5	6	1	1	1/8
9	4	1/5	9	9	1/6	1	1/8	8	1

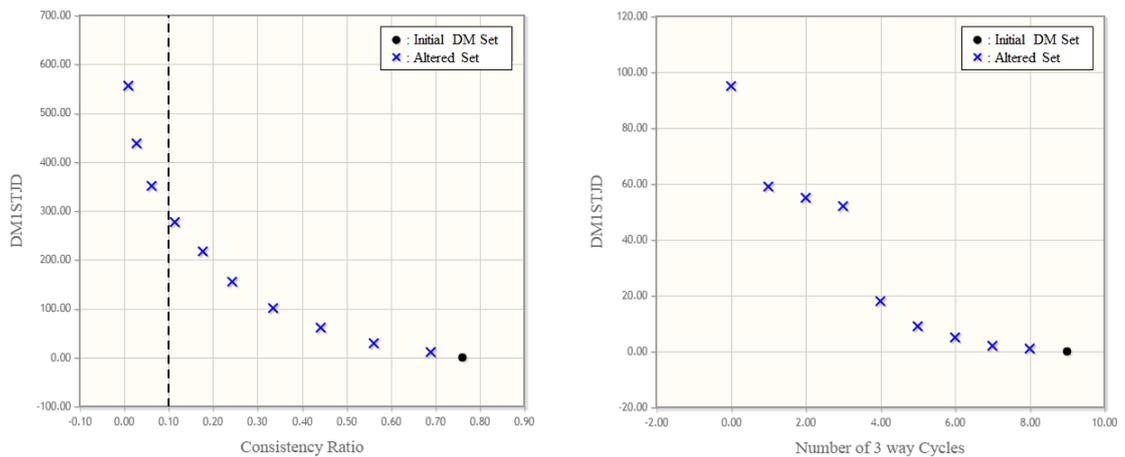


Figure 4.18: Example 4.2 Left: CR and STJD objectives. Right: L and STJD objectives

Alternatively if a DM seeks to reduce the number of cycles within their judgments, thus being more interested in ordinal inconsistency reduction, then they can instead choose L as an inconsistency objective. Given again that the DM chooses STJD as a measure of compromise objective and an archive size of 10 the solution space found is shown in Figure 4.18: Right. From this we see the shape of the objective space across the range of values of L for the minimal amount of STJD. We observe a large jump in the amount of alteration to reduce the number of cycles from 4 to 3 and again to reduce the number of cycles from 1 to 0. Such analysis of the objective space can help a DM to understand the nature of the inconsistency measure for their judgments and make an informed choice regarding reducing inconsistency.

Conversely if a DM is concerned with reducing inconsistency via looking to reduce the largest inconsistent judgment triple then the CM can instead be chosen as an inconsistency objective. Given again that the DM chooses STJD as a measure of compromise objective and an archive size of 10, the solution space found is shown in Figure 4.19: Left. From this objective space we observe the convex nature of the front for this objective pair<sup>6</sup>. We observe that there is little reduction in CM towards the edge of the initial judgments yet, at the other edge of the front larger decreases in CM are achieved for lower amounts of compromise.

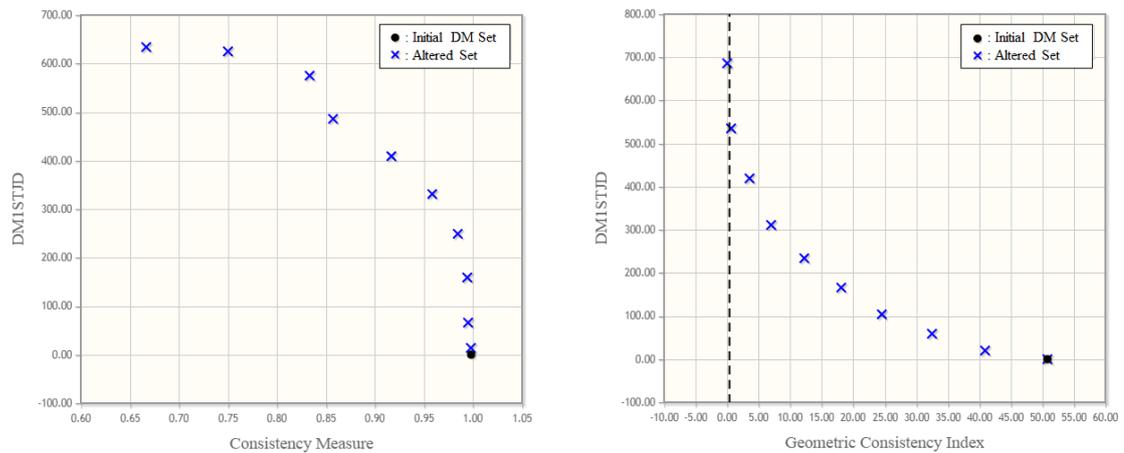


Figure 4.19 Example 4.2 Left: CM and STJD objectives. Right: GCI and STJD objectives

Alternatively if a DM instead wishes to utilise a distance-based inconsistency measure of the distance between judgments and a derived preference vector, then the GCI can instead be chosen as the inconsistency measure. The solution space with STJD as measure of compromise objective and an archive size of 10 is shown in Figure 4.19: Right. Additionally plotted as a vertical dotted line is the GCI threshold measure [72] (0.37 when  $N > 5$ ). From this plot and Figure 4.18: Left we see the strong relationship between CR and GCI as demonstrated in [72].

This example shows how the approach is versatile to the preferences of a DM regarding how inconsistency will be defined and measured in their judgments. Furthermore we see how the approach allows for knowledge of the nature of the trade-offs involved in the problem to be revealed to the DM.

<sup>6</sup> Additionally from this plot we observe that a solution with zero, or close to zero CM has not been found. Further investigations could analysis further the objective and its properties to examine this.

### 4.3.3 Example 4.3: Interactively adding constraints

Our third example explores how interactive analysis of an objective space through iteratively adding constraints can aid a DM in the selection of a single solution. DM judgments for a 5 element problem and a preference vector derived with the GM prioritization method are shown in Table 4.10 along with the initial CR inconsistency measure.

Table 4.10: Example 4.3 Judgments [CR: 1.08]

	1	2	3	4	5	$w$
1	1	8	6	1/2	1/5	0.25
2	1/8	1	7	2	7	0.30
3	1/6	1/7	1	5	1/2	0.10
4	2	1/2	1/5	1	1/2	0.12
5	5	1/7	2	2	1	0.23

Then given a DM chooses objectives of TJD and CR and an archive size of 10, the initial objective space is shown in Figure 4.20.

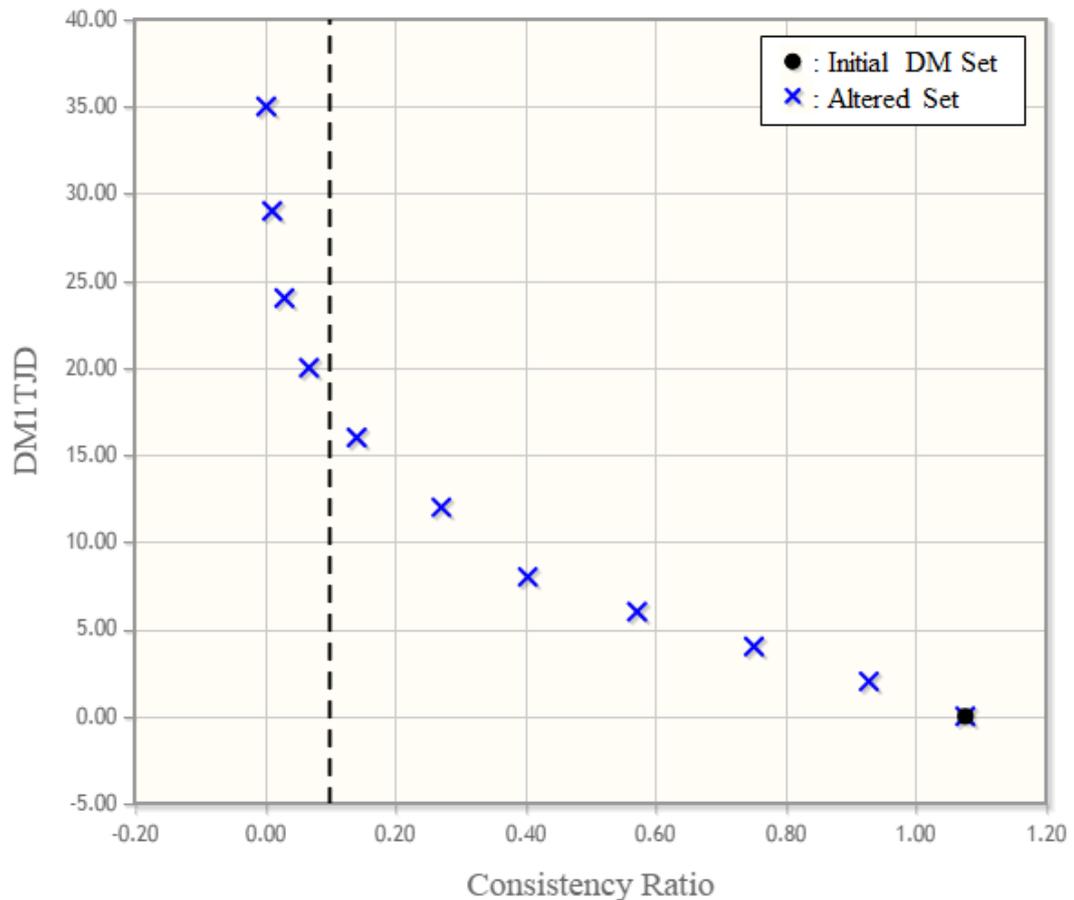


Figure 4.20: Example 4.3 Solution Space. CR and TJD objectives

From this the DM can get an overview of the objective space and the nature of the trade-offs between the objectives over the front for the problem, helping them to then add feasible constraints. For example, the DM might conjecture that it is feasible to seek a solution whose CR value is less than 0.1, so sets a constraint upon CR to only find solutions with CR of 0.1 or less. Additionally to focus upon this area they increase the archive size to a maximum of 20 and perform the search with these new parameters and constraint. The objective space with this added constraint is shown in Figure 4.21: Left with the constraint edge shown as a dotted red line. From this constrained objective space the DM might further conjecture that the amount of deviation increase past 20 then yields little further reduction in inconsistency. Therefore they could decide to add an additional constraint upon the upper amount of deviation to be less than 20 (so 19 or less). The new constrained objective space with this constraint added is shown in Figure 4.21: Right.

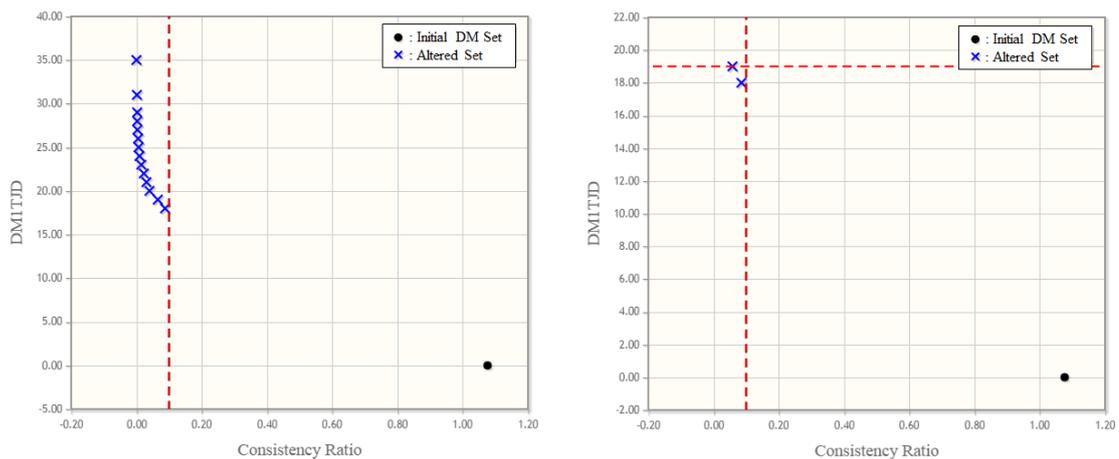


Figure 4.21 Left: Example 4.3 Objective Space with CR constraint. Right: Example 4.3 objective space with CR and TJD constraints

From this second constrained objective space the DM can observe there are only 2 solutions, with TJD values 18 and 19. The DM can then analyse these solutions, shown in Table 4.11, along with their values of the objectives and their preference vector rankings of the elements (here derived using the GM prioritization method). From analysing these solutions the DM can clearly see the compromise needed to achieve these solutions, and that both have over 90% reduction in initial inconsistency. Furthermore, with regard to the ordinal rankings of the preference vectors compared to the initial judgments preference vector only 1 change has occurred - between elements 3 and 4 - for both solutions.

Table 4.11: Example 4.3 Constrained objective space solutions

<b>TJD:18 CR:0.085</b>							<b>TJD:19 CR:0.058</b>						
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<i>w</i>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<i>w</i>
<b>1</b>	1	1	6	1	1	<i>0.24</i>	<b>1</b>	1	1	6	1	<b>2</b>	<i>0.27</i>
<b>2</b>	1	1	7	2	7	<i>0.41</i>	<b>2</b>	1	1	7	2	7	<i>0.40</i>
<b>3</b>	1/6	1/7	1	1/2	1/2	<i>0.06</i>	<b>3</b>	1/6	1/7	1	1/2	1/2	<i>0.06</i>
<b>4</b>	1	2	2	1	1	<i>0.16</i>	<b>4</b>	1	1/2	2	1	1	<i>0.16</i>
<b>5</b>	1	1/7	2	1	1	<i>0.13</i>	<b>5</b>	1/2	1/7	2	1	1	<i>0.11</i>

From analysis of these 2 solutions the DM may conjecture that for the TJD 19 solution the additional reduction in inconsistency is worth the additional deviation step. Therefore the solution with inconsistency of CR 0.058 and TJD 19 is chosen.

#### 4.3.4 Example 4.4: Larger objective sets

The approach is not constrained to using objective sets of size 2, in this example we illustrate how a DM can chose larger objective sets using the judgments from Example 4.3 in Table 4.10 to illustrate this. A DM could utilise multiple measures of compromise simultaneously, for example, they could choose to use 3 objectives of CR and TJD and also NJV. The approach will look for Altered Solutions with low CR values and with minimum deviation but also look to minimise the number of judgments that change. The 3-dimensional objective space for this 3-objective set with a large archive size defined to help emphasise the nature of the front is shown with respect to CR and NJV in Figure 4.22: Top Left. From this we observe a pattern within the objective space of multiple solutions with the same NJV value but various levels of CR; these are solutions representing increasing levels of deviation for the same amount of NJV. We see also that as CR tends towards 0 at one edge of the objective space the range of CR values for solutions with the same NJV value decreases. A DM could additionally perform analysis between the measures of compromise, for example the same 3-dimensional objective space shown this time with respect to TJD and NJV is shown in Figure 4.22: Top Right. From this we see more clearly the relationship between the measures of compromise for this set of judgments. We see that there is some relationship between the measures yet for higher amounts of compromise this relationship appears to weaken.

Furthermore a DM could utilise multiple measures of inconsistency simultaneously. For example, say a DM chooses 3 objectives of, CR and L inconsistency measures and the TJD measure of compromise. This time the MOO will look to reduce both cardinal and ordinal inconsistency simultaneously. The 3-dimensional objective space shown with respect to L and TJD is shown in Figure 4.22: Bottom Left. From this we see a large number of solutions which have all ordinal inconsistency removed but with a range of different deviation measures. Additionally a DM could perform analysis of the relationships of the inconsistency measures for this objective space as shown with respect to CR and L in Figure 4.22: Bottom Right. In this view of the objective space we see how the objectives both converge to 0 inconsistency solutions at one edge of the objective space. However we also observe the outline of 2 arcs of solutions from the initial judgment set edge of the front to a solution with all inconsistency removed edge, highlighted as dotted red arcs on the plot. This highlights the different emphases of objectives of cardinal and ordinal reduction and the importance of flexibility to allow a DM to decide upon how inconsistency reduction will be measured.

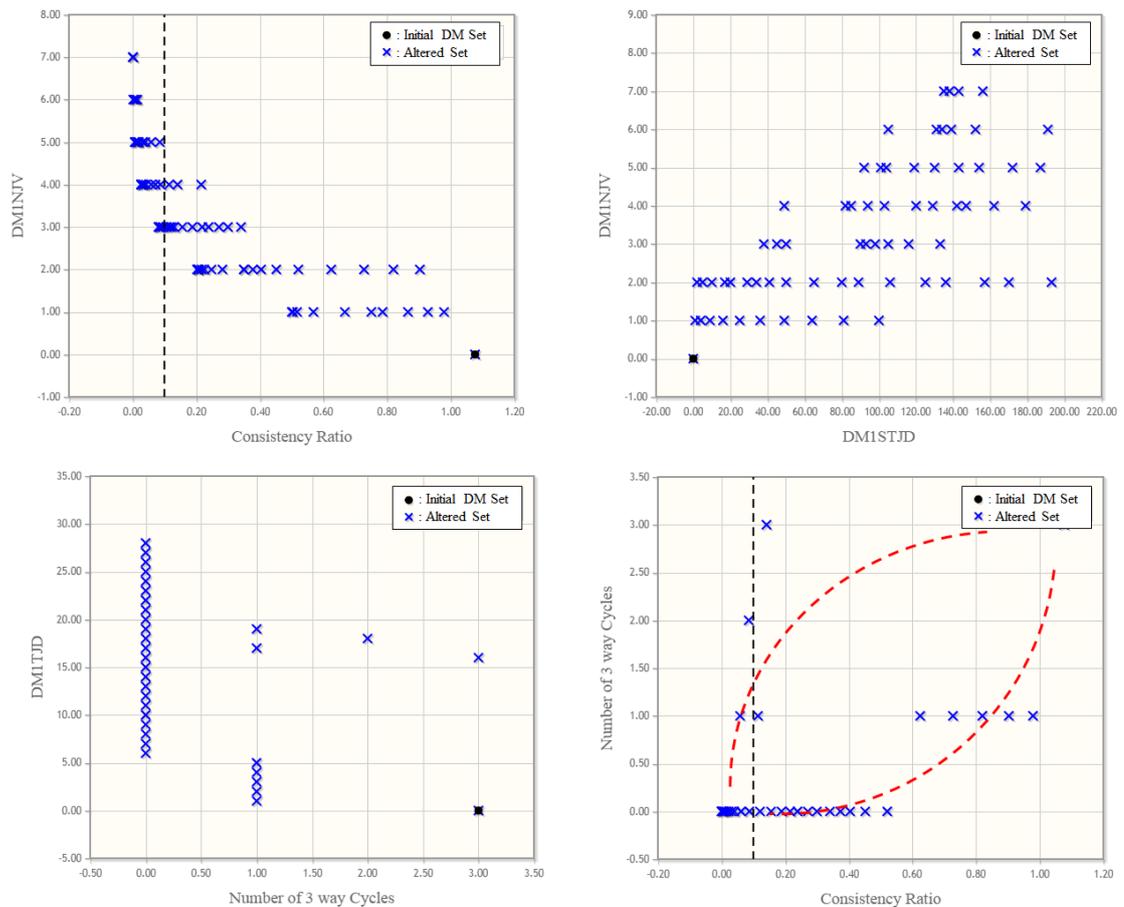


Figure 4.22: Example 4.4 Additional Objectives Solution Space analysis

## 4.4 Conclusions

In this chapter we have presented a new approach to reducing inconsistency within a set of PC judgments via a MOO approach. We have defined and discussed the stages of the approach, along with the objectives that are usable without the approach. The suitability of MOGAs to implement the approach has been demonstrated. Multiple examples comparing the approach to other approaches and highlighting the advantages and flexibility of the approach have been presented. The approach looks to optimally reduce inconsistency within a set of judgments through modelling inconsistency reduction and measuring alteration to the judgments as separate objectives to find a set of trade-off solutions between the conflicting objectives. The approach initially implements the 1-9 scale but could be extended to utilise any bounded scale. For all solutions found, any prioritization method can be utilised to derive a preference vector. The approach gives a DM control over the type of inconsistency reduction to seek as both cardinal and/or ordinal inconsistency measures can be chosen as objectives. The approach also gives a DM control over how alteration is measured to meet their needs via the measures of compromise. Using the measures of compromise additionally helps a DM glean understanding of the alteration involved and knowledge of the trade-offs involved helping facilitate traceability and sensitivity analysis. The approach seeks to alter judgments in such a way that the judgments maintain the original scale utilised by the DM, thus allowing a DM to more easily discern how their judgments have altered. The approach additionally allows a DM to set their own constraints relating to the amount of inconsistency reduction they are seeking to achieve as well as the amount of alteration they are willing to tolerate.

In this approach we have seen how MOO can be utilised to model the conflict between objectives or alteration and objectives or inconsistency reduction, to find altered judgment sets that look to minimise these conflicting objectives. Similarly when pairwise comparisons are utilised for group aggregation a consensus is sought between the DMs which, when conflicting views are present between the DMs, will result in some compromise between the DMs' views. Therefore building on from the approach to inconsistency reduction for a single DM, in the next chapter we present an approach to the aggregation of multiple DMs' judgments. The approach looks to reach aggregation through employing the measures of compromise to model as separate objectives the amount of alteration each DM's judgments undergo in the pursuit of reaching a group consensus.

# **Chapter 5 Group Aggregation of Pairwise Comparisons Using MOO**

In this chapter we present a new approach to the aggregation of PC judgments of a group of DMs via MOO. After a discussion of the problem and overview of the rationale of the approach with comparison to previous approaches, an overview of the stages of the approach is presented followed by an outline of the objectives that are usable within the approach. Next we present the various analysis the approach facilities to aid a group of DMs towards the selection of a single consensus solution. This is followed by examples of the approach including an example of the approach utilised within a group AHP decision. Finally conclusions of the approach are presented.

## **5.1 Aggregation of PC judgments of a group of DMs**

For many real-world decisions the opinions of multiple DMs are utilised, either to avail of their combined expertise or to incorporate conflicting views and experiences. In both cases there may be a level of disagreement and variance between the DMs, making the process of synthesising the DMs' judgments important. When utilised within a group environment, the process of deriving a preference vector from multiple DMs' judgments needs to incorporate the aggregation of the group of DMs' judgments into the formulation of a single preference vector for the group. Additionally, DMs are subject to irrationalities and various inconsistencies which may have adverse effects upon derived preference vectors. Therefore, seeking to additionally reduce inconsistency during the aggregation process can help to reduce its adverse effects on the group decision.

In Chapter 2 we analysed approaches to the aggregation of DM views when PC are utilised within group decision making. When eliciting separate judgments from each DM an aggregation approach then seeks to determine a single aggregated preference vector. Such aggregation may be approached via aggregation of individual priorities such as within the Weighted Arithmetic Mean Method (WAMM) or via aggregation of individual judgments such as within the Geometric Mean Method (GMM). Within both approaches Chapter 2 identified shortcoming with regard to 4 areas:

1. No explicit consideration of the amount of compromise of each DM's judgments during aggregation. Previous approaches do not provide any indication to the group of the amount of compromise in semantically meaningful terms, each DM's views have undergone to reach a consensus. This hinders a DM's validation of the outcome as well as limits the traceability and transparency of the aggregation process;
2. Further the other approaches offer no facilities for constraints to be set by DMs regarding the amount of compromise they are prepared for their views to undergo in the pursuit of a consensus solution;
3. Although previous approaches allow DM Weights of importance to be incorporated into the aggregation process they are static and defined at the start of the process, therefore they cannot be dynamically altered which is an importance consideration as defining weights is not a crisp single-pass task but rather is informed by refinement;
4. As we have seen when inconsistency is present in judgments it can adversely affect the accuracy of results therefore consideration of its reduction during aggregation would be advantageous. As WAMM aggregates individual priorities and preference vectors are derived before aggregation, it cannot consider inconsistency reduction during the aggregation process. Within aggregation of individual judgments inconsistency reduction could be considered, however GMM has no consideration of inconsistency during aggregation and in fact the amount of ordinal inconsistency may increase during GMM aggregation.

Based on this analysis, in this chapter we present an approach to the aggregation of PC judgments of a group of DMs that addresses these limitations.

## **5.2 MOO Approach to PC aggregation**

In this section we present a MOO approach to the aggregation of PCs of multiple DMs, which can simultaneously facilitate inconsistency reduction during aggregation. The approach, by modelling alteration to each DM's views as separate objectives, looks to achieve consensus for the optimally minimum amount of alteration to each DM's views as well as highlight underlying conflict between the DMs when seeking to achieve consensus. Alteration to each DM's judgments is explicitly considered through the use of the measures of compromise that give the DMs control over how alteration is measured to meet their needs as well as helping each DM glean understanding and knowledge of the trade-offs involved thus aiding traceability. When there are invariably differing views and conflict between DMs then, for the conflicting objectives of compromise to each

DM's views, there will not be a single solution that optimizes all objectives, instead a range of trade-off non-dominated solutions will exist. The approach looks to model the trade-off between the compromises needed to each DM's preferences to find aggregated PCMs - *Aggregated Consensus Solutions*. Through aggregation of individual judgments the approach is able to consider inconsistency reduction during aggregation. The approach facilitates inconsistency reduction during the aggregation process through modelling inconsistency measures as additional optional objectives to seek to find Aggregated Consensus Solutions with low inconsistency. A range of both cardinal and/or ordinal inconsistency measures can be considered giving the DMs control over the type of inconsistency reduction to seek.

The approach facilitates various analysis of the set of solutions found from the MOO process to aid the group towards reaching a final consensus. Such analysis can be via the DMs adding constraints to represent thresholds of the amount of compromise they are willing to tolerate in pursuit of consensus. The use of the compromise measures aid DMs in setting semantically meaningful constraints. The MOO approach allows for constraints to be iteratively added after the initial objective space has been viewed helping to ensure that DMs set feasible constraints. Additionally analysis of the set of solutions found can make use of knowledge of the global amount of alteration to the group of DMs' views to identify the solution(s) that represents the least overall compromise for the group. Similarly analysis of the set of solutions found can identify the solution(s) that represent the fairest amount of compromise. Furthermore the approach can facilitate weights of importance of the group of DMs to be incorporated during analysis of the set of solutions found. A weighted aggregated solution can be identified and sensitivity analysis of changing of the weights can be performed to identify the weighted aggregated solution reflecting this change.

The approach is independent of a specific prioritization method and any method can be utilised to derive a ranking from the Aggregated Consensus Solutions found, enabling the approach to be flexible to different problem scenarios and DM preferences. The approach implements the 1-9 scale to elicit judgments and this scale is utilised for the examples in this chapter however, the approach could be extended to use any bounded scale which would again allow it to be flexible to different problem scenarios and DM preferences.

In Chapter 4 a discussion and analysis of MOGAs and their suitability for MOO involving the measures of compromise was presented, along with discussions of the merits of different MOGAs. The approach to the aggregation of PC judgments of a group

of DMs via MOO is implemented via a MOGA. The approach is independent of a MOGA and various MOGAs could be applied. However for the examples and experiments presented in this chapter the MOCcell algorithm is utilised for the advantages outlined in Chapter 4, such as its ability to allow a DM to define the maximum number of non-dominated solutions that are returned. Next the stages of the approach are outlined.

### **5.2.1 Stages of Approach to Aggregation of PC judgments of a group of DMs**

The stages of the approach, shown in Figure 5.1, can be summarized as follows:

1. The number of elements of the problem is defined;
2. Judgments are elicited from each DM relating to their preferences between the elements of the problem;
3. The objectives for the MOO process are selected, consisting both of objectives of compromise to each DM's views as well as additional optional inconsistency objectives;
4. The set of objectives are utilised within MOO to find the set of Aggregated Consensus Solutions;
5. Analysis of the set of Aggregated Consensus Solutions is performed to aid the selection of a single solution from the set of solutions. Such analysis may take one or more forms:
  - i. Analysis of the set of found Aggregated Consensus Solutions with respect to the levels of conflict between the DMs. The DMs can then iteratively add feasible constraints to gradually drill down to a sub-region of the objective space to help reach a final group consensus;
  - ii. Analysis utilizing weights of importance of each DM to help to identify a solution for which the compromise to each DM's views is proportional to their weights. Sensitivity analysis of changing the weights of importance can then be performed.

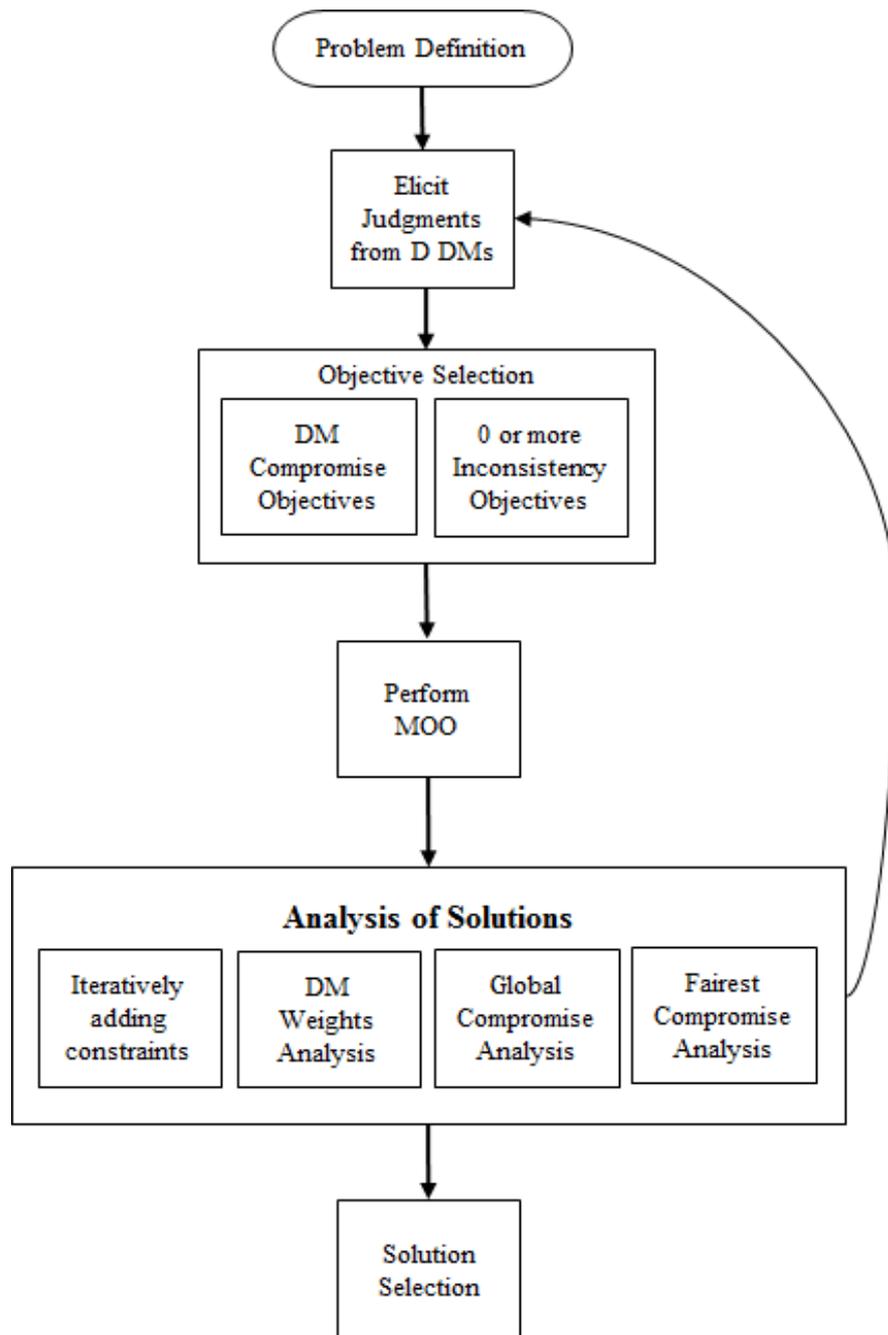


Figure 5.1: Flowchart of the stages of the approach of aggregation of PC judgments of a group of DMs

- iii. Analysis utilizing information pertaining to the global levels of compromise in the group to help identify solution(s) which represents the least overall compromise to the group of DMs;
- iv. Analysis of the set of found solutions with respect to the amount of compromise that each DM undergoes in comparison to the other DMs helping to identify solution(s) that represent the fairest compromise to the group of DMs' views;

- v. Analysis of the Aggregated Consensus Solutions and their conflict may also aid identification of scenarios with unacceptable levels of conflict within the group (in the view of the DMs) in which amendment and update of the DMs' judgments may instead be valuable. Therefore as the approach reveals the levels of conflict within the group it can highlight to the group when the conflict is high and where amendment and update of the DMs' judgments may be more beneficial than seeking aggregation under such high levels of conflict.

### 5.2.2 Multi Objective Optimisation

The approach seeks to find Aggregated Consensus Solutions through modelling the amount of compromise that each DM's judgments undergo as separate objectives, along with additional optional inconsistency measure objectives, via MOO.

Given a problem with  $n$  elements and  $D$  decision makers they each define a complete  $n$  by  $n$  PCM of their judgments:

$$\{PCM_1, PCM_2, \dots, PCM_D\} \quad (5.1)$$

From these PCMs we can extract from each their top triangle of  $J$  judgments of each DM's views. Therefore for each DM a *Judgment Set*  $O$  of cardinality  $J$  can be selected, containing enough information to reconstruct the whole of the PCM. The approach models an  $O$  representation of each DM's PCM  $\{O_1, O_2, \dots, O_D\}$ , each of which consists of  $J$  judgments  $\{o_1^k, o_2^k, \dots, o_j^k\}$ , for  $k=1, \dots, D$ .

We seek the set of non-dominated Aggregated Consensus Solutions. Again we can represent each solution as a judgment set of cardinality  $J$ , denoted as  $A = \{a_1, a_2, \dots, a_j\}$ . The set  $A$  represents the decision variables that can be obtained by minimising a set of objectives  $\Lambda$ .

The MOO problem can be formulated as:

$$\text{Minimize } [\Lambda] \quad (5.2)$$

where

$$\Lambda = \{E, B\}$$

The set of objectives  $\Lambda$  consists of two subsets. The first subset  $E$  represents measures of compromise objectives  $\varepsilon_i(A)$  of cardinality  $D$ , that each seek to minimize the measure of compromise with respect to the corresponding  $O_i$  of a DM. This subset is defined as:

$$E = \{\varepsilon_1(A), \varepsilon_2(A), \dots, \varepsilon_D(A)\} \quad (5.3)$$

The second subset B represents inconsistency objectives  $\beta_i(A)$  of cardinality  $m$ .

$$B = \{\beta_1(A), \dots, \beta_m(A)\} \quad (5.4)$$

The approach additionally allows each DM to set constraints upon the amounts of compromise they are willing to tolerate in the pursuit of reaching a consensus. They can be employed, for example, to define an upper limit to the amount of a measure of compromise that a DM's judgments can undergo during optimization.

For example, given a constraint from  $DM_i$  of  $c_i$  upon their measure of compromise, the following constraint could be defined:

$$\varepsilon_i(A) \leq c_i \quad (5.5)$$

Constraints can additionally be set upon inconsistency objectives to define bounds upon the amount of inconsistency permitted within found aggregated consensus solutions. Thus a constraint  $f_j$  upon an inconsistency objective  $\beta_j$  from objective subset B is defined as:

$$\beta_j(A) \leq f_j \quad (5.6)$$

So, the constrained multi-objective optimization problem can be formulated as:

$$\text{Minimize } \{E, B\} \quad (5.7)$$

subject to

$$\begin{aligned} \varepsilon_i(A) &\leq c_i \\ \beta_j(A) &\leq f_j \end{aligned}$$

for  $i=1,2,\dots,D$  and  $j=1,2,\dots,m$

The approach then seeks to simultaneously optimize this set of objectives to find the trade-off front of the problem, a set of *Aggregated Consensus Solutions*. A preference vector can then be derived from any of the aggregated consensus solutions found. Figure

5.2 shows an illustrative objective space found using the approach for a 4 element, 3 DM problem. Each axis represents a compromise measure objective denoting compromise for each DM. Additional inconsistency objectives could also have been employed increasing the number of dimensions of the objective space. A set of trade-off aggregated consensus solutions (in this case 8) have been found and their measures with respect to the amount of compromise to each DM are shown through their position within the objective space. From an aggregated consensus solution a preference vector of the 4 elements under consideration can then be derived.

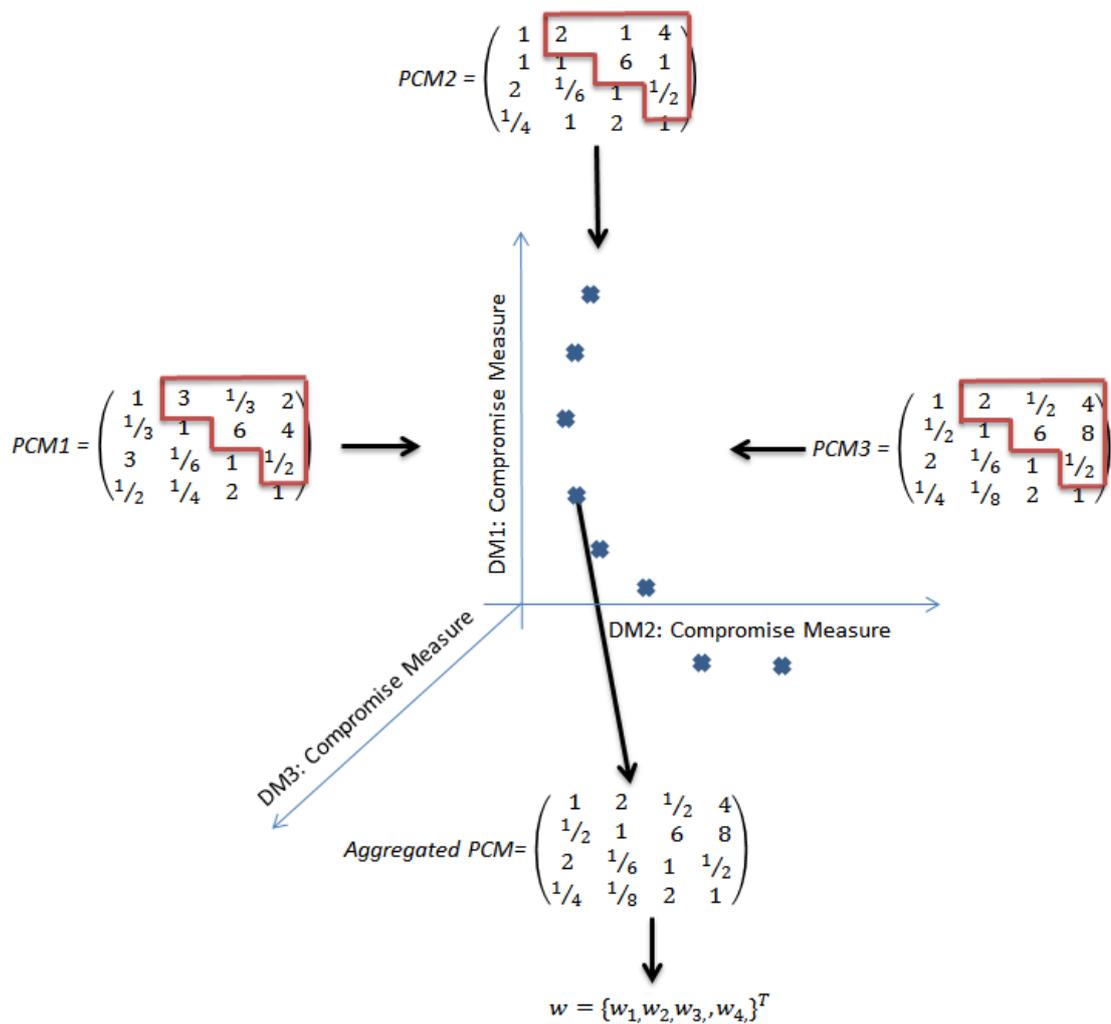


Figure 5.2: Illustration of MOO approach of aggregation of 3 DMs' preferences

Within the approach the measure of compromise that is to be utilised as a separate objective for each DM can be chosen based upon the preferences of the group so that each

DM utilises the same measure<sup>7</sup>. Through each DM utilizing the same measure of compromise additional functionality and analysis relating to total compromise information can be performed to aid the group to select a consensus solution see Section 5.2.4

The objectives that are usable within the approach, both of compromise measures and of inconsistency measures are outlined next.

### 5.2.3 Objectives usable within the Approach

The measures of compromise and inconsistency usable as objectives within an approach are outlined below. In addition to these measures, tackling the problem via a MOO framework allows for additional measures, if a new measure is devised, to be implemented into the approach: additional measures would only need to define an evaluation function to be implementable into the approach.

The levels of alteration to each DM's judgments can be measured via the measures of compromise as defined in Chapter 3.

1. **NJV**: Number of Judgment Violations: a measure of the number of judgments that have changed;
2. **TJD**: Total Judgment Deviation: a measure of the total amount of change between each judgment;
3. **STJD**: Squared Total Judgment Deviation: a variant of TJD which gives more emphasis to larger amounts of deviation to a judgment;
4. **NJR**: Number of Judgment Reversals: a measure of the number of judgments that have been reversed.

The optional addition consistency measures that can be utilised as objectives in the approach are: (for discussion of these see Chapter 2.)

1. **CR**: Consistency Ratio [33]: a measure of the level of cardinal inconsistency present, proposed by Saaty along with an additional threshold of acceptability value of 0.1;

---

<sup>7</sup> The approach, technically, could also be flexible to allow each DM to be modelled with a different compromise measure (and for a single DM to be modelled with multiple measures) but this would come at the expense of some functionality.

2. **L**: An ordinal measure of the number of 3-way cycles present within a set of judgments including consideration of equal preference;
3. **CM**: Consistency Measure [73]: a cardinal measure of inconsistency that considers the most inconsistent triple of judgments within the whole set;
4. **GCI**: Geometric Consistency Index measure [72]: an inconsistency measure based upon the distance measurements between the preference vector derived using the GM prioritization method and the original judgments.

#### **5.2.4 Analysis of solutions to aid towards selection of a single solution**

From the discovery of a trade-off set of solutions for the problem, with respect to the set of objectives, analysis of the objective space and measures concerning the conflict between the DMs can be determined. Through analysis of the nature of the objective space knowledge of the conflict between the DMs can be revealed. The approach can extract the measure of conflict between each pair of DMs with respect to the chosen measure of compromise, and create a ranking of the set of pairs. This is calculated as the distance measure of compromise between each pair of DMs within the objective space – a measure from an edge of the objective space to another edge. This allows identification of DMs with closely matching views, as well as of DMs for which there is a high level of conflict, helping to give a deeper understating of the nature of the group of DMs' views. The approach can further calculate an average of these pairings to derive a measure of agreement for each DM. These values when ranked can then highlight the “most agreeable” DM, who is most representative or typical of the group and also the “most disagreeable” DM (or least representative or typical of the group), whose views are the most outlying.

In addition to seeking to find the set of trade-off solutions for the set of conflicting objectives, additional consideration needs to be given to ultimately help the group towards selecting a group consensus solution. From a single group consensus solution, a preference vector ranking of the elements can then be derived. For the examples presented in Section 5.3 the GM prioritization method [42] is utilised to derive preference vectors from aggregated consensus solutions, however the approach is independent of a specific prioritization method and any method can be utilised to derive a ranking. Furthermore, when the approach actively seeks aggregated consensus solutions with low inconsistency, the choice of prioritization method becomes less significant. Additional analysis facilitated via the approach to aid the selection of a single consensus solution from the front of solutions can be performed through:

1. Iteratively adding constraints;
2. Global measures of compromise analysis;
3. DM weights of importance analysis;
4. Fairest compromise analysis.

These areas of analysis are discussed next.

#### **5.2.4.1 *Iteratively adding constraints***

To help the group towards the selection of a single consensus solution the approach allows DMs to define the levels of tolerance they are willing to accept in the pursuit of reaching a consensus. The approach facilitates this by allowing constraints to be iteratively defined to explore and drill down into the objective space.<sup>8</sup> Through tackling the problem of aggregation via MOO the approach facilitates an initial exploration of the objective space to reveal knowledge of the overall conflict between the DMs. This way the DMs can then look to define feasible thresholds of tolerance as constraints based upon this knowledge of the conflict. Then through iteratively adding constraints the DMs can move towards an area of the objective space to facilitate the selection of a consensus solution in an interactive and traceable manner. The approach implements constraints within a MOO framework implemented via MOGAs initially via reducing the fitness of infeasible solutions to push the population towards the feasible region of the objective space, via Constrained Pareto Dominance [119], as defined for constraint handling in NSGAI. The approach additionally introduces a hard constraint upon the archive to only allow feasible solutions to be added to the archive, thus ensuring that only feasible solutions will be presented to the DMs. See Section 4.2.5 for a more detailed discussion of constraints and the Constrained Pareto Dominance constraint strategy.

Moreover when additional inconsistency objectives are utilised by the group the approach allows constraints to be defined to these objectives. For example, when the inconsistency measure CR is employed as an additional objective a constraint could be defined upon its upper value of 0.1 (thus adhering to Saaty's threshold of acceptable

---

<sup>8</sup> Constraints are added as part of a new search utilising the constraints within the operation of the MOO search, however they could also be added to a set of found solutions to simply slice up the objective space of the solutions without re-searching if that meets the DMs' preferences.

inconsistency) to only seek to find Aggregated Consensus Solutions with acceptable levels of CR.

#### 5.2.4.2 Utilizing Global levels of Compromise information

Through analysis of information relating to the global levels of compromise within the group of DMs, the least amount of overall compromise for the group to reach a consensus can be calculated. For each aggregated consensus solution found, a global measure of the total compromise of the measure used by the DMs as objectives can be calculated. This represents the sum of the compromise measure value for each DM for the chosen measure of compromise. For example, the global Total Number of Judgment Reversals (*TNJR*) for *D* DMs can be calculated via:

$$TNJR = \sum_{i=1}^D \sum_{j=1}^J R_j \quad (5.8)$$

Global measures of compromise for the other measures of compromise NJV, TJD and STJD can be calculated in a similar way. From such calculation for each solution, a ranking of the set of aggregated consensus solutions can be made with respect to their global measure of compromise, from which the solution with the lowest global measure of compromise can be identified (or multiple solutions when more than 1 solution share the lowest global value). Calculations of the global measures of compromise for 2 DMs in which STJD is the compromise measure for each DM are illustrated visually in Figure 5.3. From this we see that there is a single solution with the lowest total value of compromise, with a TotalSTJD value of 127, located towards the middle of the front of solutions and highlighted as a yellow triangle. Such analysis can assist DMs to identify the solution with the lowest overall compromise to aid in their analysis of the objective space and aid towards reaching a consensus decision. In this example there were 2 DMs; for larger number of DMs with more complex objective spaces and larger solutions sets, the same calculation of the total of each compromise value can be calculated and could technically be utilised for any number of DMs.

Such calculation of global measures of compromise could additionally assist in the selection of a consensus solution and derivation of a preference vector from it automatically, thus allowing the approach to derive an aggregated solution from the sets of the DMs' judgments in scenarios when analysis and negotiation is not possible. In some cases analysis of global values of compromise will identify a subset of aggregated

consensus solutions that all share the lowest total measure of compromise value. In this case the approach calculates a single preference vector as the average (utilizing the geometric mean) of the preference vectors derived from this subset of aggregated consensus solutions. In this way a Global Consensus preference vector is found that represents the average of the subset of solutions that share the least overall compromise value.

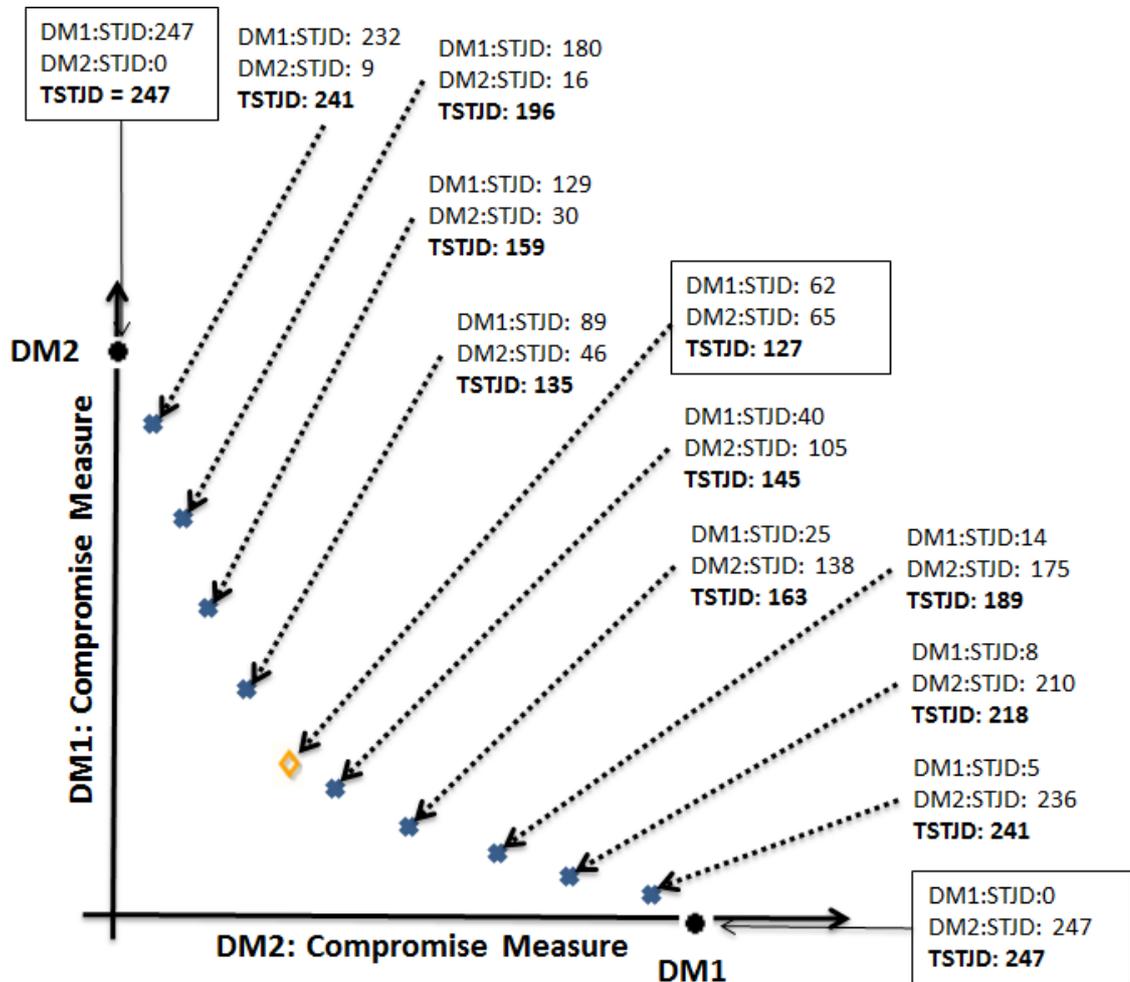


Figure 5.3: Global solution calculation illustration

### 5.2.4.3 Utilizing DM Weights of Importance analysis

Within group decision making there will be many cases due to, for example, position or expertise where the weight of importance of each DM is not the same and can be incorporated within the aggregation process through DM weights. Additionally, as we saw in Chapter 2, determination of such DM weights is not straightforward and as such weights that are incorporated dynamically will be beneficial. Although the GMM can incorporate weights of importance defined before the process they are not incorporated dynamically, similarly the WAMM incorporates weights only during calculations

between the set of separate preference vectors so they are also not incorporated dynamically. The approach allows for DM weights of importance to be incorporated into the aggregation process through identification of a weighted global solution from the front of solutions found, that is a global compromise solution that is weighted to take into account the weight of importance of each DM. Therefore the higher the weight of importance of a DM, the more weight their compromise carries within the global compromise calculations. Additionally, due to the use of the measures of compromise, the amounts of compromise each DM has to undergo within the weighted global solution are more easily comprehended by the DMs. Such visibility is not possible for solutions found using weights within the WAMM and GMM. By altering the set of DM weights in the approach sensitivity analysis can examine how changes to the DM weights of importance affect the selected weighted global solution. As the front of aggregated consensus solutions has been found such sensitivity analysis can directly be performed without the need to re-run the aggregation process (which would be required for weights incorporated in the WAMM and GMM approaches). To determine the weighted global solution from the set of solutions found, a weighted global measure of compromise is determined for each solution, from which the solution(s) with the lowest weighted global compromise can be identified. To calculate the weighted global measure of compromise for a solution, the measure of compromise value for a DM for the solution is multiplied proportionately to their weight and summed together. For example, the Weighted global Total Number of Judgment Reversals (*WTNJR*) for  $D$  DMs and with set of  $D$  DM weights  $w_1$  to  $w_d$ ,<sup>9</sup> can be calculated via:

$$WTNJR = \sum_{i=1}^D \left( \sum_{j=1}^J R_j \right) * w_i \quad (5.9)$$

Weighted global measures of compromise for the other measures of compromise NJV, TJD and STJD can be calculated in a similar way. In this way the approach still seeks low overall compromise during consideration of DM weights as the DM weights are utilised to modify the global calculation of each solution. For example, if we consider the example from Section 5.2.4.2 illustrating the global measure of compromise and given further DM weights of importance of 2/3 for DM1 and 1/3 for DM2 (that is to say DM1 is twice as

---

<sup>9</sup> So as to be comparable to equal weighted calculations the set of weights are normalised such that they sum to  $D$ , however any form of numerical representation of weights could be utilised.

highly weighted as DM2), we can look to identify the weighted global solution. The same aggregated solutions from Figure 5.3 along with their weighted total calculations for the solutions are shown in Figure 5.4: Left, the global measure again is shown as a yellow triangle and now the weighted global solution is shown as a red circle. We see a solution has been identified as the weighted solution which is closer to DM1's initial judgments due to their weight being higher. Through altering the weights we then see the effects on the weighted solution. For example, changing the weights so that DM1 is three times as important (0.75) as DM2 (0.25); the illustrative objective space of these new weights is shown in Figure 5.4: Right along with new weighted compromise calculations and identification of a weighted solution even more in favour of DM1. Such sensitivity analysis can aid the group towards the selection of a single solution.

Again, as with the global solutions calculation, the weighted solutions calculations can be applied for larger number of DMs than 2 where, given their more complex objective spaces and much larger solutions sets, the potential insights they can provide become more valuable.

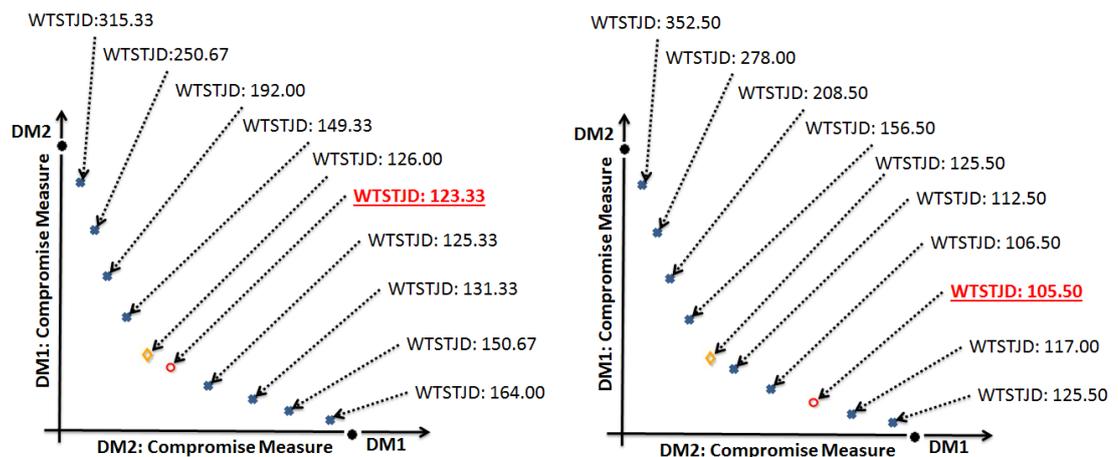


Figure 5.4 Left: Weights DM1:2/3 DM2:1/3. Right: Weights DM1: 0.75 DM2:0.25

Another way we can think of the effects of changing the weights is as affecting the shape of the curve of the objective space. We can show this for the three sets of weighted global values we have just seen, plotted in Figure 5.5 (the two sets of values calculated from the 2 weight sets, and the set of values of the un-weighted global values as essentially both DMs having equal weights). From this plot we can see the effects of incrementally giving DM1 a higher weight of importance.

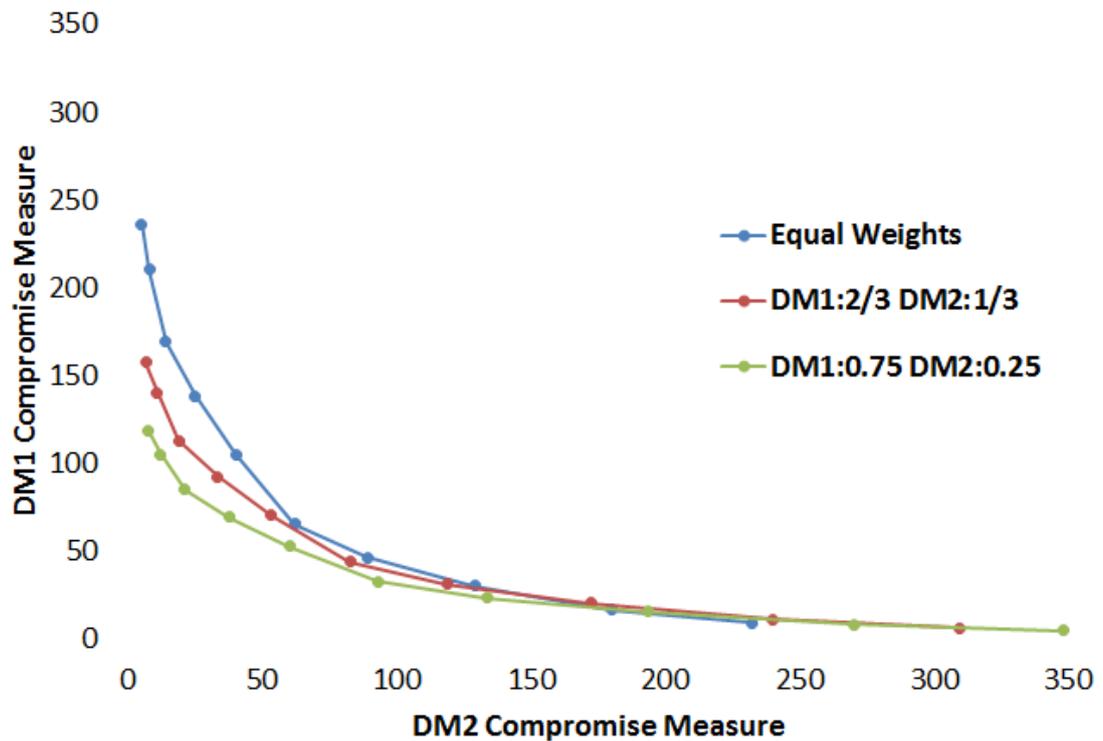


Figure 5.5: Analysis of weighted values

Calculation of weighted global values for each solution could moreover assist in the selection of a solution and derivation of a preference vector from the weighted solution(s) automatically. This enables the approach to derive an aggregated solution from the sets of DM judgments in scenarios when weights are to be incorporated, and analysis and negotiation is not possible. If multiple solutions share the lowest weighted compromise value the approach calculates a single preference vector as the average (utilizing the geometric mean) of the preference vectors derived from this subset of aggregated consensus solutions. In this way a preference vector is found that represents the average of the subset of solutions that share the weighted total compromise value.

#### 5.2.4.4 *Fairest Compromise analysis*

Analysis of the global amount of compromise in the group is useful for group of DMs seeking aggregation considering the smallest overall compromise. Weights of importance of DMs are useful when seeking aggregation in line with the weights of importance of each DM. Addition analysis of the set of found solutions can also analyse the ratio of compromise that a DM's views undergo in relation to the amount of compromise the other DMs undergo. In that way the fairness of a solution, with respect to how equal the amount of compromise each DM undergoes, can be determined. Such analysis is useful for any group of DMs seeking to gain deeper knowledge of the aggregation of their views and

especially useful for a group of DMs with competing or adversarial views and wishing to identify aggregation with respect to their views undergoing equal compromise in reaching a consensus. For each solution we can calculate the range of the amount of the compromise measure each DM undergoes within the solution. For example, the Range of the Number of Judgment Reversals (*RNJR*) for *D* DMs can be calculated via:

$$RNJR = \bigwedge_{i=1}^D R_i - \bigvee_{i=1}^D R_i \quad (5.10)$$

Range calculations for the other measures of compromise NJV, TJD and STJD can be calculated in a similar way. From such calculation for each solution, a ranking of the set of aggregated consensus solutions can be made with respect to their range of the measure of compromise, from which the solution with the fairest range measure of compromise can be identified (or multiple solutions when more than 1 solution share the lowest range value). As this analysis is performed upon the set of non-dominated solutions found such a solution represents the most equal solution from the set of solutions that make up the trade-off front of the problem. For example, 2 possible aggregated solutions for a 2 DM problem both using the TJD compromise measure could be, for solution 1 both DMs undergo 10 TJD where as in solution 2 DM1 undergoes 7 with DM2 undergoing 9. In this case the 1<sup>st</sup> solution is dominated by the 2<sup>nd</sup> so would not have made it into the final archive of found solutions to be considered for fairest calculations. Therefore the approach is focused upon finding the optimal solutions of Pareto quality between the DMs' objectives and from these solutions the equality of their compromise is then considered. From a set of found solutions it may be the case that multiple solutions share the same levels of total compromise yet the distribution of this total between the DMs will not be the same. For example, consider a 2 DM problem with STJD as the measure of compromise. Figure 5.6 show a hypothetical front of solutions with the STJD value for each DM along with the calculated total measure and the range measure for each solution. We see in this example how the solution with the least overall compromise, with a TSTJD value of 20, has a larger range value of 4 than the fairest solution, which has a range of 2 and is shown as a hollow purple triangle. We can see this more clearly in Figure 5.7 which plots this set of solutions (and initial DM judgment sets) values of both TSTJD and range.

We see, highlighted with vertical lines, the different fairest solution and global solution within the set of solutions.<sup>10</sup>

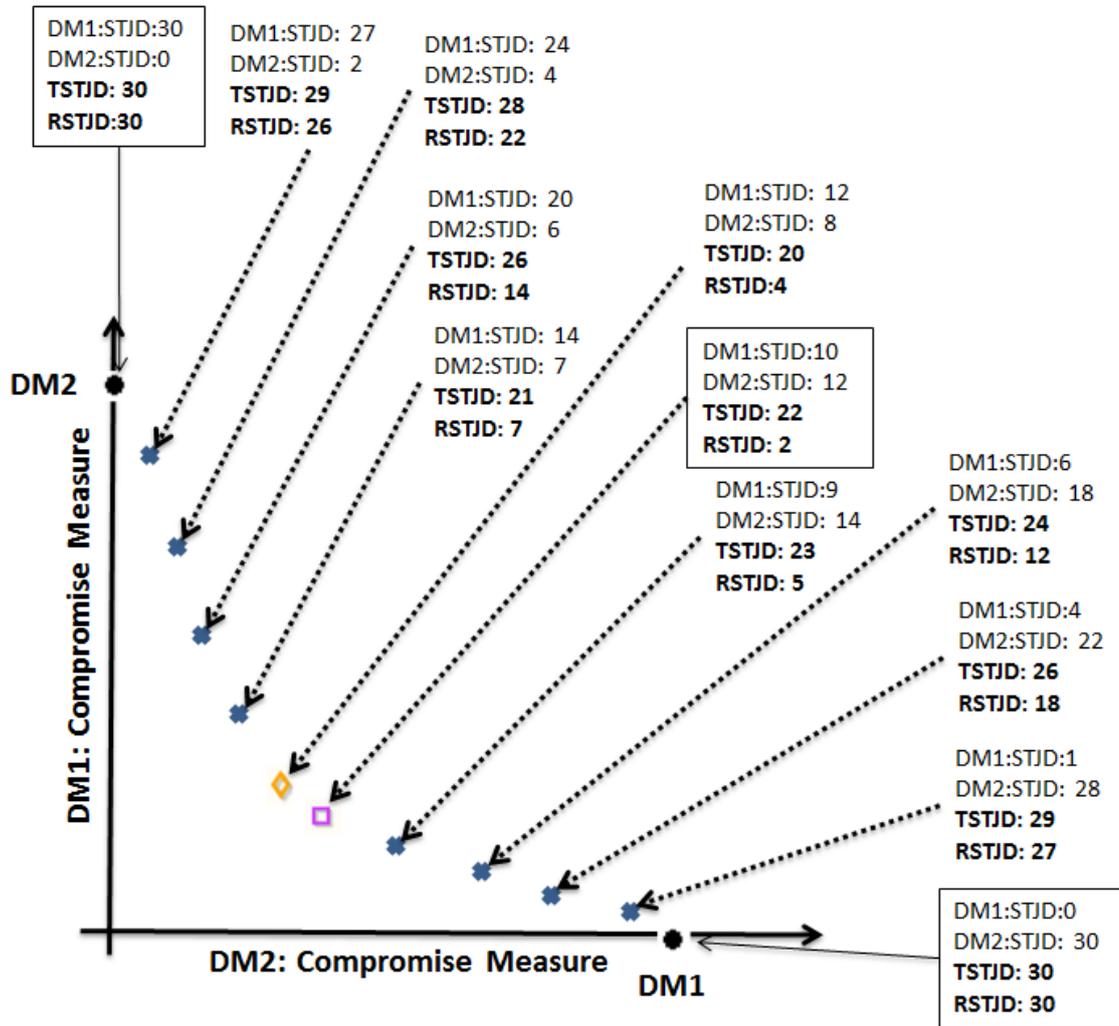


Figure 5.6: Equal compromise range calculation illustration

As with the global and weighted calculations the calculation of the range of compromise for each solution could additionally assist in the automatic selection of a solution and calculation of a preference vector from the most equal compromise solution(s). This would allow the approach to derive an aggregated solution from the sets of DM judgments in scenarios when analysis and negotiation is impossible and an aggregation is sought that represents a fair compromise to the DMs involved. If multiple solutions share the lowest range of compromise value, the approach calculates a single preference vector as the

<sup>10</sup> In such an example of 2DMs and the same measure of compromise the global and equal solutions will invariably be the same solution – however it is easier to explain and illustrate an example for just 2 DMs. See Example 5.3 in Section 5.3.3 with 3 DMs to see the difference between global optimal and equal optimal solutions.

average (utilizing the geometric mean) of the preference vectors derived from this subset of aggregated consensus solutions. In this way a preference vector is found that represents the average of the subset of solutions that share the lowest range of compromise value.

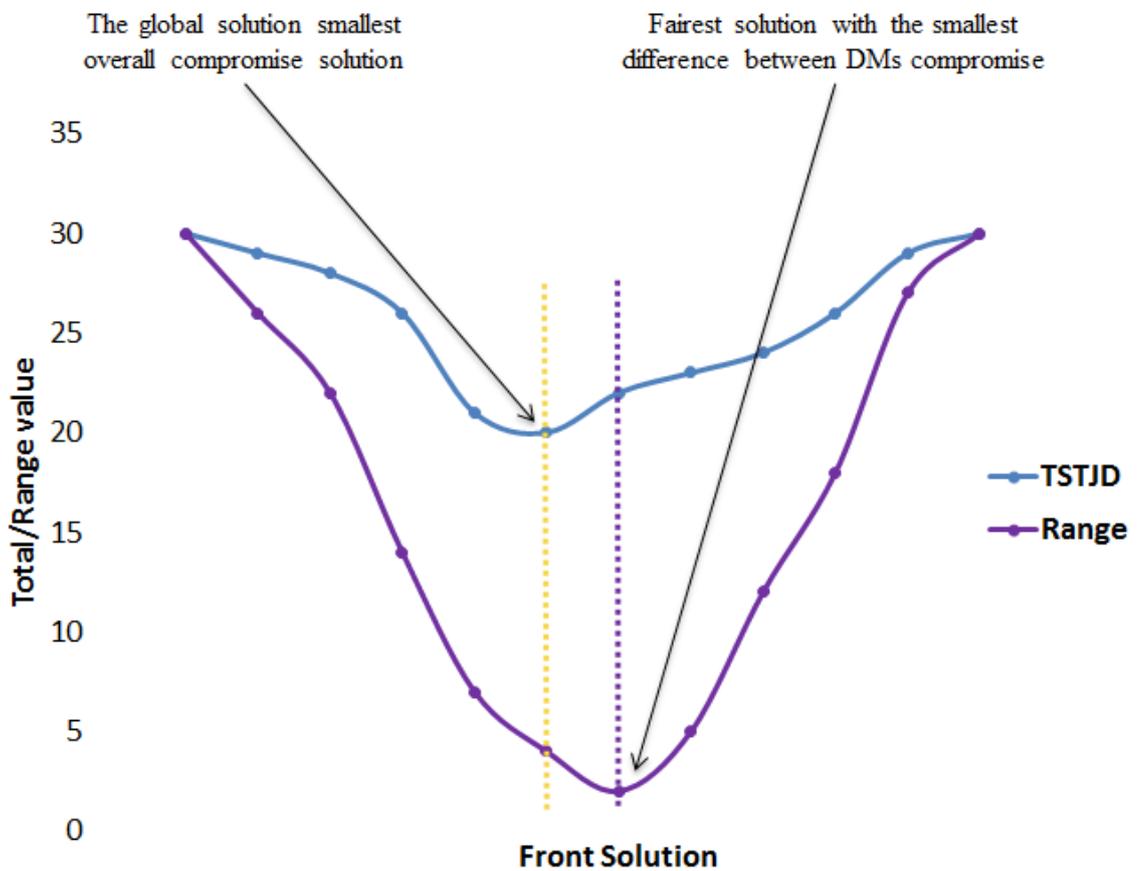


Figure 5.7: TSTJD and range values from the equal compromise illustrative example

### 5.2.5 MOO Approach to PC aggregation Discussion

We have outlined the approach to the aggregation of PC judgments of a group of DMs via a MOO approach and have outlined its advantages and the stages of the approach. The MOO definition of the problem has been outlined along with the objectives usable in the approach. Outlines have been presented of how various forms of analysis of the set of solutions found can aid the group towards the selection of a single consensus solution. In the next section examples explore the approach and its benefits along with comparisons to the GMM. This is followed by an application of the approach within an example AHP decision.

### 5.3 Examples

In this section, step-by-step examples of the approach are presented.

1. Example 5.1 explores utilizing the global calculations of compromise and compares the approach to the GMM;
2. Example 5.2 explores how DM weights of importance analysis can be utilised and dynamically altered to investigate the set of solutions found;
3. Example 5.3 illustrates how DMs can iteratively add constraints to aid towards the selection of a single consensus solution;
4. Example 5.4 explores how inconsistency reduction can be incorporated into the aggregation process to look to reduce the adverse effects of inconsistency upon rankings derived from a consensus solution;

For these examples the MOCCell algorithm was utilised with the following parameter settings: population size of 100 (10 x 10 grid); maximum evaluations count of 25,000; with selection performed via binary tournament with single point crossover (with crossover probability 0.9) and bit flip mutation (with probability 0.01) employed. The size of the archive is stated in each example (with the feedback value set to 25% of the size of the archive). Any prioritization method can be utilised within the approach to derive preference vectors from aggregated consensus solutions; the GM prioritization method is utilised in these examples.

#### 5.3.1 Example 5.1: Global solution analysis and comparison to GMM

Example 5.1 explores utilizing global measures of compromise information as well as comparing the MOO approach to the GMM. The judgments from 2 DMs for a 5 element problem are shown in Table 5.1.

Given STJD as the measure of compromise for both DMs, and an archive size of 20, the solution space for this 2-objective problem is shown in Figure 5.8. The global solution identified is shown as a hollow diamond and the solution found via GMM as a hollow circle.

Table 5.1: Example 5.1: D: 2 N: 5

		DM1							DM2				
		1	2	3	4	5			1	2	3	4	5
1	1	3	3	1/7	1/5	1	1	1/9	2	1	1/2		
2	1/3	1	7	1/4	3	2	9	1	6	1/8	1/4		
3	1/3	1/7	1	1/3	1/4	3	1/2	1/6	1	1/6	1/3		
4	7	4	3	1	1/9	4	1	8	6	1	1/2		
5	5	1/3	4	9	1	5	2	4	3	2	1		

The STJD values of each DM and Total STJD for both the GMM solution and the global MOO solution are shown in Table 5.2. From the plot of the objective space in Figure 5.8 and from the values in Table 5.2 we see that the GMM solution is dominated with respect to STJD and that the global solution found from the MOO approach has less STJD for both DMs than the GMM solution.

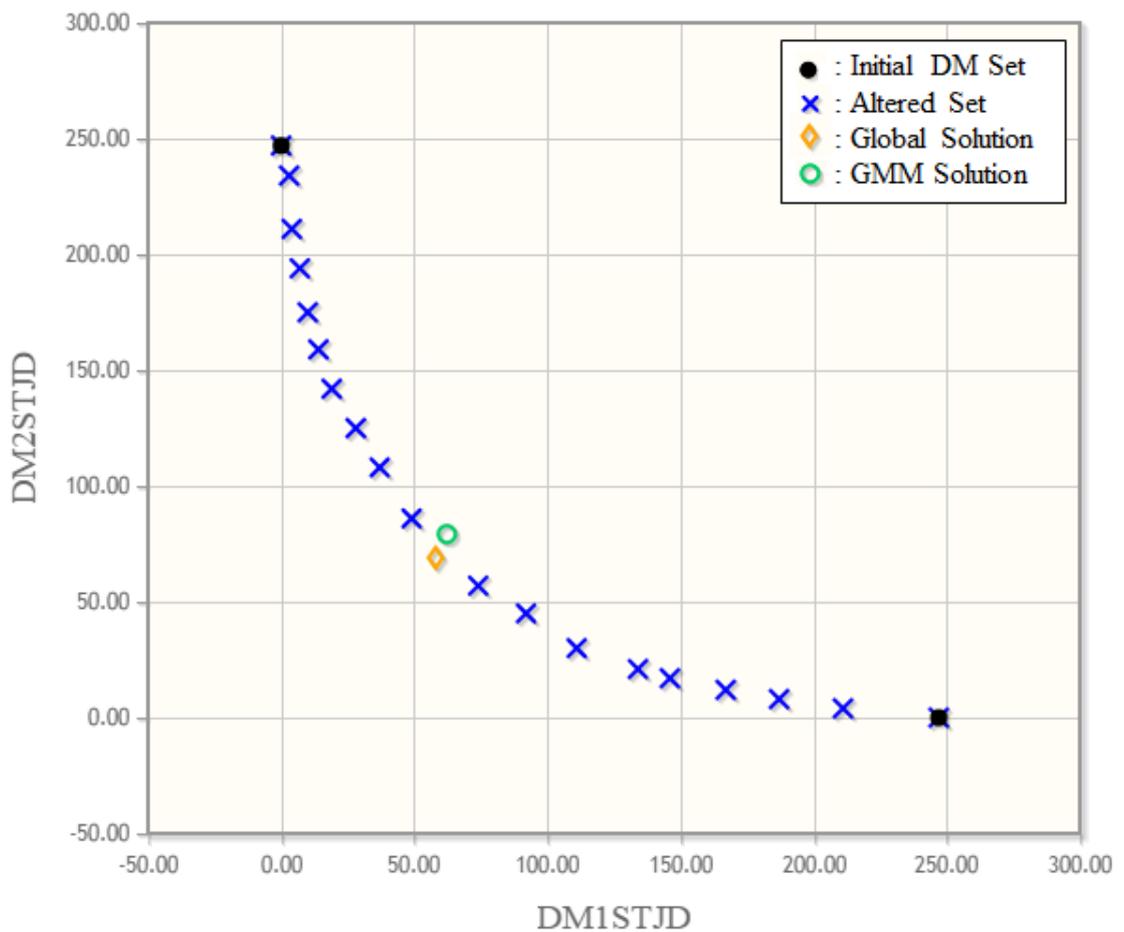


Figure 5.8: Example 5.1 objective space

Table 5.2: Example 5.1: STJD Measures

	<b>DM1:STJD</b>	<b>DM2:STJD</b>	<b>Total:STJD</b>
<b>MOO</b>	58.00	69.00	127.00
<b>GMM</b>	62.23	79.23	141.46

The PCM of the aggregated solution found via GMM is shown in Table 5.3 and the PCM of the global MOO solution found via the approach is shown in Table 5.4. From these we see how, as the approach adheres to the original judgment scale utilised by the DMs to elicit judgments, it is easier for the DMs to comprehend the aggregated solution and discern how their judgments have altered.

Table 5.3: Example 5.1: GMM PCM

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	1.00	0.5774	2.4490	0.3780	0.3162
<b>2</b>	1.7319	1.00	6.4810	0.1768	0.8660
<b>3</b>	0.4083	0.1543	1.00	0.2357	0.2887
<b>4</b>	2.6455	5.6561	4.2427	1.00	0.2357
<b>5</b>	3.1626	1.1547	3.4638	4.2427	1.00

Table 5.4: Example 5.1: MOO Global Solution PCM

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	1	1/4	2	1/4	1/4
<b>2</b>	4	1	6	1/6	1
<b>3</b>	1/2	1/6	1	1/5	1/4
<b>4</b>	4	6	5	1	1/6
<b>5</b>	4	1	4	6	1

Then if the global solution identified by the approach is chosen by the DMs as the desired consensus solution a preference vector can be derived, as shown in Table 5.5.

Table 5.5: Example 5.1: Preference vector from global solution of MOO approach

	1	2	3	4	5
MOO	0.07733	0.2041	0.05168	0.2816	0.3853

### 5.3.2 Example 5.2: DM weights of importance analysis

We now explore how DM weights of importance can be utilised dynamically within the approach. Taking the judgments from the 2 DMs in Example 5.1 we further explore the objective space with the DMs' weights of importance. We can think of the initial objective space and global solution space as the DMs initially having equal weights of importance. We then look to perform sensitivity analysis of the weights. Given weights of importance of DM1:0.65 and DM2:0.35, we see the updated objective space in Figure 5.9: Left, with the weighted solution shown as a hollow red circle. We see that within the weighted solution DM1 undergoes less compromise and also see this solution in relation to the global (equal weights) solution. Given altered weights of DM1:0.85 and DM2:0.15 the further updated objective space is shown in Figure 5.9: Right. We see the effects of increasing DM1's weight of importance upon the weighted solution identified. The DMs can additionally analysis the preference vectors from these different weighted solutions shown in Table 5.6.

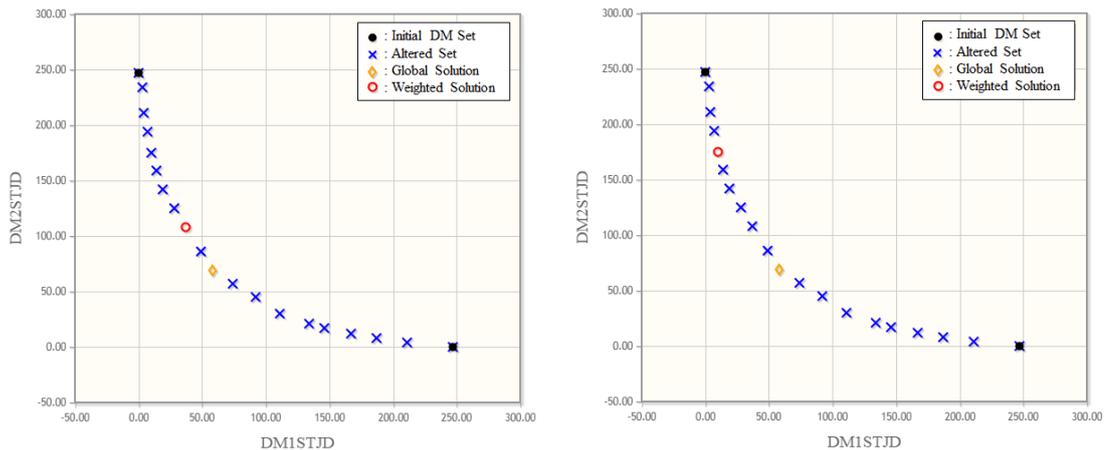


Figure 5.9 Left Weights {DM1:0.65, DM2:0.35}. Right Weights {DM1:0.85, DM2:0.15}

Table 5.6: Example 5.2: Preference vectors from weights analysis

	1	2	3	4	5
Weights 1	0.1046	0.1586	0.05299	0.2887	0.3951
Weights 2	0.1039	0.2224	0.05619	0.2624	0.3551

Weights can also be utilised for higher numbers of DMs. Given a 6-element problem, PCM judgments from 4 DMs are shown in Table 5.7. With NJR chosen as the measure of compromise for each DM we then perform the MOO for this 4-objective problem with a large archive of 150 solutions defined. Within the higher dimension space the size of the front of found solutions is much larger, from the MOO stage a set of 128 solutions was found.

Table 5.7: Example 5.2: D: 4 N: 6

		<b>DM1</b>						<b>DM2</b>							
		1	2	3	4	5	6			1	2	3	4	5	6
1	1	1	1/5	1/2	7	1/8	1/5	1	1	7	1/3	5	8	1/4	
2	5	1	3	8	8	1/9		2	1/7	1	1/7	1/5	1	1/9	
3	2	1/3	1	1/2	1/9	6		3	3	7	1	2	5	5	
4	1/7	1/8	2	1	2	4		4	1/5	5	1/2	1	1/2	1/4	
5	8	1/8	9	1/2	1	1/9		5	1/8	1	1/5	2	1	1/4	
6	5	9	1/6	1/4	9	1		6	4	9	1/5	4	4	1	

		<b>DM3</b>						<b>DM4</b>							
		1	2	3	4	5	6			1	2	3	4	5	6
1	1	1	3	8	1/7	1/5	1/5	1	1	1/7	1	5	4	1/4	
2	1/3	1	3	8	8	1/9		2	7	1	1/7	1/5	1	1	
3	1/8	1/3	1	2	1/9	6		3	1	7	1	2	5	5	
4	7	1/8	1/2	1	2	1/4		4	1/5	5	1/2	1	1/2	4	
5	5	1/8	9	1/2	1	1/4		5	1/4	1	1/5	2	1	1/4	
6	5	9	1/6	4	4	1		6	4	1	1/2	1/4	4	1	

Additionally from these 4 DMs we can demonstrate the analysis of their conflict that the approach enables. We glean knowledge from the objective space by calculating the level of conflict between each pair via the distance in the objective space between the

pair. By extracting the value of this measure for each pair of DMs, we can gauge and rank the amount of conflict between each pairing in the group. The values for each of the 6 pairings in Example 5.3 are shown in Table 5.8. From this we see that the most *agreeing* pair are DM2 and DM4 and that the most *disagreeing* pair are DM3 and DM4. By additionally calculating the averages for each pairing a DM is involved in, as shown in Table 5.9, we observe the overall “most agreeable” DM is DM2 and the “most disagreeable” is DM3.

Table 5.8: Example 5.2: Agreement values between each pair of DMs

DM Pair	Min TNJR
1&2	8.5
1&3	5
1&4	7.5
2&3	7.5
2&4	3
3&4	9.5

Table 5.9: Example 5.2: Average Agreement values of each DM

DM	Avg. TNJR
1	7.00
2	6.33
3	7.33
4	6.67

Given further weights for the 4 DMs of  $\{0.35, 0.25, 0.25, 0.15\}$ , (perhaps representing a management member as DM1, 2 team members as DM2 and DM3 and an intern as DM4) we show the objective space with the identified weighted global solution for the 4 DMs, from the view of DM1 and DM4 in Figure 5.10: Left. The amount of compromise for each DM for the identified weighted global solution is shown in Table 5.10 and the preference vector derived from this weighted solution is shown in Table 5.11.

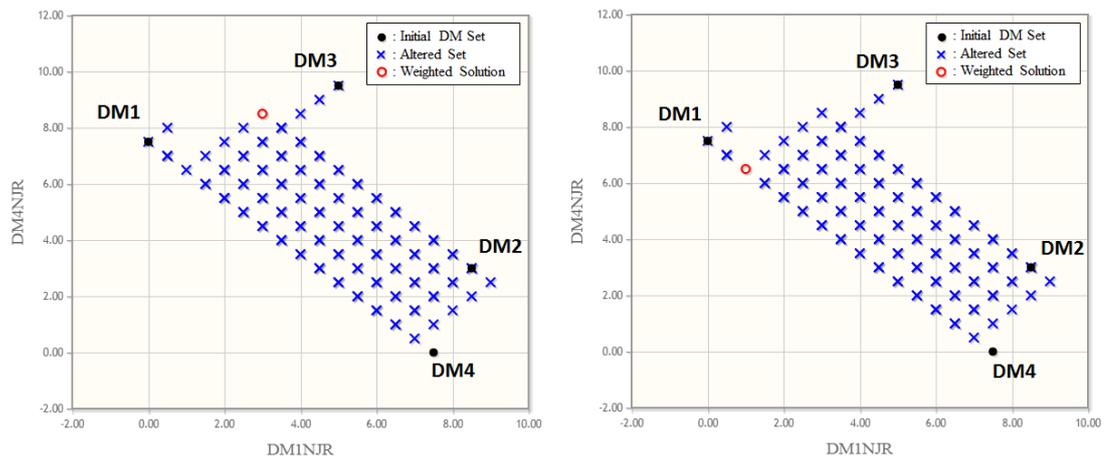


Figure 5.10: Example 5.2 Dynamic Weights Analysis DM1 & DM4 Views

We observe that DM1 undergoes much less compromise than DM4 as we would expect from their weights. Additionally, DM3 due to agreement with the strongest weighted DM DM1, as seen in Table 5.8, undergoes little compromise, seen in Figure 5.11: Left. The identified weighted global solution represents the global calculations of compromise of solutions that are weighted with respect to the DM weights<sup>11</sup>. These plots additionally help to highlight the similarity between pairs of DMs we calculated.

Table 5.10: Example 5.2: Weighted solutions compromise values

	DM1:NJR	DM2:NJR	DM3:NJR	DM4:NJR	TNJR
Weights Set 1	3	5.5	2	8.5	19
Weights Set 2	1	7.5	4	6.5	19

Then through alteration of the 4 DMs' weights to  $\{0.50, 0.20, 0.20, 0.10\}$ , increasing DM1's importance we see the effect upon the identified weighted global solution with the objective space shown from the viewpoints of DM1 and DM4 in Figure 5.10: Right and from the viewpoints of DM2 and DM3 in Figure 5.11: Right. We see the effects of these changing weights to the compromise to each DM's views in the weighted solution in Table 5.10. The preference vector derived from this weighted solution is shown in

<sup>11</sup> When weights of importance are employed the approach still ensures low overall compromise is sought as the weighted global value is a weighted calculation of the global measure of each solution. As opposed to consideration only of the weights without global compromise consideration - which would result in a more linear relationship between a DM's weight and the compromise they undergo within a weighted solution.

Table 5.11. We see DM1 now undergoes little compromise in the identified weighted solution.

This dynamic analysis of the weights illustrates how sensitivity analysis of DM weights can be carried out to analysis the impacts changes in the weights have upon aggregation. Such analysis is important as defining DM weights of importance is not a crisp task. Additionally as solutions found via the approach maintain the original judgment scale utilised for elicitation, analysis of multiple solutions can more easily be interpreted as to how a DM’s judgments have altered. Such dynamic analysis is not facilitated within the Weighted GMM approach as weights are defined before aggregation therefore changes of the weights cannot be considered. Furthermore as the judgments of a solution found via the Weighted GMM will invariably fall outside of the original judgment scale, interpretation of the judgments to analyse how a DM’s judgments have altered is problematic.

Table 5.11: Example 5.2: Weighted solutions preference vectors

	1	2	3	4	5	6
Weights Set 1	0.1288	0.2144	0.1706	0.0734	0.1230	0.2898
Weights Set 2	0.0522	0.3317	0.1692	0.1102	0.1355	0.2011

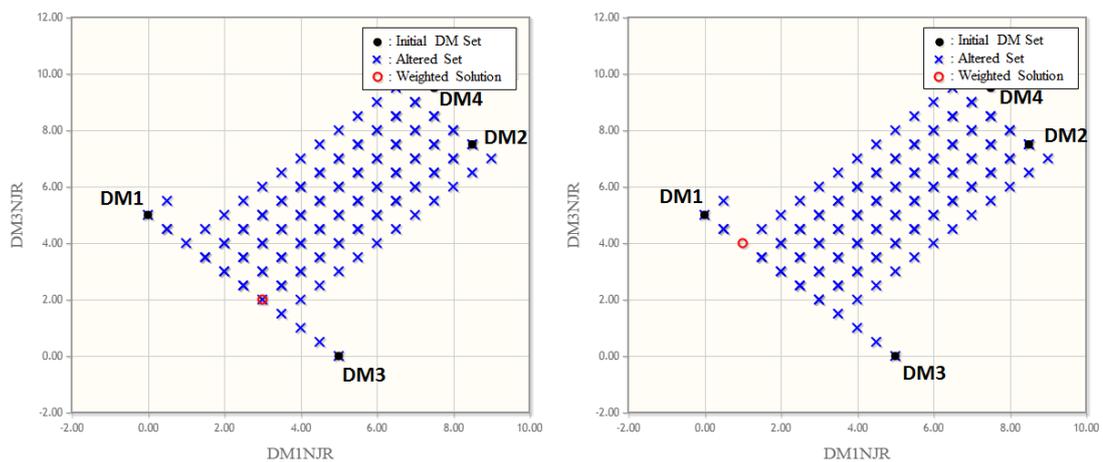


Figure 5.11: Example 5.2 Dynamic Weights Analysis DM1 & DM3 Views

### 5.3.3 Example 5.3: Interactively adding constraints

Example 5.3 explores how constraints can be iteratively utilised to aid a group of DMs to reach a consensus solution. Given a 4-element problem for 3DMs their initial judgments are shown in Table 5.12.

Table 5.12: Example 5.3: 3 DMs' Judgments

	DM1					DM2					DM3			
	1	2	3	4		1	2	3	4		1	2	3	4
1	1	4	2	1/3	1	1	1/4	1	1/3	1	1	1/3	1/6	1/7
2	1/4	1	7	1/4	2	4	1	1/7	8	2	3	1	8	1/2
3	1/2	1/7	1	3	3	1	7	1	3	3	6	1/8	1	3
4	3	4	1/3	1	4	3	1/8	1/3	1	4	7	2	1/3	1

With TJD as the measure of compromise for each DM and a maximum archive size of 50, the 3-dimensional objectives space, with respect to DM1 and DM2 is shown in Figure 5.12. In the plot the global compromise and fairest compromise solutions are plotted as is the GMM solution. The values of compromise of these 3 solutions, along with their total compromise and range values are shown in Table 5.13. Table 5.14 shows the compromise data for the initial DM sets, showing the amount of compromise needed from 2 DMs to match the views of a 3rd DM.

From the plot of the objective space in Figure 5.12 and the data in Table 5.13 and Table 5.14, we see that DM2's views differ significantly from those of DM1 and DM3 whose views are more similar. Consequently we see that the global solution of a total of 39 deviation steps require less compromise from the similar 2 DMs than the more outlying DM2. Further to this we see that due to the more outlying views of DM2 the global solution is located further away from DM2's initial judgments in the objective space than the fairest solution. For the global solution DM2 undergoes 6 more deviation steps than for the fairest solution. With regards to comparison with the GMM solution we see that the approach has identified a global solution in which the total compromise value represents almost 20% less deviation compromise than that of the GMM solution. Additionally we see that the approach has identified a single fairest solution in which the range of compromise to each DM's views is less than half the range of the compromise of the GMM solution. Additionally the total overall deviation value of this fairest solution is less than that of the GMM solution.

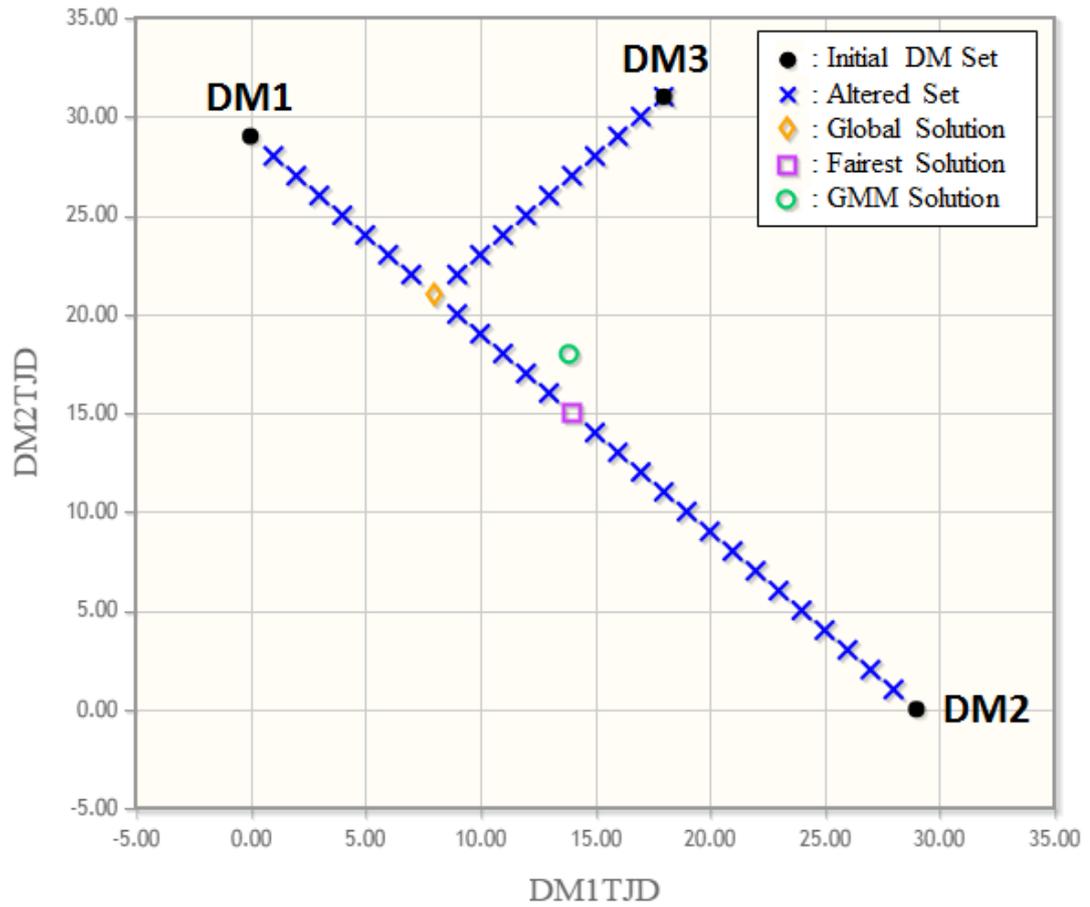


Figure 5.12: Example 5.3: Initial objective space DM1 and DM2 view

Table 5.13: Example 5.3: Initial significant solutions data

	DM1	DM2	DM3	<b>Total</b>	Range
GMM	13.86	17.98	16.14	<b>47.98</b>	4.12
Global	8	21	10	<b>39</b>	13
Fairest	14	15	16	<b>45</b>	2

Table 5.14: Example 5.3: Initial judgment sets compromise

	DM1	DM2	DM3	Total
DM1	0	29	18	47
DM2	29	0	31	60
DM3	18	31	0	49

So with this knowledge gleaned from the initial objective space and from the identified solutions, feasible constraints can be defined by the DMs. For example, if DM2 concedes that as their views are the most outlying they may choose to set a minimum constraint on their deviation equal to that of the fairest solution of 16 but also set a maximum constraint of deviation they are willing to tolerate of 25. We see the constrained objective space with DM2's constraints added in Figure 5.13: Left.

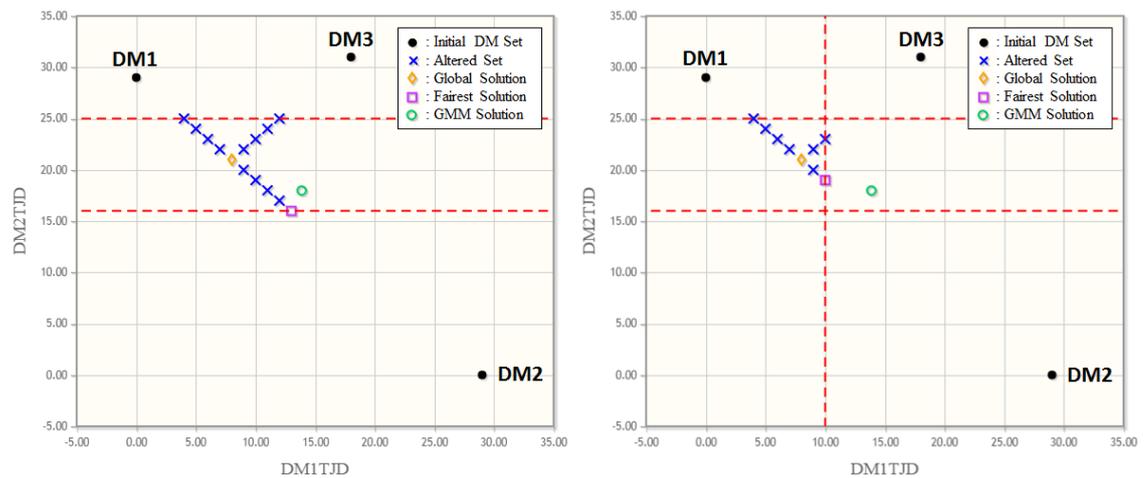


Figure 5.13 Example 5.3: Left: DM2 constraints. Right: DM2 and DM1 constraints

Additionally assume that DM1, by analysing the initial objective space, conjectures that as within the global solution their views would only undergo 8 deviation steps they will add an upper constraint to their deviation with some concession to this global value of 10. We see the constrained objective space with DM2's and DM1's constraints added in Figure 5.13: Right. From this constrained objective space we see that the GMM solution now falls outside the constrained sub-region of the objective space and highlights its inability to deal with constraints within aggregation problems. Furthermore within this constrained objective space the approach has identified a new fairest compromise solution, redefined based upon the solutions that are within the sub-region of the objectives space.

Next assume that DM3, seeing how his/her views would undergo 10 deviation steps within the global solution decides to relax only a little his/her views and defines a maximum amount of deviation of only 11. We show the constrained objective space with DM2's and DM1's constraints added from the viewpoints of DM1 and DM3 in Figure 5.14: Left. The updated constrained objective space now with DM3's constraint of 11

added is shown in Figure 5.14: Right. We see that, within this constrained objective space with constraints set by each DM, there are only 5 solutions. Going back to view the objective space from the viewpoint of DM1 and DM3 we see the 5 solutions shown in Figure 5.15.

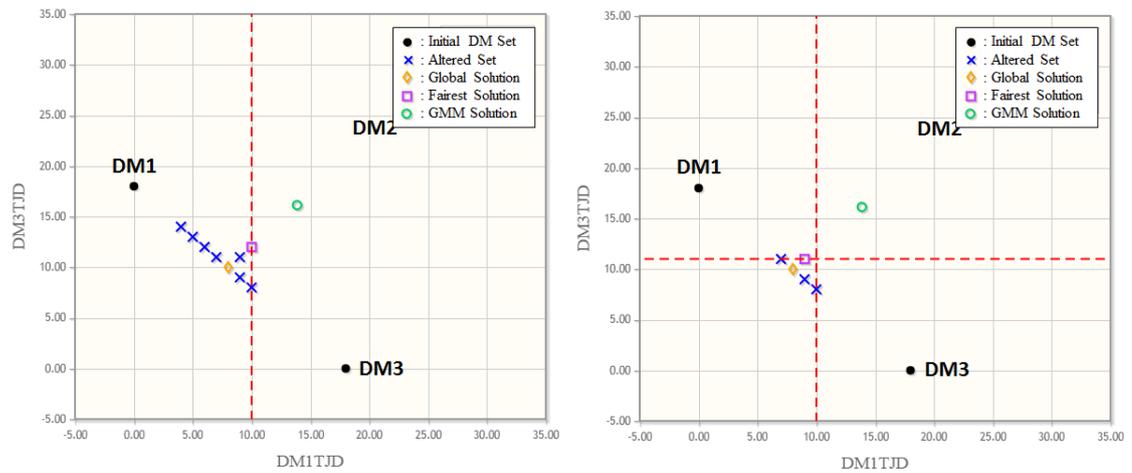


Figure 5.14 Example 5.3: Left: DM2 and DM1 constraints. Right: DM2, DM1 and DM3 constraints

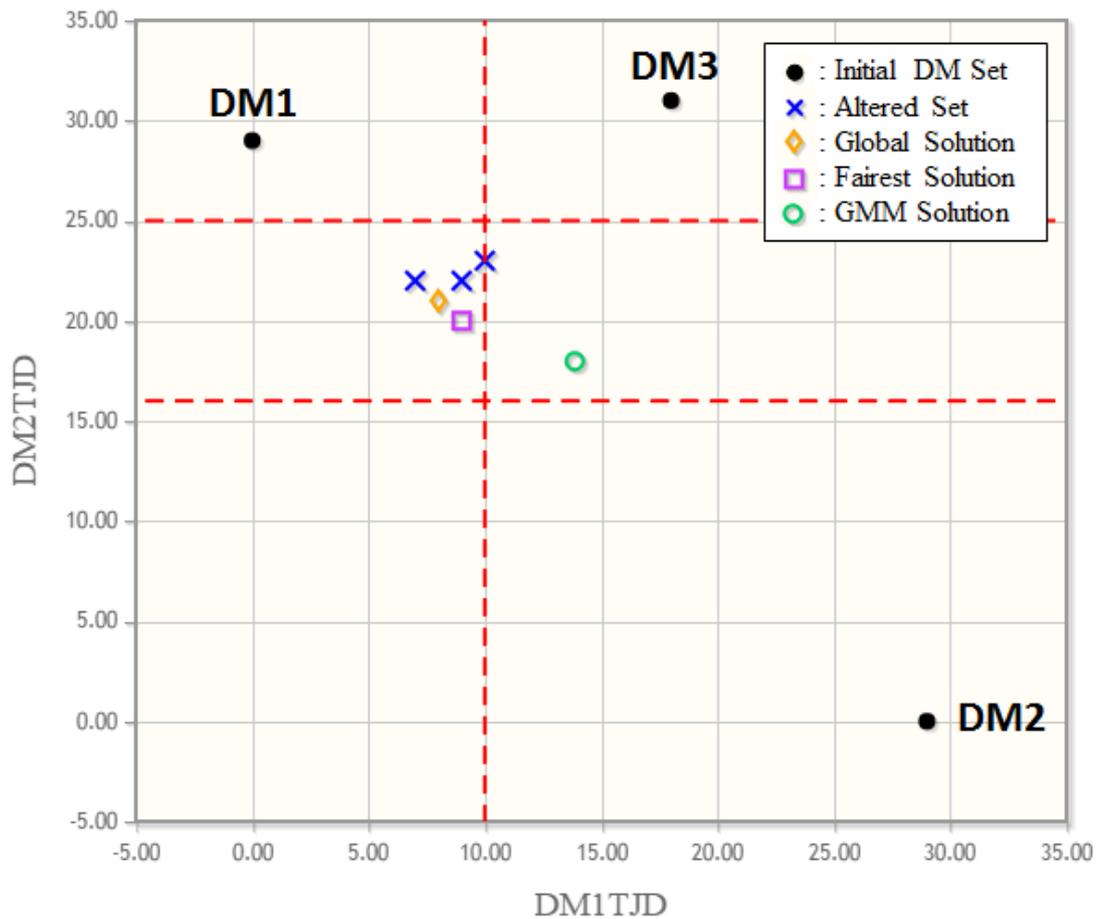


Figure 5.15: Example 5.3: Final Constrained objective space DM1 and DM2 view

Within this further constrained objective space we see the original global solution as well as another new fairest compromise solution, one that now is close to the global solution. For a solution set of this size within a constrained objective space it then becomes easier for the DMs to analyse and compare the solutions and select a consensus solution for aggregation. Additionally as the approach seeks solutions that maintain the original scale used to elicit judgments it is much easier for DMs to analyse the solutions and understand how their judgments have changed. Assume that the DMs are happy to choose the new fairest compromise solution as a consensus solution, here both DM1 and DM3 see a solution which is close to the optimal global value and DM2 sees a solution within the lower half of their deviation constraint boundaries. From this chosen solution a preference vector can then be derived. The chosen solution judgments along with the derived preference vector are shown in Table 5.15.

Table 5.15: Example 5.3: chosen solution and preference vector

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>w</b>
<b>1</b>	1	1/4	1	1/3	0.1202
<b>2</b>	4	1	7	1/2	0.4327
<b>3</b>	1	1/7	1	3	0.1810
<b>4</b>	3	2	1/3	1	0.2660

### **5.3.4 Example 5.4: Incorporating inconsistency reduction during aggregation**

Example 5.4 explores how various inconsistency measures can be utilised as additional objectives to seek to reduce inconsistency during the aggregation process, thus reducing the associated adverse effects upon preference vectors derived from aggregated solutions. Table 5.16 shows judgments from 2 DMs for a problem with 4 elements, along with initial inconsistency measures for the DMs. We can see that both DMs' judgments have a CR value above 0.5 and that both have 2 3-way cycles within their judgments.

The approach implements multiple inconsistency measures that can be used as objectives to reduce inconsistency and can be chosen based on the DMs' preferences: CR, L, CM and GCI. For example, assume that the DMs are interested in seeking low CR values within their aggregation and more specifically in ensuring that any preference vectors are derived from aggregated PCMs that have CR of 0.1 or less, thus ensuring maintenance of

Saaty's threshold of acceptability level. When the GMM is utilised for aggregation of these judgments it results in an aggregated PCM with of CR of 0.43, slightly less than the initial CR values of the DMs however still far from the 0.1 level of acceptable inconsistency. Therefore this problem with the DMs' additional inconsistency requirements cannot be incorporated within the GMM.

Table 5.16: Example 5.4 Judgments

DM1					DM2				
CR: 0.51 L:2					CR:0.52 L:2				
	1	2	3	4		1	2	3	4
1	1	1/4	1/5	3	1	1	2	1/4	2
2	4	1	2	1/2	2	1/2	1	2	1/3
3	5	1/2	1	2	3	4	1/2	1	1/7
4	1/3	2	1/2	1	4	1/2	3	7	1

Using the MOO approach with the measure of compromise of TJD as the objective for each DM and also with a 3<sup>rd</sup> objective of CR, the approach looks to find solutions with low TJD for each DM as well as solutions with low CR. To ensure that solutions with CR less than 0.1 are found, a constraint upper value of 0.1 can be added to the CR objective. The objective space for this 3-dimensional problem, with a large archive set to highlight the nature of the front, is shown in Figure 5.16: Left from the viewpoint of DM1 and DM2 deviation.

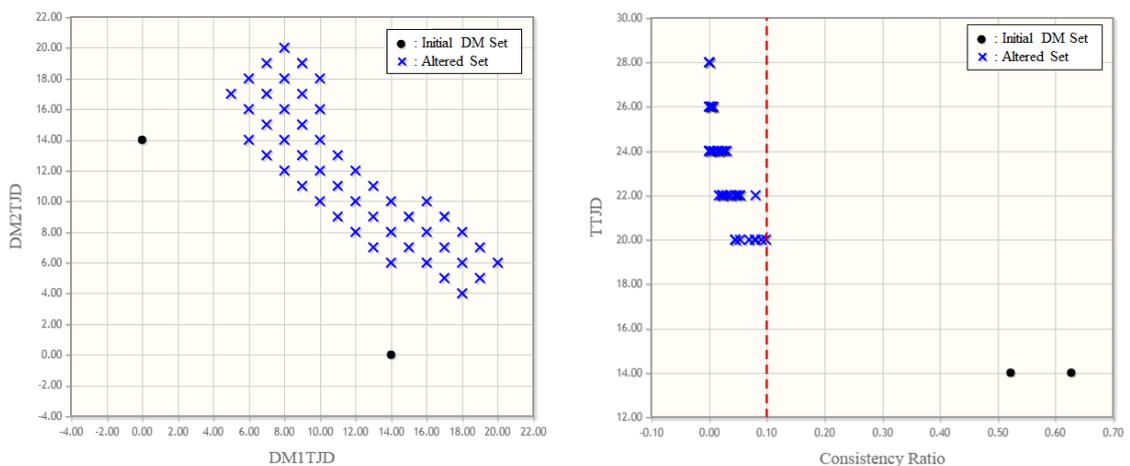


Figure 5.16 Left: Example 5.4 from DM1 and DM2 view (with CR as 3<sup>rd</sup> objective).

Right: Example 5.4 From CR and Total compromise view

The inconsistency constraint has ensured that all solutions found have a CR value below the 0.1 constraint. The 3-dimensional objective space from the viewpoint of the total deviation to both DMs (the sum of deviation for each solution found) and CR is shown in Figure 5.16: Right, along with the CR constraint as a dotted red vertical line. We see that only solutions with CR below 0.1 have been found. Therefore any solution can be selected by the DMs as a consensus solution and will have a CR below their constraint limit.

Alternately the DMs may instead be interested in seeking a low numbers of cycles within aggregated solutions. Again when the GMM is utilised for aggregation of these judgments it results in an aggregated PCM with an L value of 2, so the same number of cycles as originally present in each DM's judgments. The MOO approach can instead be utilised with an additional 3<sup>rd</sup> objective of L to seek aggregated solutions with a low numbers of cycles. Assume further that the DMs are only interested in aggregated solutions with all cycles removed and so can also set a constraint of 0 upon the L objective. The objective space for this 3-objective problem from the viewpoint of DM1 and DM2 deviations is shown in Figure 5.17: Left. The 3-dimensional objective space from the view of total deviation to both DMs and L is shown in Figure 5.17: Right, along the L objective constraint line. We see that only solutions with 0 cycles have been found (and that that each solution has a combined deviation measure of either 16 or 18). Therefore any solution can be selected by the DMs as a consensus solution and adhere to their requirements of having no cycles.

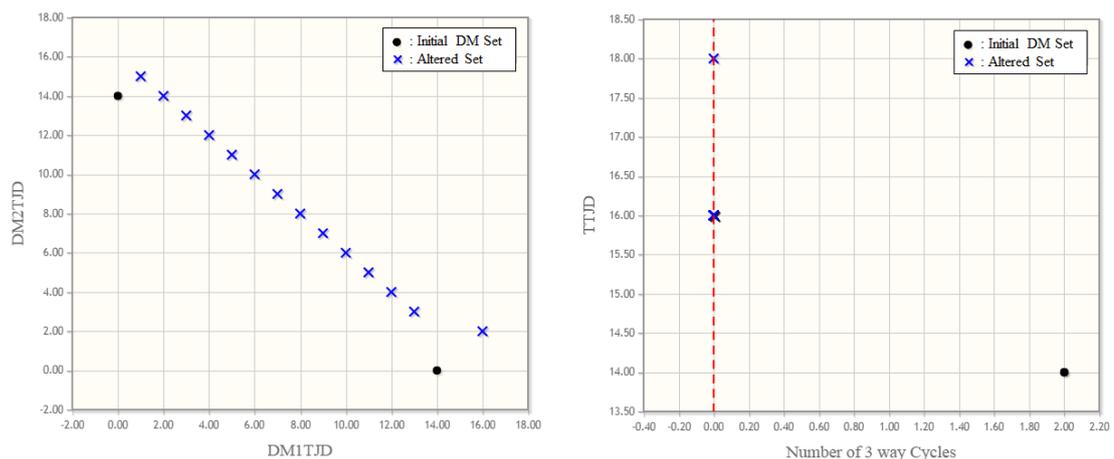


Figure 5.17 Left: Example 5.4 from DM1 and DM2 view (with L as 3<sup>rd</sup> Objective).  
 Right: Example 5.4 from L and Total compromise view.

## 5.4 MOO Approach to PC aggregation within AHP Decision

We present a step-by-step example to illustrate the use of the approach within an MCDA AHP decision problem [6]. The approach has been applied to the aggregation of PC judgments of a group of DMs whilst also looking to reduce inconsistency during the aggregation process. Firstly, an overview of the AHP decision problem is presented, this is followed by the utilization of the approach within the stages of AHP to derive a final ranking of the alternatives pertaining to the decision. The MOO aggregation approach is discussed and compared with the GMM aggregation approach during the decision problem stages.

The example explores a group of 3 DMs choosing a new renewable energy source within the community. For an overview and discussion of AHP see Chapter 2. First the formulation of the decision and its elements are presented. Following this judgment elicitation and analysis are performed. Analysis and aggregation of criteria priorities are presented followed by aggregation of alternative's priorities with respect to the criteria. Lastly a final synthesis and aggregated ranking of the alternatives are presented.

### 5.4.1 Formulation of the decision problem

In the example a group of 3 DMs, with equal weights of importance, are selecting a new renewable energy source within the community. When AHP is utilised for group decision making the formulation of the decision's elements may be defined by a single overseeing DM or via a more interactive approach between the DMs involved [18]. In this example we have an overseeing DM, and 5 criteria and 3 alternatives have been identified for which the 3 DMs' preferences will be elicited. The 3 alternatives are:

A1: **Wind Farming (WF)**

A2: **Fracking (F)**

A3: **Solar Panels (SP)**

The 5 criteria to which the alternatives are to be compared to are:

C1: **Community Impact:** The short-term and long-term impacts upon the community and land from the energy sources (**CI**);

C2: **Public Perception:** Perceived support and perceptions of the local community towards each type of energy source (**PP**);

- C3: **Infrastructure:** The set-up and deployment factors of each energy source along with accompanying legislation considerations (**I**);
- C4: **Costs:** Considerations of the costs of both initial setup and maintenance costs associated with each energy source (**Co**);
- C5: **Expansion Capacity:** The ease of future development and expansion capabilities of each energy source (**EC**).

With the elements of the decision defined we construct the AHP hierarchy, as shown in Figure 5.18, and elicit judgments from the DMs. For the aggregation at each elicitation stage within the approach the measure of compromise utilised will be NJV for each DM and an additional 4<sup>th</sup> objective of CR will be employed with an added upper constraint threshold of 0.1. This ensures that we seek at each aggregation stage only solutions with a CR below Saaty’s 0.1 recommendation. At each aggregation stage a Global aggregated solution(s) is identified (with DM weights of importance considered as equal) and utilised to derive a preference vector.

With the problem formulated and defined, judgments from the DMs can be elicited, and analysis and aggregation of the judgments can be performed. Next we firstly analyse the elicitation of the preferences of criteria from the DMs, then analyse the elicitation of DMs’ preferences between alternatives with respect to each criterion, however judgments may be elicited and aggregated in any order.

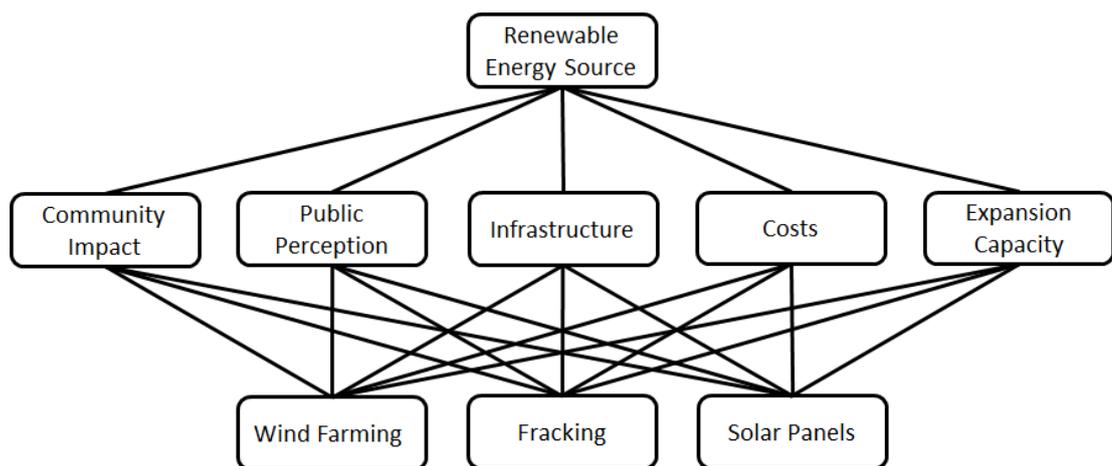


Figure 5.18: Hierarchy representation of the AHP example decision

### 5.4.2 Elicitation, aggregation and analysis of criteria priorities

To determine the importance of each criterion, we elicit judgments from the DMs with respect to the decision goal, as shown in Figure 5.19, and then derive a criteria preference vector.

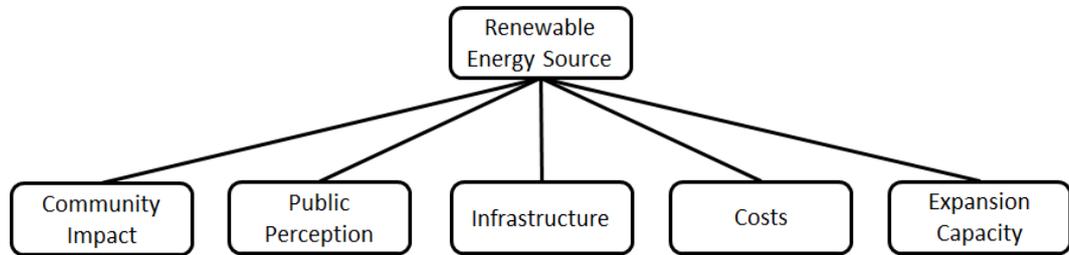


Figure 5.19: Criteria with respect to the problem

The three PCMs of preferences from the DMs for the importance of each criterion are shown in Table 5.17 along with CR values for each DM.

Table 5.17: PCM relating to criteria preferences of the three DMs

	DM1 CR: 0.54					DM2 CR: 0.5					DM3 CR: 0.7				
	CI	PP	I	Co	EC	CI	PP	I	Co	EC	CI	PP	I	Co	EC
CI	1	1/4	1/7	1	1/4	1	1/5	2	1	5	1	1/8	1/7	1/9	5
PP	4	1	1/2	1/2	6	5	1	1	3	1/3	8	1	1/4	1/2	1/3
I	7	2	1	1/2	3	1/2	1	1	1/2	1/2	7	4	1	1/2	6
Co	1	2	2	1	1/6	1	1/3	2	1	4	9	2	2	1	1/2
EC	4	1/6	1/3	6	1	1/5	3	2	1/4	1	1/5	3	1/6	2	1

As well as utilizing the MOO approach we can calculate a criteria preference vector using the GMM for comparison. For the GMM, the PCM created as the aggregation of this information is shown in Figure 5.18. We see that the level of cardinal inconsistency is above the 0.1 threshold level and consequently any preference vector derived from it will lack accuracy. However, the MOO approach, with its constrained CR objective, will only derive aggregated PCMs with CR values below the 0.1 threshold thus facilitating more accurate preference vector estimates to be derived. Additionally, as the approach seeks solutions that maintain the original judgment scale employed by the DMs during elicitation, the aggregated solutions will be easier for the DMs to discern how their judgments have altered.

Table 5.18: Aggregated PCM from GMM of the criteria preferences

CR: 0.18

	<b>CI</b>	<b>PP</b>	<b>I</b>	<b>Co</b>	<b>EC</b>
<b>CI</b>	1	0.18	0.34	0.48	1.84
<b>PP</b>	5.43	1	0.5	0.91	0.87
<b>I</b>	2.9	2	1	0.5	2.08
<b>Co</b>	2.08	1.1	2	1	0.69
<b>EC</b>	0.54	1.14	0.48	1.44	1

The preference vectors relating to the criteria weights derived from the MOO approach and GMM are shown in Table 5.19, with the most important criterion for each approach shown in bold. We see that the most important criterion calculated from the GMM is Infrastructure, whereas the most important criterion calculated from the MOO approach is Costs. As we have seen the variation in the weights can be attributed to the MOO approach being able to consider the additional information regarding inconsistency reduction.

Table 5.19: Aggregated criteria weights from MOO and GMM

	<b>Community Impact</b>	<b>Public Perception</b>	<b>Infrastructure</b>	<b>Costs</b>	<b>Expansion Capacity</b>
<b>GMM</b>	0.107	0.221	<b>0.272</b>	0.239	0.160
<b>MOO</b>	0.121	0.124	0.241	<b>0.355</b>	0.159

### 5.4.3 Elicitation, aggregation and analysis of alternatives priorities with respect to each criterion

For the 3 alternatives we then elicit judgments from the DMs of their preferences between the alternatives for each criterion. We can analyse the calculation of preferences of the DMs' preferences between the alternatives with respect to the Expansion Capacity criterion, as shown in Figure 5.20.

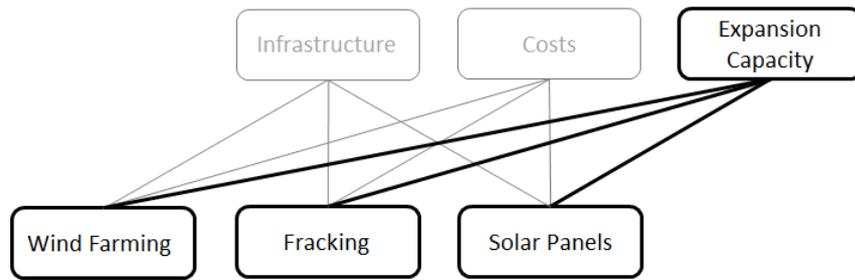


Figure 5.20: Alternatives compared with respect to Expansion Capacity criterion

The preference vectors derived from the DMs’ preferences of the alternatives with respect to the Expansion Capacity criterion, using the MOO approach and the GMM, are shown in Table 5.21. The aggregated PCM calculated during the GMM process is shown in the right hand side of Table 5.20. We see that the GMM aggregated PCM solution has a CR value greater than 0.3, again above the threshold of acceptable inconsistency. Conversely any aggregated PCM calculated during the MOO approach, with the added constraint upon the CR objective, will find PCMs below the 0.1 threshold. The single solution with the lowest overall compromise found with the MOO approach is shown in the left hand side of Table 5.20, which has a CR value of 0.

Table 5.20: MOO and GMM aggregation of alternatives with respect to Expansion Capacity

	<b>MOO</b> CR: 0.0				<b>GMM</b> CR:0.33		
	<b>WF</b>	<b>F</b>	<b>SP</b>		<b>WF</b>	<b>F</b>	<b>SP</b>
<b>WF</b>	1	1/2	2	<b>WF</b>	1	0.32	1.12
<b>F</b>	2	1	4	<b>F</b>	3.17	1	0.63
<b>SP</b>	1/2	4	1	<b>SP</b>	0.88	1.59	1

We see the effects of this in Table 5.21 with the MOO approach producing a differing ranking to the GMM approach regarding the 2<sup>nd</sup> and 3<sup>rd</sup> most preferred alternatives.

Table 5.21: Alternatives preference vectors with respect to Expansion Capacity

<b>C5</b>	<b>Wind Farming</b>	<b>Fracking</b>	<b>Solar Panels</b>
<b>GMM</b>	0.229	0.408	0.363
<b>MOO</b>	0.286	0.571	0.143

In the same manner, we can elicit judgments from the DMs of the alternatives with respect to the other 4 criteria, and then derive aggregated preference vectors using the MOO approach, as shown in Table 5.22, with the most preferred alternative for each criterion shown in bold.

Table 5.22: Aggregated preference vectors for alternatives with respect to criterion

	<b>Wind Farming</b>	<b>Fracking</b>	<b>Solar Panels</b>
<b>Community Impact</b>	0.122	0.320	<b>0.558</b>
<b>Public Perception</b>	0.165	0.225	<b>0.610</b>
<b>Infrastructure</b>	<b>0.661</b>	0.131	0.208
<b>Costs</b>	<b>0.493</b>	0.196	0.311
<b>Expansion Capacity</b>	0.286	<b>0.571</b>	0.143

#### 5.4.4 Synthesis to a final ranking

With all judgments elicited and preference vectors derived, the next stage is the synthesis of these preference vectors into a final ranking of the alternatives. The resulting final ranking of the alternatives produced via the MOO approach and via the GMM approach are presented in Table 5.23, with the most preferred alternative for each approach shown in bold. From these rankings from the 2 approaches we see that a differing final ranking has been determined from the MOO approach compared to the GMM. The highest ranking alternative from the GMM is Solar Panels; whereas for the MOO approach Wind Farming is the highest ranking alternative. Considering these differing final rankings, we have seen from the intermediate stages of the aggregation process that the MOO approach considers explicit consideration of minimising alteration and incorporates inconsistency reduction, thus provides a richer ranking.

Table 5.23: Final alternatives rankings for MOO and GMM approaches

	GMM	MOO
<b>Wind Farming</b>	0.367	<b>0.415</b>
<b>Fracking</b>	0.256	0.259
<b>Solar Panels</b>	<b>0.377</b>	0.326

#### 5.4.5 AHP Example conclusions

The AHP example has shown that the MOO approach can be utilised within a full AHP hierarchy decision problem and how through its explicit incorporation of inconsistency reduction during aggregation a richer ranking can be obtained. Moreover the MOO methodology facilitates an indicator-based approach to the aggregation process, in that DMs can decide how compromise during aggregation will be measured through choice over the measure of compromise to use. In the example NJR was utilised, however another measure such as TJD for example could have been utilised. Likewise how inconsistency is to be measured (if at all) can again be user-selected: in the example the CR objective was utilised, however different measures such as L could also or instead have been utilised.

There are additional analysis benefits that the approach facilitates that could be used in such a decision problem. Knowledge measures could be utilised at each aggregation stage to glean knowledge of the conflicts between the DMs. Constraints on measures of compromise could be employed at an aggregation stage to help drill down towards the selection of a single solution from which to derive a preference vector. Moreover for scenarios with high levels of conflict in which DMs are looking to ensure fair compromise then analysis of the fairest compromise could be incorporated at each aggregation stage. Furthermore analysis of non-equal weights of importance of each DM could be employed. For example, during the process it could become clear that one of the DM's expertise is such that their opinions should have more weight. As at each aggregation stage we are using the calculated global solution information to derive preference vectors, we could additionally utilise DM weights and analyse the final rankings and intermediate rankings, and perform some sensitively analysis on the already derived MOO fronts.

## 5.5 Conclusions

In this chapter we have presented a new approach to the aggregation of PC judgments of a group of DMs through aggregation of individual judgments via MOO, whilst simultaneously facilitating inconsistency reduction during aggregation. Through modelling alteration to each DM's views as a separate objective the approach looks to achieve consensus for the optimally minimum amount of alteration to each DM's views. The approach looks to model the trade-offs between the compromises needed to each DM's preferences to find Aggregated Consensus Solutions. Alteration to each DM's views is considered through the use of the measures of compromise giving the group of DMs' control over how alteration is to be measured. The approach additionally can facilitate inconsistency reduction during the aggregation process through modelling inconsistency measures as additional optional objectives to seek to find Aggregated Consensus Solutions with reduced inconsistency. A range of both cardinal and/or ordinal inconsistency measures can be utilised giving the DMs control over the type of inconsistency reduction to seek. We have defined and discussed the stages of the approach, the objectives that are usable within the approach and the various analysis the approach facilitates to help a group of DMs towards a consensus solution. Such analysis can be through the DMs iteratively adding constraints, to represent their tolerance level of compromise, to drill down into the objective space to help identify a consensus solution. The use of compromise measures within the approach aids the DMs to set constraints that are semantically meaningful and relatable to the degree of compromise they are willing to tolerate. Additional analysis of the global amounts of compromise, incorporation of DM weights of importance and analysis of the fairest levels of compromise can also help a group of DMs in reaching a consensus solution. Furthermore the approach is independent of a specific prioritization method so any method can be utilised to derive a ranking from Aggregated Consensus Solutions found, enabling the approach to be flexible to different problem scenarios and DM preferences. The approach implements the 1-9 scale to elicit judgments and this scale is utilised for the examples discussed in this chapter however the approach could be extended to use any bounded scale which would again facilitate flexibility for different problem scenarios and DM preferences regarding judgment scales. Additionally as the Aggregated Consensus Solutions found adhere to the original scale utilised to elicit judgments they allow a DM to discern more easily how their judgments have changed. We have presented multiple examples to illustrate the advantages and flexibility of the approach and to compare the approach to the GMM for aggregation of individual judgments. Additionally we have

presented the use of the approach within a full AHP decision problem to derive a final ranking of the decision alternatives under consideration. In the next chapter we investigate how scalable the approach is for dealing with the aggregation of larger numbers of DMs, and how to address the associated issues.

# Chapter 6 Clustering Decision Makers with respect to similarity of views

## 6.1 Introduction

In Chapter 5 we proposed an approach for the aggregation of a group of DMs' PCs. This Chapter addresses the issue of scalability for the aggregation of a large group of DMs' PCs. First an investigation of applying the aggregation approach with increasingly larger numbers of DMs is used to identify scalability issues. Following this we propose an approach to the aggregation of a large group of DMs to overcome these issues. The stages of the approach are then discussed, which first utilises clustering to divide a large group of DMs into clusters of similar views. Then an average set of views from each cluster's members are derived and utilised as objectives within MOO, to reach a final group aggregation. Following discussion of the approach, examples are presented. Finally conclusions of the approach are presented.

## 6.2 Aggregation of Many Decision Makers

The approach to the aggregation of a group of DMs' views models each DM as a separate objective within a MOO framework. Each additional DM is then simply modelled via an additional objective, and theoretically the approach can scale to deal with any size of DM group. However in practice, as the size of the set of objectives used within the MOO process becomes large, scaling issues come into play. This is due to the nature of MOO problems tackled via a MOGA when the size of the objective set becomes large. The issues relating to large objective sets are discussed next.

### 6.2.1 Multi-Objective Optimisation with large objective sets

For MOO problems, when the number of objectives is more than half a dozen, the problem is sometime termed a *Many*-objective optimization problem [134]. For such problems issues encountered may include:

1. **Increased dimensionality and size of the Pareto-front:** As the number of objectives increase the dimensionality of the objective space increases as does the size of the

Pareto-front. This results in a large number of non-dominated solutions to find, making it more difficult to accurately map out the front.

2. **Performance issues:** Within high-dimensional objective spaces performance issues surrounding stagnation in the search process can occur. In higher-dimensional objective spaces the proportion of non-dominated solutions in a population quickly becomes large making discerning between solutions in the population less effective. This then affects a MOGA's ability to diversely search over the objective space.
3. **High computational cost:** With an increased numbers of objectives, a commensurate increased amount of evaluation is required for each individual in each generation. Additionally higher computational costs will be incurred for archive maintenance.
4. **Difficulties in visualisation of the objective space:** as the dimensionality of the objective space increases it hinders interpretation and visualisation of the set of solutions found.

Consequently classical MOGAs do not perform well in many-objective optimisation problems [134]. Therefore we are likely to encounter such issues when we increase the number of DMs and consequently increase the number of objectives in the group aggregation approach. Next we explore the approach through experimentation over an increasing numbers of DMs.

### **6.2.2 Analysis of MOO Aggregation Approach for large group of DMs' PCs**

To help in evaluating the approach when utilised for aggregation of increasing larger groups of DMs we have implemented an exhaustive search of the aggregation of a group of DMs' PCs, to find the true fronts of problems. As with the exhaustive search outlined and used in our analysis in Chapter 4, an exhaustive search of group aggregation quickly becomes complex and intractable, see Chapter 4 for complexity discussion. However such complexity is with respect to the number of elements of the problem. Analysis can use relatively small values of  $N$  and increase the number of objectives without much complexity increase. We can therefore analyse the aggregation of increasingly larger groups of DMs each modelled as a separate objective, and assess how increasing the number of objectives affects MOGA performance, with respect to the usability and performance issues outlined above. Using Monte Carlo simulations we compare the average performance from a MOGA against the true fronts found via exhaustive searching over a range of objective sizes. For the experimentation, analysis of the performance for the aggregation of groups of DMs from 2 to 25 is performed. For each

$D$  value (2 to 25) 100 experiments were performed and then averages for each  $D$  value are calculated. In the results that follow the  $N$  value is 4 and the compromise objective for each DM is STJD.

Each experiment involves:

1. Creating random sets of judgments for  $D$  DMs;
2. Finding the true front of solutions for the data;
3. Using a MOGA calculate the average over 10 runs of the evaluation measures of the found solution sets.

For these experiments the MOCCell MOGA was employed (See Chapter 4 for overview of its operation) with parameter settings: population size of 100 (10 x 10 grid); maximum evaluations count of 25,000; with selection performed via binary tournament with single point crossover (with crossover probability 0.9) and bit flip mutation (with probability 0.01) employed. For the size of the archive for each experiment, as the true front of the problem is found first the size of the true front is used to define the size of the MOCCell archive, thus ensuring the archive is large enough to permit the finding of the whole front of solutions of the problem. Feedback is then defined as 25% of the size of the archive. For the MOGA performance analysis measures of these experimentations, using HyperVolume for increasing numbers of objectives becomes too costly as computing the hypervolume indicator is NP-complete [129]. Therefore we use Generational Distance (GD) and Inverted Generation distance (IGD) as performance measures in the experiments. GD [128] gives a measure of the distance between each found solution and the nearest true solution; IGD [128] gives a measure of the distance between each true solution and the nearest found solution. See Chapter 4 for discussion of these performance measures.

Next we discuss the results from these Monte Carlo experiments with respect to

1. Size of solution sets as the number of objectives increase;
2. Performance issues;
3. Increased computational costs;
4. Difficulties in visualisation of the objective space.

### 6.2.2.1 Effects of the size of solution sets as the number of objectives increase

From the experiments for each D value Figure 6.1 shows the average size of the true fronts over the range of D values as well as the average size of the fronts found by MOCell. From this we observe that the average number of non-dominated solutions quickly increases as the number of DMs increases and hence the size of the fronts become unwieldy. From a DM's perspective it becomes more difficult to analyse, objectively comprehend and extract knowledge from such a large set of solutions and to ultimately pick a single solution. For problems with such large solution fronts the population of the MOGA quickly fills up with non-dominated solutions which hinders the exploratory aspects of the algorithm. For example, Figure 6.2 shows an objective space when D=20 from the perspective of DM1 and DM2. We observe the population of solutions congregating within the middle of the objective space. Additionally, even in scenarios when only an approximate mapping of the front is desired (and therefore defining a smaller archive to find a smaller evenly spread set of solutions). A large number of solutions will still be required to be able to define an approximate mapping of the front.

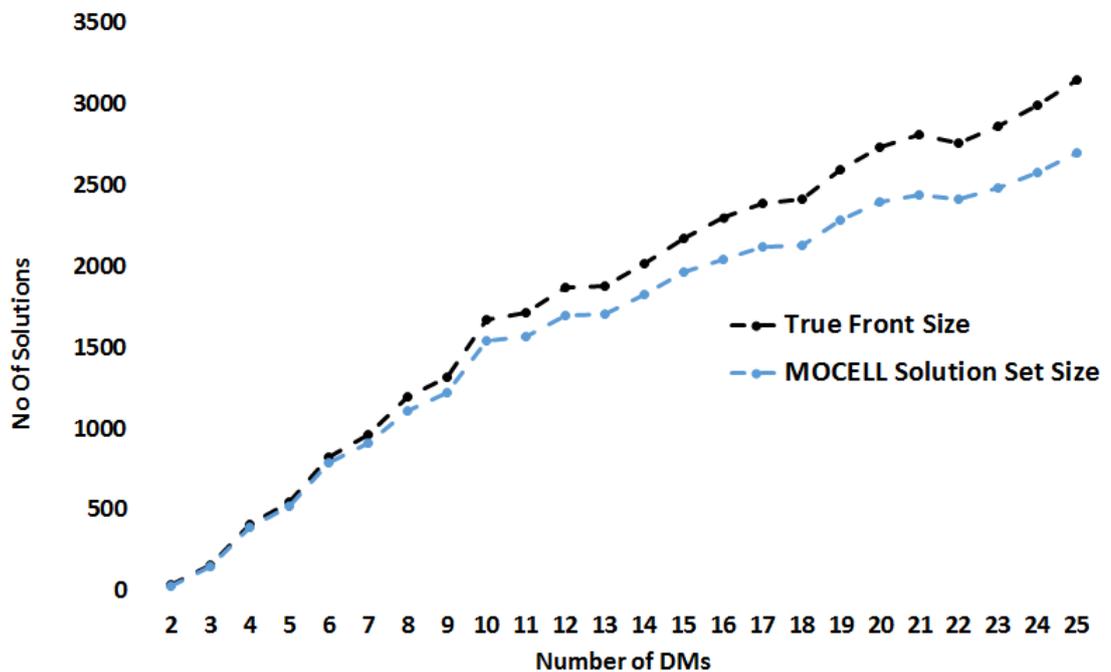


Figure 6.1: Average size of true fronts and average size of fronts from MOCell

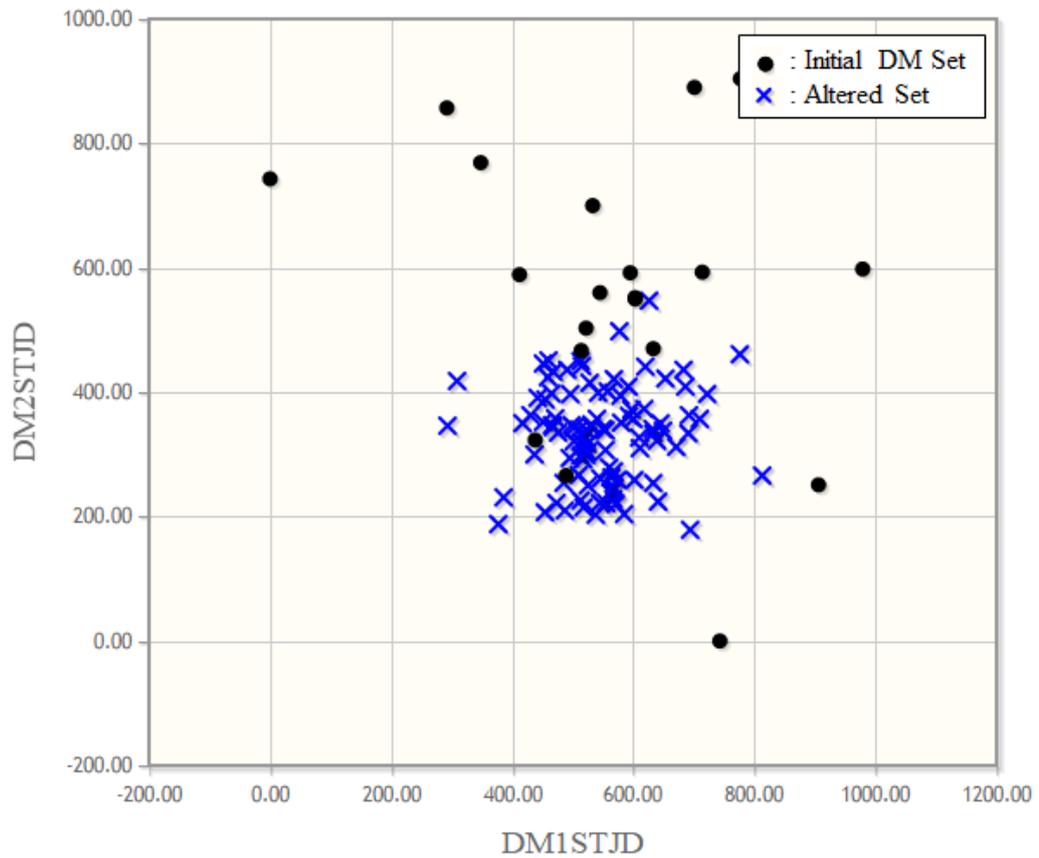


Figure 6.2: 20 DM objective space from DM1 and DM2 view

### 6.2.2.2 Performance issues

Figure 6.1 also plots the average size of the solution sets found by MOCell. From this we observe that as the number of objectives increases the size of solution sets found by the MOGA compared to the size of the true solution sets decreases, with the gap widening as the value of  $D$  increases. We visualise this by plotting the percentage of solutions found by MOCell compared to the size of the true front as shown in Figure 6.3. Here we observe that the performance - with respect to the percentage of the sizes of the fronts that has been found compared to the size of the true fronts - decreases as the number of objectives increases. Additionally the decrease appears fairly constant and suggests that the percentage of true front of solutions found will continue to decrease for higher values of  $D$ <sup>12</sup>. Furthermore, we evaluate the average performance over the range of  $D$  values using the GD and IGD measures. The average of the experiments for each  $D$  value of these measures is shown in Figure 6.4. We observe that as the number of objectives increase both GD and IDG increase, with IGD increasing faster (which we can attribute to finding

<sup>12</sup> Interestingly the performance is surprising when  $D=2$ . This is unnoticeable in Figure 6.1 as the size of the fronts when  $D=2$  are so small relative to the larger  $D$  values. These results could be explored in further investigations, see Chapter 8.

smaller percentages of the true fronts as we saw in Figure 6.3). Therefore as the number of DMs increase we see that performance deteriorates. In percentage terms the deterioration of the values approximately doubles every 7 D values. Again the increase in the plot appears fairly constant and suggests the deterioration of performance will continue for higher values of D.

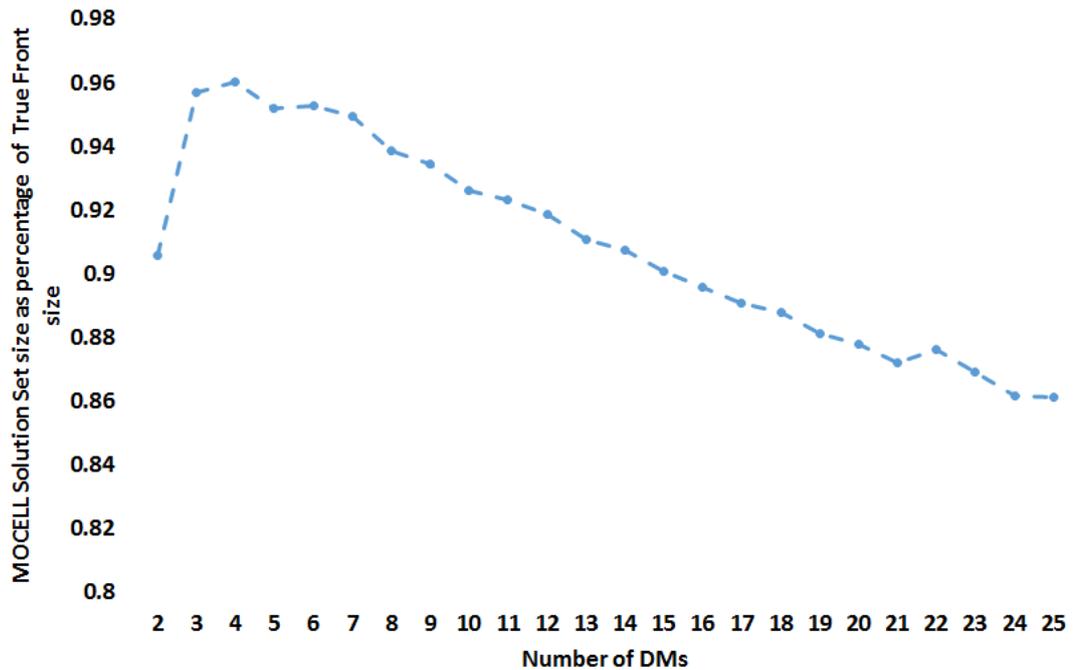


Figure 6.3: Average size of fronts found from MOCell as % of the size of true fronts

### 6.2.2.3 Increased computational costs

As the number of objectives increase the evaluations of each solution will become more costly. The evaluation costs for a solution will increase linearly with the number of objectives. Therefore a solution for a group of 20 DMs would take twice as long to be evaluated as a solution from a group of 10 DMs. Furthermore as the number of non-dominated solutions becomes very large additional costs regarding archive maintenance are incurred. Once an archive becomes large it becomes a costly process to determine if a new solution should be placed in the archive and which, if any, solutions already in the archive need to be removed. Moreover even in scenarios when only an approximate mapping of the front is desired by defining a smaller archive a large number of evaluations will still occur when determining the most appropriate subset to be in the archive over the operation of the MOGA.

#### 6.2.2.4 Difficulties in visualisation of the objective space

As the number of objectives increase the objective space becomes a higher-dimensional space, consequently it becomes harder for a DM to visualise and interpret the results and thus extract knowledge, i.e. the curse of dimensionality. Even if we could instantly find all solutions of the true front with all solutions lying exactly on the true front, we would still struggle to extract knowledge and value pertaining to the group of DMs and their conflict from the set of solutions.

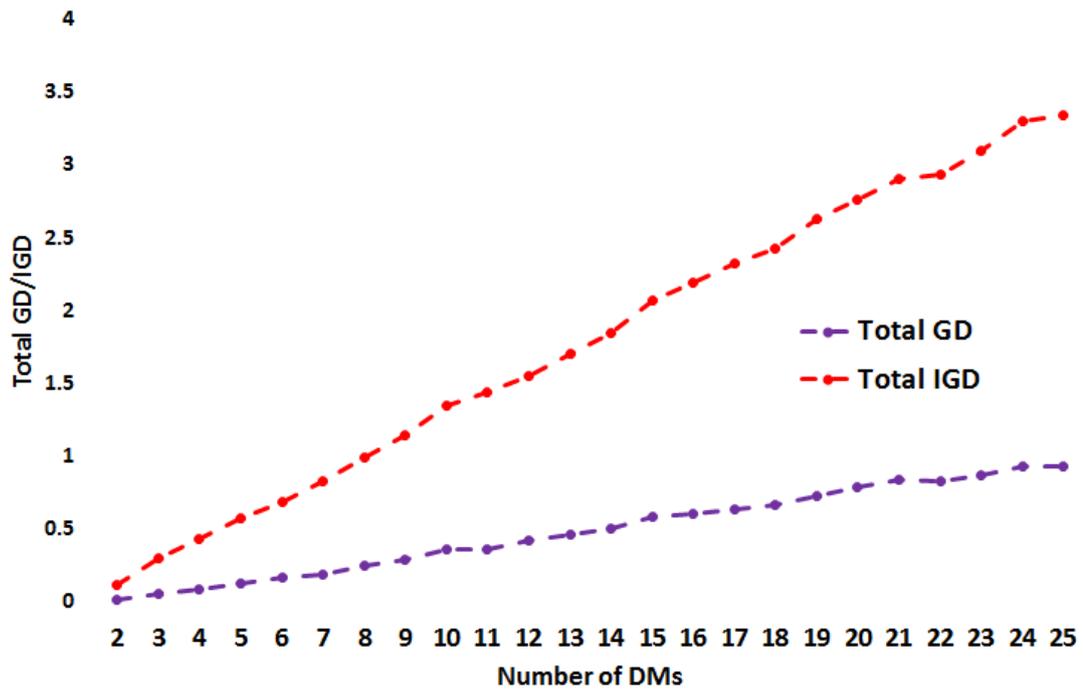


Figure 6.4 : Average GD and IGD of MOCeCell MOGA over range of DM values.

#### 6.2.2.5 Monte Carlo experimentation conclusions

From this experimentation we have highlighted the issues that occur when the number of DMs within group aggregation is scaled up. The experimentation results show constant degradation of performance and constant increase in true front sizes, suggesting that the issues will increase in a similar manner for larger group sizes than analysed. Techniques to tackle such scaling issues in the aggregation of large groups of DMs are outlined next.

### 6.2.3 Multi-objective optimisation with large numbers of objectives

We have identified the scaling issues of the aggregation of a group of DMs' views that occurs as the number of DMs increases. There are two types of approaches to tackling

MOO for large objective sets, those that maintain the full set of objectives and those that seek to reduce the number of objectives.

Approaches that maintain the full set of objectives look to overcome the scale issues through methods such as modification of the Pareto dominance relations to allow different rankings to be assigned to non-dominated solutions, or seek to utilise parallel search and aggregation methods, see [135] for a review of such approaches. These approaches still suffer from computational cost issues [136] and issues relating to diverse search effectiveness [134], as well as issues of extracting knowledge from high-dimensional objective spaces.

Alternatively various approaches have been defined that seek to reduce the number of objectives, see [137], [138], [139], [140]. Such approaches **trade-off between objective reduction and information loss**, via analysis of relationships between the objectives. From analysis of the objectives a subset can be found that preserve the problem's characteristics [123]. An approach utilizing Partition Around Medoids (PAM) clustering to reduce objectives was proposed in [134]. In this approach the objectives are clustered then the approach removes the most redundant objective identified. This process is iteratively applied removing a single objective at each iteration.

To tackle the scaling issue we propose an approach to the aggregation of PCs of a large number of DMs that looks to reduce the number of objectives for the least amount of information loss. The objective reduction is achieved through clustering DMs into similar views and then seeking a single representation of each cluster's views. The approach is presented next.

### **6.3 Approach for aggregation of a large group of DMs' PCs**

In the approach we firstly cluster a large group of DMs based on the similarity of their views. These clusters are used to calculate a single set of judgments per cluster to be used as a single objective within a MOO approach. The MOO is then performed with respect to each cluster of DMs modelled as a single objective.

As large groups of DM problems have large front sizes and higher-dimensional objective spaces tackling the issue by maintaining the full set of objectives (and looking to negate the performance issues) will not address the issues surrounding comprehension of such a large and high-dimensional objective spaces. Additionally seeking to first find redundancy between objectives to ensure only objectives in conflict remain makes for a more suitable input into a MOO process which seeks to find the trade-off front between these objectives [110]. By clustering the group of DMs we should additionally reveal

strong views within the group whilst aggregating and therefore expose some of the nature of the conflict within the group.

Previous approaches to redundancy reduction seek to derive a subset of the objective. When modelling each DM as a separate objective to remove objectives removes some DMs from the group. Instead the approach creates a set of judgments from each sub-group of DMs with similar views, which is a single set of judgments to represent the views of the members of the cluster.

The approach then treats each cluster judgment set as a single objective within MOO, thus reducing the number of objectives as each cluster is treated as a single objective. Therefore the approach tackles the scaling issues by reducing the size of the objective set. The set of trade-off solutions between the clusters is then sought via MOO. From all solutions found a preference vector can be derived that represents an aggregation of the whole group. From analysis of the clustering stage and the objective space we obtain a clear indication of the trade-offs between objective reduction and information loss, as well as indications of the nature of the group of DMs and their conflict. Additionally the approach incorporates weights of importance of each cluster based upon the number of DMs it contains. Subsequently we can identify the solution(s) on the front which represents the weighted global solution(s). Additionally through the stages of the approach we have clear traceability of the process from its start to the final group aggregated preference vector. The approach facilitates sensitively analysis of the stages, for example, selection of a different number of clusters or selection of a different solution from the front of solutions identified through MOO. Such sensitively analysis of the results and the process, aids knowledge extraction about the conflict and the views within the group as well as validation of the result. Regarding the logistical operation of the approach, for large groups of DMs it is envisaged an overseer is making choices such as the compromise measure used and performing “what k value to choose” analysis. In this way the approach is for when we want to pool a number of people’s opinions which an overseer can analysis and process to an aggregation. The stages of the approach are outlined next.

### **6.3.1 Stages of Approach for aggregation of a large group of DMs’ PCs**

The stages of the approach, shown in Figure 6.5, can be summarized as:

Stage 1: Clustering the group of DMs;

Stage 2: Deriving single representation for each cluster;

Stage 3: Multi-Objective Optimisation.

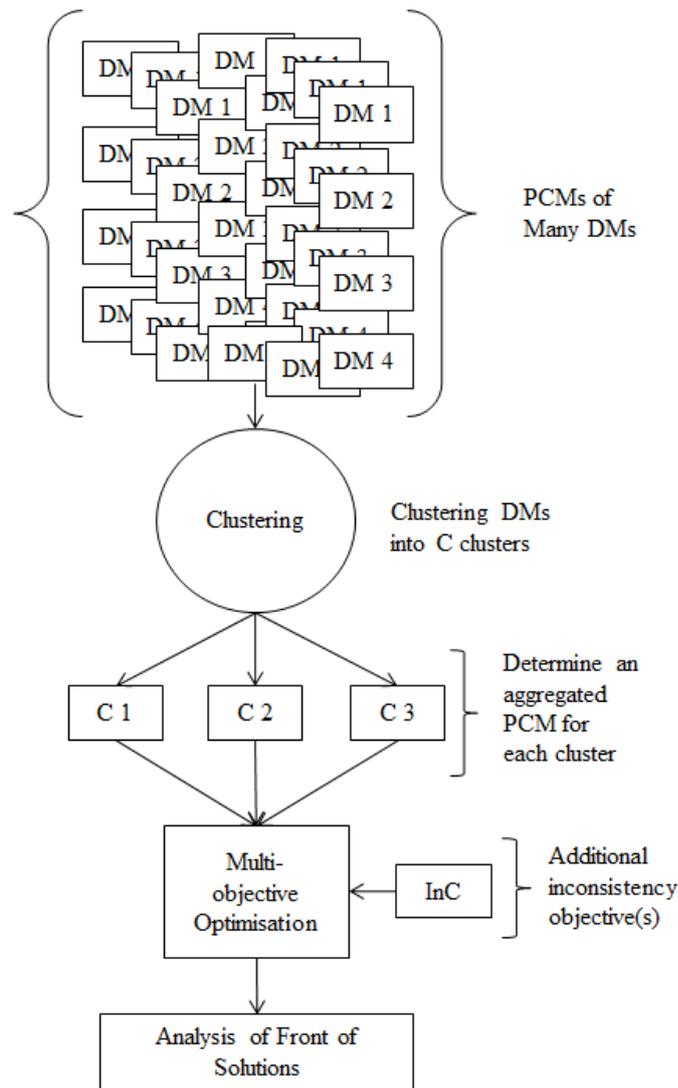


Figure 6.5: Flowchart of the approach to aggregation of a large group of DMs

### 6.3.2 Stage 1: Clustering the group of DMs

The first stage of the approach utilises clustering to divide the DMs into groups based upon the similarity of their views. The approach utilises the k-means++ algorithm to perform the clustering. K-means++ is therefore briefly outlined followed by an outline of the clustering stage of the approach.

#### 6.3.2.1 *K-means++ Clustering*

Clustering discovers natural groupings of a set of points or objects [141]. Such clustering can be performed by the k-means clustering approach. Given a set of  $d$  dimensional instances, k-means seeks to cluster the instances into a set of  $k$  clusters, such that the squared error between the mean point of a cluster and its points are minimised [142]. K-means is one of the most prominent clustering methods, requires relatively little

parameter tuning and has been identified as one of the top 10 algorithms in data mining [143]. The k-means algorithm has three stages:

1. The instances are assigned randomly into  $k$  clusters;
2. For each cluster the centroid between its members is calculated and then the distance each instance is from each cluster centroid is determined;
3. Each instance is then assigned to the cluster with the nearest centroid.

Stages 2 and 3 are repeated until no instances are assigned to a new cluster in Stage 3.

A limitation of k-means is that due to the instances initially being assigned randomly to clusters it is possible that a sub-optimal convergence will be reached. The k-means++ algorithm [144] is an enhancement of k-means that seeks to reduce this limitation through a modified initialisation stage that aims to initially assign the instances into clusters such that the initial clustering is closer to an optimal initialisation [144]. The k-means++ initialisation phase seeks to find initial cluster centers that are spread so that initial clusters are far away from each other. The k-means++ algorithm is utilised within the approach. When using the k-means++ algorithm (and for clustering in general) the selection of an appropriate value for the number of clusters is challenging [141]. The approach facilitates analysis to aid the selection of an appropriate  $k$  value, see Section 6.3.4.

### 6.3.2.2 Clustering DMs in the approach

Within the approach DMs are grouped based upon the similarity of their views regarding their judgments. Given a problem with  $D$  decision makers and  $n$  elements each DM defines a complete  $n$  by  $n$  PCM of their judgments

$$\{PCM_1, PCM_2, \dots, PCM_D\} \quad (6.1)$$

We extract from each PCM the top triangle of  $J$  judgments to represent each DM's views. For each DM a Judgment Set  $O$  of cardinality  $J$  can be selected, containing enough information to reconstruct the whole of the PCM. Using the judgment set encoding as defined in Chapter 3, the approach models an  $O$  representation of each DM's PCM  $\{O_1, O_2, \dots, O_D\}$ , each of which consists of  $J$  judgments  $\{o_1^k, o_2^k, \dots, o_j^k\}$ , for  $k=1, \dots, D$ . This set of  $D$  encoded judgment sets is utilised as the feature vector inputs for the clustering stage. An illustration of this for 3 DMs is shown in Figure 6.6. The set of  $D$  DMs are then clustered into  $C$  clusters using k-means++.

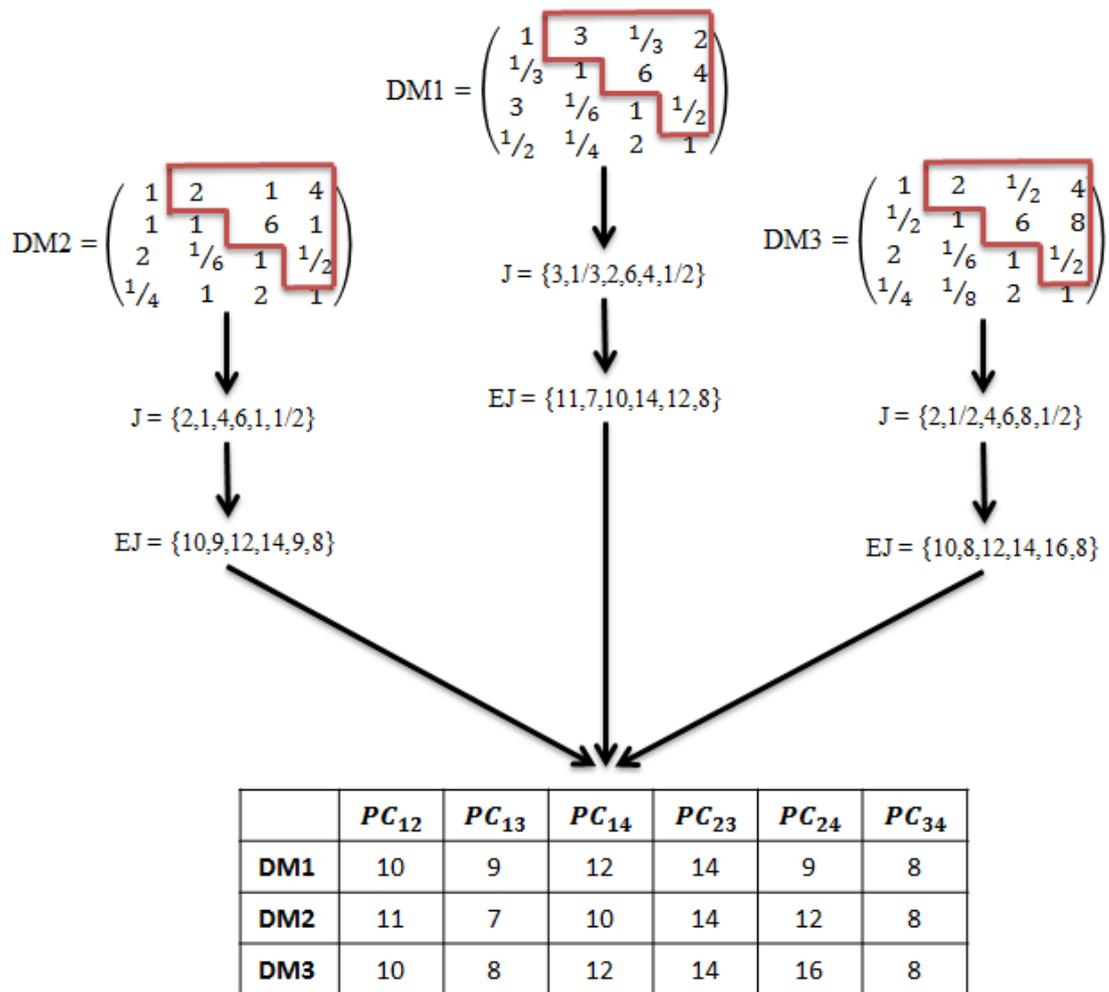


Figure 6.6: Illustration of clustering input for 3 DMs' judgments

### 6.3.3 Stage 2: Deriving a single representation for each cluster

With the DMs clustered into  $C$  clusters stage 2 of the approach then creates for each cluster a single judgment set that is a representation of its member's views. Each judgment set for each cluster is derived through single objective optimisation. For each cluster we perform single objective optimisation via a single objective genetic algorithm where the single objective is a total measure of compromise. As we saw in Chapter 5, we can calculate the total of a compromise measure across a group of judgment sets. The objectives usable in stage 2 are:

1. Total NJV
2. Total TJD
3. Total STJD
4. Total NJR

From  $C$  clusters, each of size  $S$ , we have a set of  $S$  encoded judgment sets from its members, against which we evaluate the single objective. The genetic algorithm evaluates solutions based upon the objective of total compromise. For example, for a cluster and a possible solution its TotalNJR can be calculated via:

$$TNJR = \sum_{i=1}^S \sum_{j=1}^J R_j \quad (6.2)$$

For example, given a cluster of size 3, its member's judgments are shown as DAGs in Figure 6.7 (as ordinal judgments for simplicity). Furthermore TNJR is to be used as the objective to find a single representation of these 3 DMs' views for the least amount of total compromise.

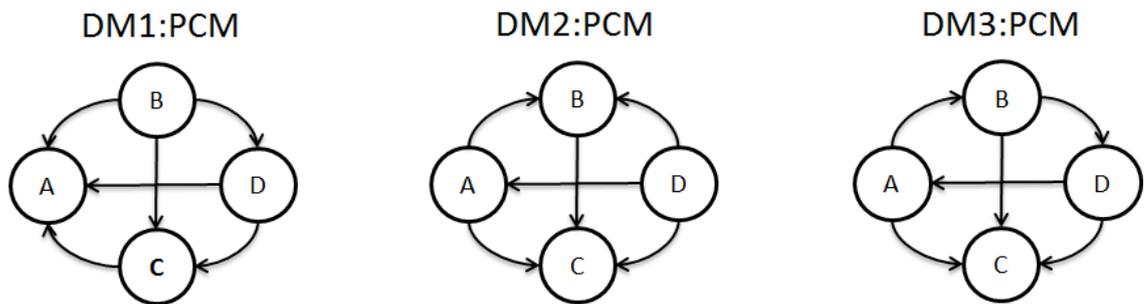


Figure 6.7: Example of 3 cluster member's judgment's as DAGs

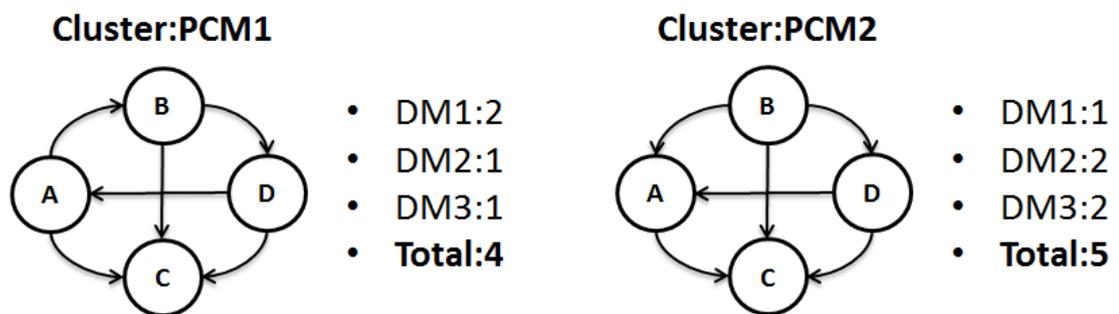


Figure 6.8: Examples of single representation judgment's as DAGs

Two such aggregated PCMs are shown in Figure 6.8. We see that for PCM2 a total of 5 reversals occur, yet for PCM1 a total of only 4 reversals will occur, hence it is a more preferred solution as a single representation of judgments for the cluster.

Therefore within each cluster we seek a representation of its members' views in a single judgment set for the smallest amount of overall compromise. As we have clustered DMs with similar views together in the same cluster, a single judgment set representation of a cluster's members should be derivable that reflects its member's views. At the end of stage 2, from the initial set of D DMs' PCMs, we have a set of C clusters and a set of C PCMs that are each a representation of the views of the members of a cluster.

Regarding the approach's clustering and subsequent creation of a judgments set for each cluster, additional consideration should be given regarding its effectiveness as a method to identify redundancy present within the objectives for a small amount of information loss. Furthermore consideration regarding how an appropriate k value can be chosen would be beneficial. The next section looks into these two aspects.

#### **6.3.4 Clustering evaluation experimentation**

Through Monte Carlo experimentation we now evaluate the approach's use of clustering as an effective way to seek redundancy to reduce the number of objectives with as little information loss as possible. Additionally such analysis also assists during operation of the approach to help an overseer to select an appropriate k value for the clustering stage.

Evaluation of the clustering of a group of DMs over a range of k-values can analyse the changing amount of total agreement within the clusters against the amount of objective reduction achieved. We determine a measure of the total agreement from the clustering stage through analysis of the amount of compromise needed in each cluster to create a single judgment set of the views of the members of the cluster, summed for all clusters. When  $k=1$  all the DMs will be in a single cluster and when a single representation of this single cluster is sought we also get a measure of the total compromise within the group. This is a useful measure for an overseer to see the total amount of compromise within the group for the given measure of compromise. For example, given a hypothetical group of 7 DMs giving judgments for an  $N=4$  problem. Figure 6.9: left shows these 7 DMs when  $k=1$ . Here the measure of compromise of TJD is utilised to seek the optimal single representation of the 7 DMs when they are clustered into a single cluster. Here we see that 58 deviation steps are required to create this representation. We also observe that as the number of objectives is reduced from 7 to 1 we have achieved an 85% objective

reduction. In a similar way we then measure the amount of compromise that occurs to reach cluster representations for larger k values, where the measure of each cluster's total compromise value is summed. For example, when k=2 as shown in Figure 6.9: Centre, the compromise in each cluster is 10 and 22, so 32 in total. Given that when k=1 we have a measure of the total compromise in the group - 100% of the compromise - we calculate subsequent k-value's compromise with respect to this initial k=1 value. For example, when k=2 the total 32 represents 55% of the original compromise. So we see that when k=2 45% less information loss occurs and also that an objective reduction of 71% will be achieved. Then again for when k=3, as shown in Figure 6.9: Right, we calculate that the total deviation steps across the 3 clusters is 22 which represents 37% of the original compromise when k=1. Furthermore k=3 results in an objective reduction of 57%.

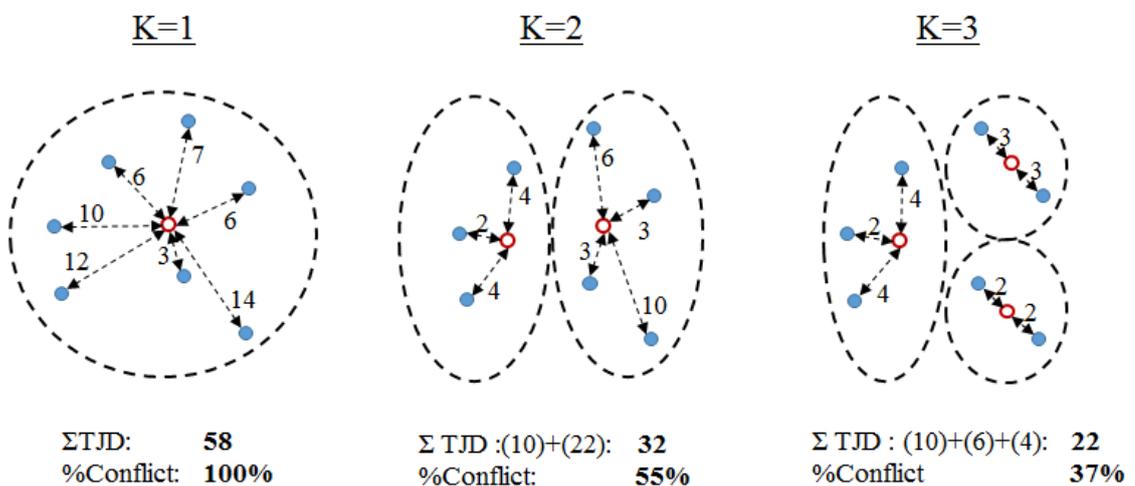


Figure 6.9: Evaluation of clustering values

We continue such analysis for further values of k all the way to D, and for each value measure the compromise and the amount of objective reduction, as shown in Table 6.1. When k=D each DM is assigned their own cluster and so each single representation would simply be the cluster's single member's judgments, and the total compromise would be 0 as would the percentage of objective reduction. In Figure 6.10 we plot the information of the trade-off between objective reduction and redundancy for this example. An overseer can then see the nature of the trade-off for this problem, as well as how the rate of redundancy changes over the range of k-values, to aid in selection of an appropriate k-value. From this an overseer might conjecture that after a k value of 3 the amount of reduction for subsequent k-values is no longer as valuable and so k=3 is chosen. Furthermore as the approach provides a traceable thread from initial DM views to final

aggregation the analysis of an appropriate k-value can also aid an overseer in performing sensitivity analysis, through selecting a different k-value and analysing how it affects the subsequent stages.

Table 6.1: Cluster analysis example data

k	Objective Percentage	Objective Reduction percentage	TotalTJD	Percentage of overall TotalTJD
1	14.29%	85.71%	58	100.00%
2	28.57%	71.43%	32	55.17%
3	42.86%	57.14%	22	37.93%
4	57.14%	42.86%	17	29.31%
5	71.43%	28.57%	12	20.69%
6	85.71%	14.29%	6	10.34%
7	100.00%	0.00%	0	0.00%

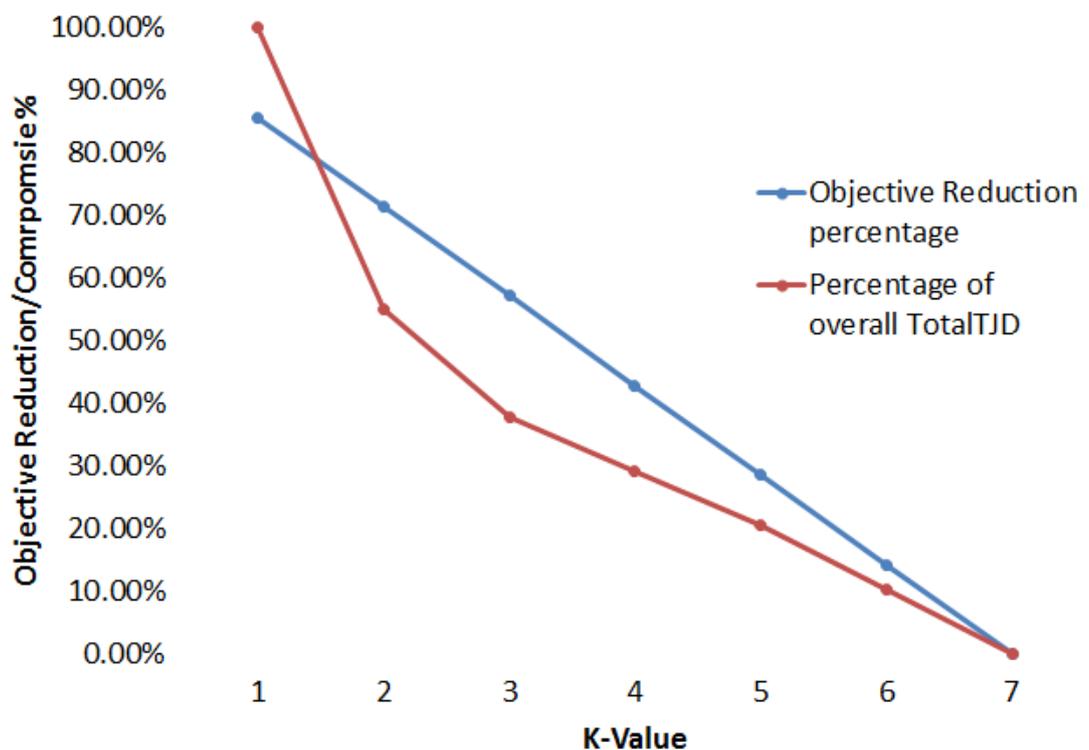


Figure 6.10: Analysis of objective reduction vs. information loss

This example shows analysis of a relatively small number of DMs. The analysis for a larger group of 50 DMs clustered using the approach is shown in Figure 6.11. Here  $N=4$  and TotalNJR has been utilised as the measure of compromise, the percentage of total compromise in the group for each k-value is plotted as a black dot and a trend line of the total compromise over the range is shown in red. In this plot an overseer can get a clear picture of the trade-off between objective reduction and information loss as well as how this trade-off changes over the range of k-values. Such analysis can aid them in selecting an appropriate k-value. Additionally from this example's plot we see that the approach has identified that for lower k values, a high amount of redundancy can be achieved from one k value to the next whilst also achieving a high level of objective reduction, thus ensuring the objective set can be reduced to a sufficiently small size to not suffer from scaling issues.

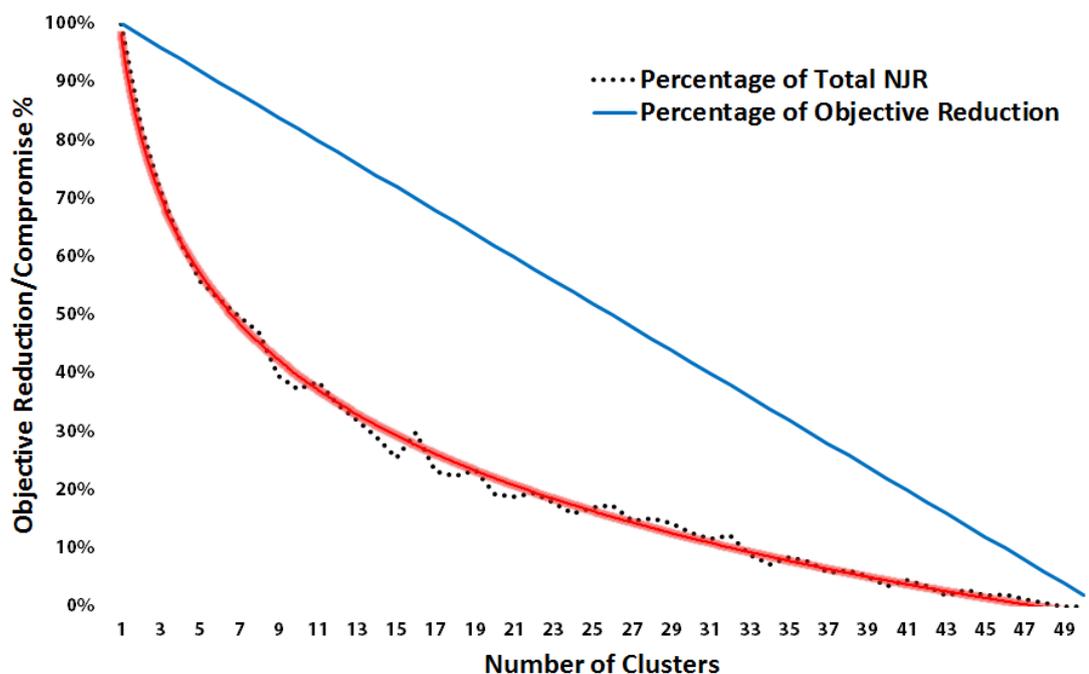


Figure 6.11: DM50 example analysis of objective reduction vs. information loss

This example shows the approach for a single set of DMs and their judgments over a single run of clustering. Utilizing Monte Carlo experimentation we analyse more generally the approach's abilities to identify redundancy. The experimentation will analyse trends regarding the trade-off between information loss and objective reduction for various group sizes (D) for a range of N values. For each N and D combination each experiment consists of:

1. Creation of random judgments for D DMs for an N element problem;
2. Performing 10 runs of clustering  $k=1$  to D and performing trade-off analysis of compromise and objective reduction for each run and deriving the averages over the 10 runs. Total NJR is utilised as the measure of compromise within the cluster analysis. Through averaging over 10 clustering runs for each data set the stochastic nature of the clustering is taken into account.

The parameters for the k-means++ clustering and single objective optimisation GA are the same as for the examples presented in Section 6.4

Then from all the experiments we derive averages of the compromise data to determine trends for the N and D combination. For each experiment new judgments are utilised and therefore for each the value of the amount of compromise when  $k=1$  will be different. However as for each experiment we are analysing over the range of k values with regards to the percentages of compromise in relation to the total compromise found when  $k=1$ , we average the data across these percentage values of the experiments. A flowchart of the Monte Carlo experimentation process is shown in Figure 6.12. Findings from the experimentation are presented next.

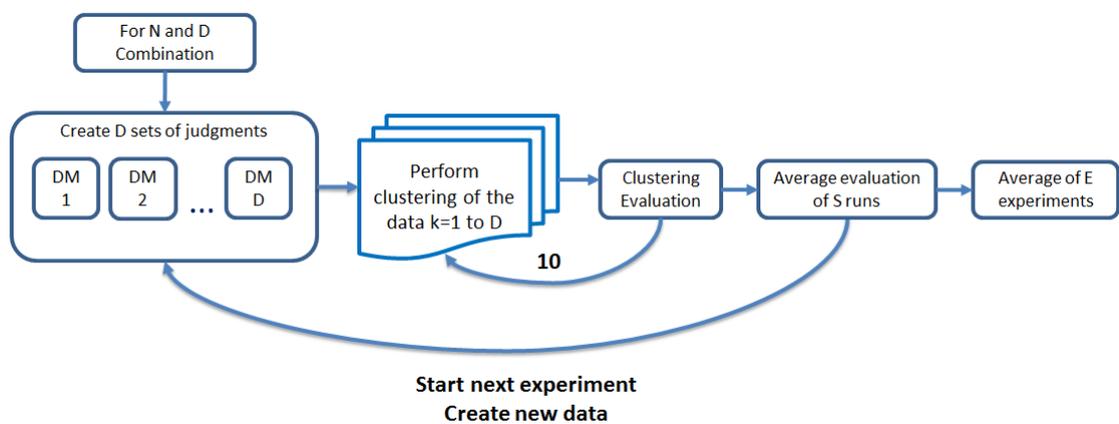


Figure 6.12: Monte Carlo experimentation process

Figure 6.13 shows experimental results from 50 experiments for  $N=3$  and  $D=20$ . The average over the experiments is shown as a red line and the upper and lower values of each k value over the experiments are shown to highlight the range between the values of the experiments. We see that for low k values the approach generally identifies high levels of redundancy from one k value to the next as the steepness of the gradient of the average compromise line shows. We additionally see that this tails off for larger k values.

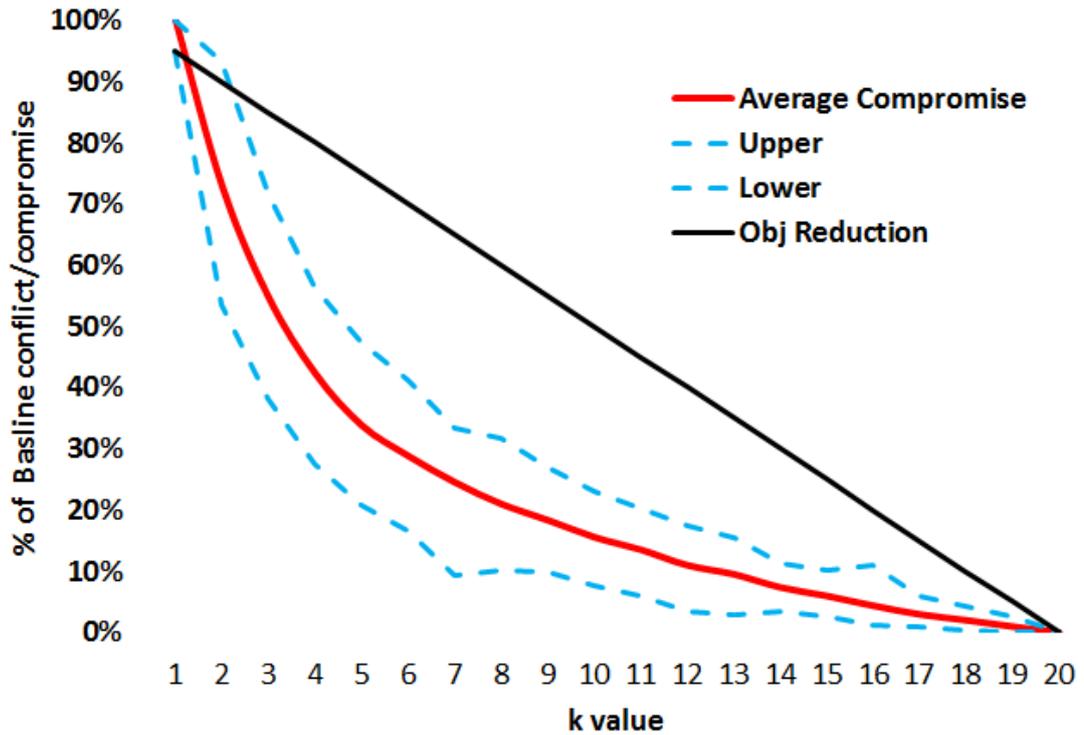


Figure 6.13: Average Redundancy N=3 D=20

We can visualise this through plotting the rate of change of compromise (and objective reduction) over the range of k-values as shown in Figure 6.14. Here we see that for a group of DMs there is high redundancy to be found for a small number of k-values and that the approach is effective in discovering it. Therefore the approach is effective when seeking to reduce an objective set such that it will suffer fewer scaling issues.

Experimentation has analysed various N values for each D value. Figure 6.15 shows the averages of the experiments for a range of N values from 3 to 5 when D=20. From this we see that for each N value the rate of change is highest for lower values of K but also that the higher the value of N the less pronounced this rate of change for lower k-values is.

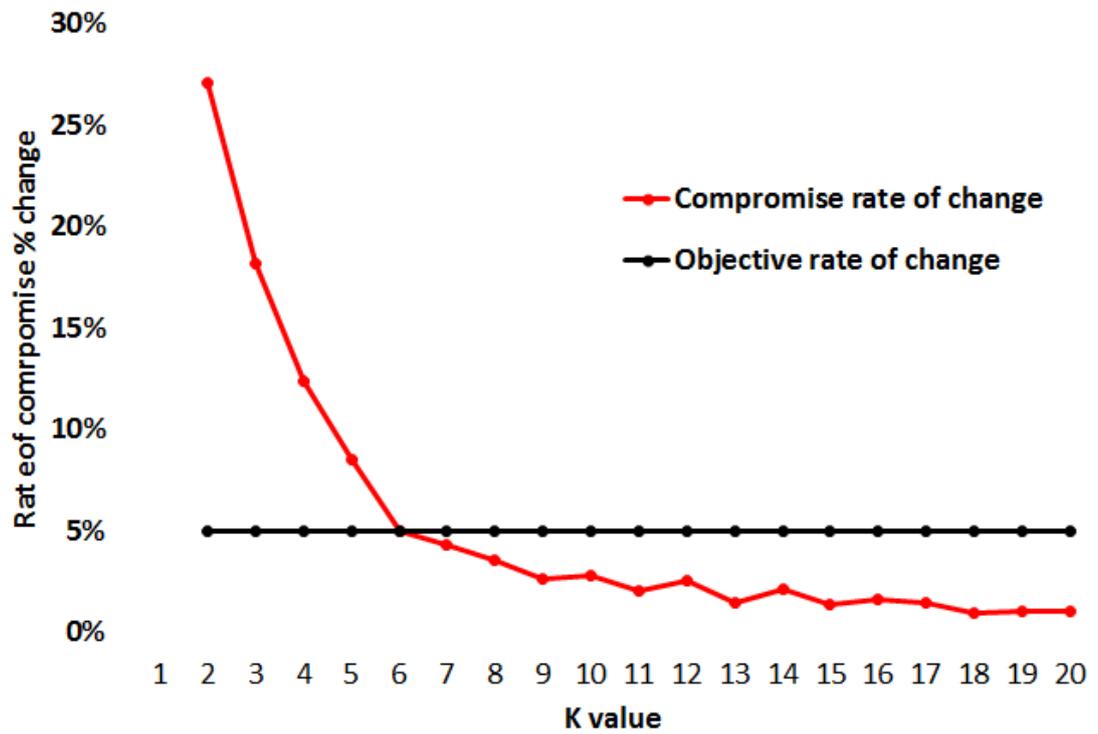


Figure 6.14: Rate of Average Redundancy N=3 D=20

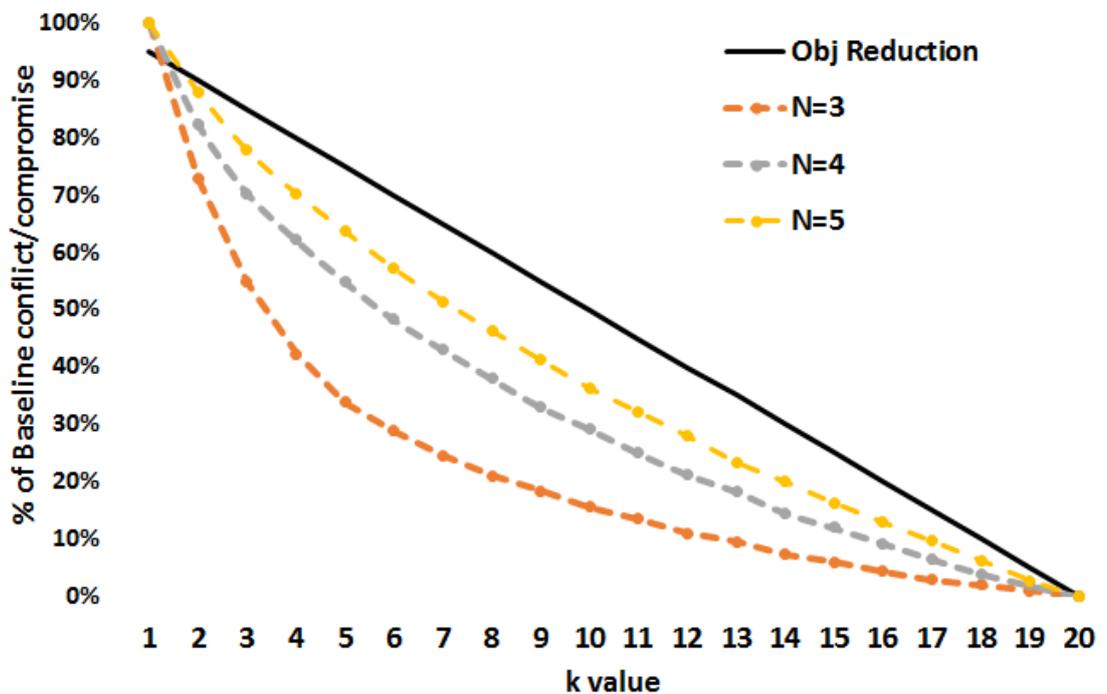


Figure 6.15: Average Redundancy over range of N values when D=20

Experimentation also analysed various sizes of DM groups. Figure 6.16 shows the average redundancy from 50 experiments when N=5 and D=50. Again here we see a high

level of redundancy found for the lower k-values and that this tails off for higher k-values. So for groups of 50 DMs we see there is redundancy to be found from grouping the DMs with low k-values and the approach is effective in discovering this. Therefore again we see that the approach is effective when seeking to reduce an objective set such that it will suffer fewer scaling issues.

From the experimentation of the approach's use of clustering we have analysed the trade-off between information loss and objective reduction. We have seen that for large groups of DMs there is a certain amount of redundancy within their views and therefore identifying it would be useful to reduce a problem's complexity. Additionally we have seen how such analysis can help an overseer to select an appropriate k-value thus helping both the traceability and the validity of a final aggregation. Such analysis can also aid sensitivity analysis of the selection of different k-values. The third stage of the approach then uses each cluster's single representation of judgments as a separate objective within MOO.

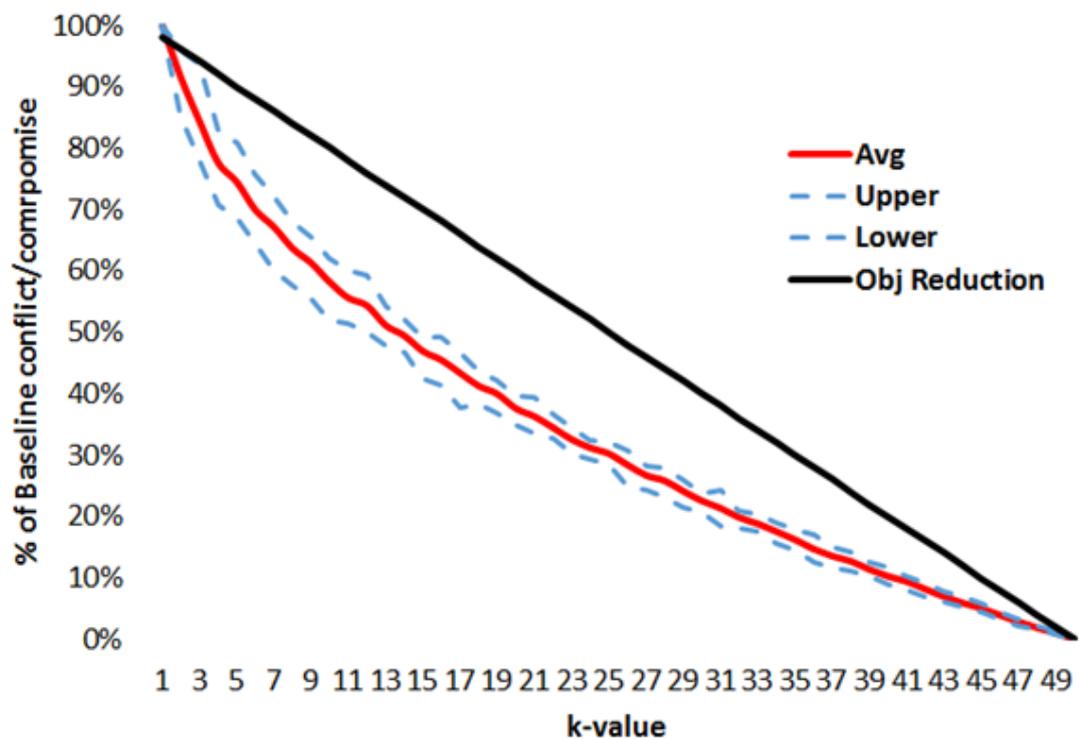


Figure 6.16: Average Redundancy N=50 D=50

### 6.3.5 Stage 3: Multi-Objective Optimisation

With the D DMs clustered into C clusters the third stage of the approach utilises the single representation of judgments for each cluster to perform MOO. The views of each cluster are modelled as a separate objective within the MOO process so the size of the objective

set will be equal to  $C$ . From the MOO the approach seeks to find aggregated judgments sets - *Aggregated Consensus Solutions* - through modelling the amount of compromise each cluster's judgments undergo as separate objectives. Given a problem with  $n$  elements and  $C$  clusters, and for each a complete  $n$  by  $n$  PCM representation of their member's judgments:

$$\{PCM_1, PCM_2, \dots, PCM_C\} \quad (6.3)$$

We represent each as a set of  $J$  judgments of each cluster's views and the approach models an  $O$  representation of each cluster's PCM  $\{O_1, O_2, \dots, O_C\}$ , each of which consists of  $J$  judgments  $\{o_1^k, o_2^k, \dots, o_j^k\}$ , for  $k=1, \dots, C$ . For each cluster with a measure of compromise as an objective we seek the set of non-dominated Aggregated Consensus Solutions. Again we can represent each solution as a judgment set of cardinality  $J$ , denoted as  $A = \{a_1, a_2, \dots, a_j\}$ . Then from any of the solutions found a preference vector can be derived.

In a similar way to the approach to aggregate smaller groups of DMs, constraints could additionally be added to the objectives. In this way the overseer could set thresholds of conflict for the amount of compromise a cluster's judgments could undergo in pursuit of aggregation. This could be useful for an overseer to define the maximum level of compromise a cluster's judgment set may undergo.

Additionally when an overseer is seeking aggregated consensus solutions with low inconsistency then additional inconsistency objectives can be added to the objective set. A range of inconsistency measures can be utilised to suit the overseer's preferences, these are Consistency Ratio (CR), number of 3-way cycles (L), Consistency measure (CM) and Geometric consistency Index measure (GCI), see Chapter 2 for a discussion of these measures.

With a front of solutions found from the MOO an overseer can then analyse the solutions found to ultimately select a single solution. For the set of solutions we could, as seen in Chapter 5, calculate for each solution a total measure of compromise and from this identify the global solution(s) from the set of solutions found. For example, given a group of 30 DMs who, by using the approach are clustered into 2 clusters, we could use total measures of compromise to identify a global solution, as shown in Figure 6.17, with the global solution shown as a hollow yellow triangle. In this figure the clusters are both the same size each having 15 members so here the total is derived evenly from each cluster's compromise. However, given that the clusters are derived based upon grouping

DMs with similar views together, it will invariably be the case that the clusters are of different sizes. Therefore the approach considers the size of each cluster within the analysis of the set of solutions found from the MOO.

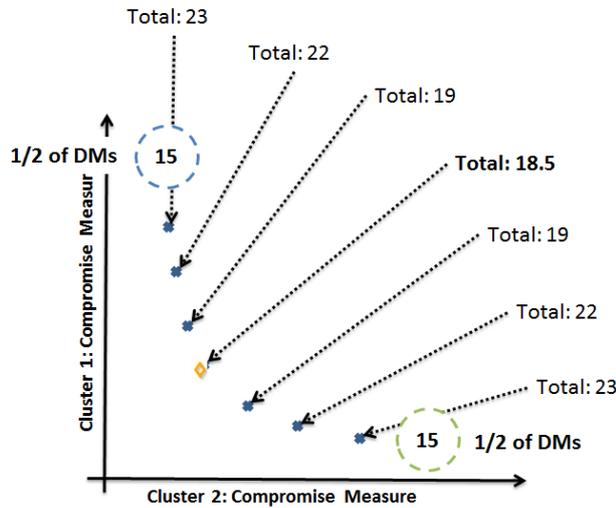


Figure 6.17: Illustration of MOO stage finding global compromise

The approach calculates for each found solution a global level of compromise for the solution based upon the solution's compromise value of each cluster modified by the size of the cluster. In this way we calculate a weighted global value of compromise for each solution where the weights are determined by the size of each cluster. In this way the overseer can identify the solution(s) that represents the weighted global solution from the set of solutions found taking into account the size of each cluster. For example, if our 30 DMs are clustered into 2 clusters of sizes 12 and 18 then Cluster2 contains 3/5 of the DMs and Cluster1 2/5. Then we can analyse the front of solutions found utilizing the sizes of the clusters as shown in Figure 6.18: Left. Here the weighted global solution is shown via a hollow red circle. In the example as Cluster2 is larger than Cluster1 we see that the weighted global solution identified favours Cluster2. If instead the 30 DMs were clustered into 2 clusters of sizes 6 and 24 then Cluster2 would be 4 times the size of Cluster1 and its weight would reflect this. In this case the weighted global solution identified by the approach would now be one which heavily favours Cluster1 due to its much larger size, as shown in Figure 6.18: Right. Identification of the weighted global solution is useful to aid an overseer select an aggregated solution. Additionally such analysis is useful for automatic selection of a solution from the approach during scenarios where analysis of the front of solutions is not possible. In such scenarios if multiple solutions share the

weighted global compromise value then a final aggregation is derived from the average of the preference vectors derived from the set of solutions that share this value.

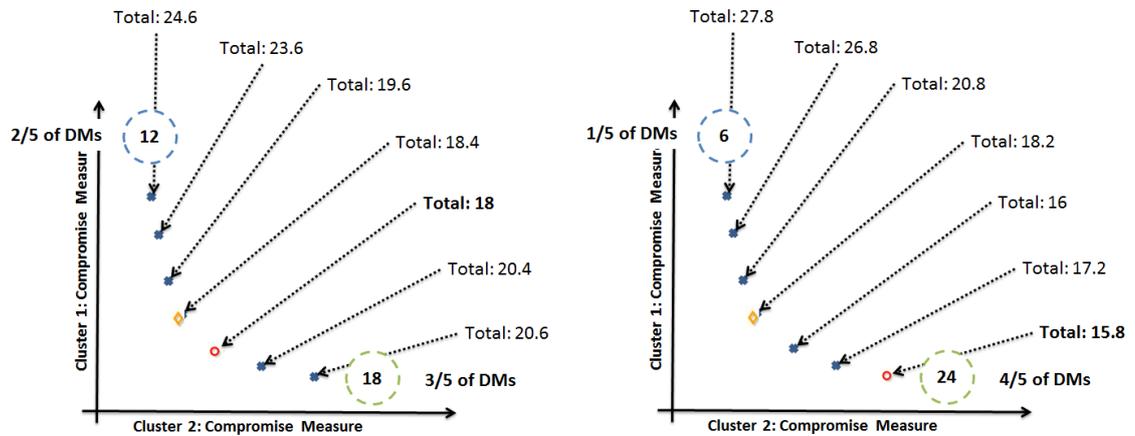


Figure 6.18: Illustration of MOO stage with cluster weights based on cluster size

Through consideration of the sizes of clusters in this way the approach has the capacity to deal with outliers. Given a group of DMs in which a single DM has highly conflicting views to the other members of the group it may be the case that their views are so contradictory that they are occupy their own cluster. For such a case during analysis of the front of solutions a cluster of such a small size would be given a suitably small weight so it would have little effect upon the identification of the weighted global solution(s), in this way softening the effects of the outlier. So the approach allows for identification of such outliers, in itself useful information, and then facilitates the softening of their effects upon analysis of the solutions. Identification of outliers in this way could additionally facilitate scenarios when an overseer deems an outlier too contradictory and so removes them from the group. As well as identifying the weighted global solution, the approach can facilitates additional analysis such as:

1. **Size:** the number of members in the cluster;
2. **Preference vector:** a ranking of the elements from the cluster's single representation - this way allowing comparisons between each cluster's views and that of the final aggregation;
3. **Information Loss for the cluster:** for the selected k-value the percentage amount of total compromise with respect to when k=1 is identified. This can further be broken down to identify how much is attributed to each cluster. Taking the 7 DMs

shown in Figure 6.9 for when  $k=3$  we can break down the 37% to reveal 18.5% is attributed to Cluster1, 11% to Cluster 2 and 7.5% to Cluster 3;

4. **Information Loss for cluster member:** the information loss per cluster value when divided by a cluster's size gives a measure of the average of the percentage of total redundancy per DM within the cluster. Taking the 7 DMs shown in Figure 6.9 when  $k=3$ , we can calculate that the average loss for cluster member to be 6% per DM for Cluster1, 5.5% in Cluster2 and 3.8% Cluster3. Such additional analysis can help reveal information regarding cohesion within the clusters.

### **6.3.6 Approach for aggregation of a large group of DMs' PCs Discussions**

We have outlined the approach to the aggregation of PC judgments of a large group of DMs. The approach tackles the scaling issues when utilizing MOO for group aggregation when the group is large by utilising clustering to group similar DMs together to look to reduce the size of the MOO objective set. The stages of the approach have been outlined along with analysis of the approach's facilities regarding aiding selection of an appropriate  $k$  value and post-MOO analysis. The next section gives examples that explore the approach and presents its benefits.

## **6.4 Examples**

In this section, step-by-step examples of the approach are presented:

1. Example 6.1 applies the approach to a relatively small number of DMs to illustrate the approach and highlight the analysis it facilitates;
2. Example 6.2 applies the approach to a large number of DMs to find a group aggregation.

For these examples the following parameter settings were employed. For the clustering stage the k-means++ algorithm is used to group the DMs. During clustering Euclidean distance is employed as the distance function and maximum iterations were set to 500. For the MOO, MOCell was employed with the following parameter settings: population size of 100 (10x10 grid); maximum evaluations count of 25,000; selection is performed via binary tournament with single point crossover (with crossover probability 0.9) and bit flip mutation (with probability 0.01) employed. Archive size is assumed to be defined by the overseer in each example. For deriving representational aggregated PCMs for each cluster each single objective GA is employed with a population size of 100 and maximum

evaluations count of 25000. Again selection is performed via binary tournament with single point crossover and bit flip mutation utilised.

The approach is independent of a specific prioritization method and any method may be utilised to derive a preference vector from any final or intermediate aggregation the approach calculates. In the example that follow the GM prioritization method is employed.

#### 6.4.1 Example 6.1: Small DM group

Example 6.1 takes a relatively small DM group of 6 to help illustrate the approach. For 4 elements the judgments from 6 DMs, along with initial preference vectors, are shown in Table 6.2.

Table 6.2: Example 6.1: 6 DMs' Judgments

DM1						DM2					
	1	2	3	4	<i>w</i>		1	2	3	4	<i>w</i>
1	1	1/5	1/2	1/7	0.06	1	1	3	6	6	0.59
2	5	1	3	8	0.59	2	1/3	1	5	1/2	0.18
3	2	1/3	1	1/2	0.14	3	1/6	1/5	1	1/2	0.07
4	7	1/8	2	1	0.21	4	1/6	2	2	1	0.17

DM3						DM4					
	1	2	3	4	<i>w</i>		1	2	3	4	<i>w</i>
1	1	5	2	7	0.57	1	1	1/2	1/3	1/3	0.11
2	1/5	1	3	2	0.21	2	2	1	2	2	0.38
3	1/2	1/3	1	1/2	0.11	3	3	1/2	1	1/2	0.21
4	1/7	1/2	2	1	0.12	4	3	1/2	2	1	0.30

DM5						DM6					
	1	2	3	4	<i>w</i>		1	2	3	4	<i>w</i>
1	1	2	1/2	7	0.38	1	1	3	1/2	7	0.41
2	1/2	1	7	1/2	0.27	2	1/3	1	8	1/6	0.19
3	2	1/7	1	1/2	0.14	3	2	1/8	1	1/2	0.14
4	1/7	2	2	1	0.20	4	1/7	6	2	1	0.26

In this example TotalSTJD is chosen by the overseer as the measure of compromise to create single representations of each cluster. To aid in selecting an appropriate k-value, the overseer analyses a range of k-values, and for each value analyses the amount of information loss as a percentage of the information loss of the total conflict in the group

when  $k=1$ . For this example results for  $k$  values 1 to 6 are shown in Figure 6.19. We observe a high rate of redundancy when  $k=2$ , with only 30% of the information loss in comparison to the overall conflict in the group. For subsequent values of  $k$  the amount of information loss reduction for each  $k$ -value is significantly less. From this we infer that within this group of DMs there appears to be 2 prevalent views and conjecture that the overseers will select a  $k$ -value of 2.

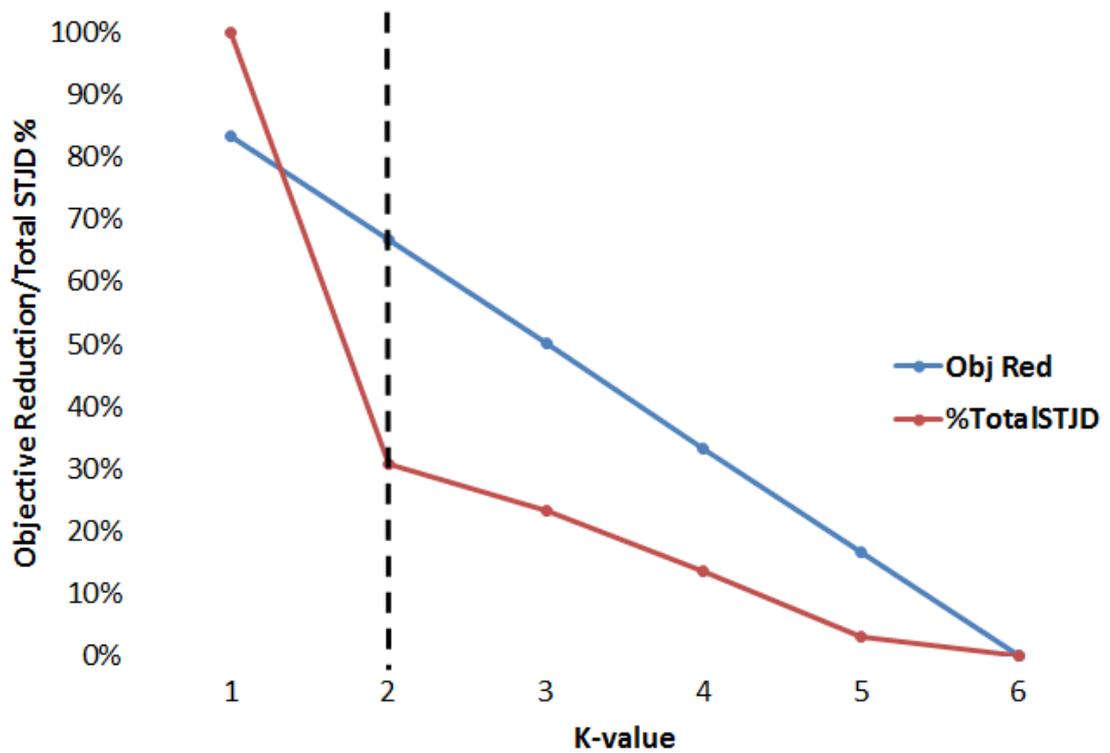


Figure 6.19: Example 6.1:  $k$ -value analysis

Given that a  $k$  value of 2 is chosen, from the clustering stage a cluster of DMs 1 and 4 and a second cluster of DMs 2, 3, 5 and 6 is derived. With TotalSTJD chosen as the single objective PCMs are derived for each cluster to represent their members, as shown in Table 6.3. Next we utilise the aggregated PCMs for each cluster to perform MOO. We have an objective set of 2 objectives that consists of the STJD compromise measure for each cluster (and an archive size of 50). The resulting objective space from the MOO is shown in Figure 6.20.

Table 6.3: Example 6.1: k=2 Aggregated PCMs

C1						C2					
{DM1, DM4}						{DM2, DM3, DM5, DM6}					
	1	2	3	4	w		1	2	3	4	w
1	1	1/4	1/3	1/5	0.07	1	1	3	2	7	0.52
2	4	1	3	5	0.54	2	1/3	1	6	1/2	0.32
3	3	1/3	1	1/2	0.16	3	1/2	1/6	1	1/2	0.09
4	5	1/5	2	1	0.23	4	1/7	2	2	1	0.18

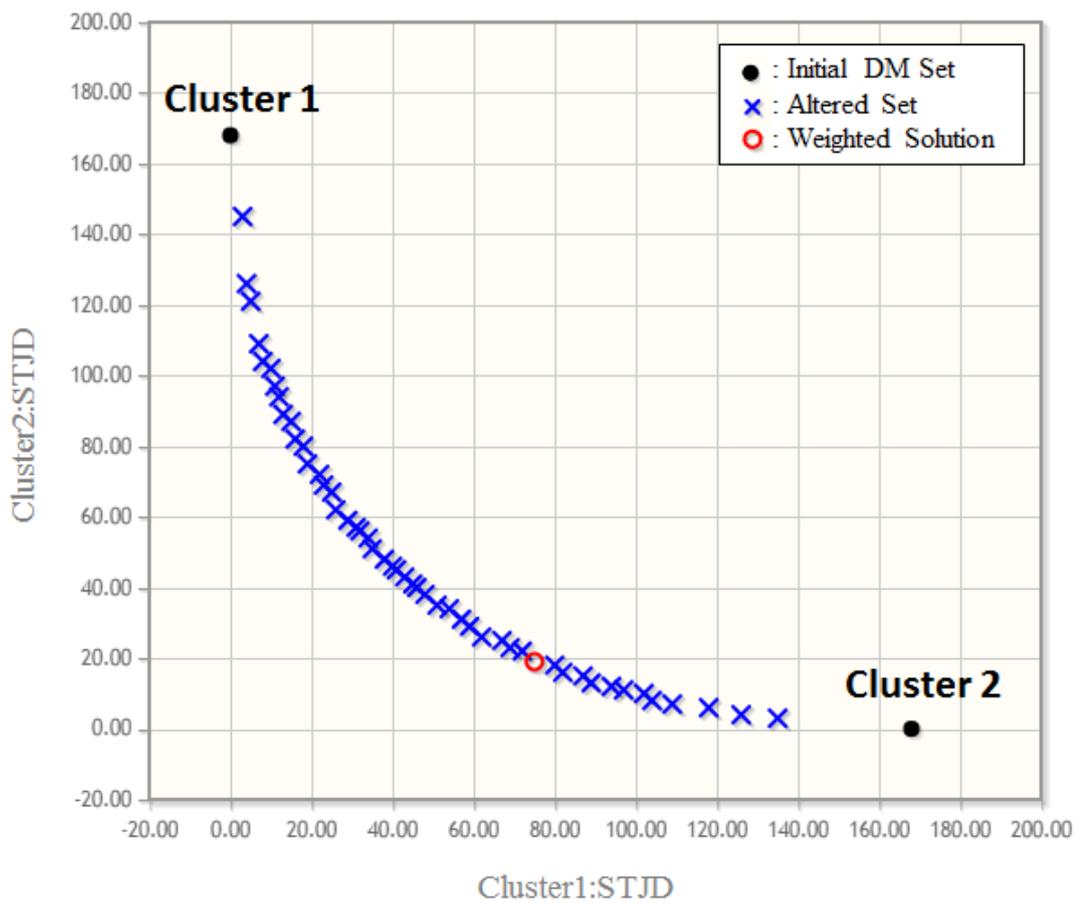


Figure 6.20: Example 6.1: k=2 objective space

To aid analysis of the front of solutions we then utilise the size of each cluster to determine weights for each to be used to identify a weighted global solution. Here with Cluster2 being twice the size of Cluster1 it is assigned a weight to indicate it is twice as important. In Figure 6.20 we observe the identified weighted solution which as we would expect favours Cluster2 due to its larger size and subsequent larger weighting. The overseer is

then free to analyse and select any solution found from which an aggregated preference vector can be derived. They may choose the identified weighted global solution from which a preference vector can then be derived representing an aggregation of the whole group of DMs, as shown in Table 6.4.

This example shows that we are able to reach an overall group aggregation from a group of DMs' PCs. Through clustering the number of objectives was reduced, in this case from six to two objectives, helping an overseer to more easily analyse the objective space resulting from the MOO. We have seen how, along with a final aggregation, the overseer was able to extract various knowledge about the group such as the prevalence of two distinct views within the group. The next example applies the approach to a much larger group of DMs.

Table 6.4: Example 6.1: Preference vector derived from weighted global solution

	1	2	3	4
w	0.3170	0.3985	0.1260	0.1585

#### 6.4.2 Example 6.2: Large number of DMs

Example 6.2 applies the approach to a much larger group of DMs. The example looks at a council deciding the location of a new recycling plant within a town by canvassing local opinion. There are five possible locations for the new recycling plant, see Figure 6.21. By canvassing the PCs of 100 local DMs regarding their preference between locations an overseer utilises the approach to aggregate and analyse their views.

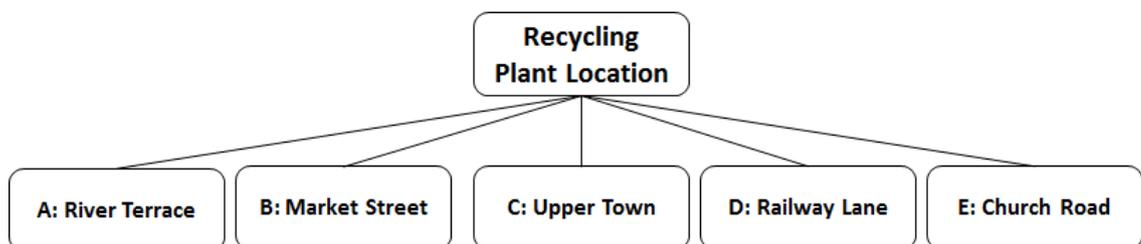


Figure 6.21: Example 6.2 recycling plant alternative locations.

First the approach clusters the group of DMs based on their views. Analysis over a range of k-values will help the overseer select an appropriate k-value. With TotalNJR utilised

as the measure of compromise the analysis of the compromise for the range of k-values from 1 to 10 is shown in Figure 6.22; as the overseer is only interested in seeking reduction of the objective set to single figures 10 is the upper limit of this analysis. We see that there are relatively high levels of redundancy found over the first couple of k values and then that the rate of change of redundancy found quickly tails off after k=3. Therefore it may be that the overseer chooses 3 as the k-value. This would represent nearly 40% reduction of information loss from when k=1 and will result in an objective reduction of 94%.

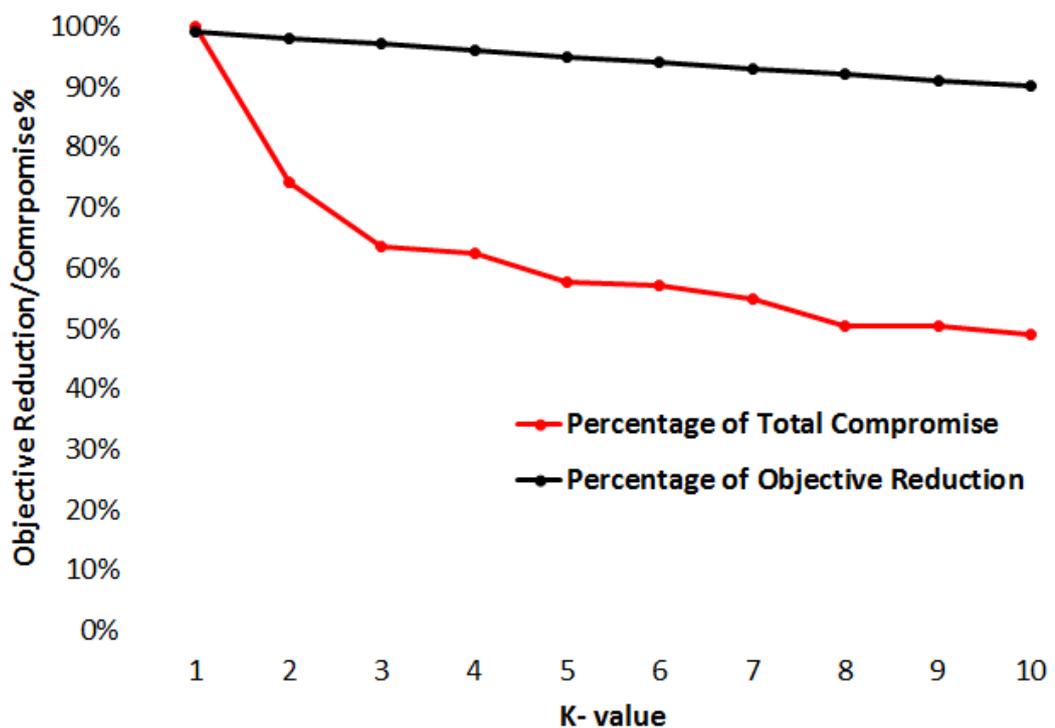


Figure 6.22: Example 6.2: Analysis over range of k values

With a k-value of 3 the clustering results in clusters of sizes 40, 24 and 36. Next a single representation PCM of each cluster's views are derived (with TotalNJR used as the single objective in each cluster), shown in Table 6.5 along with their preference vectors. From these cluster representations and resulting preference vectors it appears there are 3 distinct views with Cluster 1 heavily favouring Church Road, Cluster 2 favouring River Terrace slightly more than Railway Lane and Cluster 3 favouring Market Street slightly over River Terrace.

Table 6.5: Example 6.2: k=3 aggregated PCMs

**Cluster 1**

Size: 40

	RT	MS	UT	RL	CR	$w$
RT	1	1/2	1/6	1/9	1/8	0.03
MS	2	1	1/4	1	1/9	0.07
UT	6	4	1	1/3	1/6	0.13
RL	9	1	3	1	1/8	0.15
CR	8	9	6	8	1	0.62

**Cluster 2**

Size: 24

	RT	MS	UT	RL	CR	$w$
RT	1	7	3	5	1	0.38
MS	1/7	1	7	1/5	1/4	0.08
UT	1/3	1/7	1	1/9	1/3	0.04
RL	1/5	5	9	1	6	0.33
CR	1	4	3	1/6	1	0.17

**Cluster 3**

Size: 36

	RT	MS	UT	RL	CR	$w$
RT	1	4	5	1/8	4	0.28
MS	1/4	1	4	4	8	0.35
UT	1/5	1/4	1	3	1	0.12
RL	8	1/4	1/3	1	1	0.16
CR	1/4	1/8	1	1	1	0.09

Next we perform MOO with each cluster's single representation PCM modelled as a separate objective, with STJD employed as the compromise objective for each cluster, and an archive of 50 defined. The 3-dimensional objective space from the MOO from the viewpoint of Cluster1 and Cluster2 is shown in Figure 6.23. From the plot we observe how Clusters 1 and 3 are more similar than Cluster 2 which is the most distinct cluster. From the set of non-dominated solutions found the overseer is free to analyse and

compare all solutions for selection from which to derive a preference vector. Additionally, by using the size of each cluster to derive proportion weights, the weighted global solution has been identified and is plotted in Figure 6.23. If the overseer chooses the identified weighted global solution then the preference vector derived representing group aggregation is shown in Table 6.6. From this final group aggregation we see that Church Road is the most preferred alternative.

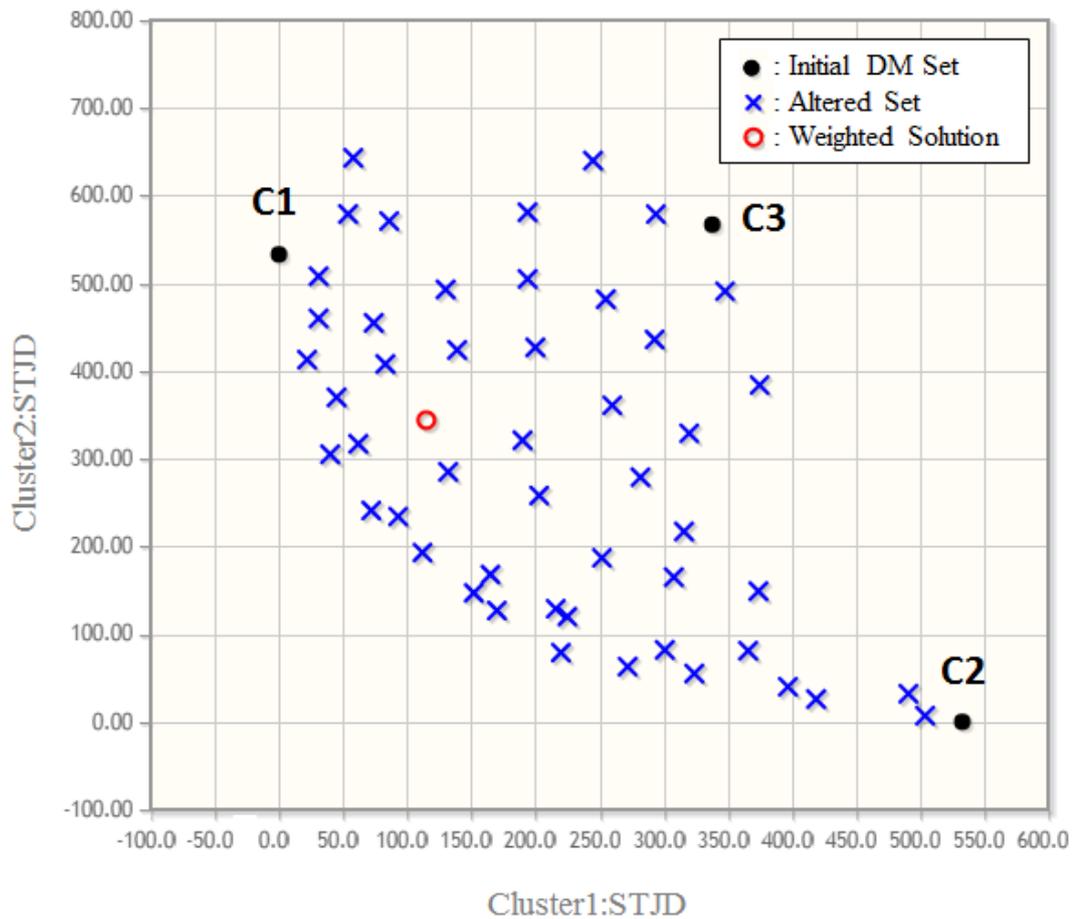


Figure 6.23: Example 6.2: k=3 objective space

Table 6.6: Example 6.2: k=3 weighted global solution preference vector

	RT	MS	UT	RL	CR
w	0.1413	0.1949	0.1949	0.184	0.2849

Additionally the overseer may compare the preference vectors from the different clusters and analyse the compromise within each cluster, as shown in Table 6.7. When k=3 the amount of redundancy as a percentage of total conflict when k=1 is 63%. In Table 6.7 the

breakdown of this 63% for each cluster is shown. We observe for example, that of this 63%, 28% is attributed to Cluster 1, 12% to Cluster 2 and 22% to Cluster 3, suggesting that Cluster 2 contains the most agreeable subset of DMs. This is similarly shown within the calculations of the average compromise per DM for each cluster, which take into account the size of each cluster to determine the average amount of information loss per DM in each cluster.

Table 6.7: Example 6.2: cluster measures analysis

	Size	Compromise Reduction %	Average Compromise % per DM
Cluster1	40	28%	0.7%
Cluster2	24	12%	0.5%
Cluster3	36	22%	0.61%
<b>When k=3</b>		<b>63%</b>	<b>0.63%</b>

Additionally the traceability of the approach facilitates further sensitivity analysis to be performed. Given an overseer's analysis of Figure 6.22 they may also wish to analyse when a k value of 2 is chosen and therefore performs sensitivity analysis to see how a different k values affects the result. With a k value of 2 the clustering now results in cluster sizes of 60 and 40. Single representations are then derived of each cluster's views (again with TotalNJR utilised as the objectives) as shown in Table 6.8. Here an overseer can observe that again a cluster of DMs who heavily favour Church Street has been derived and that the second cluster appears to have more mixed views on the alternatives (3 alternatives all similarly preferred).

Table 6.8: Example 6.2: k=2 aggregated PCMs

Cluster 1 Size: 40							Cluster 2 Size:60						
	RT	MS	UT	RL	CR	w		RT	MS	UT	RL	CR	w
RT	1	1/5	1/7	1/7	1/2	0.05	RT	1	9	7	1/8	1	0.28
MS	5	1	1/5	1	1/5	0.11	MS	1/9	1	5	9	1	0.25
UT	7	5	1	1/9	1/3	0.16	UT	1/7	1/5	1	6	1/8	0.09
RL	7	1	9	1	1/4	0.27	RL	8	1/9	1/6	1	3	0.16
CR	2	5	3	4	1	0.41	CR	1	1	1/8	1/3	1	0.22

This time the MOO will be performed with an objective set of size 2 (again with STJD utilised as the measure of compromise for each objective). The objective space found is shown in Figure 6.24. Again, through the use of the size of each cluster to derive appropriate weights, the weighted global solution has been identified and is plotted in Figure 6.24.

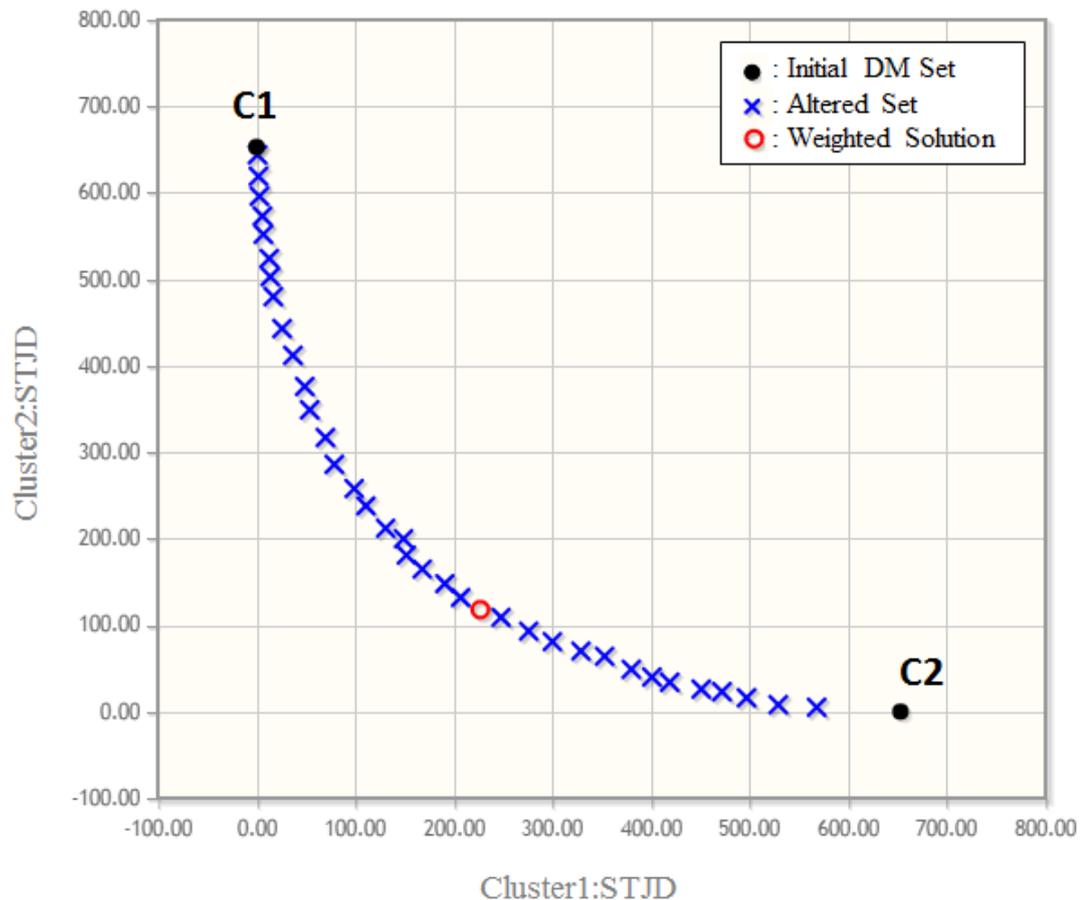


Figure 6.24: Example 6.2: k=2 objective space

If the overseer chooses this identified weighted global solution then the derived preference vector from this solution is shown in Table 6.9. Here an overseer sees that again in the final group aggregation Church Road is the most preferred alternative, this along with the intermediate analysis the approach facilitates helps the overseer in validating the final aggregation. Finally the overseer may conjecture that the final preference vector from when k=3 is chosen as here clusters or 3 distinct views were derived whereas when k=2 the second cluster contained mixed views.

Table 6.9: Example 6.2: k=2 weighted global solution preference vector

	RT	MS	UT	RL	CR
w	0.1235	0.1867	0.1305	0.2268	0.3324

## 6.5 Conclusions

This Chapter has presented an approach to the aggregation of the PCs of a large group of DMs. The approach aims to tackle the scaling issues that occur when MOGAs are utilised within MOO for the aggregation of a large group of DMs. The approach tackles the scaling issues by first clustering a large group of DMs into sub-groups based on the similarity of their views. Next from these sub-groups, a single judgment set representation of each group's views can be calculated which are then used as a single objective within a MOO approach. As the MOO is performed with respect to each cluster of DMs modelled as a single objective the approach facilitates reduction of the number of objectives compared to when each DM is modelled as a separate objective. The set of trade-off solutions between the clusters is then sought via MOO. From all solutions found a preference vector can be derived that represents an aggregation of the whole group. The approach further facilitates sensitively analysis of the stages, for example, selection of a different number of clusters or selection of a different solution from the front of solutions identified through MOO. From analysis of the clustering stage and of the objective space an overseer can discover information about the trade-offs between objective reduction and information loss, as well as indications of the nature of the group of DMs and their conflict. Additionally the approach incorporates weights of importance of each cluster based upon the number of DMs it contains and can identify the weighted global solution(s) from the set of found solutions. In this way the approach is resilient to outliers as smaller groups of outlier DMs will be assigned a small weight and hence will have less impact upon the calculation of a weighted global solution.

# **Chapter 7 MOODS: A Web Based Decision Support Tool**

## **7.1 Introduction**

In this chapter a brief overview of the MOODS (Multi-Objective Optimisation Decision Support) software tool is presented. MOODS is a web-based decision support tool that implements the theoretical approaches and methods presented in this thesis.

The tool implements the measures of compromise and the approaches defined for inconsistency reduction, group aggregation and large group aggregation. MOODS is an interactive web-based tool that runs in all major browsers utilizing native HTML code with no plugins or downloads required. In keeping with the rationale of the approaches to enhance decision making through traceability and interactivity the interface is designed to be both flexible and responsive. Realisation of the work via a free web-based tool should facilitate accessibility without the need to download and install software.

Most decision support tools are desktop-based requiring download and installation before use, such as Diviz [145], PriEsT [146], IND-NIMBUS [147], Right Choice, super-decisions, and HIPRE [148] (a web-based version is available [149] however it runs via a java applet and not in a browser). Various, more sophisticated web-based decision tools exist which are license-based such as Questfox, MakeItRational, Criterium-DecisionPlus, Smart Picker Pro and Expert choice.

The overall architecture design of MOODS is discussed next. This is followed by an overview of the interface implementation. Example uses of the tool for the scenarios of reducing inconsistency for a single DM, group aggregation and then larger group aggregation, are then presented followed by conclusions.

## **7.2 Architecture Design**

In this section a brief overview of the architecture of MOODS is presented. The elements of the architecture are shown in Figure 7.1. The design separates the presentation layer and the business logic layer. This separation should enhance the tool's ease of evolution and the reusability of the design. For example, the business logic layer could be easily utilised within a different type of interface such as a mobile device. All information

exchange between the browser and the server is performed via dynamic AJAX<sup>13</sup> calls with no page post backs required, helping facilitate a more responsive and interactive interface.

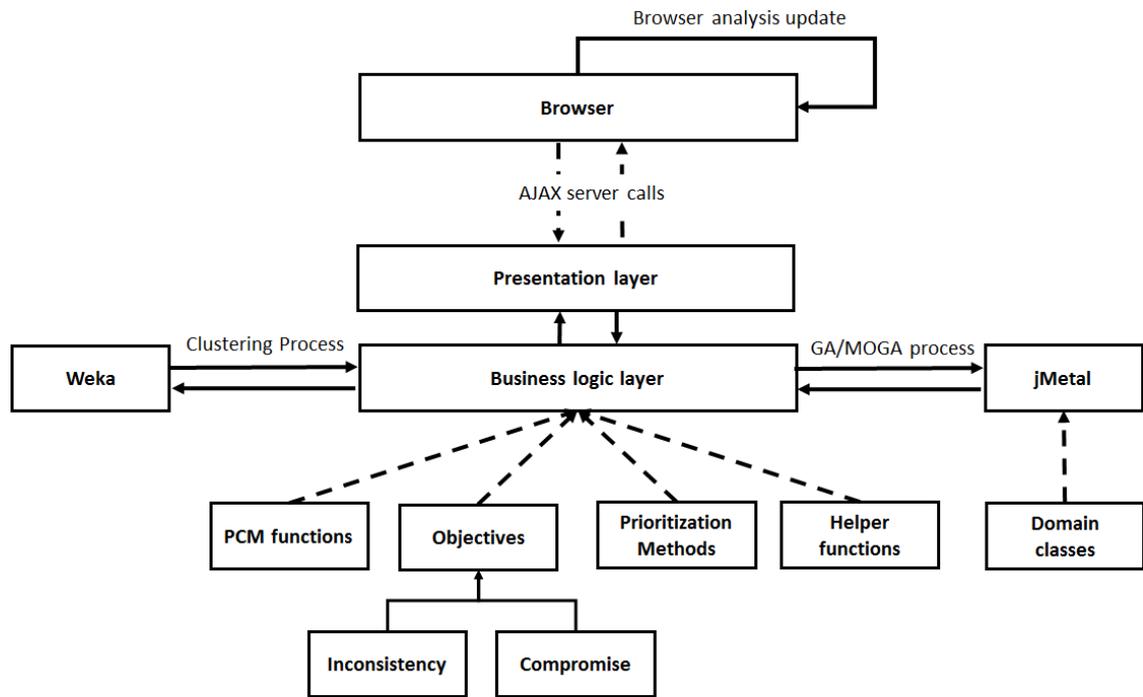


Figure 7.1: Overview of MOODS architecture

The various elements of the architecture diagram are outlined below:

1. **Browser Interface:** The interface is realised with HTML5, CSS and JavaScript.
2. **AJAX server calls:** During operation all data exchange between the browser and server is performed dynamically via AJAX calls. The tool's functionality is implemented without requiring any post-backs, helping to create a responsive interface.
3. **Browser analysis update:** The tool implements various functionality within the browser such that calls to the server are not required, thus enhancing the responsiveness of the system. For example, when solutions are plotted in the objective

<sup>13</sup> AJAX is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications.

space the plot can be updated to be viewed from different pairs of objectives without requiring a call to the server.

4. **Presentation Layer:** The presentation layer deals with the aspects of the browser view that require a server call and deals with streamlining the unpacking and packing of data between the browser and the business logic layer.
5. **Business Logic layer:** The business logic layer deals with the main functionality of the tool as well as interfacing with utilised library packages. Realised in Java this layer is split into modules of functionality, as follows:
  - a. **PCM Function:** Functions to represent, analyse and process DM judgments.
  - b. **Objectives:** Each objective usable within the tool, both those of inconsistency measures and those of measures of compromise, are implemented to a consistent design. This allows for ease of processing of objective sets as well as allowing ease of extensibility of the approach due to the use of reflection<sup>14</sup> for objective evaluation. A new inconsistency measure could easily be implemented in the framework as a new objective. Similarly a new measure of compromise could easily be implemented in the framework simply by defining a new objective.
  - c. **Prioritization methods:** All prominent methods have been implemented and can be utilised in the tool based on user preferences. Through the consistent implementation design additional methods can be swiftly implemented.
  - d. **Helper functions:** Various utility functions of more general programming functionality are contained within the helper functions module.
  - e. **Domain classes (for jMetal):** The tool interfaces with jMetal (see below) and for our approaches domain classes have been created from jMetal interfaces to define domain problems, describing how our problems are defined and encoded.

---

<sup>14</sup> Reflection allows methods calls to be evaluated at runtime [157]. Therefore a new objective could be implemented and reflected by its name without the program being aware of its existence at compile time.

6. **Machine Learning (ML) Packages:** To aid in performing the various machine learning algorithms employed within our approaches the architecture interfaces with two ML packages.
  - a. **jMetal:** Metaheuristic Algorithms in Java (jMetal) is an object-oriented Java-based framework for multi-objective optimization with metaheuristics [150].
  - b. **Weka:** A collection of machine learning algorithms for data mining tasks [151].

### 7.3 Interface Implementation

The tool's interface is realised with HTML5, CSS and JavaScript. No applets or flash plugins are required and the interface runs in most browsers. The layout of the various sections of the interface is shown in Figure 7.2. Next an outline and discussion of each section is presented.

#### 7.3.1 Problem Setup

A problem is setup via the Problem setup panel. A screenshot of the Problem Setup section is shown in Figure 7.3. First the number of DMs and the number of elements of the problem can be chosen. When utilised to reduce inconsistency a single DM is selected. Change to either of these parameters triggers the creation of an appropriate number of elicitation PCM panels within the interface (see Section 7.3.2).

Within the MOGA Setup panel the algorithm to use can be selected; MOCell is the default algorithm, however other algorithms can be utilised. Additionally the significant parameters of the archive size, defining the number of solutions that will be presented to the user and the number of evaluations to perform are selected. Additional MOGA parameters such as those relating to crossover and mutation can be defined within a folding tab that is revealed through hovering over the 'More...' link.

Within the Post MOO Analysis panel the user can setup how the found solutions for group aggregation will be analysed. The colour scheme used to denote certain types of solutions such as red for weighted solutions are used consistently within the interface's tables and charts. The user can check which analysis (if any) will be performed: Global compromise analysis, Weighted global analysis and/or Fairest compromise analysis. The analysis can be dynamically updated either to update the analysis when the DMs' weights of importance are updated or DM constraints are updated facilitating swift sensitivity analysis.

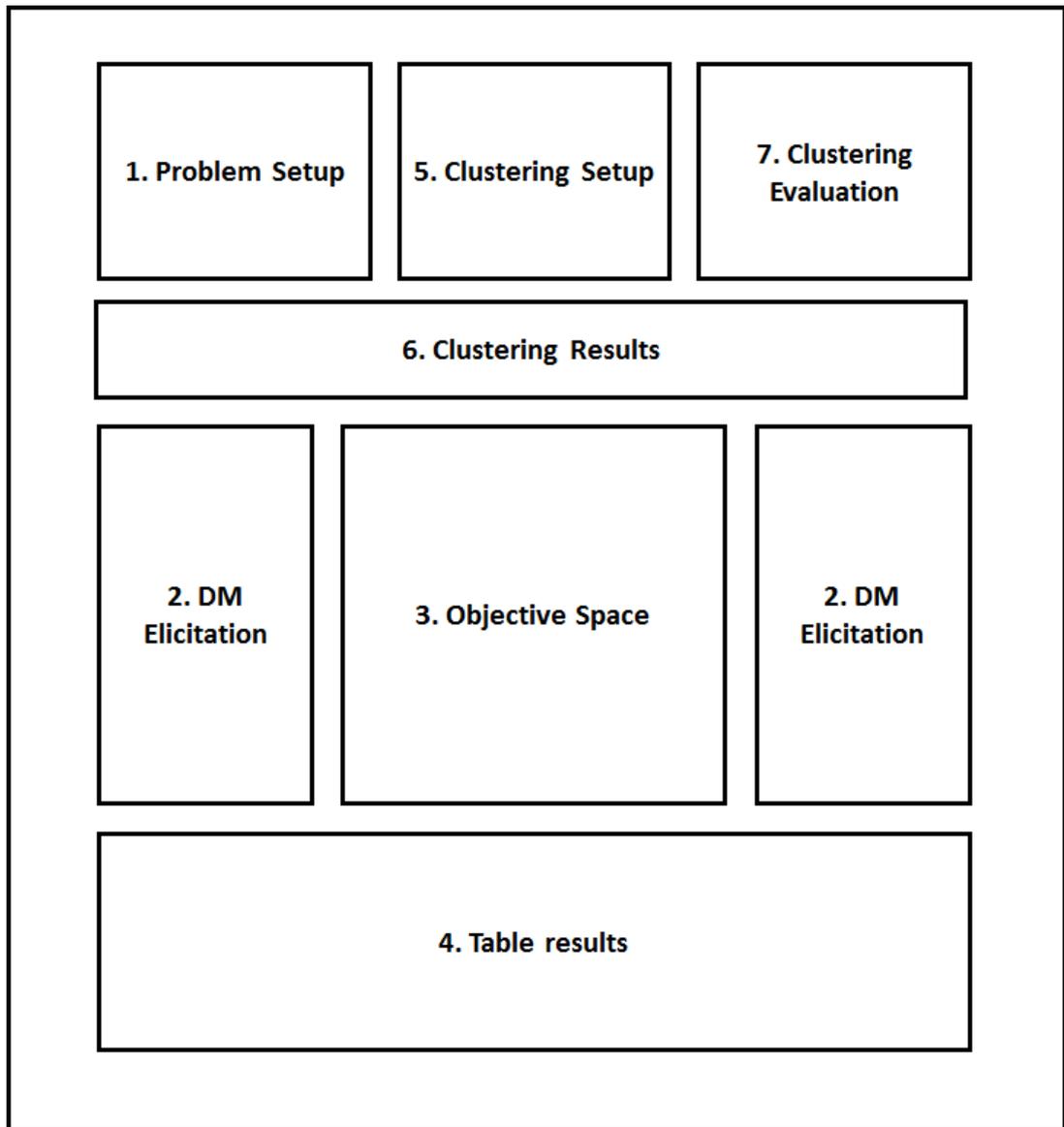


Figure 7.2: Overview of Interface

Performing the initialisation command will calculate the inconsistency values for each DM's judgments as well as calculate a preference vector and ranking of their judgments. The user can select the prioritization method to be utilised.

Finally the tool allows for problems to be saved and loaded. When a problem is saved, the parameters relating to the problem along with the data for each DM are saved within a text file (upon the user's local machine). A saved problem file can then be utilised to re-load a previous problem, once loaded the settings are populated along with all the DM elicitation panels.

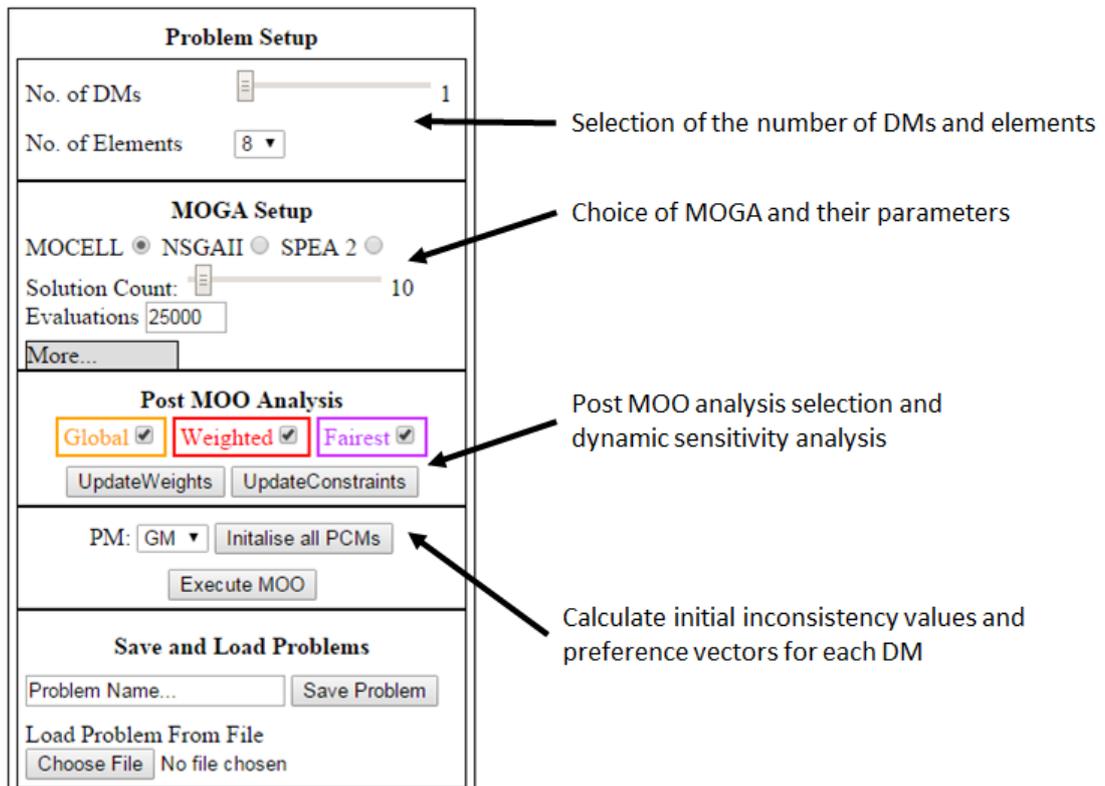


Figure 7.3: Problem Setup panel

### 7.3.2 DM Elicitation

With the number of DMs and number of elements chosen the interface automatically generates the desired number of DM panels to record the judgments of each DM. DM panels are presented one after another on either side of the objective space panel. A screenshot of a DM elicitation panel is shown in Figure 7.4. Each panel's title denotes the DM number along with the weight of importance of the DM. Implemented as a HTML 5 slider bar each DM weight of importance can easily be altered to aid swift sensitivity analysis.

The measure(s) of compromise that will be used as an objective to seek solutions with minimum alteration to the DM judgments can then be selected. Additionally constraints can be set upon any measure of compromise measure chosen, where upper and/or lower values for the compromise measure can be set.

The PCM for the DM is represented as a table of dropdown boxes from which a value from the PC scale employed can be selected. In such a representation the judgments can be elicited in any order, and changed any number of times. The tool implements the 1-9 scale, however this could be extended to implement additional scales. Additional scales would simply require the judgment dropdown boxes to be populated with the new range of possible values. In the PCM representation the trace of the matrix boxes are set to 1

and un-editable, and when any judgment is defined its reciprocal judgment is automatically set helping to ensure correct entry.

After initialisation is performed from the problem setup panel each DM's judgments are utilised to calculate the initial inconsistency measures and preference vector for each DM. The prioritization method used to calculate the ranking is selected from the problem setup panel.

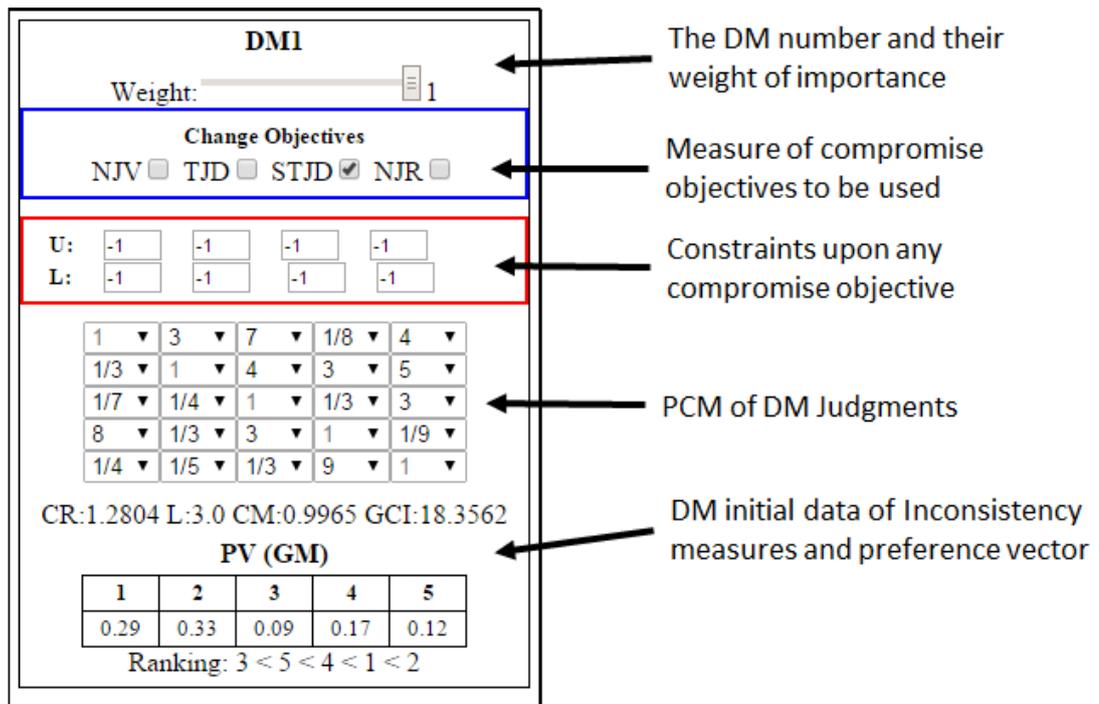


Figure 7.4: DM Elicitation panel

### 7.3.3 Objective Space

Visualisation of solution sets found via MOO as well as additional MOO setup options are contained within the objective space panel. A screenshot of an objective space showing a set of solutions found during a reduction of inconsistency for a single DM scenario is shown in Figure 7.5.

When inconsistency measures are to be utilised as objectives within the MOO process they can be selected in the objective space panel along with any constraints upon their upper and lower values. The CR, L CM and GCI measures can all be utilised as inconsistency reduction objectives.

A plot of the objective space and options regarding what is shown in the objective space and from what perspective make up the rest of the objective space panel. The solution set found via MOO is passed to the interface from the server and the solutions

are plotted in the objective space with respect to two of the objectives. The user can dynamically update the plot to view the solutions with respect to any pair of the objectives. By selecting different objectives as the x and y axis the set of solutions can be viewed with respect to any pair. Any change in an axis and subsequent updating of the plot is performed within the browser allowing for swift analysis of the objective space to be performed.

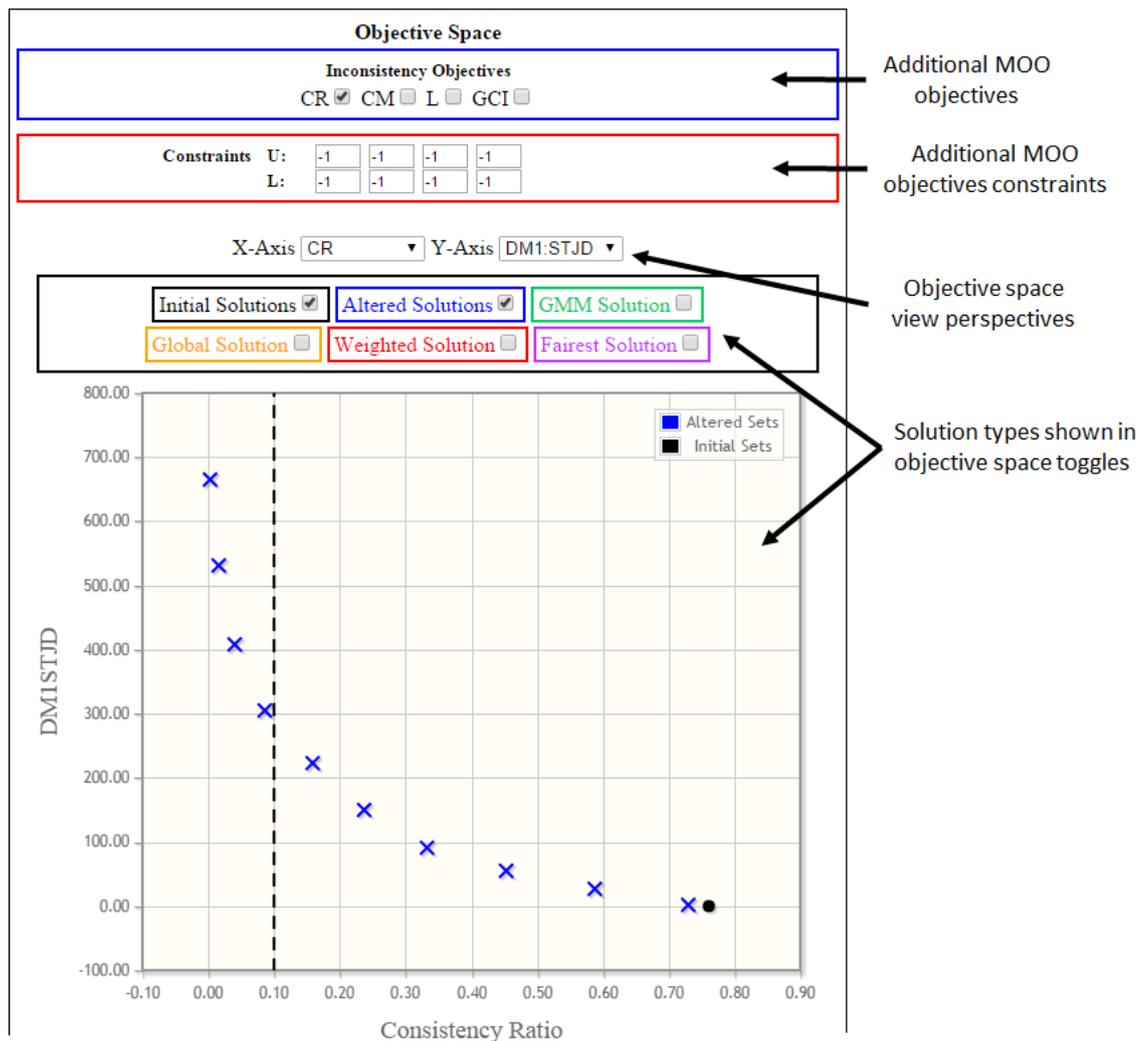


Figure 7.5: Objective Space panel

Post MOO analysis, regarding processes such as global compromise calculation, is done to identify various solutions within the solution set, such as, for example, the global solution(s). The display within the objective space of the different types of solutions that are identified, such as global solution(s) can then be toggled. The different types of solutions are altered solutions, global solutions, weighted global solutions and fairest solutions. Additionally initial solutions which represents each DM's (or cluster's) initial

judgments can be plotted and toggled within the objectives space. Moreover for comparison analysis the display of a solution found during group aggregation from the GMM can be toggled.

Additionally for objectives that have recommended threshold values, such as the CR or the GCI, the threshold values are highlighted in the objective space plot via a dotted black line. Furthermore any constraints set upon an objective are shown within the objective space (when viewing the objective space with respect to that objective) via a dotted red line.

### 7.3.4 Table Results

As well as visualisation and exploratory interactive analysis of the solutions within the objective space the solutions are displayed within the table results panel. A screenshot of table results calculated during a group aggregation of 2 DMs scenario is shown in Figure 7.6.

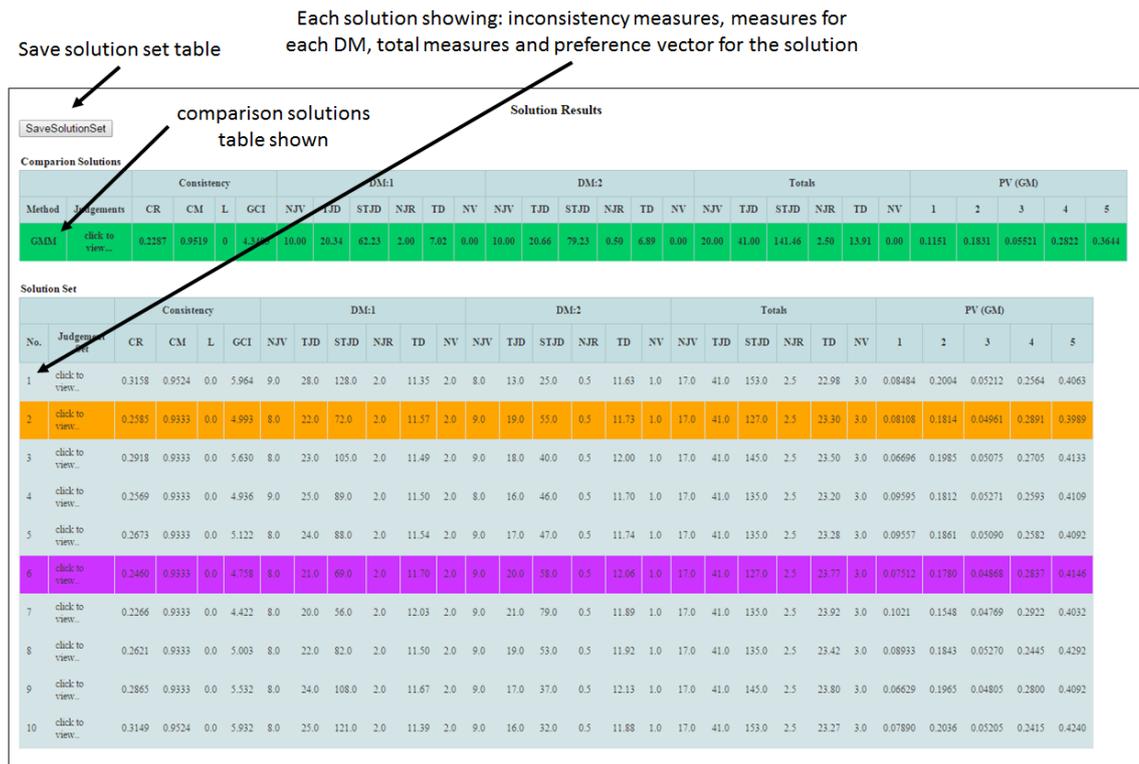


Figure 7.6: Table Results panel

Each row displays the data for a single solution: the solutions inconsistency measures, the values of each measure of compromise for each DM, the total values of each measure of compromise and the preference vector derived from the solution. The preference vector is derived utilizing the prioritization method selected within the setup panel. Clicking on

any row displays the solution's judgments as a PCM in a fold-down box below the solution row, as shown in Figure 7.7; here from the same solution set the 3<sup>rd</sup> solution's judgments are viewed. This allows the judgments of multiple solutions to be easily viewed and compared by the user. The rows utilise the same solution type colour scheme as the objective space allowing ease of identification. Additionally for comparison analysis, data relating to the solution found via the GMM during group aggregation is also displayed within the panel. Finally the solution set table can be saved (locally to the user's machine) for storage or further analysis.

**Solution Set**

No.	Judgement Set	Consistency																														
		CR	CM	L	GCI	NJV	TJD																									
1	click to view..	0.3158	0.9524	0.0	5.964	9.0	28.0																									
2	click to view..	0.2585	0.9333	0.0	4.993	8.0	22.0																									
3	click to view..	0.2918	0.9333	0.0	5.630	8.0	23.0																									
<table border="1"> <tbody> <tr> <td>1</td> <td>1/7</td> <td>2</td> <td>1/4</td> <td>1/4</td> </tr> <tr> <td>7</td> <td>1</td> <td>7</td> <td>1/6</td> <td>1/2</td> </tr> <tr> <td>1/2</td> <td>1/7</td> <td>1</td> <td>1/4</td> <td>1/4</td> </tr> <tr> <td>4</td> <td>6</td> <td>4</td> <td>1</td> <td>1/5</td> </tr> <tr> <td>4</td> <td>2</td> <td>4</td> <td>5</td> <td>1</td> </tr> </tbody> </table>		1	1/7	2	1/4	1/4	7	1	7	1/6	1/2	1/2	1/7	1	1/4	1/4	4	6	4	1	1/5	4	2	4	5	1						
1	1/7	2	1/4	1/4																												
7	1	7	1/6	1/2																												
1/2	1/7	1	1/4	1/4																												
4	6	4	1	1/5																												
4	2	4	5	1																												
4	click to view..	0.2569	0.9333	0.0	4.936	9.0	25.0																									
5	click to view..	0.2673	0.9333	0.0	5.122	8.0	24.0																									
6	click to view..	0.2460	0.9333	0.0	4.758	8.0	21.0																									
7	click to view..	0.2266	0.9333	0.0	4.422	8.0	20.0																									
8	click to view..	0.2631	0.9333	0.0	5.002	8.0	22.0																									

Figure 7.7: Table Results panel with expanded Judgments panel

### 7.3.5 Clustering Setup

For scenarios involving aggregation of a large group of DMs the tool facilitates the initial clustering stage of the approach outlined in Chapter 6 within the clustering setup panel. A screenshot of the clustering setup panel is shown in Figure 7.8. Before executing the clustering the user selects the number of clusters as well as the clustering algorithm to be utilised. The k-means++ approach is the default algorithm although k-means can be selected for comparison purposes. Future work could make available a number of additional clustering algorithms to compare.

From the clustering a single representation of the views of the members of each cluster are then calculated. The total compromise measure to be used within the optimisation process to derive these single representation is then selected by the user.

When the clustering is executed the clustering setup information along with all the DM judgments are passed to the server. The clustering is then performed followed by calculations to derive single representations for each cluster. The clustering results panel is then populated with the resulting data.

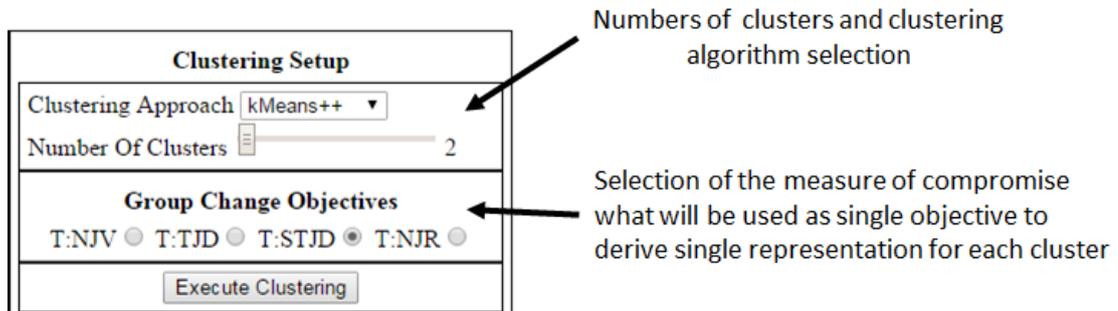


Figure 7.8 Clustering Setup panel

### 7.3.6 Clustering Results

The clustering results panel shows the overall clustering evaluation along with the data for each cluster and its single PCM judgment representation. A screenshot of clustering results for when  $k=2$  is shown in Figure 7.9.

Firstly a table of overall clustering statistics is presented representing the overall compromise values involved in the clustering. (A single compromise measure is selected to seek single representations for each cluster, this table displays the data for all compromise measure totals from the single representations).

Then the data for each cluster is presented within a separate panel. For each cluster its size and corresponding weight of importance based upon its size is displayed. The weights are also editable giving a user additional control to adjust and override weights as desired. Next the compromise values of each cluster's single representation is shown along with the list of the DMs in the cluster.

With each cluster to be used within the MOO stage the measures of compromise of objectives for the cluster are selectable along with any constraints on the objectives. The single representation PCM of the cluster is then displayed along with initial data relating to the inconsistency measures of the cluster's PCM and the preference vector derived from the cluster's PCM. The prioritization method used to calculate the ranking is selected on the problem setup panel.

Below the panels of each cluster any additional inconsistency objectives to be utilised within the MOO stage can be selected along with any constraints. When cluster

aggregation is performed the judgments of each cluster are utilised along with the selected compromise objectives (and any inconsistency objectives) and then the set of solutions found are passed back to the browser and displayed within the objective space and the table results area. Analysis to aid the selection of an appropriate k-value can be performed within the clustering evaluation panel.

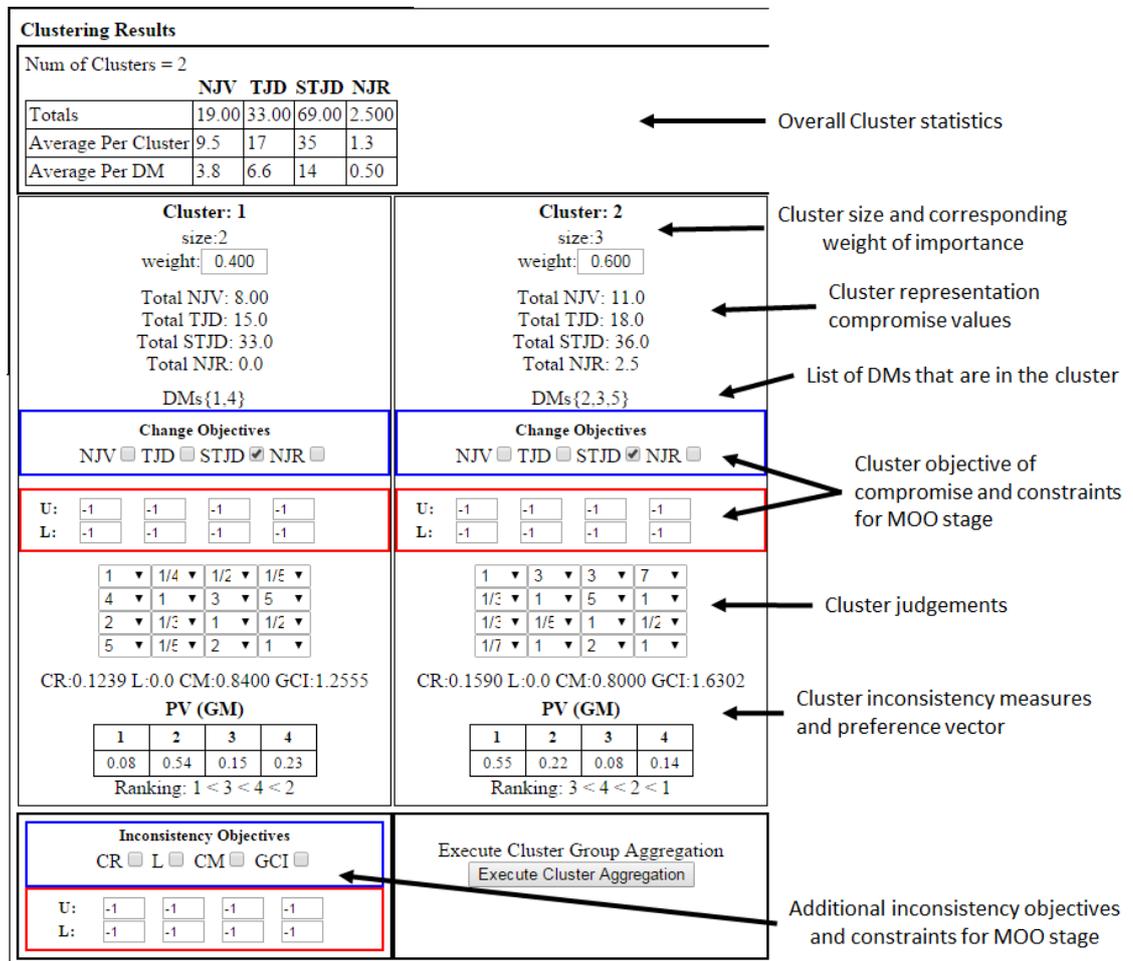


Figure 7.9: Clustering Results panel

### 7.3.7 Clustering Evaluation

To aid in the selection of an appropriate value for the number of clusters, analysis can be performed over a range of cluster values to reveal knowledge about the group and their views. In the clustering evaluation panel the user can define the upper k value of the clustering analysis. Once execution of the analysis has begun the DM data along with the clustering parameters from the clustering setup panel are passed back to the server and the clustering iteratively performed for k from 1 to the upper value chosen. After each k-value clustering is performed the results for that k value are passed back to the browser

and a new row added to the table, this way progressively displaying the results to the user. For each k value the total measure of compromise that occurs during creation of single representations for the clusters is shown along with the percentage of compromise with respect to when k=1. The percentage of objective reduction achieved for the k value is also shown. Additionally the table of analysis data can be saved (locally to the user's machine) for storage or further analysis. A screenshot of clustering analysis for group of 5 DMs is shown in Figure 7.10.

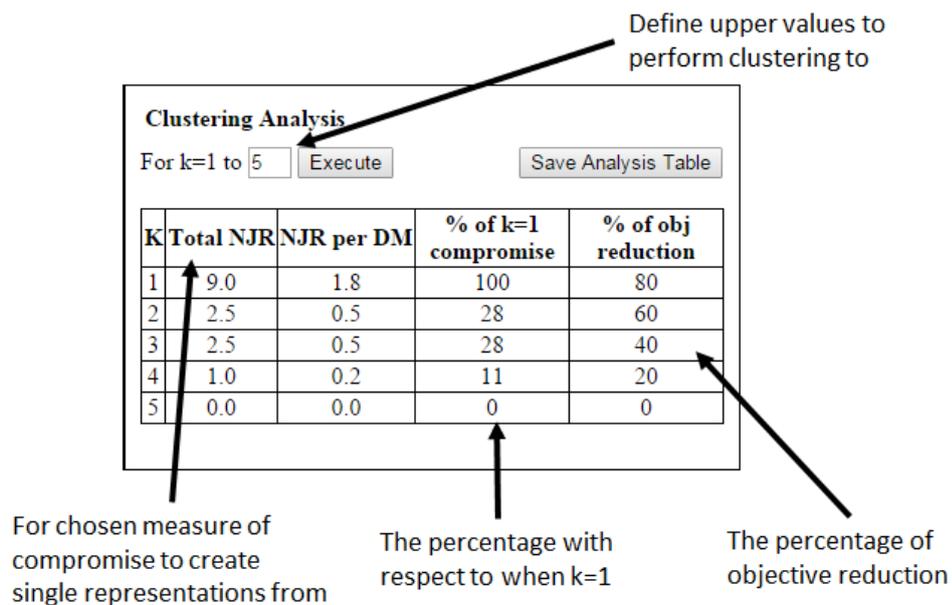


Figure 7.10: Clustering Evaluation panel

## 7.4 Usage Scenarios

Presented next are examples of the use of MOODS for the 3 scenarios of objective reduction for a single DM, group aggregation and large group aggregation.

### 7.4.1 Reducing inconsistency for single DM

MOODS can be utilised for the scenario of reducing inconsistency for a single DM. An example of the use of MOODS within such a scenario is shown in Figure 7.11. Here as the user is interested in reduction of inconsistency for a single DM the number of DMs is set to one and then a single DM elicitation panel is created. All options relating to post-MOO analysis for group aggregation are unchecked. The judgments are then entered into the elicitation panel, and the compromise measure of TJD is selected. Finally before the MOO is executed additional inconsistency objectives or CR and L are selected within the objective space. Execution of MOO for this 3-objective set is then carried out and the solution space is populated with the solutions from the perspective of the axis chosen,

here initially CR and TJD. The user can see the nature of the objective space from this view. The user can then additionally analyse the solutions returned further through altering the axis views selected to view the solutions from the perspective of CR and L. This way the nature of inconsistency within their judgments is revealed to the user. A constraint additionally can be added to the L objective of 0 – as the user is only interested in solutions without any 3-way cycles. This new objective space viewed from the perspective of CR and TJD can be analysed by the user. From this view of the constrained objective space the user could then select the first solution found CR below 0.1, highlighted with a dotted circle at the bottom of Figure 7.11. This would represent a solution without any cycles and with CR below 0.1.

#### **7.4.2 Group Aggregation**

MOODS can be utilised within a group aggregation scenario. An example of the use of MOODS for 3 DMs seeking to reach a consensus utilizing constraints is shown in Figure 7.12. With the judgments from each DM entered and with TJD as the measure of compromise to model each DM an initial set of solution can be found via MOO. From this the nature of the conflict between the 3 DMs is revealed along with both the global solution and the fairest solution. With this knowledge of the initial objective space the DMs can then add realistic and feasible constraints to drill down into the objective space to seek to reach a consensus. With the added constraints the objective space then contains only 5 solutions. Analysis of the objective space and the table of solutions of a small subset such as this is much easier than the initial larger set of solutions.

#### **7.4.3 Large Group Aggregation**

Moods can be utilised for aggregation of a large group of DMs. An example of the use of MOODS to aggregate and analyse 100 DMs' views is shown in Figure 7.13. First the data of the 100 DMs can be read in from a file, which then can generate and populate the 100 elicitation panels. To aid the overseer in the selection of an appropriate k-value they can perform analysis of the clustering for k values 1 to 5. With k=2 chosen by the overseer the clustering then results in a single representation of each cluster being derived, along with analysis of the clusters, their members and the compromise involved in reaching the single representations. With STJD chosen as the measure of compromise objective for each cluster MOO can be performed to find the set of trade-off solutions as well as analysis of the weights of each cluster to identify the weighted global solution. The inconsistency measures of the PCM of each cluster shows high CR values so the overseer can then add an additional 3<sup>rd</sup> objective of CR with an upper limit constraint of 0.1. The

3-dimensional objective space then only contains solutions in which the CR is below 0.1 and the newly weighted global solution can be chosen and its CR will be less than 0.1.

## **7.5 Conclusions**

This chapter has outlined the development and implementation of the MOODS web-based decision support tool. The tool implements the novel approaches to inconsistency reduction, group aggregation and large group aggregation presented in this thesis. The rationale for the design choices and an overview of the system architecture have been discussed. The interface design has been presented and the interface explained. Examples have been presented of the tool's use in a number of scenarios: reducing inconsistency for a single DM, performing group aggregation and performing large group aggregation. There are many enhancements and extensions that could be implemented in future versions of the tool such as integrating a database for more persistent data storage and developing a mobile phone interface extension. Such areas of development as well as future investigations of the approaches presented in this thesis are discussed in the next chapter along with overall conclusions.

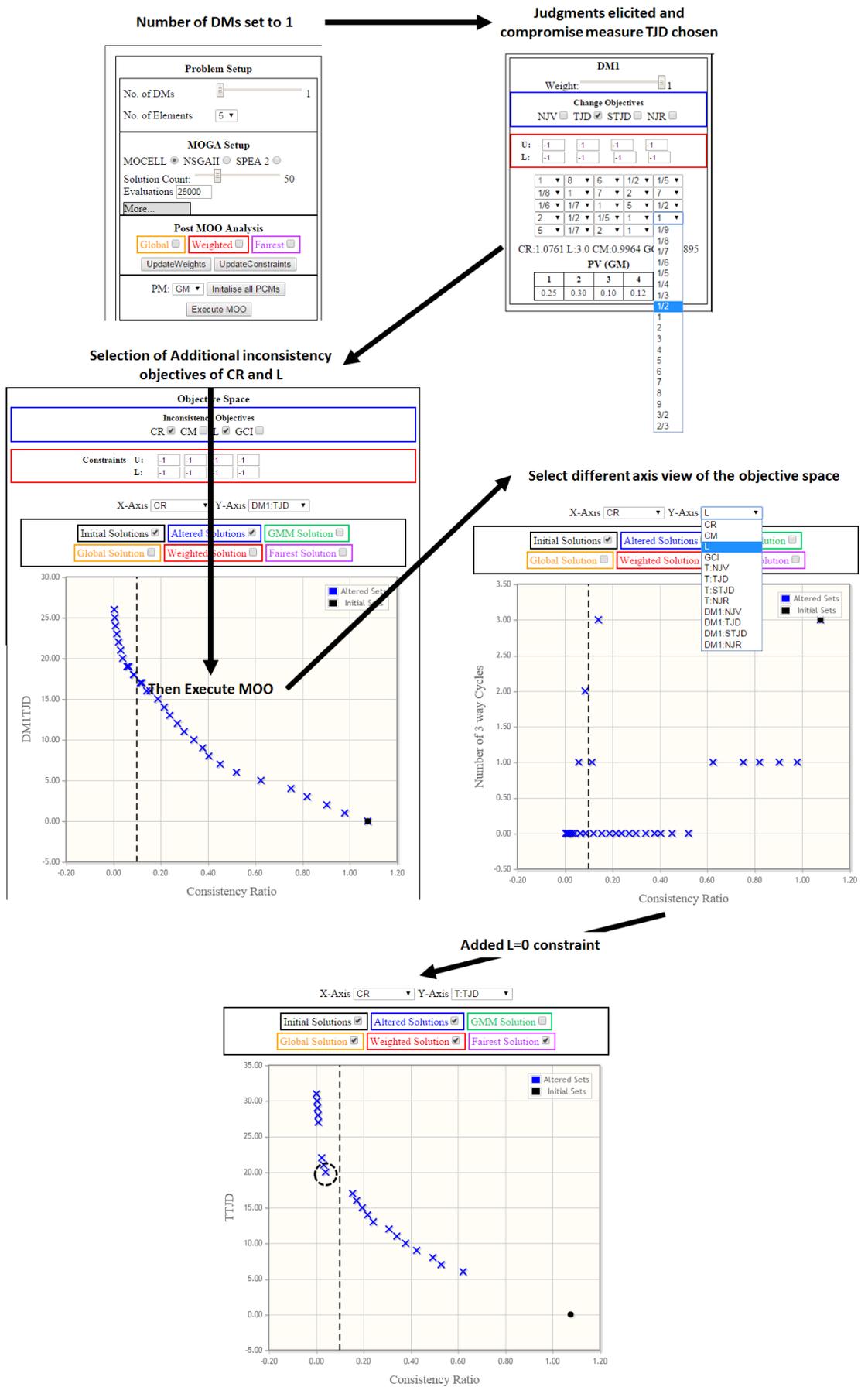
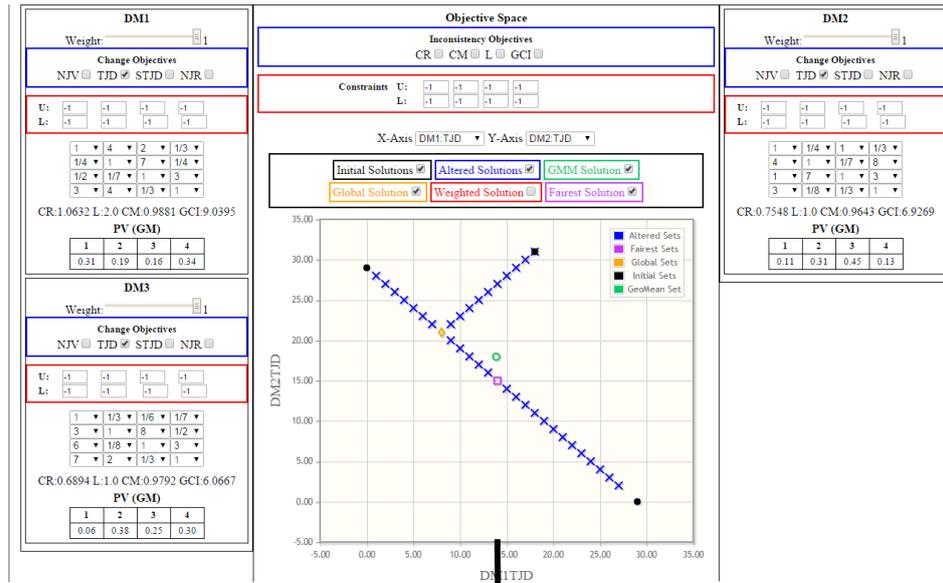
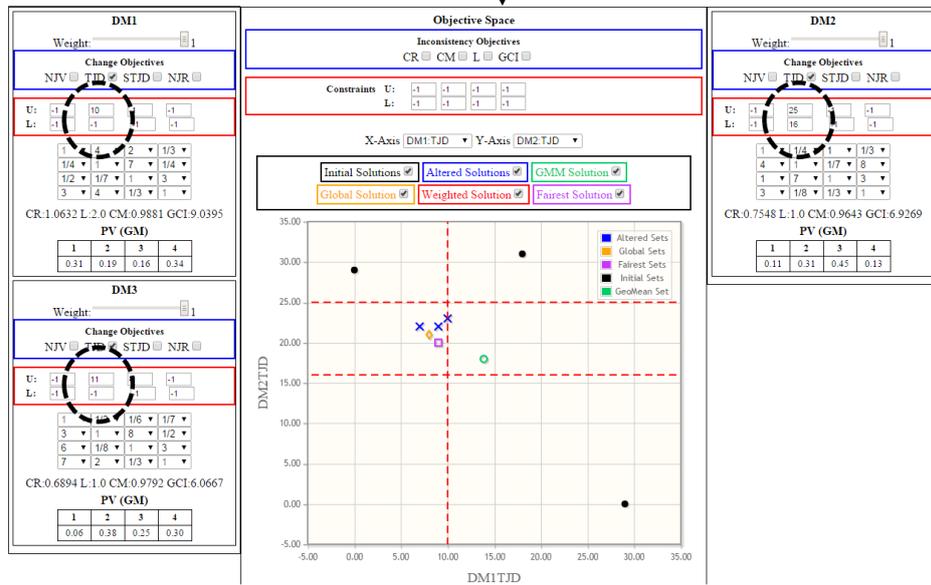


Figure 7.11: MOODS use during single DM inconsistency reduction



DM Constraints added



Constrained objective space solutions

Solution Set		Consistency				DM-1				DM-2				DM-3				Totals				PV (GM)											
No.	Judgement Set	CR	CM	L	GCI	NJV	TJD	STJD	NJR	TD	NV	NJV	TJD	STJD	NJR	TD	NV	NJV	TJD	STJD	NJR	TD	NV	TJD	STJD	NJR	TD	NV	1	2	3	4	
1	<a href="#">click to view...</a>	0.5812	0.9762	2.0	5.113	3.0	7.0	21.0	1.5	8.081	4.0	3.0	22.0	212.0	2.0	10.02	3.0	4.0	11.0	43.0	0.5	9.727	2.0	10.0	40.0	276.0	4.0	27.83	9.0	0.1499	0.3815	0.1898	0.2789
2	<a href="#">click to view...</a>	0.5726	0.9763	2.0	5.018	3.0	9.0	41.0	1.5	8.358	4.0	2.0	20.0	308.0	2.0	9.852	3.0	4.0	11.0	43.0	0.5	9.232	2.0	9.0	40.0	292.0	4.0	27.44	9.0	0.1502	0.4327	0.1810	0.2660
3	<a href="#">click to view...</a>	0.6157	0.9792	2.0	5.314	4.0	9.0	31.0	1.5	8.188	4.0	3.0	22.0	234.0	2.0	9.928	3.0	2.0	9.0	41.0	0.5	9.383	2.0	9.0	40.0	306.0	4.0	27.50	9.0	0.1309	0.4220	0.1774	0.2696
4	<a href="#">click to view...</a>	0.5708	0.9763	2.0	4.999	3.0	8.0	30.0	1.5	8.213	4.0	3.0	21.0	309.0	2.0	9.910	3.0	3.0	10.0	42.0	0.5	9.431	2.0	9.0	39.0	281.0	4.0	27.35	9.0	0.1319	0.4114	0.1848	0.2718
5	<a href="#">click to view...</a>	0.6222	0.9762	2.0	5.514	4.0	10.0	34.0	1.5	8.369	4.0	4.0	23.0	213.0	2.0	10.02	3.0	3.0	8.0	30.0	0.5	9.061	2.0	11.0	41.0	277.0	4.0	27.45	9.0	0.1137	0.4028	0.1811	0.3024

Figure 7.12: MOODS use during group aggregation

**Problem Setup**

No. of DMs:

100 No. of Elements:

**MOGA Setup**

MOCELL  NSGAI1  SPEA 2

Solution Count:

Evaluations (25000):

More...

**Post MOO Analysis**

Global  Weighted  Fairst

UpdateWeights UpdateConstraints

PM: GM  Initialise all PCMs

Execute MOO

**Save and Load Problems**

Problem Name:  Save Problem

Load Problem From File

Choose File | TestD100NST... | Final.txt

**Clustering Setup**

Clustering Input Type! Cluster By Judgements

Clustering Approach: kMeans++

Number Of Clusters:

Distance modifier!  
None (Euclidian Distance)

**Group Change Objectives**

T.NJV  T.TJD  T.STJD  T.NJR

Execute Clustering

**Clustering Analysis**

For k=1 to  Execute Save Analysis Table

K	Total NJR	NJR per DM	% of k=1 compromise	% of obj reduction
1	428.5	4.285	100	99
2	317.5	3.175	74	98
3	272.5	2.725	64	97
4	267.0	2.67	62	96
5	247.5	2.475	58	95

- Load data of group of DMs views
- With NJR as compromise measure perform range of clustering analysis to determine appropriate k value
- Performing clustering with k-value of 2 chosen

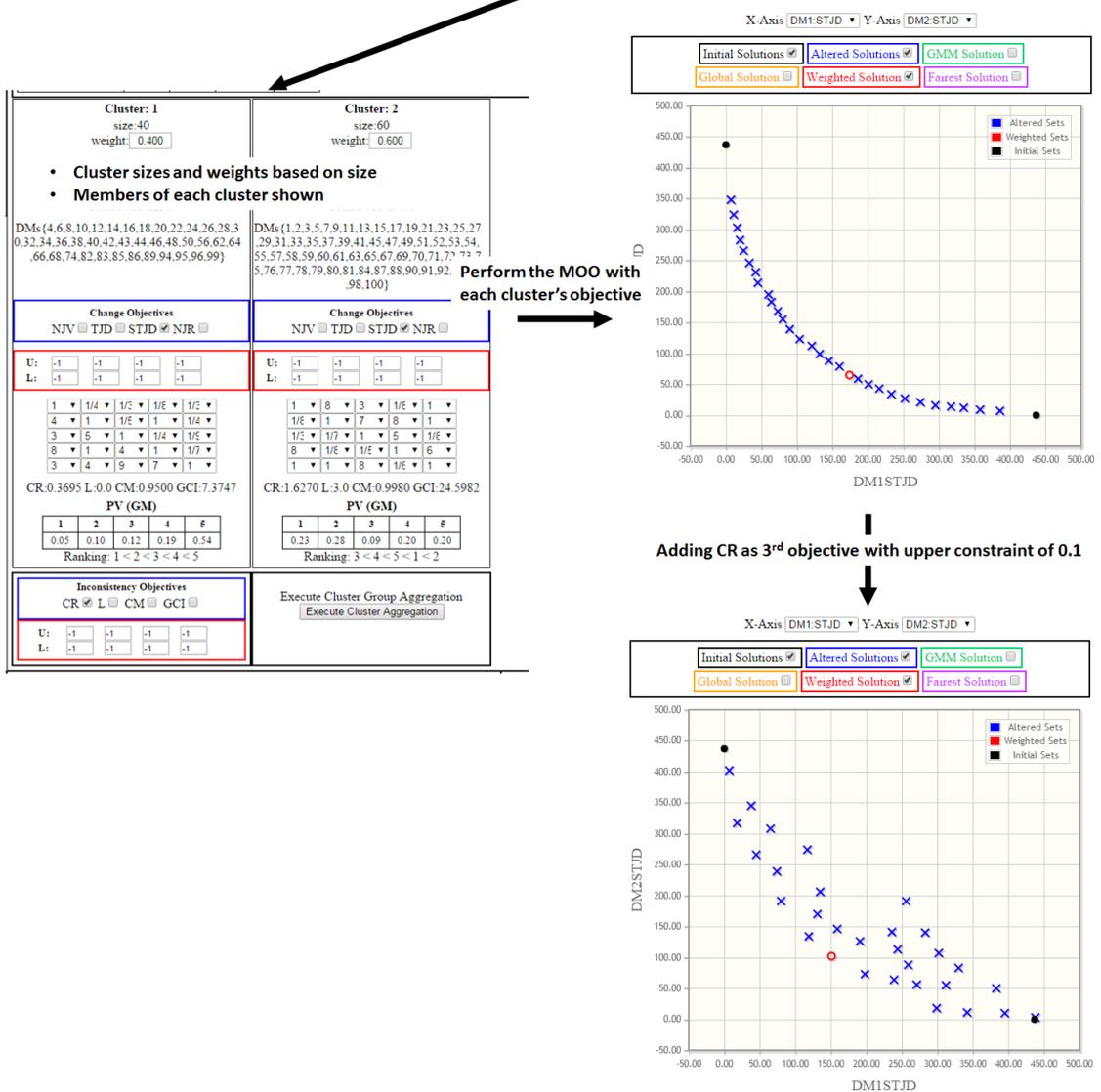


Figure 7.13: MOODS use during large group aggregation

## Chapter 8 Conclusions and Future Work

This chapter first summarises the contributions of the research before exploring future areas of investigation.

### 8.1 Summary

This research has proposed approaches that seek to enhance traceability, transparency, and auditability to help facilitate a richer decision making process. A DM's judgments undergo alteration during scenarios such as inconsistency reduction, or when looking to reach a group aggregation with other DMs. Traceability within such scenarios reveals knowledge of the scenario helping to create a richer and more comprehensible process, and aid validation of the scenario outcomes.

A range of measures of compromise have been defined to measure the amount of alteration a DM's judgments undergo in such scenarios. These measures seek to give a DM semantically meaning information of the amount of alteration their judgments have undergone to aid understanding and traceability of such scenarios. The set of measures offer various emphasis of compromise to suit different scenarios and DM preferences.

As inconsistency within a set of judgments adversely effects the accuracy of a derived preference vector its reduction is an important consideration. A new approach to the reduction of inconsistency within a set of judgments has been proposed. Unlike previous approaches the approach is not restrictive upon the type of inconsistency reduction it achieves. Alteration to a DM's judgments is modelled through the measures of compromise. Through modelling inconsistency and alteration as separate objectives the nature of the trade-offs involved between reducing inconsistency and alteration to a DM's views is revealed. The approach additionally facilitates constraints to be set upon inconsistency and compromise objectives, to define target thresholds of inconsistency, or levels of alteration allowable in the pursuit of inconsistency reduction.

For many real-world decisions the opinions of multiple DMs is utilised, either to avail of their combined expertise or to incorporate conflicting views and experiences. An approach has been proposed for the aggregation of a group of DMs' judgments. Within the approach the alteration to each DM's views is modelled as a separate objective using the measures of compromise. The amount of compromise of each DM's judgments is

revealed during aggregation which enhances the traceability and validity of the aggregation process. Interactive analysis regarding global compromise and fairest compromise aids a group towards reaching a consensus. The approach additionally enables constraints to be set by DMs as well as DM weights of importance to be incorporated dynamically into the aggregation process. The approach can also seek to reduce inconsistency during the aggregation process.

Scaling issues were identified through investigating using the newly proposed approach to group aggregation for increasingly larger groups of DMs. As a result an approach for the aggregation of a large group of DMs was proposed. The approach first utilises clustering to group the DMs based upon the similarity of their views. Next a single representation of the views of each cluster's members is derived. The approach then seeks to reach group aggregation with each cluster modelled as a separate objective. As the approach seeks to group similar DMs together before creating a single representation of each group it facilitates reduction in problem complexity through looking to identify the redundancy within the group. Additional analysis over a range of cluster values aids the selection of an appropriate number of clusters.

A Multi-objective Optimisation Decision Software (MOODS) tool has been developed that can be employed within multiple scenarios. The web-based decision support tool can be utilised by a single DM looking to reduce and understand their inconsistency implementing the new approach to inconsistency reduction. MOODS can additionally be utilised within group decision making and implements the new approach to group aggregation. Furthermore MOODS can be utilised for the aggregation of a large group of DMs and implements the new approach to the aggregation of a large number of DMs. The tool's extensible design facilitates future development to be easily implemented into its framework.

## **8.2 Future Work**

### **8.2.1 Modified Measures of Compromise**

Future work could investigate more closely the relationships between the measures of compromise, and to then seek to define additional measures that look to combine the traits of multiple measures together into modified measures of compromise.

### 8.2.1.1 *Correlation analysis of Measures of Compromise*

Investigations of the relationships between the measures of compromise could look to analyse, through experimentation, correlation between the measures. For a range of element values, we could create a random set of judgments and then through single objective optimisation eliminate the inconsistency in the judgments (for example reduce  $CR=0$ , which will also result in  $L=0$ ). Next we could calculate the values of the compromise measures to reach the set of consistent judgments. Then for each experiment a measure of the Pearson correlation [152] could be calculated between each pair of measures of compromise and averages calculated over a large number of experiments.

Results from initial testing of such experimentation and calculated correlation values when  $N=5$ , are shown in Table 8.1. We see that apart from TJD and STJD measures (which both are deviation measures) the level of correlation between the pairs appears to be below  $2/3$ . We additionally plot the results of such experimentation, for example, the values of NJR and STJD of the initial experimentation are shown in Figure 8.1.

Table 8.1: Correlation between measures of compromise,  $N=5$

	<b>NJV</b>	<b>TJD</b>	<b>STJD</b>	<b>NJR</b>
<b>NJV</b>	1	0.367	0.209	0.285
<b>TJD</b>	0.367	1	0.943	0.650
<b>STJD</b>	0.209	0.943	1	0.592
<b>NJR</b>	0.285	0.65	0.593	1

From this visualisation we see that there appears to be a general positive correlation between these two measures, however the higher the level of initial inconsistency (and thus the higher the amount of compromise required to reach full consistency) the less correlated the measures appear. This could be analysed further to see more specifically if we observe less correlation with higher initial inconsistency. Furthermore analysis looking at a range of  $N$  values would help to reveal patterns between the levels of correlation and the value of  $N$  between the measures of compromise.

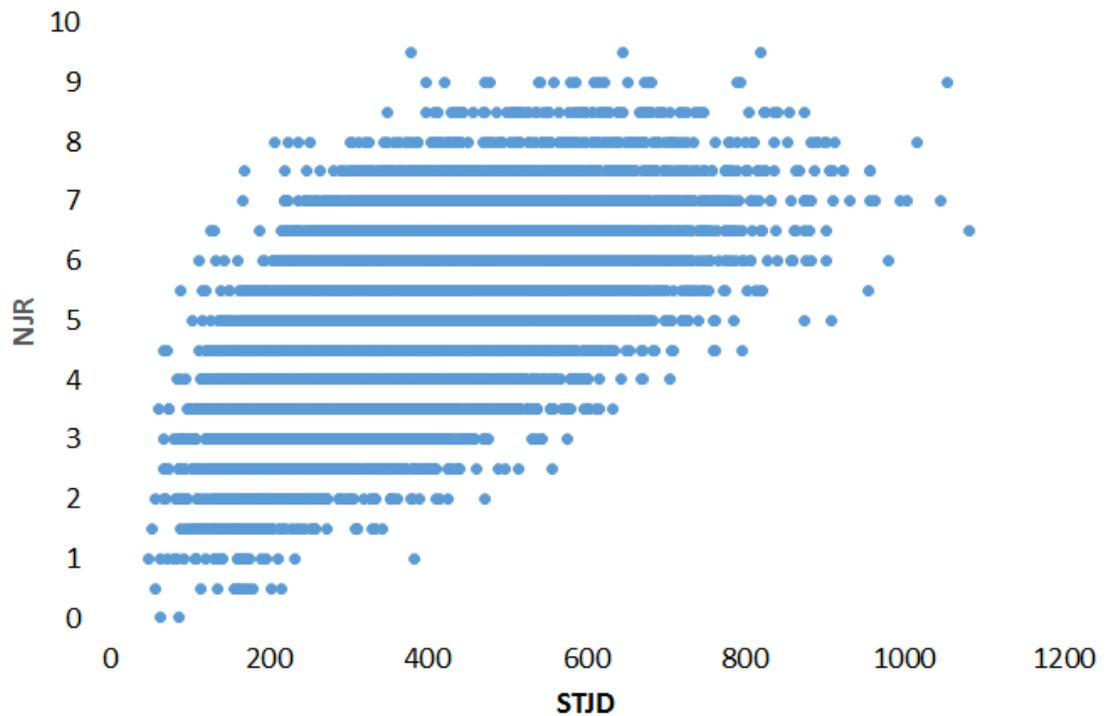


Figure 8.1: N=5. STJD and NJR correlation

Our initial correlation experimentation suggest that although there appears to be positive correlation between the measures of compromise they define compromise in different ways. Further experimentation of such correlation is an interesting avenue of investigation. Additionally as the measures of compromise define compromise in different ways a richer measure might be possible that looks to combine their traits together.

### 8.2.1.2 *Combined Measures of Compromise*

The analysis of the value of a measure of compromise over the range of scale values presented in Chapter 3 for each of the measures showed visually the difference in emphasis of each compromise measure. For example, NJR focuses upon when a reversal of a judgment has occurred without consideration of the strength of preference change, so between a judgment of 2 and a judgment of 9 NJR=0. Conversely TJD considers the amount of change without consideration of when a reversal has occurred, therefore the deviation between 2 and  $\frac{1}{2}$  is seen as less deviation compromise that the deviation between 2 and 9. These two example judgments are shown in Figure 8.2 along with their TJD and NJR measures. Further investigations could see if richer measures of compromise can be defined that look to consider the traits of multiple measures.

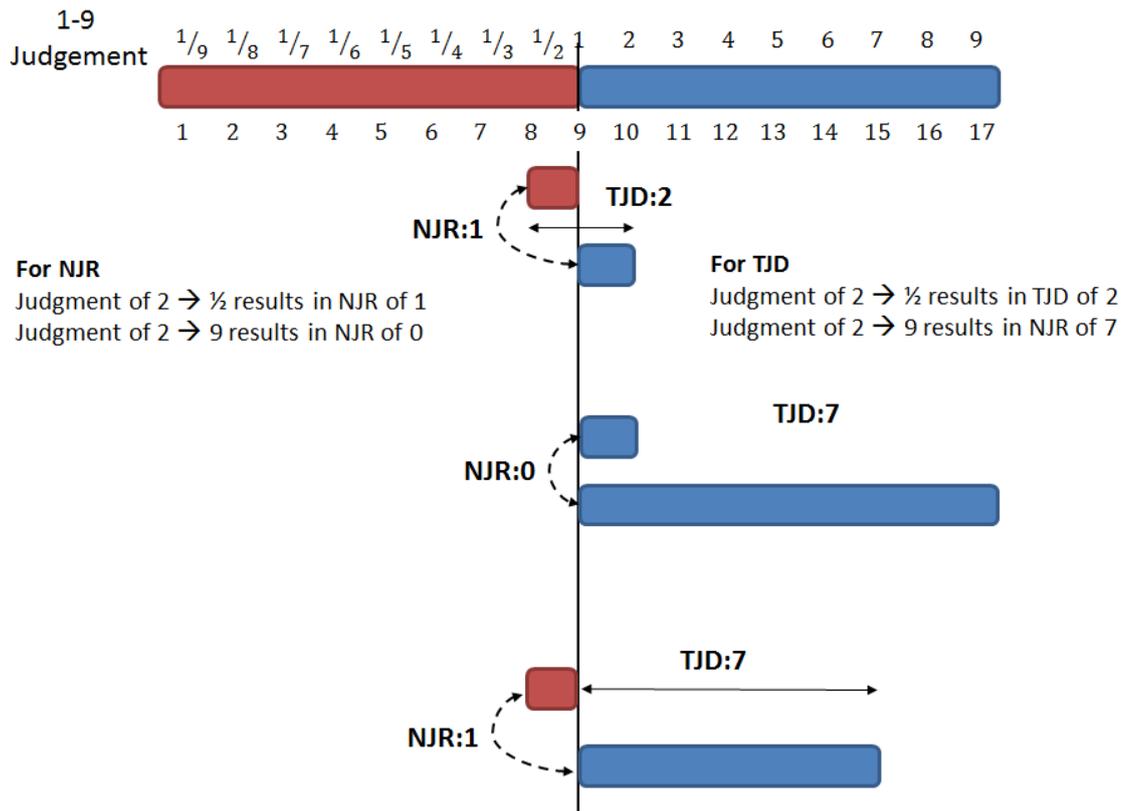


Figure 8.2 : Deviation and reversal measures comparison

A new measure for example, could be one that looks to consider both deviation and reversals within its calculation. To help consider this a 3<sup>rd</sup> judgment is shown in Figure 8.2, which involves 1 reversal and a deviation value of 7. Measured via TJD this 3<sup>rd</sup> judgment is seen as being at the same level of compromise as the 2<sup>nd</sup> judgment, and measured via NJR this 3<sup>rd</sup> judgment is seen as being at the same level of compromise as the 1<sup>st</sup> judgment. Yet we could conjecture that the 3<sup>rd</sup> judgment has undergone greater overall alteration than the 1<sup>st</sup> or 2<sup>nd</sup> judgment. One possible way we could investigate looking to define a measure that considers deviation and reversals together could be through utilising the deviation measure of a judgment with an added modifier  $m$  to give emphasis to when a reversal has occurred (and when a half reversals occurs as  $m/2$ ), as illustrated in Figure 8.3. Such a measure might be termed Modified Total Judgment Deviation (MTJD).

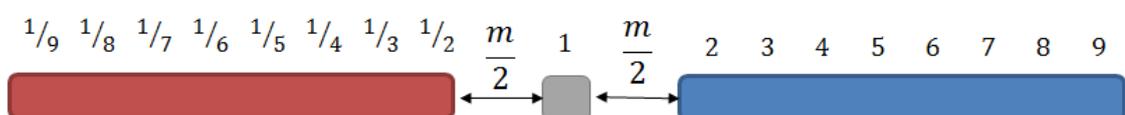


Figure 8.3: Total judgment deviation with reversals modification measure.

Further analysis could investigate rationale and formalisation of an appropriate value for  $m$ . If we initially conjecture that a reversal should represent more compromise than any deviation without a reversal (when utilizing a bounded scale), then when using the 1-9 scale the largest amount of deviation without a reversal  $2 \rightarrow 9$  should represent less compromise than the smallest deviation that also represents a reversal, which is  $\frac{1}{2} \rightarrow 2$ . Through analysis of the compromise of the judgment  $\frac{1}{2} \rightarrow 2$  over a range of different  $m$  values we summarise that  $m$  needs to be at least 6 to fulfil this rationale. When  $m=6$ ,  $\frac{1}{2} \rightarrow 2$  will measure  $2 + 6 = 8$  greater than  $2 \rightarrow 9$ , which will be 7.

When  $m=6$  the value of MTJD across the range of values from the scale is shown in Figure 8.4. From this we observe the emphasis of a reversal upon the measure. Additionally we observe that it upholds the initial conjecture that a full reversal should represent more compromise than the largest deviation without a reversal. In the plot, we observe that the compromise value of a judgment of  $\frac{1}{2}$ , altered to 2 is greater value than the compromise value of a judgment  $\frac{1}{2}$  altered to  $1/9$ .

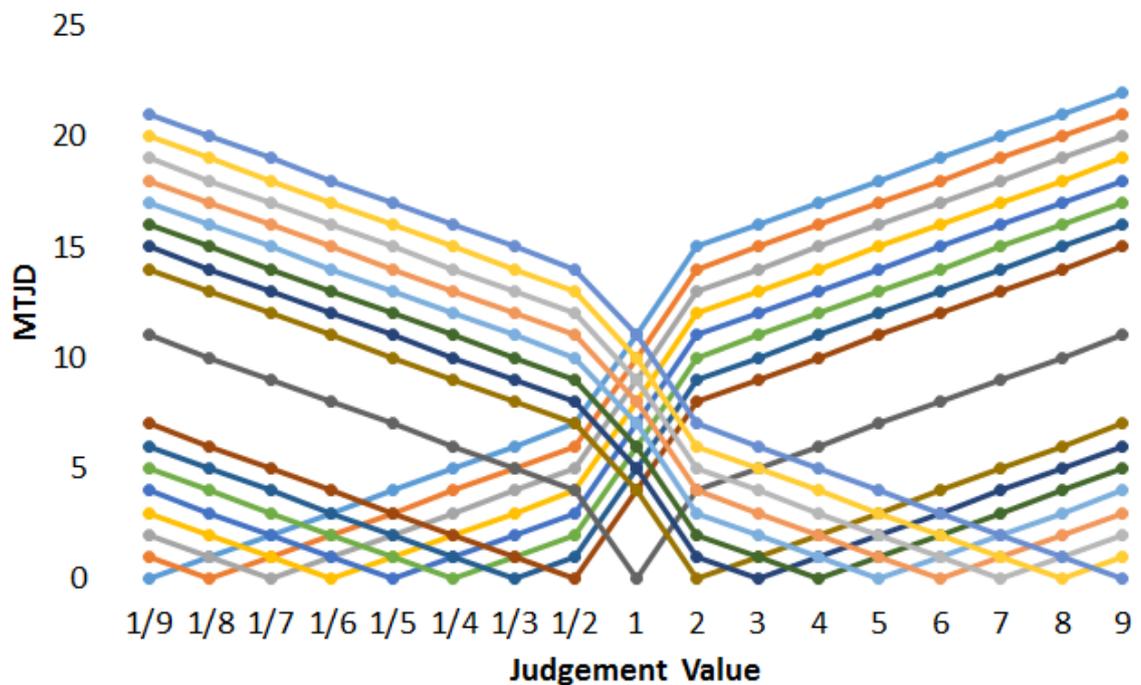


Figure 8.4: Modified TJD (MTJD).

An investigation of such ideas to combine the traits of the measures of compromise would be of interest. Further analysis could look to more formally define such combination processes as well as considering the other measures not considered above, STJD and NJV. From such analysis could we define a complete measure of compromise? Or could a

modified measure be defined that looks to actively incorporate a DM's preferences into the measure, thus defining their sentiments towards compromise into a measure?

## **8.2.2 Additional Clustering Work**

Various future investigations regarding the clustering approach could explore, analysis of clustering distance functions, analysis of utilizing different clustering algorithms and analysis of utilizing preference vectors as clustering inputs.

### **8.2.2.1 Clustering distance function analysis**

When clustering data points within algorithms such as k-means, the similarity between data points is determined via a distance function, that looks to cluster together data points in close proximity. The most commonly utilised distance measure is that of Euclidian distance [153]. Given two points we determine the Euclidian distance between them across their dimensions via the Pythagorean formula. Within the approach to the aggregation of a large number of DMs a set of judgments from each DM is utilised as the input to the clustering stage. We can discern that the Euclidian distance measure is similar to the compromise measures that are focused on deviation, without consideration of, for example, judgment reversals. Therefore further work could investigate enhancing the clustering stage via explicitly incorporating consideration of the traits of the measures of compromise into the clustering process, to seek a richer clustering result. Two possible approaches to investigate achieving this could be:

1. Creating a custom distance function, which take into consideration more complex calculations to determine the distance between instances to be clustered. Such a function could, for example, look to determine distance between DMs taking into account the amount of deviation between their views but with added consideration to emphasise when a reversal occurs.
2. Modify the encoded input vector of each DM before clustering, for example, by adding emphasis to reversals before clustering begins. For example, we could take the deviation measure of a judgment and add a modifier  $m$  to give emphasis to a reversal (and add a modifier of  $m/2$  for a half reversals), similar to the method shown in Figure 8.3. In this way we could modify an encoded judgment set to incorporate emphasis for reversals before the clustering stage.

### **8.2.2.2 Analysis of different clustering algorithms**

Within the approach for aggregating a large group of DMs the use of the k-means++ clustering algorithm facilitates a swift and stable clustering solution. Analysis over a range of k values aids selection of an appropriate k-value. Further work could investigate and compare the use of different clustering algorithms and their applicability within the clustering stage of the approach.

For example, Hierarchical clustering [154] could be employed to cluster a large group of DMs either top down, iteratively splitting a single cluster into many, or bottom up, iteratively merging clusters into one. Such an approach would then reveal a trail of the process of splitting or merging which can be visualised as a tree structure. An illustration for 6 DMs is shown in Figure 8.5: Left. Hierarchical clustering, as with k-means, needs a user to define the number of clusters to group a set of DMs, however as the distance at each iteration is calculated a distance threshold value could be defined to stop clustering when the threshold is reached. Additionally performing clustering from all DMs in separate clusters to all DMs in a single cluster would allow swift sensitivity analysis of selecting different numbers of clusters.

Similarly fuzzy clustering could be investigated. Instead of hard clustering such as k-means where every DM is assigned to only a single cluster we could investigate fuzzy clustering of DMs utilizing Fuzzy C-means (FCM) [155]. With such an algorithm after the clustering stage each DM would have a set of membership values of the probability they belong to each cluster, as illustrated in Figure 8.5: Right. We could look to incorporate this membership data into the stage of creating single representations for each cluster. We could also look to perform sensitivity analysis over various single representations of a cluster over a range of threshold of membership values.

Additionally the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [156] algorithm could be investigated. This algorithm looks to group together instances that are closely packed, that is instances which have many nearby neighbours. Its density-based approach would additionally allow for the identification of instances that lie alone in low-density regions of the clustering space. In this way the algorithm is robust against outliers affecting the results and could identify outlying DMs during the clustering stage. We could then also have more explicit consideration of outliers to help look for malicious users trying to skew results. This could be an important area of investigation for a very large group of DMs involved in crowd sourcing activities.

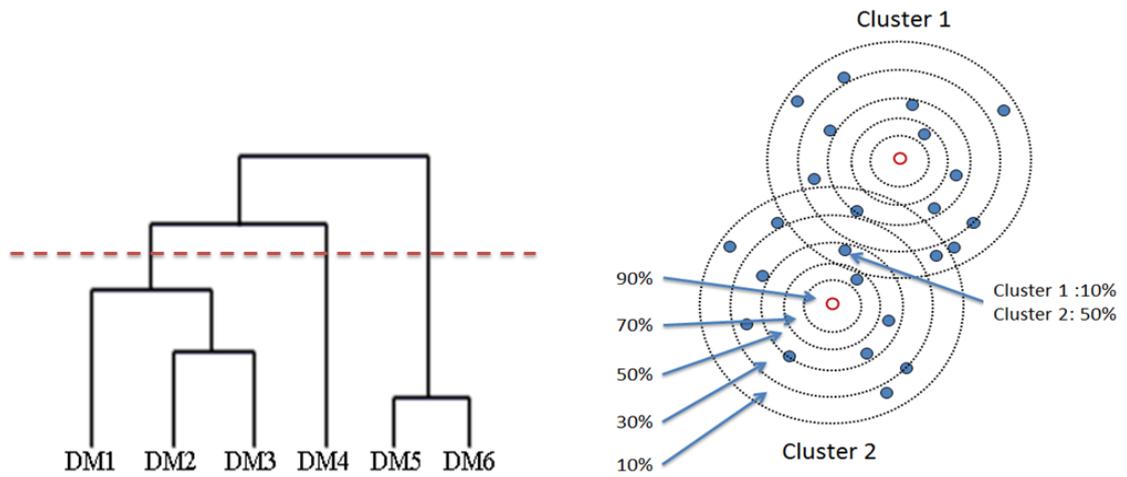


Figure 8.5 : Left: Hierarchical clustering. Right: Fuzzy clustering

### 8.2.2.3 Utilizing preference vectors as inputs

Another area of future investigations could be to look to utilise preference vectors as input from each DM for the clustering stage. Then the approach could cluster DMs utilizing different approaches to elicit their views; any method that enables creation of a ratio ranking of elements could be utilised. Such an approach would have to consider how single representations of each cluster's members would be created. With preference vectors as input the approach would then look to derive a preference vector as a single representation. This stage could utilise preference vector evaluation measures such as TD or NV as objectives to look to create a preference vector that is representational of the views of a cluster's members. These single representations could then be utilised within the MOO stage, again using preference vector evaluation measures such as TD or NV as objectives.

## 8.2.3 Further investigations of large objective set optimisation

### 8.2.3.1 Tackling scaling issue stagnation through total objective measures

Further investigations could look into alternative approaches to tackling the scaling issues of aggregation when the number of DMs is large. When stagnation occurs within a large number of objectives an interesting observed occurrence is that the addition of the corresponding Total Compromise measure into the objective set alleviates some of this stagnation. For example, when 20 DMs are aggregated each modelled as a single objective with say STJD, the stagnation that occurs during MOO can to some extent be alleviated through the addition of a 21<sup>st</sup> objective of TotalSTJD. In this case the additional objective is helping to distinguish between solutions which are non-dominated solutions with respect to the other 20 objectives. The TotalSTJD objective helps to prevent the

population from quickly becoming full of non-dominated individuals. Essentially the additional objective acts as a modification of the Pareto dominance relation so is akin to approaches to tackling many objective problems that are non-redundancy approaches [135]. Further investigations could analyse the potential of such an approach.

### **8.2.3.2 *DM group redundancy averages and formula***

From the experimentation in Chapter 6 measuring redundancy within groups of DMs over a range of  $k$  values, we were able to reveal for an  $N$  and value  $D$  pair the average level of redundancy over the range of  $k$  values 1 to  $D$ . From this experimentation we were able to identify that redundancy was present within DM groups and that the clustering approach is an appropriate technique to identify such redundancy. Further investigations could seek to define more generally the levels of redundancy over ranges of  $N$  and  $D$  values. From this a formula of redundancy could be extracted, such that given  $N$ ,  $D$  and  $K$  values the average amount of redundancy could be predicted. This could also facilitate comparison of a group of DMs to these average levels to aid in the selection of an appropriate  $k$ -value.

### **8.2.3.3 *Percentage of true fronts found for 2 DMs***

During experimentation to identify scaling issues in Chapter 6 it was observed that, as the number of DMs increase the size of solution sets found via MOGAs as a percentage of the size of the true Pareto front tails off, as shown in Figure 6.3. It was also observed that for 2 DMs the performance was noticeably low compared to 3 and 4 DMs before the performance then began to tail off for  $D$  values greater than 4. Further work could look to investigate this.

## **8.2.4 MOODS tool extensions and enhancements**

The MOODS software tool can be further developed and enhanced. An area of enhancement is to implement a database to store problems and their data, this way providing a more persistent storage solution. Such a database could also be utilised to facilitate the storage of user profiles and for group decisions DMs could be assigned through their usernames. Additionally to increase accessibility a mobile interface could be developed utilizing the functionality of the business logic layer within a mobile friendly interface.

Other enhancements could investigate facilitating increased interactivity. Investigating plotting 3-dimensional plots within JavaScript code could be investigated. This would allow the objective space to be viewed from the perspective of 3 objectives at once. Increased traceability functionality could also be implemented by, for example,

implementing a tab view within the objective space to record its state each time it is dynamically updated, thus during say the iterative adding of constraints the objective space after each stage would be reviewable through a series of tabs.

### **8.3 Conclusion**

For many decisions validation of their outcomes regarding correctness and acceptance is invariably problematic to objectively assess. Therefore to aid validation of decision outcomes, approaches with improved traceability and more semantically meaningful measurements offer an advantage to DMs by providing more evidence of the process. During scenarios such as inconsistency reduction or group aggregation, a DM's judgments undergo alteration. Traceable approaches to such scenarios reveal knowledge of the scenario helping to create a richer process, and aid validation of the scenario outcomes. Measures of compromise have been defined to measure the amount of alteration a DM's views undergo, in semantically meaningful ways, to aid traceability and understanding of the alteration. Inconsistency adversely effects decision outcomes and its reduction is important. An approach has been proposed to reduce inconsistency in a traceable way that enables understanding of the trade-offs involved between reduction and alteration to a DM's views. The problem of finding an aggregated view in group decision making from the set of DMs' views has been addressed and an approach to group aggregation has been proposed. The approach facilitates traceability and interactivity of the aggregation process to aid a group of DMs to reach a consensus. Scaling issues when the approach is utilised to aggregate the views of a large group of DMs have been identified and an approach has been proposed to overcome these limitations through utilizing clustering. Finally, a web-based software tool has been developed that implements these approaches within a responsive browser-based interface. The research has concluded with identification and discussions of several areas of future investigations.

# Bibliography

- [1] S. French, J. Maule, and N. Papamichail, *Decision Behaviour, Analysis and Support [Paperback]*. Cambridge University Press; 1 edition, 2009.
- [2] P. G. W. Keen and M. S. S. Morton, *Decisión support systems: an organizational perspective*. Addison-Wesley Pub. Co., 1978.
- [3] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956.,” *Psychol. Rev.*, vol. 101, no. 2, pp. 343–352, 1956.
- [4] E. U. Choo and W. C. Wedley, “A common framework for deriving preference values from pairwise comparison matrices,” *Comput. Oper. Res.*, vol. 31, no. 6, pp. 893–908, May 2004.
- [5] R. Słowiński, “Basic Notions of Multiple Criteria Decision Aiding,” in *European Spring School on MCDA*, 2014.
- [6] G. Box and N. Draper, “Empirical Model-Building and Response Surfaces,” *PsycCRITIQUES*, vol. 34, no. 5. 1989.
- [7] E. Triantaphyllou, *Multi-criteria Decision Making Methods: A Comparative Study*. Springer; 2000 edition, 2000.
- [8] 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,” *Rev. la Real Acad. Ciencias Exactas, Fis. y Nat. Ser. A. Mat.*, vol. 102, no. 2, pp. 251–318, 2008.
- [9] B. Roy, “The Optimisation Problem Formulation: Criticism and Overstepping,” *J. Oper. Res. Soc.*, vol. 32, no. 6, pp. 427–436, Jun. 1981.
- [10] C. A. Costa, “Les problématiques de l’aide à la décision : vers l’enrichissement de la trilogie choix-tri-rangement,” *RAIRO - Oper. Res. - Rech. Opérationnelle*, vol. 30, no. 2, pp. 191–216.
- [11] R. L. Keeney, “Value-focused thinking: Identifying decision opportunities and creating alternatives,” *Eur. J. Oper. Res.*, vol. 92, no. 3, pp. 537–549, Aug. 1996.
- [12] C. Hwang and K. Yoon, *Multiple attribute decision making: methods and applications : a state-of-the-art survey*. Springer-Verlag, 1981.
- [13] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge University Press, 1993.
- [14] E. Jacquet-Lagrange and J. Siskos, “Assessing a set of additive utility functions for multicriteria decision-making, the UTA method,” *Eur. J. Oper. Res.*, vol. 10, no. 2, pp. 151–164, Jun. 1982.
- [15] S. Greco, V. Mousseau, and R. Słowiński, “Ordinal regression revisited: Multiple criteria ranking using a set of additive value functions,” *Eur. J. Oper. Res.*, vol. 191, no. 2, pp. 416–436, Dec. 2008.
- [16] J. R. Figueira, S. Greco, and R. Słowiński, “Building a set of additive value functions representing a reference preorder and intensities of preference: GRIP method,” *Eur. J. Oper. Res.*, vol. 195, no. 2, pp. 460–486, Jun. 2009.
- [17] A. Ishizaka and P. Nemery, *Multicriteria decision analysis: methods and software*. Wiley, 2013.

- [18] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, 1980.
- [19] F. Zahedi, "The Analytic Hierarchy Process--A Survey of the Method and its Applications," *Interfaces (Providence)*, vol. 16, no. 4, pp. 96–108, 1986.
- [20] A. Ishizaka, "Clusters and pivots for evaluating a large number of alternatives in AHP," *Pesqui. Operacional*, vol. 32, no. 1, pp. 87–102, Apr. 2012.
- [21] A. Ishizaka, C. Pearman, and P. Nemery, "AHPSort: an AHP-based method for sorting problems," *Int. J. Prod. Res.*, vol. 50, no. 17, pp. 4767–4784, Sep. 2012.
- [22] B. Schoner and W. C. Wedley, "Ambiguous Criteria Weights in AHP: Consequences and Solutions," *Decis. Sci.*, vol. 20, no. 3, pp. 462–475, 1989.
- [23] T. L. Saaty, *Decision Making with Dependence and Feedback: The Analytic Network Process : the Organization and Prioritization of Complexity*. Rws Publications, 2001.
- [24] J. S. Dyer, "Remarks on the analytic hierarchy process," *Manage. Sci.*, vol. 36, no. 3, pp. 249–258, 1990.
- [25] J. P. B. Bertrand Mareschal, "Prométhée: a new family of outranking methods in multicriteria analysis," *Oper. Res. - ORIJ*, 1984.
- [26] J. P. Brans and B. Mareschal, "Promethee Methods," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco, and M. Ehrgott, Eds. Springer Verlag, 2005, pp. 163–196.
- [27] J. P. Brans, P. Vincke, and B. Mareschal, "How to select and how to rank projects: The Promethee method," *European Journal of Operational Research*, vol. 24, no. 2, pp. 228–238, 1986.
- [28] J. P. Brans and B. Mareschal, "Promethee-V - Mcdm Problems with Segmentation Constraints," *INFOR*, vol. 30, no. 2, pp. 85–96, 1992.
- [29] B. Roy, "Classement et choix en présence de points de vue multiples," *RAIRO - Oper. Res. - Rech. Opérationnelle*, vol. 2, no. V1, pp. 57–75.
- [30] J. Figueira, V. Mousseau, and B. Roy, "ELECTRE Methods," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, 2005, pp. Vol. 78, 133–172.
- [31] L. L. Thurstone, "A law of comparative judgment.," *Psychol. Rev.*, vol. 101, no. 2, pp. 266–270, 1927.
- [32] J. Til, C. Groothuis-Oudshoorn, M. Lieferink, J. Dolan, and M. Goetghebeur, "Does technique matter; a pilot study exploring weighting techniques for a multi-criteria decision support framework.," *Cost Eff. Resour. Alloc.*, vol. 12, no. 1, p. 22, Jan. 2014.
- [33] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *J. Math. Psychol.*, vol. 15, no. 3, pp. 234–281, Jun. 1977.
- [34] A. Ishizaka, D. Balkenborg, and T. Kaplan, "Influence of aggregation and measurement scale on ranking a compromise alternative in AHP," *J. Oper. Res. Soc.*, vol. 62, no. 4, pp. 700–710, 2011.
- [35] J. S. Finan and W. J. Hurley, "Transitive calibration of the AHP verbal scale," *Eur. J. Oper. Res.*, vol. 112, no. 2, pp. 367–372, 1999.
- [36] P. T. Harker and L. G. Vargas, "The theory of ratio scale estimation: Saaty's analytic hierarchy process," *Manage. Sci.*, vol. 33, no. 11, pp. 1383–1403, 1987.
- [37] F. A. Lootsma, "Conflict resolution via pairwise comparison of concessions," *European Journal of Operational Research*, vol. 40, no. 1, pp. 109–116, 1989.

- [38] D. Ma and X. Zheng, "9/9-9/1 Scale Method of AHP," in *2nd Int. Symposium on AHP*, 1991, pp. 197–202.
- [39] A. A. Salo and R. P. Hämäläinen, "On the measurement of preferences in the analytic hierarchy process," *J. Multi-Criteria Decis. Anal.*, vol. 6, no. 6, pp. 309–319, 1997.
- [40] C. C. Lin, "A revised framework for deriving preference values from pairwise comparison matrices," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 1145–1150, 2007.
- [41] B. Srdjevic, "Combining different prioritization methods in the analytic hierarchy process synthesis," *Comput. Oper. Res.*, vol. 32, no. 7, pp. 1897–1919, 2005.
- [42] G. B. Crawford, "The geometric mean procedure for estimating the scale of a judgement matrix," *Math. Model.*, vol. 9, no. 3–5, pp. 327–334, 1987.
- [43] S. Siraj, L. Mikhailov, and J. Keane, "Enumerating all spanning trees for pairwise comparisons," *Comput. Oper. Res.*, vol. 39, no. 2, pp. 191–199, Feb. 2012.
- [44] H. N. Gabow and E. W. Myers, "Finding All Spanning Trees of Directed and Undirected Graphs," *SIAM J. Comput.*, vol. 7, no. 3, pp. 280–287, Aug. 1978.
- [45] A. Chu, R. E. Kalaba, and K. Spingarn, "A comparison of two methods for determining the weights of belonging to fuzzy sets," *J. Optim. Theory Appl.*, vol. 27, no. 4, pp. 531–538, 1979.
- [46] B. Golany and M. Kress, "A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices," *Eur. J. Oper. Res.*, vol. 69, no. 2, pp. 210–220, 1993.
- [47] E. Blankmeyer, "Approaches to consistency adjustment," *J. Optim. Theory Appl.*, vol. 54, no. 3, pp. 479–488, Sep. 1987.
- [48] W. D. Cook and M. Kress, "Deriving weights from pairwise comparison ratio matrices: An axiomatic approach," *European Journal of Operational Research*, vol. 37, no. 3, pp. 355–362, 1988.
- [49] L. Mikhailov, "A fuzzy programming method for deriving priorities in the analytic hierarchy process," *Journal of the Operational Research Society*, vol. 51, no. 3, pp. 341–349, 2000.
- [50] L. Mikhailov, "Multiobjective Prioritisation in the Analytic Hierarchy Process using Evolutionary Computing," in *Applications of Soft Computing: Advances in Intelligent and Soft Computing Volume 36*, A. Tiwari, J. Knowles, E. Avineri, K. Dahal, and R. Roy, Eds. Springer, 2006, pp. 321–330.
- [51] S. Siraj, L. Mikhailov, and J. Keane, "Preference elicitation from inconsistent judgments using multi-objective optimization," *Eur. J. Oper. Res.*, vol. 220, no. 2, pp. 461–471, Jul. 2012.
- [52] N. Bryson, "A Goal Programming Method for Generating Priority Vectors," *J. Oper. Res. Soc.*, vol. 46, no. 5, pp. 641–648, 1995.
- [53] C. Lin, "An enhanced goal programming method for generating priority vectors," *J. Oper. Res. Soc.*, vol. 57, no. 12, pp. 1491–1496, 2006.
- [54] R. E. Jensen, "Comparisons of eigenvector, least squares, chi square, and logarithmic least squares methods of scaling a reciprocal matrix," 1983.
- [55] Z. S. Xu, "Generalized Chi Square Method for the Estimation of Weights," *J. Optim. Theory Appl.*, vol. 107, no. 1, pp. 183–192, 2000.
- [56] P. Laininen and R. P. Hämäläinen, "Analyzing AHP-matrices by regression," *Eur. J. Oper. Res.*, vol. 148, no. 3, pp. 514–524, 2003.
- [57] R. Ramanathan, "Data envelopment analysis for weight derivation and aggregation in the analytic

- hierarchy process,” *Comput. Oper. Res.*, vol. 33, no. 5, pp. 1289–1307, 2006.
- [58] M. W. Herman and W. W. Koczkodaj, “A Monte Carlo study of pairwise comparison,” *Inf. Process. Lett.*, vol. 57, no. 1, pp. 25–29, 1996.
- [59] L. Mikhailov and J. Knowles, “Priority elicitation in the AHP by a pareto envelope-based selection algorithm,” *Lect. Notes Econ. Math. Syst.*, vol. 634, pp. 249–257, 2010.
- [60] I. Ali, W. D. Cook, and M. Kress, “On the Minimum Violations Ranking of a Tournament,” *Manage. Sci.*, vol. 32, no. 6, pp. 660–672, 1986.
- [61] T. L. Saaty, “Eigenvector and logarithmic least squares,” *European Journal of Operational Research*, vol. 48, no. 1, pp. 156–160, 1990.
- [62] C. R. Johnson, W. B. Beine, and T. J. Wang, “Right-left asymmetry in an eigenvector ranking procedure,” *J. Math. Psychol.*, vol. 19, no. 1, pp. 61–64, 1979.
- [63] W. W. Koczkodaj and S. J. Szarek, “On distance-based inconsistency reduction algorithms for pairwise comparisons,” *Log. J. IGPL*, vol. 18, no. 6, pp. 859–869, 2010.
- [64] S. Lipovetsky and W. Michael Conklin, “Robust estimation of priorities in the AHP,” *Eur. J. Oper. Res.*, vol. 137, no. 1, pp. 110–122, 2002.
- [65] R. Sugden, “Why be Consistent? A Critical Analysis of Consistency Requirements in Choice Theory,” *Economica*, vol. 52, no. 206, pp. 167–183, 1985.
- [66] S. I. Gass, “Tournaments, transitivity and pairwise comparison matrices,” *J. Oper. Res. Soc.*, vol. 49, no. 6, pp. 616–624, 1998.
- [67] M. G. Kendall, “Further Contributions to the Theory of Paired Comparisons,” *Biometrics*, vol. 11, no. 1, pp. 43–62, 1955.
- [68] Y. Iida, “The Number of Circular Triads in a Pairwise Comparison Matrix and a Consistency Test in the AHP,” *J. Oper. Res. Soc. Japan*, vol. 52, no. 2, pp. 174–185, 2009.
- [69] R. E. Jensen and T. E. Hicks, “Ordinal data AHP analysis: A proposed coefficient of consistency and a nonparametric test,” *Math. Comput. Model.*, vol. 17, no. 4–5, pp. 135–150, 1993.
- [70] M. Kwiesielewicz and E. van Uden, “Inconsistent and contradictory judgements in pairwise comparison method in the AHP,” *Comput. Oper. Res.*, vol. 31, no. 5, pp. 713–719, 2004.
- [71] F. Harary and L. Moser, “The Theory of Round Robin Tournaments,” *Am. Math. Mon.*, vol. 73, no. 3, pp. 231–246, 1966.
- [72] J. Aguaron and J. M. Moreno-Jimenez, “The geometric consistency index: Approximated thresholds,” *Eur. J. Oper. Res.*, vol. 147, no. 1, pp. 137–145, 2003.
- [73] W. W. Koczkodaj, “A new definition of consistency for pairwise comparisons,” *Inf. Process. Lett.*, vol. 52, no. 5, pp. 273–276, 1993.
- [74] G. Crawford and C. Williams, “A note on the analysis of subjective judgment matrices,” *J. Math. Psychol.*, vol. 29, no. 4, pp. 387–405, 1985.
- [75] J. I. Peláez and M. T. Lamata, “A new measure of consistency for positive reciprocal matrices,” *Comput. Math. with Appl.*, vol. 46, pp. 1839–1845, 2003.
- [76] P. Ji and R. Jiang, “Scale transitivity in the AHP,” *Journal of the Operational Research Society*, vol. 54, pp. 896–905, 2003.
- [77] P. T. Harker, “Derivatives of the Perron root of a positive reciprocal matrix: With application to the analytic hierarchy process,” *Appl. Math. Comput.*, vol. 22, no. 2–3, pp. 217–232, May 1987.

- [78] T. L. Saaty, "Decision-making with the AHP: Why is the principal eigenvector necessary," *Eur. J. Oper. Res.*, vol. 145, no. 1, pp. 85–91, Feb. 2003.
- [79] W. Holsztyński and W. W. Koczkodaj, "Convergence of inconsistency algorithms for the pairwise comparisons," *Inf. Process. Lett.*, vol. 59, no. 4, pp. 197–202, 1996.
- [80] S. Siraj, "Preference Elicitation from Pairwise Comparisons," The University of Manchester, 2011.
- [81] Z. Xu and C. Wei, "Consistency improving method in the analytic hierarchy process," *Eur. J. Oper. Res.*, vol. 116, no. 2, pp. 443–449, 1999.
- [82] D. Cao, L. C. Leung, and J. S. Law, "Modifying inconsistent comparison matrix in analytic hierarchy process: A heuristic approach," *Decis. Support Syst.*, vol. 44, no. 4, pp. 944–953, 2008.
- [83] S. Siraj, L. Mikhailov, and J. Keane, "A heuristic method to rectify intransitive judgments in pairwise comparison matrices," *Eur. J. Oper. Res.*, vol. 216, no. 2, pp. 420–428, Jan. 2012.
- [84] J. F. da S. Costa, "A Genetic Algorithm to Obtain Consistency in Analytic Hierarchy Process," *Brazilian Journal of Operations & Production Management*, vol. 8, no. 1, pp. 55–64, 2011.
- [85] C. Wang, S. Liu, and C. Pang, "Using genetic algorithm improve the consistency of fuzzy analytic hierarchy process," in *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, 2012, pp. 977–982.
- [86] X. Sun, Q. Liu, and L. Zhang, "Consistency Modification of Judgment Matrix Based on Genetic Algorithm in Analytic Hierarchy Process," *2011 Third Pacific-Asia Conf. Circuits, Commun. Syst.*, pp. 1–4, 2011.
- [87] A. Ishizaka and A. Labib, "Selection of new production facilities with the Group Analytic Hierarchy Process Ordering method," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7317–7325, Jun. 2011.
- [88] E. N. Weiss and V. R. Rao, "AHP DESIGN ISSUES FOR LARGE-SCALE SYSTEMS," *Decis. Sci.*, vol. 18, no. 1, pp. 43–61, Jan. 1987.
- [89] R. Ramanathan and L. S. Ganesh, "Group preference aggregation methods employed in AHP: An evaluation and an intrinsic process for deriving members' weightages," *Eur. J. Oper. Res.*, vol. 79, no. 2, pp. 249–265, 1994.
- [90] S. Jaganathan, J. J. Erinjeri, and J. Ker, "Fuzzy analytic hierarchy process based group decision support system to select and evaluate new manufacturing technologies," *Int. J. Adv. Manuf. Technol.*, vol. 32, no. 11–12, pp. 1253–1262, Apr. 2006.
- [91] Y. Cho and K. Cho, "A loss function approach to group preference aggregation in the AHP," *Comput. Oper. Res.*, vol. 35, no. 3, pp. 884–892, Mar. 2008.
- [92] E. Condon, B. Golden, and E. Wasil, "Visualizing group decisions in the analytic hierarchy process," *Comput. Oper. Res.*, vol. 30, no. 10, pp. 1435–1445, Sep. 2003.
- [93] T. Saaty, "Group Decision Making and the AHP," in *The analytic Hierarchy Process. Applications and Studies*, B. Golden, E. Wasil, and P. Harker, Eds. Springer Berlin Heidelberg, 1989, pp. 59–67.
- [94] F. A. Lootsma, J. M. Sluijs, and S. Y. Wang, "Pairwise comparison of concessions in negotiation processes," *Gr. Decis. Negot.*, vol. 3, no. 1, pp. 121–131, Apr. 1994.
- [95] T. L. Saaty and L. Vargas, "The possibility of group welfare functions," *Int. J. Inf. Technol. Decis. Mak.*, vol. 04, no. 02, pp. 167–176, Jun. 2005.
- [96] J. Aczél and T. L. Saaty, "Procedures for synthesizing ratio judgements," *J. Math. Psychol.*, vol. 27, no. 1, pp. 93–102, Mar. 1983.

- [97] R. C. Van Den Honert and F. A. Lootsma, "Group preference aggregation in the multiplicative AHP The model of the group decision process and Pareto optimality," *Eur. J. Oper. Res.*, vol. 96, no. 2, pp. 363–370, 1997.
- [98] E. Forman and K. Peniwati, "Aggregating individual judgments and priorities with the analytic hierarchy process," *European Journal of Operational Research*, vol. 108, no. 1, pp. 165–169, 1998.
- [99] M. T. Escobar, J. Aguaron, and J. M. Moreno-Jimenez, "A note on AHP group consistency for the row geometric mean prioritization procedure," in *European Journal of Operational Research*, 2004, vol. 153, no. 2, pp. 318–322.
- [100] Z. Xu, "On consistency of the weighted geometric mean complex judgement matrix in AHP," *Eur. J. Oper. Res.*, vol. 126, no. 3, pp. 683–687, Nov. 2000.
- [101] L. Mikhailov, "Group prioritization in the AHP by fuzzy preference programming method," *Comput. Oper. Res.*, vol. 31, no. 2, pp. 293–301, Feb. 2004.
- [102] P. Gargallo, J. M. Moreno-Jiménez, and M. Salvador, "AHP-Group Decision Making: A Bayesian Approach Based on Mixtures for Group Pattern Identification," *Gr. Decis. Negot.*, vol. 16, no. 6, pp. 485–506, Jan. 2007.
- [103] N. Bryson and A. Joseph, "Generating consensus priority interval vectors for group decision-making in the AHP," *J. Multi-Criteria Decis. Anal.*, vol. 9, no. 4, pp. 127–137, Jul. 2000.
- [104] C. Madu and C. Kuei, "Stability analyses of group decision making," *Comput. Ind. Eng.*, vol. 28, no. 4, pp. 881–892, Oct. 1995.
- [105] T. L. Saaty and L. G. Vargas, "Dispersion of group judgments," *Math. Comput. Model.*, vol. 46, no. 7–8, pp. 918–925, Oct. 2007.
- [106] B. Mareschal and J.-P. Brans, "Geometrical representations for MCDA," *Eur. J. Oper. Res.*, vol. 34, no. 1, pp. 69–77, Feb. 1988.
- [107] E. Alpaydin, *Introduction to Machine Learning*. MIT Press; 2nd Revised edition edition, 2010.
- [108] C. Macharis and J. P. Brans, "The GDSS PROMETHEE procedure," *J. Decis. Syst.*, vol. 7, pp. 283–307, 1998.
- [109] J. W. Sammon, "A Nonlinear Mapping for Data Structure Analysis," *IEEE Trans. Comput.*, vol. C-18, no. 5, pp. 401–409, May 1969.
- [110] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Comput. Intell. Mag.*, pp. 28–36, 2006.
- [111] M. Gen, R. Cheng, and L. Lin, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*. Springer, 2008.
- [112] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, 1975.
- [113] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning [Hardcover]*. Addison Wesley; 1 edition, 1989.
- [114] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*, vol. 1, G. J. E. Rawlins, Ed. Morgan Kaufmann, 1991, pp. 69–93.
- [115] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms [Hardcover]*. Wiley-Blackwell; 2nd Edition edition, 2004.
- [116] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," ... *1st Int.*

*Conf. Genet. Algorithms*, pp. 93–100, 1985.

- [117] C. Fonseca and P. Fleming, “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization,” *Proc. Conf. Genet. Algorithms*, no. July, pp. 416–423, 1993.
- [118] J. Horn, N. Nafpliotis, and D. E. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” *Proc. First IEEE Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, 1994.
- [119] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [120] N. Srinivas and K. Deb, “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [121] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.
- [122] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization,” in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.
- [123] E. Alba, B. Dorronsoro, F. Luna, A. J. Nebro, P. Bouvry, and L. Hogue, “A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs,” in *Computer Communications*, 2007, vol. 30, no. 4.
- [124] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, “AbYSS: Adapting scatter search to multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 439–457, 2008.
- [125] D. Corne, N. Jerram, J. Knowles, M. Oates, and J. Martin, “PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’2001)*, 2001, pp. 283–290.
- [126] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the Pareto Archived Evolution Strategy,” *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.
- [127] E. Zitzler and K. Simon, “Indicator-Based Selection in Multiobjective Search,” in *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, 2004, pp. 832–842.
- [128] D. A. Van Veldhuizen and G. B. Lamont, “Multiobjective Evolutionary Algorithm Research : A History and Analysis,” pp. 1–88, 1998.
- [129] K. Bringmann and T. Friedrich, “Approximating the volume of unions and intersections of high-dimensional geometric objects,” *Comput. Geom.*, vol. 43, no. 6–7, pp. 601–610, Aug. 2010.
- [130] A. Nebro, J. Durillo, C. Coello Coello, F. Luna, and E. Alba, “A Study of Convergence Speed in Multi-objective Metaheuristics,” in *Parallel Problem Solving from Nature – PPSN X SE - 76*, vol. 5199, G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, Eds. Springer Berlin Heidelberg, 2008, pp. 763–772.
- [131] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, “A Review of Hybrid Evolutionary Multiple Criteria Decision Making Methods,” 2014.
- [132] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. Boston, MA: Springer US, 2007.
- [133] C. A. C. Coello, “A Survey of Constraint Handling Techniques used with Evolutionary Algorithms.”
- [134] X. Guo, Y. Wang, and X. Wang, “Using Objective Clustering for Solving Many-Objective Optimization Problems,” *Math. Probl. Eng.*, vol. 2013, pp. 1–12, 2013.

- [135] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2419–2426.
- [136] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.
- [137] A. López Jaimes, C. C. Coello, and J. Uriás Barrientos, "Online Objective Reduction to Deal with Many-Objective Problems," in *Evolutionary Multi-Criterion Optimization SE - 34*, vol. 5467, M. Ehrgott, C. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Springer Berlin Heidelberg, 2009, pp. 423–437.
- [138] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: theory and applications.," *Evol. Comput.*, vol. 17, no. 2, pp. 135–66, Jan. 2009.
- [139] K. Deb and K. Saxena, D., "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, 2006, pp. 3353–3360.
- [140] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08*, 2008, p. 673.
- [141] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [142] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [143] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Dec. 2007.
- [144] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," pp. 1027–1035, Jan. 2007.
- [145] J. C. Ros, "Introduction to Decision Deck-Diviz: Examples and User Guide," 2011.
- [146] S. Siraj, L. Mikhailov, and J. Keane, "PriEsT : A Tool to Estimate Priorities from Inconsistent Judgments," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, 2013, pp. 44 – 49.
- [147] K. Miettinen, "ND-NIMBUS for Demanding Interactive Multiobjective Optimization," *Mult. Criteria Decis. Mak.*, pp. 137–150, 2006.
- [148] R. P. Hämmäläinen and H. Lauri, "HIPRE 3+ User's Guide," 1995.
- [149] J. Mustajoki and R.P. Hämmäläinen:, "Web-HIPRE - A Java applet for AHP and value tree analysis," in *5th International Symposium on the Analytic Hierarchy Process*, 1999.
- [150] A. J. Nebro and J. J. Durillo, "jMetal 4.5 User Manual," 2014.
- [151] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software : An Update," *SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009.
- [152] S. M. Stigler, "Francis Galton's Account of the Invention of Correlation," *Stat. Sci.*, vol. 4, no. 2, pp. 73–79, May 1989.
- [153] J. C. Gower, "Euclidean distance geometry," *Math. Sci.*, vol. 7, no. 1, pp. 1–14, 1982.

- [154] S. C. Johnson, "Hierarchical clustering schemes.," *Psychometrika*, vol. 32, no. 3, pp. 241–54, Sep. 1967.
- [155] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," *SIAM Review*, vol. 25, no. 3. pp. 442–442, 1983.
- [156] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
- [157] J. Malenfant, M. Jacques, and F. Demers, "A Tutorial on Behavioral Reflection and its Implementation," 1996.