Collaborative optimisation in building design with a Pareto-based genetic algorithm

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/35017

Publisher: © Fan Yang

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
University Library

Author/Filing Title: YANG, F.

Class Mark: 1

Please note that fines are charged on ALL overdue items.
Collaborative Optimisation in Building Design with a Pareto based Genetic Algorithm

By

Fan Yang, BSc (Eng), MSc (Eng)

A Doctoral Thesis Submitted in Partial Fulfilment of the Requirements for the Award of Doctor of Philosophy of Loughborough University

May 2008

© Fan Yang, 2008
ABSTRACT

Large-scale building design is a constantly evolving discipline. Design managers are consistently trying to identify means for producing a 'better' product in a 'shorter' period of time. Hence there is a need for design assistant tools that can help designers understand the big picture. It is becoming hard to improve the system performance of building design based merely on advances in individual disciplines. In other words, improvements in individual disciplines alone are not sufficient to affect the improvements in the whole system. To achieve higher quality, system-orientated, holistic, multidisciplinary approaches to building design are needed (NSF, 1996). For this reason, this research investigates the applicability of multidisciplinary disciplinary optimisation (MDO) methodology in building design. The MDO methods divide a single system into a group of smaller sub-systems and effectively manage interactions between sub-systems. In the context of building design, the single system refers to the whole building design, and sub-system could be each disciplinary design. Such approaches could reduce the time and cost associated with the multidisciplinary design cycle.

This thesis describes the work of developing collaborative optimisation framework with a Pareto based genetic algorithm (COPGA). A conceptual COPGA framework is designed based on a thorough analysis of the nature, characteristics, and needs of building design, problems in application of MDO in engineering design and existing MDO formulation. A multi-objective collaborative optimisation framework is modified in order to enhance ability to solve multi-objective multidisciplinary building design problems. Finally a two-cycle COPGA framework is established.

This framework is implemented in two case studies. In the first study, a simple mathematical problem is used to test and verify the framework. The second study applies the framework in a building design scenario, which is used to evaluate the main features and performance of the framework. Results obtained from semi-structure interviews revealed that the COPGA framework is a good design decision support tool due to three obvious characteristics, namely a systematic design approach, a powerful design space search tool, and practical mechanism for multi-objective problems. Furthermore, this research not only contributes to the improvement on multidisciplinary building design, but also provides an effective approach for the multi-level MDO formulations.

Key words: Multidisciplinary, Multi-objective, building design, Pareto optimality, Genetic algorithm
ACKNOWLEDGMENTS

I am most grateful to my supervisor, Professor Bouchlaghem N.M., for his patient guidance, invaluable suggestions, and constant encouragement throughout this research project. Without his help in obtaining funding, this research and my PhD would not be possible.

I also wish to express my thanks to Professor John Miles at Cardiff University. During the process of my research, he has given me a lot of support.

Additionally I am very much indebted to my parents and my husband, for their continued love, support and patience.

Finally many colleagues and friends, either from China or the UK, have helped me through all these years. I would also like to express my appreciation to them.
TABLE OF CONTENTS

ABSTRACT i
ACKNOWLEDGMENTS ii
TABLE OF CONTENTS iii
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF ABBREVIATIONS xii
LIST OF NOMENCLATURES xiv

CHAPTER ONE: INTRODUCTION

1.1 Introduction 1
1.2 Background 1
1.3 Justification for the Research 4
1.4 Aim and Objectives 5
1.5 Summary of Methodology 5
1.6 Contribution to Knowledge 7
1.7 Limitations of the Research 8
1.8 Thesis Organisation 9

CHAPTER TWO: RESEARCH METHODOLOGY

2.1 Introduction 13
2.2 Research Strategy and Methodology 13
2.2.1 Overview 13
2.2.2 Type of Research Methods 15
2.2.3 Quantitative Research 15
2.2.4 Qualitative Research 17
2.2.5 Triangulation 20
2.3 The Methodology Adopted for the Research

2.3.1 Research Design

2.3.2 Building of Knowledge

2.3.3 COPGA Framework Development

2.3.4 Modelling for Building Design

2.3.5 Evaluation

2.4 Summary

CHAPTER THREE: BUILDING DESIGN OPTIMISATION

3.1 Introduction

3.2 Design Optimisation

3.3 Taxonomy of Optimisation Algorithms

3.3.1 Gradient-based Algorithm

3.3.2 Derivative-free Algorithm

3.3.3 Comparisons between Gradient-based and Derivative-free Algorithm

3.4 Characteristics of Building Design Optimisation

3.4.1 Multi-variable Design

3.4.2 Multi-type Variable

3.4.3 Multi-objective Design

3.4.4 Multi-disciplinary Design

3.4.4.1 Architecture

3.4.4.2 Structural Engineering

3.4.4.3 Building Services Engineering

3.5 The Needs to Improve Building Design Optimisation

3.5.1 Optimisation Formulations

3.5.2 Building Simulation Integration

3.5.3 Systematic Design

3.6 Industry Requirements

3.6.1 Interviewee’s Background

3.6.2 Findings from the Interviews

3.7 Summary
### CHAPTER FOUR: MULTIDISCIPLINARY DESIGN OPTIMISATION (MDO)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>79</td>
</tr>
<tr>
<td>4.2</td>
<td>Overview of MDO</td>
<td>79</td>
</tr>
<tr>
<td>4.3</td>
<td>Challenges to the Application of MDO in Engineering Design</td>
<td>82</td>
</tr>
<tr>
<td>4.3.1</td>
<td>MDO Formulation</td>
<td>82</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Computing Requirements</td>
<td>83</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Data Exchange</td>
<td>84</td>
</tr>
<tr>
<td>4.4</td>
<td>Classification of MDO Formulation</td>
<td>85</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Main Elements of MDO Problem</td>
<td>86</td>
</tr>
<tr>
<td>4.4.2</td>
<td>MDO Formulation</td>
<td>87</td>
</tr>
<tr>
<td>4.5</td>
<td>Review of Collaborative Optimisation and Analytic Target Cascading</td>
<td>93</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Collaborative Optimisation (CO)</td>
<td>94</td>
</tr>
<tr>
<td>4.5.1.1</td>
<td>CO Overview</td>
<td>94</td>
</tr>
<tr>
<td>4.5.1.2</td>
<td>CO Formulation</td>
<td>96</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Analytic Target Cascading Optimisation (ATC)</td>
<td>99</td>
</tr>
<tr>
<td>4.5.2.1</td>
<td>ATC Overview</td>
<td>99</td>
</tr>
<tr>
<td>4.5.2.2</td>
<td>ATC Formulation</td>
<td>101</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Discussion</td>
<td>105</td>
</tr>
<tr>
<td>4.5.3.1</td>
<td>Comparisons of CO and ATC</td>
<td>105</td>
</tr>
<tr>
<td>4.5.3.2</td>
<td>Selection of Formulation</td>
<td>107</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>109</td>
</tr>
</tbody>
</table>

### CHAPTER FIVE: COLLABORATIVE OPTIMISATION FRAMEWORK WITH A PARETO BASED GENETIC ALGORITHM (COPGA) DEVELOPMENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>110</td>
</tr>
<tr>
<td>5.2</td>
<td>Multi-objective Optimisation Solution</td>
<td>110</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Weighting Factor Method</td>
<td>111</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Pareto Optimality</td>
<td>112</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Optimisation Algorithm for This Study</td>
<td>113</td>
</tr>
<tr>
<td>5.3</td>
<td>Test Problem</td>
<td>115</td>
</tr>
<tr>
<td>5.4</td>
<td>Implementation by the MOCO Framework</td>
<td>116</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Formulations of the MOCO Framework</td>
<td>116</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Limitations with the MOCO Framework</td>
<td>118</td>
</tr>
<tr>
<td>5.5</td>
<td>Pareto GA Based CO Framework Development</td>
<td>121</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Reasons for Use of the Pareto-based GA</td>
<td>121</td>
</tr>
<tr>
<td>5.5.2</td>
<td>The Conceptual COPGA Framework</td>
<td>122</td>
</tr>
<tr>
<td>5.5.2.1</td>
<td>Internal Cycle of the COPGA Framework</td>
<td>123</td>
</tr>
<tr>
<td>5.5.2.2</td>
<td>External Cycle of the COPGA Framework</td>
<td>125</td>
</tr>
<tr>
<td>5.6</td>
<td>Pilot Study</td>
<td>128</td>
</tr>
<tr>
<td>5.6.1</td>
<td>The COPGA Formulation</td>
<td>128</td>
</tr>
<tr>
<td>5.6.2</td>
<td>The COPGA Results</td>
<td>132</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Validation of the COPGA Formulation</td>
<td>134</td>
</tr>
<tr>
<td>5.6.3.1</td>
<td>All-at-Once (AAO) Formulation</td>
<td>135</td>
</tr>
<tr>
<td>5.6.3.2</td>
<td>Comparisons between COPGA Results and AAO Results</td>
<td>135</td>
</tr>
<tr>
<td>5.7</td>
<td>Suggestions for a Future Case Study</td>
<td>137</td>
</tr>
<tr>
<td>5.8</td>
<td>Summary</td>
<td>138</td>
</tr>
</tbody>
</table>

CHAPTER SIX: DESIGN SCENARIO

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>140</td>
</tr>
<tr>
<td>6.2</td>
<td>Design Scenario</td>
<td>140</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Design Structure Matrix</td>
<td>141</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Summary of all Information in Design Scenario</td>
<td>145</td>
</tr>
<tr>
<td>6.3</td>
<td>Analysis Model</td>
<td>148</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Structural Analysis</td>
<td>148</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Validation of Structural Analysis</td>
<td>150</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Thermal Analysis</td>
<td>152</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Validation of Thermal Analysis</td>
<td>153</td>
</tr>
<tr>
<td>6.4</td>
<td>Analysis Formulation</td>
<td>157</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Structural Analysis Formulation</td>
<td>157</td>
</tr>
<tr>
<td>6.4.1.1</td>
<td>Beam Design</td>
<td>158</td>
</tr>
</tbody>
</table>
7.5 Discussions
7.5.1 Results
7.5.2 Appropriateness of the Evaluation Approach
7.6 Summary

CHAPTER EIGHT: CONCLUSIONS AND RECOMMENDATIONS

8.1 Introduction
8.2 Summary
8.3 Conclusions
8.4 Limitations of the Research
8.5 Recommendations for Future Work
  8.5.1 Further Development of the COPGA Framework
  8.5.2 Recommendations in Community of MDO Applications
  8.5.3 Integrating MDO Method with Agent Technology
8.6 Closing Remarks

REFERENCES

APPENDICES

| Appendix One: | List of Publications Arising from the Research |
| Appendix Two: | Template for Industry Semi-Structured Interviews |
| Appendix Three: | The COPGA Results of the Mathematical Problem |
| Appendix Four: | The COPGA Results of Design Scenario |
| Appendix Five: | Matlab Program Code |
| Appendix Six: | Evaluation Questionnaire |
LIST OF FIGURES

Figure 1.1 Research Structure 10
Figure 2.1 Nested Approach of Research Methodology 15
Figure 2.2 Triangulation of Quantitative and Qualitative Data 21
Figure 2.3 The Research Process 23
Figure 2.4 Major Keywords Used in Searching the Relevant Literature 27
Figure 2.5 Process of Framework Development 28
Figure 2.6 Modelling Process and Research Method Adopted 32
Figure 3.1 Optimisation vs. Decision-based Design 41
Figure 3.2 Process of Simulated Annealing Algorithm 49
Figure 3.3 Process of Genetic Algorithm 50
Figure 3.4 Value of Wind Bracing 60
Figure 3.5 A Monolithic Framework for BSI 66
Figure 3.6 A Loose Integrated Framework for BSI 67
Figure 4.1 Two-disciplinary Coupled System 86
Figure 4.2 Iteration Loops in a Series of Analyses 94
Figure 4.3 Collaborative Optimisation Architecture 96
Figure 4.4 Schematic of Analytical Target Cascading Process for a Top-down and bottom-up, Level-by-level Solution Sequence 100
Figure 4.5 Links between Decision and Analysis Model in ATC Framework 100
Figure 4.6 Information Flow Up and Down in the ATC Hierarchy 103
Figure 4.7 The Process of CO Formulation 106
Figure 4.8 The Process of ATC Formulation 106
Figure 5.1 Pareto Rank 114
Figure 5.2 Crowding Distance 115
Figure 5.3 Formulation of the Internal Cycle 123
Figure 5.4 The Mathematical Problem in the COPGA Framework 131
Figure 5.5 The COPGA Solutions of the Mathematical Test Problem 132
Figure 5.6 System-level Co-ordination of Variable of $x_i$ 133
Figure 5.7 System-level Co-ordination of Variable of $x_i$ 134
Figure 5.8 AAO Formulation of the Mathematical Test Problem 135
Figure 5.9 Comparison between the COPGA and AAO Results 137
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 6.1</td>
<td>Office Floor Plan</td>
<td>140</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>Three Configurations that Characterise a System in DSM Analysis</td>
<td>141</td>
</tr>
<tr>
<td>Figure 6.3</td>
<td>Definition of a Module Placed on N-square Diagram</td>
<td>144</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>Interdisciplinary Variables between Disciplines</td>
<td>144</td>
</tr>
<tr>
<td>Figure 6.5</td>
<td>Separation of Structure and Services Zone</td>
<td>148</td>
</tr>
<tr>
<td>Figure 6.6</td>
<td>Two-span-two-story Structural Problem</td>
<td>151</td>
</tr>
<tr>
<td>Figure 6.7</td>
<td>Dimensions of Cross-section</td>
<td>158</td>
</tr>
<tr>
<td>Figure 6.8</td>
<td>Design Scenario in the COPGA Framework</td>
<td>172</td>
</tr>
<tr>
<td>Figure 6.9</td>
<td>COPGA Solutions of the Design Scenario</td>
<td>174</td>
</tr>
<tr>
<td>Figure 6.10</td>
<td>Trends of the Sixty Solutions of Design Scenario</td>
<td>176</td>
</tr>
<tr>
<td>Figure 6.11</td>
<td>System-level Co-ordination of Value of Window's Height</td>
<td>177</td>
</tr>
<tr>
<td>Figure 6.12</td>
<td>Progresses in System-level Objective Function</td>
<td>179</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>A General Prototyping Framework and Evaluation Approach</td>
<td>182</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Five Ordinal Measures of Agreement on Likert scale</td>
<td>187</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1.1  Relationships between Research Objectives and Methods 6
Table 2.1  Type of Research and Corresponding Example 14
Table 2.2  Distinguishing Characteristics of Quantitative and Qualitative Methods 22
Table 2.3  Overview of the Research Process 25
Table 3.1  Comparison of Properties between Gradient-based and Derivative-free Algorithm 52
Table 3.2  The Interviewee’s Background 71
Table 3.3  Response to Characteristics of Building Design 72
Table 3.4  Response to the Conventional Building Design practices 73
Table 3.5  Response to Approaches to Coordinate Conflicts between Disciplines 74
Table 3.6  Response to Properties Expected in the Multidisciplinary Design Model 76
Table 4.1  Building Multidisciplinary Optimisation Problems: Example of Variables and Design Functions 87
Table 4.2  Summaries of MDO Formulations 90
Table 5.1  Five Representative Solutions of the Mathematical Test Problem 133
Table 5.2  Comparison between the COPGA and AAO Formulations 136
Table 6.1  Application Parameter-based DSM in Design Scenario 143
Table 6.2  Objective, Variable and Constraints Information 146
Table 6.3  Constructional and Occupancy Details 146
Table 6.4  Summary of All Kinds of Variables in the Design Scenario 147
Table 6.5  Comparison of Results of Structural Analysis 152
Table 6.6  U and Y Value in the Thermal Analysis Example 154
Table 6.7  Comparison of Results of Thermal Analysis 157
Table 6.8  Data Detailed in Design Scenario 163
Table 6.9  Surface Areas, U-value and Y-Value in the Design Scenario 163
Table 6.10 Six Representative Solutions of Design Scenario 175
Table 7.1  Response to Features of the COPGA Framework 192
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>T Value of Satisfaction with the COPGA Framework between Architect and Structural Engineer</td>
<td>193</td>
</tr>
<tr>
<td>7.3</td>
<td>T Value of Satisfaction with the COPGA Framework between Structural and HVAC Engineer</td>
<td>194</td>
</tr>
<tr>
<td>7.4</td>
<td>Summaries of Comments from Interviewees Regarding the COPGA Prototype</td>
<td>201</td>
</tr>
</tbody>
</table>
LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAO</td>
<td>All-At-Once</td>
</tr>
<tr>
<td>AIAA</td>
<td>American Institute of Aeronautics and Astronautics</td>
</tr>
<tr>
<td>ATC</td>
<td>Analytical Target Cascading</td>
</tr>
<tr>
<td>BDO</td>
<td>Building Design Optimisation</td>
</tr>
<tr>
<td>BLISS</td>
<td>Bi-Level Integrated System Synthesis</td>
</tr>
<tr>
<td>CIBSE</td>
<td>Chartered Institution of Building Services Engineers</td>
</tr>
<tr>
<td>CO</td>
<td>Collaborative Optimisation</td>
</tr>
<tr>
<td>COPGA</td>
<td>Collaborative Optimisation Framework with a Pareto based Genetic Algorithm</td>
</tr>
<tr>
<td>CSCW</td>
<td>Computer Support Cooperative Work</td>
</tr>
<tr>
<td>CSSO</td>
<td>Concurrent Subspace Optimisation</td>
</tr>
<tr>
<td>DSM</td>
<td>Design Structure Matrix</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilating, Air Conditioning</td>
</tr>
<tr>
<td>IDF</td>
<td>Individual Disciplinary Feasible</td>
</tr>
<tr>
<td>IFC</td>
<td>Industry Foundation Classes</td>
</tr>
<tr>
<td>MDF</td>
<td>Multidisciplinary Feasible</td>
</tr>
<tr>
<td>MDO</td>
<td>Multi-disciplinary optimisation</td>
</tr>
<tr>
<td>MOCO</td>
<td>Multi-objective Collaborative Optimisation</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SQP</td>
<td>Sequential Quadratic Programming</td>
</tr>
</tbody>
</table>
LIST OF NOMENCLATURES

The COPGA Framework

- $d_i$: $i^{th}$ subsystem-level objective;

- $X^0$: Vector of system level-variable (i.e. shared and coupling variables), namely interdisciplinary variable;

- $X_{sh}$: Vector of shared variable;

- $Y_{ij}$: Vector of coupling variable, namely $i^{th}$ subsystem send value to $j^{th}$ subsystem;

- $X_{LB}, X_{UB}^0$: Vector of system-level variable's lower and upper bound;

- $X_{LB}^i, X_{UB}^i$: Vector of $i^{th}$ subsystem-level variable's lower and upper bound;

- $g_i$: Vector of $i^{th}$ subsystem constraints;

- $X^i$: Vector of subsystem $i^{th}$ variables;

- $X_{local}^i$: Vector of subsystem $i^{th}$ local variables;

- $(\star)^0$: Variable in system level

- $(\star)^i$: Interdisciplinary variable in the corresponding $i^{th}$ subsystem level;

- $(\star)^*$: Optimal value of variable;

- $(\odot)^i$: Target value of interdisciplinary variable sent from system level to sub-system level

Structural Analysis:

- $A$: Area;

- $A_{eff}$: Effective cross-sectional area;

- $D$: Depth of section;

- $d$: Depth of web;

- $E$: Modulus of elasticity of steel;

- $F_c$: Compressive axial force;

- $I$: Second moment of area;

- $K$: Stiffness matrix;

- $l$: Length of structural element;

- $M$: Bending moment;

- $M_c$: Bending moment capacity;
LIST OF NOMENCLATURES (continued)

$P_c$  Compression resistance;

$p_b$  Bending strength (lateral-torsional buckling);

$p_e$  Compressive strength;

$p_y$  Design strength of steel;

$r_x$  Radius of gyration about the major axis;

$r_y$  Radius of gyration about the minor axis;

$T$  Thickness of flange or transferred matrix from the local coordinates into the global coordinates;

$t$  Thickness of web;

$S_x$  Plastic modulus about the major axis;

$Z_x$  Section modulus about the major axis;

$\lambda$  Slenderness, i.e. the effective length divided by the radius of gyration;

$\lambda_{LT}$  Equivalent slenderness (lateral-torsional buckling);

$\Delta$  Displacement

**Thermal Analysis:**

$C_v$  Ventilation conductance (W/K);

$f$  Decrement factor;

$F_{ea}$  Room conduction factor with respect to dry resultant temperature;

$F_{oe}$  Room admittance factor with respect to dry resultant temperature;

$f_r$  Thermal response factor;

$F_1, F_2$  Factors related to characteristics of heat source with respect to dry resultant temperature;

$N$  Number of air change per hour (h$^{-1}$);

$Q_{av}$  Heat gain from ventilation (W);

$Q_{con}$  Convective component of the internal gain (W);

$Q_f$  Fabric heat gain (W);

$Q_{fa}$  Fabric gain to the air node (W);
LIST OF NOMENCLATURES (continued)

$Q_k$  Total sensible cooling load to the air node (W);

$Q_{rd}$  Radiant component of the internal gain (W);

$Q_{sg}$  Solar gain through glazing (W);

$Q_l$  Total gain loss (W);

$t_{oo}$  Outside air temperature ($^\circ$C);

$t_e$  Dry resultant temperature at centre of room ($^\circ$C);

$t_{eo}$  Sol-air temperature ($^\circ$C);

$U$  Thermal transmittance of material (W/m$^2$ K);

$V$  Room volume (m$^3$);

$Y$  Thermal admittance (W/m$^2$ K);

$\theta$  Time (h);

$\sum A$  Sum of room surface area (m$^2$);

$\sum AU$  Sum of the product of surface area and corresponding thermal transmittance over surfaces through which heat flow occurs (W/K);

$\sum AY$  Sum of the product of surface area and corresponding thermal admittance over surfaces through which heat flow occurs (W/K);

$\phi$  Time lag associated with decrement factor (h);

(*)  Mean gains;

(•)  Cycle gains
CHAPTER ONE: INTRODUCTION

1.1 Introduction
This chapter presents a general overview of the thesis, consisting of a brief introduction and description of the subject matter of the research as well as the specific problems being studied. It also sets out aim and objectives, the methodologies which the research is carried out, a summary of its achievements and the structure of the thesis.

1.2 Background
The construction industry is regarded as one of the largest economic sectors in the world, typically representing 10-25% of the gross national product (GNP) of a nation (Veeramani et al., 1998). In the UK, output from the construction industry is about £58 billion in 1998, which is equivalent to around 10% of GNP (Egan, 1998). In the USA, output from the construction industry is around $850 billion per year (equivalent to 13% of GNP) and employs 10 million people (Kalay, 1999). The construction industry has a reputation for low productivity, waste, low technology, and poor quality (Egan, 1998). Over the last decade, some new paradigms and advanced information & communication technologies have been adopted for building design, construction process planning, process execution and control, and project management to enable reductions in cost and lead-time of construction projects. This thesis mainly focuses on improving building design with optimisation paradigm and collaborative working.

Building design is a complex, multidisciplinary engineering activity that requires making difficult compromises to achieve a balance between competing objectives, including safety, reliability, performance, and cost (Grierson and Khajehpour, 2002; Sisk et al., 2003). In such a design, clients are in position to initiate the project, employ professional teams, and find sufficient resources; they also describe their expectation with respect to functions, attributes or other special feature of the building that satisfies the business needs. Consequently these teams, including architects, structural engineers, building services engineers, quantity surveyors, contractors, material suppliers, etc — work together for a relatively short period on the design and construction of a building and satisfy client’s needs.
To support designers in making decisions that can ultimately determine the success or failure of the end product, optimisation with detailed analysis/simulation tools is sometimes needed. Due to often compressed design schedules, these analysis tools could be capable of a rapid turn-around analyses without compromising accuracy. Optimisation can also enable designers to explore the design space efficiently to help them make 'smart' decisions quickly. Within a optimisation framework, a design is cast in an objective-oriented decision-making model. Objectives are defined by desired building performances or economic requirements. In addition, a causal relationship is presumed between design decision variables and its performance. Due to these presumptions of causality which implies that design decisions are essentially driven by performance (within a given context), design optimisation is a natural formalisation for performance-based building design. It improves or fine-tunes the design in terms of one or more performance aspects by systematically searching for design variables that meet these stated objectives. Within the past fifty years, different optimisation algorithms have been applied to numerous complex problems including architectural design (Szykman and Cagan, 1997; Yin and Cagan, 2000; Michalek et al., 2002), structural design (Balling and Yao, 1997, Isenberg et al., 2002, Middendorf, 2003), thermal and HVAC design (D'Cruz and Radford, 1987, Bouchlaghem and Letherman, 1990; Wright et al., 2002). In spite of benefits obtained from these applications, it has been found that optimised one-discipline designs do not always fit together to produce the best overall design.

On the other hand, the fragmentation of knowledge in the building industry has created a symmetry of ignorance (Kalay, 1999), where no single professional has all the knowledge needed to design a complex facility, and where it is no longer possible to design a building without consulting many specialists (Cuff, 1991). The collaboration between these different disciplines is essential for the success of a building design (Cheng, 2003). To provide supportive environments for collaboration, design systems must provide participants with facilities for information sharing, task coordination, and conflict resolution (Wang et al., 2002).

With the increasing capabilities of the computer as a communication device, collaborative systems with shared information have made great advances. Most of these systems can provide access to catalogues of design information and facilitate
Chapter 1

Introduction

communication between multidisciplinary design team members in multimedia formats (Fruchter et al., 1995; Fruchter, 1996).

However coordination of conflicts in the design process is critical for successful collaborative design (Klein, 1992). These conflicts could stem from different disciplinary terminologies (Wang et al., 2002). Faced with this problem, the Industry Foundation Classes (IFC) developed schemas that define a standard file format that can be used as a mechanism for sharing building information between CAD systems and the ever-expanding range of design analysis tools (IAI, 2007), while the Semantic Web provides a framework for sharing definitions of terms, resources and relationships between disciplines (Aziz et al., 2004). These technologies have been widely applied in the field of building design in order to resolve semantic conflicts (Pan, 2006; Plume and Mitchell, 2007).

Furthermore, design conflicts can be related to the requirements and dependencies between discipline-specific design tasks. For example, the architect wants maximum flexibility of floor space usage and high comfort level while the structural engineer desires the most economical and safe structure. It is apparent that optimum floor flexibility may conflict with the lightest structure, as column and girder layouts that achieve a least-weight structure may limit the floor space usage. Traditionally, a complex activity such as this was facilitated mostly through negotiations and face-to-face meetings. Few mechanisms that automate this negotiation process in the computer environment should be developed (NSF, 1996).

In fact, researchers in other engineering industries (e.g. aerospace and automotive) have invested a lot of effort on exploring new methods of solving the above two problems: system optimisation and automatic coordination of multidisciplinary design. The American Institute of Aeronautics and Astronautics (AIAA) Committee in 1991 named these methods that facilitate multidisciplinary designs as Multidisciplinary Design Optimisation (MDO). So far these MDO methods have been applied in launch-vehicle design (Braun et al., 1997), trajectory optimisation (Huque and Jahingir, 2002), aeroelastic design (Rodriguez et al., 1998), supersonic aircraft optimisation (Jun et al., 2004), conceptual bridge design (Balling and Rawlings, 2000), undersea vehicles design (Belegundu et al., 2000), race car design (McAllister
et al., 2005), but the studies of MDO in building design are still in the early stages. Therefore the main aim of this research is to formulate, demonstrate, and evaluate an appropriate MDO method for building design optimisation.

1.3 Justifications for the Research.

The current loosely-coupled design environments, where the disciplinary experts use different methods and models with different objectives, use separate computing platforms, and are sometimes remotely located from each other, create difficulties for a structured system design. Thus, to aid engineers in making decisions that can ultimately determine the success of the end product, it is important to explore the application of MDO in building design. The reasons are:

- A number of decision support systems based on optimisation techniques have been developed for the building design, these optimisation techniques are able to explore a set of alternative design solutions, and select the preferred option. This process of performing optimisation matches a decision-making process in engineering design (Chen et al., 1998; Azarm and Narayanan, 2000).

- Optimisation techniques are usually integrated with analysis software in order to speed up the design process, increase efficiency and enable the comparison of a broader range of design options, leading to a more optimal design (Augenbroe, 2002). Hence it can be argued that designs obtained using optimisation techniques are more reliable than designs based on designers' experience.

- Past work on building design optimisation is limited to discipline-based research, while building design is multidisciplinary; therefore it is important to improve synchronous collaborative design, asynchronous collaborative design, and design coordination (Gross et al., 1998).

- The standard practice currently used for coordinating conflicts among multidisciplinary designs is face-to-face meetings and a final resolution decided by the chief designer, which is a time-consuming process. With the short time frame allowed for design, designers need to obtain compatible solutions rapidly and reliably.

- For large and complex engineering design, some form of problem decomposition becomes necessary (Papalambros, 2002). Decomposition in line
with discipline is beneficial because it allows the specialised analysis and decision-making focused on individual design tasks.

1.4 Aim and Objectives
The process of building design is complex, requiring skills and knowledge from all design disciplines involved. For this reason, there is a need for the different specialists to work together as an integrated team. The aim of this research is to develop and implement a framework for the application of multidisciplinary design optimisation techniques to building design. To achieve this goal, the following specific objectives are pursued:

- To investigate the existing applications of optimisation algorithms in building design and define the characteristics of building design optimisation and challenges (Objective One);
- To review state-of-the-art MDO applications in other industry sectors (e.g. aircraft design, automotive design, etc.) and to identify the major issues that inhibit using MDO in engineering design (Objective Two);
- To develop and test Collaborative Optimisation framework with a Pareto based Genetic Algorithm (COPGS) for building design (Objective Three);
- To implement the COPGS framework within the building design context (Objective Four); and
- To evaluate the COPGS framework by the way of expert assessment (Objective Five).

1.5 Summary of Methodology
To achieve the research objectives, a combination of research methods was adopted. These included literature review, expert interview, scenario, simulation, rapid prototyping, and evaluation. Table 1.1 illustrates the relationship between the research objectives and the research methods adopted to achieve them.
<table>
<thead>
<tr>
<th>Objectives</th>
<th>Literature Review</th>
<th>Expert Interview</th>
<th>Mathematical Problem Testing</th>
<th>Modelling</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective One</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objective Two</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objective Three</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Objective Four</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Objective Five</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The following section briefly summarise the research methods used. Chapter 2 provides full details of the research methods adopted, including justification for their use.

**Literature review:** The research started with an extensive literature review which focused on two major subjects. Firstly, a structured review was undertaken in the areas of characteristics of building design optimisation, problems with multidisciplinary and multi-objective building design. Secondly, a review of multidisciplinary design optimisation (MDO) was carried out, including challenges on the application of MDO in engineering problem, main elements of MDO problems, and MDO formulation. Relevant sources were identified such as referred journals, books, the Internet, government reports, conference and workshop proceedings, and doctoral dissertations. The literature review was aimed at identifying the problems associated with building design optimisation that can be improved and to examine the applicability of the MDO formulation. The review was an ongoing process, carried out simultaneously with other stages of the research project.

**Expert interview:** The iterative development of the COPGS framework resulted in a series of papers and reports at different development stages of the project. They were reviewed critically. Moreover during the process of developing the COPGS framework, discussions with experts in construction and other industries were frequently undertaken, which affirmed needs of industry participants, and the author's
understandings of MDO and the consideration of some important issues when implementing the COPGS framework.

**Modelling for building design:** A model was developed based on the COPGS framework within the Matlab environment. There were four main steps in developing the prototype:

- **Establish design scenario:** The literature review findings were used to establish a design scenario that demonstrated the obvious characteristics of building design optimisation, namely multi-objective and multidisciplinary;
- **Simulate discipline-specific analysis:** Structural and HVAC analyses were required when applying the COPGS framework for the above design scenario; hence the structural and thermal programs were coded based on the matrix displacement method, and the steady-state and dynamic method guided by CIBSE respectively;
- **Use the prototype:** The prototype system was demonstrated so that designers can evaluate its performance and ensure that it meets an acceptable level of accuracy and efficiency; and
- **Revise and enhance the prototype:** The suggestions made by the designers and researchers were used to refine the prototype.

**Evaluation:** Semi-structured interviews were adopted to evaluate the COPGS framework, which involved two architects, four structural designers, two HVAC designers, one academic researcher and one professional software developer. The evaluation interviews were undertaken consisted of three main elements: a presentation on the background to the COPGS framework, a demonstration, the completion of an evaluation questionnaire, and discussions on key issues relating to the framework. The relevant comments and suggestions were used to refine and improve the framework.

**1.6 Contribution to Knowledge**

This study presents a two-level non-hierarchical design optimisation framework that has been extended to integrate interrelated building performance areas. The specific contribution of this research can be summarised as:
First, this research makes a thorough review of the nature and characteristics of building design optimisation in the published literature. Problems underlying building design optimisation are discussed; and possible solutions are suggested.

Second, this research examines the properties of multi-level MDO formulation based on the requirements of a building design optimisation problem.

Third, the COPGS framework provides an explicit mechanism for achieving consistent and collaborative design solutions in scenarios that require coordination between different performance requirements.

Fourth, the prototype of framework for multi-objective and multi-disciplinary building design optimisation has been developed.

Fifth, this study also provides a new approach to invoking and coordinating multiple analyses in the decision-making process.

1.7 Limitations of the Research
This research advocates MDO as a system approach to improve collaborative building design in two studies. The first study verified the collaborative optimisation framework using a simple mathematical problem. The second study used a realistic design scenario that involved structural and building services design. These studies reveal both the benefits and challenges associated with formalising a collaborative design scenario.

Experiences gained from both studies are used to derive a generalised framework for building design for a multi-disciplinary and multi-objective problem, these studies also help identify future work necessary to apply such a framework in practice. In addition they investigate dependencies between structural and building service design based on the corresponding analysis software, which can be coordinated within the COPGS process, thereby offering a new solution to integrating building analysis.

Work presented in this thesis studied the applicability of MDO framework for a trade-off building design approach. However, there are some limitations to this research owing to time and technique constraints.
• Firstly, some aspects of building design are essentially qualitative, such as aesthetics. These aspects cannot be modelled using numerical optimisation and are not considered in this study. Hence this study only focuses on those quantitative performance areas.

• Secondly, the COPGS framework is applied to test problems with a sufficiently small number of variables to make it possible for the results to be interpreted.

• Thirdly, one major criterion in selecting an optimisation algorithm is the type of design variables (e.g. continuous and discrete); this study handles all variables as continuous and takes approximate values with regard to discrete variables.

• Finally, the implementation of the COPGS framework is computationally intensive due to the large number of iterations required to obtain reliable and compatible design solutions.

The COPGS framework remains in a proof-of-concept prototype development stage, further developments are still required.

1.8 Thesis Organisation
Figure 1.1 shows the overall research process carried out to achieve the specific objectives of the research. The thesis is structured into eight chapters, a brief description of each chapter is given below:
Chapter One: Introduction

This chapter provides an introduction to the research project undertaken. It briefly describes the research background; justifies the need for research; outlines the associated aim and objectives; presents a summary of methodologies used and limitations of the research; and illustrates the contributions of the research.

Chapter Two: Research Methodology

This chapter reviews the relevant research methodologies, discusses the methodological consideration for this study, and justifies the adopted research methods.
Chapter Three: Building Design Optimisation (BDO)
The literature review on building design optimisation is the focus of Chapter Three. It also discusses the optimisation algorithms, followed by a discussion of the characteristics of building design optimisation. The chapter also summarises challenges on current building design optimisation and requirements from industry participants.

Chapter Four: Multidisciplinary Design Optimisation (MDO)
This chapter reviews the definition and classification of MDO formulation and discusses issues that influence the application of MDO in engineering designs. It then describes in detail both Collaborative Optimisation (CO) and Analytic Target Cascading (ATC) with regard to their major features and mathematical formulations. Finally the most suitable MDO formulation is selected based on requirements of building design optimisation and comparisons between CO and ATC.

Chapter Five: Pareto Genetic Algorithm based on Collaborative Optimisation (COPGS) Framework Development
This chapter analyses the existing multi-objective optimisation solutions and selects Pareto optimality for the proposed framework. It also demonstrates the developed COPGS framework that reflects the specific context of BDO. This framework is tested using a mathematical problem. Finally, the COPGS framework is validated through comparisons of results generated by both the COPGS and All-at-Once formulations.

Chapter Six: Design Scenario
A design scenario presenting the distinguishing features of the COPGS framework within the building design optimisation context is developed in this chapter. At the same time, the structural and HVAC analyses are presented and validated. The COPGS framework is applied on the design scenario. Finally the results obtained from the COPGS framework are analysed in term of coordination process among disciplines and the sensitivity of objective function to design variables.
Chapter Seven: Evaluation

This chapter describes the system evaluation process. It starts with an introduction to the evaluation aim and objectives, followed by a description of the evaluation method. The evaluation results are discussed, and further used to improve the developed prototype system. The benefits and limitations of the system are also presented.

Chapter Eight: Conclusions and Further Study

This chapter presents the summary and conclusions of the thesis. It covers the major findings, the conclusions of the study, and the limitations of the study; and provides recommendation for both industry practitioners and future research.
CHAPTER TWO: RESEARCH METHODOLOGY

2.1 Introduction
This chapter provides an introduction to general research methodologies and presents an overview of the methodology used for this research. The first part of the chapter examines the philosophical perspective and underlying principle of research process. It also describes different research methods to justify the choices that have been made in the selection of an appropriate research strategy. The chapter ends with a description of the research methods adopted and demonstrates how they achieve the research objectives. The whole research process presents the 'Building-Testing-Refinement' cycle, which is adopted to develop the proposed framework.

2.2 Research Strategy and Methodology

2.2.1 Overview
The Concise Oxford Dictionary (2004) defines 'research' as 'careful search or inquiry' endeavour to discover new or collate old facts etc, by scientific study of the subject, or by course of critical investigation'.

According to Love et al. (2002), two research philosophies appear to dominate the study of construction: the Interpretivist (otherwise known as phenomenological) approach and the positivist approach. Interpretivists argue knowledge development and theory building through developing ideas induced from observed and interpreted social construction (qualitative approach). Blumberg et al., (2005) promote the positivists view that knowledge is developed by investigating social reality through observing objective facts (quantitative approach). These principles include various types of research; its context; the effects of knowledge, experience and bias; and the meaning of generalisation and particularisation in a research context. Table 2.1 lists a few types of research and corresponding example.
Table 2.1 Type of Research and Corresponding Examples (Phillips and Pugh, 2000)

<table>
<thead>
<tr>
<th>Type of Research</th>
<th>Scope of Research</th>
<th>Examples in this Kind of Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploratory</td>
<td>The research work needs to examine what theories and concepts are appropriate to developing new ones if it is necessary, and whether any existing methodology can be adopted.</td>
<td>Correlation-prediction: statistically significant correlation coefficients between and among a number of factors are sought and interpreted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Theory construction: an attempt to find or describe principles that explain how things work the way they do.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trend analysis: prediction or forecasting of the future direction of events</td>
</tr>
<tr>
<td>Testing-out</td>
<td>This research tries to find the limitations of previously proposed generalisations.</td>
<td>Analysis: classes of data are collected and studies conducted to discern patterns and formulate principles that might guide future action.</td>
</tr>
<tr>
<td>research</td>
<td></td>
<td>Comparison: two or more existing situations are studied to determine their similarities and differences.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Evaluation: research to determine whether a program or project followed the prescribed procedure and achieved the stated outcomes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Experiment: one or more variables are manipulated and the results analysed.</td>
</tr>
<tr>
<td>Problem solving</td>
<td>This research starts from a particular problem in the real world, and brings together all the intellectual resources that can be brought to bear on its solution. The problem has to be defined and the method of solution has to be discovered.</td>
<td>Case study: the background, development, current conditionals and environmental interaction of one or more individual, groups, communities, business or institution are observed, recorded and analysed of stages of patterns in relation to internal and external influences.</td>
</tr>
<tr>
<td>research</td>
<td></td>
<td>Design-demonstration: new systems or programs are constructed, tested and evaluated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Survey-questionnaire: behaviours, beliefs and observations of specific groups are identified, reported and interpreted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Status: a representative or selected sample of one or more phenomena is examined to determine its special characteristics.</td>
</tr>
</tbody>
</table>


2.2.2 Type of Research Methods

In general, there are three types of research methodologies: quantitative, qualitative and a combination of both methods called triangulation or mixed method (Punch, 1999; Fellows and Liu, 2003; Neuman, 2006). Research is a systematic investigation to find an answer to a problem (Blaxter et al., 2006) while research methodology refers to the principles and procedures of logical thought processes which are applied to this investigation (Fellow and Liu, 2003; Klien and Myers, 1999). Kagioglou et al. (2000) introduce a nested approach to describe a hierarchical model of research methods that is divided into three main interrelated themes: research philosophy, research approaches and research technique, as show in Figure 2.1. The research philosophy (e.g. positivist and interpretivist) found in the outer ring guides the inner research approach and research technique. The research approach consists of domain theory generation and testing methods, such as quantitative and qualitative. Research technique (e.g. questionnaire and interview) comprises data collection tools.

![Figure 2.1 Nested Approach of Research Methodology (Kagioglou et al., 2000)](image)

Prior to discussing the methodology adopted in this study, the following sections review the characteristics, advantages and disadvantages of these research methods.

2.2.3 Quantitative Research

The quantitative research method is regarded as the specific and positivist research methods. It is about gathering factual data and studying relationships between facts in order to find out how such facts and relationships accord with the theories of previous research (Fellows and Liu, 2003), and determines whether a hypothesis holds true (Creswell, 1994). In addition, quantitative methods can also be employed to establish general laws or principles (Burns, 2000). It often involves the collection of a large amount of data sets if compared with the qualitative approach (O’Leary, 2004).
main approaches are used in data collection for quantitative research (Fellow and Liu, 2003), namely:

- Asking questions of respondents by questionnaires and interviews;
- Carrying out experiments; and
- ‘Desk Research’ using data collected by others.

Here two commonly used quantitative research methods are explained:

a) Surveys
This is a research method in which the research systematically asks a large number of people the same question and then records their answers (Neuman, 2006). It is appropriate for analysis of groups’ interactions; the collection of original data for describing a population too large to observe directly; investigating attitudes and orientation in a large population; and describing the characteristics of a large population.

There are two main types of data collection methods in survey research, which includes: face-to-face or telephone interviews and the questionnaire survey. The advantage of this survey lies in gathering data from a relatively large number of respondents within a limited time frame. It is thus concerned with a generalised result when data is abstracted from a particular sample or population (Naoum, 1998). The disadvantage is that little insight is usually obtained regarding the causes or the processes behind the phenomenon being studied. Also, survey studies are subject to some well-known biases. For example, respondents may change their answers either consciously or unconsciously, to show themselves in a better light or to confirm to the expectation of those who are studying them.

b) Experimental Research
Experimental research is best suited to known problems or issues where the variables involved are identified, or are, at least, hypothesised with some confidence (Fellows and Liu, 2003). Hence, it can be thought of as systematic trial and observation trial because the answer is not known beforehand, observation because the result must be carefully recorded, and systematic because all good research is planned and purposeful (Melville and Goddard, 1996).
According to Fellows and Liu (2003), there are two approaches to experimental research - laboratory experiments and field experiments. Laboratory experiments are usually carried out to test relationship between identified variables, by holding all except one variable constant and then testing the effect on dependent variables by changing one independent variable. This is done with a view of making generalisable statements applicable to real world situation. Field experiments are not conducted in specially built laboratories but in dynamic social, industrial, economic and political areas (Gallier, 1992). The key strength of experimental research is its control and logical rigour in establishing evidence for causality. In general, experiments tend to be easier to replicate, less expensive and less time consuming than the other techniques (Neuman, 2006), but it is extremely difficult in a study involving human individuals (Alasuutari, 1998).

2.2.4 Qualitative Research

The qualitative research method is regarded as a naturalist, subjectivist or interpretivist research method and tends to focus on exploring in much detail a smaller number of instances which are seen as being interesting or illuminating (Blaxter et al., 2001). Its data sets are relatively small scale (O'Leary, 2004) and chiefly non-numeric, such as in the form of text and image (Punch, 1998). This is because it aims to investigate and gain insight into the beliefs, understandings, views, opinions, etc. of people involved in depth rather than breadth (Fellows and Liu, 2003). The tools for qualitative research are action research, case study, ethnographic research and ground theory (Neuman, 2006). The analysis of qualitative data involves filtering, sorting and other manipulations to prepare them for analytic techniques (Fellows and Liu, 2003). Detailed discussions of qualitative research approach are presented as follows:

a) Ethnographic Research

Ethnographic research in its broadest sense may be defined as the science of cultural description and is best accomplished by immersing oneself in the socio-cultural situation under study (Lang and Heiss, 1984). The focus of investigation is on the everyday behaviours (e.g. interactions, language, rituals) of the people in the group, with an intent to identify cultural norms, beliefs, social structures, and other cultural patterns (Leedy and Ormrod, 2001). Some researchers in the field of information
systems appear to turn to ethnographic research for information technology management (Davies and Nielsen, 1992) the development of information systems (Hughes et al., 1992), and design and evaluation of information system (Myers, 1999). The key strength of this method is that it gives a detailed view of the entire cultural scene by pulling together all aspects learned about the group and showing its complexity. The disadvantages are that it may have limited generalisibility to other topics or domains and it takes a lot longer than most other kinds of research (Mohamed, 2006).

b) Action Research
Action research is a vague concept but it has been defined as research that involves practical problem-solving which has theoretical relevance (Humford, 2001). Active involvement by the researcher is essential for identifying, promoting and evaluating problems and potential solutions (Fellows and Liu, 2003; Foster, 1972). Action researches intend not only to contribute to existing knowledge but also to help resolve some of the practical concerns of the people, or clients, who are trying to deal with a problematic situation (Gill and Johnson, 2002).

O’Brien (2001) indicates that although action research is used in a real situation rather than in contrived and experimental research, it can be used for preliminary or pilot research, especially when the situation is too ambiguous to frame to a precise research question. Lau (1997) reviewed the use of action research in information systems studies and proposed a term System Development (SD), which covers various methods used in analysis, design, development and implement of information and decision support system. As a special type of action research, SD is deemed that the development of a method or system can provide ‘a perfectly acceptable piece of evidence’ (an artifact) in support of a proof, where proof is taken to be any convincing argument in support of a worthwhile hypothesis. SD could be thought of a ‘proof-by-demonstration’ (Nunamaker et al., 1990). SD research has also been referred to as ‘engineering’ type research (Cecez-Kecmanovic, 1994).

Prototyping is also a type of action research method that is used in system development (Baskerville and Wood-Harper, 1998). In the context of information systems research, the theory/concept proposed usually leads to the development of a
prototype system with the intention of illustrating the theoretical framework (Burstein and Gregor, 1999). In this sense, prototyping and SD are similar methods. The development of a conceptual/ theory demonstrator and prototype is also a method of evaluation that is appropriate at the early stage of a software development life-cycle. It attempts to illustrate some or all of the proposed functionality of a system (Duke, 2001).

c) Case Studies

The case study approach is problem-oriented and is applicable to an individual, a group of people, an institution, or a whole community (Lang and Heiss, 1984.). Yin (2003) defined the case study research method as an empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and with multiple sources of evidence. It differs from action research in that the case study researcher seeks to study (organisational) phenomena and not to change them, unlike the action researcher who is often directly involved in planned organisational change (Avison et al., 2001).

According to Johnston et al. (1999), good and effective case study research should have the following elements:

• The research must begin with hypotheses developed by theory;
• The research design must be logical and systematic; and
• The findings must be independently evaluated.

Hence case studies are best used in studies that require deeper understanding of how and why things happen (Yin, 2003) rather than testing the relationships between them (Gordon and Langmaid, 1998).

Case studies can be either single or multiple. This single case study is analogous to a single experiment, and many of the same conditions that justify a single experiment also justify a single case study. It is appropriate where the objective is to develop a new theory rather then to test, develop or prove an existing theory or to establish statistical generalisation. When there is more than one single case, the study has to use multiple-case studies. In this situation, the term (single or multiple case studies) refers
Chapter 2 Research Methodology

to the way in which the results of the study can be interpreted—in other words, what is
the best way to consider the study either as serial (single) or parallel (multiple)
designs (Ganah, 2003).

The key strength of case study research is that it suitable for learning more about a
little known or poorly understood situation (Leedy and Ormrod, 2001). It also enables
the researcher to compare a number of different approaches to the same problem in
sufficient detail as to be able to draw out lessons which have general applicability
(Moore, 2000). In addition, case studies can help in achieving greater realism in
research, and requires a reasonably holistic research (Graham, 2000). They may also
be useful for investigating how an individual or program changes over time, perhaps
as the result of certain circumstances or interventions.

The weakness of cases studies is that they are usually restricted to a single event or
organisation; providing a limited basis for traditional ‘scientific’ generalisation (Yin,
2003). Hence they are often used for complex processes, their antecedents and
outcomes; this process may last for months or years and the concerned may not wait
for publication of the research result and when they are published may become out-of-
date. Another weakness is that the data collection and analysis process may be
influenced by the researcher’s interpretation of events, documents and interviews
(Drake et al., 1998).

2.2.5 Triangulation

Triangulation is the combination of quantitative or qualitative methods in the study of
the same phenomenon (Amaratunga et al., 2002). It can be a very powerful tool to
gain insight and results, to assist in making inference and in drawing conclusion, as
illustrated in Figure 2.2. The initial and obvious benefit of this is that it will involve
more data, thus being likely to improve the quality of the research (Denscombe, 2003).
Furthermore, researchers see things from different perspectives and understand the
topic in a more rounded and complete fashion than would be the case with data drawn
from just quantitative or qualitative approaches (Fellow and Liu, 2003; Denscombe,
2003). For example, using a quantitative method such as a questionnaire survey can
provide a abroad idea of the subject studied, and combining it with qualitative
methods such as interviews or case studies provides a better understanding of the
same study. Most importantly the triangulation approach provides an opportunity to corroborate findings that can enhance the validity of the data. They do not prove that the researcher has 'got it right', but they do give some confidence that the meaning of the data has some consistency across methods and that the findings are not closely tied with a particular method used to collect the data (Denscombe, 2003).

![Figure 2.2 Triangulation of Quantitative and Qualitative Data (Fellow and Liu, 2003)](image)

According to Easterby-Smith et al. (2001), there are four distinct categories of triangulation: theoretical, data, investigator and methodological:

- **Theoretical triangulation** involves borrowing models from one discipline and using them to explain situations in another discipline;
- **Data triangulation** refers to research where data is collected over different time from different sources;
- **Investigator triangulation** is where different people collect data on the same situation and data, and the results are then compared; and
- **Methodological triangulation** uses both quantitative as well as qualitative methods of data collection. These are extremely diverse and include questionnaires, interviews, telephone surveys, and field studies.
Through the aforementioned review, these quantitative and qualitative research methods have their own distinguishing characteristics, strengths and weaknesses. This is why the use of triangulation research is often encouraged. Table 2.2 shows a summary of comparisons between the two methods.

### Table 2.2 Distinguishing Characteristics of Quantitative and Qualitative methods

(Neuman, 2006; Abdullah, 2003; Amaratunga et al., 2002 and Leedy and Ormrod, 2001)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Quantitative Research</th>
<th>Qualitative Research</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>To explain and predict</td>
<td>To describe and explain</td>
</tr>
<tr>
<td></td>
<td>To confirm and validate</td>
<td>To explore and interpret</td>
</tr>
<tr>
<td></td>
<td>To test theory</td>
<td>To building theory</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Focused</td>
<td>Holistic</td>
</tr>
<tr>
<td></td>
<td>Known variables</td>
<td>Unknown variable</td>
</tr>
<tr>
<td></td>
<td>Established guidelines</td>
<td>Flexible</td>
</tr>
<tr>
<td></td>
<td>Statistic design</td>
<td>Emergent design</td>
</tr>
<tr>
<td></td>
<td>Context free</td>
<td>Context-bound</td>
</tr>
<tr>
<td></td>
<td>Detected view</td>
<td>Personal view</td>
</tr>
<tr>
<td><strong>Research Procedures</strong></td>
<td>Procedures are standard, and replication is frequent</td>
<td>Research procedure are particular, and replication is very rare</td>
</tr>
<tr>
<td><strong>Data Collection</strong></td>
<td>Representative, large sample</td>
<td>Informative, small sample</td>
</tr>
<tr>
<td></td>
<td>Standardized instruments</td>
<td>Observation, interviews</td>
</tr>
<tr>
<td><strong>Theory</strong></td>
<td>Theory is largely caused and is deductive</td>
<td>Theory can be causal or non-causal and is often inductive</td>
</tr>
<tr>
<td><strong>Data Analysis</strong></td>
<td>Analysis proceeds by using statistic, tables or charts and discussing how they show relates to hypothesis</td>
<td>Analysis proceeds by extracting themes or generalisation form evidence and organising data to present a coherent, consistent picture</td>
</tr>
<tr>
<td><strong>Reporting Finding</strong></td>
<td>Numbers</td>
<td>Word</td>
</tr>
<tr>
<td></td>
<td>Statistic, aggregated data</td>
<td>Narratives, individual quotes</td>
</tr>
<tr>
<td></td>
<td>Formal voice, scientific style</td>
<td>Personal voice, literary style</td>
</tr>
<tr>
<td><strong>Strengths</strong></td>
<td>Provide wide converge of the range of situation</td>
<td>Data gathering methods seen as natural than artificial</td>
</tr>
<tr>
<td></td>
<td>Fast and economical</td>
<td>Ability to look at change process over time</td>
</tr>
<tr>
<td></td>
<td>Where statistic are aggregated from large samples, they may be considerable relevance to policy decisions</td>
<td>Ability to understand people’s meaning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contribute to theory generation</td>
</tr>
<tr>
<td><strong>Weakness</strong></td>
<td>Tend to be rather inflexible and artificial</td>
<td>Data collection can be tedious and require more resources</td>
</tr>
<tr>
<td></td>
<td>Not very effective in understanding process</td>
<td>Analysis and interpretation of data may be more difficult</td>
</tr>
<tr>
<td></td>
<td>Not very helpful in generating theories</td>
<td>Harder to control the pace progress and end-points of research process</td>
</tr>
</tbody>
</table>
Major research methodologies have been briefly reviewed above. In the following sections, the research is designed and methods are selected based on in-depth consideration of the research objectives.

2.3 The Methodology Adopted for the Research

The main aim of the research is to develop a viable and consistent framework to improve building design optimisation through the application of MDO. This framework will enhance collaborative multidisciplinary design and accomplish various building performance objectives (e.g. safety, comfortable indoor environment, economic) at the same time. In order to achieve this aim and the specific objectives (refer to Section 1.4) in a complete fashion, combination of quantitative and qualitative research approaches were adopted.

2.3.1 Research Design

Research design is a logical plan for getting from here to there, where here may be defined as the initial set of questions to be answered, and there is set of conclusions (answers) (Yin, 2003). Another way of thinking about a research study is as a 'blueprint' of research, dealing with at least four problems: what questions to study, what data is relevant, what data to collect, and how to analyse the results (Philliber et al., 1980). Hence good design can help the researcher to avoid a situation in which the evidence does not address the initial research questions.

![Figure 2.3 The Research Process](image)
Figure 2.3 illustrates the research design, including the four major components

- Building of Knowledge: This stage started with an extensive literature review for building design optimisation, then the studies about MDO, including definition, decomposition schemes, formulations, were conducted.

- COPGA Framework Development: In this stage, the conceptual COPGA framework was developed through in depth analysis of the literature on the application of MDO and discussion with experts in the construction and other industries. To validate this framework, a mathematical problem was used.

- Modelling for Building Design: The COPGA framework was applied to a building design problem at this stage. It aimed at presenting how to implement this framework in the context of building design optimisation, such as formulating the design problem in a mathematical form and developing the corresponding design performance analysis programs.

- Evaluation: The last stage is to verify and validate the optimisation model for building design using semi-structural interviews.

However, the overall process is cyclic and interactive. After obtaining feedback and comments through the evaluation, the conceptual framework needed to be refined. Lessons learned from any of the other three stages are fed back to the “building of knowledge” stage, adopted and further explored. Finally such a research design helped to achieve the overall aim and specific objectives. Table 2.3 describes the key research issues in each stage such as: what questions have been answered, what objectives are met and which method is used.
Table 2.3: Overview of the Research Process

<table>
<thead>
<tr>
<th>Stage</th>
<th>Objective</th>
<th>Question</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building of Knowledge</td>
<td>• Objective One</td>
<td>• What are the characteristics of building design optimisation?</td>
<td>• Literature review</td>
</tr>
<tr>
<td></td>
<td>• Objective Two</td>
<td>• What are challenges to building design optimisation?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• What is MDO?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• How can MDO improve building design?</td>
<td></td>
</tr>
<tr>
<td>COPGA Framework Development</td>
<td>• Objective Three</td>
<td>• What the industry participants require in building design?</td>
<td>• Literature survey</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• How can the framework solve multi-objective multidisciplinary building design?</td>
<td>• Expert review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Which co-ordination strategy can be used in the framework?</td>
<td>• Mathematical problem testing</td>
</tr>
<tr>
<td>Modelling for Building Design</td>
<td>• Objective Four</td>
<td>• What kind of building design problem is suited to be implemented?</td>
<td>• Scenario</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• What are the variables, constraints and objectives to be considered in this building problem?</td>
<td>• Simulation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• How can this building problem be formulated in the proposed framework?</td>
<td>• Rapid prototyping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• How can this building problem be implemented in a computed-based prototype system?</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>• Objective Five</td>
<td>• Who can evaluate this framework?</td>
<td>• Rapid prototyping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Which features of this framework need to be evaluated?</td>
<td>• Semi-structured interview</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• What approach can be used to evaluate this framework?</td>
<td></td>
</tr>
</tbody>
</table>

The objective of each methodology used, type of data generated and the rationale for its adoption is discussed as follow. Various types of data collected are also used to support the triangulation approach in this research.

### 2.3.2 Building of Knowledge

In the first stage, the researcher refined, synthesised and used the theory of MDO to identify the limitations of current multi-discipline building design optimisation, and then developed meaningful research objectives. This stage involved a substantial literature review. The work conducted in this stage was mainly collecting and synthesising existing knowledge, rather than discovering or creating new knowledge.
A literature review is a systematic, explicit, reproducible method for identifying, evaluating and synthesising the existing body of completed and recorded work produced by researchers, scholars and practitioners (Fink, 2005). A good literature review helps to place the research in the context of what has already been done, allowing comparisons to be made and providing framework for further researcher. It also prevents researchers from repeating previous errors or redoing work that has already been done (Blaxter et al., 2006).

The specific objectives of literature review in this study were:

- To evaluate state-of-the-art building design optimisation with a view of design characteristics, optimisation algorithm adopted and relevant design objectives;
- To establish the theoretical background of MDO and its applications in other industries (e.g. manufacturing, aerospace and automotive);
- To study the factors which were addressed for the implementation of MDO in other industries;
- To explore possible frameworks to implement MDO in building design optimisation.

The initial stage of the research involved a wide spectrum of topics, which were related to the concept of optimisation and collaborative design paradigm. This is considered as the foundation of this research in reviewing the current practices of building design optimisation. It then focused specifically on the adoption of MDO as the main strategy to support the multi-objective multi-disciplinary building design optimisation. Since MDO is a methodology initiated from the manufacturing industry, it is still a relatively new subject to the construction industry. The literatures collected from other industries needed to be examined carefully from the building design perspectives.
The main keywords used in the literature survey are shown in Figure 2.4.

**Figure 2.4: Major Keywords Used in Searching the Relevant Literatures**

The sources of the material for the literature came from:

- Text books, journals, theses, reports, etc. from Loughborough University Library;
- Internet (World Wide Web) searches (e.g. Google Scholar);
- Conference Notes/Proceedings;
- Media articles; and
- Organization brochure and publications.

### 2.3.3 COPGA Framework Development

The second stage is overtly one of new knowledge creations. The researcher engaged in the creative and innovative design activity for the framework development including defining component, selecting the appropriate optimisation algorithm, developing the COPGA framework for building design, and validating the framework. The framework development followed an interactive process illustrated in Figure 2.5.
Figure 2.5 Process of Framework Development

Such an interactive process involves a set of the research step of analysing documents obtained from the literature survey, developing the COPGA framework, reviewing the framework by academic and industry experts, and testing and verifying the proposed framework using a mathematical problem. As a result, a viable and consistent COPGA framework was achieved.

At each step, various research methods were adopted. Detailed usages of these methods are discussed as follows:

1) Literature Survey

Based on the intensive literature survey, the characteristics of building design optimisation were summarised, and the initial thought of application of MDO in building design was established. The next question was what kind of MDO formulation could be developed for building design optimisation. A large body of literature about MDO applications (especially collaborative optimisation and analytic target cascading formulation) in other industries, ranging from PhD theses to the
American Institute of Aeronautics and Astronautics technical reports, has been collected. Qualitative analysis of these textual literatures is to accomplish the following objectives:

- To evaluate the existing MDO applications in other industries according to some specific characteristics, such as the decomposition scheme, co-ordination strategy, variable classifications, implementing technical issues (detailed information in Section 4.4);
- To select existing MDO formulation with the focus on building design properties (i.e. available analysis tools and design methods.) (Detailed information in Section 4.5.3);
- To select the appropriate optimisation algorithm based on the computation efficiency and obtaining global results (detailed information in Section 5.2.3).

In this research, the reports and thesis that describe the applications of the multi-level MDO formulation were analysed based on the following three criteria.

- Firstly, most applications of MDO are related with large-scale complex engineering designs. It is an attractive attribution of multi-level MDO formation dividing a large problem into a few manageable sub-problems. System partitioning methods differ from decomposition by component, discipline, sequential or matrix. Therefore what kind of decomposition schemes used in the MDO formulation becomes the first criterion of the category.
- Secondly, if the large problem has been decomposed into a few small sub-problems, it is inevitable some internal links among sub-problems are uncovered, which cannot be discovered when the problems are handled as a whole. Which kind of measure is used to identify relationship among sub-problems is the second criterion.
- Finally, after finding these links, some measures must be taken to obtain the same results from the MDO formulations as from the formulation without decomposition. In this study, these measures are described as the co-ordination strategy, the last criterion.
After such in-depth analyses, both ATC and CO formulation was effectively understood. Considering the characteristics of building design optimisation and comparisons between ATC and CO formulations, the CO formulation were selected for the proposed framework. However the CO formulation is merely suitable for single-objective problems, the Pareto-based genetic algorithm was proposed to extend the multi-objective capacity to the CO formulation; the conceptual COPGA framework was thus established.

2) Mathematical Problem Testing
Like other studies related with applications of optimisation techniques, a simple mathematical problem was chosen to validate the COPGA framework. To achieve this objective, the following three questions must be answered carefully:

1) Which kind of mathematical problem can be used to test the COPGA framework?
The development of the COPGA framework aims at solving building design optimisation problems involving complex links among disciplines and conflicting objectives. Hence the mathematical problem used to test the COPGA framework should present all these properties. On the other hand, this mathematical problem should be simple enough to manage. In other words, the number of each type of variables is not too large and with only two overall objectives.

2) How can this mathematical problem be implemented based on the COPGA framework?
Implementations of this mathematical problem using the COPGA framework must involve a large amount of calculation, which resulted from the iterative process in optimisation. It is impossible to solve this problem by manual methods; hence this problem should be implemented easily in a computer-based environment.

3) How is this result of the COPGA framework validated with regard to this mathematical problem?
Since the mathematical problem is small, the outcomes of the COPGA framework can be validated through comparing them with the results obtained from the ‘All-at-Once’
frameworks, namely, without decomposition (refer to Section 5.6.3.1.) Such a validation approach has been widely used in other MDO applications (Budianto, 2000; Choudhary, 2004). Detailed results are presented in Chapter Five.

If the result from the COPGA framework is correct in the mathematical problem, it implies that the COPGA framework works well in computer environments, and that there are no technical errors. If not, this framework will be further refined through expert review.

Operations

3) Expert interview

There are two reasons for the application of this approach. The first one is to confirm the industry participants’ requirements and current problems about multi-objective and multidisciplinary building design identified in the literature review. In order to achieve this objective, semi-structured interviews were conducted. The experts include architects, structural and building services engineers, an academic researcher and a specialised software developer. Experts in these interviews merely focus on design participant, but future studies could include other stakeholders, such as quantity surveyor, client, facility managers, contractor and material supplier. The questionnaire template used in these interviews is attached in Appendix Two. Section 3.6 presents the analysis of the data obtained from these interviews, this contributed to the development of an appropriate framework for building design.

The second purpose is to verify the author’s understandings of the theoretical concept through communication with researchers from the construction and other industries. As discussed above, the application of MDO gained popularity in the aerospace and automotive industries, and researchers in these industries have developed wider knowledge. The author discussed some personal thoughts about MDO with these experienced researchers by e-mail or telephone. These experts affirmed the author’s understandings of MDO and emphasised some issues considered whilst implementing the MDO framework. For example, with regard to post-optimality \(^1\) (Braun and Kroo, 1993), it is not necessary to integrate it into the MDO framework because the main

---

\(^1\) The post-optimality information generated through first-order computations can be used to accurately predict the effect of constraint and parameter perturbations on the optimal solution.
purpose of post-optimality is to decrease the computational expense. Additionally, some researchers with experience in coordinating conflicts in multi-disciplinary design were also contacted at this stage. At the beginning, it was intended to integrate game theory into the MDO framework as a coordination strategy, but through critical review of other researchers' works. This idea was rejected because the concept of game theory ignores the interaction between disciplines. Finally, this framework is also further improved from other researchers' comments and suggestions at conferences and workshops. The iterative development of the framework resulted in a series of papers at different development stages of the project, which are listed in Appendix one.

2.3.4 Modelling for Building Design

Modelling is the process of constructing a model, a representation of a designed or actual object, process or system. For a representation of reality, it must include the essential features of reality whilst being reasonably cheap to construct and operate and easy to use (Fellows and Liu, 2003). The modelling process in this research is depicted in Figure 2.6.

![Figure 2.6 Modelling Process and Research Method Adopted](image-url)
The COPGA framework has been developed then validated using a mathematical problem. The following stage was to implement the model to demonstrate that the COPGA framework could work well in a multi-objective multidisciplinary building design optimisation case. With this in the mind, three methods were used to accomplish this objective:

1) The Scenario

Initially, it was planned to implement the COPGA framework in a completed building design project, and then evaluate the framework by comparing the results generated with the real design solutions. However, owing to the complexity of the practical project (e.g. a large number of variables, complicated specialised knowledge, etc.), it was difficult to be achieved by one person within the limited time frame. Also, detailed design information is confidential for unauthorised users to access. Due to these constraints, a design scenario derived from real industrial cases was adopted to depict a multi-objective multidisciplinary building design problems.

The scenario design often combines with other research methods, such as case study or survey (interviews and questionnaires) in order to identify all possible scenarios and key factors in each scenario (Glenn and Gordon, 2003). Because MDO is a new research area, few people know about it and those who know are often merely aware of the term with general meaning, especially in the construction industry. Also the implementation of MDO involves many specialised terms, such as post-optimality, global optimisation and coupling variable. These terms are unfamiliar to many building designers. Therefore construction practitioners without MDO background can hardly provide suitable information to the scenario design. Thus the survey and interview methods are not appropriate at the current stage. A possible method to design the scenarios is to analyse existing building projects and to identify key factors that the COPGA framework focuses on, such as complex relationships among disciplinary design. Therefore this scenario is an imaginary design task, but the problems demonstrated in it exist in a real building design environment.
The other reason for the use of the design scenario is that the simple mathematical problem is so abstract that building designers cannot understand what kind of problem this framework faces on. The scenario, however, provides a context for understanding the features and benefit of the COPGA framework (detailed information in Sections 6.2). The rationality of the scenarios was assessed via presenting it on academic conference in the early framework development stage and evaluating them with a prototype system at the final stage of the research.

2) Prototyping

Research can be generally individual work which discovers and describes existing reality (explorative research) or which aims at creating a new reality (e.g. new technology or processes) that needs to be evaluated and justified. The research described in this thesis aims at developing a MDO framework that can be used in building design to solve multi-objective multidisciplinary problems. An important element of the methodology used was the automation and implementation of the COPGA framework. More specifically, the intention was to encapsulate the developed COPGA framework into a prototype, which is known as rapid prototyping. Laudon and Laudon (2002) defined prototyping as a process of building an experimental system quickly and inexpensively for demonstration and evaluation, so that the users can gain better information. The key benefits of the prototyping include: short development time; short user reaction time (feedback from user); improved user' understanding of the system, its information needs, and its capabilities; and low cost (Turban and Aronson, 1998). However, Laudon and Laudon (2002) stressed that prototyping could not gloss over essential steps in systems development. Even if the completed prototype works reasonably well, the manager may not believe there is no need for reprogramming, redesign, or full documentation and testing to build a polished production system.

In this research the prototyping system was used to implement the COPGA framework in the design scenario. This design scenario involved two overall objective functions and two disciplines, - structural and HVAC (Heating, Ventilation and Air Conditions). In the context of this scenario, the COPGA framework was made up of
the internal and external cycle, whilst the internal cycle could be divided into five components, including one system-level optimisation block, two subsystem-level optimisation blocks and two subsystem-level analysis blocks. The optimisation algorithms in one system-level block and two subsystem-level optimisation blocks directly utilised optimisation toolkits in Matlab. The two subsystem-level analysis blocks is used to perform subsystem-level optimisation through simulation technology. The detailed simulation process is described in the next section. The Pareto-based genetic algorithm used in the external cycle was coded in the Matlab programming language.

After finishing compiling the program in each block, the next step was to connect them into an integrated system. It was crucial to identify what kind of data and when were required to transfer from one block to other blocks. For example the steps of initialisation of population, crossover and mutation operation would call the internal cycle to obtain the compatible values of the interdisciplinary variables during the process of Pareto-based genetic algorithm, whilst the system-level optimisation block in the internal cycle required the values of interdisciplinary variables from two subsystem-level optimisation blocks and keeping those local variables at subsystem level (detailed explanation in Section 5.5.2.2). Finally, the prototyping system was developed through effective organisation of different blocks and right data transmission.

3) Simulation

Simulation is used to assist prediction of the behaviour of intangible not human reality and/or to revise a model to enhance its predictive accuracy or predictive capability; Morgan (1984) suggests a variety of purposed for simulation;

- Examine the performance of alternative technique;
- Check complex mathematical/ analytic model
- Evaluate the behaviour of complex random variables, the precise distribution(s) of which is (are) unknown.
In the context of this research, simulation was used to examine the structural and thermal performances with different sets of design variables, and were coded within Matlab programming environment.

Firstly, the relevant literatures were thoroughly reviewed to determine design variables and objective functions considered, analysis method adopted and the corresponding building regulations. These literatures included the CIBSE (Chartered Institution of Building Services Engineers) Guide, HVAC design, Steel design handbook, Steel design code (BS5950) and so on. Then, a few assumptions were made to simplify the design problem. For instance, some unnecessary parameters, such as solar coefficient, were fixed as constant (detailed information in Section 6.2.2). In a word, the aim of building these simulations was to test the COPGA framework in the context of the design scenario. Hence simulation programmes in the prototype system were merely required to easily and quickly development with the necessary accuracy, but do not work as the commercial analysis software which involved the large amount of variables.

2.3.5 Evaluation

Evaluation will be used to represent the appraisal of the whole system. Within evaluation, it is generally agreed that there are two sub-sections: verification and validation, which are defined as follows (Miles et al., 2000):

- Validation is the process which determines whether or not a system meets the required specification and is suitable for its intended purpose. Validation ensures that the software has been formulated in the intended manner.
- Verification is the process of ensuring that the product does not contain any technical errors. Verification ensures that the software has been formulated correctly.

The prototype system in this study was designed to implement the COPGA framework in a real building design problem. The semi-structured interviews were conducted to evaluate this prototype system. Compared with other evaluation approaches (e.g. questionnaire and evaluation workshop), such a face-to-face two-way
conversation provides the opportunity for the interviewers to obtain more valuable and in-depth comments through a clear understanding of the research context. Participants in this evaluation included the architects, structural engineers and HVAC engineers, hence the discipline-specific terminologies were used to describe the COPGA framework. This means that some more technical terms, such as GA and Pareto optimality, could be explained clearly based on the interviewer's background.

The aim of this work was to evaluate the COPGA framework which focused on the multi-objective multi-disciplinary building design optimisation problem. To achieve this aim, the semi-structured interviews contained four parts which reflected the usefulness, appropriateness, correctness and directions of further development of the COPGA framework. The first part was aimed at evaluating the key features of the COPGA framework, such as coordination strategy and optimisation algorithm adopted. The second part was to validate the performances of the COPGA in this scenario. The last part was incorporated open questions with regard to the limitations for the application of the COPGA framework in reality. The semi-structured interview template is presented in Appendix Five. Data obtained from the semi-structured interviews are analysed in Chapter 7.

However, in the first trial interview, the interviewee reported two problems: one was that it took too long to answer all questions in details; the other was that it was difficult to answer whether the results from the COPGA framework were right because too much detailed design information was presented in this design scenario, and consequently the interviewee could not check the results in a short time. Considering these comments, the method of validating the results of this framework was changed. Since the COPGA framework has been validated through the mathematical problem during the process of development, it proved that there were no technical errors, such as data transfer between different blocks, application of optimisation algorithm. Compared with implementation of the mathematical problem, the main different factor was that performance simulations (i.e. structural and thermal analysis) were required to integrate into the prototyping system. If these simulation programmes could be validated, this implied that the outcome of the prototyping system was correct since the whole process of optimisation had been validated in the mathematical problem.
Chapter 2 Research Methodology

There are three main possible techniques for assessing the accuracy of simulation models (Bowman and Lomas, 1985): analytical verification, inter-model comparisons and empirical validation. This research adopted the inter-model comparisons that the author's simulation programs were validated by the best-known commercial software. The structural simulation program based on the matrix-displacement methods would be evaluated by the STAAD pro software, the commercial software called DesignBuilder was used to assess the thermal analysis program designed in line with calculation introduced by CIBSE Guide A (1999). Detailed comparisons of results between two simulation programs are presented in Section 6.3.

2.4 Summary

This chapter has described the theoretical background of methodology used for model development in this research with the rationales for their adoption. These methodologies used include:

- Literature
- Expert review
- Mathematical problem testing
- Scenario
- Simulation
- Prototyping
- Semi-structured interview

Each methodology is carefully selected to meet the research requirements in each stage. The earlier stage of the research requires a general scoping study of the research topic; and the literature review achieves the objectives of exploring the subject in greater detail. It also helps the researcher to identify the specific issues that need to be investigated further. The next stage adopts the qualitative content analysis to determine the conceptual COPGA framework; critical reviews from construction and other industry experts to further refine and improve this framework, and a mathematical problem to validate it. The work of the third stage is implementation of the COPGA framework in the context of building design; the methodologies in this stage include design scenario, simulation, and prototyping. Finally semi-structured
interviews were conducted to evaluate this research, including rationale of this study and major features, performances and limitations of the COPGA framework.

The findings from each part of the research are used to contribute to development of the next stage of the study, and to confirm the findings by a ‘triangulation’ approach to ensure that every issue is fully explored. The detailed analysis of the data generated from these methodologies adopted is discussed in the following chapters where the main findings of the research are presented.
CHAPTER THREE: BUILDING DESIGN OPTIMISATION

3.1 Introduction
The proposition of design by optimisation in building and construction was earlier compiled in 1966 (Torres et al., 1966). However major changes have taken place in the field of optimisation techniques available for solving the problems, and the nature of the problems themselves. Hence this chapter starts with reviews on two kinds of optimisation algorithms, including gradient-based and derivative-free algorithms, and describes their distinguishing properties. It is then followed by the studies of past and current optimisation work ranging from architectural, to structural and HVAC design. Finally challenges to improve building design optimisation are identified from the literature and semi-structured interviews.

3.2 Design Optimisation
In its pure definition, optimisation refers to the studies of problems in which one seeks to minimise or maximise a real function by systematically choosing the values of real or integer variables within an allowed set (Wikipedia, 2007b). This is not the definition that is used here; the engineering interpretation of optimisation could be referred to as:

- Optimisation is a key to accelerating exploration of search space, and thus provides greatest opportunity to reduce design cycle time (Rowell and Korte, 2003).
- Design improvement, not optimality, may guide a user in the course of optimisation (Kroo and Manning, 2000).
- One part of optimisation is the evaluation of the design proposal. The second part is the generation of new and hopefully better designs. Thus, optimisation consists of both analysis (evaluation) and synthesis (generation of new solutions) (Andersson, 2001).
- Optimisation means improving or fine-tuning the design in terms of one or more performance aspects (Papalambros, 2002).
Based on the above definitions, optimisation can be seen as a technique which helps designers to explore more design space to create a better design solution rather than obtaining a minimal or maximal solution. This is why researchers in the community of optimisation take much effort on finding new optimisation search strategies, such as direct searches and genetic algorithm (referred to Section 3.3.2).

Generally engineering design involves the generation of design alternatives and the selection of the best one. Since the number of possible design options is practically infinite for most products, human judgement is needed to decide which options to include in the consideration of alternative designs and which to dismiss. Thus the process of engineering design is a decision-making process (Chen et al., 1998; Azarm and Narayanan, 2002). The term 'optimisation' is widely used in a rather loose way to indicate doing something better than the way we are currently doing it. In product development this term is often used in a similar manner to indicate making product decisions that yield a better product. Figure 3.1 shows that there exists a close relationships between decision-based design and optimisation.

A careful comparison between decision making and optimisation reveals that the options space is equivalent to the set of permissible value of \( x \) in feasible region; the expectation of any given \( x \) is assigned by \( F(x) \); and the preference is stated that

---

**Figure 3.1 Optimisation vs. Decision-based Design** (Renaud 2002)
more is better (maximisation) or less is better (minimisation). Thus optimisation can be used to capture the properties of decision made. This recognition allows the application of rigorous optimisation techniques to the case of decision-making design (Renaud, 2002).

3.3 Taxonomy of Optimisation Algorithms
The formal model of an optimisation problem is a statement of the form:

\[
\text{Minimise } f(x) \\
\text{Subject to } h(x) = 0; \quad g(x) \leq 0; \quad x \in \mathbb{N} \subseteq \mathbb{R}^n \quad \text{(Equation 3.1)}
\]

Where \( f(x) \) is the objective function; \( h(x) \) is the equality constraint and \( g(x) \) is the inequality constraint; \( \mathbb{N} \) is the set constraints of the n-dimensional real space \( \mathbb{R}^n \). After formulating the problem with a set of variables, objective function and constraints, optimisation algorithms can be applied to find the optimal solution(s).

Normally these optimisation problems are defined with the following six elements:

- Function characteristics: An optimisation algorithm that minimises or maximises one or more objective functions subject to a number of constraints. The nature of objective and constraint functions can influence the choice of optimisation algorithm, where the functions are differentiable. The problem may lend itself to an exact solution by a calculus-based optimisation algorithm. However the first and second objective functions of practical engineering design problems cannot be estimated analytically because of non-quantifiable design objectives and constraints such as legal and aesthetic requirements. These problems are regarded as derivative-free optimisation problems.

- Single or multiple variables: The number of variables optimised can also influence the choice of optimisation algorithm. Some problems exist only as one variable. However most problems have more than one variable. The complexity of optimisation increases with the number of variables.

- Discrete or continuous variables: If the number of the variables value is finite, then the problem is discrete and the optimum consists of a certain
combination of variable values. However some problems are regarded as mixed-integer optimisations as both discrete and continuous variable exist.

- Constrained or unconstrained: The range of variable values is often restricted by simple bounds or constraint functions. The constraints can be formed as either equalities or inequalities. There are numerous ways of dealing with the constraints; each particular method often depends greatly on the optimisation algorithm adopted. The constraints can act as constraint functions to limit an operation within the problem, as well as acting as bounds on the variables.

- Local or global optimisation: When a solution is better than its neighbouring points, it is a local optimum. If more than one local optimum exist, then the lowest of all local optima is the global optimum. Usually there is no guaranteed optimisation algorithm for finding the global optimum.

- Linear or nonlinear problem: An optimisation problem in which the objective and constraint functions are linear functions of their variables is referred to as a linear programming problem. On the other hand, if at least one of the objective or constraint functions is nonlinear, then it is referred to as a nonlinear programming problem.

In addition to these, another important aspect is whether it is a single objective or multi-objective optimisation problem. The optimisation algorithms with regard to the single objective problems are studied in the following section, while reviews on multi-objective optimisation algorithm are presented in Section 5.2.

3.3.1 Gradient-based Algorithm

Gradient-based algorithms use the gradient of the objective function, $\nabla f(x)$, at the current iteration point to gather information about the structure of the function and to determine the direction of the next step in the iteration. For differentiable objective functions the gradient is an analytic expression. In other cases the gradient must be approximated numerically. The numerical approximation is not an easy task and can cost several extra function evaluations in each iteration. The method may also be sensitive to errors in the gradient approximation.
The gradient-based algorithms can be grouped into unconstrained methods and constrained methods.

In the former method, if the $\nabla f(x)$ is zero and Hessian matrix is positive for a given $x$, this ensures that the design is at least a relative minimum. For example, Newton methods, the process starts with some arbitrary initial values, then the next vector of variable $x_{k+1}$ is decided by the current point $x_k$ at iteration $k$ using Equation 3.2

$$x_{k+1} = x_k - \frac{f(x_k)}{\nabla f(x_k)} \quad \text{(Equation 3.2)}$$

Finally the process is stopped when $\nabla f(x_k)$ is less in magnitude than a specified constant (e.g. $10^{-4}$).

The latter method is preferable in most cases of gradient-based algorithm. Karush-Kuhn-Tucker (KKT) conditions presented in Equation 3.3 are a set of necessary conditions for constrained optimality.

1. $x^*$ is feasible
2. $\lambda_j g_j(x^*) = 0 \quad j=1, \ldots, m \quad \lambda_j \geq 0$
3. $\nabla f(x^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*) + \sum_{k=m}^{k+m} \lambda_k \nabla h_k(x^*) = 0 \quad \text{(Equation 3.3)}$

Where $x^*$ is the optimum design, $\lambda$ is the Lagrange multiplier. There are $m$ inequality constraints and $k$ equality constraints that are active at the optimum.

Sequential quadratic programming (SQP) is a standard gradient-based algorithm for solving a constrained optimisation problem, and has been shown to have a very fast convergence rate (Schittkowski, 1985). The SQP algorithm approximates the problem into a quadratic sub-problem (QP) by taking a second order approximation of the objective function, and a first order approximation of the constraints in the QP sub-problem are linear, KKT conditions reduce to a system of linear equations that can be solved explicitly. The solution to the QP sub-problem produces a search direction for the original problem. Next, the algorithm proceeds by determining an appropriate step size, iterating until the termination criteria have been met. The criteria could be a
maximum number of iterations, absolute or relative changes in objective function and KKT conditions (Vanderplaats, 1984).

However the gradient-based algorithm is usually trapped in a local optimal result. Hence a multi-start strategy should be applied, which means that the optimisation is run repeatedly with different starting points, and the lowest of the local minima found is taken be a good local optimum or possibly the global optimum. At the same time, this requires the user to have an intuition about the design problems which result in choosing the starting points to find the local minima around the points of interest (Loosemore, 2003).

### 3.3.2 Derivative-free Algorithm

In the many cases, even if it is known that objective function is smooth, its analytical expression is available, but computation of its value may be expensive or affected by the present of noise. As a result the first order derivatives cannot be explicitly or approximately calculated. The derivative-free algorithm does not attempt to directly compute the unavailable derivative information, but work through repeated function evaluation accepting and rejecting candidate solutions, and the search for an optimum proceeds iteratively. This motivates the increasing interest in the use of derivative-free methods.

The search method in this kind of algorithms could vary from high intelligent search techniques to a simple random search. Such search methods are particularly attractive for discrete variables and discontinue functions. Moreover they can do global searches. These algorithms, on the other hand, usually result in taking a long time to convergent to the optimum point, especially if the problem is large, due to lack of gradient information.

Most of the derivative-free algorithms are to minimise or maximise the objective functions as an unconstrained function but to provide some penalties to limit constraint violations. The classical approach is to create a pseudo-objective function of the form shown in Equation 3.4 (Papalambros and Wilde, 2000):
\[ \theta(x,r) = f(x) + P(g(x),r) \]  

(Equation 3.4)

Where \( f(x) \) is the original objective function, \( P(g(x),r) \) is an imposed penalty function. The scalar \( r \) is a multiplier which determines the magnitude of the penalty. If a small value for \( r \) is chosen, the resulting function \( \theta(x,r) \) is easily optimised but may yield major constraint violations. While a large value of \( r \) could ensure near satisfaction of all constraints but create a very poorly conditioned optimisation problem (Vanderplaats, 1984).

Unlike the gradient-based algorithm, there is no canonical condition (i.e. KKT condition) to define optimal point in the derivative-free algorithm. Instead various algorithms adopt specific terminal criterion based on optimisation problem. This broad spectrum of derivative-free methods can be classified into direct and stochastic search.

**Direct Search**

The direct search is a sequential examination of trial solutions involving the comparison of each trial solution with the best obtained up to that time together with a strategy for determining what the next trial solution will be (Kolda et al., 2003). Examples of direct search methods are the Nelder-Mead simplex method (Nelder and Mead, 1965) and Hooke-Jeeves method (Hooke and Jeeves, 1961).

In the simplex method the value of objective function is evaluated at a number of points constituting a grid in the feasible region. The grid is moved towards the minimum (i.e. the desired point) through successive changes, whereby the worst point (where the objective function has the highest value) is replaced by another one in a favourable direction at any single move through three main operations. They are (Bouchlaghem, 1990):

- **Reflection**: where \( x_h \) is replaced by:
  \[ x_r = (1 + \alpha)x_0 - \alpha x_h \quad \alpha > 0 \]

- **Expansion**: where \( x_e \) is expanded in the direction along with a further improvement of the function value may be expected.
\[ x_c = \alpha x_h + (1 - \alpha) x_0 \quad \gamma > 1 \]

- Contraction: By which the simplex is contracted:
\[ x_c = \beta x_h + (1 - \beta) x_0 \quad 0 < \beta < 1 \]

Where,
- \( x_h \) is the vertex corresponding to the highest value of the objective function;
- \( x_0 \) is the central point of all \( x_i \) except \( i = h \) and is given by:
\[ x_0 = \frac{1}{n} \sum_{i=h}^{n} x_i. \]

Compared with the simplex method, the search approach used by the Hooke-Jeeves method is less cautious and more speculative (Kolda et al., 2003). This method takes steps along the valley of the objective function. It assumes it is worthwhile to make further explorations in a direction that was successful in a previous step. To illustrate the exploratory moves, suppose that iteration \( k-1 \) was successfully (no exploratory step is attempted when \( k = 0 \)). The iteration \( k \) begins by conducting a co-ordination search about a trial point \( x_p = x_k + (x_k - x_{k-1}) \), rather than about the current value of \( x_k \). The idea is that since the step \( x_k - x_{k-1} \) from \( x_{k-1} \) to \( x_k \) led to an decrease in \( f(x) \) (i.e. desire to minimisation ), then further progress may be possible in the general direction of \( x_k - x_{k-1} \). Such a step is called a pattern step. The objective is evaluated at the trial point \( x_p \) and the algorithm then proceeds to conduct a co-ordination search about \( x_p \), even if \( f(x_p) \geq f(x_k) \). If the co-ordination search for \( x_p \) is successful, and finds the next point \( x_s \) such that \( f(x_s) < f(x_k) \) then \( x_s \) is accepted as the new value of \( x_{k+1} \). If no such point is found around \( x_p \), then the pattern step is deemed unsuccessful, and the method is reduced to coordinate a search for \( x_k \) (Hooke and Jeeves, 1961).

Like other derivative-free algorithms, the direct search remains popular with practitioners because they do not require rigorous mathematical properties and could obtain a global optimal result. However, performance based on these algorithms could require the large amount of computational time as the number of variables increased. It was asserted that they were best suited for problems with a small number of variables (Meza et al., 1996). In addition, slow asymptotic convergence is the other
drawback. This stems from the lack of valuable derivative information, which often takes a very long time to locate the exact minimum (Kolda et al., 2003).

**Stochastic Search**

A stochastic search involves random elements in their search strategy. The methods have often been developed by analogies to a phenomenon. Stochastic algorithms are usually easier to implement and are able to handle a wide range of optimisation problems associated with discrete and unordered design variables, non-differentiable and non-continuous objective functions. Simulated annealing and Genetic algorithms are popular examples of stochastic search used in optimisation.

Simulated annealing (SA) algorithms were originally the Monte Carlo method (Metropolis et al., 1953). The first proper SA algorithm was investigated by Kirkpatrick (1983). This algorithm exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure and the search for a minimum in a general system (Metropolis et al., 1953). Such a random search is controlled by two parameters, including the initial temperature and cooling rate. The initial temperature determines the level of randomness in the algorithm. If the higher annealing temperature provides the particles of the solid with a very higher mobility, this means worse points are likely to be accepted. Meanwhile the cooling rate determines how quickly the level of randomness decreases as the number of iterations of the algorithm increase. A slow cooling rate could be more likely to find a global optimum; it also increases convergence time. The whole process of SA is illustrated in Figure 3.2.
Figure 3.2 Process of Simulated Annealing Algorithm (Ali et al., 2002)

The major advantage of SA is the ability to find global optimum, because it employs a random search which not only accepts changes that decrease objective function (i.e. the desired solution), but also some changes that increase it. The latter is accepted with a probability ($p$).

$$p = \exp(-\frac{\delta f}{T})$$

Where $\delta f$ is the increase in objective function and $T$ is a control temperature.

SA can also be used to solve problems involving both continuous and discrete variables (Ali et al., 2002). All in all, the SA algorithm is most efficiently used for getting a ‘good enough’ solution in a reasonable run time.
Introduced by John Holland in 1973, Genetic Algorithms (GAs) is a heuristic search method derived from natural selection and evolution (Goldberg, 1989). In the context of GAs, each design candidate is represented by a string that is called *individual*. Each string is made up of *chromosomes* that stand for design variables, whilst the number of design candidates forms a *population*. The objective functions of each design are evaluated as *fitness* functions.

![Figure 3.3 Process of Genetic Algorithm (Sisk, 1999)](image)

The GA process illustrated in Figure 3.3 involves a competitive selection to remove poor solutions. This process starts with generating an initial population of possible solutions, usually a fixed size. If the size is too small, the GA's population risks being dominated by one, possibly sub-optimal, solution within a short number of generations. An excessively large population is computationally inefficient. For most applications, a population size of between 20 and 100 individuals is recommended (Barclay, 1993).

Once an initial population has been generated, the GA performs a loop of fitness assessment. For many problems, formulation of a fitness function that distinguishes between the possible solutions can be rather complex owing to involving more than one aspect of the problem (e.g. minimise capital cost and life-cycle cost). After the
fitness assessments, the genetic operation follows until the population of possible solutions are found using a certain satisfying criteria, such as reaching a plateau of average fitness or a set number of generations. The genetic operations including selection, crossover and mutation aim at producing a new generation. These operations also result in obtaining the global solutions. At the first stage in the process of genetic operation, individuals in the current generation are selected and copied to a 'mating-pool'. Although there are various selection approaches (e.g. stochastic tournament and deterministic sampling), the fundamental of them is that individuals with higher fitness have a proportionately higher chance of being selected (Hooper et al., 1992). The crossover operator is analogous to the natural sexual mating process. It performs an exchange of chromosome between two randomly selected individuals, resulting in two new child individuals. However if the genetic material needed to get the optimal solution, is not presented within the initial population, the GA can never satisfy its chief criteria through selection and crossover alone. Mutation introduces new genetic material into the population. In this respect mutation maintains population diversity by introducing complete novel genetic material. This new material enables the GA to arrive at some undiscovered optimum.

Compared with algorithms based on direct search, both SAs and GAs are more versatile and as a result applicable to a wider range of problems. They can also handle relatively larger problems because they do not search the design space exhaustively.

3.3.3 Comparisons between Gradient-based and Derivative-free Algorithm

An optimisation algorithm, if used effectively, can greatly reduce engineering design time and improve, efficient, and economical designs. However the successes of these algorithms rely to a large extent on understanding of their advantages and limitations, and then use them in the proper environment. The following Table 3.1 describes their differences on the basis of optimisation properties.


<table>
<thead>
<tr>
<th>Properties</th>
<th>Gradient-based algorithms</th>
<th>Derivative-free algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope of application</strong></td>
<td>They are proven to converge to minima for formulations with continuous and smooth functions with respect to all design variables.</td>
<td>They demonstrate no special problem with optimisation formulation with discontinuous function objective, multiple local minima, discrete variables and nonsmooth design space.</td>
</tr>
<tr>
<td><strong>Local or global optimal solution</strong></td>
<td>These methods are prone to converging on local minima.</td>
<td>The extraordinary features combined with these algorithms enable to find global solution from a large number of alternatives, such as genetic operations in the GAs.</td>
</tr>
<tr>
<td><strong>Dimension of variable</strong></td>
<td>Large number of design variables and constraints can be handled.</td>
<td>The solution time for this kind of algorithm rise at a fast rate with increase of number of variable.</td>
</tr>
<tr>
<td><strong>Number to call analysis model</strong></td>
<td>The number to call analysis model is relatively small comparing with derivative-free algorithm.</td>
<td>These methods require a large of number of call for constraint evaluation.</td>
</tr>
<tr>
<td><strong>Convergence rate</strong></td>
<td>With the guide of derivative information they can quickly attain some level of convergence.</td>
<td>The convergence rate of these methods is generally slow. It is hard to decide how long the algorithm should be run to be able to arrive at a ‘good solution’.</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td>Once mathematical optimisation model is developed, these codes can be executed with little human involvement until their completion.</td>
<td>They are arduous to set up because the user must choose how to present and encode practical problem in optimisation process. Jo and Gero (1998) showed it was difficult to formulate design knowledge into representation of GAs.</td>
</tr>
<tr>
<td><strong>Robustness</strong></td>
<td>The gradient-based algorithm is able to find an answer, starting from arbitrary initial points.</td>
<td>They, unlike gradient-based algorithms, cannot find a unique optimal result in different performances, but these results are near to a real optimum.</td>
</tr>
</tbody>
</table>
3.4 Characteristics of Building Design Optimisation

Some characteristics of building design make the application of optimisation techniques more difficult, such as nonlinear objective and constraint functions, continuous and discrete variables, competing performances and the involvement of different experts. Facing on these difficulties, previous researches had been undertaken to explore the use of different optimisation algorithms. This section mainly focuses on reviewing these works.

3.4.1 Multi-variable Design

The first characteristic is a high-dimensional building space to be explored when implementing optimisation. This problem stems from the following two reasons. Firstly the number of variables to describe individual building performance is very large, these performance areas include safety, comfort, aesthetic, and function (Coley and Schukat, 2002 and Saporito et al., 2001). Coley and Schukat (2002) presented that there is a large number of constructional parameters (more than 100 variables) in just a building environmental model. Wright et al (2002) demonstrated there is a total 200 design variables to configure control system setpoints and the size of HVAC components.

Secondly, the number of combinations of design variables increase an exponentially with the number of variables. Hence an efficient optimisation algorithm needs to cope with the computation demand generated by the exponential growth in the search space. Saporito et al., (2001) indicated there were 243 design combinations to test in a office building case despite of only four variables being involved, so the lattice method was used, in which the multi-dimensional problem was transformed into a one-dimensional problem, consequently reducing the number of alternative solutions from 243 to 9. Parvin and Serpen (1999) presented the application of artificial neural network (ANN) in a truss with a fixed geometric configuration; optimisation for better design could be achieved by changing the cross-sectional area of different structural members. The ANN algorithm could perform computation from each mesh node in parallel, which drastically reduced the search time for the optimisation process. Michalek et al., (2002b) put forward an interactive optimisation method for a floor layout problem. The advantage is that the designer can interactively and intuitively
define and refine the optimisation problem statement and guide the search algorithm into the area of interest during the optimisation process. For example, if a designer sees the design is moving into an undesirable area of design space, he/she can intervene and force a search into a new area of the space. This method used the designers' experience and intuition to avoid some unnecessary searches.

3.4.2 Multi-type Variables

Both continuous and discrete variables are used to describe the building design problem. The continuous design variables are real numbers, e.g. orientation of building, which may be varied continuously between the lower and upper bounds. Building design also involves the selection of components, which is a discrete process. Therefore, design variables specifying the selection of building components may be represented by integer values, i.e. vary in a stepwise fashion between the boundaries depending on the type of component. Some gradient-based optimisation methods, such as sequential quadratic programming, are fast and efficient for the continuous variable, while these methods treat discrete variable as continuous and employ a rounding-off technique that gives approximate values which may affect the optimum final solution (Bouchlaghem and Letherman, 1990). Furthermore the derivative of the objective function with respect to variables is necessary for this kind of optimisation algorithm (Isenberg et al., 2002). It is difficult for building design to be formulated fully in algebraic form (Medjdoub and Yannou, 2000). Focusing on these problems, some researchers used stochastic optimisation algorithms to deal with discrete variables. For example, Gonzalez-Monroy and Cordoba (2000) used simulated annealing to model energy supply, which include a number of different devices of transformation and storage expressed in discrete variables. Wright et al. (2002) solved problem of discrete variables, such as equipment control status and fan type, by the use of GAs. Kang and Zong (2004) adopted a harmony search heuristic algorithm to solve 25-bar space truss with the consideration of the cross-section areas as discrete variable, because the structural components are in standard sizes in practice. However, when implementing these stochastic optimisations, a suitable step size for continuous variable must be determined. If the step is too large many possible solution may be missed; if is too small the search will require a lot of computational time.
3.4.3 Multi-objective Design

A reason for the use of an optimisation algorithm is that they can help designers select the best option from a number of feasible solutions. Thus it is necessary to determine some measures of evaluation of these alternative solutions. These measures are called objectives or criteria. In some cases, the evaluation of a design uses the single-criteria, e.g. energy saving. However, most building designs cannot be assessed using one criterion, but using many independent criteria (Jedrzejuk and Marks, 2002, Li et al., 1999, Marler and Arora, 2004). Evaluations could be based on cost or performance.

With regard to the cost issues, both the initial cost and annual operating costs are usually considered simultaneously (Grierson and Khajephour, 2002; Nielsen and Svendsen, 2002). In these cases, the initial capital cost at the time of building construction accounts for the cost of land, superstructure cost, facade costs and the cost of HVAC systems. While the annual operating costs incurred after completion of the building construction accounts for the cost of energy consumed, maintenance work done, property taxes etc. (Grierson and Khajephour, 2002).

On the other hand, in the view of design performance, each discipline has an individual design objective.

- For the architect, spatial configuration is a main consideration, which includes finding feasible locations and dimensions of a set of interrelated objects (Michalek et al., 2002a). Evaluations of this issue are related with not only common engineering objectives such as cost and performance, but also aesthetic and usability qualities of a layout (Michalek et al., 2002a; Medjdoub and Yannou, 2000). Jo and Gero (1997) presented two architectural layout objectives, one was interactive costs which were calculated based on the adjacency needs among two functional units, the other was the distance between two units' assigned locations.

- The structural engineers determine an optimum material distribution in a given design space (Middendorf, 2003). Minimizing the weight of structural members in design whilst ensuring the safety (e.g. lateral and vertical deflection constraints and axial load constraints) is a normal design objective,
regardless of reinforced concrete frames (Balling and Yao, 1997) or steel framework (Soegiarso and Adeli, 1997, Foley and Schinler, 2003). Apart from this objective, other objectives are required to be optimised, such as maximum stiffness, minimum displacement at specific structural points, maximum natural frequency of free vibration, maximum structural strain energy (Li et al., 1999; Kicinger et al., 2005).

- The main consideration of the building services engineers is to provide an energy efficient building and provide a comfortable internal environment for potential users. The indoor environment design includes the thermal, visual and aural subsystem (Manning, 1994). In broad terms, the criteria of assessing the building environmental design are grouped into two types of building, naturally ventilated and air-conditioned buildings. In the former, both percentage mean vote (PMV) and percentage of people dissatisfied (PPD) are usually used to measure thermal comfort (Fanger, 1970). Minimising energy consumption is a key part of the design of either type of building. The designer considers the design variables, such as size of HVAC, the lighting power density (Wright et al. 2002, Ghisi and Tinker, 2005).

In order to solve these multi-objective problems, three types of the approaches are used, including approaches with a priori articulation of preference (e.g. weighted factor), approaches with a posteriori articulation of preferences (e.g. physical programming), and approaches with no articulation of preferences (e.g. Pareto optimality). Detail studies about these approaches are presented in Section 5.2.

3.4.4 Multi-discipline Design

The different professionals that take charge of building design and comprise a ‘design team’ usually include architect, structural engineer, mechanical and electrical services engineer, construction engineer, and quantity surveyor (Grierson and Khajehpour, 1999). Each member has a range of responsibilities and interests, however to produce a good design solution that effectively satisfies cost, time and functionality constraints, the co-operative efforts of this group of professionals are essential (Mathew and Rafiq, 1994).
3.4.4.1 Architecture

The application of optimisation to architectural design is relatively new and requires a careful formulation of the problem. Besides some common optimisation features (i.e. discrete and continuous variables, multi-objective, local and global optimisation algorithm and non-smooth objective and constraints), some characteristics further inhibit the use of optimisation. These include (Al-Homoud, 2001; Medjdoub and Yannou, 2000; Michalek et al., 2002a,b):

- Architectural design is characterised by being difficult to quantify numerically, because it is specifically concerned with aesthetic and usability qualities of a layout which are generally difficult to model or quantify mathematically.

- Architectural design as a system is composed of a number of components or sub-systems (e.g. function room). An overall optimum solution of all building components is difficult. In practice, they make a series of separate decisions for each component or sub-system. Combination of these sub-systems sometimes results in complicated formulations of the problem and an inability to reach a true optimum.

- A big constrain when using optimisation in the conceptual design is short time frame. Hence the applications of optimisation must allow the designer to quickly generate high quality layouts and receive both visual and computational feedback.

The main tasks in architectural design are topology and geometry (Michalek et al., 2002a; Medjdoub and Yannou, 2000):

- Topology refers to logical relationships between layout components. It could be evaluated based on topological qualities, such as openness, proximity, directionality, or symmetry.

- Geometry refers to the position and size of each component in the layout.

In order to complete these tasks, researchers put forward different solutions. Traditionally, topological and geometrical problems have been implemented separately. The topological problem is often implemented using grammar whereas the
geometrical problem has been solved using mathematical programming or related optimisation.

Medjdoub and Yannou (2000) proposed an ARCHiPLAN model with two solution levels: topological and geometrical, which is close to the methodology used in architectural design. They used a technique of first enumerating all topologies, the designer is then able to review the feasible topological possibilities and select those that she/he wanted to explore geometrically. This technique reduces computation dramatically, and they have shown success for up to twenty rooms.

Michalek et al., (2002a) developed a mathematical model for the geometric decision in the layout problem that allows an efficient solution with gradient-based and hybrid SA/SQP methods. This model was then embedded into another model used for topology decision that was solved with heuristic global methods. The difference with the model developed by Medjdoub and Yannou (2002) is that each iteration of topology model needs to call the geometry model. In the geometry model, there are four types of units based on their function, namely room, boundaries, hallways and access ways. Whilst ten variables for each unit include a referent point location in the coordinates, distances to north, south, east and west walls respectively, and the size of any windows on each wall. Using these variables the user can formulate design objectives and constraints, for example Unit i must be inside Unit j, while Unit K must not be adjacent to Unit j. However this study did not consider structural elements and route of pipe and duct when sizing these units.

Jo and Gero (1998) proposed the GA formulation that was only used in the field of unit topology problem while the size and function of each unit had been decided in advance. The problem configuration and the evaluation criteria were drawn from the Liggett’s approach (1981). The main contribution of this study was to present a design rule in the form of gene schema which can be manipulated in the genetic search engine. It also demonstrated the coupling of a GA technique with a design process could produce very good results, especially for large scale problems which was computationally difficult.
3.4.4.2 Structural Engineering

The problems addressed by structural optimisation can be divided into three major categories (Kicinger et al., 2005):

- Topology (layout) optimisation also known as topological optimum design (TOD) – looking for an optimal material layout of an engineering problem;
- Shape optimisation (SO) – seeking optimal contours, or shape, of a structural system whose topology is fixed;
- Sizing optimisation – searching for optimal cross-sections, or dimensions, of an element of a structural system whose topology and shape are fixed.

The three categories are closely related to three major stages of engineering design, i.e. TOD is conducted in the conceptual design stage, SO in the preliminary design stage, and finally sizing optimisation is performed in the detailed design stage.

Most previous applications are related to sizing optimisation in a simple structural system. Isenberg et al., (2002) considered the cross-section area and bending moment as design variables with the optimisation goal of minimising the weight of the structure. In this study, the objective function was linear; the optimal solution should lie on a constraint boundary or intersection of boundaries, so an interactive numerical optimisation procedure was proposed to reduce the weight of the structure until the most stringent constraint is satisfied with equality. The partial derivatives of the constraint functions (i.e. total frame drift) with respect to design variables were necessary to be calculated in this algorithm. However the standard cross sections come in a limited variety and the closest to a given one will usually have different values. This is why a number of researchers tended to use stochastic algorithm for optimising dimensions of cross-sections, such as Pezeshk et al., (2000); Coello et al., (1994); Hajela (1992); Goldberg and Samtani (1986).

Compared to optimal sizing member, topology optimisation is a more of a complex structural design problem. Kicinger and Arciszewki (2004) employed a simple GA based on aggregating functions for multi-objective problems. It is to identify the configuration of wind bracing, beam and column supports in a 36-storey building design. One important stage using GA is to enumerate all possible values of every
variable and code them in genotypic values. Figure 3.4 graphically presents all values of wind bracing for each structural grid in this study. The figure also helps readers to understand what the topology structural optimisation focuses on.

![Figure 3.4 Value of Wind Bracing](Kicinger and Arciszewski, 2004)

The formal optimisation of reinforced concrete frameworks is more challenging than the optimisation of steel frame because of the complexity associated with reinforcement design. Balling and Yao (1997) put forward a multi-level framework to minimise the total cost (i.e. material, fabrication and placement cost) with respect to cross-section dimension, reinforcement topology, bar selection, bar position and so on. This multilevel frame includes system optimisation (SO) and individual member optimisation (IMO). The SO level considered values for cross-section dimension and called analysis programs to determine deflection and drift constraints, it then sent values of the section dimension and internal-force (axial force, shear force and bending moment) as constant parameters to IMO. IMO exploited an exhaustive search to choose a topology and quantity of steel bars. This two-level framework can fully optimise reinforcement details.

### 3.4.4.3 Building Services Engineering

In the field of building services design, most researchers focus on the building thermal optimisation and HVAC plants optimisation

- Building thermal performance

D'Cruz et al., (1983) listed the factors affecting thermal performance, which are building shape, building mass, building orientation, window size, glass type, shading, surface finishes, material properties, ventilation and infiltration. Bouchlaghem and Letherman (1990) undertook thermal optimisation research to minimise the degree of
discomfort which is a function of indoor environmental temperature by using a combination of simplex and non-random complex optimisation algorithms.

Miles et al., (2001) used GAs to aid the design in the early stage of building design to reduce conceptual complexity for the designer. The structural form of the building is considered in conjunction with environmental impact and integrated with the services strategy.

Caldas and Norford (2002) utilised the principle of GAs. The authors optimise the annual energy cost by assessing the thermal and lighting performance using DOE2, which is an unbiased computer program to predict the hourly energy use and energy cost of a building and gives hourly weather information. This information is then used to optimise the placing and size of windows within a building. The optimisation algorithm is modified to minimise the computational complexity of the problem.

- HVAC plants optimisation

The ultimate aim of HVAC plants optimisation is to minimise the operational and installation cost of the system, satisfying requirements of comfortable indoor environment. Wright and Hanby (1987) identified that the kind of optimisation had three main elements:

1) The identification of a number of possible system configurations;
2) For each configuration the optimisation of the size of the system components;
3) The assessment of the system performance by the selection of criteria values to be used as quantitative parameters, thus enabling the selection of an optimum system.

Wright et al., (2002) adopted Pareto-based genetic algorithms to examine the relationship to the optimum sizing of HVAC system (i.e. supply air temperature and flow rate), simultaneously with the optimisation of its supervisory control strategy in order to identify the pay-off between the system energy use and occupant comfort. The final Pareto curves illustrate the trade-off energy cost against room thermal
comfort (i.e. Percentage of People Dissatisfied) in winter, summer and swing design day.

However the parameters related with structural design are usually fixed in the applications of thermal and HVAC optimisation design, for example, Wright *et al.*, (2002) assumed the material and dimension of external and internal walls, floors and ceiling had been decided before optimisation.

### 3.5 The Needs to Improve Building Design Optimisation

As complexity of building design and cost have increased, there is a need for enhancing the capability of building design optimisation. The following three major improvements are required (Papalambros, 2002; Rowell and Korte, 2003; Choudhary and Michalek, 2005):

- Improvements in building optimisation models, especially in the architectural design;
- Improvement in the methods that address the problems of multidisciplinary analysis;
- Improvement in the design of the system that coordinates the execution of the coupled engineering disciplines to reduce work load and design cycle time.

#### 3.5.1 Optimisation Formulations

Formulating a good optimisation model and choosing appropriate methods to solve the problem are crucial for finding meaningful solutions that meet the stated goals in an efficient manner (Choudhary and Michalek, 2005). Herein a good optimisation model requires experts to deduce the relationships between design variables and objective/constraint functions, and to develop realistic models that can be implemented in computer codes. A typical optimisation problem is expressed in Equation 3.1.

Careful model building is a pre-requisite in optimisation. A fair portion of this task depends on the modeller’s understandings of the problem and the model properties that will affect the solution process (Choudhary and Michalek, 2005). Most previous
studies of design optimisation demonstrate problems in a mathematical model which can be categorised into two groups. An optimization problem in which the objective and constraint functions are linear functions of their variables is referred to as a linear programming (LP) problem. On the other hand, if at least one of the objectives or constraint functions is nonlinear, then it is referred to as a nonlinear programming (NLP) problem.

However such mathematical models are limited for the following reasons:

- The problems are over-simplified in most cases in order to make them possible to solve using mathematical methods. The solution time for these methods tends to grow exponentially as the number of decision variables increases because of the combinatorial nature of the problem.

- The solution time may be highly variable, i.e. a small problem may take much more time than a large problem, minor variations in the problem may lead to a significant increase in solution time, and a single-solution algorithm does not work best for all types of mathematical problems (Chandra, 1991).

- Mathematical model have the disadvantage of being difficult and sometimes it is impractical to formulate the problem into a mathematical model. The non-numerical, an ill-defined with non-quantifiable criteria nature of many architectural problems have contributed to this difficulty (Al-Homoud, 2005).

- Most mathematical optimization applications are suited and developed for continuous design variables. In discrete optimization this problem becomes a difficult task.

- The chances of obtaining an optimum solution by mathematical programming method depend on the proximity of the initial solution provided for the problem. Often the decision-maker is unable to give a good solution to these problems.

GAs address most of the above problems (e.g. discrete variables and objective functions) and have been popular with diverse applications in the field of architecture, structural and building services design. A good feature of a model based on GA is that it does not require explicit expressions between dependent and independent variables.
For example, it is hard for the steel structural designers to present the complex formula between the universal cross-section of beams and their properties (i.e. elastic modulus, moment capacity). This can be achieved through coding a one-to-one relationship carefully in the GA process. However, this model is criticised because of the huge computation demand. This mainly stems from that it tries every design solution even if some solutions are not feasible.

By comparing to the above two approaches (i.e. mathematical and the GA-based model), each approach demonstrates certain advantages and disadvantages. Therefore the selection of an appropriate approach depends on the particular engineering design problem to be solved. For example, Bouchlaghem and Letherman (1990) dealt with the thermal design as a continuous variables optimisation problem with a nonlinear objective function and linear constrains on the variables in a mathematical model, but Wright et al., (2002) solved the design problem in the GA model and considered variables as discrete.

As a result, whether the optimisation module for a design problem is good or not, it does not only depend on the selection of mathematical, GA and other approaches, also on whether:

- The optimisation approach considers issues on the practical design (e.g. standard cross-section size in steel structural design);
- The optimisation approach is easy to implement in a computer environment (e.g. transferring design knowledge to computing code);
- The optimisation approach utilises available resources (e.g. computing resource and analysis software); and
- The optimisation approach meets the requirements of the design (e.g. design stage, design accuracy).

### 3.5.2 Building Simulation Integration (BSI)

The role of simulation tools in the building design has been firmly established over the last two decades. Simulation is credited with speeding up the design process, increasing efficiency, and enabling the comparison of a broader range of design
variants, leading to more optimal designs (Augenbroe, 2002). Simulation algorithms have historically been designed to predict answers for different performance areas such as thermal, HVAC or structural problems. These algorithms develop and evolve naturally within their discipline, and each discipline maintains an expertise in their operation, for example SPA2000 and ANSYS in the field of structural design, and Energy Plus and BLAST for thermal analysis. Furthermore, engineers working with different models may be situated in geographically dispersed locations. In order to predict the properties of design as a whole, the different models have to be interconnected. Therefore, in order to manage cross-functional teams, the integration of multi-domain simulation is necessary (Citherlet et al., 2001). Combining all design decisions and evaluating them simultaneously poses significant challenges to multidisciplinary engineering design (Augenbroe et al., 2004; Clarke, 2001). Therefore, many past researches and developments have dealt with this integration, and existing solutions can be classified into two groups as follows (Citherlet et al., 2001; Malkawi 2004):

1) A monolithic, synthesis approach

This approach presented in Figure 3.5 is the most basic solution for multiple-domain simulation; it allows compiling as a single executable code that contains internal modules or subroutines to accomplish each disciplinary analysis. Clark (1985) developed a simulation program that integrate thermal, ventilation, air quality, electrical power and lighting calculation. A monolithic synthesis code has the advantage of being fast and executable by a single designer but tends to exclude other-domain experts from the design process and can quickly become outdated without continued support and improvement. In addition, these monolithic codes can be difficult to extend or modify (Rowell and Korte, 2003).
2) A loosely integrated framework in which each disciplinary software uses a different programming language and need to exchange data with other software.

Such a framework is normally established using three approaches, namely model exchange, model sharing and coupled program.

The approach of model exchange presented on the left of Figure 3.6 allows to build discipline-based models first, and then transform the whole or part of the model by using a data exchange generally based on a standardised file format. The SFSAS system (Ren et al., 2007) is designed to integrate fire simulation with structural analysis by a kernel database. This core database has been developed to support the data store and exchange in the SFSAS system and bridge the connection between different models. In addition, the Initial Graphics Exchange Specification (IGES) is a neutral data format that allows the digital exchange of information among different simulation systems (Nagel et al., 1980).

The concept of model sharing presented in the middle of Figure 3.6 requires all design information to use a shared model and carry out the analysis separately. The COMBINE project (Augenbroe, 1995), a typical example, focuses on a data-exchange infrastructure via sharing building simulation models which formalise the input-
output relations of relevant performance aspects. Design information is automatically exchanged between different models through an interoperability layer, while the Building Design Advisor software sets a single, object-oriented building representation which describes the building (Papamichael et al., 1997).

However the previous two approaches are critical because a project practitioner would probably recognise that much time and effort is still required to locate, translate, enter, exchange and update data between different analysis models, and further information is constantly required (Citherlet et al., 2001). Under this condition, the coupled concept presented in the right of Figure 3.6 emerges, in which one discipline-based model controls the whole analysis process and calls other disciplinary analysis when necessary. ESP-r (Clarke, 2001) addresses the issue of integration by simultaneously processing mathematical models of building physics (i.e. thermal and ventilation) within a simulation. This model applies customised solvers to each disciplinary simulation. Integration is achieved by linking the outcome of one disciplinary simulation to the coefficients and source terms of the equation of another related discipline. The inconvenience of the coupled approach is the maintenance of data and link consistency which depend on the separate evolution of each coupled application and creating difficulties in any change or improvement (Citherlet et al., 2001).

![Figure 3.6 A Loose Integrated Framework for BSI (Citherlet et al., 2001)](image)

These existing solutions in simulation integration still have some limitations, which provide new possibilities to unattended gaps in this field. Therefore a rigorous and consistent framework that enables collaborative design, distributed computing
platforms, and the inclusion of high accuracy analysis solution is urgently required (Rowell and Korte, 2003).

### 3.5.3 Systematic Design

Building design optimisation always qualifies as a complex system due to large scale and strong interactions. The term of ‘large scale’ refers to the large number of design variables and members (e.g. participants or physical components). It is difficult to handle the entire set of design variables by one person or team. Whilst strong interaction between members exists if the state of the member affects how the system responds to change in another member. Improvements in one member alone are not sufficient to affect the improvements in products and processes needed in the future. System level synthesis, analysis, and optimisation tools will be required (NSF, 1996). The methods are advocated that can exploit interactions to achieve a better overall system than can be achieved by ignoring interactions. Decomposition-coordination is a valuable and necessary approach in solving the complex system (Papalambros, 2002; Al-Homoud, 2005; Choudhary and Michalek, 2005). A good decomposition should allow (Papalambros, 2002):

- improved coordination and communication;
- conceptual simplification of the design;
- modularity and parallel computation;
- simpler and more efficient computation procedure; and
- use of optimisation technique tailored to specific sub-problems.

Wagner (1993) proposed four types of decomposition schemes:

- **Sequential decomposition** is appropriate for flow processes. Chemical or manufacturing processes may be partitioned by making cuts in flow paths.
- **Matrix decomposition** is applied to large systems of mathematical equations.
- **Object decomposition** involves dividing a system by physical or functional component. For example, an automotive design may be partitioned by object into body, powertrain, and suspension subsystems.
• Aspect decomposition divides the system by discipline. For example, aeroplane design could be partitioned into structural, aerodynamic, and dynamics disciplines.

The key requirement of coordination strategy is that they converge to the same solution set as that of the un-decomposed problem (Papalambros, 2002). One kind of coordination approach is realised by gradient-based optimal sensitivity in the high level (Braun and Kroo, 1993), such as MDO formulation. The other kind is based on artificial intelligent technology, including artificial neural network (Chen et al., 2005) and agent technology (Anumba et al., 2003).

In fact, these decomposition-coordination methods have been used in building and infrastructure design in different ways.

1) Balling and Rawlings (2000) used a two-level optimisation framework in bridge design. In this case, bridge design was decomposed into a superstructure group and a deck group located at subsystem level. While the system level is responsible for the overall design objective of minimisation cost and co-ordination among subsystems. The coupling variables between the two groups originated from the interactive force between the cable and the deck of a typical suspension bridge. The system-level optimiser sent the target values of coupling variables to the corresponding groups; when the target value matched the actual value which satisfies the constrains at the subsystem level, it meant that a consistent overall design is found.

2) Choudhary et al. (2003) presented a multi-level optimisation framework for building performance analysis using an analytical target cascading (ATC) method. This framework was based on object decomposition where a complex healthcare facility was divided into six function zones. Every zone was then further partitioned into several rooms and cubicles hierarchically. Every part partitioned was regarded as a decision model which linked the relevant analysis model. Linking variables represented decisions shared between two or more decision models at the same level. The vertical relationships between decomposed levels were embodied through building performance targets. The
Chapter 3  

Building Design Optimisation

ATC framework coordinates multiple decision-making tasks in a compatible solution, provided explicit decision support including tradeoffs between performance goals and purposeful use of simulation tools in decision-making.

3) Pushkar et al. (2005) introduced a methodology for environmentally optimal building design based on the sequential decomposition that was usually used in the chemical and manufacturing industries. Through the proposed methodology, the entire design variables were divided into production and construction group, operational energy group, and maintenance to demolition group according to the extent of their environmental impacts. In such a manner, environmental optimisation could be performed within each group separately, and then the partial decisions are combined. Finally the overall environmentally optimal solution for the entire building would be obtained.

The major drawback of the above applications is that they only consider improving design based on the single discipline's view (e.g. building services or structural design). The design and optimisation of complex system poses a unique challenge: where physical components or disciplines must be designed simultaneously so that they are compatible and consistent with one another while delivering properties that, in combination, achieve the objective of the overall system (Design Decisions Laboratory, 2007).

The methodology of multidisciplinary design optimisation (MDO) based on the concept of decomposition-coordination has been successfully applied in the aircraft and automotive industry and resulted in more reliable and better products, it is rarely used in the building design. Therefore the application the MDO in construction can be an important research direction to improve overall performance of building design.

3.6 Industry Requirements

In order to confirm the findings from previous literature review (e.g. approaches of coordinating design conflicts, characteristics of building design), semi-structured interviews were conducted. Questions in these interviews are related to the characteristics and conventional practices in contemporary building design,
approaches of dealing with multidisciplinary design and their drawbacks. Analyses of data obtained from this interviews are presented below.

### 3.6.1 Interviewee's Background

From the response to the two questions regarding the interviewees’ industrial backgrounds, it can be seen that the interviewees held a wide variety of roles in industry as shown in Table 3.2. Some participants often had experience in more than one discipline.

<table>
<thead>
<tr>
<th>Architect</th>
<th>Structural engineer</th>
<th>HVAC engineer</th>
<th>Academic researcher</th>
<th>Specialised software developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Their work experience in industry ranges from 3 to 21 years with an average of 10 years. Ninety percent of the interviewees have more than five years working experience. The description of the positions held in their organisations produces a wide variety of answers including architect, structural engineer, HVAC engineer, researcher and professional software developer. Furthermore, sixty percent of interviewees played more than one roles during their career life, such as designers and contractor, designer and researcher, client and contractor. The interviewees' experiences demonstrate that they are qualified for this interview. Experts in these interviews merely focus on design participant, but future studies could include other stakeholders, such as quantity surveyor, client, facility managers, contractor and material supplier.

### 3.6.2 Findings from the Interviews

All interviewee’s opinion about these questions included in Appendix Two are shown in Table 3.3 to Table 3.7.

**Question 3**

The interviewees were asked to identify the characteristics of building design. Here, three characteristics (i.e. Statement 1-3 in Table 3.3) were given, namely multi-variable, multi-objective and multidisciplinary. This question also asked them to select the most obvious characteristic.
Results:

Table 3.3 Response to Characteristics of Building Design

<table>
<thead>
<tr>
<th>Statement</th>
<th>Percentage of respondent (%)</th>
<th>Number of respondents who was regarded this as the most obvious characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 A large amount of parameters to be considered in design</td>
<td>70%</td>
<td>1</td>
</tr>
<tr>
<td>2 Several objectives to be achieved such as functional requirements,</td>
<td>70%</td>
<td>2</td>
</tr>
<tr>
<td>economic requirements, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 A number of design disciplines involved such as structural, electrical,</td>
<td>90%</td>
<td>6</td>
</tr>
<tr>
<td>mechanical design and other experts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As Table 3.3 shows, the response to Statement 3 received the highest (90%), whilst 6 out of 10 interviewees reflected that the most obvious feature of building design was multidisciplinarity. Although these interviewees were involved different kinds of projects, such as electrical power factories, water reservoirs, nuclear power stations and buildings, the same collaborative working between disciplinary designs, such as geotechnical, hydraulic electrical, mechanical, process, building services and civil engineer exists. Apart from the characteristic of multidisciplinarity, the interviewees also stated that the multi-objective nature of building design is another important characteristic. For example the high value is always the main consideration in a current construction projects, hence designers need to minimise capital and running cost, optimising building performance, and minimise impacts on the environment.

Besides the characteristics of building design shown in Table 3.3, the interviewees indicated others as follow:

- Client requirements. Building design is a client-based consultancy service, thus clarification of client requirements is a key factor in the success in design.
- Time constraints: Building designers are usually bound by tight time frames,
• Low innovation: designs in the construction projects often lag behind those of other industries in terms of adoption of innovative approaches.

• Intangible factors: It is difficult for identification, analysis and translation of explicit and implicit design requirements into design specification. For example client requirements combine with site, environmental and regulation requirements, which are difficult to trace and correlate design decision.

Question 4
This question reviewed conventional building design practices; interviewees were required to determine a level of agreement from 1 to 5 on each statement in Table 3.4.

Results:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Ranking</th>
<th>Mean Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Sequential independent disciplinary design plus system level review</td>
<td>3 1 4 2</td>
<td>3.5</td>
</tr>
<tr>
<td>5 Design limited to each discipline (mature design approach, analysis tool, design code)</td>
<td>1 4 5</td>
<td>3.9</td>
</tr>
<tr>
<td>6 Resolution of interdisciplinary conflicts non-automated</td>
<td>5 5</td>
<td>4.5</td>
</tr>
<tr>
<td>7 Relying heavily on previous experience</td>
<td>2 2 3 3</td>
<td>3.7</td>
</tr>
</tbody>
</table>

Data obtained from Question 4 (Table 3.4) indicated that the range of responses were fairly low for Statements 4 and 7. Evaluators were further asked the reasons for these statements. They were:

• Most of interviewees confirmed that design tasks were based on each discipline, but the process was concurrent with some overlaps between disciplines rather than being sequential. Regarding these overlaps, the system level review was necessary to deal with conflicts between disciplines and obtain a compatible design solution.

• Interviewees stated that experience was really important for design; it was helpful to identify the design schemes and decide design parameters quickly. However in recent years, some design support systems emerged, such as
expert and knowledge-based systems; consequently designers reduced the extent of dependence on experience. In addition, previous design experience is usually challenged by some new problems in the new and complex construction projects.

Question 5

This question provided three approaches for coordinating conflicts between disciplines. The interviewees were asked to respond to the frequency of use of these approaches by ticking their opinion on a five-point scale. In addition interviewees were also allowed to present other approaches used according to their experience.

Results:

The scores were assigned to the ratings frequently=5, most times=4, sometimes=3, rarely=2, never=1. Table 3.5 shows the results, a simple arithmetic equation was applied to calculate the mean ranking.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Frequency</th>
<th>Mean Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Chief designer (e.g. architect) proposes a resolution for coordinating different disciplinary design based on experience.</td>
<td>Never</td>
<td>Rarely</td>
</tr>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9 Negotiation amongst other designers in design meetings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Using some parameters based on building design regulations, for example a structural designer may leave some holes for water pipes through floor in advance.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>Internet or Intranet within the scope of the company</td>
<td>Specialist commercial software: such as Revit architecture</td>
</tr>
</tbody>
</table>
Results in Table 3.5 demonstrated:

- Face-to-face meetings were the most common approach to solve conflicts between disciplines because it could be used at any stage of the design and provides designers with the chance to explain clearly their requirements, but interviewees also indicated some drawbacks of this approach. For example in some cases it was hard to ensure all designers to attend such a meeting simultaneously, this approach could also be expensive for designers who were located in different places.

- Using a chief designer was ranked second in the three approaches. One reason for this result was that informal discussions between disciplines were the first choice to coordinate conflicts, if they couldn’t reach agreements, then they would ask for help from the high-level chief designers. However problems sometimes arise in complex projects that might be beyond the chief designers’ capabilities.

- The lowest mean ranking was received for the approach of using design code. Most interviewees stated that the code merely provided values of design variables to satisfy legal requirements in single aspects of design performance (i.e. deflection), this did often not meet specific design requirements.

Interviewees also put forward other approaches in addition to those listed in Table 3.5, they are:

- Internet or Intranet within the company. Interviewees reflected that this approach was often used to transfer drawing files, leave messages about design changes and consult with other participants through e-conferences.

- Specialist software. One interviewee said that a commercial software called Revit architecture was used in his company to automatically coordinate any changes made by other designers.

Question 6:

This question was open-ended and asked interviewees to identify constraints on the application of optimisation in building design.
Results:

There was a wide spectrum of responses to this question with the interviewees giving a number of issues and constraints on the application of optimisation in building design. By far, the most common opinion was the lack of drives for the application of optimisation. The main reason was that optimisation often required additional iterative work between disciplines, this resulted in increasing time and cost, and hence designers were not often willing to conduct design optimisation.

Another major barrier to the adoption of optimisation is the difficulties in formulating an optimisation model. Interviewees, particularly architects, indicated that some qualitative aspect of building design were hard to be formulated in optimisation models, such as the aesthetics of a building and flexible space planning design. However most interviewees expressed that they were still interested in the search capabilities provided by optimisation techniques.

Question 7

This question was multiple-choice, the interviewees were ask to state their expectation of functions in a multidisciplinary design model, four options (Statement 11 to 14 in Table 3.6) were given, the interviewees were also encouraged to include other functions that they required.

Results:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Percentage of respondent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 Integrating different specialised software within one model</td>
<td>70%</td>
</tr>
<tr>
<td>12 Automatically resolving conflicts between disciplines.</td>
<td>90%</td>
</tr>
<tr>
<td>13 Provide as many feasible design solutions as possible</td>
<td>60%</td>
</tr>
<tr>
<td>14 Considering various objectives concurrently</td>
<td>80%</td>
</tr>
<tr>
<td>Other:</td>
<td></td>
</tr>
<tr>
<td>• Informing any changes from other designers quickly</td>
<td></td>
</tr>
</tbody>
</table>

Based on results in Tables 3.6, a few points need to be discussed:
• Firstly within the four potential functions, most interviewees suggested that it was great if coordination of conflict between disciplines (i.e. Statement 12) could be accomplished by the proposed model, because the current approaches (i.e. face-to-face meetings and chief designer) were time-consuming and have limited scope in the application.

• Secondly interviewees maintained that exploring different alternative designs in order to find the most suitable solution (i.e. Statement 13) was a good function at the beginning of every project. However it brought difficulties in making decisions if too many solutions were provided. For example an interviewee indicated that up to ten different structures, girds and material might be considered during the conceptual design stage for a project, whilst 3-4 design schemes on average were provided to clients by architects.

• Thirdly, in the response to Statement 11 interviewees gave a relatively low agreement (70%). Because the interviewees suggested that the use of specialised knowledge should not be limited to analysis software, some lessons and experiences from previous projects also should support the design models.

3.7 Summary
In this chapter, the characteristics and applications of building design optimisation are studied through a comprehensive literature review and industry interviews. The optimisation in building design is more complex due to the large number of variables, the various types of variables involved (e.g. continuous and discrete), conflicting building performance requirements (e.g. safety and aesthetic) and close interactions between disciplines. Concentrating on these difficulties, the selection of an appropriate optimisation algorithm is critical for success in building design. Thus existing optimisation algorithms are discussed under the gradient-based and derivative-free groups. The main algorithms in each group are described in the terms of main features, application scope, terminal criteria, advantages and disadvantages. These algorithms include the Newton method, SQP, the simplex method, the Hooke-Jeeves method, SA and GA. Their distinguishing properties are helpful for choosing the appropriate optimisation for implementing the proposed framework in the later
stages. Most important in this chapter are the research priorities in the field of building design optimisation are presented; these are the development of optimisation models, building simulation integration, and system design optimisation. These challenges encourage the application of multidisciplinary design optimisation to achieve improvements in building design.
CHAPTER FOUR: MULTIDISCIPLINARY DESIGN OPTIMISATION

4.1 Introduction
This chapter reviews multidisciplinary design optimisation (MDO) in general. Firstly, the background and concept of MDO and its definitions are presented, then benefits obtained from the application of MDO are summarised. Problems with the use MDO are also explained in this chapter. After analysing these problems, the focus of this research is on the MDO formulation. The basic elements of the MDO problem and formulations are then presented. Two of these MDO formulations (collaborative optimisation, and analytic target cascading) are studied in depth in terms of their main features, advantages and mathematical formulations. Based on comparisons between the two formulations and the characteristics of building design optimisation presented in the last chapter, the collaborative optimisation formulation is selected for solving complex building design.

4.2 Overview of MDO
In 1991 the American Institute of Aeronautics and Astronautics (AIAA) used research and development as a means to help engineers master interdisciplinary couplings and to enable them to exploit the associated synergism, towards improving efficiency and effectiveness of the design process and obtaining better quality of the final product. Because an aerospace vehicle is an engineering system whose performance depends on interactions of many disciplines or parts and whose behaviour is governed by a very large set of coupled equations. In practice, engineers deal with these equations by partitioning them into subsets corresponding to the major disciplines. In this process of pragmatic partitioning, the couplings among the subsets tend to be increased in number so it is burdensome to account strictly for them all (AIAA, 1991). Under these conditions, the concept of multidisciplinary design optimisation (MDO) emerged; it is defined (AIAA, 1991):

"A methodology for the optimal design of complex engineering system and sub-system that coherently exploits the synergism of mutually interacting phenomena using high fidelity analysis with formal optimisation."
In order to explore further understanding of this definition, two key questions arise.

1) What is MDO?
MDO is a methodology to enable design, not a design. It provides a collection of tools and methods that permit the trade-off between different disciplines involved in the design process (Giesing and Barthelemy, 1998). This methodology broadly covers the human interface, computing and optimisation aspect of design. Specifically speaking:

- The MDO methodology discards the 'push button design' idea in favour of a realistic approach that recognises the role of the human mind as the leading force in the design process and the role of mathematics and computers as indispensable tools (AIAA, 1991).
- The MDO methodology should enable the use of multiprocessor/distributed/parallel computing for executing a method that originated in a serial computer environment (AIAA, 1991).
- The MDO methodology mathematically traces a path in the design space from the initial to improved designs and it operates a large number of variables and functions simultaneously (AIAA, 1991).

2) What problems does MDO solve?
The MDO methodology mainly focuses on the complex engineering system design. A system may be qualified as complex due to its large scale (i.e. large number of disciplines or variables), and due to strong interactions between disciplines. The analysis of a complex system as an undivided whole is usually found to be inefficient and intractable. An alternative strategy is to partition the system into small sub-systems; these sub-systems (Wagner, 1993) could be functional components, disciplines, design stages or analysis models. Considering the sub-systems individually and their interactions may render the system task feasible and more efficient. However these interactions are more difficult than the constituent members. The co-ordination strategy for sub-system interactions is a key element in an MDO problem (Sobieszczanski-Sobieski and Haftka, 1997).

Therefore, MDO is an appropriate methodology, especially for complex engineering systems that are distributed, governed by mutually interacting physical phenomena, made up of distinct interacting sub-systems (Sobieszczanski-Sobieski, 1993).
Furthermore, it is a concurrent engineering design tool for a large-scale, complex system design that can be affected by the design of several smaller functional units or sub-systems (McAllister et al., 2005).

Since the MDO methodology was put forward, researchers in various engineering industries (e.g. aerospace and automotive) have invested a lot of effort in exploring its applications, such as in aircraft design (Kroo et al., 1994), aircraft engine design (Tappeta et al., 1999), aircraft wing design (Jun et al., 2004) underwater vehicle design (Belegundu et al., 2000), ship design (Campana et al., 2005) and race car design (McAllister et al., 2005). From these applications, the MDO methodology demonstrates its particular advantages (Willcox, 2004):

- Systematic, logical design procedure: MDO provides an effective coordination strategy between disciplines in design/optimisation procedures with consideration of the system objective and individual objectives (Kroo et al., 1994; Giesing and Barthelemy, 1998; Balling and Sobieszczanski-Sobieski, 1996);

- Increased number of design variables & constraints to be handled: Design variables will increase significantly if a new discipline is added in the design. A decomposition scheme allows each discipline to only consider those discipline-specific variables. This improves the overall system’s capability in handling variables, compared with the traditional sequential optimisation;

- Not bias by intuition or experience. In the process of MDO, every discipline optimiser is able to obtain the result from corresponding analysis module. Unlike in the past, designers heavily rely on previous experience to determine appropriate value for design parameters (Wang et al., 2002; Jedrzejuk and Marks, 2002); and

- Reduction in design time. An MDO environment facilitates effective communication between disciplines so that the design cycle time will be decreased (Sobieszczanski-Sobieski and Haftka, 1997).

Like other industries, the construction industry deals with complex building design; design firms are typically grouped by discipline (e.g. architecture, structure, building
services). However, the potential benefits of the MDO methodology have not been realised in the industry, this research attempts to develop a MDO framework to support collaborative building design.

4.3 Challenges to the Application of MDO in Engineering Design
Engineering design has been taking advantage of the application of MDO. However, these applications have also highlighted some of the generic problems and challenges. The following section explains three main problems, namely MDO formulation, computing requirements, and data exchange.

4.3.1. MDO Formulation
Typical elements in the MDO formulation include problem analysis, sensitivity analysis, approximations, design space search algorithms, decompositions, etc. Among these elements, the accurate formulation of couplings between disciplines often has a significant impact on both convergence and quality of system design performance (Alexandrov and Hussaini, 1997).

Hence specific attention has to be paid in order to ensure the correctness of the MDO formulation. In the last decade, several MDO formulations were developed. Single-level formulations, such as the All-at-One (AAO) method, and the Individual Discipline Feasible (IDF) method (Cramer et al., 1994), have been established and widely applied in the aerospace industry, but they are confined to only small and conceptual level problems (Brown, 2004). New multi-level MDO formulations, such as Concurrent Subspace Optimization (Sobieszczanski-Sobieski, 1988), Two-Level Integrated System Synthesis (Sobieszczanski-Sobieski, et al., 1998), Analytical Target Cascading (Michelena et al., 1999) and Collaborative Optimization (Braun and Kroo, 1995) have been proposed for the global optimization of large and complex systems (definitions of single and multi-level formulation refers to Section 4.4).

Current practice relies upon the engineer’s insight to understand whether the system is single-level, multi-level or hybrid and to choose an appropriate formulation scheme (Logan, 1990). For a large system, this decision is difficult and can be ad hoc. Giesing and Barthelemy (1998) pointed out that the optimisation formulation must be re-
configurable and tailored to the specific problem encountered, and the integration and synthesis approach for multidisciplinary problem formulation are much more important.

4.3.2 Computing Requirements

The interdisciplinary couplings inherent into MDO tend to present computing requirement challenges in three aspects. Firstly, most problems faced by MDO are about complex system design. Such a complex problem results in a large amount of variables being handled due to many disciplines or physical components involved. Secondly, in order to take advantage of specialised knowledge and mature analysis tools, the decomposition of large system design is necessary in line with the discipline. This is bound to add extra variables to coordinate conflicts between sub-systems. The increase in the number of design variables caused from the above two reasons results in an increase in computational cost for optimisation. Finally, most analytical methods (e.g. computational fluid dynamics and finite element analysis) used by an optimiser usually require high levels of computer effort. Hence high-performance computing is needed to reduce analysis time to a reasonable level.

In response to the requirements of computing resources, many researchers (Giesing and Barthelemy, 1998; Sobieszczanski-Sobieski and Haftka, 1997) demonstrate that both distributed computing technologies (e.g. Parallel Computing, Grid Computing) and approximate models (e.g. neural networks based response surface and sensitivity based Taylor series linearisation) are two possible solutions. The distributed computing technology is a necessity for the future to generate enough computing power to perform optimisation that requires complex analysis and to drive multi site operations. Whilst the efficient incorporation of approximation models in the MDO methodology can help reduce the computational expense for three reasons (Koch et al., 1998). Firstly they provide a natural way of implementing coarse-gained parallelisation. Secondly they are computationally inexpensive to evaluate and therefore avoid potential numerical difficulties. Thirdly they often use high accuracy analysis with inherently smooth models (Sobieski and Kroo, 2000).
4.3.3 Data Exchange

MDO promises to obtain reliable design solutions through linking analysis tools for performance analysis (Sobieszczanski-Sobieski and Haftka, 1997). The analysis data for each discipline have to be made to interact with one another for the purpose of system analysis and system optimisation. The transfer of data from the output of one disciplinary software to another is rarely automated and frequently requires few man-hours of processing by one or more persons. On the other hand, in the multi-level formulation data transfer also exists between system and sub-system level. Hence a computing framework should be provided for efficient transfer, storage and access of data (Kodiyalam and Sobieszczanski-Sobieski, 2001)

With regard to the problem of data exchange, Kodiyalam and Sobieszczanski-Sobieski (2001) provided a database through structured query language interface for data storage/access/manipulation. Amitay et al., (2003) proposed a building block for analysis wrapping which was the process of integration of analysis models by modifying the source or writing programs to handle input/output files.

In addition to the above three problems, researchers like Wakayama and Kroo (1998), Bennett (1998), Young et al. (1998), Hoenlinger et al. (1998) also identified some other problems with the application of MDO such as:

- **Organisation structure**: Organisation is formed along disciplinary lines where each technology group is responsible for maintaining technical excellence. However, no one is in charge of the overall MDO process. An improved organization would benefit from the use of the MDO methodology.

- **Training**: The lack of familiarity with the processes associated with MDO limits its use. Therefore it is necessary to offer MDO-oriented training.

- **Visualization**: It is important for the designer to have user-friendly processes for displaying the design space and interpreting the result of the optimisation.

Although all these problems influence the application of MDO in engineering design, this research only focuses on the study of the MDO formulations. This is because of:
• the importance of the MDO formulation: Compared with the other issues of the MDO application (e.g. high-performance computing resource and effective data exchange), the selection of an appropriate MDO formulation has significant impact on the MDO implementation because these issues depend fully on the formulation of the problem. These issues are often difficult to solve before the establishment of the MDO formulation (Alexandrov and Lewis, 2004).

• the lack of effective MDO formulation: Although several MDO formulations have been applied in the aerospace and automotive industries, it is still in the early stages within the construction industry. The MDO formulation depends on the design methods and analysis tools, and the choice of decomposition schemes; these features must be decided through consideration of the particular processes, characteristics and difficulties of building design.

4.4 Classification of the MDO formulation

As the last section explains, due to the extreme complexity of most MDO problems, it is necessary to focus on problem formulation and its interdependence with the programming algorithm. Above all, it is important to clearly distinguish between formulation and algorithm. Formulation means expressing the problem as a set of mathematical statements amenable to solution, while algorithm means defining a procedure for solving the problem once the problem has been defined (Cramer et al., 1994). An analysis of an MDO formulation considers such attributes as consistency, well-posed, equivalence to other formulations and optimality conditions. An analysis of an optimisation algorithm for solving a given formulation of an MDO problem should consider local convergence, global convergence properties and iteration cost (Alexandrov and Lewis, 1999). The optimisation algorithms available were introduced in Section 3.2. This section focuses on reviewing the fundamental MDO formulations that emerged.
4.4.1 Main Elements of the MDO Problem

MDO as an effective methodology has been gaining popularity and was being increasingly used to design complex engineering systems over the past decade. For instance, an aircraft wing is designed for aircraft performance, aerodynamics and structure discipline (Sobieszczanski-Sobieski and Kroo, 1996); vehicle design was to consider powertrain, chassis and body component (Kim et al., 2002). In spite of multi-disciplinary or multi-component design, they demonstrate a number of basic features of the MDO problem. Herein for illustrative purposes a two-disciplinary system is used to explore the main elements of the MDO problem. In order to help readers understand these notations, Table 4.1 identifies how these variables and functions apply to a building design example. The reader is encouraged to refer back to this table to put an instance of a physical meaning on variables, functions and there terms are also used in the description of the fundamental MDO formulations.

![Two-disciplinary coupled system](image)

**Figure 4.1. Two-disciplinary coupled system**

It is assumed that all analytical models in the system are run within the disciplines. In Figure 4.1 all inputs and outputs of each discipline are vectors and are defined in the nomenclature

- **Design objective (\(f_1, f_2\)):** Design objectives normally minimise cost and maximise benefit and, therefore, may be competing.
- **Local variable \((x_1, x_2)\):** These variables are mutually exclusive sets of design variables. They represent the independent inputs that distinguish one design from another.
- **Shared variable \((x_s)\):** This vector of variables is needed by more than one discipline.
• Coupling variable \((y_{12}, y_{21})\): where \(y_u\) contains those functions computed in discipline \(i\) that are needed as input to discipline \(j\). These coupling variables complicate the order of execution of the disciplines and are responsible for iterations among disciplines.

• Discipline-specific variable. This kind of variable represents all variables that are required for one disciplinary. For example, discipline-specific variables for discipline 1 include \(x, x_1, s_1\) and \(y_{21}\).

• Design inequality constraint \((g_1, g_2)\): The design constraints normally guard against failure and otherwise unacceptable behaviour.

• Design equality constraint \((h, h_2)\). These constraints that are state equations may be equations of equilibrium, compatibility, constitution and conservation. Note the values of state variables (i.e. \(s_1, s_2\)) are used to solve these state equations.

### Table 4.1 Building Multidisciplinary Optimisation Problem: Examples of Variables and Design Functions

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_{12})</td>
<td>Depth of floor, wall and roof</td>
</tr>
<tr>
<td>(y_{21})</td>
<td>Position and weight of water tank</td>
</tr>
<tr>
<td>(x_1)</td>
<td>Window size</td>
</tr>
<tr>
<td>(x_1)</td>
<td>Shape of cross-section of beam</td>
</tr>
<tr>
<td>(x_2)</td>
<td>Orientation of window</td>
</tr>
<tr>
<td>(g_1)</td>
<td>Stress and displacement limits</td>
</tr>
<tr>
<td>(g_2)</td>
<td>Air velocity limit in the duct</td>
</tr>
<tr>
<td>(f_1)</td>
<td>Minimum weights of structural components</td>
</tr>
<tr>
<td>(f_2)</td>
<td>Minimum dissatisfaction (thermal comfort indices)</td>
</tr>
</tbody>
</table>

Notes: Discipline 1 stands for structure design, Discipline 2 stands for HVAC design

#### 4.4.2 The MDO Formulation

The above standard terms will be used throughout this thesis to present the MDO formulation. Now the classification of approaches for formulating and solving MDO problems are discussed. These approaches include All-at-Once (AAO), Individual Discipline Feasible (IDF), Concurrent Subspace Optimization (CSSO) using the Global Sensitivity Equations (GSE), Collaborative Optimization (CO), and Bi-Level Integrated System Synthesis (BLISS).
There are different schemes of classification. Cramer et al. (1994) argued that the key distinguishing feature in alternative MDO formulations was how to maintain discipline design feasibility at every objective function, constraint or sensitivity needed during each optimisation iteration. The criterion for understanding the difference between them proposed by Alexandrov and Lewis (1999) was based on the way that a formulation handles the explicit or implicit constraints, including disciplinary analysis constraints, design constraints and interdisciplinary consistency constraints.

However this research follows Balling and Sobieszczanski-Sobieski’s (1996) classification, in which the MDO formulations could be categorised into two groups. One group is called single-level optimisation formulation. In this group, optimisation is performed only at the system level, and the role of the discipline is limited to analysis and function evaluation. The other group is called multi-level optimisation formulation. In this group, the discipline-specific variables are determined by disciplinary optimisers while the system design variables are determined by the system optimiser. Herein concepts of disciplinary analyser and evaluator have to be explained. The disciplinary analysers seek values for the state variables that reduce the residual (i.e. $r_1, r_2$) in the state equations to zero. The disciplinary evaluators evaluate the residuals in the state equations for given values of the state variables. Obviously the computational effort required by a disciplinary evaluator is significantly less than that required by a disciplinary analyser (Balling and Sobieszczanski-sobieski, 1996).

In addition Balling and Sobieszczanski-sobieski (1996) introduced the system used to concisely describe the structure of an MDO formulation. These notations are:

- **SO** - System Optimiser
- **SA** - System Analyser
- $O_i$ - Disciplinary i Optimiser
- $A_i$ - Disciplinary i Analyser
- $E_i$ - Disciplinary i Evaluator
- [ ] - Nested execution
- $\parallel$ - Parallel execution
- $\rightarrow$ - Sequential execution
Some typical MDO formulations are briefly described based on the single and multi-level group in Table 4.2. This table also illustrates the structure of each formulation using Balling’s notations and symbols.
Table 4.2 Summaries of the MDO Formulations

<table>
<thead>
<tr>
<th>Classification</th>
<th>Existing MDO formulation</th>
<th>Description</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-level Formulation</td>
<td>Multi-Disciplinary Feasible (MDF)</td>
<td>The system optimiser supplies the system analyser with shared, local and state variables. The system analyser coordinates all of the disciplinary analyser. In addition, a system analyser is called at each iteration of the system optimiser to determine the value of the coupling variables. Hence the MDF is completely non-hierarchic in nature. There are no restrictions on data communication between the sub-systems. In a purely computation text, this formulation is desirable if the sub-systems are weakly coupled. The Balling’s notation is $SO[SA][A_1 \rightarrow A_2]$</td>
<td>![Diagram of MDF]</td>
</tr>
<tr>
<td></td>
<td>Individual Disciplinary Feasible (IDF)</td>
<td>The system optimiser in the IDF gains additional responsibility for the solution process over the MDF approach. The system optimiser must also control the values for the coupling variables. Thus the IDF has improved robustness over MDF, since the sub-systems must no longer wait for the result of other analyses before commencing their own analysis, all of the sub-systems can be evaluated in parallel. However the dimension of the optimisation problem is increased and the data communication requirements are higher. The Balling’s notation is $SO[A_1][A_2]$</td>
<td>![Diagram of IDF]</td>
</tr>
</tbody>
</table>
All-at-Once (AAO)  

AAO is a highly centralized approach. Instead of utilising analysers to complete the analysis for each sub-system, evaluators are used that compute only the residuals of the state equations. The system optimiser is now saddled with three sets of decision variables: the original design variables \((x_1, x_2, x_3)\), the coupling variables \((y_{12}, y_{21})\), and the state variables \((s_1, s_2)\). This high degree of centralisation offers impressive efficiency in some situations, yet it is sometimes difficult to map too many organisational structures due to its centralisation and specialised structure.

The Balling's notation is \(SO[E1\|E2]\).

Disciplinary Constraint Feasible (DCF)

A disciplinary optimiser is associated with each sub-system, and is charged with ensuring any local design constraints are satisfied, and minimizing any applicable objective functions (which may be modified from the original design problem) with respect to sub-system-level variables. The system optimiser is responsible to ensure system consistency, and minimize the system objective function, with respect to system-level variables.

The Balling's notation is \(SO[O1\|A1]\|O2\|A2]\).
| Collaborative Optimization (CO) | The CO formulation utilises the disciplinary evaluator instead of the disciplinary analyser in the DCF. Thus disciplinary optimisers that simultaneously determine the disciplinary and state variables are called at each iteration of the system optimiser. The system optimiser simultaneously determines the system design. A detailed description is shown in the next section. The Balling's notation is \( SO[O1[E1]|O2[E2]]. \) |
|--------------------------------|---|
| Analytical Target Cascading (ATC) | A system optimiser in the ATC fills the same roll as in the DCF formulation, but the intermediate sub-system optimisers are now charged with coordinating any sub-systems that are directly below them in the system hierarchy in addition to their own analysis. A detailed description is shown in the next section. The Balling's notation is \( SO[O1[A1]|O3[A3]|O4[A4]|O2[A2]|O5[A5]|O6[A6]]. \) |
Most of the MDO applications initially involve the direct integration of multiple disciplinary analyses with an optimiser, using a single-level formulation. For small problems, wiring all disciplinary designs together in such a system is quite feasible and usually efficient. However it is difficult to solve this kind of problems as the number of disciplines and interactions grow larger. Furthermore, in order to allow expert or group leaders to make decisions of discipline-specific variables using existing resources, it is required to use the multi-level formulations in which the disciplinary optimisers are located. Such formulations provide the discipline autonomy that is frequently mapped as a formulation to an existing organisational structure (Balling and Yao, 1997). On the other hand, the multi-level formulation can reduce communication requirements, since local design variables must no longer be passed to the system optimiser. The system optimiser is limited to coordinating the disciplinary interactions and guiding the entire process to a system-optimal design.

This research assumes that building design is a large-scale and complex problem. Obviously the single-level formulations are inappropriate solutions; hence two examples of typical multi-level formulations are introduced, namely collaborative optimisation, and analytical target cascading.

4.5 Review of Collaborative Optimisation and Analytic Target Cascading

Both the collaborative optimisation (CO) and the analytical target cascading (ATC) have been widely applied over the last decade. The CO is a two-level optimisation formulation used for non-hierarchical systems (Kroo et al., 1994; Budianto, 2000; Balling and Rawlings, 2000). The newer formulation of the ATC, on the other hand, decomposes a hierarchical system into two or more levels (Kim, 2001; Michalek, et al., 2005a; Choudhary et al., 2005). In hierarchical systems, children disciplines are coupled only to parent disciplines and not to each other. Non-hierarchical systems are more general since no restrictions are placed on how disciplines are coupled (Ballinger and Sobieszczanski-sobieski, 1996). This section reviews and compares the two formulations in depth.
4.5.1 Collaborative Optimisation

The collaborative optimisation (CO) formulation was proposed by Kroo et al., (1994) and then improved by Balling and Sobieszczanski-Sobieski (1996). The first application of CO for space vehicle design was cited by Braun and Kroo (1995) and for aircraft configuration by Sobieszczanski-Sobieski and Kroo (1996).

4.5.1.1 CO Overview

Sequential disciplinary designs and informal iteration can lead to designs that are sub-optimal. In many ad-hoc design procedures, individual design teams are assigned subsets of the design variables, parts of analysis, and local objective functions that are only vaguely aligned with the overall goal of the system. Sequential choices by these design teams lead to a kind of non-cooperative game, which may reach an equilibrium that is not an optimum for the system (Braun and Kroo, 1995). The left side of Figure 4.2 demonstrates the sequential analysis process using feedback and feedforward loops. In this context, decomposition in the CO formulation is based on the line of discipline. Problems in sub-system optimisation are managed by disciplinary experts (such as structural and mechanical designers), who have specialised knowledge and can use the corresponding analytical software, but the system-level optimiser needs to minimise the overall objective and coordinate between disciplinary designs. The right side of Figure 4.2 shows the elimination of feedback and feedforward of iteration loops through the introduction of an auxiliary variable (refer to Section 4.5.1.2.) in the CO formulation. Thus each disciplinary designers work on their own design in parallel with others.

Figure 4.2 Iteration Loops in a Series of Analyses (Kroo et al., 1994)
When considering the characteristics of decomposition and synthesis of the CO formulation, it is regarded as an effective MDO formulation for the following reasons (Braun and Kroo, 1995; Sobieszczanski-Sobieski and Haftka, 1997):

1) To decrease computational burden. The solution time and computational cost for most analysis and optimisation increases at a high rate with the increase in size and complexity of the design problem. The CO formulation enables to eliminate the local disciplinary variable from the system level problem. This means that the detailed sub-system (disciplinary) designs are, in a sense, hidden in system-level problem. Therefore the workload and communication requirement of the system-level co-ordination process are significantly reduced, while the sub-system optimiser is the responsibility of discipline-specific designs, without consideration of influences from other disciplinary designs.

2) To fit a conventional organisation. Large design projects involve a number of participants from various design disciplines, usually in separate locations. Disciplinary decomposition should be well suited for use in conventional design organisations. The CO design formulation is analogous to the current design process where a team leader (system-level optimiser) is responsible for minimising the overall objective while guiding a set of disciplinary experts (sub-system optimiser) into agreement.

3) To not require analysis software integration. The salient feature of an MDO problem is the large and monolithic analysis and design codes, and the implementations of design analyse become more complex with the increase of the number of disciplines. As a result analysis codes have grown so large that they are incomprehensible and difficult to maintain (Kroo et al., 1994). The CO formulation allows designers to use disciplinary applications independently.

4) To keep disciplinary design autonomy. In the CO formulation, the disciplinary problem is accomplished by providing design freedom within the sub-system optimisation process and enforcing multidisciplinary compatibility at system-level. This means that the disciplinary designer is free to specify the values of
variables that are unique to the disciplinary design but also those that might be an output from another disciplinary analysis.

### 4.5.1.2 CO Formulation

Many of the fundamental characteristics of CO are apparent in its various forms. This research follows a simple L2 measure for the sub-system objective (DeMiguel and Murray, 2000). The L2 measure adopts the sum of the square of the differences between the local values for a variable and requested system target values as an objective function of the sub-system optimisation. Figure 4.3 illustrates the communication channels between the system optimiser and the sub-systems.

![Diagram of Collaborative Optimisation Architecture](Braun and Kroo, 1995)

With the general CO formulation known, the relevant terminology can be formally introduced. The design variables of the original design problem are separated into local variable $X_i$, shared variable $X_s$, and coupling variable $Y_{ij}$ according to discipline partitioning. The auxiliary variable $X^0$ is imported as a new variable to replace the shared variable $X_s$ and coupling variable $Y_{ij}$ at the system level. Such auxiliary
variables break the links between disciplines to enable each discipline to execute design concurrently. In addition, compatible constraint functions (e.g. \(d_i^*(X^0)\)) are introduced to achieve consistency in the design process. Now that the terminology has been established, the detailed formulation of system and sub-system formulation are presented in Equation 4.1 and 4.2.

**System-level Formulation**

Minimise \(f_s(X^0)\)

With regard to \(X^0 = (X_s^0, Y_{ij}^0)\)

Satisfy: \(d_i^*(X^0) = (x_i^0 - x_{ij}^*)^2 + (y_{ij}^0 - y_{ij}^*)^2 = 0\)

\(d_i^*(X^0) = (x_i^0 - x_{ij}^*)^2 + (y_{ij}^0 - y_{ij}^*)^2 = 0\) \hspace{1cm} (Equation 4.1)

**Sub-system-level i Formulation**

Minimise: \(d_i(X_i) = (\tilde{x}_i^0 - x_i)^2 + (\tilde{y}_i^0 - y_i)^2\)

With regard to \(X_i = (x_i, x_{ij}, y_{ij})\)

Satisfy: \(g_i(X_i) < 0; \ h_i(X_i) = 0\) \hspace{1cm} (Equation 4.2)

Where,

\(f_s(X^0)\) is the objective function in the system level;

\(X^0\) is the vector of system-level variable;

\(x_i^0\) is the vector of the shared variable in the system level;

\(y_{ij}^0\) is the vector of the coupling variable in the system level;

\(x_{ij}^*\) is the vector of optimal value of shared variable sent by the \(i^{th}\) sub-system;

\(y_{ij}^*\) is the vector of optimal value coupling variable sent by the \(i^{th}\) sub-system;

\(d_i^*(X^0)\) is the \(i^{th}\) compatible constraint in system level;

\(\tilde{X}_i^0\) is the vector of target value of system-level variable that are sent to the \(i^{th}\) sub-system;

\(d_i(X_i)\): is the objective function in the \(i^{th}\) sub-system;

\(X_i\) is the vector of decision variables in the \(i^{th}\) sub-system;

\(x_i\) is the vector of local variable of the \(i^{th}\) sub-system

\(x_i^0\) is the vector of shared variable corresponding to the \(i^{th}\) sub-system;
\( y_{ji} \) is the vector of coupling variable that is input to \( i^{th} \) sub-system but is output from the \( j^{th} \) sub-system;
\( x_{ji}^0 \) is the vector of target value of shared variable sent from system level to the \( i^{th} \) sub-system;
\( y_{ji}^0 \) is the vector of target value coupling variable sent from system level to the \( i^{th} \) sub-system.

In the system level, the optimiser manages disciplinary interactions and changes the target values of auxiliary variables (\( X_{i}^0, Y_{i}^0 \)), seeking to minimise the system objective \( f_s(X^0) \). The optimiser for sub-system \( i \) receives the system target values pertinent to sub-system \( i \), and minimises the discrepancy between the system target values (i.e. \( \tilde{X}_{i}^0 \)) and their corresponding sub-system values (i.e. \( x_{ji} \) and \( y_{ji} \)) by the \( i^{th} \) sub-system, subject to satisfying the sub-system's local design constraints (i.e. \( g_{i} \) and \( h_{i} \)). While the compatible constraint functions at system level is the same as the objective functions at sub-system level (i.e. \( d_{s} f(X^0) \)). Once the compatibility constraints reach zero, it means that all values of interdisciplinary variables are agreed between disciplines. Although the sub-systems' objective functions take the same form as the compatibility constraint functions of the system level, both \( x_{ji} \) and \( y_{ji} \) are variables in the \( i^{th} \) sub-system level, while both \( x_{ji}^0 \) and \( y_{ji}^0 \) are variables at the system level. In other words, both objective functions of sub-system level and constraints functions of system level are the same with regard to different variables that have the same physical meaning.

In the CO formulation, each sub-system optimiser is given sufficient degrees of freedom to achieve a design that is feasible with respect to its local constraints, because the sub-system-level optimisers are able to manage all discipline-specific variables (i.e. \( X_{i} \)) which include the local variable \( x_{i} \), the shared variable \( x_{ji} \) and the coupling variable \( y_{ji} \). It should be noted that the target values of auxiliary variables corresponding to the \( i^{th} \) sub-system (i.e. \( X_{ji}^0 \)) keep constant during the sub-system optimisation, while the optimal values of auxiliary variables sent from the sub-system optimisers (i.e. \( x_{ji}^* \)) are fixed parameter during the system optimiser.
4.5.2 Analytic Target Cascading Optimisation

The analytic target cascading (ATC) method was first introduced by Michelena and Papalambros (1997) and further detailed by Kim (2001). This formulation organises a set of design tasks in a hierarchy. This section introduces the ATC process and its mathematical formulation.

4.5.2.1 ATC Overview

ATC is a multilevel, multidisciplinary design methodology to find an optimal system design, ensuring consistency between sub-systems or disciplines and achieving the overall targets assigned at the top of the hierarchy (Kim, 2001; Michelena et al., 2003). In the whole process of ATC, the original design problem is partitioned into a set of sub-problems constituting system, sub-systems and components shown in Figure 4.4. Design targets are specified at the top level of the multilevel design formulation and ‘cascaded down’ to lower levels. Sub-problems at lower levels are formulated so that all elements included in the hierarchy match the cascaded targets consistent with the overall system targets. Design targets derived at lower levels are co-ordinated with the higher level ones by iteratively adjusting values of targets and decision variables (Kokkolaras et al., 2002).

Hence the major benefits of ATC are a reduction in design cycle time through avoidance of design iterations at the later stage and an increased likelihood that physical prototypes will be closer to production quality. Furthermore the ATC facilitates concurrency in the system design. Once targets are identified for systems, sub-systems, components and the latter elements can be isolated and designed in detail independently, allowing the outsourcing of sub-systems and components to suppliers (Michelena et al., 2003).
Each sub-problem in the ATC hierarchy requires a decision model and one or more analysis models. The decision model of a sub-problem is its formulation as a design optimization model. It requires representation of sub-problem performance $R$, decision variables $x$, and all relevant constraints $g$ and $h$ on decision variables. The decision model also embodies the links of each sub-problem to upper and lower level sub-problems in the hierarchy. It is through these links that top-level targets are propagated down and lower-level responses are rebalanced up the hierarchy. Each decision model is associated with one or more analysis models to compute performance $R$ as a function of decision variables $x$. The analysis models take values of decision variables as input and returns their corresponding performance response as output (Figure 4.5). Every decision model requires an analysis tool (a simulation for example) or an analytic function $r$ from which performance $R$ can be derived with respect to decision variables $x$. In the building simulation context, the simulation will typically return a dataset (for example, a vector of room temperatures at every timestep), which is processed by the analysis model into the required performance response (for example, maximum daily temperature).

Figure 4.4: Schematic of the Analytical Target Cascading Process for a Top-down and Bottom-up, Level-by-level Solution Sequence (Michelena et al., 2003)

Figure 4.5: Links between Decision and Analysis Model in ATC Framework (Choudhary, 2004)
Every sub-problem in the ATC hierarchy is formulated and solved independently, and posed to optimise “target matching” with its upper and lower level sub-problems. The ATC problem is solved iteratively for meeting all targets as closely as possible by a co-ordination strategy. Once compatible targets are derived from the ATC process, individual sub-problems can be isolated and outsourced to be solved in further detail, thereby enabling truly concurrent design (Kokkolaras et. al., 2002). Formulating a design problem in the ATC framework requires (Choudhary, 2004):

- Identifying appropriate decomposition;
- Hierarchical organization of decomposed sub-problems and identifying key links between them;
- Formulation of sub-problems as decision models and identifying suitable optimization algorithms to solve them; and
- Building and mapping appropriate analysis models to each decision model.

Once formulated, steps involved in solving an ATC problem can be summarized as (Choudhary, 2004):

- Specifying values of overall design targets (referred to formally as “target setting”);
- Propagating specified top-level targets to lower levels and optimizing all sub-problems to match targets as closely as possible; and
- Iteratively searching for an overall consistent solution by applying a co-ordination strategy.

4.5.2.2 ATC Formulation

The first step in setting up an ATC formulation is to decompose the problem. Once the decision model is determined, appropriate analysis models are associated with each decision model in the hierarchy. An analysis model is appropriate if it can compute performance values as functions of its decision variables. Every analysis model evaluates design decisions by taking variables and parameters as input and returning performance values as output.
In the ATC formulation decomposition of problems also includes identifying common links between sub-problems. Horizontal links between sub-problems are called linking variables. Linking variables represent decisions shared among two or more decision models at the same vertical level (Kim, 2001). Their value is determined individually by the decision models that are coordinated by the upper level parent problem. Thus sub-problems that share linking variables must also have a common decision model at the upper level.

The vertical relationships between decomposed levels are embodied by performance targets and responses. Figure 4.6 shows the information flow up and down in the ATC hierarchy for a three-level problem with one system, three sub-systems, and two components. Rectangular boxes represent the decision models and oval boxes are the analysis models. As an example, sub-system B model receives its target values $R$ from the system level model S. At an iteration $k$ in the ATC process, sub-system B also has responses values (i.e. $R_{b1}$ and $R_{b2}$), and linking variable (i.e. $y_{b1}$ and $y_{b2}$) from the component levels. Sub-system B is solved for determining values of its local decision variables, values of component level response $R_{b1}$ and $R_{b2}$, and values of coordinating linking variables $y_{b}$ such that deviations from information received from upper and lower levels are minimized. This includes minimizing deviations between $R_{b1}$ and $R_{b1}^L$, $R_{b2}$ and $R_{b2}^L$, $y_{b}$ and $y_{b1}^L$, $y_{b}$ and $y_{b2}^L$, $R_{b}$ and $R_{b}^L$.

Sub-system B computes its response $R_b$ by using its analysis model, which requires values of local decision variables $x_b$ and lower level responses $R_{b1}$ and $R_{b2}$ as input. The analysis model returns the sub-system response $R_b$ as output. Note that in these interactions responses of a particular level are decision variables input to the analysis model at the upper level. Another condition in this organization is that linking variables are shared between children of a parent problem.
For target matching a problem for the j-th design model at i-th level, the general formalisation of the optimisation model is stated in Equation 4.3:

Minimise: \( \| R_y - R_y^i \|^2 + \| y_y - y_y^i \|^2 + \epsilon_y^s + \epsilon_y^f \)

With regard to \( \vec{x}_y, y_{(i+1)}^i, e_y^s, e_y^f \)

Subject to: \( \sum_{k \in C_y} \| R_{(i+1)k} - R_{(i+1)k}^{(L)} \|^2 \leq \epsilon_y^{(L)} \)
\( \sum_{k \in C_y} \| y_{(i+1)k}^L - y_{(i+1)k}^L \|^2 \leq \epsilon_y^{(L)} \)
\( g_y(\vec{x}_y) \leq 0 ; h_y(\vec{x}_y) = 0 \) \hspace{1cm} (Equation 4.3)

Where,
\( \vec{x}_y = [\vec{x}_y, y_y, R_{(i+1)k}, ... R_{(i+1)k_c}]^T \) is the vector of all decision variables of element \( j \) at level \( i \); 
\( R_y = r_y(\vec{x}_y) \), where \( r_y \) is the vector function that represents the analysis model. It calculates the responses for element \( j \) at level \( i \) by taking in all its decision variables as input;
\( C_y = \{ k_1, ..., k_{c_y} \} \) and \( c_y \) is the number of child elements;
\( y_{(i+1)} = y_{(i+1)k}, ... y_{(i+1)k_c} \) and \( R_{(i+1)} = R_{(i+1)k}, ... R_{(i+1)k_c} \); 
\( \vec{x}_y \) is the vector of local decision variables for element \( j \) at level \( i \); 
\( y_y \) is the vector of linking variable for element \( j \) at level \( i \);
\( \varepsilon_y^R \) is the tolerance variables for consistency of target set at element \( j \) at level \( i \) and the responses of \( j \)'s children;

\( \varepsilon_y^Y \) is the tolerance variables for consistency of linking variables coordinated at element \( j \) at level \( i \) for child element at the (i+1)th level;

\( R_y^U \) is vector of response values cascaded to element \( j \) at level \( i \) as targets from its parent at level (i-1);

\( y_y^U \) is the vector of coordinating linking variables for the linking variables in the children of element \( j \) at level \( i \). This vector includes one copy of each linking variables from all element \( j \)'s children;

\( R_{(i+1)k}^L \) is vector of response variables values cascaded to the element \( j \) at level \( i \) as targets from its kth child at level (i-1);

\( y_{(i+1)k}^L \) is the vector of linking variables values cascaded element \( j \) at level \( i \) from its kth child at level (i-1);

\( g_y \) and \( h_y \) are vector functions representing inequality and equality design constraints

\( \| \cdot \|^2 \) represents the square of the \( L_2 \) norm.

Equation 4.3 presents the ATC formulation at the intermediate level. At the top level of hierarchy the problem is formulated in Equation 4.4:

Minimise: \( \| R_y - T_j \|^2 + \varepsilon_y^R + \varepsilon_y^Y \)

With regard to \( x_{ij}, y_{(i+1)j}, \varepsilon_y^R, \varepsilon_y^Y \)

Subject to:

\[ \sum_{k \in C_j} \| R_{(i+1)k} - R_{(i+1)k}^L \|^2 \leq \varepsilon_y^R \]

\[ \sum_{k \in C_j} \| y_{(i+1)k} - y_{(i+1)k}^L \|^2 \leq \varepsilon_y^Y \]

\( g_y (x_y) \leq 0; \ h_y (x_y) = 0 \) (Equation 4.4)

where \( T_j \) is overall system targets which are specified before performing optimisation.

However for the sub-problems at the bottom level, they do not have any children, thus the formulations shown in Equation 4.5 do not contain the tolerance variable for coordinating lower level information.

Minimise: \( \| R_y - R_y^o \|^2 + \| y_y - y_y^o \|^2 \)

With regard to \( x_y \)

Subject to: \( g_y (x_y) \leq 0; \ h_y (x_y) = 0 \) (Equation 4.5)
Chapter 4 Multidisciplinary Design Optimisation

Note: \( \bar{x}_v \) merely include \( \bar{x}_y \) and \( y_y \) at the bottom level.

4.5.3 Discussion

Both the CO and the ATC are classed as the multi-level MDO and are used for complex modern engineering design problems. However the specific characteristics of each formulation enable them to apply to various industry designs. Allison et al., (2005) investigated the distinctions between CO and ATC based on formulation and solution process of each method. This section presents these differences between CO and ATC and the best-suited formulation for building design optimisation.

4.5.3.1 Comparisons of CO and ATC

Based on the understandings of the CO and the ATC formulation, three attributes are used to differentiate between them. These are:

- Decomposition scheme

With regard to the complex MDO problems, the main approach is to partition the original design problem into smaller and easier to solve sub-problems, and then coordinate the optimisation of each sub-problem in order to achieve a consistent solution that is optimal for the overall system. There are various decomposition schemes for the MDO problems which have been reviewed in Section 3.5.3. The CO formulation decomposes along discipline boundaries which are often dictated by the organisation structure or the analysis tools available. While the ATC paradigm is based on hierarchical organisational and analysis structure, which are typically partitioned by object. Therefore the CO formulation is motivated by multidisciplinary analyse needs in the aerospace industry, while the ATC formulation is motivated by product development needs in the automotive industry (Allison et al., 2005).

- Formulation process

After being decomposed by discipline or physical component, it is necessary to examine input-output relationship of simulations that link with each discipline and component. The CO formulation can then be established. The left sides of Figure 4.7 and Figure 4.8 demonstrate type of problem for CO and ATC formulation respectively, while the right sides present corresponding formulations. Based on observations of the CO and the ATC formulation, it is easy to identify two differences.
Firstly, in the CO formulation the top level performs the co-ordination among sub-system-level problems, and does not stand for any discipline; all disciplines are located at the lower level and given equal importance. While the ATC formulation is based on hierarchical analysis structures with unidirectional functional dependencies. In other words, the parent problems in the ATC formulation include all children problems. To sum up, problems in different levels have dependent relationship in ATC formulation but independent relationship in the CO formulation.

Secondly, Allison (2004) demonstrates that the linking variable within the ATC context includes all shared variables and part of the coupling variables in the CO context. This is because the coupling variables consist of feedback and feedforward variables. For example, in Figure 4.7, Y1 and Y2 are feedback variables from simulation I to simulation II and simulation III respectively. A1 is the feedforward...
variable from simulation II to simulation I and simulation III. While the ATC formulation presented does not accommodate feedback coupling. For example the system in Figure 4.8 does not support the relationship where outputs generated by the parent element analysis (i.e. Simulation I) are required as inputs to the analysis of the child element analysis (i.e. Simulation II):

- Co-ordination strategy

A co-ordination strategy is applied to the iteration through the multi-level MDO formulation. The main requirement from a co-ordination strategy is that it should converge to the same solution set as that of an un-decomposed problem (Papalambros, 2002). ATC shares the idea of co-ordination strategy to achieve consistency with CO, namely adding compatible constraints to minimise deviations between children problems and the parent problems. The difference is that the ATC adopts inequality constraints with a certain tolerance (e.g. \( \sum_{k \in C_k} \| y_{(i+1)k} - y_{(i+1)k}^L \|^2 \leq \epsilon_{\Gamma}^C \)), while the CO formulation uses equality constraints (e.g. \( d'_i(X^o) = 0 \)). In addition, both co-ordination processes consist of nested loops. The ATC has a nested co-ordination strategy while the CO has a nested optimisation. A nested co-ordination strategy means that while parent problems are usually executed multiple times, these problems are provided as a static set of responses from child problems and a static set of target from higher level problems. Each optimisation problem can be executed autonomously. Whilst the system-level optimiser in the CO formulation needs to wait for results from the sub-system-level optimisation problem at each optimisation iteration. In other words, the sub-system-level's optimisation problem in the CO formulation can consider constraints from the system level which is activated in each system-level's iteration. Therefore both formulations motivate the use of low accuracy analysis models to avoid a huge amount of computational demand.

4.5.3.2 Selection of Formulation

The choice of the exact formulation will depend on the availability of models (i.e. simulation/analysis tool), so this task must be done carefully with minimum variations to the existing design methods (Kim. et al., 2002). Based on the reviews of state-of-the-art building design optimisation in Chapter 3, decomposition along discipline is
more advantageous over physical object decomposition for the following three reasons.

1) Characteristics of building design team. The groups of different professionals that together take charge of building design and thus constitute a ‘design team’ usually fall into the categories of architect, structural engineer, mechanical and electrical service engineer, construction engineer and quantity surveyor (Grierson and Khajehpour, 2002). Each member has a range of responsibilities and interests. However to produce a good design solution that effectively satisfies cost, time and functionality constraints, the co-operative efforts of this group of professionals are essential (Mathews and Rafiq, 1994). Hence in the field of building design, design teams are formed with consideration of discipline boundaries.

2) Maturation of the discipline-based analysis tools. Both ATC and CO formulations allow each optimiser to connect certain simulation/analysis software, which help designers predict/evaluate design performances. Hence the better formulation needs to enable a more effective and efficient use of emerging building performance analysis tools. The best-known solver for the building thermal, HVAC and network airflow is DOE-2 (Lawrence, 1979). It utilises hourly weather data to calculate the hour-by-hour performance and response of a building with a known description. A host of commercial software (e.g. eQuest and EnergyPlus) use DOE-2. The structural design software (e.g. SAP 2000 and ETABS) use the displacement method, force method, and more advanced finite element analysis to visualise or predict the stress and deflection of each structure component.

3) Different design methods. Each disciplinary designer has formed his/her own design method. Building services designers analyse building thermal behaviour based on functional room or zone (e.g. office and toilet), because these rooms have different design requirements in terms of indoor temperature, ventilation, lighting etc. The structural design is based on different structural units which can be a frame or a load bearing wall.
Based on these above reasons, this research adopted the CO formulation to solve the multidisciplinary building optimisation for thermal and structural design.

4.6 Summary
This chapter mainly focused on the methodologies used for multidisciplinary design optimisation. This demonstrated that MDO is the collection of methods which are used to obtain a consistent and overall optimal system design. Thus some complex engineering designs will benefit from the applications of the MDO methodology when dealing with a systematic, logical design procedure, increased number of design variables & constraints to be handled, not biased by intuition or experience, and achieves a reduction in design time. At the same time some issues (i.e. the MDO formulation, computing requirement and data exchange) can inhibit further application of MDO. Taking these issues and status of the MDO application in the construction industry into account, the establishment of an appropriate MDO formulation of building design optimisation is the emphasis of this research. Thus the MDO formulations that emerged were described in terms of the single and multi-level formulation. However the multi-level formulation demonstrates some advantages over the single-level formulation (such as number of variables, utilisation of specialised knowledge and software), so both CO and ATC that are multi-level are studied in depth. In order to select one for building design optimisation, they were compared in term of decomposition scheme, procedure of formulation and co-ordination strategy. Finally the CO formulation was chosen because of the characteristics of the building design, the analysis software available and the design method. Herein it is worth mentioning that all MDO formulations described in this chapter are used to solve a single-objective design problem, which is not suited for the current multi-objective engineering problem. Hence the next chapter will introduce how to integrate the MDO formulations with multi-objective optimisation algorithms.
CHAPTER FIVE: COLLABORATIVE OPTIMISATION FRAMEWORK WITH A PARETO BASED GENETIC ALGORITHM (COPGA) DEVELOPMENT

5.1 Introduction
This chapter explains the process of developing collaborative optimisation framework with a Pareto based genetic algorithm (COPGA). This process starts with reviews of multi-objective optimisation solutions, and then identifies problems within the multi-objective collaborative optimisation (MOCO). Changes in the MOCO are made to facilitate the adoption of the Pareto-based GA algorithm. Then the development of the COPGA framework is described. In order to validate this framework, the COPGA and All-at-once results for the same mathematical problem are compared. Finally some suggestions are put forward for the application of the COPGA framework on a building design problem in Chapter 6.

5.2 Multi-objective Optimisation Solution
In many building design optimisation problems, there are multiple measures of performance, cost, comfort etc, which should be optimised simultaneously. It is possible to optimise each separately however this rarely gives a suitable solution to the global problem. With a single-objective optimisation algorithm a single 'perfect' solution can be obtained, this is rarely the case with multi-objective optimisation problems. Instead the multi-objective problems tend to be characterised by a group of alternative solutions each of which is considered feasible. The aim is to present the decision-maker (engineer) with the selection of alternatives, permitting him/her to make an informed choice. The purpose of these methods is to help the engineer to make the right decision in conflicting situations (Loosemore, 2003). Marler and Arora's (2004), and Miettinen (1999) classified approaches for multi-objective problems into four groups based on the point in the optimisation process at which the decision maker expresses preference in the choice of solution.
A non-preference articulation: This type of approaches does not use any preference information. Example is the Min-max formulation (Steuer, 1986);

A priori articulation of preference information: In this type of approaches, before the actual optimisation is conducted the different objectives are somehow aggregated to one single-objective, such as the case in the weighting factor methods, and goal programming (Charnes et al., 1955);

A progressive articulation of preference: This class of approaches is generally referred to as interactive. They rely on progressive information about the decision-makers preferences simultaneously as they search through the solution space. Typical examples are the STEM approach (Benayoun et al., 1971); and

A posterior articulation of preference: This kind of approaches enables to first search the solution space for a set of non-inferior solutions, such as in the Pareto optimality (Pareto, 1906).

This section is focused on studies of the weighting factor method and Pareto optimality, which belong to the category of: on priori articulation of preferences, and posterior of preferences respectively, as these two approaches are commonly used in building design (detailed reviews presented in Section 3.4.3).

5.2.1 Weighting Factor Method

Weighting objectives is the oldest multi-objective solution technique. In the very basic form of the weighting method presented in Equation 5.1, each objective is assigned a weight depending on the decision maker’s preference and the judged importance of each objective. The objectives are then combined together to form a single equation for optimisation. In doing this the users can obtain a truly multi-objective solution to the problem as the objectives are restricted by the decision makers judgement. When fixing the weights, the solution of the optimisation may converge on a point that is not the true optimum.
Chapter 5

Minimise $\sum_{i=1}^{K} w_i f_i(X)$

Where $w_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{K} w_i = 1$  

(Equation 5.1)

This approach usually has difficulties in that the objective functions are generally of different magnitudes and therefore might have to be normalised first. In addition although the formulation is simple, the method is somewhat ad-hoc, as there is no clear relation between the weightings and the obtained solution (Andersson, 2000).

5.2.2 Pareto optimality

With single-objective optimisation algorithm, the optimum is a single point within the feasible solution. When there is more than one objective to be optimised the notion of an optimal solution is replaced by a less definitive concept. One powerful approach used in multi-objective design optimisation is known as Pareto optimality.

Pareto optimality is a required measure of multi-objective optimisation, independent of whether the optimisation procedure uses Pareto optimality as a method of progression in the search. The concept of Pareto optimality defines the optimum solution for any multi-objective problem (Loosemore, 2003)

Practical problems are often characterised by several non-commensurable and often competing measures of performance, or objectives. Assuming a minimisation problem, Pareto optimality is defined as follows (Fonseca and Fleming, 1993):

A vector of decision variables is said to be Pareto optimal if no feasible vectors of design variable exists which would decrease some objectives without causing a simultaneous increase in at least one other objective.

Hence Pareto-optimal solutions are also called efficient, non-dominated, and non-inferior solutions. In addition, this concept of Pareto optimality almost always gives a set of solutions called the Pareto optimal set rather than a single solution. For a given Pareto optimal set, the corresponding objective function values in the objective space are called the Pareto frontier. Both the weighted method and genetic algorithm (GA) are regarded as the most common approaches to obtain a Pareto optimal set (Andersson, 2000; Marler and Arora 2004). Miettinen (1999) stated that the weighted
methods could be used in a repetitive process by varying the preference applied to each of the objectives, a set of Pareto optimal solutions can then be attained. This method is the simplest and most straightforward way of obtaining multiple points on the Pareto-optimal frontier. However this method is associated with some major drawbacks. Depending on the scaling of the different objectives and the shape of the Pareto frontier, it is hard to select the weightings to ensure that the points are spread evenly on the Pareto frontier (Das and Dennis, 1997). A detailed explanation for Pareto-based GA is described as follow.

5.2.3 Optimisation Algorithm for This Study

The two previous methods are widely adopted in building design optimisation. The weighting factor method is one of the most computationally efficient, easy-to-use, and common approach. However, in practice precise design preferences are rarely known before optimisation, whilst this method is fully dependent on the decision-maker’s preferences, in some cases the analysis have to be performed a few times because of changes in weighting factors. Pareto optimality combines the objective functions in a fair and unbiased way, which complies with the common practice that is to improve at least one objective without worsening the others (Papalambros, 2002), and the Pareto set would not change as long as the problem description is unchanged. Under these principles the Pareto optimality was the focus of increasing amounts of research over the past years.

This research attempts to use a Pareto-based genetic algorithm for a multi-objective optimisation problem. Several varieties of this algorithm were developed such as the multi-objective genetic algorithm (Fonseca and Fleming, 1993), the niched Pareto Genetic algorithm (Horn et al., 1994), the non-dominated sorting genetic algorithm (NSGA) (Srinivas and Deb, 1994), the strength Pareto evolutionary algorithm (Zitzler and Thiele, 1999), and the fast non-dominated sorting genetic algorithm (NSGA-II) (Deb et al., 2002). This study adopted the NSGA-II that has two important features, namely: Pareto rank, and crowding distance.
The approach of Pareto rank explicitly utilises the concept of Pareto optimality in evaluating fitness or assigning the selection probability to solutions. In the NSGA-II, the process of Pareto rank, presented in Figure 5.1, consists of ranking all the solutions that are not dominated by any other as ‘1’. This set is then removed from the ranking procedure and the next set of non-dominated solutions is assigned the rank ‘2’, and so on.

**Objective 2**

![Figure 5.1 Pareto Rank](image)

The approach of crowding distance aims to obtain a uniform spread of solutions along a best-known Pareto frontier. This approach is a measure of population density around a solution. To get an estimate of the density of a particular solution “i” in the population, the average distance of two solutions on either side of solution “i” along each of the objective is computed, the quantity $cd_i$ serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbours with the same Pareto rank (this is called the crowding distance). In Figure 5.2 the crowding distance of the $i^{th}$ solution (marked with filled circles) is the average side-length of the cuboid (shown by a dashed box). Equation 5.2 is used to calculate the crowding distance of each point in each Pareto set.

$$cd_k(x_{i,k}) = \frac{f_k(x_{i+l,k}) - f_k(x_{i-l,k})}{f^{\text{max}}_k - f^{\text{min}}_k}$$

$$cd(x_i) = \sum_{k=1}^{l} cd_k(x_i) \quad \text{(Equation 5.2)}$$

Where,

$l$: Number of the objective functions;

114
\(x_{i,k}^{(1)}\): The \(i^{th}\) solution with respect to the \(k^{th}\) objective function;

\(f_k(x_{i+1,k})\): Value of the \(k^{th}\) objective function with the \((i+1)^{th}\) solution;

\(f_k^{\text{max}}, f_k^{\text{min}}\): The maximal and minimal value of the \(k^{th}\) objective function in the generation.

Objective 2

\[\begin{array}{c}
\bullet \\
i-1 \\
\circ \\
i \\
\bullet \\
j+1
\end{array}\]

Cuboid

Objective 1

**Figure 5.2 Crowding Distance**

In the NSGA-II, if the solutions are in the same Pareto set, the solution with a higher crowding distance wins. Otherwise, the solution with the lowest rank is selected.

### 5.3 Test Problem

In order to explore the problem with an existing MDO framework and develop a proposed framework, Equation 5.3 defines a small multi-objective multidisciplinary test problem.

Min: \(F_1\) and \(F_2\)

\[F_1 = x_2^2 + x_3 + y_{12}; \quad F_2 = -y_{21}\]

Inequality constraints \(g_1 = \frac{y_{12}}{8} - 1 \geq 0; \quad g_2 = \frac{y_{21}}{10} - 1 \geq 0\);

Equality constraints \(y_{12} = x_1^2 + x_2 + x_3 - 0.2y_{21}; \quad y_{21} = x_1 + x_3 + \sqrt{y_{12}}\)

Bounds: \(-10 \leq x_1 \leq 10; \quad 0 \leq x_2 \leq 10; \quad 0 \leq x_3 \leq 10\)  

**Equation 5.3**

There are two coupled disciplines and two objective functions (i.e. \(F_1\) and \(F_2\)) in this problem. In addition, according to definition of the local, shared and coupling
variables in Section 4.4.1, \( x_i \) and \( x_j \) are shared variables, \( y_{i2} \) and \( y_{21} \) are coupling variables, \( x_2 \) is a local variable in the sub-system 1, while there is no local variable in sub-system 2.

## 5.4 Implementation by the MOCO Framework

The Collaborative Optimisation framework with a Pareto based genetic algorithm (COPGA) is developed based on the Multi-objective Collaborative Optimisation approach (MOCO) (Tappeta and Renaud, 1997), which is the first extension onto collaborative optimisation (CO) (Kroo et al., 1994) for multi-objective multidisciplinary optimisation problems. The MOCO framework is a two-level architecture in which the system-level optimiser handles the shared and coupling variables to co-ordinate designs among disciplines, minimising the design objective functions, while discipline-specific design problems are completed in the sub-system level. The detailed MOCO formulations are presented in the first part of this section, and then limitations with this framework are identified.

### 5.4.1 Formulations of the MOCO Framework

Formulations of both system-level and sub-system-levels for the problem in the Equation 5.3 are demonstrated below:

#### System-level formulation:

Min: \( F(X^0) = w_1 F_1 + w_2 F_2 \)

Equality Constrains:

\[
d^*_1(X^0) = (x_1^* - x_1^0)^2 + (x_3^* - x_3^0)^2 + (y_{i2}^* - y_{i2}^0)^2 = 0;
\]

\[
d^*_2(X^0) = (x_2^* - x_2^0)^2 + (x_3^* - x_3^0)^2 + (y_{21}^* - y_{21}^0)^2 = 0;
\]

Bounds: \(-10 \leq x_i^0 \leq 10; 0 \leq x_j^0 \leq 10\)

With respect to \( X^0 = [x_1^0, x_2^0, y_{i2}^0, y_{21}^0] \)

#### Sub-system 1 formulation:

Min: \( d_s(X^1) = (x_1^1 - \bar{x}_1^0)^2 + (x_3^1 - \bar{x}_3^0)^2 + (y_{i2}^1 - \bar{y}_{i2}^0)^2 \)

Inequality constraint: \( \frac{y_{i2}^1}{8} - 1 \geq 0 \)
Equality constraint: \( y^{1}_{12} = x_1^2 + x_2^2 + x_3 - 0.2y_{21}^0 \)

Bounds: \(-10 \leq x_1^1 \leq 10; 0 \leq x_2 \leq 10; 0 \leq x_3^1 \leq 10\)

With respect to \( X^1 = [x_1^1, x_2, x_3, y_{12}^1] \)

Sub-system 2 formulation:

\[
\text{Min } d_2(X^2) = (x_1^2 - \overline{x}_1^2)^2 + (x_2^2 - \overline{x}_2^2)^2 + (y_{21}^2 - \overline{y}_{21}^0)^2
\]

Inequality constraint: \( \frac{y_{21}^2 - 1}{10} \geq 0 \)

Equality constraint: \( y_{21}^2 = x_1^2 + x_2^2 + \sqrt{y_{12}^0} \)

Bounds: \(-10 \leq x_1^2 \leq 10; 0 \leq x_2^2 \leq 10; \)

With respect to \( X^2 = [x_1^2, x_2^2, y_{21}^2] \)

Where,

\( \bullet^0 \): Variable in system level

\( \bullet^i \): Discipline-specific variables in the \( i \)th sub-system level

\( \bullet^* \): Optimal value of variable

\( \overline{\bullet} \): Target value of interdisciplinary variable sent from system level to sub-system level

In the MOCO framework, auxiliary design variables (i.e. \( X^0 \)) are introduced to replace interdisciplinary variables including the shared variables (i.e. \( x_1 \) and \( x_3 \)) and coupling variables (i.e. \( y_{12} \) and \( y_{21} \)) in order that the sub-systems can work in parallel. The values of these auxiliary variables (e.g. \( \overline{x}_1^0 \) and \( \overline{x}_3^0 \)) are constant in the course of sub-system level optimisation while they (e.g. \( x_1^0 \) and \( x_3^0 \)) could be adjusted by the system-level optimiser. The sub-system-level optimisers are free to control local variable (e.g. \( x_2 \)), and also manage the interdisciplinary variables (e.g. \( x_{1}^1, x_3^1 \) and \( y_{12}^1 \) in the sub-system 1). Thus a situation will emerge in which the same variable in the system level receives different values from the each sub-system (e.g. \( x_1^* = x_1^{**} \)). In order to achieve consistency in the design process compatibility constraints (e.g.}
$d_i^*$ and $d_j^*$) are introduced at system level. These constraints are the sum of square differences between values of system-level variables (e.g. $x_1^0$ and $y_{12}^0$) and optimal values of these in the corresponding sub-system level (e.g. $x_1^{1^*}$ and $y_{12}^{1^*}$). If they are set to zero, it will imply that $x_1^{1^*} = x_i^0$. However some engineering problems can allow small error tolerances, so in these cases the compatibility constraint can be given a deviation tolerance such as $d_i^* \leq 10^{-2}$.

With regard to multi-objective problems, the MOCO framework adopted a weighting method and solved it in the system level. While the single objective function in the sub-system level is to minimise the sum of square difference between target values of interdisciplinary variables sent from system level (e.g. $x_i^0$ and $y_{12}^0$) and values of these in the corresponding sub-system level (e.g. $x_i^1$ and $y_{12}^1$). It is worth noting that the formulation of system-level constraint functions and sub-system-level objective functions are similar, but during the course of sub-system-level optimisation, the value of the interdisciplinary variables in the corresponding sub-system level, such as $x_i^1$ in the sub-system 1 and $x_i^2$ in the sub-system 2, can be varied, while target values sent from the system-level, such as $x_i^0$, are constant. In the process of system-level optimisation, values of the interdisciplinary variables in the corresponding sub-system level, such as $x_i^{1^*}$ and $x_i^{2^*}$, have been optimised and are fixed, while the target value of these can be varied.

5.4.2 Limitations with the MOCO Framework

There are some limitations with the implementation of the MOCO framework by Tappeta and Renaud (1997).

1) **Step size:**

In the MOCO framework, once the sub-system optimisation problems are solved, the change in the optimal sub-system objective function with respect to any fixed parameter (e.g. $x_i^0$ and $y_{12}^0$) can be calculated, Equation 5.4 presents an example of the sub-system 1 function objective with regard to $x_i^0$. 

118
\[
\frac{\Delta d^*_i}{\Delta \bar{x}_i^0} = \frac{\partial d_i^*}{\partial \bar{x}_i^0} + \lambda_i^* \frac{\partial g_i}{\partial \bar{x}_i^0}
\]

Equation 5.4

Where: \( \lambda^* \) is the Lagrange multiplier vector for the optimum solution.

Because constraint \( g_1 \) is not a function of \( \bar{x}_i^0 \), Equation 5.4 can be simplified as Equation 5.5:

\[
\frac{\Delta d^*_i}{\Delta \bar{x}_i^0} = \frac{\partial d_i^*}{\partial \bar{x}_i^0} = -2(x_i^* - \bar{x}_i^0)
\]

Equation 5.5

Because the objective functions at the sub-system level are expressed in the same way as the constraint functions at system-level, then the optimal sensitivity value (e.g. \( \frac{\Delta d^*_i}{\Delta \bar{x}_i^0} \)) can be obtained from the sub-system optimal solution. This can then help determine the system-level sensitivity for the compatibility constraint functions and therefore the direction of the next value of system-level variables is guided.

For example \( \frac{\Delta d^*_1}{\Delta \bar{x}_1^0} = -4 \) can be obtained after completing sub-system 1 optimisation, this implies that if \( \bar{x}_1^0 \) increases, \( d_1^* \) will decrease. Based on this sensitivity information, the system-level optimiser will increase the value of \( x_1^0 \) in order to reduce \( d_1^* \) in the next iteration that indicates the direction of the search. The first problem is how to choose a step size for the iterations that can help find optimal results within a reasonable time frame.

The second problem arises when there are conflicts between sub-systems. For example if \( \frac{\Delta d^*_1}{\Delta \bar{x}_1^0} = -4 \) and \( \frac{\Delta d^*_2}{\Delta \bar{x}_2^0} = 1 \), sub-system 1 will require an increase \( x_1^0 \) to reduce the value of \( d_1^* \), while sub-system 2 will require reducing \( x_2^0 \) to decrease the value of \( d_2^* \). Under this condition, the problem is how the system-level optimiser selects the value of interdisciplinary design variables for the next iteration.
2) **Local optimum:**

Sequential Quadratic Programming (SQP) was used for the optimisation at both the system and sub-system level in the Tappeta's work (1997). Such a gradient-based optimisation algorithm can result in a local optimum.

3) **Multi-objective problems:**

In the MOCO framework, weighting factors that reflect the relative importance of the different objective functions are used to change the multi-objective problem into a single-objective optimisation. In most multi-objective problems there is no single global solution, and these weighting factors are hard to define before optimisation.

4) **Delinquent nature of the MOCO formulation:**

After completing the system-level optimisation, the optimal value of $x_i^0$ has to satisfy the following Karush-Kuhn-Tucker (KKT) condition:

$$\frac{\partial F}{\partial x_i^*} + \lambda_1 \frac{\partial d_i^*}{\partial x_i^*} + \lambda_2 \frac{\partial d_i^*}{\partial x_i^*} = 0$$

Equation 5.6

In Equation 5.6, $x_i^*$ is the optimal result, which must be a feasible variable, thus system-level constraints are satisfied, namely $d_i^* = d_i^0 = 0$. In other words, Equation 5.7 is achieved:

$$x_i^* = x_i^0 = x_i^*$$

Equation 5.7

Hence both $\frac{\partial d_i^*}{\partial x_i^*} = -2(x_i^* - x_i^0)$ and $\frac{\partial d_i^*}{\partial x_i^*}$ are equal to zero. Thus the KKT condition presented in Equation 5.6 becomes $\frac{\partial F}{\partial x_i^*} = 0$ no matter what the value of the Lagrange multipliers ($\lambda_1$ and $\lambda_2$) and what the system-level objective functions are. This can become the main source of computational difficulties because the Lagrange multipliers are not really computable when the values of variables become feasible at system level. Sequential quadratic programming (SQP) and other gradient-based optimisation algorithms that employ such Lagrange multipliers in their merit functions have serious difficulty in making a decision about the termination of optimisation process. In other words, the MOCO formulation causes the KKT
conditions not to work, causing it to stop at any points without reaching the optimum (Alexandrov and Lewis, 2000; Lin, 2004).

5.5 Pareto GA based on CO Framework Development

Based on the understanding of the MOCO framework and associated problems, some improvements to this framework are made to suit the application of the Pareto-based genetic algorithm for the multi-objective problem in the system level. The first part of this section explains the reason for the use of the Pareto-based GA, the next part describes the COPGA framework.

5.5.1 Reasons for Use of the Pareto-based GA

In order to overcome the difficulties with the MOCO framework presented in Section 5.4.2, this study proposes the application of a Pareto-based genetic algorithm in the system level. The reasons are:

1) Global optimisation algorithm

The genetic algorithm is the best-known global optimisation. If this algorithm is used for the system-level optimisation, both system and sub-system levels avoid being trapped in a local optimum, even if there is a gradient-based optimisation algorithm used in the sub-system level, because target values of interdisciplinary variables decided by system-level significantly influence the sub-system-level optimisation through the objective function of the sub-system level.

2) Multi-objective approach

In the MOCO framework, the approach of the weighting method is used to solve multi-objective problems. This approach requires determining the weighting factors for different objectives before performing optimisation. Normally values of these factors are more subjective depending on the decision-maker, and these values are hard to be decided because changes often happen in the process of building design, for example, clients set a bigger value of weighting factor for running cost against capital cost; however clients may change these values during the design process.
However Pareto optimality can provide several optimal solutions with different sets of weighting factors in a single design process.

3) Delinquent nature of the MOCO formulation

As the previous section discussed, the MOCO formulation results in the invalidation of the KKT condition, which is a terminal criterion in most gradient-based optimisation algorithms, such as SQP. The terminal criterion of GA is often to satisfy the predefined number of generations, hence avoiding this problem.

The above three reasons justify the use of Pareto-based GA algorithm in the system-level optimisation of the MOCO framework. However this algorithm has some difficulties in handling constraints, for example when it adopts penalty function strategies to transfer a constrained problem to an unconstrained problem, problems arise due to fact that the fitness assignment is usually based on the Pareto rank of a solution, not on its objective function values (Jimenez et al., 2002). Hence this study proposes the two-cycle framework, internal and external, where the MOCO's system-level optimisation can be regarded as an unconstrained problem. The detailed description of this framework is presented as follow.

**5.5.2 The Conceptual COPGA Framework**

The COPGA framework to be developed consists of two cycles, namely internal and external, and there are both system and sub-system levels in the internal cycle. The internal cycle aims to transfer the stochastic values of interdisciplinary variables including shared and coupling variables to feasible values through a two-level framework. The process of the Pareto-based genetic algorithm is completed in the external cycle in order to evaluate the original design objective functions (e.g. minimise capital cost) and generate new stochastic values of interdisciplinary variables for running the internal cycle again, while those constraint functions in the system level of the MOCO are added up, becoming the single objective function in the system level of the COPGA.

Here 'stochastic' means that the values are chosen within the bounds of each variable at random; 'feasible' means that the value of variables can satisfy all design
constraints (e.g. allowance for shear force of beam). The processes used in the two cycles will be explained in the next section.

5.5.2.1 Internal Cycle of the COPGA Framework

In the internal cycle, like the MOCO framework, auxiliary design variables (i.e. $X^0$) are introduced to replace the shared and coupling variables. The values of these auxiliary variables are also constant in the course of sub-system-level’s optimisation while they could be adjusted by the system-level optimiser. The sub-system-level optimiser can adjust the values of interdisciplinary variables in the corresponding sub-system (e.g. $X_i^1$ and $X_i^2$). An important feature in the COPGA framework is a compatible objective function (e.g. $d_i^1 + d_j^2$) is introduced at the system level in order to achieve consistency in the design process. If this objective function can be minimised to zero, a compatible solution is obtained. Here ‘compatible’ means that there is an equal value with regard to the same variable (e.g. $X_i^1 = X_i^2 = X_i^0$). The mathematical formulations of system and sub-system levels are presented in Figure 5.3.

![Diagram of Internal Cycle Formulation](image-url)

Figure 5.3: Formulation of the Internal Cycle
Where,

\( d_i : \) i\textsuperscript{th} sub-system-level objective

\( X^0 : \) Vector of system level-variable (i.e. shared and coupling variables), namely interdisciplinary variable

\( X_{sh} : \) Vector of shared variable

\( Y_j : \) Vector of coupling variable, namely i\textsuperscript{th} sub-system send value to j\textsuperscript{th} sub-system.

\( X^0_{LB}, X^0_{UB} : \) Vector of system-level variable's lower and upper bound

\( X^i_{LB}, X^i_{UB} : \) Vector of variable's lower and upper bound in i\textsuperscript{th} sub-system-level

\( g_i : \) Vector constraints in the of i\textsuperscript{th} sub-system

\( X^i : \) Vector of variables in the of i\textsuperscript{th} sub-system

\( X_{local} : \) Vector of local variables in the of i\textsuperscript{th} sub-system

\((\bullet)^0 : \) Variable in system level

\((\bullet)^i : \) Discipline-specific variables in the i\textsuperscript{th} sub-system level

\((\bullet)^* : \) Optimal value of variable

\((\bullet) : \) Target value of interdisciplinary variable sent from system level to sub-system level

As Figure 5.3 shows, the internal cycle starts from receiving stochastic values of interdisciplinary variables, and then these are sent to the sub-system level. After completing the optimisations of each sub-system, the optimal values of interdisciplinary variables (e.g. \( X^i_{sh}, Y_j^i \)) are generated by varying discipline-specific variables (e.g. \( X^i \)), satisfying constraints of sub-system level (e.g. \( g_i \)). These optimal values are sent up to system-level. Based on this information from the sub-systems, the system-level optimiser will determine the new values of interdisciplinary variables (i.e. \( X^0 \)) to reduce the compatible function (i.e. \( \sum_{i=1,n} d^*_i (X^0) \)). Re-optimisations in the sub-system level are followed in accordance with these new target values (i.e. \( \bar{X}^0 \)). It is obvious that the system-level optimiser needs to call the sub-system optimisation in each iteration at system level. The iterations between the sub-systems and system level cease when changes in the compatible objective function at the system level are less than the predefined value.
The sub-system-level optimisers are free to control the local variables in their own 
sub-system (i.e. $X_{\text{local}}$), and manage interdisciplinary variables (i.e. $x_{sh}$ and $y_{ij}$).
Although the system-level optimiser can also adjust the value of these 
interdisciplinary variables (i.e. $x_{sh}^{0}$ and $y_{ij}^{0}$), they are different parameters with the 
same physical meaning within the two levels. The target values of interdisciplinary 
variables decided by the system-level optimiser are fixed during the process of sub-
system-level optimisation, the optimal values of these (i.e. $x_{sh}^{*}$ and $y_{ij}^{*}$) decided by the 
sub-system-level optimiser are fixed during the process of the system-level's 
optimisation. Furthermore, both the objective functions at system and sub-system 
level have similar formulation. The difference is that $X_{sh}$ and $Y_{ij}$ are adjusted and 
$\bar{X}_{sh}$ and $\bar{y}_{ij}$ are constant parameters in the sub-system level, while $X_{sh}^{0}$ and $Y_{ij}^{0}$ are 
changed and $X_{sh}^{*}$ and $Y_{ij}^{*}$ are constant parameters in the system-level

5.5.2.2 External Cycle of the COPGA Framework

Although the internal cycle attempts to obtain compatible solutions, in some cases the 
compatible objective can’t be minimised to zero or near to zero. Hence one aim of the 
external cycle enables these incompatible solutions to be excluded in the next 
generation in the course of Pareto-based GA. Furthermore in the external cycle new 
stochastic values of the interdisciplinary variables are generated through Genetic 
operation, namely selection, crossover and mutation. The Non-dominated Sorting 
GA-II (NSGA-II) algorithm (Deb et al., 2002) is used in the external cycle. This 
algorithm is better than the simple Pareto-based GA algorithm (Fonseca and Fleming, 
1993) because it introduces the parameter of crowding distance (Definition in Section 
5.2.3). The process of the external cycle is described below:

1) Input interdisciplinary variables

The process starts with choosing the values of interdisciplinary variables at random 
within the bounds of these to create an initial population in the GA process. These are 
then used to start the internal cycle.

2) Undertake the internal cycle (See Section 5.5.2.1 for details)
3) Check for compatibility with interdisciplinary variables between disciplines

This stage identifies those interdisciplinary variables that have different values in various disciplines. If the cumulative compatible objective \( d_1^i + d_2^i \) is close to zero, the interdisciplinary variables have equal values in all disciplines; the design objective functions (e.g. minimise capital cost) are then calculated. For the other interdisciplinary variables, the corresponding objective functions are assigned arbitrary large values in order for their related variables to be excluded from the next generation.

4) Evaluate the objective function values

The objective functions in the external cycle are real design objectives, such as capital cost or annual running cost. Their values are assessed based on Pareto rank and crowding distance (for definitions refer to Section 5.1.3). The design solutions with the low Pareto rank and high crowding distance are regarded as the better ones. Every design solution is firstly assessed using the Pareto rank, for those design solutions with the same rank, the crowding distance is compared.

5) Check for the terminal criterion in the external cycle

The terminal criterion in this study is to complete a predefined number of generations. If this criterion is met, the external cycle will be terminated. Otherwise the external cycle goes to the next step, the genetic operators explained below.

6) Execute the genetic operators

The genetic operators define the new individuals called off-springs for the next generation. It starts by selecting two individuals at random in the current generation and then choosing the one with the lowest Pareto Rank or the highest crowding distance. This process is called tournament and is repeated until a predefined number of individuals is obtained, these are the parent individuals. Then the crossover performs an exchange of every chromosome between two randomly picked parent individuals resulting in two new (child) individuals. If the values of variables within the initial population do not include a set that achieves the optimal solution, then the mutation introduces new values into the population.
7) Repeat step 2 to 6 until the predefined number of generations is reached. In each
generation the offspring created in the external cycle are sent to internal one.

From the overview of the whole process of the COPGA framework, it is apparent that it incorporates the best features of several concepts.

- Coordination strategy. The COPGA framework simulates coordination approach in the MOCO framework. In this approach, both the auxiliary variable and compatible functions are imported to achieve the concurrent design and coordination among disciplines.

- Two-cycle framework. The equality constraint functions of system-level in the MOCO framework are added up, and then become the objective function of system-level in the internal cycle of the COPGA framework, thus the KKT conditions can work well in the internal cycle of the COPGA framework even using SQP algorithm. In addition, problems in the external cycle can be cast as unconstrained problems which are easy to implement using the Pareto-based GA algorithm.

- Capacity of multi-objective problem. The concept of Pareto optimality is used for multi-objective design problem in the COPGA framework. Compared with the MOCO framework, the COPGA framework can optimise design objectives with different units and magnitudes; it does not require determination of preferences for various objectives before implementing optimisation process.

- Robust design space search. The external cycle’s optimisation is performed by the use of the genetic algorithm, which have powerful ability to search in the design space due to operations such as crossover and mutation. In addition, the different initial values of interdisciplinary variables generated by the GA are sent from the external cycle to the internal cycle in the COPGA framework, which avoids being trapped in a local optimum even if the gradient-based optimisation algorithm is used in the internal cycle.
5.6 Pilot Study

In this section a pilot study is undertaken to apply the COPGA framework to the problem in Equation 5.3. In order to validate the COPGA framework, this mathematical test problem is also solved using an all-at-once (AAO) method. Comparisons between the COPGA and AAO results are presented.

5.6.1 The COPGA Formulation

The mathematical test problem in Equation 5.3 can be decomposed into two subproblems, including two shared variables (i.e. \(x_1\) and \(x_3\)), two coupling variable (i.e. \(y_{12}\) and \(y_{21}\)) and one local variable (i.e. \(x_2\)). There are two objective functions to be minimised (i.e. \(F_1\) and \(F_2\)). Based on requirements of the COPGA formulation, in the internal cycle, the system-level optimiser coordinates two sub-system problems through the minimisation of the compatible function \((d_1^* + d_2^*)\), which is a function of interdisciplinary variables (i.e. \(x^0_1, x^0_3, y^0_{12}, \text{ and } y^0_{21}\)). The optimisation model for the system level problem of the internal cycle is formally stated as:

Formulation of system level in the internal cycle:

Minimise: \(d_1^* + d_2^*\)

\[
d_1^*(X^0) = (x_1^* - x_1^0)^2 + (x_3^* - x_3^0)^2 + (y_{12}^* - y_{12}^0)^2
\]

\[
d_2^*(X^0) = (x_1^* - x_1^0)^2 + (x_3^* - x_3^0)^2 + (y_{21}^* - y_{21}^0)^2
\]

With respect to: \(X^0 = [x_1^0, x_3^0, y_{12}^0, y_{21}^0]\)

Bounds: \(-10 \leq x_1^0 \leq 10; 0 \leq x_3^0 \leq 10\)

In this case the system-level optimiser does not control \(x_2\), which is a local variable in sub-system 1, the value of this variable is sent with corresponding set of interdisciplinary variables to calculate objective function \(F_1\) when finishing the internal cycle.

Within the COPGA framework, the sub-system-level optimisers are free to control the interdisciplinary variables (e.g. \(x_1^1, x_3^1, y_{12}^1\) in the sub-system 1) and also retain control
of their local variables (e.g. \(x_2\) in the sub-system 1). It also receives target values for each interdisciplinary variable from the system level (i.e. \(\bar{x}_i, \bar{y}_{i2}\) and \(\bar{y}_{i2}\)). The sub-system optimiser's task is to meet the target values provided by the system optimiser as best as possible by varying the discipline-specific variables whilst satisfying local constraints. The optimisation models for the sub-system level of the internal cycle problem are formally stated as:

Formulation of sub-system 1 in the internal cycle:

Minimise: \(d_1(X') = (x_1' - \bar{x}_1)' + (x_2' - \bar{x}_2)' + (x_{i2}' - \bar{y}_{i2})'\)

With respect to: \(X' = [x_1', x_2', x_{i2}']\)

Inequality constraint: \(\frac{y_{i2}}{8} - 1 \geq 0\)

Equality constraint: \(x_1' + x_2 + x_{i2}' - 0.2\bar{y}_{i2} = y_{i2}'\)

Bounds: \(-10 \leq x_1' \leq 10; \ 0 \leq x_2 \leq 10; \ 0 \leq x_{i2}' \leq 10;\)

Formulation of sub-system 2 in the internal cycle:

Minimise: \(d_2(X^2) = (x_1^2 - \bar{x}_1)' + (x_2^2 - \bar{x}_2)' + (y_{21} - \bar{y}_{21})'\)

With respect to: \(X^2 = [x_1^2, x_2^2, y_{21}^2]\)

Inequality constraint: \(\frac{y_{21}}{10} - 1 \geq 0\)

Equality constraint: \(x_1^2 + x_2 - \sqrt{y_{21}} = y_{21}^2\)

Bounds: \(-10 \leq x_1^2 \leq 10; \ 0 \leq x_2^2 \leq 10;\)

The external cycle of the COPGA framework is to optimise the original objective functions, namely \(F_1\) and \(F_2\), the mathematical formulation of the external cycle optimisation is as follows:

Formulation of external cycle's optimisation

Minimise: \(F_1\) and \(F_2\)

\(F_1 = x_2^2 + x_3 + y_{12}; \ F_1 = -y_{12}\)
With respect to: \( x_1, x_3, y_{12}, y_{21} \)

Bound: \(-10 \leq x_1 \leq 10; \ 0 \leq x_3 \leq 0\)

Figure 5.4 shows the whole process of the COPGA framework for the problem in Equation 5.3. It includes data transfer between internal and external cycle, and inputs and outputs between the system and sub-system level in the internal cycle.

This process of the COPGA framework starts from the external cycle; values of interdisciplinary variables (i.e. \( x_1, x_3, y_{12} \) and \( y_{21} \)) are initialised and sent to the two sub-systems in the internal cycle. After completing optimisation at sub-system level, optimal values of interdisciplinary variable are sent to system level (e.g. \( x_1^* \) and \( x_3^* \)), the system-level optimiser decides new value for the interdisciplinary variables (i.e. \( x_1^0, x_3^0 \)) to minimise the objective function \( d_1^* + d_2^* \), then they are sent back to the two sub-systems to start the optimisation again. Such iterations among system and sub-system level are stopped when the change in the objective function \( d_1^* + d_2^* \) in successive two iterations is less than \( 10^{-5} \). Finally the values of \( d_1^* + d_2^* \) with the corresponding interdisciplinary variable are sent out from the internal cycle, this also means the internal cycle is completed.

The first step of the external cycle is to evaluate whether the design solutions are compatible using the value of \( d_1^* + d_2^* \). In this case if \( d_1^* + d_2^* \leq 0.01 \), it implies that it is a compatible design, then the external cycle’s objective functions are calculated (\( F_1 \) and \( F_2 \)); otherwise both \( F_1 \) and \( F_2 \) are assigned \( 10^3 \). After that, the objectives of \( F_1 \) and \( F_2 \) are assessed based on the principle of Pareto rank and crowding distance. Then the terminal criterion of the external cycle is evaluated, this is the predefined number of iteration, which is 60 generations in this study. If this criterion is not met, the GA operations including selection, crossover and mutation are followed. This step is to generate new stochastic values of interdisciplinary variables for conducting the internal cycle again.
Figure 5.4 The Mathematical Problem in the COPGA Framework
5.6.2 The COPGA Results

Sequential Quadratic Programming (SQP), which is the built-in the Matlab optimiser, is used for the optimisation in both system-level and sub-system-level in the internal cycle. SQP is a gradient-based optimisation algorithm, which is fast for small problems and produces a local optimal design. The non-dominated sorting GA-II (NSGA-II) algorithm is adopted for the external cycle's optimisation. All parameters of NSGA-II algorithm are decided on the basis of Deb's suggestions (2002). These parameters includes: number of generation=60; population size=60; crossover probability is below 0.9; mutation probability is below 0.25. The whole optimisation process was run in Matlab.

After completing 60 iterations in the external cycle, the Pareto frontier shown in Figure 5.5 is generated with regard to the problem in Equation 5.3. This curve of trade-off between two objectives (i.e. F₁ and F₂) indicates that a further decrease of F₁ will be at the expense of increasing F₂. Sixty feasible design solutions that are listed in Appendix Three make up this curve. Five representative solutions of the problem in Equation 5.3 is described in Table 5.1, they are selected from all design solutions based on the maximal or minimal value of objectives or variables.

![Figure 5.5 The COPGA Solutions of the Mathematical Test Problem](image-url)
Table 5.1 Five Representative Solutions of the Mathematical Test Problem

<table>
<thead>
<tr>
<th>Number of Solution</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$x_1$</th>
<th>$x_3$</th>
<th>$y_{12}$</th>
<th>$y_{21}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Max $F_1$; Min $F_2$; Max $x_1$; Max $y_{12}$; Max $y_{21}$)</td>
<td>112.918</td>
<td>-30.009</td>
<td>9.960</td>
<td>9.850</td>
<td>103.077</td>
<td>30.000</td>
</tr>
<tr>
<td>No.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Min $F_1$; Max $F_2$; Min $y_{21}$, Min $x_3$)</td>
<td>12.866</td>
<td>-10.011</td>
<td>2.357</td>
<td>4.719</td>
<td>8.123</td>
<td>10.011</td>
</tr>
<tr>
<td>No.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Min $x_1$)</td>
<td>18.2154</td>
<td>-13.757</td>
<td>1.149</td>
<td>9.695</td>
<td>8.475</td>
<td>13.757</td>
</tr>
<tr>
<td>No.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Max $x_3$)</td>
<td>43.3039</td>
<td>-21.003</td>
<td>5.234</td>
<td>10.000</td>
<td>33.304</td>
<td>21.003</td>
</tr>
<tr>
<td>No.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Min $y_{12}$)</td>
<td>16.149</td>
<td>-12.479</td>
<td>1.535</td>
<td>8.112</td>
<td>8.033</td>
<td>12.479</td>
</tr>
</tbody>
</table>

Although solutions from the internal cycle cannot be guaranteed to be compatible, the following example could be regarded as a compatible design because the values of the system-level objective function (i.e. $d_1^* + d_2^*$) is less than 0.01. This example demonstrates a collaborative convergence process. In this process starting values of $x_1$, $x_2$, $y_{12}$ and $y_{21}$ in system-level variable are 1, 2, 3 and 4; After completing the internal cycle, value of these variables become 2.2309, 4.6518, 9.7662 and 10.007.

Figure 5.6 System-level Co-ordination of Variable of $x_1$

Figure 5.6 shows discipline-specific convergence histories for 8 system-level iterations with respect to the variable of $x_1$ in this example. In the first iteration, the
system level proposes an $x_1$ value of 1. After meeting all of its discipline-specific constraints, the sub-system 1 agrees on a value of 1, while the sub-system 2 returns with a request to increase this value to 3.634. Based on information provided by the solution of each sub-system, the system-level optimiser insists on the $x_1$ value of 1. In this iteration sub-system 2 requests increasing the $x_1$ value to 3.382, while sub-system 1 still agrees on the value of 1. In the third iteration, the $x_1$ value is changed to 3.9286 by a system-level optimiser. In this case, the sub-system 1 returns a request of reducing this value to 2.5949, while the sub-system 2 requires decreasing the value to 3.5435. Such a process of negotiation keeps going to the eighth iteration. In this iteration, the system-level optimiser sends the value of $x_1$ as 2.309 to the sub-system level. With this value, both sub-systems are able to remain disciplinary constraints feasible while providing agreement on this value of $x_1$.

The negotiation process between two sub-systems with regard to $x_3$ is presented in Figure 5.7

![Figure 5.7 System-level Co-ordination of Variable of $x_3$](image)

5.6.3 Validation of the COPGA Formulation

This section aims to evaluate the COPGA framework by the use of the all-at-once (AAO) framework which a single-level MDO framework. Hence the first part of this section explains what the AAO framework is and demonstrates the AAO formulation about the problem in Equation 5.3. The next part is to compare the characterise and results of the COPGA and AAO formulations.
5.6.3.1 All-at-once (AAO) Formulation

The all-at-once (AAO) is one of the common single-level MDO formulations. The AAO is highly centralised. The system optimiser manages three kinds of design variables: the local, shared and coupling variables; and minimises the objective functions while satisfying constraints. Any sub-systems dependencies and interaction is addressed through an integrated analyse. For a given set of design variables, the integrated analysis returns constraints and objective values for evaluation by the optimiser. The AAO formulation of the problem in Equation 5.3 is illustrated in Figure 5.8.

Optimiser:
Minimize $F_1$ and $F_2$
Constraints: $g_1 \leq 0; g_2 \leq 0; h_1 = 0; h_2 = 0$
Bound: $-10 \leq x_1 \leq 10; 0 \leq x_2 \leq 10; 0 \leq x_3 \leq 10$
With regard to: $x_1, x_2, x_3, y_{12}, y_{21}$

Evaluator 1:
$g_1 = 1 - \frac{y_{12}}{8}$
$h_1 = y_{12} - x_1^2 - x_2 - x_3 + 0.2y_{21}$
$F_1 = x_1^2 + x_2 + y_{12}$

Evaluator 2:
$g_2 = 1 - \frac{y_{21}}{10}$
$h_2 = y_{21} - x_1 - x_3 - \sqrt{y_{12}}$
$F_2 = -y_{21}$

Figure 5.8 AAO Formulation of the Mathematical Test Problem

This AAO formulation is implemented by the use of NSGA-II in C code (Kanpur Genetic Algorithm Laboratory, 2005). These parameters of the multi-objective NSGA-II includes: number of generation=100; population size=100; crossover probability is below 0.9; mutation probability is below 0.25.

5.6.3.2 Comparisons between COPGA Results and AAO Results

The COPGA framework requires optimisers at both system and sub-system level in the internal cycle, and the external cycle. A considerable amount of effort and time is
spent formulating these optimisation problems and implementing the appropriate algorithm to solve them. All-at-Once (AAO) framework, on the other hand, has simple problem setups, since each disciplinary sub-system is only required to perform the analysis (no sub-system optimisation). A single optimiser controls all of the variables and deals with all of the constraints.

Table 5.2 Comparisons between the COPGA and AAO Formulation

<table>
<thead>
<tr>
<th>Formulation methods</th>
<th>Design Module</th>
<th>Number of variables</th>
<th>Number of constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPGA framework</td>
<td>External cycle design</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Internal cycle design</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>System-level design</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Sub-system 1 design</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Sub-system 2 design</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>All-at-Once framework</td>
<td>System design</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

As Table 5.2 shows, the numbers of variables and constraints managed by the different-level's optimisers in the COPGA framework are slightly less than those in system optimiser in the AAO framework for the problem in Equation 5.3. It seems the COPGA framework does not have many advantages over the AAO framework. This phenomenon emerges mainly because optimisations of the problem in Equation 5.3 at disciplinary level are simple, for example the number of local variables and discipline-specific constraints is not too large. Hence the COPGA framework is preferred for scalability and organisational reasons. This scalability feature refers to dimensionality and complexity of optimisation problems. The decomposition of the problem in the COPGA framework allows the large-scale problem to be solved. Furthermore the COPGA framework is more suited to the structure and culture found in most multidisciplinary design teams. Providing disciplinary experts with autonomy, more control and responsibilities, and more opportunities to contribute to the overall system integration and optimisation, the COPGA is a more accepted method.

Since the problem in Equation 5.3 is small, the results of the COPGA framework can be validated by comparing the results obtained from the AAO framework. Such a validation approach has been widely used in other MDO applications (Budianto, 2000;
Choudhary, 2004). Figure 5.9 illustrates the results from the AAO and COPGA formulation for the problem in Equation 5.3. The red points represent the Pareto frontier of the AAO results, while blue points stand for the Pareto frontier of the COPGA results. These blue points are not the same density as those red points because 60 populations are set to run the COPGA optimisation, while 100 populations are set in the AAO process. However, the two curves can match each other, which implies that the COPGA framework gives similar results to the AAO approach.

![Figure 5.9 Comparisons between COPGA and AAO Results](image)

### 5.7 Suggestions for a Future Case Study

The next stage of this research attempts to demonstrate the application of the COPGA framework in a real building design. Based on lessons from the pilot study in this chapter, some suggestions of establishing and implementing a design scenario are summarised as follow:

1) Issue of computation burden

The COPGA framework results in increasing computation consumption for two reasons. One is that implementation of the Pareto-based GA in the external cycle requires a large amount of computation time. The other is that optimisation in the internal cycle further demands a significant amount of computation time due not only to the difficult coordination problem in the system level, but also the constraints analysis in the sub-system level. Hence the design scenario is assumed to include
small number of variables, and the disciplinary analysis linking with sub-system optimisation should not be too complex.

2) Scope of design scenario

The COPGA framework is developed for multidisciplinary design problems. However each disciplinary designer different elements to analyse. For instance building services designers choose a room or zone to analysis owing to individual requirements in terms of indoor temperature, ventilation, lighting etc. While structural design is based on different structural units which can be a frame or a load bearing wall. Hence the design scenario developed facilitates the implementation of different disciplinary analysis simultaneously, such as a framed room.

3) Identifying the relationship between disciplines

The key step in implementing a COPGA framework is to classify variables into local, shared and coupling groups. The design structure matrix (DSM) is a possible approach that helps to identify these kinds of variable. However this approach is often used to reduce iterations among parameters or activities through adjusting their sequences, so changes in the tradition DSM should be made for the application of the COPGA framework.

5.8 Summary

This chapter presents the process of developing the COPGA framework. The COPGA framework is based on the MOCO framework; hence the formulation and problems with MOCO framework are studied. In order to respond to these problems (i.e. local optimum, multi-objective approach and delinquent of MOCO formulation), the NGSA-II algorithm, which is a combination of the Pareto optimality with genetic algorithm is proposed to perform optimisation in the system level of the MOCO framework. However owing to limitation of the NGSA-II algorithm (i.e. handling constraint functions), changes to the MOCO framework have been made. The COPGA framework consists of an internal and external cycle optimisation. The internal cycle simulated the two-level framework of the MOCO formulation. The differences are that the compatible constraint functions of the system-level in the
MOCO are added up, consequently becoming the single compatible objective function in the COPGA framework, while the original design objective functions (e.g. minimisation of capital cost) are handled in the external cycle of the COPGA framework. Thus the optimisation problem of the external cycle becomes non-constrained, which facilitates the use of the NGSA-II algorithm. After the establishment of the COPGA framework, a multi-objective multidisciplinary mathematical problem presented in Equation 5.3 is used to test this framework through comparing solutions with an AAO formulation. In addition to presenting the results from two formulations, their characteristics are also compared to demonstrate that the COPGA framework has advantages over the AAO framework in large-scale engineering problems. Finally in order to apply the COPGA framework for building design in the next stage of this research, some suggestions are made for developing a suitable design scenario in the aspect of computation issues, scope of design scenario, and approach of identifying interdisciplinary variables.
CHAPTER SIX: DESIGN SCENARIO

6.1 Introduction

This chapter explains how the COPGA framework is applied to a three-storey office building design scenario. Data in this design scenario are collected through inspection of design documents and interviews with designers. It is therefore a realistic representation of a design task and is sufficiently complex to demonstrate both merits and limitations of the COPGA framework in multi-objective multidisciplinary building design. Wherever required data was unavailable, appropriate assumptions are made based on common standards.

6.2 Design Scenario

A south-faced room on an intermediate floor in a three-storey office located in London is chosen as a design example; this is shown in Figure 6.1. A design structure matrix (DSM) is used to identify interdisciplinary variables among structural and HVAC design in this design example.

![Office Floor Plan](image)

Figure 6.1: Office Floor Plan
6.2.1 Design Structure Matrix

Several variables and parameters contain interdisciplinary dependencies, which greatly influence not only the final system design, but the response of the disciplinary design objective function as well. The parameter-based design structure matrix (DSM) method is adopted here, this is a graphical tool to aid the designer in organising and structuring the design synthesis process, and showing the relationships between the various disciplines involved in a design problem (Yassine and Braha, 2003; Pektas and Pultar, 2006). There are three basic blocks for describing relationships among disciplines in the DSM method: parallel, sequential and coupled, which are illustrated in Figure 6.2.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Parallel</th>
<th>Sequential</th>
<th>Coupled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph representation</td>
<td><img src="image" alt="Parallel Graph" /></td>
<td><img src="image" alt="Sequential Graph" /></td>
<td><img src="image" alt="Coupled Graph" /></td>
</tr>
<tr>
<td>DSM representation</td>
<td><img src="image" alt="Parallel DSM" /></td>
<td><img src="image" alt="Sequential DSM" /></td>
<td><img src="image" alt="Coupled DSM" /></td>
</tr>
</tbody>
</table>

**Figure 6.2. Three Configurations that Characterise a System in DSM Analysis**

(MIT, 2005)

The relationships between the 23 parameters for this design scenario are shown in Table 6.1. The elements in this table represent the original sequence of design, which is made up of a square matrix. Dependencies among variables are represented with the help of 'X' marks in the off-diagonal cells. In this study, one difference with the conventional parameter-based DSM is that all variables are partitioned by each discipline, thus the first row and column in Table 6.1 presents the disciplinary name. The other is that dependencies within the disciplinary design do not be marked, for example the calculation of a beam dimension requires the weight of the floor.

Reading across a row shows input variables; reading down a column shows output variables. Those interdisciplinary variables are presented with marks of 'X'. For
instance the marks in row O of Table 6.1 denote that variable O (i.e. heating gain from floor, wall and roof) requires information from variables F, H, I, and J (i.e. structural component material; floor, wall and roof dimension). If the design variables could be made in order of A through W, it will be desirable for all information required by each variable to have been already generated by a predecessor design variable. It can be seen in Table 6.1 that this is not the case for some of the variables. Variable H, for example, requires information from variable R, because position and size of heating equipment affect the value of live load on floor, which determined the depth of floor. In practice the value of variable R could not be available before generating the value of variable H.
Table 6.1 Application Parameter-based DSM in Design Scenario

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Architecture Design Variable</th>
<th>Structural Design Variable</th>
<th>HVAC Design Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture Design Variable</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Orientation of building</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height of floor-to-ceiling</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window size and type</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural system type</td>
<td>E</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>Structural component material</td>
<td>G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column layout</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Floor dimension</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wall dimension</td>
<td>J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roof dimension</td>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross-section of beam</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height of floor-to-floor</td>
<td>M</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Cross-section of column</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solar gain</td>
<td>N</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Heating gain from floor, wall and roof</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Heating gain from occupant and equipment</td>
<td>P</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HVAC system type</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heating equipment size and position</td>
<td>R</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Cooling equipment size and position</td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air condition-beam integration scheme</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air diffuse layout</td>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air duct size</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply air temperature</td>
<td>W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Although the DSM is very useful for identifying interdisciplinary variables, it is hard to read because too many local variables (e.g. size of beam and column) are mixed with interdisciplinary variables. Figure 6.4 presents the relationship among architecture, structural and HVAC design for this design scenario, while Figure 6.3 helps to read Figure 6.4.

![Figure 6.3 Definition of a Module Placed on N-square Diagram](image)

![Figure 6.4 Interdisciplinary Variables between Disciplines](image)

As Figure 6.4 shows, some variables are sent from architectural design to structural and HVAC design simultaneously, including space function, height of floor-to-ceiling, orientation of building and size of window. In practice these variables are fixed as a constant parameter at the beginning of structural and HVAC design. When structural and HVAC engineers cannot meet their own design constraints, they consult with the architect to change them.
In addition Figure 6.4 illustrates dependencies between structural and HVAC design. The HVAC designer, for example, requires the materials and dimensions of the structural components to calculate U values of wall, floor and roof. On the other hand, the structural designer requires information about weight and position of heating and cooling equipments, which influence the live loads on the floor. The variables of the air duct’s size and AC-beam integration scheme are also needed by structural designer, and are function of height of floor-to-floor.

6.2.2 Summary of All Information in Design Scenario

In order to simplify this design scenario, some variables are assigned a fixed value before starting optimisation, such as orientation of building, height of floor-to-ceiling. Finally this case study includes two objectives, six variables and ten constraints that are presented in Table 6.2. Table 6.3 describes constructional and occupancy information. In terms of analysis by the use of DSM, it is easy to divide variables related to this design scenario into shared, coupling and local variables between the structural and HVAC disciplines, there are listed in Table 6.4
Table 6.2: Objective, Variable and Constraint Information

<table>
<thead>
<tr>
<th>Objective</th>
<th>Variable</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) To minimise weight of column and beam ( F_1(x_1, x_2, x_3, x_4) )</td>
<td>1) Height of window ((x_1)) mm</td>
<td>1) Allowance for beam bending moment (g_1(x_1, x_2, x_3, x_4))</td>
</tr>
<tr>
<td>2) To minimise sum of peak cooling and heating load in summer and winter ( F_1(x_1, x_1) )</td>
<td>2) Depth of external wall ((x_2)) mm</td>
<td>2) Allowance for beam shear force (g_2(x_1, x_2, x_3, x_4))</td>
</tr>
<tr>
<td></td>
<td>3) Depth of air duct ((x_3)) mm</td>
<td>3) Allowance for beam deflection (g_3(x_1, x_2, x_3, x_4))</td>
</tr>
<tr>
<td></td>
<td>4) Depth of beam ((x_4)) mm</td>
<td>4) Allowance for column slenderness ratio (g_4(x_3, x_4, x_5))</td>
</tr>
<tr>
<td></td>
<td>5) Depth of column ((x_5)) mm</td>
<td>5) Allowance for cross-section capacity of column with moments in plan buckling (g_5(x_1, x_2, x_3, x_4, x_5))</td>
</tr>
<tr>
<td></td>
<td>6) Supply air temperature in summer ((x_6)) (\degree C)</td>
<td>6) Allowance for cross-section capacity of column with moments out-of-plan buckling (g_6(x_1, x_2, x_3, x_4, x_5))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7) Allowance for column axial force (g_7(x_1, x_2, x_3, x_4, x_5))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8) Allowance for depth of external wall (g_8(x_2, x_5))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9) Allowance for air change rate (g_9(x_1, x_2, x_6))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10) Allowance for air velocity in the duct (g_{10}(x_1, x_2, x_3, x_6))</td>
</tr>
</tbody>
</table>

Table 6.3: Constructional and Occupancy Details

<table>
<thead>
<tr>
<th>Item</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Wall (opaque)</td>
<td>105mm outer brickwork; (x_1) mm inner brickwork; 13mm lightweight plaster</td>
</tr>
<tr>
<td>Internal partition wall</td>
<td>13mm lightweight plaster; 105mm brickwork; 13mm lightweight plaster</td>
</tr>
<tr>
<td>Internal floor/ceiling</td>
<td>50mm screed, 150mm dense cast concrete, 25mm wood block; 16mm plasterboard ceiling (density 391.2kg/m², (U=1.5), (Y=2.9))</td>
</tr>
<tr>
<td>Window</td>
<td>Double glazed</td>
</tr>
<tr>
<td>Lighting</td>
<td>18.75 W/m² of floor area; in use 0900-1700h</td>
</tr>
<tr>
<td>Occupancy</td>
<td>Occupied 0900-1700 h by 6 people, 80W sensible heat output per person</td>
</tr>
<tr>
<td>Electrical equipment</td>
<td>Four computers of 150 W, in use 0900-1700h</td>
</tr>
<tr>
<td>Mechanical ventilation</td>
<td>10.5L/S fresh air per person</td>
</tr>
</tbody>
</table>
Table 6.4: Summary of All variables in the Design Scenario

<table>
<thead>
<tr>
<th>Type of variable</th>
<th>Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared variable</td>
<td>Height of window (x_1)</td>
<td>In structural design, window size is a function of self-weight of exterior wall, which affects dead load to beam and column. While in thermal design, it mainly affects heating loss in the room.</td>
</tr>
<tr>
<td>Coupling variable (input of HVAC from output of structural design)</td>
<td>Depth of wall (x_2)</td>
<td>Depth of wall is decided during structural design, but calculation of U value of wall needs this variable in thermal design</td>
</tr>
<tr>
<td>Coupling variable (input of structural from output of HVAC design)</td>
<td>Depth of ventilation duct (x_3)</td>
<td>The services engineer will adjust depth of duct to satisfy noise requirement, the structural designer will calculate the height of floor-to-floor based on this variable</td>
</tr>
<tr>
<td>Structural Local variable</td>
<td>Cross-section of beam and column (x_4, x_5)</td>
<td>This is a local variable in structural design. It is adjusted to satisfy structural strength, stability and so on.</td>
</tr>
<tr>
<td>HVAC local variable</td>
<td>Supply air temperature in summer (x_6)</td>
<td>This is a local variable in the HVAC discipline to control supply air rate.</td>
</tr>
</tbody>
</table>

The calculation of this case is based on the following assumptions:

1) The structural calculation is based on the combination dead loads and live loads. Both wind loads and seismic loads are not considered.

2) All structural components are calculated based on elastic analysis.

3) There is the same height of floor-to-floor in the every story.

4) Density of steel is \(7850 \text{ kg/m}^3\)

5) The dry resultant temperature in adjoining room is equal; hence heat flow occurs only through the outside window and wall.

6) The office is located in the centre of London, so the window must be closed all day to avoid traffic noise. The air infiltration rate is equivalent to 1 air change per hour. The mechanical ventilation is based on a minimum fresh air requirement per person.
7) With regard to the AC-beam integration scheme, this case adopts separation of service and structure zones that is presented in Figure 6.5.

8) Value of thermal admittance (Y-value) must not be changed because for multi-layered structure, the admittance is primarily determined by the characteristic of materials in the layers nearest to the internal surface.

9) The peak-heating load is assumed to happen at 12 at noon in January, and the peak-cooling load happens in July.

![Figure 6.5 Separation of Structural and Services Zone](image)

6.3. Analysis Model

Disciplinary analysis is required by each sub-system optimiser in the COPGA framework. This section describes the analysis method adopted for this design example.

6.3.1 Structural Analysis

There are numerous ways to analyse statically indeterminate structures in building design. The statically indeterminate structure is one where the static equilibrium equations are not sufficient for determining the internal forces and reaction on that structure (Wikipedia, 2007). Buick and Graham (2003) summarise the differences
among these classical methods including force method, slope-deflection methods, moment-distribution method, and unit load method.

The force method is a method for calculating the response of statically indeterminate structures by which unknowns are force quantities (the redundant force $X_1, X_2, \ldots, X_n$) and the equations used to solve the unknowns are based on geometrical conditions (compatibility condition at the location of each redundant force). While in displacement method, unknowns are displacement quantities and the equations used to solve the unknowns are based on statically conditions (equilibrium conditions).

In this study, a framed structural system is adopted. The connections of beam-to-column are constrained in bending moment while the connections of column-to-base are fixed-end, thus the displacement method is better than the force method because it has less unknown variables. In addition, the Matrix-Displacement method developed from the displacement method must be used because of the large number of structural elements to be calculated, this is a computer-based analysis method. Herein steps of this matrix-displacement method are outlined (Buick and Graham, 2003):

1) Calculate element stiffness matrix in local coordinates ($[k']$)

$$[k'] = \begin{bmatrix}
\frac{EA}{l} & 0 & 0 & \frac{-EA}{l} & 0 & 0 \\
0 & \frac{12EI}{l^3} & \frac{6EI}{l^2} & 0 & \frac{-12EI}{l^3} & \frac{6EI}{l^2} \\
0 & \frac{6EI}{l^3} & \frac{4EI}{l^2} & 0 & \frac{-6EI}{l^3} & \frac{2EI}{l^2} \\
\frac{-EA}{l} & 0 & 0 & \frac{EA}{l} & 0 & 0 \\
0 & \frac{-12EI}{l^3} & \frac{-6EI}{l^2} & 0 & \frac{12EI}{l^3} & \frac{-6EI}{l^2} \\
0 & \frac{6EI}{l^2} & \frac{2EI}{l} & 0 & \frac{-6EI}{l^2} & \frac{4EI}{l}
\end{bmatrix}$$

2) Calculate element stiffness matrix in global coordinates ($[k']$)

$$[k'] = T^T[k'][T]$$

3) Assemble element stiffness matrix ($[k']$) in the global stiffness matrix ($[K]$)
4) Calculate element load vector ([\( p' \)]) in the local coordinates, assuming full fixity at the joints of each element

5) Calculate element load vector in the global coordinator

\[ [p'] = T^T[p'] \]

6) Assemble element load vector ([\( p' \)]) in the global load vector ([\( P \)])

7) Generate an equation of moment equilibrium at each joint;

\[ \{P\} = [K] \cdot \{\Delta\} \]

8) Solve the system of equations for the unknown joint displacements (\( \Delta \));

\[ [\Delta] = [K]/[P] \]

9) Calculate element force vector ([\( F' \)]) using the expression derived in step 2 and the values of joint displacements calculated in Step 8;

\[ [F'] = [k'] \cdot [\Delta]' \]

The processes of the matrix-displacement method are coded in Matlab and saved in plane_gangjia.m file.

### 6.3.2 Validation of Structural Analysis

The process of validation starts with the analysis of a two-span-two-storey steel framed structural problem shown in Figure 6.6 using the STAAD pro, which is a mature commercial structural analysis software (STAAD pro. 2007), this problem is then solved using the method described in the previous section. Finally the results obtained from the two methods are compared.
In this test problem, there are 9 nodes and 10 structural elements that are marked with the numbers in Figure 6.6. A uniform dead load is applied to each beam, whose value is 20N/mm. Both UB762×267×173 and UC254×254×89 are adopted for the cross-section of beam and column respectively. The elastic modulus (E) is equal to 205KN/mm². The self-weight of column and beam are not calculated during the process of analysis. In addition, all beams have the same size of cross-section, so do the columns, thus these critical positions, at which the maximal axial or shear force or bending moment emerges, determines the model of cross-section. Table 6.5 presents the comparison of the results from STAAD Pro and those obtained from a programme based on the matrix-displacement method written by the author.
Table 6.5 Comparison of Results of Structural Analysis

<table>
<thead>
<tr>
<th>Position</th>
<th>Result of STAAD pro</th>
<th>Result of program based on matrix-displacement method</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Axial force (N)</td>
<td>Element 8 and 10</td>
<td>5030</td>
<td>5190</td>
</tr>
<tr>
<td>Shear force (N)</td>
<td>End of element 10</td>
<td>6910</td>
<td>6938</td>
</tr>
<tr>
<td>Bending moment (kNm)</td>
<td>End of element 7</td>
<td>68.615</td>
<td>70.391</td>
</tr>
<tr>
<td>Column</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Axial force (N)</td>
<td>Element 3</td>
<td>276000</td>
<td>276566</td>
</tr>
<tr>
<td>Shear force (N)</td>
<td>Element 2 and 6</td>
<td>5030</td>
<td>5190</td>
</tr>
<tr>
<td>Bending moment (kNm)</td>
<td>End of element 2</td>
<td>10.979</td>
<td>11.321</td>
</tr>
</tbody>
</table>

These differences may stem from two reasons. One is that the STAAD Pro software is based on the finite element analysis, which is more accurate compared with the matrix-displacement method. The other reason is it is plastic analysis in the STAAD-pro software. However these differences between the two methods of analysis in Table 6.5 can be accepted.

6.3.3 Thermal Analysis

When sizing heating systems it is normally sufficient to consider steady-state conditions with an allowance for intermittent operation. Whereas for the purpose of calculating cooling loads in the summer time, it is essential to take account of the building dynamics and hence non-steady climatic conditions (CIBSE Guide A, 1999).

Steps for the steady-state heating loss calculation using dry resultant temperature in the wintertime are listed (CIBSE Guide A, 1999):

1) Heating loss from infiltration and mechanical ventilation ($Q_v$)
2) Heating loss from structural fabric ($Q_f$)
3) Total heating loss ($Q_t$)

$$Q_t = F_1 Q_f + F_2 Q_v$$

Note: both $F_1$ and $F_2$ are factors related to the characteristics of the heat source. For building with an average external $U$ value in the range 0.60-3.0 W/m²K including openings, which covers the majority of habitable structures, value of these factors are
\( F_1 = 1.00 \) and \( F_2 = 1.10 \) for a panel radiator heating system; \( F_1 = 0.92 \) and \( F_2 = 1.23 \) for a forced warm air heating system (with an accuracy of 5.0\%) (Chatterton, 1994, p58). In this study, the forced warm air heating system is adopted.

For calculations of air conditioning cooling load the process is (CIBSE Guide A, 1999):

1) Solar gain through glazing (\( Q_{sg} \))

2) Mean fabric gain at air node (\( \bar{Q}_{fa} \))

3) Cyclic conduction gains at air node (\( \tilde{Q}_{fa} \))

4) Internal gains (\( Q_{con}, Q_{rad} \))

5) Infiltration and mechanical ventilation gain (\( Q_v \))

6) Total sensible cooling load (\( Q_k \)): \( Q_k = \bar{Q}_a + \tilde{Q}_a + Q_{sg} + Q_v \)

Where:
\[
\bar{Q}_a = \bar{Q}_{fa} + F_{w} 1.5 \sum Q_{rad} + \sum Q_{con} - 0.5 \sum Q_{rad}
\]
\[
\tilde{Q}_a = \tilde{Q}_{fa} + F_{w} 1.5 \sum Q_{rad} + \sum Q_{con} - 0.5 \tilde{Q}_{rad}
\]

6.3.4 Validation Results of the Thermal Analysis

A south-facing room which is 50\% glazed in an intermediate floor is used as an example to apply the thermal analysis. The building is located in central London. The dimensions of this room are \( 6 \times 4.5 \times 3.6 \)m. Thermal transmittances (U-value) and admittance (Y-value) for each structural component are described in Table 6.6. The schedule of occupancy and office equipment whose heating gain is 10W/m\( ^2 \) is from 9am to 5pm, the schedule of lighting whose heating gain is 18.75W/m\( ^2 \) is from 7am to 7pm. The Dry-resultant indoor temperature is kept around 21\( ^\circ \)C. The extreme outside air temperature in winter is -4 \( ^\circ \)C. This example is analysed using both the method given by CIBSE and DesignBuilder software. The DesignBuilder software was developed to run EnergyPlus input files and display results graphically (DesignBuilder, 2007). Comparison of the results obtained from the two methods are presented in Table 6.7
Table 6.6 U and Y values in the thermal analysis example

<table>
<thead>
<tr>
<th>Surface</th>
<th>Area (m²)</th>
<th>U value (W/m²K)</th>
<th>(A×U) (W/K)</th>
<th>Y value (W/m²K)</th>
<th>(A×Y) (W/K)</th>
<th>Decrement Factor</th>
<th>Time lag (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Wall (opaque)</td>
<td>10.8</td>
<td>0.537</td>
<td>5.80</td>
<td>3.7</td>
<td>39.96</td>
<td>0.26</td>
<td>10</td>
</tr>
<tr>
<td>Internal partitioned wall</td>
<td>54</td>
<td>1.16</td>
<td>2.4</td>
<td>129.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal floor</td>
<td>27</td>
<td>4.24</td>
<td>3</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ceiling</td>
<td>27</td>
<td>4.24</td>
<td>1.4</td>
<td>37.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>10.8</td>
<td>2.758</td>
<td>29.78</td>
<td>3.2</td>
<td>34.56</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>∑</td>
<td>129.6</td>
<td>35.58</td>
<td>322.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Peak heating load calculation in winter:**

**Step One:** heat loss from infiltration and mechanical ventilation (Qv)

For an office, the air infiltration rate in winter is equal to 1/h (CIBSE Guide A, 1999)

\[
C_{v1} = \frac{NV}{3} = \frac{6 \times 4.5 \times 3.6 \times 1}{3} = 32.4 \text{ W/K}
\]

There are two people in the office, mechanical ventilation requires the minimum fresh air per person to be 10.5l/s

\[
C_{v2} = \frac{10.5 \times 10^{-3}}{\frac{1}{3} \times 2} = 25.2 \text{ W/K}
\]

\[
C_v = C_{v1} + C_{v2} = 32.4 + 25.2 = 57.6 \text{ W/K}
\]

\[
Q_v = C_v(t_c - t_w) = 57.6 \times (22 + 4) = 1440 \text{ W}
\]

**Step Two:** heat loss from structural fabric (Qf)

\[
Q_f = (\sum A \times U) \times (t_c - t_w) = 35.58 \times (22 + 4) = 889.5 \text{ W}
\]

**Step Three:** total heat loss (Qt)

\[
Q_t = F_1Q_f + F_2Q_c = 0.92 \times 889.5 + 1.23 \times 1440 = 2589 \text{ W}
\]

**Peak cooling load calculation in summer:**
Before the calculation of the peak cooling load is undertaken, the response factor \( f_r \) that reflects building response to changes in the environmental temperature is calculated because it affects solar gains.

\[
    f_r = \frac{\sum (AY) + C_v}{\sum AU + C_v} = \frac{322.92 + 32.4}{35.58 + 32.4} = 5.22
\]

This building is a slow-response building because of \( f_r > 4 \) (CIBSE Guide A, 1999).

**Step one:** solar gain through glazing \( Q_{sg} \) (the mean total solar irradiance for a south facing surface in July is 238W/m², correction factor for a slow-response building with shading is 0.62).

\[
    Q_{sg} = 0.62 \times 238 \times 10.8 = 1593W
\]

**Step two:** mean fabric gain at air node \( \bar{Q}_{fa} \)

Transfer coefficients

\[
    F_{cu} = \frac{3(C_v + 6\sum A)}{\sum AU + 18\sum A} = \frac{3(32.4 + 6 \times 129.6)}{35.58 + 18 \times 129.6} = 1.026
\]

\[
    F_{cy} = \frac{3(C_v + 6\sum A)}{\sum AY + 18\sum A} = \frac{3(32.4 + 6 \times 129.6)}{322.92 + 18 \times 129.6} = 0.915
\]

The mean solar-air and air temperature are:

Opaque wall (south facing, light) \( \tilde{t}_{so} = 22.6^\circ C \)

Window \( \tilde{t}_{so} = 19.6^\circ C \)

Mean gain for a mean dry resultant temperature of 22 \(^\circ\)C

\[
    \bar{Q}_{fa} = F_{cu} \sum (AU)(\tilde{t}_{so} - \bar{t}_c)
    \]

\[
    (\bar{Q}_{fa})_{wall} = 1.026 \times 5.8 \times (22.6 - 22) = 3.57W
\]

\[
    (\bar{Q}_{fa})_{window} = 1.026 \times 29.78 \times (19.6 - 22) = -73.3W
\]

\[
    \bar{Q}_{fa} = 3.57 - 73.3 = -69.73
\]
Chapter 6  Design Scenario

**Step Three:** cyclic conduction gains at air node ($\bar{Q}_{fa}$)

The sol-air and air temperatures are obtained from external design data for the time of day corresponding to 1230h minus the time lag appropriate to the decrement factor for the surface. $\bar{Q}_{fa} = F_a \sum (AU) f_{rec(t-d)}$

For wall (hour-ending 0300h) $\bar{t}_{wo} = (10.9 - 22.6) = -11.7^\circ C$

For window (hour-ending 1300h) $\bar{t}_{wo} = (24.2 - 19.6) = 4.6^\circ C$

$\bar{Q}_{fa}\text{wall} = 0.915 \times 5.8 \times 0.26 \times (-11.7) = -16.14\text{W}$

$\bar{Q}_{fa}\text{window} = 0.915 \times 29.78 \times 1 \times 4.6 = 112.7\text{W}$

$\bar{Q}_{fa} = -16.14 + 112.7 = 96.56\text{W}$

**Step Four:** internal gains ($Q_{con}, Q_{rad}$)

The convective and radiant components of the mean internal gain for an 8-hour occupancy are as follows:

$Q_{con} = (40 \times 2.25 + 168 \times 2 \times 0.76) \times (8/24) + (6 \times 4.5 \times 18.75 \times 0.41) \times (12/24) = 218.9\text{W}$

$Q_{rad} = (40 \times 2.25 + 168 \times 2 \times 0.24) \times (8/24) + (6 \times 4.5 \times 18.75 \times 0.59) \times (12/24) = 206.2\text{W}$

Hence the convective and radiant components of the swing in internal gain (i.e. value at the calculation hour minus mean value) are as follows:

$Q_{con} = (40 \times 2.25 + 168 \times 2 \times 0.76 + 6 \times 4.5 \times 18.75 \times 0.41) - 218.9 = 334.02\text{W}$

$Q_{rad} = (40 \times 2.25 + 168 \times 2 \times 0.24 + 6 \times 4.5 \times 18.75 \times 0.59) - 206.2 = 469.24\text{W}$

**Step Five:** Infiltration gain and mechanical ventilation gain ($Q_v$)

**Infiltration gain**

$Q_{vi} = C_v (t_{wo} - t_e) = \frac{6 \times 4.5 \times 3.6}{3} \times (27 - 22) = 162\text{W}$

**Mechanical ventilation gain:**

There are two people; the minimum fresh air per person is 10.5l/s

$Q_{v2} = \frac{10.5 \times 10^{-3}}{1} \times \frac{1}{3} \times 2 \times (27 - 22) = 126\text{W}$

$Q_v = Q_{vi} + Q_{v2} = 162 + 126 = 288\text{W}$
total sensible cooling load \( (Q_k) \)

\[
Q_k = Q_{ja} + F_{g} + F_{r} - 0.5 \sum Q_{rad} - 0.5 \sum Q_{con} \\
= -69.73 + 1.026 \times 1.5 \times 206.2 + 218.9 - 0.5 \times 206.2 = 363.41
\]

\[
Q_a = Q_{ja} + F_{g} + F_{r} - 0.5 \sum Q_{rad} \\
= 95.56 + 0.915 \times 1.5 \times 469.24 + 334.02 - 0.5 \times 469.24 = 839.02
\]

\[
Q_k = Q_a + Q_{sg} + Q_v \\
= 363.41 + 839.02 + 1593 + 288 = 3083W
\]

<table>
<thead>
<tr>
<th>Peaking heating load (in January)</th>
<th>Results calculated manually</th>
<th>Results calculated by DesignBuilder software</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peaking cooling load (in July)</td>
<td>2589W</td>
<td>2517W</td>
<td>2.86%</td>
</tr>
<tr>
<td></td>
<td>3083</td>
<td>2920</td>
<td>5.58%</td>
</tr>
</tbody>
</table>

The main cause for such differences originates from the different analysis methods. The analysis method adopted in DesignBuilder follows the American society of heating, refrigerating and air-conditioning engineers (ASHRAE) method.

### 6.4 Analysis Formulation

The COPGA framework is a mathematical model; so all the analysis models must be presented in mathematical form. This section formulates the structural and HVAC analysis of this design scenario based on the two analysis methods explained in the previous section.

#### 6.4.1 Structural Analysis Formulation

For a simple steel structure analysis it is based on the behaviour of individual members, these members could be the form of a column, beam, wall or floor slab. Under incremental loading, this deformation or displacement response of a steel member is the process of elastic stage, elastic-plastic stage, plastic stage and final collapse. The structural design in this scenario assumes the response of member occur
elastic stage. The whole procedure is guided by BS 5950 code and steel designer’s manual (Buick and Graham, 2003).

The structural system in this design is a framed system with a fixed connection between column and foundation and semi-rigid connections between column and beam. All cross-sections adapt S275 rolled I-section. Figure 6.8 shows the cross-section’s dimension.

![Figure 6.7 Dimensions of Cross-section](image)

6.4.1.1 Beam Design

The use of S275 steel, no greater than 40mm thick, takes $p_y = 265N/mm^2$ $D = x_4 mm$, $B = \frac{x_4}{3} mm$, because beam calculation is based on elastic analysis, $9\varepsilon < b/T < 13\varepsilon$ is required ($\varepsilon = \sqrt{275/p_y} = 1.018$). The value of $b/T$ is assumed to be 10. Hence $T = \frac{x_4}{60} mm$. In addition the value of $d/t$ is assumed to be 40, hence $t = \frac{29x_4}{1200} mm$.

Beam Load Calculation:

1) Dead load=floor self-weight + wall self-weight
   
   $8.802 + (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-5} N/mm$

2) Live load: 5.625N/mm

3) Combination load=1.4 Dead load+1.6 Live load
\[ w = 21.323 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-3} \text{ N/mm} \]

4) Top floor load
\[ 1.4 \times 8.802 + 1.6 \times 1.05 \times 4.5 = 19.8828 \text{ N/mm} \]
\[ 21.77 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-5} \text{ N/mm} \]

Properties of Beam Cross-section Calculation:

1) Moment of inertia of section (I)
\[ I_x = \sum \left( \frac{bh^3}{12} + Ad^2 \right) = 2 \times \left[ \frac{x_4}{3} \times \left( \frac{x_4}{60} \right)^3 + \frac{x_4}{3} \times \frac{x_4}{60} \times \frac{x_4}{2} \times \frac{x_4}{2} \right] + \frac{29x_4}{1200} \times \frac{x_4 - x_4}{30} = 4.51 \times 10^{-3} x_4^4 \text{ mm}^4 \]
\[ I_y = \sum \left( \frac{bh^3}{12} \right) = 2 \times \frac{x_4}{60} \times \frac{\left( \frac{x_4}{30} \right)^3}{12} + \frac{29x_4}{1200} \times \frac{\left( \frac{29x_4}{30} \right)^3}{12} = 1.03 \times 10^{-4} x_4^4 \text{ mm}^4 \]

2) Elastic section module of strong axis (Z)
\[ Z = I \frac{c}{x_4} = \frac{4.51 \times 10^{-3} x_4^4}{x_4/2} = 9.02 \times 10^{-3} x_4^3 \text{ mm}^3 \]

3) Plastic section module of strong axis (S)
\[ S = \sum Ad = 2 \times \left[ \frac{x_4}{3} \times \frac{x_4}{60} \times \frac{59x_4}{120} + \frac{29x_4}{60} \times \frac{29x_4}{1200} \times \frac{29x_4}{120} \right] = 0.011x_4^3 \]

4) Gross section area: \( A = 2 \times \frac{x_4}{3} \times \frac{x_4}{60} + \left( \frac{x_4}{30} \right) \times \frac{29x_4}{1200} = 0.034x_4^2 \text{ mm}^2 \]

5) Bending strength (\( p_b \))
\[ r_y = \sqrt{\frac{I}{4A}} = \sqrt{\frac{1.03 \times 10^{-4} x_4^4}{0.034x_4^2}} = 0.055x_4 \]
\[ \lambda = \frac{L_E}{r_y} = \frac{9000}{0.055x_4} = \frac{1.63 \times 10^5}{x_4} \]
\[ \lambda_{LT} = \frac{0.9 \times 0.905\lambda}{\left( 1 + 0.05(\lambda/60)^2 \right)^{0.25}} \]
\[ p_b = \frac{P_E}{\frac{1}{\phi_{LT} + (\phi_{LT} - P_E)^{0.5}}^{0.5}} \]
In which
\[ p_g = \left( \frac{\pi^2 E}{4 \lambda_{LT}^2} \right) \]
\[ \phi_{LT} = p_y + \left( \frac{\eta_{LT} + 1}{2} \right) p_e \]

6.4.1.2 Column Design

The use of S275 steel, no greater than 16mm thick, takes \( p_y = 275N/mm^2 \), \( D = x_s \), \( B = x_t \), because column calculation is based on elastic analysis, \( 9\varepsilon < b/T < 13\varepsilon \) is required \( \varepsilon = \sqrt{275/p_y} = 1 \). Value of \( b/T \) is assumed to be 10. Hence \( T = \frac{x_s}{20} mm \). In addition Value of \( d/t \) is assumed to be 15, hence \( t = \frac{3x_t}{50} mm \). 

Column Load Calculation:

1) Force from top column:
\[ 7.585 \times 0.154 x_s^2 \times (3000 + x_3 + x_4 + 100) \times 10^{-5} + 95952 \]

2) Total axial force:
\[ [95953 + 0.063 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3)] \times 2 + 95952 + 2 \times 7.585 \times 0.154 x_s^2 \times (3000 + x_3 + x_4 + 100) \times 10^{-5} N \]

Property of Column Cross-section Calculation:

1) Moment of inertia of section
\[ I_x = \sum \left( \frac{bh^3}{12} + Ad^2 \right) = 2 \times \frac{x_s \times \left( \frac{x_s^2}{20} \right)^3}{12} + \frac{x_s \times \left( \frac{x_s}{20} \right)^2}{2} + \frac{3x_s}{50} \times \frac{\left( x_s - \frac{x_s}{10} \right)^3}{12} = 0.0261 x_s^4 \text{mm}^4 \]
\[ I_y = \sum \left( \frac{bh^3}{12} \right) = 2 \times \frac{x_s \times \left( \frac{x_s}{20} \right)^3}{12} + \frac{9x_s \times \left( \frac{3x_s}{50} \right)^3}{12} = 8.34 \times 10^{-3} x_s^4 \text{mm}^4 \]

2) Elastic section module of strong axis (Z)
\[ Z = \frac{I}{c} = \frac{0.0261 x_s^4}{\frac{x_s}{2}} = 0.0522 x_s^3 \text{mm}^3 \]
3) Plastic section module of strong axis (S)

\[ S = \sum A d = 2 \times \left[ x_5 \times \frac{x_5}{20} \times \frac{19x_5}{40} + 0.45x_5 \times \frac{3x_5}{50} \times \frac{0.45x_5}{2} \right] = 0.05965x_5^3 \]

4) Section area (A)

\[ A = x_5 \times \frac{x_5}{20} \times 2 + (x_5 - \frac{x_5}{10}) \times \frac{3x_5}{50} = 0.154x_5^2 \text{mm}^2 \]

5) Radium of gyration

\[ r_x = \sqrt{\frac{I}{A}} = \sqrt{\frac{0.0261x_5^4}{0.154x_5^2}} = 0.411x_5 \]

\[ r_y = \sqrt{\frac{I}{A}} = \sqrt{\frac{0.00834x_5^4}{0.154x_5^2}} = 0.232x_5 \]

6) Compressive strength (\( P_c \))

\[ P_c = \frac{P_x P_y}{\phi + (\phi^2 - P_x P_y)^{0.5}} \]

in which

\[ \phi = \frac{P_x + (\eta + 1)P_z}{2} \]

\[ P_z = (\pi^2 E / \lambda^2) \]

6.4.1.3. Constraints of Structural Optimisation

1) Allowance for beam bending moment:

\[ M_{\text{max}} = \frac{w d^2}{8} = \frac{[21.323 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-4}] \times 9000^2}{8} \]

\[ = 0.21 \times 10^9 + 141.75 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \]

\[ < \frac{P_x Z_c}{1} = 0.00902x_4 P_b \]
2) Allowance for beam shear force:

\[ R_{\text{max}} = \frac{wl}{2} = \frac{[21.323 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-5}] \times 9000}{2} \]

\[ = 95953 + 0.063 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \]

\[ < 0.6P_yA_{\text{eff}} = 0.6 \times 265 \times \frac{29x_4}{1200} \times (x_4 - \frac{x_4}{30}) = 3.71x_4^2 \]

3) Allowance for beam deflection:

\[ \Delta_{\text{max}} = \frac{wl^4}{384EI} = \frac{[21.323 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-5}] \times 9000^4}{384 \times 205000 \times 4.51 \times 10^{-3} x_4^4} \]

\[ = 1.8 \times 10^{10} \times \frac{[21.323 + 1.4 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) \times 10^{-5}]}{x_4^4} \]

\[ < \frac{\text{span}}{360} = \frac{9000}{360} = 25 \text{mm} \]

4) Allowance for column slenderness ratio:

\[ \lambda = \frac{KL}{r_x} = \frac{(3000 + x_3 + x_4 + 100) \times 1.5}{0.411x_5} \leq 180 \]

\[ \lambda = \frac{KL}{r_y} = \frac{(3000 + x_3 + x_4 + 100) \times 1.5}{0.232x_5} \leq 180 \]

5) Allowance for cross-section capacity of column with moments in plan buckling:

\[ \frac{F_c}{P_{cr}} + \frac{m_xM_x}{M_{cx}} \frac{(1 + 0.5 \frac{F_c}{P_{cr}})}{1} \leq 1 \]

6) Allowance for cross-section capacity of column with moments out-of-plan buckling:

\[ \frac{F_c}{P_{cy}} + \frac{m_zM_y}{M_{cy}} \leq 1 \]

7) Allowance for column axial force:

\[ 2.87 \times 10^5 + 0.126 \times (3000 - x_1 + x_3 + x_4 + 100) \times (1.7x_2 + 186.3) + \]

\[ 2.33x_2 \times (3000 + x_3 + x_4 + 100) \times 10^{-5} < P_cA = P_c0.154x_3^2 \]

8) Allowance for the depth of external wall:

\[ 105 + x_2 + 13 \leq \frac{x_4}{3} \]
6.4.2 HVAC Analysis Formulation

The office module is to have a single-duct air-conditioning system, there is to be a roof-mounted air-handling and refrigeration plant. It will use a psychometric chart and data to find the peak summer and winter design load and air conditions.

Table 6.8 Detailed Data in Design Scenario

<table>
<thead>
<tr>
<th>Items</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme winter outdoor temperature</td>
<td>-4 °C</td>
</tr>
<tr>
<td>Peaking cooling load through the south-facing glazing (at noon in July)</td>
<td>238 W/m²</td>
</tr>
<tr>
<td>Solar-gain correction factor for slow-response building with shading</td>
<td>0.62</td>
</tr>
<tr>
<td>Comfort indoor temperature in winter</td>
<td>21 °C</td>
</tr>
<tr>
<td>Comfort indoor temperature in summer</td>
<td>22 °C</td>
</tr>
</tbody>
</table>

Table 6.9 Surface Areas, U-value and Y-value in the Design Scenario

<table>
<thead>
<tr>
<th>Surface</th>
<th>Area (m²)</th>
<th>U value (W/m²/K)</th>
<th>(A×U) (W/K)</th>
<th>Y value (W/m²/K)</th>
<th>(A×Y) (W/K)</th>
<th>Decrement Factor</th>
<th>Time lag(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Wall (opaque)</td>
<td>(3−(\frac{x_1}{1000}))×9</td>
<td>1</td>
<td>(\frac{1}{0.376}+\frac{x_2}{620})</td>
<td>16740−5.58(x_1)</td>
<td>3.7</td>
<td>(\frac{3−\frac{x_1}{1000}}{1000})×33.3</td>
<td>0.26</td>
</tr>
<tr>
<td>Internal partitioned wall</td>
<td>54</td>
<td>0.61</td>
<td></td>
<td>2.5</td>
<td>135</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal floor</td>
<td>40.5</td>
<td>0.4</td>
<td></td>
<td>2.9</td>
<td>117.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ceiling</td>
<td>40.5</td>
<td>0.4</td>
<td></td>
<td>1.4</td>
<td>56.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>(\frac{x_1}{1000})×9</td>
<td>3.3</td>
<td>0.0297(x_1)</td>
<td>3.3</td>
<td>0.0297(x_1)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

6.4.2.1 Peak Cooling Load

The objective is to determine the maximum cooling load, which is assumed to occur in extremely high outdoor temperature, namely in July.

Step One: Solar gain through glazing (\(Q_{sg}\))

\[
Q_{sg} = 0.62×\frac{x_1}{1000}×9×238 = 1.328x_1
\]
**Step Two:** Mean fabric gain at air node ($\overline{Q}_{fa}$)

Transfer coefficients

\[
F_{cu} = \frac{3(C_v + 6\sum A)}{\sum A U + 18\sum A} = \frac{3038}{2916 + \frac{16740 - 5.58x_1}{233.12 + x_2} + 0.0297x_1}
\]

\[
F_{cy} = \frac{3(C_v + 6\sum A)}{\sum A U + 18\sum A} = \frac{3038}{2916 + (409.05 - 0.0036x_1)} = \frac{3038}{3325.05 - 0.0036x_1}
\]

The mean solar-air and air temperature are

Opaque wall (south facing, light) $\bar{t}_e = 22.6^\circ C$

Window $\bar{t}_e = 19.6^\circ C$

Mean gains for a mean dry resultant temperature of 22 degrees

\[
\overline{Q}_{fa} = F_{cu} \sum (AU)(\bar{t}_e - \bar{t}_c)
\]

\[
(\overline{Q}_{fa})_{\text{wall}} = \frac{3038}{2916 + \frac{16740 - 5.58x_1}{233.12 + x_2} + 0.0297x_1} \times \frac{16740 - 5.58x_1}{233.12 + x_2} \times (22.6 - 22)
\]

\[
(\overline{Q}_{fa})_{\text{window}} = \frac{3038}{2916 + \frac{16740 - 5.58x_1}{233.12 + x_2} + 0.0297x_1} \times 0.0297x_1 \times (19.6 - 22)
\]

\[
\overline{Q}_{fa} = \frac{0.305 \times 10^8 - 0.61 \times 10^5 x_1 - 216.5x_1 x_2}{(2916 + 0.0297x_1) \times (233.12 + x_2) + 16740 - 5.58x_1}
\]

**Step Three:** Cyclic conduction gains at air node ($\tilde{Q}_{fa}$)

The sol-air and air temperatures are obtained from external design data for the time of day corresponding to 1230h minus the time lag appropriate to the decrement factor for the surface. $\tilde{Q}_{fa} = F_{cy} \sum (AU)f\bar{t}_{eo(\theta - \phi)}$

For wall (hour-ending 0300h) $\bar{t}_{eo} = (10.9 - 22.6) = -11.7^\circ C$

For window (hour-ending 1300h) $\bar{t}_{eo} = (24.2 - 19.6) = 4.6^\circ C$

\[
(\tilde{Q}_{fa})_{\text{wall}} = \frac{3038}{3325.05 - 0.0036x_1} \times \frac{16740 - 5.58x_1}{233.12 + x_2} \times 0.26 \times (-11.7)
\]

\[
(\tilde{Q}_{fa})_{\text{window}} = \frac{3038}{3325.05 - 0.0036x_1} \times 0.0297x_1 \times 1 \times 4.6
\]

\[
\tilde{Q}_{fa} = \frac{-1.55 \times 10^8 + 1.49 \times 10^5 x_1 + 415.5x_1 x_2}{(3325.05 - 0.0036x_1) \times (233.12 + x_2)}
\]
**Step Four:** Internal gains ($Q_{\text{con}}, Q_{\text{rad}}$)

The convective and radiant components of the mean internal gain for an 8-hour occupancy are as follows:

$$ Q_{\text{con}} = (40 \times 6 + 150 \times 4 \times 0.76 + 9 \times 4.5 \times 18.75 \times 0.41) \times (8/24) = 335.7W $$

$$ Q_{\text{rad}} = (40 \times 6 + 150 \times 4 \times 0.24 + 9 \times 4.5 \times 10 \times 0.59) \times (8/24) = 277.34W $$

Hence the convective and radiant components of the swing in internal gain (i.e. value at the calculation hour minus mean value) are as follows:

$$ \tilde{Q}_{\text{con}} = (40 \times 6 + 150 \times 4 \times 0.76 + 9 \times 4.5 \times 18.75 \times 0.41) - 335.7 = 671.4W $$

$$ \tilde{Q}_{\text{rad}} = (40 \times 6 + 150 \times 4 \times 0.24 + 9 \times 4.5 \times 18.75 \times 0.59) - 277.34 = 554.68W $$

**Step Five:** Infiltration gain and mechanical ventilation ($Q_{i}$)

$$ Q_{i} = C_{v}(t_{\text{avg}} - t_{c}) = \frac{9 \times 4.5 \times 3}{3} \times (27 - 22) = 202.5W $$

$$ Q_{i2} = \frac{1.05 \times 10^{-3}}{1} \times \frac{1}{3} \times 2 \times (27 - 22) = 378W $$

$$ Q_{i} = 202.5 + 378 = 583.5W $$

**Step Six:** total sensible cooling load ($Q_{c}$)

$$ Q_{c} = Q_{f} + F_{w} \sum 1.5 \sum \tilde{Q}_{\text{rad}} + \sum \tilde{Q}_{\text{con}} - 0.5 \sum \tilde{Q}_{\text{rad}} $$

$$ = 3.305 \times 10^{4} - 0.61 \times 10^{4} x_{1} - 216.55 x_{3} - 0.13 \times 10^{7} x_{1} - 0.58 x_{1} $$

$$ = 3.305 \times 10^{4} - 0.61 \times 10^{4} x_{1} - 216.55 x_{3} + 197.03 $$

$$ = 3.305 \times 10^{4} - 0.61 \times 10^{4} x_{1} - 216.55 x_{3} + 197.03 $$

$$ Q_{c} = Q_{f} + F_{w} \sum 1.5 \sum \tilde{Q}_{\text{rad}} + \sum \tilde{Q}_{\text{con}} - 0.5 \sum \tilde{Q}_{\text{rad}} $$

$$ = 4.25 \times 10^{4} + 1.49 \times 10^{4} x_{1} + 415.05 x_{3} + 2.5 \times 10^{4} x_{2} + 415.05 x_{3} x_{2} + 394.06 $$

$$ = 4.25 \times 10^{4} + 1.49 \times 10^{4} x_{1} + 415.05 x_{3} + 2.5 \times 10^{4} x_{2} + 415.05 x_{3} x_{2} + 394.06 $$

$$ = 4.25 \times 10^{4} + 1.49 \times 10^{4} x_{1} + 415.05 x_{3} + 2.5 \times 10^{4} x_{2} + 415.05 x_{3} x_{2} + 394.06 $$
\[ Q_k = \bar{Q}_a + \bar{Q}_a + Q_{sg} + Q_v \]
\[ = \frac{3.305 \times 10^8 - 0.61 \times 10^5 x_1 - 216.55 x_1 x_2 + 0.13 \times 10^7 x_2}{(2916 + 0.0297 x_1) \times (233.12 + x_2)} + 16740 - 5.58 x_1 \]
\[ + \frac{4.25 \times 10^8 + 1.49 \times 10^5 x_1 + 2.5 \times 10^6 x_2 + 415.05 x_1 x_2}{(3325.05 - 0.0036 x_1) \times (233.12 + x_2)} + 1.328 x_1 + 1174.59 \]

### 6.4.2.2 Peak Heating Load

A wide variety of heating equipment is available that can heat the occupied space either directly by combustion of a fuel or indirectly by utilizing air, water or steam as a heat transfer fluid. In our case the panel radiator heating system is adopted.

#### Step One: Infiltration and mechanical ventilation conductance \((C_v)\)

For a typical office, the air infiltration rate in winter is equal to 1/h (CIBSE Guide A, 1999)

\[ C_v = \frac{NV}{3} = \frac{9 \times 4.5 \times 3 \times 1}{3} = 40.5 \text{W/K} \]

There are six people; the minimum fresh air per person is 10.5l/s

\[ C_{v2} = \frac{10.5 \times 10^{-3}}{1} \times \frac{1}{3} \times 6 = 75.6 \text{W/K} \]

\[ C_v = C_{v1} + C_{v2} = 40.5 + 75.6 = 116.1 \text{W/K} \]

#### Step Two: Total heat loss \((Q)\)

\[ Q = [F_1 \Sigma (AU) + F_2 C_v](t_c - t_{so}) \]

\[ F_1 = 1 \text{ and } F_2 = 1.10 \text{ for the panel radiator heating system (with an accuracy of 5.0%)} \]

(Chatterton, 1994, p58).

Hence

\[ Q = [1 \times \left( \frac{16740 - 5.58 x_1}{233.12 + x_2} + 0.0297 x_1 \right) + 1.11 \times 116.1] \times (21 + 4) = \frac{4.19 \times 10^7 - 139.5 x_1}{233.12 + x_2} + 0.74 x_1 + 3222 \]

### 6.4.2.3 Constraints of HVAC Optimisation

Supply airflow rate \(Q = \frac{Q_k \times (273 + x_6)}{357 \times (21 - x_6) \times 1000} \text{ m}^3 / \text{s} \)
1) Allowance for air change rate

\[
4 \leq \frac{Q_k \times (273 + x_6)}{357 \times (21 - x_6)} \times 9 \times 4.5 \times 3 = 8.29 \times 10^{-5} \times \frac{Q_k \times (273 + x_6)}{(21 - x_6)} \leq 20
\]

2) Allowance for air duct velocity

\[
2 \text{m/s} \leq \frac{Q_k \times (273 + x_6)}{0.357 \times (21 - x_6) \times x_3} \leq 4.5 \text{m/s}
\]

6.5 Application of the COPGA Framework to the Design Scenario

The COPGA framework includes two cycles, internal and external. The aim of the internal cycle is to obtain the feasible values of interdisciplinary variable (i.e. \(x_1\), \(x_2\) and \(x_3\)) through a two-level formulation, while the external cycle is used to implement the process of the Pareto-based GA algorithm for optimising the objective functions \(F_1\) and \(F_2\). Detailed formulations with regard to the two cycles are presented as follows.

6.5.1 Internal Cycle Formulation

The design scenario posed in Section 6.2 can be decomposed in line with the disciplines (i.e. structural and HVAC disciplines). The design problem is represented at the system level of the internal cycle where the single objective function is a compatible function (i.e. \(d_1^* + d_2^*\)). The problem is decomposed into two sub-problems at the sub-system level. Sub-system 1 represents the structural design and sub-system 2 represents the HVAC design.

The system-level problem determines the value of interdisciplinary variables, namely shared variable \((x_1)\) and coupling variables \((x_2\) and \(x_3)\) for optimising the objective function \(d_1^* + d_2^*\). The sub-system problems are solved to minimise discrepancies between the values of interdisciplinary variable passed down from system-level and the local values of these, while also satisfying discipline-based design constraints. The terminal criterion of the internal cycle is that the change in the value of the system-level objective function in successive iterations is less than a predefined value. In this...
study, this predefined value is $10^{-5}$. The following mathematical formulations are used to describe the internal cycle processes of the COPGA framework for this design scenario. The symbols used throughout this case are given in Table 6.2.

6.5.1.1 System Level Formulation

At the system level of the internal cycle, the optimiser co-ordinates two disciplinary designs by selecting values for all the interdisciplinary variables to minimise $d_1^* + d_2^*$. In each iteration, the system-level optimiser calls two sub-system-level optimisations. Such iteration is stopped when the change in the value of $d_1^* + d_2^*$ is less than $10^{-5}$.

The optimisation model for the system level problem is formally stated as:

**System level model**

Minimise: $d_1^* + d_2^*$

$$d_1^* = (x_1^* - x_1^0)^2 + (x_2^* - x_2^0)^2$$

$$d_2^* = (x_1^* - x_1^0)^2 + (x_2^* - x_2^0)^2$$

With respect to: $x_1^0, x_2^0, x_3^0$

Bounds: $1000mm \leq x_1^0 \leq 3000mm$; $0mm \leq x_2^0 \leq 120mm$, $100mm \leq x_3^0 \leq 700mm$

Both $x_1^*$ and $x_2^*$ are optimal values of the sub-system 1; both $x_1^*$ and $x_2^*$ are optimal values of the sub-system 2. During the system-level optimisation they are fixed, while the interdisciplinary variable (i.e. $x_1^*, x_2^0, x_3^0$) can be adjusted to minimise the function of $d_1^* + d_2^*$.

6.5.1.2 Sub-system Level Formulation

The structural designer determines the dimension of the structural components (e.g. floor, wall) to effectively support all the loads imposed on the building; the building services designer is to provide occupant comfort indoors by way of designing the building configuration (e.g. size of window) or adding mechanical equipment (e.g. heating system and air condition). In this study, the dimensions of beam and column ($x_4, x_5$) and the air supply temperature in the duct in summer ($x_6$) are local variables in the structural and HVAC sub-system respectively. Within the COPGA framework,
the sub-system optimisers also have the freedom to control interdisciplinary variables (i.e. $x_1, x_2, x_3$). Firstly they receive target values of the height of window and depth of wall and duct from the system level, then is as close to these values as possible, by varying the discipline-specific variable (e.g. $x'_i, x'_2, x'_4, x'_5$ in the structural sub-system) while satisfying local constraints.

The mathematical formulation of the structure and HVAC sub-system are stated as follows:

Sub-system 1: Structural sub-system formulation
Minimise: $d_i = (x'_i - \bar{x}_i)^2 + (x'_5 - \bar{x}_5)^2$
With respect to: $x'_1, x'_2, x'_4, x'_5$
Constraints:
Allowance beam bend moment $g_1(x'_1, x'_2, \bar{x}_3, x_4)$;
Allowance beam shear force $g_2(x'_1, x'_2, \bar{x}_3, x_4)$;
Allowance beam deflection $g_3(x'_1, x'_2, \bar{x}_3, x_4)$;
Allowance column slenderness ratio $g_4(\bar{x}_5, x_4, x_5)$;
Allowance cross-section capacity of column with moments in plan buckling $g_5(x'_1, x'_2, \bar{x}_5, x_4, x_5)$;
Allowance cross-section capacity of column with moments out-of-plan buckling $g_6(x'_1, x'_2, \bar{x}_5, x_4, x_5)$;
Allowance column axial force $g_7(x'_1, x'_2, \bar{x}_5, x_4, x_5)$;
Allow the depth of external wall $g_8(x'_2, x_5)$;
Bounds: $1000mm \leq x'_1 \leq 3000mm$; $0mm \leq x'_2 \leq 120mm$; $200mm \leq x_4 \leq 1200mm$;
$100mm \leq x_5 \leq 400mm$

Sub-system 2: HVAC sub-system formulation
Minimise: $d_2 = (x'_2 - \bar{x}_7)^2 + (x'_3 - \bar{x}_7)^2$
With respect to: $x'_3, x'_7, x_6$
Constraints:
Allowance air change rate: $4 \leq g_s(x_1^2, x_2, x_5) \leq 20$;
Allowance air duct velocity $2 \leq g_{sl}(x_1^2, x_2, x_1^3, x_5) \leq 4.5$;
Bounds: $1000 \text{mm} \leq x_1^2 \leq 3000 \text{mm}; 100 \text{mm} \leq x_2 \leq 700 \text{mm}; 15^\circ \text{C} \leq x_5 \leq 20^\circ \text{C}$;

$x_1$, $x_2$, and $x_5$ are target values sent from system level and are fixed at the sub-system level. The detailed formulas of local constraints are presented in Section 6.3.

6.5.2 External Cycle Formulation

The external cycle of the COPGA framework is to optimise the original design objectives. In this study, two design objective functions are minimised, one is total weight of beam and column, and other is sum of peak heating load in winter and cooling load in summer. Some assumptions are taken to formulate two objective functions. With regard to the first objective, dimensions of cross-section in all beams are the same, so are all the columns, calculation of this objective function is based on the weight of one beam and column. The second objective function is calculated on the basis of assumption that occurrence of the hottest day is in July in London.

The mathematical formulation of the external cycle optimisation are stated as follows:

Minimise: $F_1$ and $F_2$

$F_1$ is the total weight of column and beam:

$F_1 = [(18000 + 18x_4) \times 9000 + (3600 + 8x_5) \times (3000 + x_3 + x_4 + 100)] \times 7850 \times 10^{-9}$

$F_2$ is the total peaking cooling load and heating load:

$F_2 = Q_c(x_1, x_2) + Q_r(x_1, x_2)$

With respect to: $x_1, x_2, x_3$

Bound: $1000 \text{mm} \leq x_1 \leq 3000 \text{mm}; 0 \text{mm} \leq x_2 \leq 120 \text{mm}; 100 \text{mm} \leq x_3 \leq 700 \text{mm}$

The optimiser in the external cycle merely manages the interdisciplinary variable (i.e. $x_1, x_2, x_3$), while the local variables (e.g. $x_4, x_5$) are fixed during the course of
optimisation. In addition, there are no constraints in the optimisation of the external cycle. This characteristic is favoured to the use of Pareto-based GA algorithm.

6.5.3 The COPGA Optimisation

Figure 6.8 shows the flow chart for the internal and external cycle, as well as inputs and outputs between the system and sub-system level in the internal cycle.
This flow chart starts from the external cycle, values of interdisciplinary variables (i.e. $x_1, x_2, x_3$) are initialised and sent to two sub-systems in the internal cycle. After completing the optimisation in the sub-system level, optimal values of
interdisciplinary variables are sent to the system level (i.e. $x_1^{0, *}, x_2^{0, *}, x_3^{0, *}$), the system-level optimiser determines new values of interdisciplinary variable (i.e. $x_1^+, x_2^+, x_3^+$) to minimise the objective function $d_1^{+} + d_2^{+}$, and then these new target values (i.e. $x_1^+, x_2^+, x_3^+$) are sent back to the two sub-systems to start the optimisation again. Such iterations among system and sub-system levels are stopped when the change in the objective function of $d_1^{+} + d_2^{+}$ in two successive iterations is less than $10^{-5}$. Finally the value of $d_1^{+} + d_2^{+}$ with the corresponding interdisciplinary variables is sent out of the internal cycle. This means the internal cycle has been completed.

The first step of the external cycle is to evaluate whether the design solutions are compatible through the value of $d_1^{+} + d_2^{+}$. 'Compatibility' here means the same interdisciplinary variable have equal value in the two sub-systems. If $d_1^{+} + d_2^{+} \leq 0.01$, it implies that it is a compatible design, then the external cycle's objective functions are calculated ($F_1$ and $F_2$); otherwise both $F_1$ and $F_2$ are assigned a value of $10^6$. After that, the objectives of $F_1$ and $F_2$ are assessed based on the principle of Pareto rank and crowding distance (detailed explanation in Chapter five). Then the terminal criterion of the external cycle is evaluated, namely the predefined number of iteration in this study. If this criterion is not met, the genetic operations including selection, crossover and mutation are performed. This step is to generate new stochastic values for the interdisciplinary variables to implement the internal cycle again.

6.5.4 Implementation Setup

Sequential Quadratic Programming (SQP) optimisation toolkit in Matlab was used to perform both the subsystem and system level optimisation in the internal cycle because the SQP is fast and efficient for small problems. The Non-dominated sorting GA-II (NSGA-II) algorithm was adopted for the external cycle optimisation. The algorithm was written in Matlab as a separate program. In this program the simulated binary crossover and polynomial mutation was used, crossover probability is below 90%, and mutation probability is below 1/n, where n is the number of decision variables for real-code GAs. The distribution indices for crossover and mutation are
20 and 20 respectively (Deb, et al., 2002). Hence in this study, these parameters includes: number of generation=60; population size=60; crossover probability is below 0.9; mutation probability is below 0.25. Furthermore, all design variables in this study are regards to be continuous.

6.5.5 The COPGA Results

In this section, the results of the design scenario optimised through the COPGA formulation are analysed based on 60 COPGA results and co-ordination in one internal cycle.

6.5.5.1 Analyses of COPGA Results

After completing 60 iterations in the external cycle, the Pareto frontier shown in Figure 6.9 is generated. This curve of trade-off between two objectives (i.e. \( F_1 \) and \( F_2 \)) indicates that a further decrease of \( F_1 \) will be at the expense of increasing the \( F_2 \). Sixty feasible design solutions that are listed in Appendix Four make up this curve. Six representative design solutions described in Table 6.10 are selected from all design solutions based on the maximal or minimal value of objectives or variables.

![Figure 6.9 COPGA Solutions of the Design Scenario](image-url)
The sixty sets of variables including two objectives and three interdisciplinary variables are illustrated in Figure 6.10. As Figure 6.10 shows, the curve of the window’s size ($x_1$) has a similar shape to the curve of the second objective ($F_2$), while it has an inverse shape to the curve of the first objective ($F_1$). This suggests that increasing the value of $x_1$ will increase the value of $F_2$, but decrease the value of $F_1$. In addition, although the depth of wall ($x_2$) is also the function of $F_1$ and $F_2$, Figure 6.10 shows that changes in the function of $F_1$ and $F_2$ do not follow changes in this variable. It implies that the functions of $F_1$ and $F_2$ are more sensitive to $x_1$ than $x_2$. This is why the design solution with minimum $F_1$ and maximum $F_2$ is also where $x_1$ is at the maximum (e.g. No.1 solution in the Table 6.10), while the design solution with minimum $x_2$ may not be one with minimum $F_1$ and maximum $F_2$ (e.g. No.6 solution in the Table 6.10).

<table>
<thead>
<tr>
<th>Number of solution</th>
<th>$F_1$ (kg)</th>
<th>$F_2$ (W)</th>
<th>$x_1$ (mm)</th>
<th>$x_2$ (mm)</th>
<th>$x_3$ (mm)</th>
<th>$x_4$ (mm)</th>
<th>$x_5$ (mm)</th>
<th>$x_6$ ($^\circ$C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.1 (min $F_1$, max $F_2$, max $x_1$, min $x_2$, min $x_4$, min $x_5$)</td>
<td>1529</td>
<td>10251</td>
<td>1812</td>
<td>0</td>
<td>389</td>
<td>731</td>
<td>220</td>
<td>15.2</td>
</tr>
<tr>
<td>No.2 (max $F_1$, min $F_2$, min $x_1$, max $x_2$, max $x_4$, max $x_5$)</td>
<td>1808</td>
<td>8647</td>
<td>1000</td>
<td>120</td>
<td>394</td>
<td>791</td>
<td>242</td>
<td>15.1</td>
</tr>
<tr>
<td>No.3 (min $x_3$)</td>
<td>1556</td>
<td>9466</td>
<td>1311</td>
<td>0.3</td>
<td>354</td>
<td>740</td>
<td>221</td>
<td>15.2</td>
</tr>
<tr>
<td>No.4 (max $x_3$)</td>
<td>1661</td>
<td>8865</td>
<td>1000</td>
<td>31.3</td>
<td>501</td>
<td>759</td>
<td>230</td>
<td>16.6</td>
</tr>
<tr>
<td>No.5 (min $x_6$)</td>
<td>1550</td>
<td>9704</td>
<td>1464</td>
<td>0.8</td>
<td>371</td>
<td>736</td>
<td>221</td>
<td>15.0</td>
</tr>
<tr>
<td>No.6 (max $x_6$)</td>
<td>1596</td>
<td>8981</td>
<td>1000</td>
<td>0</td>
<td>454</td>
<td>744</td>
<td>226</td>
<td>16.7</td>
</tr>
</tbody>
</table>
Figure 6.10 Trends of the Sixty Solutions of Design Scenario

Note: values of variables and objective functions in Figure 6.10 are reduced with certain scales in order that they could be presented as a drawing.

6.5.5.2 Analysis of the Co-ordination in the Internal Cycle

Because not all results from the internal cycle are compatible design solutions, the negotiations between two sub-systems occur in the internal cycle. An example to demonstrate the collaborative convergence process is described here where the starting values of \(x_1\), \(x_2\) and \(x_3\) in system-level variable are 2000mm, 112mm and 600mm. After completing the internal cycle, the values of these variables become 1075mm, 0mm, 562mm where the value of the system-level objective function \((d_1 + d_2)\) is \(1.51 \times 10^{-10}\). Such a solution could be regarded as a compatible design.
Figure 6.11 System-level Co-ordination of Value of Window’s Height

Figure 6.11 shows discipline-specific convergence histories for 3 system-level iterations with respect to the variable of $x_1$. In the first iteration, the system level proposes an $x_1$ value of 3000mm. After meeting all of its discipline-specific constraints, the structural sub-system agrees with the value of 3000mm, while the HVAC sub-system responds with a request to reduce this value to 1955mm. Based on the optimal solution provided by each sub-system, a system-level step is taken which alters the system-level value of $x_1$ to 914mm. In this case, both sub-systems request increasing this value to 1000mm after conducting their own optimisation. In the third iteration, the system-level increases the $x_1$ value to 1075mm. Under this situation, both sub-systems are able to remain disciplinary constraints feasible while finally agreeing on this value of $x_1$. Through such repeated collaborations, the system-level optimiser orchestrates the interdisciplinary compatibility process. On the other hand fixed initial values are assigned to the sub-systems optimisation, for example 2000mm and 1500mm are the initial values of height of window in each iteration of structural and HVAC sub-system optimisation respectively. This is why the values of each sub-system in Figure 6.11 have a big jump between the end of the system-level iteration and beginning of the next one.
Chapter 6

Design Scenario

The number of iterations in the system-level optimisation in this example is three, whilst 9 iterations in Budianto's application (2000) and 89 iterations in Braun's application (1996). The reasons that caused less iterations in this example are:

- Number of system-level variables

In this example, the number of system-level variables is three. There are nine variables in the Budianto's application and 23 variables in the Braun's application. The more variables there are will obviously result in more iteration.

- Changes in the COPGA framework

The system-level optimiser in the COPGA framework merely co-ordinates conflicts among sub-systems to obtain the compatible solution. Besides accomplishing it, the system-level optimiser in Budianto's and Braun's applications also needs to optimise the design objective function (e.g. $F_1$) to an optimal solution. Therefore more iteration must be spent on comparing these compatible solutions to find out the optimal one.

- Initial values of system-level variable

The gradient-based optimisation algorithm adopted in this example is similar to that used in Budianto (2000) and Braun (1996). Such an algorithm is more sensitive to the initial values than other algorithms (e.g. stochastic-based algorithm). In other words, if the initial values are much closer to the optimal one, the gradient-based algorithm will spend less iteration.

- Optimisation algorithm adopted in the system-level

SQP is adopted for the system-level optimisation in this example, while Powell's method is used in the Budianto's application. There would be 161 system-level iterations in this example if using the Powell's method, which is an unconstrained optimisation algorithm. This kind of algorithm must search more space than the constrained optimisation algorithm, such as SQP.
In this example, the three system-level iterations shown in Figure 6.12 require 18 iterations for each sub-system. The values of system-level objectives (i.e. $d_1 + d_2$) in this figure are scaled such that they have the same order of magnitude. This figure also illustrates the trends of wave of $d_1 + d_2$ towards zero in each system-level iteration, although the two sub-system’s values of $x_1$ in Figure 6.9 remain unchangeable after a few sub-systems’ iterations in the first and second system-level iterations. This phenomenon implies that the system-level optimiser co-ordinates two sub-systems through adjusting a set of variables (i.e. $x_1$, $x_2$ and $x_3$), not just a single variable (i.e. $x_1$).

6.6. Summary

This chapter extends the COPGA framework to a realistic multi-objective multidisciplinary design scenario. It also give a step-by-step illustration of the main steps in the process including understanding the design scenario, decomposing this design scenario in line with discipline, formulating mathematical analysis models, setting up optimisation formulation based on the COPGA framework, implementing optimisation. In this scenario, structural and HVAC disciplines were involved to minimise the two overall objective functions, namely total weight of beam and
column, and the sum of peak heating and cooling load. In order to support this design, both the structural and thermal analysis models were developed based on the theory of matrix-displacement method and steady and dynamic state thermal analysis. Furthermore, this chapter presents the process of identifying dependencies among disciplinary design by the use of design structure matrix.
CHAPTER SEVEN: FRAMEWORK EVALUATION

7.1 Introduction

Evaluation is an integral part of the framework development. This chapter describes the framework evaluation process. Firstly the objectives of the evaluation are introduced, and then the methodology adopted is explained, including the selection of evaluation techniques, choosing the evaluators and the design of the evaluation questionnaire. The following section consists of two parts, the first part discusses the approaches to analyse data obtained from semi-structured interviews, which guarantees the rate of feedback; and the lower cost and risk of evaluation practice comparing with case study; the second part presents the evaluation process and analysis of the responses. The analysis covers all the major aspect of the system, including the background of respondents, the main features and mechanisms of the COPGA framework, performance of the COPGA framework, and limitations and suggestions for the industry application of the COPGA framework. Finally the results and methods of the evaluation are discussed.

7.2 Evaluation Aim and Objectives

The aim of the evaluation is to determine the appropriateness and functionality of the COPGA framework in solving multi-objective multidisciplinary building design problems. To achieve this aim, the specific objectives are:

- To verify the process of the COPGA framework ensuring that this framework does not contain any technical errors (Objective One);
- To assess main features and mechanisms of the COPGA framework (Objective Two);
- To assess the performance of the COPGA prototype (Objective Three); and
- To obtain feedback on the limitations and recommendations for improving the COPGA prototype (Objective Four).

It is therefore important to adopt an appropriate evaluation methodology in order to achieve the above specific objectives.
7.3 Evaluation Methodology

According to Gediga et al., (1999) evaluation plays an important part in software development. Without evaluation, mistakes in the development would not be noticed and errors could be repeated in new projects. Evaluation can help by identifying such things as inappropriate knowledge representation approaches and search mechanisms, as well as clarifying the human knowledge within the system (Miles and Moore, 1994). Formal evaluation instigates feedback from the users (Mile and Moore, 1994); involving the end-users in a system's development increases its usefulness and ultimately its future acceptance (Davies et al., 2004).

There are two main categories of evaluation of software systems; formative and summative (Obonyo et al., 2005; Gediga et al., 1999). Figure 7.1 illustrates the proposed general framework for the life cycle of a prototype system (Davies et al., 2004; Smith, 1991).

![Figure 7.1 A General Prototyping Framework and Evaluation Approach (Davies et al., 2004; Smith, 1991)](image)

Figure 7.1 makes a distinction between formative and summative evaluations, primarily in terms of when they take place in a standard linear model of the system.
life cycle. The formative evaluation is undertaken during the development phase in order to improve a system iteratively, until the desired design objectives are reached and weaknesses of the software are eliminated (Anumba and Scott, 2001). There are three key characteristics during the process of this kind of evaluation (Remenyi and Smith, 1999).

- Formative evaluation applied correctly is a frequent, if not quasi-continuous process;
- An evaluator’s perception of what is being evaluated changes and the value put on his/her perceptions changes as he/she learns more about the project; and
- The objective of the system development will evolve during the formative evaluation process.

Summative evaluation is an evaluation of a final design regarding guidelines, standards, or other objectives of the evaluation (Gediga et al., 1999). According to Remenyi and Smith (1999) the purpose of the summative evaluation is to assess and confirm or refute the value of the realised system. This evaluation may be performed before or after system installation. Davies et al., (2004) assert that much of the summative evaluation is managed and performed by those who have designed the system being implemented. The most frequently evaluated criteria seem to be those of information quality (e.g. accuracy, timelines, adequacy and appropriateness) along with facilitating criteria, such as user satisfaction and attitudes.

### 7.3.1 Evaluation Approach

The evaluation process in this research can be divided into two stages. The first stage is a formative evaluation that is conducted during the development of the COPGA framework. The objective of this stage is to test the validity of the overall framework namely Objective One in Section 7.2. Specifically speaking, this stage evaluates whether:

- The optimisation algorithms and the MDO formulation selected are understood and utilised properly, such as NSGA-II algorithm and MOCO formulation; and
• The proposal framework is encapsulated into the computer-based environment in the right way, including elements of system, data transfer among elements etc.

Hence a simple mathematical problem that demonstrates the multi-objective and multidisciplinary characteristics was used to verify the COPGA framework through comparing results between AAO and COPGA formulation. Detailed data analyses of this evaluation are discussed in Section 5.6.3.

The second stage conducts a summative evaluation, in which the COPGA framework is validated ensuring to meet the requirements of building design and solve a design problem by using the proposed approaches. The COPGA prototype is used on a building design problem to assess its results. After the summative evaluation, Objective Two-Four in Section 7.2 can be achieved.

Miles et al., (2000) indicated two popular approaches for summative evaluation. One approach is case studies, in which a trial of the system is provided for industry users to make use of it over a prolonged time period (e.g. a number of weeks); evaluators use a diary to record their practices including information on any difficulties that occurred and any features that they felt lacking (Miles et al., 2000). The advantage of this approach is that the evaluators are given an opportunity and enough time to 'get used to the system's functionality and form an opinion on whether the stated benefits are actually achieved'. The main limitation lies in the difficulties in finding organisation or groups of people prepared to be experimented upon (Ren, 2002).

Another approach is to use a focus group, in which an evaluation workshop is held in a single location with all the evaluators participating simultaneously, and then the evaluators are guided through a usage scenario with the use of appropriate notes. This workshop is followed by the distribution and completion of a questionnaire by each evaluator (Miles et al., 2000). The advantage of this approach is that evaluators can interact with the system designer, it guarantees the rate of feedback; and the cost and risk of evaluation practice is lower than in case studies. However, this approach
requires a relatively large number of evaluators to be available for a short period of time.

Considering that there are a number of algorithms in the COPGA framework, it would take time for the evaluators to understand these and evaluate the system. A semi-structured interview, which combines advantages of the above two methods, is used to evaluate the COPGA framework. This interview begins with a brief explanation of the main algorithms (i.e. GA, Pareto optimality, co-ordination strategy) and their utilisation. This is then followed by a description of a design scenario on which the COPGA prototype is used. Then the interviewees are given a brief note describing the objectives of this evaluation. Lastly they are asked to answer a series of questions and are encouraged to give their suggestions and ideas for improvement.

The advantages of the semi-structured interview as an evaluation method are as follows:

- The face-to-face interview provides two-way communication. As a result the evaluators have an in-depth understanding of the COPGA framework, to ensure reliable results;
- Evaluators do not have to simultaneously attend in the same place, which is more flexible than the focus group method. This can reduce the evaluation cost significantly and can involve more evaluators.

### 7.3.2 Evaluators

Easton (1998) stated that a scientific study should be necessary to obtain numbers for the sample to be representative of all levels and types of users. An evaluation for practical rather than scientific purposes may not need to be rigorous about the number but it should include the full range of users. The governing philosophy behind the COPGA framework is the use of an integrated approach for multi-objective multidisciplinary design, in order that each member of the design team can gain an appreciation of the other members of the design team’s contributions.
Therefore, it is crucial that those involved with its development and evaluation come from various disciplines. In addition, by using more than one evaluator a broader spectrum of opinion can be obtained, which helps to ensure an integrated approach to the design is incorporated into the COPGA framework.

The COPGA framework is evaluated at this stage by:

- Architects
- Structural engineers
- Building services engineers

Apart from the end-user’s viewpoint, the COPGA framework is also evaluated from a developer’s viewpoint, including an academic researcher and a professional software developer. This researcher undertook studies about solving multidisciplinary building design using multi-agent technology, thus having enough knowledge to assess the strategy adopted in the COPGA framework. At the same time, the professional software developer participated in developing the commercial software, which can automatically co-ordinate conflicts made by various designers. Thus his main focus were on technical issues, such as user interface, and the approach of integrating graphic data with design analysis.

**7.3.3 Evaluation Questions Design**

The questions are an important part of the evaluation. They consisted of four major parts that roughly correspond to various aspects of the evaluation objectives. It included questions about:

- The background of the interviewee in terms of the numbers of years of practical experience in industry and the role s/he has held;
- The various features and mechanisms of the COPGA framework;
- The performances/outcomes of the COPGA prototype;
- The comments on barriers in the application in practice and improvements of the COPGA prototype.
The questionnaire is included in Appendix Six. These questions are designed to allow interviewees to express their views in both quantitative and qualitative manners. Quantitative questions consist of a number of statements to which the interviewees can express their level of agreement by circling a number on a five-point scale. The five-point scale is chosen as it is deemed to provide a sufficient range of response without being overly complex (Fellows and Liu, 2003). Interviewees are also allowed to make specific comments via open questions. Thus, the evaluators' responses are not limited by the format of questions proposed by the system developer.

7.4 Evaluation Results

7.4.1 Data Analysis Method

7.4.1.1 Mean Ranking

In this evaluation, there are 8 questions to evaluate main features of the COPGA framework. Every question has a Likert’s scale of five ordinal measures of agreement (from 1 to 5) towards the importance of each feature presented (Figure 7.2). The rating levels are presented to the interviewees as 1 for strong disagreement to 5 for strong agreement.

Ordinal scale of 1 to 5 in ascending order

1 2 3 4 5

Increasing degree of agreement

Figure 7.2 Five Ordinal Measures of Agreement on Likert scale

The main approach used to analyse the data generated from the interviewees is by using the mean ranking technique. The computation of mean ranking is given by the following formulae:

\[
\text{Mean Ranking} = \frac{\sum (1n_1 + 2n_2 + 3n_3 + 4n_4 + 5n_5)}{n_1 + n_2 + n_3 + n_4 + n_5}
\]
Where, 
\(n_1, n_2, n_3, n_4\) and \(n_5\): number of interviewees

### 7.4.1.2 T-Tests

T-test introduced by William Sealy Gossett (Siegel, 1956) is one of the most commonly used techniques for testing a hypothesis on the basis of a difference between two groups of samples (Wikipedia, 2008). Two groups must be (Lowry, 1999):

- Randomly drawn from normally distributed populations; and
- The measures of which two groups are composed are equal-interval.

Compared with other analysis techniques for two groups of samples (e.g. Mann-Whitney U test, Wilcoxon signed-rank test (Lowry, 1999)), the t-test deals with the problem associated with inference based on ‘smaller’ samples, for instance, there must be more than five samples in one group in the Mann-Whitney U test (Hambury, 1987). There are two ways of calculating the t-value in terms of independent and dependent groups. In this research, the t-test is used to evaluate whether different disciplinary designers have the same opinion about the COPGA framework, hence various groups are independent. Step-by-step computation procedures for the t-test used on two independent groups are presented as follows (Lowry, 1999):

1) For the two groups, A and B, of size \(N_a\) and \(N_b\) respectively, calculate the mean of group A and B, \(M_{xa}\) and \(M_{xb}\); the sum of squared deviation of group A and B, \(ss_a\) and \(ss_b\).

2) Estimate the variance of the source population as:
\[
\sigma^2 = \frac{ss_a + ss_b}{(N_a - 1)(N_b - 1)}
\]

3) Estimate the standard deviation of the sampling distribution of sample mean differences as:
\[
\sigma_{M-M} = \sqrt{\frac{S_s^2}{N_a} + \frac{S_s^2}{N_b}}
\]
4) Calculate $t$ as:

$$t = \frac{M_{22} - M_{12}}{\sigma_{M-M}}$$

5) Refer the calculated $t$-value to the critical $t$-value.

As a result the hypothesis can be accepted/refused with a certain confidence. If the confidence is 95%, this means that the likelihood of an experimental result having come through mere random variability, such as mere chance coincidence, sample error, the luck of the scientific draw, is somewhat less than 5%.

In this research, because the number in the sample is not too large, both the mean ranking and $t$ value are calculated manually, not using the statistical package SPSS.

### 7.4.2 Data Analysis Result

Using feedbacks from the semi-structured interviews, an analysis on the various aspects of the COPGA framework and prototype is conducted.

#### 7.4.2.1 Evaluator's Background

The interviewees in this evaluation are the same as in the interviews conducted before development of the COPGA framework. Hence detailed evaluator's background can refer to Section 3.6.1. However interviewees were also asked whether to work as a design coordinator. Results showed that forty percent of evaluators worked in this role.

Based on the information about the interviewees' level of experience, roles played and position held, it is considered that the evaluation group is sufficiently qualified to provide a fair assessment of the system.

#### 7.4.2.2 Responses to Questions

Many open and close questions with the five-point scale are used to collect evaluator's opinions about the concept of the framework and prototype, including main features, performance of the prototype and recommendations for improvements. These opinions are analysed by the use of mean ranking and $t$-test.
This is followed by a selection of the most pertinent responses to the qualitative questions. The achievements against objectives are assessed at each stage.

Section One: Response to main features and mechanisms of the COPGA framework

This section contains an analysis of the questions about the various aspects of the overall COPGA framework and component, such as decomposition schemes, coordination strategy, search tool and functionality.

Question 4

This research used the design structure matrix (DSM) to identify dependent parameters between disciplines (described in Section 6.2.1), hence this question asked respondents to evaluate whether DSM was an effective and useful approach.

Results:

The majority of interviewees gave positive feedback on the use of DSM. They commented that:

- The DSM approach was easy to be learnt and used;
- The graphical representation provided clear views about dependent parameters between disciplines.

Interviewees also expressed some doubt about whether the approach could work well in a large-scale project. An interviewee mentioned there were around 40 parameters in the case of suspended ceiling design; hence users may find it elaborate to list all the parameters sequentially by the use of DSM.

Question 5

This research covered the decomposition of the design in line with disciplines, in comparison with other decomposition schemes based on components or design phases. Hence this question required interviewees to assess whether the decomposition based on disciplines was suitable for building design.
Results:
Most interviewees agreed with the decomposition scheme used in this research because design approaches and computer aided design tools/software were developed well within each disciplinary field. Although the modern method of construction in which design was divided by components was encouraged in recent years, designs in each function room also included different disciplines. In summary, disciplinary decomposition was a fundamental approach and could be applied for various kinds of construction projects.

Question 6
In this question, interviewees assessed eight attributes of the COPGA framework listed in Table 7.1. Their opinions indicated the extent of agreement on each aspect on a five-point scale.

Results:
Two steps were taken to analyse results from the above question. In the first stage, data from all ten interviewees were expressed in terms of mean ranking. In the second stage, and using the t-test technique, results were analysed based on the three groups, namely: architect, structural, and HVAC engineers. There were 2, 4, and 2 interviewees in the respective group.
Table 7.1 Response to Features of the COPGA Framework

<table>
<thead>
<tr>
<th>Statements</th>
<th>Ranking</th>
<th>Mean Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 The COPGA framework ensures that all essential characteristics of building design optimisation are represented.</td>
<td>2 8</td>
<td>4.8</td>
</tr>
<tr>
<td>Do you agree that the COPGA framework adopts a robust decomposition scheme?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2 Implement easily this decomposition</td>
<td>4 6</td>
<td>4.6</td>
</tr>
<tr>
<td>1.3 Reduce the dimension of design problem</td>
<td>2 1 7</td>
<td>3.5</td>
</tr>
<tr>
<td>1.4 Follow current organisational structure for building design</td>
<td>2 5 3</td>
<td>4.1</td>
</tr>
<tr>
<td>1.5 Take full advantage of disciplinary knowledge and analyses</td>
<td>1 1 3 5</td>
<td>4.2</td>
</tr>
<tr>
<td>1.6 The COPGA framework is a good decision support tool of building design</td>
<td>3 7</td>
<td>4.7</td>
</tr>
<tr>
<td>1.7 The COPGA framework improves computational efficiency for building design optimisation.</td>
<td>2 5 3</td>
<td>3.2</td>
</tr>
<tr>
<td>1.8 The COPGA framework is easy to extend for more than two disciplinary designs.</td>
<td>1 4 5</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Table 7.6 presents the results of the first stage; the interviewees gave relatively high ranking to Statement 1.1, 1.2, 1.4, 1.5, 1.6 and 1.8. The results suggested that interviewees in general felt that the principal features of the COPGA framework were effective. In other words, they agreed that:

- The COPGA framework addressed the essential characteristics of building design, namely multi-objective multidisciplinary;
- Decomposition based on disciplines was easy to use in building design, and such a decomposition scheme also followed the current organisational structure. This is consistent with the comments in Question 5;
- The subsystems of the internal cycle in the COPGA framework kept disciplinary autonomy. The COPGA structure allows each designer to complete their own designs by the use of specialised experience and software;
- The COPGA framework was promising in developing a system as an aid to the decision-making tool of building design; and
- Only two disciplinary designs were involved in the demonstrated COPGA framework. Most interviewees agreed that this framework could be extended for the application of design which involved more than two disciplines.

However, the interviewees gave relatively low score to two questions regarding the decrease in the number of variable to be handled (Statement 1.3) and improving computational efficiency (Statement 1.7). The reasons that the interviewees did not agree with these two statements were also investigated.

With regard to Statement 1.3, most interviewees with experience in structural and HVAC designs indicated that the number of variables in their discipline design was not be reduced in the COPGA framework; in contrast architects gave positive feedbacks on this statement because the COPGA framework enables them to consider only interdisciplinary variables while leave discipline-based variable to structural and building services engineers to analyse.

A more important discussion was about Statement 1.7. Although the coordination of conflicts between disciplines is performed automatically in the COPGA framework, it could reduce time in obtaining a compatible solution. However the genetic operations of crossover and mutation in the GA algorithm might generate some unacceptable design schemes, which could be a time-wasting strategy. Therefore the evaluators thought that the COPGA framework may not improve the overall computational efficiency for building design.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Mean Rank</th>
<th>T value</th>
<th>Decision Critical t=2.13 (at (\alpha=0.05)) one tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>All features from 1.1 to 1.8</td>
<td>4.63</td>
<td>3.99</td>
<td>Accept (H_1)</td>
</tr>
</tbody>
</table>

The null hypothesis (\(H_0\))- There is no statistically significant difference in mean level of satisfaction between architect and structural group with regard to features of the COPGA framework.
The research hypothesis \((H_1)\)- The mean level of satisfaction in the architect group is significantly greater than in the structural group with regard to features of the COPGA framework.

Table 7.2 shows the mean rank of architect and structural groups respectively and a t value at 95 percent confidence level. The t value obtained is larger than the critical value. Hence, the research hypothesis \((H_1)\) can be accepted concluding that the architect gave more positive views on the capability of the COPGA framework than the structural engineers. Since the principle underlying the COPGA framework is a systematic design approach, which is implicit in the way architects work, the framework can be more effective in solving architectural problems than other disciplines.

### Table 7.3 T Value of Satisfaction with the COPGA Framework between Structural and HVAC Engineers

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Mean Rank</th>
<th>t value</th>
<th>Decision Critical t (=2.78) (at (\rho=0.05)) two tail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Structural engineer</td>
<td>HVAC engineer</td>
<td></td>
</tr>
<tr>
<td>All features from 1.1 to 1.8</td>
<td>3.99</td>
<td>3.995</td>
<td>0.0227</td>
</tr>
</tbody>
</table>

The null hypothesis \((H_0)\)- There is no statistically significant difference in mean level of satisfaction between the structural and HVAC groups with regard to features of the COPGA framework.

The research hypothesis \((H_1)\)- There is statistically significant difference in mean level of satisfaction between the structural and HVAC groups with regard to features of the COPGA framework.

Table 7.3 shows the mean rank of the structural and HVAC groups respectively and t value at 95 percent confidence level. The t value obtained is smaller than the critical value. Hence, the research hypothesis \((H_0)\) can be accepted by concluding
that both structural and HVAC engineer have the same attitudes towards the capability of the COPGA framework.

Question 7
This open question asked interviewees to assess the co-ordination strategy adopted in the COPGA framework.

Results:
Most interviewees considered this co-ordination strategy to be an effective approach to solving conflicts between disciplines. The reasons were:

- The co-ordination strategy used in the COPGA framework was similar to the interaction process of multidisciplinary design in practice. For example structural and HVAC engineers were required to analyse a design solution proposed by an architect. If this solution was not feasible, the architect would make some changes, and then asked the engineers to undertake the analysis again. Such iteration was simulated in the COPGA framework through the introduction of the target value of the interdisciplinary variables.

- Interviewees commonly agreed that the classification of variables best addressed problems in multidisciplinary design. In this co-ordination strategy, variables were grouped into local, shared and coupling variables. Through such a classification, relationships between disciplines were clearly demonstrated. Furthermore, this co-ordination strategy proposed different approaches to handling interdisciplinary variables (i.e. shared and coupling variables) and local variables respectively.

Question 8
This open-ended question asked interviewees to assess the use of the genetic algorithm in the COPGA framework.

Results:
Most interviewees showed a positive view on the usage of GA. They found:
The idea of using a GA modelled on the Darwinian theory of evolution was very interesting, the most useful aspect of GA was the ability to create design options, and it also provided the user with a selection of possible solutions.

The involvement of an optimisation search strategy facilitated the coordination of conflicts, because the adoption of an optimisation algorithm could find more disciplinary design solutions, which increases the possibility of achieving agreements between disciplines.

Section Two: Response to performances of COPGA prototype

In this section, a demonstration of the COPGA prototype was presented to interviewees, this demonstration illustrated a typical office room design in which the structural and HVAC design were considered in the form of two objective functions (i.e. weight of the structural components and sum of peak heating and cooling loads) needed to be minimised simultaneously. The interviewees were required to evaluate the performances of the COPGA prototype through Question 9 and Question 10.

Question 9
After the demonstration of the COPGA prototype, the interviewees were asked to give feedback on some aspects of the application.

Results:
Answers obtained from interviewees covered functionalities and outcomes of the COPGA prototype. Four major points are summarised:

- An effective design support tool. Most interviewees felt the COPGA prototype could be an aid in decision-making to the client/developer as well as the design team. They envisaged the client consulting the COPGA prototype with another member of the design team to assess the impact changes on the overall design as the COPGA prototype could re-evaluate the changed design and check conflicting design objectives simultaneously;
• A systematic design. Construction projects could be conceived as a system design that required contributions from various specialists. Most interviewees supported the idea that an integrated approach to system development in the COPGA prototype would reap some real benefits because interactions between disciplines were also taken into account, not merely one component of system;

• Good mechanism for a multi-objective problem. In practice, designers set weighted factors on various design objectives. In some cases these factors were hard to be decided in advance, whilst the Pareto optimality adopted in the COPGA prototype could provide solutions with different sets of weighted factors in one calculation;

• Increased confidence in the design solution. Most interviewees expressed that the COPGA could help increase confidence in decisions made because it enabled the user to rapidly consider many design options, and the design solutions were also checked using accurate analysis software.

Question 10
The main purpose of this question was to check respondents’ concerns about the application of the COPGA prototype.

Result:
With regard to this question, interviewees indicated:

• The design scenario used in the COPGA prototype was sufficient and good enough to demonstrate the proof of concept. However they were worried about capability in applying the COPGA prototype in a complex construction project, because it was difficult to predict all conflict situations between disciplines before commencing design.

• There was a need for some post-processing. On one hand, the large amount of output information was generated by the COPGA prototype; textual information did not facilitate easy comparisons. It is more effective to use a graphical representation for such a purpose, mainly in the form of graphs. On the other hand most industry designers did not understand the physical
meaning of the Pareto frontier, and therefore further interpretations should be provided.

• The COPGA prototype handled all variables as continuous. As such, it was possible that this could not meet requirements in the practical design. For example, the variable of the standard steel cross-section was often discrete.

Section Three: Response to industry applications of the COPGA prototype

Through this section, information about limitations and suggestions for improvement of the COPGA prototype were obtained from interviewees. Two open-ended questions were used for this.

Question 11

This question asked the interviewees on aspects that inhibited users from utilising the COPGA prototype.

Results:

The comments made by the interviewees included:

• Difficulties in modelling the multidisciplinary design process. Construction designs were thought too complex and important, so it was hard to be handled by a computer. As a result, users doubted computer system capabilities to mimic human issues during the process of design, such as design culture, client changes, non-quantitative design aspects.

• Large efforts on editing input files. The fundamental feature of the COPGA prototype was considering every disciplinary design of a project at the same time. In other words, users must spend a lot of time and energy on deciding the value of all disciplinary inputs to run this prototype.

• Identification of all conflict between disciplines. The premise of using the COPGA prototype was identification of all interdisciplinary variables, which might be beyond designer capabilities in the new and complex project design.

• Time constraints. Time is a major constraint when creating alternative designs. The COPGA prototype could spend more time than other decision
support systems to obtain results owing to the use of GA, Pareto optimality and co-ordination strategy for multidisciplinary design.

Question 12

In this question interviewees were given the chance to express their suggestions for improving the COPGA prototype in the future.

Results:

The majority of interviewees made at least one comment regarding this question. The main suggestions are summarised as follow:

- Adding interaction between system and human. The COPGA prototype should allow the users to make changes in design to suit their preference during the whole process, thus enabling them to see how their changes affect the overall design.

- Interoperability with other input file formats. The comment on Question 11 demonstrated that a large number of inputs could be a barrier to using the COPGA prototype, so the COPGA prototype should be compatible with existing disciplinary design tools to avoid repetitive work. For example architects are accustomed to using a graphical package and quantity surveyors established large databases that enable them to quantify and estimate future cost by the use of Microsoft Excel.

- Knowledge sharing. The COPGA prototype should take full advantage of valuable experiences and lessons, which can help in obtaining optimal solutions quickly. For example a respondent suggested parameters, like overall building height, uniformity, net/gross ratio, wall/floor ratio, were good indicators of cost-efficient building design as opposed to calculating capital costs.

- Improvement of computational speed. As the comments on Question 11 pointed out, time was always the major constraints on the application of the COPGA prototype. A respondent with good IT knowledge suggested the deployment of the COPGA prototype in the high-performance computing environment should be a promising solution. Other interviewees expressed
some rules of thumb, such as span/depth ratios to estimate the structure depth of member, could be used, which would significantly reduce the burden of computation.

7.5 Discussion

The following sections distil the results of the evaluation and further discuss the effectiveness of the evaluation methods.

7.5.1 Results

The responses received from the evaluators were very positive and the system itself generated a lot of enthusiasm. This was evident from the number of suggestions about ways in which the COPGA prototype could be more useful and provide even more design solutions. From these demonstrations, it can be concluded that the COPGA prototype used as a decision support system is acceptable, as it allows the users to stay in control of the design process and has the potential to be a very useful tool to designers. However the interviewees put forward some barriers for the adoption of the COPGA prototype in practice, expectations on improvement of this prototype are also expressed. All comments are presented in brief as follows:
Table 7.4 Summaries of Comments from Interviewees Regarding the COPGA Prototype

<table>
<thead>
<tr>
<th>Benefits of the prototype system</th>
<th>Barriers of the prototype system</th>
<th>Suggestions for prototype improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Address the major characteristics and problems of building design optimisation;</td>
<td>• Difficulties in modelling multidisciplinary design process;</td>
<td>• Involvement of human interaction;</td>
</tr>
<tr>
<td>• Facilitates decision making for disciplinary designers and clients;</td>
<td>• Large effort on establishing building model;</td>
<td>• Compatible with existing input software, such as AutoCAD, Excel;</td>
</tr>
<tr>
<td>• A systematic approach of multidisciplinary design;</td>
<td>• Identification of all conflicts between disciplines;</td>
<td>• Output interpretation;</td>
</tr>
<tr>
<td>• Improving capability of searching design space;</td>
<td>• Time constraints;</td>
<td>• Increasing scope of using GA, such as solving discrete variable in disciplinary design;</td>
</tr>
<tr>
<td>• A reasonable approach of solving a multi-objective problem;</td>
<td>• Interoperability with other input file formats</td>
<td>• Knowledge sharing;</td>
</tr>
<tr>
<td>• Appropriate and suitable well decomposition scheme;</td>
<td></td>
<td>• Deployment in distributed computing environment</td>
</tr>
<tr>
<td>• Increase of confidence in design solutions;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Integrates specialised analysis software</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7.5.2 Appropriateness of the Evaluation Approach

The evaluation undertaken achieved its set objectives. Appropriateness in this context means that the objectives are clearly defined, measurable and quantifiable. The evaluation approach conducted demonstrates its effectiveness due to several points, including:

- Approach of collecting data. The reasons behind the selection of the semi-structured interviews evaluation method have been discussed in Section 7.3.1. In addition the quantitative and qualitative data are collected through these open and closed questions in the interviews.
• Approach of analysing data. Both mean ranking and t-test technique are used to analyse the quantitative data. These quantitative results are in general positive and give a good indication that the COPGA framework is considered to be appropriate for multidisciplinary multi-objective building design optimisation. The qualitative results are obtained from the interviewees to highlight areas that could be improved and to suggest what those improvements might be.

• Evaluators: All the interviewees have considerable experiences in building design. The range of interviewees covers the fields of architecture, structures, and building services. Furthermore the interviewees are encouraged to raise their queries about their understanding of the COPGA framework and state whether this framework addresses the need of building design.

7.6 Summary

This chapter has described the process of evaluation of the COPGA framework, including formative and summative evaluation. The verification of this framework in the formative evaluation is discussed in Chapter 5, while this chapter focuses on discussing approaches adopted in the summative evaluation. A semi-structured interview was carried out to assess the main features and mechanisms of the COPGA framework and performance of application of the COPGA prototype in an office room design. The results obtained through the quantitative and qualitative analyses are highly encouraging and indicate that the objectives of the COPGA framework have been achieved. The majority of interviewees think that the COPGA prototype is a powerful decision support system for building design. The benefits and limitations for the adoption of the system are also pointed out. These points will contribute to the recommendations for further development, which are addressed in the next chapter.
8.1 Introduction

This chapter concludes this research project, which explored the application of MDO to building design optimisation. It first summarises the work that was carried out to achieve each of the research objectives. Conclusions are then drawn the COPGA framework could be an effective and powerful tool in supporting building design optimisation. The last section of the chapter presents the limitations of the research and makes recommendations for further work.

8.2 Summary

The process of building design is complex, requiring skills from several disciplines, including architecture, structural engineering, building services, etc. Well-informed interdisciplinary decisions ensure compatibility and promote design solutions that effectively satisfy cost, and functionality and performance objectives. The rationale for undertaking this research was the need for systematic methods for multi-objective multidisciplinary building design optimisation. To fulfil this need, the research developed the COPGA framework based on MDO. The aim was achieved through several objectives of the research including:

- To investigate the existing applications of optimisation algorithms in building design and define the characteristics of building design optimisation and challenges;
- To review state-of-the-art MDO applications in other industry sectors (e.g. aircraft design, automotive design, etc.) and to identify the major issues that inhibit using MDO in engineering design;
- To develop and test a Pareto Genetic Algorithm based on Collaboration Optimisation (COPGA) framework for building design;
- To implement the COPGA framework within the context of building design; and
- To evaluate the COPGA framework by the way of expert assessment.
The specific tasks undertaken in this research, with respect to research objectives are summarised below:

**Objective 1: To investigate the existing applications of optimisation algorithms in building design**

Initially comprehensive reviews provided general background on building design optimisation. It showed that, although many efforts have been made to solve each disciplinary design optimisation, the systematic multidisciplinary optimisation is still in the early stages in building design. The main reason is the complex process of coordination of conflicts between disciplines.

The characteristics of building design optimisation applications were found to be:

- The large number of variables: Some researchers adopted algorithms like a lattice method and ANN, or utilised designers' experience to reduce computational requirements which result from a large number of variables.
- Multi-type variable: With regard to continuous variables, gradient-based optimisation algorithms (i.e. Newton, SQP) are often used. While the derivative-free algorithms (i.e. genetic algorithm and simulated annealing algorithm) are good for handling discrete variables.
- Multi-objective design: The applications of a weighting factor method and Pareto optimality are widely used to solve conflict between economic and design performance requirements.
- Multi-disciplinary design. Variables and objective functions were identified for architectural, structural, and building services design respectively.

Although various techniques have been adopted to improve the efficiency and effectiveness of building design optimisation, the fields of development of building optimisation models, building simulation integration and system design are still be investigated (as presented in Section 3.5). A MDO framework has good potential to satisfy the above the requirements.

204
Chapter 8

Conclusions and Recommendations

Objective 2: To review state-of-the-art MDO applications in other industry sectors (e.g. aircraft design, automotive design, etc.) and to identify the major issues that inhibit using MDO in engineering design;

In order to identify the key features, enablers, barriers and potential benefits of applying the MDO to building design optimisation, relevant implementations in other industries (i.e. aircraft and automotive) were reviewed as presented in Chapter Four. The literature review revealed there were some challenges to the application of MDO in engineering designs, including appropriate MDO formulation, high-performance computing requirement, data exchanges between different analysis software etc. After considerations of these challenges, this research focused on the development of an appropriate MDO formulation. Therefore studies on the main elements of MDO problems and fundamental MDO formulation were undertaken. The findings from these studies demonstrated that the single-level MDO formulation was not suitable for large-scale building design. Thus this research concentrated on the two multi-level formulations: collaborative optimisation (CO), and analytic target cascading optimisation (ATC). Based on the main features of the two formulations and characteristics of building design practices, it was established that the CO formulation has advantages over the ATC formulation for the following reasons:

- characteristics of building design teams, which are defined by disciplines;
- maturation of discipline-based analysis tools; and
- different design methods used for each discipline.

The researcher used these findings from the literature as a guide to develop the proposed framework which is based on the CO formulation.

Objective 3: To develop and test a Pareto Genetic Algorithm based on Collaboration Optimisation (CO-PGA) framework for building design optimisation.

Based on the framework proposed for improving multidisciplinary building design optimisation through the application of MDO, the problem formulation and
optimisation algorithms involved in the framework were established, as presented in Chapter Five. The process of developing this framework included four steps:

1) Investigation and analysis of the implementation of exiting MOCO framework shows some limitations, these include the determination of step size at the system level, local optimum, the approach used for multi-objective problems, and the delinquent nature owing to the MOCO formulation. These limitations encouraged the adoption of the Pareto-based GA algorithm for the proposed framework.

2) A Two-cycle framework called COPGA was developed. The main purpose of the internal cycle is the coordination of conflicts between disciplines; while the Pareto-based GA algorithm was used to achieve the building design objectives in the external cycle.

3) A simple mathematical problem was used to test the COPGA framework. Variables in this mathematical problem were divided into local, shared and coupling groups, and then used in the COPGA framework.

4) Results obtained from the COPGA and a single-level AAO formulation were compared, to verify the COPGA framework. Furthermore findings from these results demonstrated that the COPGA framework was preferred for scalability and organisational reasons.

During the process of developing the framework, a series of papers and reports were worked out, and discussions with experts at conferences and workshops were conducted. These activities were helpful in refining this framework.

**Objective 4: To implement the COPGA framework within the context of building design**

This objective considered the implementation of the COPGA framework in the context of building design. This included four steps as follow:
Propose an implementation scenario: Based on literature reviews and the inspection of design documents, a typical office in a three-storey building case study was defined in Section 6.2;

Develop analysis program: In order to optimise the design scenario, structural and thermal analysis programs were developed based on the matrix-displacement method and the steady-state and dynamic method guided by CIBSE Guide respectively, and verified as presented in Section 6.3;

Identify shared and coupling variables. The parameter-based DSM methods were used to graphically present interdisciplinary variables between structural and HVAC design in the scenario;

Use the design case study in the COPGA framework (as presented in Section 6.5.1 and 6.5.2); and

Implement the design case study in the computing environment and analyse results.

Objective 5: To evaluate the COPGA framework by the way of expert assessment

The COPGA framework was evaluated by an academic researcher, software developer, and industry participants including architects, structural and HVAC engineers. The evaluation approach adopted was semi-structured interviews. The evaluation questions were designed to achieve the following objectives:

To assess the main features of the COPGA framework;

To assess the performance of the COPGA framework;

To identify the limitations and improvements for the COPGA framework.

The analysis of the feedback results were described in Chapter Seven. These results of the evaluation showed that:

The coordination strategy, the GA search mechanism, and the approach of Pareto optimality adopted in the COPGA framework satisfied all major requirements of building design, such as multidisciplinarity, multi-objective problem.
• The interviewees gave a useful feedback on the framework, its further developments, and most importantly, on further applications of COPGA in the industry.

The evaluation confirmed that the COPGA framework could be an useful decision support tool for building design.

8.3 Conclusions

This thesis has demonstrated the applicability of MDO in building design optimisation, and developed a robust COPGA framework. A number of main conclusions can be drawn from this research:

• The contemporary building design is more complex due to the large number of design variables, the conflict design objectives including economic and performance aspects, and the close interaction between disciplines.

• Most existing applications of optimisation in building design focused on single disciplinary design, with little available for multidisciplinary design optimisation framework.

• Although the MDO methodology has been popular in the aerospace and automotive industry for a few years, its applications in building design is still at the early stages, which provides a few potential research needs, such as appropriate MDO formulation, problem execution framework etc.

• The COPGA framework is an effective decision support tool for building design owing to the following features:
  o Systematic design. A formal model provides explicit means of achieving consistent and concurrent design solutions in scenarios that require coordination between multiple disciplinary performance specifications. Such a systematic process can go beyond where intuition often fails.
o A robust search tool. This framework used a genetic algorithm as a search tool, investigating the design space quickly and reliably, and presenting the users with a selection of optimal designs.

o Approach for a multi-objective problem: The Pareto optimality adopted by this framework allows the users to consider more feasible design solutions.

- Breaking down a large-scale design in line with disciplines can be a fairly straightforward step in the context of building design. As discussed in Section 4.5.3.2, such a decomposition scheme could fully exploit specialised design methods and analysis software, and also align with the current organisational structures.

- The COPGA framework offers a new approach to integrating multiple analyses in the decision-making process. This approach allows the use of existing analysis software with minimal modification, and these analyses software can be also distributed geographically.

- The COPGA framework demonstrate an optimisation algorithm as a rational decision-making tool for building design, whilst this multi-level design optimisation methodology can be extended for different kinds of construction projects, such as dam, highway, healthcare projects etc.

- The COPGA framework can provide designers with a better design methodology that integrates different disciplines in the design process. However such design tools cannot replace the role of humans in making decisions, therefore the best feature of any design support systems is to provide more design alternatives to the decision maker.

8.4 Limitations of the research

All research studies have their limitations, and this study is no exception. The main limitations were as follow:
Conclusions and Recommendations

- The COPGA framework was merely applied for an office in a three-storey building. Some problems may arise when it is applied to a large-scale project. The main reason is that those interdisciplinary variables are hard to identify before optimisation.

- The COPGA framework is a mathematical model. This requires all design variables and objective functions to be formulated in the numerical format. This may cause difficulties for some design areas, such as functional requirement and aesthetic.

- The research is constrained by the evolving nature of the MDO methodology. Some technical issues, such as theoretical convergence of MDO formulation (Burgee et al., 1996), are still being questioned. Therefore, this research cannot demonstrate all the advantages that building design can obtain from the MDO.

8.5 Recommendations for further work

This research explored the applicability of MDO in building design to obtain a systematic design solution. However the COPGA framework is still at a proof-of-concept stage, therefore some technical and operational aspects need to be further developed.

8.5.1 Further developments of the COPGA framework

While the value of the COPGA framework was demonstrated in this research, there is scope for further developments along the following lines:

- Utilise designer’s experience

The typical engineering optimisation problem is non-linear and non-convex, effective designers’ instruction (e.g. switch optimisation methods) can reduce time in obtaining optimal design. Furthermore the COPGA framework should support users’ interaction through adjusting some parameters set during the design cycle, which can avoid deadlock in the optimisation iteration process.
• Incorporate other legacy system
In practice, designers have been used for some applications to undertake their own design, for example, architects use AutoCAD while quantity surveyors use Excel. These software establish initial designs, which are inputs of the COPGA framework. If the PCACO framework can’t communicate with these design tools, designer will spend a lot time and energy on completing input files, which inhibit users from using the COPGA framework.

• Apply the GA in disciplinary design
GA in this research was merely used for the external cycle optimisation, which limited the full advantages of GA. If the use of GA can extend for the disciplinary design, more feasible design solutions will be explored. Furthermore the obvious feature of GA can deal with discrete variables very well, while most variables in the disciplinary design are not continuous.

• Interpret output files
From the evaluation of COPGA, it became apparent that there is a need for some post-processing because of the large amount of output information generated. It would be more effective to use graphical representations to facilitate easy comparison, e.g. steel weight versus overall height of building.

• Undertake industry field work for further evaluation
In this research the COPGA prototype application on the design scenario was enough to demonstrate the functionality of the COPGA framework, however further study should apply the system to real projects for capturing design knowledge and interdisciplinary design variables, which is a good method to evaluate and improve the performance of the prototype system.

8.5.2 Recommendations to the MDO community
The MDO formulation is the first element of the application of MDO; there are other relevant fields which is under active development, as outline below:
• Development of a computational framework to execute the MDO application

This framework creates the sophisticated computational procedure to support the implementation of the MDO application; consequently the designers would be able to concentrate more on the application and less on the programming details. The framework should support the integration of various processes of the MDO application. For example maintaining data used by multiple disciplines, visualise intermediate and final optimisation results, giving notification about any changes from other disciplines, automatically processing and moving data, some mechanism for fault tolerance etc (Salas and Townsend, 1998).

• Application of grid computing in the MDO application

The MDO application could be very time-consuming and computationally intensive due to involvement of accurate analysis and coordination among sub-problems. Generating a better solution within a restricted time scale always inhibits the application of MDO in engineering design. Furthermore the multiple users in the MDO environment can be distributed when carrying the optimisation in the heterogeneous environment. Based on the above two reasons, Grid computing is a promising solution. The Grid allows the sharing and combining of computing power, data resources and software applications over the Internet (Foster et al., 2001). By providing scalable, secure, high-performance mechanisms for discovering, accessing and unitising remote research in the heterogeneous environment, Grid technology should be successfully utilised to reduce the computational time requirement of the MDO application.

8.5.3 Integrating MDO method with Agent Technology

The MDO methods improve the quality of large-scale multidisciplinary design through an appropriate decomposition scheme, classification of design variables, and effective management of interactions amongst different disciplines. However some limitations still exist as discussed in Chapter Seven.
Recent developments of intelligent agent technologies make it an ideal platform for the implementation of the MDO models. Distributed agents have the potential to strengthen the MDO method in the following aspects:

- **Incorporation of judgement and experience**: A limitation of the MDO method is the difficulties in formulating ill-structured problems, which does not have an explicit, clearly defined algorithmic solution. No specific rules can be followed when integrating disciplines. Experienced designers deal with system integration using judgement and experience. The agent-based approaches offer an effective method to tackling the ill-structured design problems by taking advantage of the knowledge-based agent systems.

- **Intelligence**: A key part of the MDO process is the use of appropriate formulation. Various MDO formulation (e.g. CO, ATC, BLISS and CSSO) may be introduced in terms of problem decomposition, different dependencies and coordination strategy. The capability of intelligence in the agent-technology can be employed for task dispatching and planning, namely selection of MDO formulation.

- **Distributed architecture**: The MDO problem is often high-computational and time-consuming. Therefore it is natural to adopt a distributed architecture consisting of many intelligent agents, which implicates parallel computing richly utilises computing resource on network, and reduces computing time greatly.

- **Learning ability**: The coordination strategy adopted in the MDO methodology is relatively rigid. Many recorded trends of resolutions between disciplines are generated by accumulating knowledge from a large number of design projects. A set of design methodologies are constructed by using learning techniques in agent-based systems to generalise these trends. These methodologies can be then used to guide design teams through future design projects.

- **Adaptability**: Agent-based systems allow the addition or deletion of agents. Thus, new knowledge can be added, and old knowledge removed rapidly. This provides a way to systematically incorporate new design knowledge into the problem solving process. A system that discovers design methodologies can
constantly be fed with the new design knowledge, hence producing design methodologies that are based on the latest technologies, such IT technology, construction materials etc.

8.6 Closing Remarks

The successful adoption of MDO should considerably improve quality and reduce the cycle time of building design. The research in this thesis demonstrated how the COPGA framework improves building design by the use of the MDO. Such capabilities of this framework such as automatically coordinating conflicts between disciplines, satisfying multiple design objectives simultaneously, and powerful design space search tool, allow it to obtain a systematic design solution that is difficult to resolve through other approaches. Moreover, the obvious feature of the COPGA framework is to decompose a large building design problem into a related grouping of smaller, more tractable, coupled discipline-based sub-systems. Such a decomposition scheme is suitable in terms of design process, and analysis software available.
References


215
References (continued)


References (continued)


References (continued)


218
References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


References (continued)


Appendix One:

List of Publications Arising from the Research
Conference Paper:
Appendix Two:

Template for Industry Semi-Structured Interviews
Information on Respondents

1) The respondent's experience has involved working as a:
   Architect Designer (...), Structural Engineer (...), HVAC Engineer (...),
   Client ( ) Other, please specify _______

2) Your experience in building design ________ years.

Needs for building design

3) In your opinion, what are the characteristics of contemporary building design? (tick all that apply)
   □ 1 A large amount of parameters to be considered in design
   □ 2 Several objectives to be achieved such as functional requirements, economic requirements, etc.
   □ 3 A few design disciplines involved such as consultant, structural, electrical, mechanical design and other experts.
   □ 4 Others, please specify ________________________________

   If you choose more than one option, please select the most obvious characteristic ________

4) Some conventional building design practices are listed below, to what extent do you agree with them based on your experience? Scale these from 1 to 5 (where 1= Strongly disagree  2= Disagree  3= Neutral  4= Agree  5= Strongly agree)

   a Sequential independent disciplinary design plus system level review
   1  2  3  4  5

   b Design limited to each discipline (mature design approach, analysis tool, design code)
   1  2  3  4  5

   c Resolution of non-automated interdisciplinary conflicts
   1  2  3  4  5

   d Relying heavily on previous experience
   1  2  3  4  5
5) Some traditional approaches of coordinating conflicts amongst different design disciplines are listed here, please fill in the following table (tick all that apply)?

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Frequency (please tick)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Frequently</td>
</tr>
<tr>
<td>a Chief designer (e.g. architect) proposes a resolution for coordinating different disciplinary designs based on experience</td>
<td></td>
</tr>
<tr>
<td>b. Negotiation amongst other designers in design meeting</td>
<td></td>
</tr>
<tr>
<td>c. Using some parameters based on building design code, for example structural designer leaves some holes for water pipe through floor in advance</td>
<td></td>
</tr>
<tr>
<td>d. Others, please specify:</td>
<td></td>
</tr>
</tbody>
</table>

6) In your experience, what are the constraints on the application of optimisation in building design?

7) In your opinion, do you need a model with the following properties which satisfy requirements of multidisciplinary building design in practice? (tick all that apply)

- Integrating different specialised software within one model
- Automatically smoothing conflict between disciplines
- Providing as many feasible design solutions as possible
- Considering various objective concurrently
- Others, please specify
Appendix Three: The COPGA Results of the Mathematical Problem
### Appendix Three

#### The COPGA Results of the Mathematical Problem

<table>
<thead>
<tr>
<th>Number of Solution</th>
<th>$x_1$</th>
<th>$x_3$</th>
<th>$y_{12}$</th>
<th>$y_{21}$</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.9602</td>
<td>9.8500</td>
<td>103.0676</td>
<td>30.0000</td>
<td>112.9177</td>
<td>-30.0000</td>
</tr>
<tr>
<td>2</td>
<td>2.3572</td>
<td>4.7199</td>
<td>8.1233</td>
<td>10.0111</td>
<td>12.8660</td>
<td>-10.0111</td>
</tr>
<tr>
<td>3</td>
<td>8.8304</td>
<td>10.0000</td>
<td>82.3667</td>
<td>27.9220</td>
<td>92.3673</td>
<td>-27.9220</td>
</tr>
<tr>
<td>6</td>
<td>7.7723</td>
<td>9.9992</td>
<td>65.1583</td>
<td>25.8547</td>
<td>75.1636</td>
<td>-25.8547</td>
</tr>
<tr>
<td>7</td>
<td>3.4696</td>
<td>9.9199</td>
<td>18.5527</td>
<td>17.0777</td>
<td>28.4913</td>
<td>-17.0777</td>
</tr>
<tr>
<td>8</td>
<td>8.5194</td>
<td>9.9970</td>
<td>77.1327</td>
<td>27.3310</td>
<td>87.1302</td>
<td>-27.3310</td>
</tr>
<tr>
<td>9</td>
<td>2.5023</td>
<td>5.9762</td>
<td>10.0826</td>
<td>11.5536</td>
<td>16.0897</td>
<td>-11.5536</td>
</tr>
<tr>
<td>12</td>
<td>7.5259</td>
<td>10.0000</td>
<td>61.5369</td>
<td>25.3809</td>
<td>71.5376</td>
<td>-25.3809</td>
</tr>
<tr>
<td>15</td>
<td>6.1340</td>
<td>10.0000</td>
<td>43.0721</td>
<td>22.7041</td>
<td>53.0723</td>
<td>-22.7041</td>
</tr>
<tr>
<td>17</td>
<td>5.9215</td>
<td>9.9315</td>
<td>40.6468</td>
<td>22.2285</td>
<td>50.5875</td>
<td>-22.2285</td>
</tr>
<tr>
<td>18</td>
<td>1.9104</td>
<td>9.8554</td>
<td>10.5379</td>
<td>15.0072</td>
<td>20.3945</td>
<td>-15.0072</td>
</tr>
<tr>
<td>19</td>
<td>6.6524</td>
<td>9.9924</td>
<td>49.3698</td>
<td>23.7065</td>
<td>59.3807</td>
<td>-23.7065</td>
</tr>
<tr>
<td>20</td>
<td>3.2088</td>
<td>10.0000</td>
<td>16.8159</td>
<td>17.3107</td>
<td>26.8162</td>
<td>-17.3107</td>
</tr>
<tr>
<td>23</td>
<td>5.0414</td>
<td>10.0000</td>
<td>31.3906</td>
<td>20.6688</td>
<td>41.4023</td>
<td>-20.6688</td>
</tr>
<tr>
<td>26</td>
<td>5.3419</td>
<td>10.0000</td>
<td>34.2899</td>
<td>21.2097</td>
<td>44.2899</td>
<td>-21.2097</td>
</tr>
<tr>
<td>32</td>
<td>1.3494</td>
<td>8.8146</td>
<td>8.1898</td>
<td>13.0199</td>
<td>17.0295</td>
<td>-13.0199</td>
</tr>
<tr>
<td>34</td>
<td>8.2778</td>
<td>10.0000</td>
<td>72.9459</td>
<td>26.8195</td>
<td>82.9907</td>
<td>-26.8195</td>
</tr>
</tbody>
</table>

240
<table>
<thead>
<tr>
<th>Number of Solution</th>
<th>$x_1$</th>
<th>$x_3$</th>
<th>$y_{12}$</th>
<th>$y_{21}$</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>9.4063</td>
<td>9.9997</td>
<td>92.6742</td>
<td>29.0343</td>
<td>102.6739</td>
<td>-29.0343</td>
</tr>
<tr>
<td>38</td>
<td>6.5261</td>
<td>9.9729</td>
<td>47.8531</td>
<td>23.4487</td>
<td>57.8264</td>
<td>-23.4487</td>
</tr>
<tr>
<td>39</td>
<td>7.4698</td>
<td>9.9960</td>
<td>60.6631</td>
<td>25.2715</td>
<td>70.6650</td>
<td>-25.2715</td>
</tr>
<tr>
<td>41</td>
<td>1.5347</td>
<td>8.1123</td>
<td>8.0329</td>
<td>12.4788</td>
<td>16.1490</td>
<td>-12.4788</td>
</tr>
<tr>
<td>43</td>
<td>5.9233</td>
<td>9.9328</td>
<td>40.6701</td>
<td>22.2335</td>
<td>50.6126</td>
<td>-22.2335</td>
</tr>
<tr>
<td>45</td>
<td>2.1814</td>
<td>5.5844</td>
<td>8.1056</td>
<td>10.6171</td>
<td>13.7029</td>
<td>-10.6171</td>
</tr>
<tr>
<td>47</td>
<td>8.4391</td>
<td>9.9950</td>
<td>75.2644</td>
<td>27.1475</td>
<td>85.5292</td>
<td>-27.1475</td>
</tr>
<tr>
<td>48</td>
<td>9.6244</td>
<td>9.9962</td>
<td>96.5764</td>
<td>29.5139</td>
<td>106.5941</td>
<td>-29.5139</td>
</tr>
<tr>
<td>50</td>
<td>9.8319</td>
<td>9.9829</td>
<td>101.1634</td>
<td>29.8608</td>
<td>111.3829</td>
<td>-29.8608</td>
</tr>
<tr>
<td>52</td>
<td>8.2828</td>
<td>9.9761</td>
<td>73.4451</td>
<td>26.8334</td>
<td>83.4745</td>
<td>-26.8334</td>
</tr>
<tr>
<td>53</td>
<td>5.2439</td>
<td>10.0000</td>
<td>33.3039</td>
<td>21.0036</td>
<td>43.3039</td>
<td>-21.0036</td>
</tr>
<tr>
<td>55</td>
<td>2.2413</td>
<td>5.5361</td>
<td>8.5154</td>
<td>10.6955</td>
<td>14.0606</td>
<td>-10.6955</td>
</tr>
<tr>
<td>56</td>
<td>9.7460</td>
<td>9.9922</td>
<td>98.9438</td>
<td>29.7347</td>
<td>108.9433</td>
<td>-29.7347</td>
</tr>
<tr>
<td>60</td>
<td>9.5047</td>
<td>9.9720</td>
<td>94.7262</td>
<td>29.2279</td>
<td>104.7663</td>
<td>-29.2279</td>
</tr>
</tbody>
</table>
Appendix Four:

The COPGA Results of the Design Scenario
### The COPGA Results of the Design Scenario

<table>
<thead>
<tr>
<th>Number of Solution</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1812.4</td>
<td>0</td>
<td>388.97</td>
<td>730.66</td>
<td>219.66</td>
<td>15.184</td>
<td>1528.5</td>
<td>10251</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>120</td>
<td>394.12</td>
<td>791.3</td>
<td>242.27</td>
<td>15.069</td>
<td>1808.2</td>
<td>8647.1</td>
</tr>
<tr>
<td>3</td>
<td>1128.8</td>
<td>3.5539</td>
<td>376.28</td>
<td>743.18</td>
<td>222.88</td>
<td>16.149</td>
<td>1580.1</td>
<td>9168.5</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>7.59E-12</td>
<td>453.99</td>
<td>743.91</td>
<td>226.43</td>
<td>16.637</td>
<td>1595.7</td>
<td>8981.1</td>
</tr>
<tr>
<td>5</td>
<td>1726.9</td>
<td>0.21998</td>
<td>380.79</td>
<td>731.97</td>
<td>219.77</td>
<td>15.127</td>
<td>1532.8</td>
<td>10116</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>94.82</td>
<td>413.53</td>
<td>784.62</td>
<td>239.91</td>
<td>15.181</td>
<td>1777.9</td>
<td>8696.9</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>86.482</td>
<td>410.94</td>
<td>778.82</td>
<td>236.91</td>
<td>15.148</td>
<td>1748.1</td>
<td>8715.1</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>110.27</td>
<td>394.17</td>
<td>787.73</td>
<td>241.07</td>
<td>15.07</td>
<td>1791.9</td>
<td>8665.5</td>
</tr>
<tr>
<td>9</td>
<td>1157.4</td>
<td>1.6472</td>
<td>373.27</td>
<td>741.25</td>
<td>222.47</td>
<td>16.02</td>
<td>1572</td>
<td>9220.7</td>
</tr>
<tr>
<td>10</td>
<td>1369.6</td>
<td>1.42E-06</td>
<td>361.99</td>
<td>743.18</td>
<td>226.43</td>
<td>16.637</td>
<td>1595.7</td>
<td>8981.1</td>
</tr>
<tr>
<td>11</td>
<td>1293.6</td>
<td>0</td>
<td>382.64</td>
<td>738.64</td>
<td>221.93</td>
<td>15.02</td>
<td>1561.9</td>
<td>9439.8</td>
</tr>
<tr>
<td>12</td>
<td>1781</td>
<td>0.003452</td>
<td>388.32</td>
<td>731.15</td>
<td>219.75</td>
<td>15.246</td>
<td>1530.4</td>
<td>10202</td>
</tr>
<tr>
<td>13</td>
<td>1900</td>
<td>41.825</td>
<td>422.68</td>
<td>761.55</td>
<td>229.02</td>
<td>16.323</td>
<td>1664.8</td>
<td>8831.6</td>
</tr>
<tr>
<td>14</td>
<td>1001.9</td>
<td>23.913</td>
<td>489.87</td>
<td>755.08</td>
<td>228.59</td>
<td>16.609</td>
<td>1644</td>
<td>8892.7</td>
</tr>
<tr>
<td>15</td>
<td>1000</td>
<td>10.325</td>
<td>453.25</td>
<td>748.54</td>
<td>226.88</td>
<td>16.631</td>
<td>1613.6</td>
<td>8939.4</td>
</tr>
<tr>
<td>16</td>
<td>1000</td>
<td>81.156</td>
<td>412.6</td>
<td>776.92</td>
<td>235.09</td>
<td>15.169</td>
<td>1736.5</td>
<td>8727.3</td>
</tr>
<tr>
<td>17</td>
<td>1678.9</td>
<td>0.010513</td>
<td>376.73</td>
<td>1678.9</td>
<td>0.010513</td>
<td>376.73</td>
<td>1535.2</td>
<td>10042</td>
</tr>
<tr>
<td>18</td>
<td>1493.9</td>
<td>0.26548</td>
<td>366.68</td>
<td>735.4</td>
<td>220.54</td>
<td>15.048</td>
<td>1546.1</td>
<td>9752.1</td>
</tr>
<tr>
<td>19</td>
<td>1001.6</td>
<td>69.827</td>
<td>414.27</td>
<td>772.6</td>
<td>232.75</td>
<td>16.32</td>
<td>1714.6</td>
<td>8757.2</td>
</tr>
<tr>
<td>20</td>
<td>1000.4</td>
<td>53.67</td>
<td>429.1</td>
<td>766.49</td>
<td>230.82</td>
<td>16.61</td>
<td>1687.9</td>
<td>8797.7</td>
</tr>
<tr>
<td>21</td>
<td>1000</td>
<td>60.754</td>
<td>432.94</td>
<td>769.42</td>
<td>231.89</td>
<td>16.607</td>
<td>1701.7</td>
<td>8777.9</td>
</tr>
<tr>
<td>22</td>
<td>1196.5</td>
<td>1.3438</td>
<td>375.63</td>
<td>740.56</td>
<td>222.3</td>
<td>16.012</td>
<td>1569.3</td>
<td>9283</td>
</tr>
<tr>
<td>23</td>
<td>1000</td>
<td>5.4873</td>
<td>466.26</td>
<td>746.56</td>
<td>226.9</td>
<td>16.634</td>
<td>1607.2</td>
<td>8958.5</td>
</tr>
<tr>
<td>24</td>
<td>1002.4</td>
<td>48.736</td>
<td>428.19</td>
<td>764.43</td>
<td>230.12</td>
<td>16.605</td>
<td>1678.5</td>
<td>8815.1</td>
</tr>
<tr>
<td>25</td>
<td>1000</td>
<td>14.145</td>
<td>452.63</td>
<td>750.23</td>
<td>227.03</td>
<td>16.629</td>
<td>1620.1</td>
<td>8924.8</td>
</tr>
<tr>
<td>26</td>
<td>1419.3</td>
<td>0</td>
<td>374.68</td>
<td>736.59</td>
<td>221.11</td>
<td>15.586</td>
<td>1552.2</td>
<td>9636.3</td>
</tr>
<tr>
<td>27</td>
<td>1000</td>
<td>60.007</td>
<td>429.44</td>
<td>769.06</td>
<td>231.67</td>
<td>16.602</td>
<td>1699.6</td>
<td>8779.9</td>
</tr>
<tr>
<td>28</td>
<td>1310.7</td>
<td>0.28626</td>
<td>353.96</td>
<td>740.03</td>
<td>221.3</td>
<td>15.162</td>
<td>1556.3</td>
<td>9465.6</td>
</tr>
<tr>
<td>29</td>
<td>1000</td>
<td>94.146</td>
<td>412.62</td>
<td>782.99</td>
<td>239.47</td>
<td>15.167</td>
<td>1770.5</td>
<td>8698.3</td>
</tr>
<tr>
<td>30</td>
<td>1252.3</td>
<td>0.045572</td>
<td>366.72</td>
<td>739.08</td>
<td>221.72</td>
<td>15.667</td>
<td>1562.1</td>
<td>9375.1</td>
</tr>
<tr>
<td>31</td>
<td>1000</td>
<td>16.416</td>
<td>465.2</td>
<td>751.44</td>
<td>227.34</td>
<td>16.628</td>
<td>1626.1</td>
<td>8916.4</td>
</tr>
<tr>
<td>32</td>
<td>1578.6</td>
<td>0</td>
<td>369.63</td>
<td>734.05</td>
<td>220.2</td>
<td>15.018</td>
<td>1540.7</td>
<td>9885.2</td>
</tr>
<tr>
<td>33</td>
<td>1000</td>
<td>3.1924</td>
<td>454.63</td>
<td>745.35</td>
<td>226.58</td>
<td>16.635</td>
<td>1601.3</td>
<td>8967.8</td>
</tr>
<tr>
<td>34</td>
<td>1523</td>
<td>4.82E-08</td>
<td>368.21</td>
<td>734.89</td>
<td>220.43</td>
<td>15.174</td>
<td>1544.2</td>
<td>9798.3</td>
</tr>
<tr>
<td>35</td>
<td>1637.1</td>
<td>0</td>
<td>372.74</td>
<td>733.18</td>
<td>219.99</td>
<td>15.135</td>
<td>1537.4</td>
<td>9976.7</td>
</tr>
</tbody>
</table>
# The COPGA Results of the Design Scenario (continued)

<table>
<thead>
<tr>
<th>Number of Solution</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>1546.3</td>
<td>1.94E-08</td>
<td>364.85</td>
<td>734.48</td>
<td>220.23</td>
<td>15.013</td>
<td>1542.1</td>
<td>9834.8</td>
</tr>
<tr>
<td>37</td>
<td>1000</td>
<td>0.76217</td>
<td>371.15</td>
<td>736.14</td>
<td>220.83</td>
<td>15.005</td>
<td>1549.7</td>
<td>9704.4</td>
</tr>
<tr>
<td>38</td>
<td>1129.5</td>
<td>1.0048</td>
<td>377.27</td>
<td>741.72</td>
<td>222.75</td>
<td>16.168</td>
<td>1574.6</td>
<td>9179.5</td>
</tr>
<tr>
<td>39</td>
<td>1464.4</td>
<td>0.53776</td>
<td>366.76</td>
<td>735.85</td>
<td>220.66</td>
<td>15.005</td>
<td>1548</td>
<td>9719.8</td>
</tr>
<tr>
<td>40</td>
<td>1000</td>
<td>89.74</td>
<td>414.83</td>
<td>780.79</td>
<td>238.19</td>
<td>15.206</td>
<td>1759</td>
<td>8707.9</td>
</tr>
<tr>
<td>41</td>
<td>1000</td>
<td>52.682</td>
<td>427.25</td>
<td>766.06</td>
<td>230.64</td>
<td>16.602</td>
<td>1685.8</td>
<td>8799.9</td>
</tr>
<tr>
<td>42</td>
<td>1473.8</td>
<td>0.53776</td>
<td>366.76</td>
<td>735.85</td>
<td>220.66</td>
<td>15.005</td>
<td>1548</td>
<td>9719.8</td>
</tr>
<tr>
<td>43</td>
<td>1239.5</td>
<td>0</td>
<td>367.97</td>
<td>739.28</td>
<td>221.82</td>
<td>15.726</td>
<td>1563.1</td>
<td>9355.3</td>
</tr>
<tr>
<td>44</td>
<td>1000</td>
<td>72.349</td>
<td>414.46</td>
<td>773.61</td>
<td>233.21</td>
<td>15.202</td>
<td>1719.5</td>
<td>8748.3</td>
</tr>
<tr>
<td>45</td>
<td>1199.7</td>
<td>4.50E-05</td>
<td>376.97</td>
<td>740.04</td>
<td>222.26</td>
<td>16.042</td>
<td>1567.4</td>
<td>9293.1</td>
</tr>
<tr>
<td>46</td>
<td>1002.1</td>
<td>76.323</td>
<td>415.61</td>
<td>775.12</td>
<td>233.95</td>
<td>16.395</td>
<td>1727.1</td>
<td>8742.2</td>
</tr>
<tr>
<td>47</td>
<td>1437.4</td>
<td>0</td>
<td>374.94</td>
<td>736.31</td>
<td>221.03</td>
<td>15.561</td>
<td>1551</td>
<td>9664.7</td>
</tr>
<tr>
<td>48</td>
<td>1793.8</td>
<td>0</td>
<td>388.6</td>
<td>730.95</td>
<td>219.71</td>
<td>15.222</td>
<td>1529.7</td>
<td>10222</td>
</tr>
<tr>
<td>49</td>
<td>1000</td>
<td>92.894</td>
<td>413.45</td>
<td>781.24</td>
<td>238.93</td>
<td>15.18</td>
<td>1762.5</td>
<td>8701</td>
</tr>
<tr>
<td>50</td>
<td>1002.5</td>
<td>18.281</td>
<td>494.54</td>
<td>752.7</td>
<td>227.96</td>
<td>16.599</td>
<td>1634</td>
<td>8913.6</td>
</tr>
<tr>
<td>51</td>
<td>1614.4</td>
<td>0</td>
<td>369.92</td>
<td>733.5</td>
<td>220.03</td>
<td>15.025</td>
<td>1538.4</td>
<td>9941.3</td>
</tr>
<tr>
<td>52</td>
<td>1000</td>
<td>31.303</td>
<td>501.4</td>
<td>758.47</td>
<td>229.98</td>
<td>16.57</td>
<td>1660.6</td>
<td>8846.7</td>
</tr>
<tr>
<td>53</td>
<td>1002.3</td>
<td>48.727</td>
<td>428.27</td>
<td>764.43</td>
<td>230.12</td>
<td>16.605</td>
<td>1678.5</td>
<td>8815</td>
</tr>
<tr>
<td>54</td>
<td>1001</td>
<td>79.459</td>
<td>422.56</td>
<td>760.96</td>
<td>228.83</td>
<td>16.524</td>
<td>1662.2</td>
<td>8837.3</td>
</tr>
<tr>
<td>55</td>
<td>1345.5</td>
<td>8.90E-07</td>
<td>358.68</td>
<td>737.5</td>
<td>221.05</td>
<td>15.247</td>
<td>1554.4</td>
<td>9521</td>
</tr>
<tr>
<td>56</td>
<td>1000.5</td>
<td>30.859</td>
<td>489.32</td>
<td>758.07</td>
<td>229.5</td>
<td>16.581</td>
<td>1657.3</td>
<td>8867</td>
</tr>
<tr>
<td>57</td>
<td>1660</td>
<td>3.81E-07</td>
<td>378.4</td>
<td>732.91</td>
<td>220.03</td>
<td>15.18</td>
<td>1536.8</td>
<td>10012</td>
</tr>
<tr>
<td>58</td>
<td>1000</td>
<td>79.21</td>
<td>413.98</td>
<td>776.23</td>
<td>234.48</td>
<td>15.191</td>
<td>1732.5</td>
<td>8731.8</td>
</tr>
<tr>
<td>59</td>
<td>1000</td>
<td>14.09</td>
<td>395.19</td>
<td>788.66</td>
<td>241.55</td>
<td>15.072</td>
<td>1796.2</td>
<td>8658.1</td>
</tr>
<tr>
<td>60</td>
<td>1000</td>
<td>28.4</td>
<td>489.31</td>
<td>757.03</td>
<td>229.2</td>
<td>16.585</td>
<td>1652.7</td>
<td>8874.3</td>
</tr>
</tbody>
</table>
Appendix Five: Matlab Programme Code
Starting Programme

function nsga_2(pop,gen)

%% function nsga_2(pop,gen)
% is a multi-objective optimization function where the input arguments are
% pop - Population size
% gen - Total number of generations

if nargin < 2
    error('NSGA-II: Please enter the population size and number of generations as input arguments.');
end

if isnumeric(pop) == 0 || isnumeric(gen) == 0
    error('Both input arguments pop and gen should be integer datatype');
end

pop = round(pop);
gen = round(gen);

[M, V, min_range, max_range] = objective_description_function();

chromosome = initialize_variables(pop, M, V, min_range, max_range);

chromosome = non_domination_sort_mod(chromosome, M, V);

for i = 1 : gen
    pool = round(pop/2);
    tour = 2;
    parent_chromosome = tournament_selection(chromosome, pool, tour);

    mu = 20;
    mum = 20;
    offspring_chromosome = genetic_operator(parent_chromosome, M, V, mu, mum, in_range, 
                                           max_range);

    [main_pop,temp] = size(chromosome);
    [offspring_pop,temp] = size(offspring_chromosome);
    clear temp
    intermediate_chromosome(1:main_pop,:) = chromosome;

Appendix Five

Matlab Programme Code

intermediate_chromosome(main_pop + 1 : main_pop + offspring_pop, 1 : M+V) = offspring_chromosome;
intermediate_chromosome = non_domination_sort_mod(intermediate_chromosome, M, V);

chromosome = replace_chromosome(intermediate_chromosome, M, V, pop);
if ~mod(i,100)
cle
fprintf('%d generations completed\n',i);
end
end

save solution.txt chromosome -ASCII

if M == 2
plot(chromosome(:,V + 1),chromosome(:,V + 2),'*');
elseif M == 3
plot3(chromosome(:,V + 1),chromosome(:,V + 2),chromosome(:,V + 3),'*');
end

Objective Function Description

function [number_of_objectives, number_of_decision_variables, min_range_of_decesion_variable, max_range_of_decesion_variable] = objective_description_function()

g = sprintf('Input the number of objectives: ');

number_of_objectives = input(g);
if number_of_objectives < 2
error('This is a multi-objective optimization function hence the minimum number of objectives is two');
end
g = sprintf('Input the number of decision variables: ');

c = input(g);
for i = 1 : number_of_decision_variables
c
g = sprintf('Input the minimum value for decision variable %d : ', i);
min_range_of_decesion_variable(i) = input(g);
Appendix Five

Matlab Programme Code

Initialize Variable

function f = initialize_variables(N, M, V, min_range, max_range)

min = min_range;
max = max_range;
K = M + V;
for i = 1 : N
    for j = 1 : V
        a(i,j) = min(j) + (max(j) - min(j))*rand(1);
    end
J= sosopt(a(:,:));
f(i,1:V)=J(1:V);
local(i,1:2)=J(4:5);
f(i,V + 1: K) = evaluate_objective(f(i,1:V),local(i,1:2),M, V);
end
Sort the Population based on Pareto Rank

function f = non_domination_sort_mod(x, M, V)
[N, m] = size(x);
clear m
front = 1;
F(front).f = [];
individual = [];
for i = 1 : N
    individual(i).n = 0;
    individual(i).p = [];
    for j = 1 : N
        dom_less = 0;
        dom_equal = 0;
        dom_more = 0;
        for k = 1 : M
            if (x(i,V + k) < x(j,V + k))
                dom_less = dom_less + 1;
            elseif (x(i,V + k) == x(j,V + k))
                dom_equal = dom_equal + 1;
            else
                dom_more = dom_more + 1;
            end
        end
        if dom_less == 0 && dom_equal <= M
            individual(i).n = individual(i).n + 1;
        elseif dom_more == 0 && dom_equal <= M
            individual(i).p = [individual(i).p j];
        end
    end
    if individual(i).n == 0
        x(i,M + V + 1) = 1;
        F(front).f = [F(front).f i];
    end
end
while ~isempty(F(front).f)
    Q = [];
    for i = 1 : length(F(front).f)
        if ~isempty(individual(F(front).f(i)).p)
for \( j = 1 : \text{length}(\text{individual}(\text{F(front).f(i)).p)) \)
\[ \text{individual}(\text{individual}(\text{F(front).f(i)).p(j))).n = \ldots \]
\[ \text{individual}(\text{individual}(\text{F(front).f(i)).p(j))).n - 1; \]
if \( \text{individual}(\text{individual}(\text{F(front).f(i)).p(j))).n == 0 \)
\[ x(\text{individual}(\text{F(front).f(i)).p(j)),M + V + 1) = \ldots \]
\[ \text{front} + 1; \]
\[ Q = [Q \text{individual}(\text{F(front).f(i)).p(j))]; \]
end
end
end
end
\]
\[ \text{front} = \text{front} + 1; \]
\[ \text{F(front).f} = Q; \]
end

[\text{temp, index_of_fronts}] = sort(x(:,M + V + 1));
for \( i = 1 : \text{length}(\text{index_of_fronts}) \)
\[ \text{sorted_based_on_front}(i,:) = x(\text{index_of_fronts}(i,:),); \]
end
\[ \text{current_index} = 0; \]
for \( \text{front} = 1 : (\text{length}(	ext{F}) - 1) \)
\[ \text{distance} = 0; \]
\[ y = []; \]
\[ \text{previous_index} = \text{current_index} + 1; \]
for \( i = 1 : \text{length}(	ext{F(front).f}) \)
\[ y(i,:) = \text{sorted_based_on_front}(\text{current_index} + i,:); \]
end
\[ \text{current_index} = \text{current_index} + i; \]
\[ \text{sorted_based_on_objective} = []; \]
for \( i = 1 : M \)
\[ [\text{sorted_based_on_objective, index_of_objectives}] = \ldots \]
\[ \text{sort}(y(:,V + i)); \]
\[ \text{sorted_based_on_objective} = []; \]
for \( j = 1 : \text{length}(\text{index_of_objectives}) \)
\[ \text{sorted_based_on_objective}(j,:) = y(\text{index_of_objectives}(j,:),); \]
end
\[ f_{\text{max}} = \ldots \]
\[ \text{sorted_based_on_objective}(\text{length}(\text{index_of_objectives}), V + i); \]
\[ f_{\text{min}} = \text{sorted_based_on_objective}(1, V + i); \]
Appendix Five

\[ y(index\_of\_objectives(length(index\_of\_objectives)),M + V + 1 + i) = \text{Inf}; \]
\[ y(index\_of\_objectives(1),M + V + 1 + i) = \text{Inf}; \]
for \( j = 2 : \text{length(index\_of\_objectives)} - 1 \)
\[ \text{next\_obj} = \text{sorted\_based\_on\_objective}(j + 1,V + i); \]
\[ \text{previous\_obj} = \text{sorted\_based\_on\_objective}(j - 1,V + i); \]
if \( f_{\text{max}} - f_{\text{min}} = 0 \)
\[ y(index\_of\_objectives(j),M + V + 1 + i) = \text{Inf}; \]
else
\[ y(index\_of\_objectives(j),M + V + 1 + i) = \frac{(\text{next\_obj} - \text{previous\_obj})}{f_{\text{max}} - f_{\text{min}}}; \]
end
end

distance = [];
distance(:,1) = zeros(length(F(front).f),1);
for \( i = 1 : M \)
\[ \text{distance}(:,1) = \text{distance}(:,1) + y(:,M + V + 1 + i); \]
end
\[ y(:,M + V + 2) = \text{distance}; \]
\[ y = y(:,1: M + V + 2); \]
\[ z(previous\_index:current\_index,:) = y; \]
end
f = z();

Crossover and Mutation Operator

function f = geneticOperator(parent\_chromosome, M, V, mu, mum, 1\_limit, u\_limit)
[N,m] = size(parent\_chromosome);
clear m
p = 1;
was\_crossover = 0;
was\_mutation = 0;
for \( i = 1 : N \)
if rand(1) < 0.9
\[ \text{child\_1} = []; \]
\[ \text{child\_2} = []; \]
\[ \text{parent\_1} = \text{round}(N*\text{rand}(1)); \]
if parent\_1 < 1
\[ \text{parent\_1} = 1; \]
end

251
end

parent_2 = round(N*rand(1));
if parent_2 < 1
    parent_2 = 1;
end

while isequal(parent_chromosome(parent_1,:),parent_chromosome(parent_2,:))
    parent_2 = round(N*rand(1));
    if parent_2 < 1
        parent_2 = 1;
    end
end

parent_1 = parent_chromosome(parent_1,:);
parent_2 = parent_chromosome(parent_2,:);

for j = 1 : V
    u(j) = rand(1);
    if u(j) <= 0.5
        bq(j) = (2*u(j))^((1/(mu+1)));
    else
        bq(j) = ((1/(2*(1 - u(j))))^((1/(mu+1)));
    end
    child_1(j) = ...
        0.5*(((1 + bq(j))*parent_1(j)) + (1 - bq(j))*parent_2(j));
    child_2(j) = ...
        0.5*(((1 - bq(j))*parent_1(j)) + (1 + bq(j))*parent_2(j));
    if child_1(j) > u_limit(j)
        child_1(j) = u_limit(j);
    elseif child_1(j) < l_limit(j)
        child_1(j) = l_limit(j);
    end
    if child_2(j) > u_limit(j)
        child_2(j) = u_limit(j);
    elseif child_2(j) < l_limit(j)
        child_2(j) = l_limit(j);
    end
end

J= sysopt(child_1);
child_1=J(1:3);
local_1=J(4:5);
child_1(:,V+1:M+V) = evaluate_objective(child_1,local_1,M, V);
J= sysopt(child_2);
child_2 = J(1:3);
local_2 = J(4:5);
child_2(:, V+1:M+V) = evaluate_objective(child_2, local_2, M, V);
was_crossover = 1;
was_mutation = 0;
else
parent_3 = round(N * rand(1));
if parent_3 < 1
parent_3 = 1;
end
child_3 = parent_chromosome(parent_3, :);
for j = 1 : V
r(j) = rand(1);
if r(j) < 0.5
delta(j) = (2 * r(j))^-((1/(muu + 1))) - 1;
else
delta(j) = 1 - (2 * (1 - r(j)))^-((1/(muu + 1)));
end
child_3(j) = child_3(j) + delta(j);
if child_3(j) > u_limit(j)
child_3(j) = u_limit(j);
elseif child_3(j) < l_limit(j)
child_3(j) = l_limit(j);
end
end
J = sysopt(child_3);
child_3 = J(1:3);
local_3 = J(4:5);
child_3(:, V+1:M+V) = evaluate_objective(child_3, local_3, M, V);
was_mutation = 1;
was_crossover = 0;
end
if was_crossover
child(p,:) = child_1;
child(p+1,:) = child_2;
was_crossover = 0;
p = p + 2;
elseif was_mutation
child(p,:) = child_3(1:1:M+V);
was_mutation = 0;
Appendix Five

Evaluate Objective Function

function f = evaluate_objective(x,l,M,V)

f = []; 

\[ f(1) = (0.034*9000*l(1)^2+0.154*l(2)^2*(3000+x(3)+l(1)+100))*7.85e-6; \]

\[ f(2) = \frac{(3.305e8-0.61e5*x(1)+1.3e6*x(2)-216.55*x(1)*x(2))/((2916+0.0297*x(1))*(233.12+x(2))+16740-5.58*x(1))+4(2.5e8+1.49e5*x(1)+2.5e6*x(2)+415.05*x(1)*x(2))/((3325.05-0.0036*x(1))*(233.12+x(2)))+1328*x(1)+1174.59+(3.85e5-128.34*x(1))/(233.12+x(2))+0.68*x(1)+3570 \]

if length(f) ~= M 
    error('The number of decision variables does not match you previous input. Kindly check your objective function');
end

Replace chromosome

function f = replace_chromosome(intermediate_chromosome,M,V,pop)

[N, m] = size(intermediate_chromosome);
[temp,index] = sort(intermediate_chromosome(:,M+V+1));
clear temp m
for i = 1:N
    sorted_chromosome(i,:) = intermediate_chromosome(index(i),:);
end
max_rank = max(intermediate_chromosome(:,M+V+1));
previous_index = 0;
for i = 1:max_rank
    current_index = max(find(sorted_chromosome(:,M+V+1) == i));
    if current_index > pop
        remaining = pop - previous_index;
        temp_pop = ... 
        sorted_chromosome(previous_index + 1 : current_index, :) = ... 
        [temp_sort,temp_sort_index] = ... 
        sort(temp_pop(:, M + V + 2), 'descend');
        for j = 1:remaining 
            f(previous_index + j,:) = temp_pop(temp_sort_index(j,:)); 
        end 
    end 
end
end
return;
elseif current_index < pop
   f(previous_index + 1 : current_index,:) = ...
   sorted_chromosome(previous_index + 1 : current_index,:);
else
   f(previous_index + 1 : current_index,:) = ...
   sorted_chromosome(previous_index + 1 : current_index,:);
   return;
end
previous_index = current_index;
end

Tournament selection

function [pop, variables] = size( chromosome);
rank = variables - 1;
distance = variables;
for i = 1 : pool_size
   for j = 1 : tour_size
      candidate(j) = round(pop*rand(1));
      if candidate(j) == 0
         candidate(j) = 1;
      end
      if j > 1
         while ~isempty(find(candidate(1 : j - 1) == candidate(j)))
            candidate(j) = round(pop*rand(1));
            if candidate(j) == 0
               candidate(j) = 1;
            end
         end
      end
   end
end
for j = 1 : tour_size
   c_obj_rank(j) = chromosome(candidate(j),rank);
   c_obj_distance(j) = chromosome(candidate(j),distance);
end
min_candidate = ...
find(c_obj_rank == min(c_obj_rank));
if length(min_candidate) == 1
   max_candidate = ...
   find(c_obj_distance(min_candidate) == max(c_obj_distance(min_candidate)));
   if length(max_candidate) == 1
      max_candidate = max_candidate(1);
   end
   f(i,:) = chromosome(candidate(min_candidate(max_candidate)),:);
else
   f(i,:) = chromosome(candidate(min_candidate(1)),:);
end
end

System-level optimisation

function J = sysopt(z)
Appendix Five
Matlab Programme Code

[xstar,Jstar] = fmincon('Jsys',z,[],[],[],[],[700,0,100],[3000,120,700],[],optimset('TolFun',1e-3));
J=[xstar,Jstar];
if Jstar<0.02
    M = ss1opt(xstar);
    N = ss2opt(xstar);
    J=[xstar,M(3),M(4), N(3)];
else
    J=[le6,Je6,Je6,Je6,Je6,Je6];
end

Subsystem 1 optimizer

function Jxstar = ss1opt(z)
[xstar,Jstar] =
    fmincon(@subobj1,[1000,100,200,200],[],[],[],[],[500,0,600,100],[3000,120,1200,400],@g1,optimset('Display','off','TolX', 1e-7),z);
Jxstar = [xstar,Jstar];

Subspace 2 optimizer

function Jxstar = ss2opt(z)
[xstar,Jstar] =
    fmincon(@subobj2,[1000,300,16],[1000,100,15],[3000,700,20],@g2,optimset('display','off','TolX', 1e-7),z);
Jxstar = [xstar,Jstar];

Subsystem 1 Objective Function

function f=subobj1(x,z)
f=(x(1)-z(1))^2+(x(2)-z(2))^2;

Subsystem 2 Objective Function

function f=subobj2(x,z)
f=(x(1)-z(1))^2+(x(2)-z(3))^2;

Subsystem 1 Constraint Function

function [c,ceq]=g1(x,z)
%this function is for structural analysis
[beam_moment,beam_shearforce,column_axialforce,column_moment]=plane_gangjia (x,z);
Pb=bending_strength(x);
Pcx=compressive_strength((3.65*(3000+z(3)+x(3)+100))/(x(4)));
Pcy=compressive_strength((6.46*(3000+z(3)+x(3)+100))/(x(4)));
Pbc=bending_strength_column(x,z);
c=[
    %Allowance beam bend moment
    (beam_moment)/(0.00902*Pb*x(3)^3)-1;
    %Allowance beam shear force
    (beam_shearforce)/(3.71*(x(3))^2)-1;
    %Allowance beam deflection
    ((3.92e11+2.576e5*(3000-x(1)+z(3)+x(3)+100)*(1.7*x(2)+186.3))/(x(3)^4))/25-1;
];
% Allowance column slenderness ratio
\((3.65*(3000+z(3)+x(3)+100)/(x(4)))/180-1;\)
\((6.46*(3000+z(3)+x(3)+100)/(x(4)))/180-1;\)

% For major axis in plane buckling
\((0.154*Pcx*x(4)^2)+((0.5*column\_moment)/(14.355*x(4)^3))*(1+(0.5*column\_axialforce)/(0.154*Pcx*x(4)^2))-1;\)

% For out-of-plane buckling
\((0.154*Pcy*x(4)^2)+(0.44*column\_moment)/(0.0522*Pbc*x(4)^3)-1;\)

% Comparing external wall depth with width of beam
\(354+3*x(2)-x(3);\)

ceq=[]

**Subsystem 2 Constraint Function**

function [c,ceq]=g2(x,z)

% This function is for thermal analysis.

c=[% Air change rate
\((3.305e8-0.61e5*x(1)+1.3e6*z(2)-216.55*x(1)*z(2))/(2916+0.0297*x(1))*(233.12+z(2))+16740-5.58*x(1)+4.25e8+1.49e5*x(1)+2.5e6*z(2)+415.05*x(1)*z(2))/(3325.05-0.0036*x(1))*(233.12+z(2))+1.328*x(1)+1174.59)*(233.12+z(2))+240975;
\]-(3.305e8-0.61e5*x(1)+1.3e6*z(2)-216.55*x(1)*z(2))/(2916+0.0297*x(1))*(233.12+z(2))+16740-5.58*x(1)+4.25e8+1.49e5*x(1)+2.5e6*z(2)+415.05*x(1)*z(2))/(3325.05-0.0036*x(1))*(233.12+z(2))+...
\]1.328*x(1)+1174.59)*(233.12+z(2))]-0.714;

% Duct noise
\((3.305e8-0.61e5*x(1)+1.3e6*z(2)-216.55*x(1)*z(2))/(2916+0.0297*x(1))*(233.12+z(2))+16740-5.58*x(1)+4.25e8+1.49e5*x(1)+2.5e6*z(2)+415.05*x(1)*z(2))/(3325.05-0.0036*x(1))*(233.12+z(2))+...
\]1.328*x(1)+1174.59)*(233.12+z(2))]-0.714;

ceq=[]

**Structural Analysis**

function [beam\_moment,beam\_shearforce,column\_axialforce,column\_moment]=plane\_gangjia(x,z)

cle
tic
format short

% Collect information about structural node and element (Step 1)

elem\_num=21;
n=16;

%T=['C2:Q' num2str(3*n+1)];
%A\_total\_data\_input\_info=xlsread('matrix\_info\_get.xls',1,T);
%num=A\_total\_data\_input\_info([1:elem\_num],1);
i\_num=[1,2,3,5,6,7,9,10,11,13,14,15,2,3,4,6,7,8,10,11,12,13,14,15];
%j\_num=A\_total\_data\_input\_info([1:elem\_num],2);
xi=A\_total\_data\_input\_info([1:elem\_num],3);
x2=A\_total\_data\_input\_info([1:elem\_num],4);

% Collect information about structural node and element (Step 1)

x1=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
x2=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

% Collect information about structural node and element (Step 1)

y1=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
y2=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

257
Appendix Five

.

•

-

Mat/ab Programme Code

O+z(3)+x(3)),31 OO+z(3)+x(3),2 *(31 OO+z(3 )+x(3)),3 *(31 OO+z(3)+x(3)),31 OO+z(3)+x(3 ),2 *(31 OO+z(3 )+
x(3)),3*(31 OO+z(3)+x(3))];
%y2=A_total_data_input_ info([ I :elem_ num], 6);
y2=[31 OO+z(3 )+x(3 ),2 *(31 OO+z(3 )+x(3) ),3 *(31 OO+z(3 )+x(3)),31 OO+z(3)+x(3 ),2 *(31 OO+z(3 )+x(3)),3 *(
31 OO+z(3 )+x(3)),31 OO+z(3 )+x(3 ),2 *(31 OO+z(3 )+x(3)),3 *(31 OO+z(3)+x(3)),31 OO+z(3 )+x(3),2*(31 OO+z
(3 )+x(3) ),3 *(31 OO+z(3 )+x(3) ),31 OO+z(3 )+x(3),2*(31 OO+z(3)+x(3)),3 *(31 OO+z(3 )+x(3)),31 OO+z(3)+x(
3),2 *(31 OO+z(3)+x(3)),3 *(31 OO+z(3 )+x(3 )),31 OO+z(3 )+x(3 ),2 *(31 OO+z(3 )+x(3)),3 *(31 OO+z(3 )+x(3) )] ;
%d=A_total_data_input_info([l:elem_num],7);
d=[x(4),x(4 ),x(4),x(4 ),x(4),x(4),x(4 ),x(4 ),x(4),x(4),x(4 ),x(4),x(3 ),x(3 ),x(3 ),x(3 ),x(3),x(3 ),x(3),x(3),x(3)
];

%type=A_total_data_input_info([l:elem_num],8);
type=[l, 1,1, 1,1, 1, l, 1,1, 1,1, 1,2,2,2,2,2,2,2,2,2];
%E=A_total_data_input_info([l:elem_num],9);
E=[205000,205000,205000,205000,205000,205000,205000,205000,205000,205000,205000,205000,20
5000,205000,205000,205000,205000,205000,205000,205000,205000];
fori= I :elem num
ifx2(i)-xl(i)=O
ify2(i)-yl(i)=O
disp
return
else ify2(i)-yl(i)>O
theta(i)=90;
else
theta(i)=270;
end
end
else
k=(y2(i)-y I (i))/(x2(i)-xl (i));
ifk>=O
theta(i)=atan(k)*!80/pi;
else
theta(i)= 180+atan(k)*l80/pi;
end
end
switch type(i)
case I %column cross-section property
A(i)=O.l54*d(i)A2;
I(i)=0.026l*d(i)A4;
case 2 %beam cross-section property
A(i)=0.034*d(i)A2;
I(i)=4.5le-3*d(i)A4;
end
end
%zhizuo_info=A_total_data_input_info([l :(3 *n) ], I 0);
zhizuo_info=[O 0
0
I
I
I
I
I
I
I
0
0
0
I
I
I
I
I
I
I
0
0
0
I
I
I
I
I
I
I
0
0
0
I
I
I
I
I
I
I
I
I
I
I];
%a_nonode=A_total_data_input_info([l :elem_num],ll );
a_nonode=[O
0
0
0
0
0
0
0
0
0
0
0
9000
9000
9000
9000
9000
9000
9000
9000];
9000
%key_nonode=A_total_data_input_info([l:elem_num],l2);
key_nonode=[8 8
8
8
8
8
8
8
8
8
8
8
I
I
I
I
I
I
I
I
I];
%node_load_info=A_total_data_input_info([l :(3*n)], 13);
0
0
0
0
node_load_info=[O
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

258


Appendix Five Matlab Programme Code

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
q_nonode=total_data_input_info([1:elem_num,14]);
for j=1:length(L)
    E1=E(j);A1=A(j);I1=I(j);L1=L(j);theta1=theta(j);
k=elemframe_jicheng(E1,A1,I1,L1,theta1);
    h(j)=k;
end
%Calculate global stiffness matrix
K=total_K_assemble(K,k,i_num(j),j_num(j));

%calculate element stiffness matrix (step 2)
L=sqrt((x2-x1)^2+(y2-y1)^2);
K=zeros(3*n);
for j=1:length(L)
    El=E(j);A1=A(j);I1=I(j);L1=L(j);theta1=theta(j);
    k=elemframe_jicheng(El,A1,I1,L1,theta1);
    h(j)=k;
%Calculate global stiffness matrix (step 3)
K=total_K_assemble(K,k,i_num(j),j_num(j));
end
%Calculate element load (step 4)
nonnode_load_info=zeros(1,n*3);
non_node_load_by_dangang=[];
for j=1:length(L)
    q=q_nonode(j);a=a_nonode(j);l=L(j);key=key_nonode(j);
switch key
case 1
    X1=X_flfSt=0;
    X2=X_flfSt=0;
    Y1=Y_flfSt=q*a*(1-2*l^2+2*l^3)/(2*l^3);
    Y2=Y_flfSt=q*a*(1-2*l^2+2*l^3)/(2*l^3);
    M1=M_flfSt=q*a*(6-8*l^2+3*l^3)/(12*l^2);
    M2=M_flfSt=q*a*(6-8*l^2+3*l^3)/(12*l^2);

case 2
    X1=X_flfSt=0;
    X2=X_flfSt=0;
    Y1=Y_flfSt=q*b*(1+2*l^2)/(l^2);
    Y2=Y_flfSt=q*b*(1+2*l^2)/(l^2);
    M1=M_flfSt=q*b*(2-3*l^2)/(l^2);
    M2=M_flfSt=-q*b*(2-3*l^2)/(l^2);

case 3
    X1=X_flfSt=0;
    X2=X_flfSt=0;
    Y1=Y_flfSt=-6*q*a*(1-3*l)/l^3;
    Y2=Y_flfSt=-6*q*a*(1-3*l)/l^3;
    M1=M_flfSt=q*a^3*(6-8*l^2+3*l^3)/12*l^2;
    M2=M_flfSt=-q*a^3*(6-8*l^2+3*l^3)/12*l^2;

case 4
    X1=X_flfSt=0;
    X2=X_flfSt=0;
    Y1=Y_flfSt=25*q*a*(2-3*l^2)/(2*l^2+1.6*l^3)/l^3;
    Y2=Y_flfSt=25*q*a*(2-3*l^2)/(2*l^2+1.6*l^3)/l^3;
    M1=M_flfSt=q*a^3*(3-l^2-a^2)/6;
    M2=M_flfSt=-q*a^3*(3-l^2-a^2)/6;

case 5
    X1=X_flfSt=q*a*(1-5*l)/l^2;
    X2=X_flfSt=-5*q*a^2/l;
    Y1=Y_flfSt=0;
    Y2=Y_flfSt=0;
    M1=M_flfSt=0;
    M2=M_flfSt=0;

case 6

259
Appendix Five

\[
X_{\text{first}} = -q*(l-a)/l; \\
X_{\text{end}} = q*a/l; \\
Y_{\text{first}} = 0; \\
Y_{\text{end}} = 0; \\
M_{\text{first}} = 0; \\
M_{\text{end}} = 0;
\]

\[\text{case 7}\]
\[
X_{\text{first}} = 0; \\
X_{\text{end}} = 0; \\
Y_{\text{first}} = q^2*a^2*(a/l+3*(l-a)/l)^2; \\
Y_{\text{end}} = q^2*a^2*(a/l+3*(l-a)/l)^2; \\
M_{\text{first}} = q^2*a^2/l; \\
M_{\text{end}} = q^2*a^2*(l-a)/l^2;
\]

\[\text{case 8}\]
\[
X_{\text{first}} = 0; \\
X_{\text{end}} = 0; \\
Y_{\text{first}} = 0; \\
Y_{\text{end}} = 0; \\
M_{\text{first}} = 0; \\
M_{\text{end}} = 0;
\]

\[
X_{Y_M} = [X_{\text{first}}, X_{\text{end}}, Y_{\text{first}}, Y_{\text{end}}, M_{\text{first}}, M_{\text{end}}];
\]

\[
X_{Y_M} = [X_{\text{first}}, Y_{\text{first}}, M_{\text{first}}, X_{\text{end}}, Y_{\text{end}}, M_{\text{end}}];
\]

non_node_load_by_dangang = [non_node_load_by_dangang; X_{Y_M}];
non_node_load_info(3*i_num(j)-2) = nonnode_load_info(3*i_num(j)-2) + X_{Y_M}(1);
non_node_load_info(3*i_num(j)-1) = nonnode_load_info(3*i_num(j)-1) + X_{Y_M}(3);
non_node_load_info(3*i_num(j)) = nonnode_load_info(3*i_num(j)) + X_{Y_M}(5);
non_node_load_info(3*j_num(j)-2) = nonnode_load_info(3*j_num(j)-2) + X_{Y_M}(2);
non_node_load_info(3*j_num(j)-1) = nonnode_load_info(3*j_num(j)-1) + X_{Y_M}(4);
non_node_load_info(3*j_num(j)) = nonnode_load_info(3*j_num(j)) + X_{Y_M}(6);
end

\[
\text{load_node_total} = \text{nonnode_load_info} + \text{node_load_info};
\]

\[
\text{load_node_total} = \text{load_node_total};
\]

\[
\text{u_info} = \text{find}(\text{zhizuo_info} = 0);
\]

\[
\text{K_tiaozheng} = \text{K}(\text{u_info'}, \text{u_info'});
\]

\[
\text{load_node_total_tiaozheng} = \text{load_node_total}(\text{u_info'});
\]

\[
\text{u_weiyi} = \text{K_tiaozheng}\text{load_node_total_tiaozheng};
\]

\[
\text{U} = \text{zeros}(3*n, 1);
\]

\[
\text{U}(\text{u_info}(1), 1) = \text{u_weiyi}(1);
\]

\[
\text{F_neili} = \text{K}\times\text{U};
\]

\[
\text{for} \ i = 1: \text{length}(\text{u_info})
\]
\[
\text{E1}=\text{E}(1); \text{A1}=\text{A}(1); \text{L1}=\text{L}(1); \text{theta1}=\text{theta}(1);
\]
\[
\text{u}(:, i) = \text{U}(3*i_num(i)-2:3*i_num(i), 3*j_num(i)-2:3*j_num(i), 1); \\
\text{u1} = \text{u}(:, i);
\]
\[
\text{F_element_node_vector}(, i) = \text{gangjia_Element_Forces}(	ext{E1}, \text{A1}, \text{L1}, \text{theta1}, \text{u1});
\]
\[
\text{F_element_node_vector}(, i) = \text{F_element_node_vector}(, i) - \text{non_node_load_by_dangang}(:, i);
\]
\end{enumerate}

\[
\text{F_element_node_vector} = \text{F_element_node_vector};
\]

\[
\text{beam_moment} = \text{abs}(\text{F_element_node_vector}(16, 3));
\]
\[
\text{beam_shearforce} = \text{abs}(\text{F_element_node_vector}(19, 2));
\]
\[
\text{column_axialforce} = \text{abs}(\text{sum}(\text{F_element_node_vector}(16:21, 2)));
\]
\[
\text{column_moment} = \text{abs}(\text{F_element_node_vector}(4, 6));
\]

260
Appendix Six:
Evaluation Questionnaire
Appendix Six

Evaluation Questionnaire

Semi-structured interview template for evaluating the Pareto Genetic Algorithm based on Collaborative Optimisation Model

The aim of this work is to evaluate the model that was developed to facilitate solving multi-objective multidisciplinary building design problems. This evaluation process begins with the interviewee’s understandings of the contemporary complex building design; the evaluator then briefly explains features and capabilities of this model. This is followed by description of the outcome from this model. The completion of the interview will validate this research from three aspects of research rational, model’s process and model’s outcome.

Information on respondents

1) The respondent’s experience has involved working as a:
Architect Designer (...), Structural Engineer (...), HVAC Engineer (....)  
Client ( )  Other, please specify ________

2) Your experience in building design ________ years.

3) Did you work as design coordinator?  
   Yes( ) / No( )

3.1) If yes, Can you briefly describe this post’s duties?

Part One: Model Assessment

4) Is the design structure matrix developed in this research an effective approach of identifying relationships amongst different disciplines? Why?

5) In your opinion, is decomposition based on discipline better than other approaches, such as based on function room, design phase? If not, do you have other approach of decomposing the complex building design?
6) To evaluation framework

<table>
<thead>
<tr>
<th>Item</th>
<th>Statement</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The COPGA framework ensures that all the essential characteristics of building design optimisation are represented.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Do you agree that the COPGA framework adopts a robust decomposition scheme?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a) Implement easily this decomposition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b) Reduces the dimension of the design problem.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c) Follows current organisational structure for building design</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d) Takes full advantage of disciplinary knowledge and analyses.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The COPGA framework is a good decision support tool for building design.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>The COPGA framework improves computational efficiency for building design optimisation.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The COPGA framework is easy to extend for more than two disciplinary designs.</td>
<td></td>
</tr>
</tbody>
</table>

7) Do you think the coordination strategy used in the model is an effective approach to solving conflicts between different disciplinary designs? Why?  

8) In your opinion, do generic Algorithm (GA) adopted in this model improve the capability of exploring potential design solution? Why?  

Part Two: Outcome Assessment

9) Which parts of this COPGA prototype impressed you most and why?  

10) Which parts of this COPGA prototype fell short of your expectations and why?  

263
Part Three: Industry application

11) What might discourage people from using the COPGA prototype?

12) In what ways can the COPGA prototype be further improved?