NEW GENETIC ALGORITHMS FOR CONSTRAINED OPTIMISATION AND APPLICATIONS TO DESIGN OF COMPOSITE LAMINATES

Tat Loong Yu

Thesis submitted to the
University of Wales
in candidature for the degree of
Doctor of Philosophy

December 2004

Cardiff School of Engineering
Cardiff University

DECLARATION AND STATEMENTS

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (Candidate)
Date ....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated.

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (Candidate)
Date ....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (Candidate)
Date ....

ACKNOWLEDGEMENTS

There are those who lead, and there are those who guide, and then there are those who do both. I would like to thank my supervisor Dr David Kennedy for leading and guiding me throughout this research work. In times when progress was slow, his unswerving support and patience is highly appreciated. I particularly appreciated the freedom bestowed on me to manage my research at my own pace.

I would also like to extend my gratitude out to Dr Chris Williams for introducing me to genetic algorithms during my undergraduate years at the University of Bath. It is he who revealed to me the beauty in the structures of nature.

I would also like to thank my parents, Mr. Yu Tiong Seng and Madam Lim Yoke Khim, for their love, encouragement and support. I cannot forget my sister and brother in-law, Madam Yu Su Yin and Mr. Cheng Heng Jiang, for showing me the true meaning of grit and determination in times of adversity. It is also with their blessings that this work is possible.

I like to thank Mr. Mah Sau Cheong for giving all the encouragement and support to pursue my dreams and ambitions. He is a great mentor and a guiding light at the end of a long dark tunnel.

Last but not least, I thank the love of my life, Miss Chiam Fui Suan, for her undivided love, patience and understanding for the many years that I have been away pursuing my research. Like the break of dawn, everything comes gradually, and at its appointed hour.

ABSTRACT

A general purpose constraint handling technique for genetic algorithms (GA) is developed by borrowing principles from multi-objective optimisation. This is in view of the many issues still facing constraint handling in GA, particularly in the number of control parameters that overwhelms the user, as well as other GA parameters, which are currently lacking in heuristics to guide successful implementations. Constraints may be handled as individual objectives of decreasing priorities or by a weighted-sum measurement of normalised violation, as would be done in multi-objective scenarios, with full consideration of the main cost function. Rather than the unnecessary specialisation seen in many new heuristics proposed for GA, the simplicity, generality and flexibility of the technique is maintained, where several options such as partial or full constraint evaluation, tangible or Pareto-ranked fitness, and implicit dominance evaluation are presented. By reducing the number of constraint evaluations, these options increase the probability of discovering optimal regions, and hence increase GA efficiency. Studies in applications to a constrained numerical problem, and to the design of realistic composite laminate plates and structures, serve to demonstrate the ease of implementation and general reliability in heavily constrained problems. The difference in the dynamics of partial or full violation knowledge showed that while the former reduced the overall number of constraint evaluations performed, the latter compromises for the expense of full constraint evaluations in the reduced number of GA generations, whether in terms of discovering feasible regions or optimal solutions. The benefit of partial or full constraint evaluations is subjective, as it ultimately depends on the trade-off in the computational cost of constraint evaluations and GA search.

其出弥远，其知弥少。
The more one knows, the less one understands.

To my parents.

TABLE OF CONTENTS

DECLARATION AND STATEMENTS .... **I**

ACKNOWLEDGEMENTS .... **II**

ABSTRACT .... **III**

TABLE OF CONTENTS .... **V**

LIST OF FIGURES .... **IX**

LIST OF TABLES .... **XV**

PRINCIPAL NOMENCLATURE .... **XVII**

1 INTRODUCTION **1**
1.1 MOTIVATIONS, AIMS AND OBJECTIVES 1
1.2 LITERATURE REVIEW 1
1.2.1 GENETIC ALGORITHMS 2
1.2.1.1 DEVELOPMENTS IN SCHEMA EVOLUTIONARY THEORIES 2
1.2.1.2 GENERAL PROGRESS 4
1.2.2 COMPOSITE PLATES AND PANEL ASSEMBLIES 6
1.2.2.1 PLATE DESIGN AND OPTIMISATION 6
1.2.2.2 PANEL ASSEMBLIES DESIGNS AND OPTIMISATIONS 7
1.3 GENERAL OVERVIEW 9

2 GENETIC ALGORITHMS **11**
2.1 INTRODUCTION 11
2.2 FRAMEWORK AND TERMINOLOGY 12
2.3 GENETIC STRUCTURES, $\mathbf{X}^{E}$ 16
2.3.1 BINARY ENCODING 17
2.3.2 DIRECT ENCODING 19
2.4 MEASUREMENTS OF PERFORMANCE, $\boldsymbol{\mu}$ 20
2.4.1 EVALUATIVE FEEDBACK AND THE ROLES OF $\boldsymbol{\mu}$ 20
2.4.1.1 GENETIC CO-ADAPTATION 21
2.4.1.2 SCHEMA GROWTH 22
2.4.2 FITNESS EVALUATIONS AND PROBABILITY DISTRIBUTIONS 23

2.4.3 SCALING OF FITNESS 24
2.4.4 CONSTRAINT HANDLING 26
2.5 GENETIC OPERATORS, **Ω** 27
2.5.1 REPLICATION 27
2.5.1.1 STOCHASTIC METHODS 28
2.5.1.2 STATIC AND OTHER METHODS 31
2.5.2 INTERACTION 33
2.5.2.1 CROSSOVERS AND SEXUAL OPERATORS 33
2.5.2.2 MUTATIONS AND ASEXUAL OPERATORS 36
2.5.3 REPLACEMENT 38
2.6 ADAPTIVE PLANS, **τ** 40
2.6.1 PARALLEL AND DISTRIBUTED IMPLEMENTATIONS 41
2.6.2 CLUSTERS AND NICHE IDENTIFICATION 42
2.6.3 EVOLUTIONARY ALGORITHMS 43
2.7 CLOSING REMARKS 44

3 CONSTRAINED OPTIMISATION WITH GENETIC ALGORITHMS 46
3.1 INTRODUCTION 46
3.2 CONSTRAINTS IN GENETIC ALGORITHMS – A LITERATURE REVIEW 48
3.2.1 DEATH SENTENCE 50
3.2.2 PENALTY FUNCTIONS 51
3.2.3 MULTI-OBJECTIVE CONSTRAINT HANDLING 55
3.3 FEASIBILITY OF GENOTYPES: THE MULTI-OBJECTIVE APPROACH 57
3.3.1 LEXICOGRAPHICAL COMPARISON OF PRIORITISED CONSTRAINTS 60
3.3.2 SUM VIOLATION OF NON-PRIORITISED CONSTRAINTS 63
3.3.3 FITNESS OF INFEASIBLE GENOTYPES 65
3.3.3.1 PARETO-RANKING OF GENOTYPES 65
3.3.3.2 PENALTY-ASSIGNMENT OF GENOTYPES 67
3.3.4 PARTIAL KNOWLEDGE OF VIOLATION 69
3.4 CASE STUDY ON CONSTRAINED NUMERICAL OPTIMISATION 71
3.4.1 QUADRATIC TEST FUNCTION 72
3.4.2 EXPERIMENTATION FRAMEWORK 72
3.4.2.1 NON-PRIORITISED PARETO-RANKING 74
3.4.2.2 NON-PRIORITISED PARETO-RANKING WITH LAZY EVALUATION 75
3.4.2.3 PRIORITISED PARETO-RANKING 75
3.4.2.4 PRIORITISED PARETO-RANKING WITH LAZY EVALUATION 76
3.4.2.5 NON-PRIORITISED PENALTY-ASSIGNED 76
3.4.3 RESULTS AND DISCUSSION 77
3.5 CONCLUSION AND CLOSING REMARKS 85

4 ANALYSIS OF COMPOSITE LAMINATES 87
4.1 INTRODUCTION 87
4.2 COMPOSITE LAMINATES 88
4.2.1 STRESS-STRAIN RELATIONSHIP OF AN ORTHOTROPIC LAMINA 91
4.2.2 CONSTITUTIVE RELATIONSHIP OF A LAMINATE 93
4.3 PRISMATIC PLATE ASSEMBLIES 96
4.3.1 FORCE-DISPLACEMENT RELATIONSHIP OF A COMPONENT PLATE 98
4.3.2 STIFFNESS MATRIX OF A COMPONENT PLATE 100
4.3.3 TRANSFORMATION OF THE PLATE STIFFNESS MATRIX 102
4.3.4 BUCKLING AND THE EIGENVALUE PROBLEM 105
4.3.4.1 WITTRICK-WILLIAMS ALGORITHM 106

4.3.4.2 BOUNDARY AND SUPPORT CONDITIONS 108
4.4 DESIGN AND ANALYTICAL TOOLS 110
4.4.1 VIPASA 111
4.4.2 VICON 111
4.4.3 VICONOPT 112
4.5 CLOSING REMARKS 114

5 DESIGN OF LAMINATES 116
5.1 INTRODUCTION 116
5.2 MASS MINIMISATION DESIGN PROBLEM FORMULATION 118
5.2.1 BUCKLING CONSTRAINT 119
5.2.2 BALANCED AND SYMMETRIC CONSTRAINT 120
5.2.3 DISCRETE THICKNESS CONSTRAINT 121
5.3 MASS MINIMISATION WITH GENETIC ALGORITHMS 122
5.3.1 GENETIC STRUCTURES 122
5.3.1.1 REPRESENTATION OF FIBRE ORIENTATION 123
5.3.1.2 REPRESENTATION OF DISCRETE THICKNESS 124
5.3.2 MEASUREMENT OF PERFORMANCE 125
5.3.3 GENETIC OPERATORS 126
5.3.4 ADAPTIVE PLAN 126
5.4 BENCHMARK LAMINATE 127
5.4.1 PROBLEM DEFINITION 128
5.4.2 PARAMETER ENCODING 129
5.4.3 COMPARISON LAMINATE 129
5.4.4 RESULTS AND DISCUSSION 131
5.5 LAMINATE WITH LONGITUDINAL VOIDS 135
5.5.1 PROBLEM DEFINITION 136
5.5.2 PARAMETER ENCODING 138
5.5.3 RESULTS AND DISCUSSION 138
5.6 GENERALISED LAMINATE WITH LONGITUDINAL VOIDS 143
5.6.1 PROBLEM DEFINITION 143
5.6.2 PARAMETER ENCODING 144
5.6.3 RESULTS AND DISCUSSION 145
5.7 CONCLUSIONS AND CLOSING REMARKS 147

6 DESIGN OF STIFFENED PANEL ASSEMBLIES 150
6.1 INTRODUCTION 150
6.2 BOX-SECTION ANALOGY OF LAMINATE WITH LONGITUDINAL VOIDS 151
6.2.1 PROBLEM DEFINITION 153
6.2.2 GA PROPERTIES AND SETUP 154
6.2.3 RESULTS AND DISCUSSION 155
6.3 DISCRETE THICKNESS DESIGN OF STIFFENED PANELS 159
6.3.1 PROBLEM DEFINITION 161
6.3.2 OPTIMISER CONFIGURATION AND SETUP 163
6.3.3 RESULTS AND DISCUSSION 164
6.4 WEIGHT AND COST DESIGN OF STIFFENED SANDWICH-PANELS 169
6.4.1 PROBLEM DEFINITION 171
6.4.2 WEIGHT-COST TRADEOFFS 172
6.4.3 OPTIMISER PARAMETERS AND CONFIGURATION 173
6.4.4 RESULTS AND DISCUSSION 175
6.4.5 VARIANT MATERIALS 181

6.5 CONCLUSION AND CLOSING REMARKS 185

7 CONCLUSIONS AND FUTURE WORK 187
7.1 GENETIC ALGORITHMS 187
7.1.1 LOW CARDINALITY GENETIC ALPHABETS 187
7.1.2 SEARCH CONSTRAINTS 188
7.1.2.1 DYNAMIC PRIORITISATION OF CONSTRAINTS – FUTURE WORK 189
7.1.2.2 CLUSTERING FOR CONSTRAINTS SPECIATION – FUTURE WORK 190
7.1.3 IMPROVING SEARCH RESOLUTION 191
7.1.3.1 DIVIDE-AND-CONQUER – FUTURE WORK 192
7.1.3.2 GENOTYPE CREEP – FUTURE WORK 193
7.2 COMPOSITE LAMINATES 194
7.2.1 DESIGN CONSTRAINTS AND OBJECTIVES 195
7.2.2 SHAPE OPTIMISATION 195
7.3 CLOSING REMARKS 197

REFERENCES 198

APPENDIX I: GRAY CODE CONVERSIONS 211

APPENDIX II: UNIQUE SOLUTIONS TO THE BENCHMARK PLATE 212

APPENDIX III: BUCKLING MODES FOR LAMINATE WITH 5 LONGITUDINAL VOIDS 214

APPENDIX IV: ISOMETRIC PLOTS OF J3 PANEL BUCKLING 215

LIST OF FIGURES

Figure 2-1 Visualisation of RWS with 8 genotypes in the population with corresponding probability of selection. 30

Figure 2-2 Visualisation of SUS, which is similar to RWS, except with the addition of equally spaced markers. 30

Figure 2-3 1PC on two parent genotypes, resulting in two offspring. 34

Figure 2-4 Uniform mutation and the flip from a 0 to 1 at the 5th locus. 37

Figure 3-1 Directional approach path in prioritised constraints. 62

Figure 3-2 Omnidirectional approach paths in non-prioritised constraints. 65

Figure 3-3 Convergence behaviour of all ten GAPR runs under steady-state replacement strategy with respect to the population average objective value. 79

Figure 3-4 Population standard deviation in terms of objective value of all ten GAPR runs.. 79

Figure 3-5 Number of generations required on discovering the optimum of QTC 80

Figure 3-6 Number of generations required on discovering the first fully feasible (zero violation) solution to QTC. 83

Figure 3-7 Normalised violation for the best genotype observed in every generation at the start of the evolution for all runs of methods GAPR and GAPR-P on QTC. 83

Figure 3-8 Increase in the total number of constraint evaluations over successive generations for all runs of GAPR and GAPR-P on QTC. 84

Figure 3-9 Increase in the total number of constraint evaluations over successive generations for all runs of GAPR and GAPR-LZ on QTC. 84

Figure 3-10 Savings in constraint evaluations through the integration of artificial prioritisation and lazy evaluations of constraints for QTC........ 85

Figure 4-1 Constituents of a fibre-reinforced composite laminate and the main material axes together with stresses acting on it........ 90

Figure 4-2 Lamina with an offset angle θ between its material axes and the governing laminate stresses with respect to the laminate reference axes........ 93

Figure 4-3 Lamina numbering system........ 93

Figure 4-4 Sign conventions for membrane forces and bending moments........ 94

Figure 4-5 Basic force and axis system for a component plate in an assembly........ 98

Figure 4-6 Perturbation edge forces, displacements and nodal lines of a component plate in an assembly........ 99

Figure 4-7 Connection modelling between component plates........ 102

Figure 4-8 Plate eccentric edge connections together with corresponding dimensions and displacements of a component plate........ 103

Figure 4-9 Rotational transformation from member axes to global axes........ 103

Figure 4-10 Straight nodal lines for isotropic plates or anisotropic plates without shear, compatible with simply support transverse ends........ 109

Figure 4-11 Skewed nodal lines as a result of anisotropy and/or shear loading, resulting in incompatibility with simply supported transverse ends assumption........ 109

Figure 4-12 VICONOPT continuous design strategy........ 113

Figure 5-1 Fibre orientations in consideration and corresponding encoded chromosomal representation........ 123

Figure 5-2 Developed GA optimiser framework with prioritisation and lazy evaluation of constraints........ 127

Figure 5-3 Design variables, layer fibre orientation and corresponding discrete thickness, of the benchmark laminate. 128

Figure 5-4 Comparison laminate for the benchmark problem. 130

Figure 5-5 Number of generations required until convergence for benchmark laminate problem. 133

Figure 5-6 General convergence trend of runs conducted for the benchmark laminate. 133

Figure 5-7 Typical laminate with 3 longitudinal voids. 136

Figure 5-8 General setup of the 3 cases studied for laminate with longitudinal voids. 137

Figure 5-9 Population average mass of runs from the case with 3 longitudinal voids 140

Figure 5-10 Scaled dimensions of a repeating element in the best solution of the laminate with 5 voids. 140

Figure 5-11 Contour plot of the governing critical buckling mode, which is local, for the laminate with 5 longitudinal voids 141

Figure 5-12 Overall global discrete optimum mass of laminates with different number of voids with regions A and B composing of only 90° and 0° plies respectively, and the distribution of mass between the regions 142

Figure 5-13 Schematic diagram of the generalised laminate with longitudinal voids. 143

Figure 5-14 Constraints on the breadth of region B in a single repeating element. 145

Figure 5-15 Overall global discrete optimum mass of laminate with different number of voids with regions A and B. 146

Figure 5-16 Scaled diagrams illustrating the breadth to thickness ratio of region B in the best obtained laminates. 147

Figure 5-17 Orthogrid panel concept. 149

Figure 6-1 Typical box-section with 3 voids. 152

Figure 6-2 Support conditions for an arbitrary box-section. 152

Figure 6-3 Box-section design cases with 3, 4 and 5 repeating elements. 153

Figure 6-4 Distribution of mass between skin and web of a fixed-breadth box-section as number of elements increases with standard stacking sequence. 157

Figure 6-5 Scaled diagram illustrating the dimensional relationships of the best solution of a repeating element of the box-section with 5 elements. 157

Figure 6-6 Optimum mass for a box-section with 5 elements. 158

Figure 6-7 Three cases conducted for each set of stiffened panel assemblies. 161

Figure 6-8 Components of a stiffened panel and connection boundaries considered in the design. 162

Figure 6-9 Skin contour plot of the governing buckling mode of the best solution for the J3 panel. 167

Figure 6-10 Scaled diagrams of the best solution found for each of the six cases of stiffened panels considered, together with longitudinal boundary support conditions. 168

Figure 6-11 Typical cross-section of sandwich construction, and forces acting on an element of a panel in flexure. 170

Figure 6-12 Stacking sequence of all component plates, i.e., the skin, web and flange of the stiffened panel. 171

Figure 6-13 Graphical representation of a Pareto front with convex and concave regions. 173

Figure 6-14 Two phenotypic representation depending on the additional on/off switch chromosome. 174

Figure 6-15 Effects of the switch chromosome on web chromosomes. 175

Figure 6-16 Percentage change in mass and cost relative to the minimum mass case (1.0 factor) with respect to trade-off factor. 177

Figure 6-17 Distribution of genotypes in terms of mass and cost at selection generations. 178

Figure 6-18 Distribution of best solutions (sub-optimal) of each generation of all runs across all tradeoffs and Pareto points... 179

Figure 6-19 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.96 – 1.00 and Pareto points.. 179

Figure 6-20 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.90 – 0.94 and Pareto points.. 180

Figure 6-21 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.00 – 0.80 and Pareto points.. 180

Figure 6-22 Concatenation of the material 1-bit chromosome to the genotype.................... 182

Figure 6-23 Spread of best individuals (sub-optimal) at each generation at all tradeoffs of all runs conducted for the CFRP/GFRP variant case and Pareto points.......................... 183

Figure 6-24 Spread of best individuals (sub-optimal) at each generation at factor 1.0 of all runs conducted for the CFRP/GFRP variant case and Pareto points.......................... 183

Figure 6-25 Spread of best individuals (sub-optimal) at each generation at factor 0.9 of all runs conducted for the CFRP/GFRP variant case and Pareto points.......................... 184

Figure 6-26 Spread of best individuals (sub-optimal) at each generation at factor 0.8 of all runs conducted for the CFRP/GFRP variant case and Pareto points.......................... 184

Figure 6-27 Spread of best individuals (sub-optimal) at each generation at factors 0.0 to 0.7 of all runs conducted for the CFRP/GFRP variant case and Pareto points...................... 185

Figure 7-1 Feasible regions imposed by the first and second constraints, and the fully feasible region located at the intersection of both.. 190

Figure 7-2 Boundary induced by the creep operator and movement from an initial to a locally optimal location. 194

Figure 7-3 Variation in repeating geometrical shapes and resulting structural members...... 196

Figure 7-4 Topological grid with locations that require mass (shaded) and a possible interpretation of the resulting shape.. 196

Figure AI-1 Conversion procedures for Binary-to-Gray and vice-versa. 211

Figure AI-2 Equivalent decimal, standard binary and Gray codes. 211

Figure AII-1 List of unique optima found for benchmark laminate problem. 213

Figure AIII-1 Contour plots of non-critical buckling modes for laminate with 5 longitudinal voids problem.. 214

Figure AIV-1 Isometric plots of the critical buckling modes of the best found solution for J3 panel.. 215

LIST OF TABLES

Table 3-1 Number of genes required for each design variable in QTC to reflect the number of significant figures.......... 73

Table 3-2 GA parameters and their corresponding values used in the QTC experimentation framework.......... 74

Table 3-3 Results for 10 independent runs of penalty-based methods, M1 to M5, together with proposed methods and variants, GAPR to GAPA, on QTC.......... 77

Table 4-1 Properties of a selection of fibre-reinforced composite materials.......... 90

Table 5-1 Material properties for CFRP.......... 129

Table 5-2 Design variables for the comparison study and the bounds, which is the number of standard plies.......... 131

Table 5-3 Global discrete optima obtained on the best run out of 10 independent runs for the comparison laminate problem.......... 131

Table 5-4 Number of unique optima found in each of the ten runs conducted.......... 134

Table 5-5 Unique set of discrete optima returned from the first run of the benchmark laminate problem.......... 134

Table 5-6 Best solutions, in terms of mass, returned for each of the 3 cases with longitudinal voids.......... 139

Table 5-7 Best solutions found from all runs of all cases of the laminate with longitudinal voids.......... 139

Table 5-8 Critical buckling load factor at several of the wavelengths considered for the laminate with 5 voids.......... 142

Table 5-9 Optima for laminates with 3 and 4 voids if restricted only to 90° and 0° plies in regions A and B. 142

Table 5-10 Mass penalty incurred by the best observed feasible solution of other configurations of composite laminate. 146

Table 6-1 Fixed web breadth of the cases considered for the design of box-sections. 153

Table 6-2 Best unique solutions found for box-sections with 3, 4 and 5 repeating elements respectively and their corresponding discrete mass. 156

Table 6-3 Distribution of mass between skin and web for the best known solution of the box-sections with fixed web breadth. 156

Table 6-4 Improvement in mass discovered by allowing the breadth of the web of box-sections to vary. 158

Table 6-5 Selection of best solutions found for the box-section design including the number of elements and web breadth as design variables. 159

Table 6-6 Loading conditions considered for the GARTEUR benchmark stiffened panel design with the highest maximum in-plane stress resultants in a Dornier wing. 160

Table 6-7 Best solution found for each case of blade-stiffened panels (I3 – I5) and flange-stiffened panels (J3 – J5) 164

Table 6-8 Critical buckling load factor and governing half-wavelength for flange-stiffened panel J3 for each of the loading condition. 166

Table 6-9 Material properties and price factors considered for the plies and core. 171

Table 6-10 Best solutions found at various trade-off factors. 176

Table 6-11 Material properties between a carbon and glass fibre reinforced epoxy. 182

Table 6-12 Results for mass-cost tradeoff at various levels for a mixed materials CFRP/GFRP optimisation 182

PRINCIPAL NOMENCLATURE

$\mathbf{A}$	Laminate In-Plane Stiffness Matrix
$\mathbf{B}$	Laminate Coupling Stiffness Matrix
$\mathbf{C}$	Material Stiffness Matrix
C_{ij}	Chromosome
$\mathbf{D}$	Displacement Vector
$\mathbf{D}$	Laminate Out-of-Plane Stiffness Matrix
$\mathbf{d}_j$	Displacement Vector at Edges $j = 1,2$
$\mathbf{E}$	GA Environment
E_1	Primary Young's Modulus (parallel to fibre)
E_2	Secondary Young's Modulus (perpendicular to fibre)
F	Load Factor
G	Genotype
G_{12}	In-Plane Shear Modulus
g_{ijk}	Gene
J	Number of Eigenvalues Exceeded at a Trial Load Factor or Frequency
J_{m}	Number of Eigenvalues of Component Plate if Edges are Clamped
J_{S}	Number of Eigenvalues of Substructure if Edges are Clamped
J_0	Number of Eigenvalues if Structure is Edges are Clamped
$\mathbf{K}$	Overall Stiffness Matrix
$\mathbf{k}$	Plate Stiffness Matrix
$\mathbf{k}_{ij}$	Plate Partial Stiffness Matrix
ℓ	Plate/Assembly Length
$\mathbf{m}$	Perturbation Bending and Twisting Moments Vector
m_x, m_y, m_{xy}	Perturbation Bending and Twisting Moments per Unit Length
$\mathbf{n}$	Perturbation Membrane Forces Vector

N_L, N_T, N_S	In-Plane Forces per Unit Length
n_x, n_y, n_{xy}	Perturbation Membrane Forces per Unit Length
$\mathbf{P}$	Force Vector
p_j	Perturbation Force Vector at Edges $j = 1,2$
$\mathbf{Q}$	Reduced Material Stiffness Matrix
$\overline{\mathbf{Q}}$	Transformed Reduced Material Stiffness Matrix
Q_{ij}	Reduced Material Stiffness Matrix Coefficients
$\overline{Q}_{ij}$	Transformed Reduced Material Stiffness Matrix Coefficients
$s[\mathbf{K}]$	Sign Count of the Overall Stiffness Matrix
$\mathbf{T}_{ej}$	Eccentricity Transformation Matrix at Edges $j = 1,2$
$\mathbf{T}_r$	Rotation Transformation Matrix
u, v, w, ψ	Perturbation Displacement Amplitudes
V_f	Fibre Volume Fraction
$\mathbf{X}_{jk}^A$	Alleles
$\mathbf{X}^E$	Genetic Structures
γ_{12}	Plane Shear Strain
$\boldsymbol{\varepsilon}$	Strain Vector/Perturbation Membrane Strains
ε_1, ε_2	Plane Strain
$\boldsymbol{\kappa}$	Perturbation Curvature and Twists Vector
κ_x, κ_y, κ_{xy}	Perturbation Curvature and Twists
λ	Half-wavelength of Response (Buckling or Vibration)
$\boldsymbol{\mu}$	Measurement of Performance
ν_{12}, ν_{21}	Major and Minor Poisson's Ratio
ξ	Schema in GA Environment
$\boldsymbol{\Xi}^E$	Schemata in GA Environment
π	Pi
$\boldsymbol{\sigma}$	Stress Vector
σ_1, σ_2	Plane Stress
$\boldsymbol{\tau}$	Adaptive Plan
τ_{12}	Plane Shear Stress
$\boldsymbol{\Omega}$	Genetic Operators

ω	Frequency
ω_i	Genetic Operator

1 Introduction

1.1 Motivations, Aims and Objectives

To borrow a quote from Richard Dawkins (2003) in *A Devil's Chaplain*, "With evolution, a great light breaks through into the deepest recesses, into every corner, of the science of life... You understand not only what is, but why." Life has shown tremendous fortitude in the most adverse of natural environment, adapting to the most extreme and hostile of conditions in beautiful and weird ways. Nature has an unmistakeable infallibility of evolving, via the process of natural selection, in producing the unique variations that abound everywhere. Yet this beauty in design is only but transient and in constant flux, for deep within, forces that drive evolution are blind and selfish.

With better understanding of such natural systems, so too is the ability to simulate, with increasing accuracy, such robust mechanisms to assist in the discovery of designs and solutions to a multitude of problems. Genetic algorithms (GA) are one such simulation. However, there remain many areas where improvements can be made, just as evolution continues unceasingly in its quest for perfection, only to be inhibited by environmental constraints. Here, practical and modest aims are sought, and can be generalised into two parts:

1. Explore alternate approaches to handling design requirements (constraints) to reduce sensitivities of parameters associated with the techniques and so minimise the approach-specific effects on design space, and
2. Investigate such alternate approaches in computationally expensive design and optimisation of prismatic composite laminated plate assemblies.

1.2 Literature Review

The current work covers inter-disciplinary methods and techniques by using GA as an optimisation tool and search process to discover optimal designs of composite laminates. Extensive coverage of all published literatures is impossible. This section covers, instead, those that cover necessary groundwork of models and theories used here as well as developments that, in a way, influenced the current work. The definitions of GA and details of analytical models used in the design of composite laminates are presented elsewhere.

1.2.1 Genetic Algorithms

GA found its start with Holland (1975) initially as *genetic plan*, although the change to the former term was advocated by doctoral students in Ann Arbor to emphasise the centrality of computation in its implementation. In his preface to the 1992 edition of his monograph, he stressed the change of the use of the term ‘intrinsic parallelism’ to ‘implicit parallelism’ to reflect the implicit parallel processing via schemata. The crux of GA lies in its modelling of the natural process of evolutionary genetics as its search algorithm. The process is that of natural selection, which generally guarantees the survival of the fittest on a statistical rather than on an individual level.

Several other evolutionary-based algorithms, or evolutionary algorithms (EA), also emerged around this timeframe, namely evolution strategies (ES) and evolutionary programming (EP). Nevertheless, GA is still considered as the most widely known type of EA (Bäck, 1994). For a more detailed discussion of components relevant to a generic adaptive plan framework, see Chapter 2.

1.2.1.1 Developments in Schema Evolutionary Theories

Much of the early work concentrated primarily on modelling GA in a simplistic framework involving fixed-length binary digit structures and simple operators, hence the often termed Simple GA (SGA), also known widely as the canonical GA. Holland (1975) used such a model to develop a theoretical framework. The resulting Schema Theorem (Holland, 1975; Goldberg, 1989a) states that under a set of considered genetic operators and reproductive plans, schemata with above-average average fitness receive exponentially increasing number of representations (or trials) in subsequent generations.

Due to the many assumptions and generalisations made by Holland (such as using only the bit representations), the Schema Theorem has been considered only valid in that restricted scenario. Mitchell (1996) pointed out that such derived properties gave limited insight into the workings of GA, which led to the conclusion that perhaps a more useful approach to better understanding and predicting the behaviour of GA would be analogous to that of statistical mechanics in physics, where the traditional aims were to describe laws on a macroscopic level (pressure, temperature, etc.) rather than on the microscopic components (particles, molecules, etc.). This will lead to laws that describes GA behaviour on a statistical level, like 'mean fitness in the population'. Unfortunately, developments of theories for explaining the success of GA have been few and far between, where most concentration lies mainly in developing extensional components.

This disparity in development pace often leads to contradictions, with one prime example in Goldberg's Building Block Hypothesis (Goldberg, 1989a). It states generally that genetic alphabets of low cardinality, or 'building blocks', are important in ensuring successful GA deployment, with binary encoding as a main candidate. Even Goldberg (1990b) was drawn to the increasingly successful empirical usage of floating-point genes although it contradicts any simple analyses. Successes in floating-point representations have been widely reported, such as by Davis (1991), although it was also cautioned that a problem-sensitive approach is probably best, in which domain knowledge can and should be incorporated. Although the Building Block Hypothesis is now generally accepted as an article of faith by those employing GA, much still remains in making it more precise and characterising the types of fitness landscapes on which it is valid (Mitchell et al., 1991). However, work elsewhere showed discrepancies in the hypothesis.

The success of GA is often attributed to its property of 'implicit parallelism', which dictates an exponential allocation of trials based on implicit competitions amongst schemata. Attention should, however, be drawn to work by Bertoni and Dorigo (1993), which showed that Holland's (1975) estimate, where at least the full population size of trials are processed usefully, is correct only in the particular case when the actual population size is proportional to 2^n, where n is the number of genes. In fact, Grefenstette and Baker (1989) previously showed that GA does not solve the k-armed bandit problem[1], in the sense of choosing the hyper-plane with the best mean objective function value within a given hyper-plane partition. Indeed, the accuracy of the Schema Theorem has been brought into question following other works (Vose, 1991; Mühlenbein, 1991).

Following such reports, a later theoretical discussion (Grefenstette, 1991) suggested that the Building Block Hypothesis, built on analogies to the k-armed bandit problem, should probably be re-examined. It was then concluded that it is unlikely that a universal GA useful

[1] Holland (1975) used this as an analogy to GA behaviour, and thus illustrating the workings of implicit parallelism.

across all applications can be developed[1]. On a positive note, analysis showed that desirable results obtained are shown to be less dependent on sampling mechanisms or recombination operators. Perhaps this best explains the flurry of specialised operators and GA variants that seem to display robustness under specific contexts. Nonetheless, it also warns against unnecessary attempts at developing novel genetic operators and GA implementations.

An alternative analysis of schema evolution (Stephens and Waelbroeck, 1999) has also shown that the Building Block Hypothesis is only partially true. With the notion of 'effective' fitness, a remarkably similar conclusion to the original Schema Theorem[2] was made, in that schemata of high effective fitness received exponentially increasing number of trials as a function of evolution time. The similarity, however, ends here. Although it was shown that a building block in the form of a unit-gene schema cannot be destroyed by crossover, there is no preference over short, low-order schemata. On the contrary, typically large schemata may be favourable if schema reconstruction dominates.

1.2.1.2 General Progress

Early uses of GA showed an important factor influencing convergence, and hence, reliability. This phenomenon, known as epistasis, has also been observed in natural systems, and therefore, the term is often used interchangeably with deception. Initial work can be traced to back to Holland (1975), where the inversion operator was also proposed to encourage co-adaptation.

Further investigative work was also performed, such as one by Goldberg (1987), which also led to various proposals (Goldberg, 1989b; Goldberg, 1990a; Goldberg et al., 1991), with some being easier to implement than others. One suggestion assumes prior knowledge of the objective function, so that the resulting genotypes are able to arrive at optimal solutions via 'tight' building blocks (citing binary encoding as examples), i.e., genes that are clustered together. Inversion was also suggested as an alternative, although it had largely fallen out of use. Perhaps the most pervasive solution to the problem of epistasis is the proposal of the

[1] It is important, therefore, to note that specialised GA with specialised operators, shown to work on a limited number of problem classes, should not be indiscriminately applied to other areas without careful analysis.

[2] The original Schema Theorem by Holland (1975) was based on actual fitness.

messy genetic algorithm[1] (mGA) with interesting features unlike standard GA.

Following insights into schema processing, such as those by Grefenstette (1991), work on deceptive problems may well be only partly accurate. This is because much of the work on epistasis was grounded on the Building Block Hypothesis.

The infancy of GA is still very much obvious, as there is still a lack of general guidelines on how best to configure and tune related GA parameters to achieve good response, despite numerous heuristic extensions to GA. Early research into mutation rates (Fogarty, 1989; Hesser and Männer, 1991; Bäck, 1992) showed that non-uniform mutation that is time-dependent may improve convergence. Previously, Holland (1975) concluded, based on the Schema Theorem, that mutation should be seen as nothing more than a 'background' operator. This statement has since been challenged by many others, and it is now generally accepted that mutation has a more important role to play in the overall search than was predicted by Holland.

Improvements can also be found when mutation rates are also encoded into genotypes to allow self-adaptation, as reported by Bäck (1993). Results indicated that mutation rates should ideally be 1 bit to the total number of bits in the genotype for pseudo-Boolean functions with one optimum, as rates decreasing towards this value will only accelerate the search slightly during optimisation. It is also mentioned that Gray code (see Chapter 2) also assures unimodality of the fitness function whenever the objective function itself is unimodal. With multimodal fitness functions – for example when using standard binary encoding – suggestions were to use values other than that suggested for unimodal fitness functions above to overcome local optima. Hence, the suggestion that self-adaptation of mutation rates may be beneficial. However, the current work adopted only the fixed mutation rate throughout all case studies conducted to prevent additional design variables from being introduced into the search.

There is still an abundance of reports on best-practice for GA parameter tuning, such as investigating effects of population size and functionalities of various crossovers (De Jong and Spears, 1990; Spears and De Jong; 1990, 1991). To prevent redundancy, further reviews are

[1] Coverage of mGA is ignored here as it bears little relevance to the current work, since only fixed-length genotypes and standard GA components are modelled in developments here. See cited references for more details.

left where appropriate.

1.2.2 Composite Plates and Panel Assemblies

Designs and optimisations of composite laminates often revolve around three important factors – modelling, analysis and optimisation. Often, the details of the analysis depend largely on the modelling. For simple plates, the focus has been primarily on the well-established classical plate theory (CPT). However, stiffened composite panels often require rather more expensive analytical and complicated modelling, resulting often in the use of finite element method (FEM), with tools such as STAGS (Almroth and Brogan, 1978), FEMSYS (Peters, 1979), EAL (Whetsone, 1980), and NASTRAN, or approximate or exact finite strip methods, such as VIPASA[1] (Wittrick and Williams, 1974), VICON (Anderson et al., 1983), and STIPSY (Baburaj and Kalyanaraman, 1993). Finite strip analysis partitions panels into strips, where displacement fields are described using polynomial and trigonometric functions.

1.2.2.1 Plate Design and Optimisation

The unidirectional properties of fibre-reinforced composites give the freedom to generate bespoke designs. In the case of composite laminates, composed of many layers of thin fabrics or unidirectional layers in standard thicknesses, changes in ply thicknesses and fibre orientations allow tailoring to suit specific properties. Therefore, most practical laminates require integer multiples of a standard ply thickness, which then involves integer optimisations. Combinatorial optimisations are also possible and brings in fibre orientations of individual layers as design variables, usually limited to the common set of 0°, 90°, and $\pm 45^\circ$, to produce custom stacking-sequences with integer multiples of standard ply thicknesses that are most efficient for a variation of loading conditions.

For discrete optimisation, GA has proven to be the more popular optimisation tool. Callahan and Weeks (1992) have used GA to maximise stiffness and strength of a laminate under in-plane and flexural loads. Many such optimisations have been reported by Haftka, Gürdal, and co-workers (LeRiche and Haftka, 1993; Gürdal et al., 1994; Kosigo et al., 1994; Todoroki and Haftka, 1997), using specialised GA as their primary stacking sequence optimisation tool

[1] Chapter 4 covers in detail the theoretical groundwork for the VIPASA model, since it is used as the analytical tool for subsequent design and optimisation.

under buckling and strength constraints.

Sargent et al., (1995) presented a comparison between GA and other random search techniques for strength design of composite laminated plates. Liu et al., (1998) developed a Gene-Rank GA and repair strategies for constraint handling and applied it to the maximisation of buckling load of composite laminated plates. Several specialised GA have also been developed, noticeably those by LeRiche and Gaudin (1998), and Potgeiter and Stander (1998), for ply angles and stacking sequence selection for dimensional stability optimisation subjected to thermal and hydro expansions, and strain energy minimisation subjected to lateral loads respectively. Costa et al., (2000) also approached the combinatorial problem of stiffness maximisation with the option of including cost constraints and employing an analysis based on the first and third order shear deformation theories.

Grosset et al., (2001) considered a multi-objective weight-cost optimisation of a rectangular laminated plate using GA, subjected to constraints on the first natural frequency, by allowing a mixture of graphite-epoxy and glass-epoxy in the resulting plate. Grosset et al., (2002) again investigated the improvement of GA in in-plane stiffness maximisation of a symmetric and balanced laminated plate with additional statistical sampling.

1.2.2.2 PANEL ASSEMBLIES DESIGNS AND OPTIMISATIONS

Most panel designs and optimisations have concentrated on continuous design variables. Early examples can be seen in work by Stroud and Agranoff (1977), performing minimum mass designs on panels using exact finite strip analysis. The analysis was conservative when anisotropy effects were considered. An efficient computer program was introduced by Williams and Wright (1978) for buckling and vibration analyses of prismatic assemblies of longitudinally compressed isotropic plates, with capabilities of handling sub-structures. Attention should, however, be drawn to a more general analysis for plate assemblies by Wittrick and Williams (1974), employing the Wittrick-Williams algorithm, that allows anisotropy and combined loadings, which should prove more useful for composite laminates. Improvements to the analysis was later resolved by Anderson et al., (1983), accounting better for shear and anisotropic effects.

These developments led to the VICONOPT design computer program (Williams et al., 1990a; Williams et al., 1990b). Butler and Williams (1992) went on to report design studies using VICONOPT on stiffened panels with buckling and strength constraints. More practical loading conditions were explored by Butler (1994). Comparisons of VICONOPT analytical capabilities with FEM showed good agreement. Butler (1995) later obtained design charts for minimum weight designs for a range of loads.

A practical design for real structures can be seen in a full-scale parametric study of optimum designs for benchmark stiffened wing panels of a Dornier aircraft by a GARTEUR (Group for Aeronautical Research and Technology in Europe) Action Group using VICONOPT by York et al., (1993). The continuous optimisation performed included design variables ranging from ply thicknesses and stiffener dimensions, to ply angles. It was also reported that allowance for transverse shear deformation made negligible difference as the design was governed by overall buckling and a maximum stiffener height constraint.

Relaxation of the height constraint for variant cases of the heaviest loaded panel of the Dornier aircraft showed that designs obtained using CPT failed to carry the design load by up to 7% when analysed using the first order shear deformation plate theory (SDPT), as reported by Williams et al., (1996). On the other hand, redesign using SDPT incurred a mass penalty of approximately 2%. Mass and cost optimisations have also been performed using VICONOPT, as reported by Edwards et al., (1998) where the common minimum mass objective was replaced with a minimum cost objective or a combination of both.

A report by Kennedy et al., (1999) explored further practical applications into aerospace panels using VICONOPT. A two-stage optimisation process was performed to arrive at discrete optimum designs of blade-stiffened panel assemblies, showing negligible penalties of less than 1% when continuous thickness designs were forced to be discrete. In the same report, discontinuous cost functions were also successfully optimised as did inclusions of fundamental frequency and frequency-free band constraints.

Kennedy et al., (2004) presented a strategy of discrete optimum design with application to discrete mass design of composite plates with longitudinal voids. Bounds on the discrete optimum mass are first established by finding the continuous optimum and by binary partial enumeration in its vicinity. These bounds then permit an efficient branch-and-bound search for the discrete global optimum. Results showed a 50% reduction in the mass penalty associated with satisfying discrete thickness requirements, for plates with and without longitudinal voids.

An alternative optimum design program, PANDA/PANDA2, was reported by Bushnell (1987a; 1987b). It uses a wide range of simple physical models to capture individual failure modes. Reports have also been published that have verified optimum designs from PANDA2 empirically in experiments (Bushnell, 1988), and using FEM (Bushnell, 1994) via the analytical tool STAGS.

Integer representations for ply thicknesses, fibre orientations and stiffener geometry used for a simple GA optimisation of a minimum mass design with buckling, strain and contiguity constraints were reported by Nagendra et al., (1993). A new and improved GA was later presented (Nagendra et al., 1996) that exhibited improved convergence and reliability, but still requires thousands of analyses.

Since analyses of a population of genotypes are expensive in GA, Harrison et al., (1996) explored the use of response surface methodologies (RSM) in an attempt to reduce computation overhead. Exact analytical data obtained from PANDA2 in the early stages of the search were used to construct a polynomial surface to serve as good approximations to failure loads for feasibility evaluations of genotypes in subsequent generations. GA was also used by Crossley and Laananen (1996) as the optimisation tool to maximise energy absorption with ply thicknesses, fibre orientations and stiffener types as design variables.

1.3 General Overview

Optimisation has been a goal, long sought in all aspects, from engineering applications through to natural systems, to find the most efficient systems and configuration of elements to fulfil the demands and requirements of the governing bodies. In nature, the driving force behind adaptation, leading to increasingly complex organisms, stems from the process of evolution since time immemorial.

Engineering designs, however, do not possess the luxury of time as solutions are expected in as little time as possible. As computational power increases, so is the ability to include more design variables in engineering designs and optimisations and the mimicry of more time-consuming natural processes invoked by natural adaptive systems. Genetic algorithm (GA) is one such example of a search procedure derived from analogies to natural systems. Many other similar methods have been successfully developed and deployed, ranging from evolutionary strategies (ES) to ant colony optimisation and swarm intelligence. Chapter 2 provides a general overview of components involved in a typical GA framework. Practically, all systems have to satisfy certain requirements and conditions, known as *constraints*. A novel proposal for efficient handling of constraints in GA is outlined in Chapter 3, which also presents a study into a constrained numerical optimisation problem.

Such design capabilities, sometimes in hybrid systems involving traditional optimisation tools, have increased competitions for ever-increasing efficient designs. In structural applications, this implies a demand for better load carrying capacity without excessive penalties in weight and/or costs by using stronger and lighter materials. The ability to perform design analysis cheaply and rapidly also provides facilities for observing effects induced by changes to materials, geometry, physical dimensions, and other factors.

Chapter 4 covers the analytical concepts necessary to proceed on to the design and optimisation of composite laminated plates and prismatic plate assemblies. Chapter 5 presents the design of plates with longitudinal voids. Chapter 6 extends the constraint handling capabilities of the GA framework to design of a variation of stiffened panel assemblies, including box-sections, blade-stiffened and web-stiffened panels, before finally proceeding to a multi-objective mass-cost optimisation of a blade-stiffened panel involving structural foam core materials. It also includes a variant of the multi-objective optimisation that considers two composite materials considering carbon and glass fibres in an epoxy matrix.

Chapter 7 then provides a closing overview and conclusion of the various proposals and case studies conducted. Future perspectives are also presented as possible developments from this work.

2 GENETIC ALGORITHMS

This chapter provides a general overview of Genetic Algorithms (GA) and the various objects and processes encapsulated by it. A generic framework is first presented followed by a close association to the standard optimisation formulation. Then a more detailed coverage of the various objects and processes is presented, ending with a general discussion of the different implementation models.

2.1 INTRODUCTION

GA is a set of stochastic algorithms derived from the fundamental principles underlying Darwinist evolution and Mendelian genetics – evolutionary genetics, first presented and proposed by Holland (1975). Its processes are not unlike an evolving biological system, using micro-level genetic interactions coupled with macro-level environmental feedback to drive the emergence of organisms that has adapted itself to maximising the exploitation of environmental resources.

GA performs a stochastic search of a domain space of a simulated environment by modelling the processes of natural selection. Natural selection, in biological terms, is the process that stochastically guarantees organisms that are better adapted to a particular environment a higher probability of survival. This then ensures that better genetic blueprints are propagated to organisms in successive generations. Organisms in future generations can then build upon this 'memory' imprinted in the blueprints and improve upon it, hence becoming more adapted to the environment while continuously transmitting improved knowledge of the landscape onwards.

In general, natural selection can be thought of as a continuous process of searching through a space for better solutions, which, in turn, can be viewed as a process of solving a problem – an optimisation process. For problems with a small solution space, an exhaustive search may be feasible but overhead increases dramatically when solution space becomes larger. In lieu of an exhaustive search, a more intelligent approach must be employed. Natural selection, observed in nature, has produced elegant results that are able to exploit the most hostile and varied of environments. GA mimics it hoping to reproduce the same success through an equivalent algorithm.

GA encourages evolutionary changes in structures via low-level simple and basic interactions in their sub-structures. These interactions, pivotal in GA, encourage these structures to exchange meta-information about their relative adaptability and performances, and recombine them to produce potentially more accurate and complex information. Thus, GA is evidently a bottom-up algorithm that, driven by evaluative feedback loops, is capable of self-learning on the sub-structure level without a priori information about the environment. Such a phenomenon of recognisable patterns and behaviours manifesting with simple seemingly inconsequential interactions is known as emergence (Holland, 1998; Johnson, 2001). Since GA model the behaviour of natural selection, they rely heavily on terminology borrowed from evolutionary genetics to emphasise this analogy.

2.2 Framework and Terminology

The definitions of the terminology are set into the descriptive and simplified framework outlined below to introduce the overall workings of a GA. This will be later placed into the context of a general optimisation problem to provide an application-level overview and correspondence to components in a GA. The framework is simplified in the sense that it models commonly perceived objects and processes of a typical GA. It will also illustrate the differences between GA and other conventional optimisation techniques.

There are three major components central to a typical GA framework:

1. An environment, $\mathbf{E}$ on which a GA operates,
2. An adaptive plan, $\boldsymbol{\tau}$ that determines the low-level interactions between sub-structures, and
3. A measurement of performance, $\boldsymbol{\mu}$ from which a GA obtains evaluative feedback on the environment.

The landscape or environment $\mathbf{E}$ can be defined as the domain space in which a GA operates and progressively evolves genetic structures or solutions that are increasingly adapted to the environment and exploitative of it. The full set of structures derivable from $\mathbf{E}$ and bounded by it is commonly known as the gene pool of genetic structures, $\mathbf{X}^{E}$. The genetic structures may be sets of design variables, sequences of instructions, probability distributions, or any other collection of structures that form the basis of the concerned field of study to which a GA is applied.

The genetic structure G of the modelled environment is known as a genotype, organism or individual (amongst other names). It should be noted that transformation of design variables to a genetic representation is often performed, e.g., floating-point values to binary strings, and will be discussed later. Therefore, the term genotype is often only used to imply the result of that transformation, while the true representation or the actual observed characteristics of the genotypes are called phenotypes. And it follows that for any genotype G, the relationship $G \in \mathbf{X}^G$ and $\mathbf{X}^G \subseteq \mathbf{X}^E$ holds. The reason for introducing $\mathbf{X}^G$ is that there exists a group of GA that allows groups of genotypes to evolve simultaneously and independently in different regions of $\mathbf{E}$, or niches. In the more commonly used group of GA without niches, then the above relationship simplifies to $G \in \mathbf{X}^E$ because $\mathbf{X}^G = \mathbf{X}^E$. This latter case will be used throughout, without loss of generality, because it is not erroneous to consider niches as a single environment provided that genotypes in these niches remain independent and do not interact, i.e. the set they represent do not intersect.

A sub-structure of a genotype is called a chromosome, C_{ij} where the subscripts i and j points to the j-th chromosome of the i-th genotype. If all genotypes $G \in \mathbf{X}^E$ have l chromosomes and the j-th chromosome is subjected to $C_j \in \mathbf{X}_j^C$, then the set of the environment becomes the combination of all chromosomes

$$\mathbf{X}^E = \prod_{j=1}^{l} \mathbf{X}_j^C \qquad 2\text{-}1$$

A chromosome can be further decomposed into smaller building blocks, drawn from GA's analogy to evolutionary genetics, and these building blocks are known as genes, g_{ijk} where the definitions of the subscripts i and j remain as before, and k points to the gene's position in the chromosome, also known as the gene's locus. The set of values that any one gene can carry are known as alleles. Thus, if the elements of $\mathbf{X}_j^C$ are to represent chromosomes with m specified genes, where the k-th gene has a set of alleles $\mathbf{X}_{jk}^A$, then $\mathbf{X}_j^C$ is the set of all combination of alleles

$$\mathbf{X}_j^C = \prod_{k=1}^{m} \mathbf{X}_{jk}^A \qquad 2\text{-}2$$

It then follows that

$$\mathbf{X}^{E} = \prod_{j=1}^{l} \left(\prod_{k=1}^{m} \mathbf{X}_{jk}^{A} \right) \qquad 2\text{-}3$$

It should be noted that $\mathbf{X}^{E}$ is potential rather than the actual set a GA will cover. It is the responsibility of the GA to exploit $\mathbf{X}^{E}$ to its full potential in order to find the set of highly adapted genotypes, but without the compromise of evolution time.

Typically, a GA starts by giving all elements in $\mathbf{X}^{E}$ equal probability distribution and readjusts that upon evaluative feedback. The adjustments are performed by the adaptive plan, $\boldsymbol{\tau}$. The plan, $\boldsymbol{\tau}$, typically iterates across small discrete instances of time, t, rather than continuously on a subset of $\mathbf{X}^{E}$, denoted $\mathbf{X}^{E}(t)$. This subset at time t, a generation, is known as a population of genotypes. It is the responsibility of $\boldsymbol{\tau}$ to traverse through the landscape exploiting existing knowledge of the environment while exploring the potential in $\mathbf{X}^{E}$, by making successive selection of $\mathbf{X}^{E}(t+1)$ from $\mathbf{X}^{E}(t)$ using a set of, usually linearly implemented, (genetic) operators, $\boldsymbol{\Omega}$.

From the previous statement, we can see that a GA has two important roles to perform, and immediately defines the two pivotal tasks of genetic operators – exploration and exploitation. This distinction of the tasks does not necessarily imply a discontinuity of roles, where in actual implementation the roles of operators often overlap.

At generation t, the environment $\mathbf{E}$ responds to a genotype by producing a signal. Depending on the field of study, this signal may, possibly after transformation, be a payoff (game theory), utility (economics), error functions (control) or any other performance measure, $\boldsymbol{\mu}$. This performance measure is known in GA as fitness, and the function transforming the signal into a suitable $\boldsymbol{\mu}$ is called the fitness function. The plan $\boldsymbol{\tau}$ then proceeds as such: $\mathbf{X}^{E}(t)$ is transformed by $\boldsymbol{\tau}$ into $\mathbf{X}^{E}(t+1)$ by assigning a probability distribution over $\mathbf{X}^{E}(t)$ and then applying $\boldsymbol{\Omega}$. This proceeds until some termination criterion is satisfied.

It is important to note that the relationship of fitness to allelic values and vice-versa, in most non-trivial landscapes, cannot be easily determined. This is because in such environments, the presence of certain alleles at certain genes may mask and inhibit the phenotypic effects of other alleles at different loci, resulting in different adaptability. This phenomenon is known in GA as epistasis, and is most observable in non-linear systems, where solutions cannot be superposed to produce solutions.

To summarise, a GA framework encapsulates the set of objects $\mathbf{X}^E$, $\boldsymbol{\mu}$, $\boldsymbol{\Omega}$, and $\boldsymbol{\tau}$, where at generation t,

1. $\mathbf{X}^E(t) = \{G_1, G_2, \ldots\} \in \mathbf{X}^E$ is the population,
2. $\boldsymbol{\mu}$ is the set of fitness of genotypes in the population as a result of direct or transformed feedback from the environment,
3. $\boldsymbol{\Omega} = \{\omega_1, \omega_2, \ldots\}$ is the set of genetic operators acting on the population $\overline{\mathbf{X}}^E(t+1)$, which is a transitional population derived from $\mathbf{X}^E(t)$, until eventually $\boldsymbol{\Omega}$ finalises it into $\mathbf{X}^E(t+1)$, and
4. $\boldsymbol{\tau} : \boldsymbol{\mu} \times \mathbf{X}^E(t) \to \boldsymbol{\Omega}$ is a GA which, on the basis of the probability distributions over population of generation t, determines what operator(s) to apply on to $\mathbf{X}^E(t)$ to obtain $\mathbf{X}^E(t+1)$.

A generalized GA can be defined using the above objects. To illustrate the components of the above framework and place it in the context of optimisation, consider the standard formulation of such a problem:

$$\begin{array}{llll}
\textbf{Optimise} & & & \\
 & f(\mathbf{x}), & \mathbf{x} \in \mathbf{X}, & \\
\textbf{such that} & & & \\
 & h_i(\mathbf{x}) = 0, & i = 1,2,\ldots,m, & \\
 & g_i(\mathbf{x}) \le 0, & i = m+1,\ldots,n, & \text{2-4} \\
\textbf{where} & & & \\
 & \mathbf{x}^{LB} \le \mathbf{x} \le \mathbf{x}^{UB} & &
\end{array}$$

A vector of design variables, $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, is analogous to a genotype, where the design variables and the corresponding objective function $f(\mathbf{x})$ in this case are observable characteristics or phenotypes. Therefore, each generation will be composed of a set of solution vectors. A GA is able to operate on $\mathbf{x}$ only usually after some mapping to its genotypic equivalence, and the result of transformation of a single design variable $x \in \mathbf{x}$ is a chromosome. The number of genes in the chromosomes and the set of alleles governing the genes will obviously depend on this transformation. The environment $\mathbf{E}$, defined by the boundaries imposed by $\mathbf{X}$, then responds to a genotype in a particular population by applying the objective function, $f(\mathbf{x})$, the set of equality constraints, $h_i(\mathbf{x})$ and inequality constraints, $g_j(\mathbf{x})$ and assigning a fitness that is representative of the cumulative effect of optimality of $\mathbf{x}$ in $f(\mathbf{x})$ coupled with satisfaction of the constraints.

The plan $\boldsymbol{\tau}$ then assigns a probability distribution over the population and derives stochastically, using operators in $\boldsymbol{\Omega}$, the next population based on currently available knowledge. Upon the availability of the next population, the algorithm is repeated until eventually the optimal set of $\mathbf{x}$ is attained.

The above illustration can also indicate the differences between GA and other conventional optimisation techniques (Goldberg, 1989a), and are summarised as follows:

1. GA works with genotypes and resulting sub-structures and not the solution vectors themselves, i.e., the transformed space of the environment proper to a GA landscape, particularly when encoders and decoders are used to map environment variables to and from genotypes (e.g., binary encoding),
2. GA works on a population of genotypes and not a single solution vector,
3. GA works on performance measurements of genotypes or fitness and not on derivatives, gradients or other auxiliary data, and
4. GA works on stochastic, not deterministic transition rules.

2.3 Genetic Structures, $\mathbf{X}^E$

In the summary of the preceding section, it has been shown that GA works on genotypes rather than solution vectors, and not only on a single instance but a population of them at any generation t. Genotypes are elements from the set of genetic structures $\mathbf{X}^E$, which is the transformed domain of the environment, $\mathbf{E}$. Hence

$$\begin{aligned} &T:\mathbf{E}\rightarrow\mathbf{X}^E \\ &T^{-1}:\mathbf{X}^E\rightarrow\mathbf{E} \end{aligned} \qquad 2\text{-}5$$

where the encoder, T, transforms elements in $\mathbf{E}$ to the corresponding genetic domain, $\mathbf{X}^E$, while the decoder, T^{-1} reverses the transformation of T.

2.3.1 Binary Encoding

The most commonly applied transformation traditionally, especially to decimal numbers, is the binary or base-two encoding. This simply transforms decimal numbers to a sequence of bits (or binary digits), thereby producing a binary vector or string. Hence, genotypes are also commonly referred to as simply strings. For example, the decimal value 25 is represented by a binary string[1] 11001. It is immediately obvious that the chromosome has a length of 5 and at every locus, the alleles are of the set $\mathbf{X}_{jk}^{A} = \{0,1\}$.

Typically, a chromosome's length is defined by the bounds of the variable it represents. The binary encoder-decoder also needs to accommodate negative values and floating-point numbers for any variable $x \in \mathbf{x}$. The genotype is then a vector of chromosomes[2]. For example, consider the Rosenbrock function, which has been commonly used to test the robustness of GA and is presented below[3]:

Minimise

$$f(\mathbf{x}) = 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_1\right)^2 \qquad \text{2-6}$$

where

$$-2.047 \le x_i \le 2.048$$

The problem consists of two design variables, where x_i can be represented as a 12-bit binary string and thus defines the chromosome that represents x_i. The genotype is simply the vector of two chromosomes of identical lengths to form a 24-bit binary string where a random sample from the domain space might produce (in matrix representation)

$$G = [111010101001 \quad 100101111000]$$

[1] This presentation format is known as *big-endian*, where the bits have decreasing significance from left to right, as opposed to *little-endian* where the order of significance is reversed. Do note that this is the presentation format typically adopted by GA literature and the convention is observed here.

[2] The term 'chromosome' is also used in literature to imply the whole genotype. In such cases, the distinction between genotype and chromosome is unclear. The convention used here is aimed at preventing this ambiguity, so that a one-to-one correlation between a design variable to a chromosome, and a solution vector (of design variables) to a genotype, may be implied directly without the need for further disambiguation throughout.

[3] The usual lower bound is -2.048, but is here taken to be -2.047 because then it divides exactly into a 12-bit binary string with 4096 discrete states, giving a 0.001 difference between each state.

giving phenotypic values

$$\begin{aligned} x_1 &= 1.706 \\ x_2 &= 0.377 \\ f(\mathbf{x}) &= 642.3282 \end{aligned}$$

Generally, for any solution vector $\mathbf{x}$ with n elements, a genotype can be represented as a vector of n chromosomes or as a $1 \times n$ matrix. Therefore, all the genotypes of a population of size m can be represented by a $m \times n$ matrix.

With the binary encoding, moving a discrete step from an arbitrary value of x to x' often involves changing of more than one bit in the binary string. For example, consider a move from $x = 7$ to $x' = 8$ in a 4-bit string. This will require 4 bit-flips in the corresponding chromosome from $C = [0111]$ to $C' = [1000]$. The implication of this is that obtaining a small shift of value in C may require considerably more effort on genetic operators than a large shift. On top of that, the set of random bit flips of any arbitrary gene leading to considerable phenotypic changes is large particularly in long binary strings. As a result, many other forms of encoding, especially for real numbers, have been used to reduce this discrepancy.

Gray encoding, or the mapping of domain variables into Gray codes, is a common encoding that is used to overcome the disadvantages of binary encoding. The term 'Gray code' often refers to the binary reflected Gray code.

A Gray code[1] equally has bits as alleles when used, but the difference lies in the fact that it represents the numbers in the sequence of integers $\{0,\ldots,2^{m-1}\}$ as a bit-string of length m where adjacent numbers differ only by a single bit. Thus, genetic operators need only operate on one bit to obtain the next phenotypic value.

This 'adjacency' property of Gray codes makes it more attractive when particular genetic operators are considered, and has been reported as performing slightly better than binary encoding. However, while the largest phenotypic change in value due to a random bit flip on a gene of a binary-encoded chromosome of length m is 2^{m-1}, the change in a Gray-encoded chromosome of an equal length is larger. Nevertheless, the set of random bit flips on any

[1] See Appendix I for conversion rules from Gray to binary, and vice-versa.

arbitrary gene of a Gray-encoded string leading to considerable phenotypic changes is smaller than its binary-encoded equivalent.

2.3.2 Direct Encoding

There is a growing preference, especially in the research of applications of GA, to directly apply GA to real parameters representative of the environment. For example, the floating-point representation of genetic structures is often used for numerical optimization.

The use of direct encoding allows specialisations to specific problems by defining genetic operators that exploit any a priori knowledge of the domain. Although GA are intended to be a general plan for exploiting and exploring any 'uncharted' domain space, their robustness particularly in environments fraught with noises and local optima, which traditional search and optimisation methods have difficulties in traversing, prove attractive. Coupled with specialist 'knowledge', even more robust hybrid methodologies can be developed for practical applications.

For example, consider again the Rosenbrock function (Eq. 2-6). With floating-point encoding, the actual values of the design variables x_1 and x_2 can be represented directly, where a random sampling across the domain might yield a genotype

$$G = [1.706 \quad 0.377].$$

Direct encoding does not imply only using floating-point values, but other forms of structural representations that are immediately indicative of the environment or problem domain. These are like, but not restricted to, tree structures, base-n values and subsets of the domain space (niches).

There is strong advocacy against using floating-point values in numeric optimisations by GA 'purists' (Goldberg, 1990b). The use of such direct encoding, particularly when decomposable into smaller packets of information (like bits), tends to disagree with the Building Block Hypothesis. The hypothesis, proposed by Goldberg (1989a), states that complex adaptations evolve from simpler structures not unlike 'a child [creating] magnificent fortresses through the arrangement of simple blocks of wood [...]'.

Although reports have been made that verify the hypothesis, others have also proven (empirically) that using floating-point values give significant advantages. The consequence is of course the development of new genetic operators that must deal with the inability of traditional operators to cope with non string-like representations. The edge recombination operator for the Travelling Salesman Problem (TSP) (Whitley et al., 1991) is an example of this.

2.4 Measurements of Performance, $\boldsymbol{\mu}$

GA, as mentioned earlier, are bottom-up algorithms driven by evaluative feedback. While GA do not presuppose any derivative, gradient or other auxiliary data, a measurement of performance, $\boldsymbol{\mu}$, is required in order to gauge the adaptability of genotypes in exploiting environmental resources.

For each genotype $G \in \mathbf{X}^E$, a measure of performance, $\boldsymbol{\mu}$ determines the probability of survival of new structures derived partially or fully from it. In other words, the fitness of a genotype must be an indication of its optimality in $\mathbf{E}$. Thus, for any genotype in $\mathbf{E}$, there is a set of corresponding interpreters, $\mathbf{I}$ which decodes a set of environmental response, $\mathbf{R}^E$ into a corresponding fitness value.

$$\mathbf{I} : \mathbf{R}^E \to \boldsymbol{\mu} \qquad 2\text{-}7$$

Traditionally, particularly in applications into optimisations, $\mathbf{I}$ is a monotonic fitness function that scales the response, $\mathbf{R}^E$ obtained directly from the formulation of the optimisation problem. This response is normally a value that is representative of the cumulative effect of the objective function and any constraints. Methods for deriving suitable fitness functions for such optimisation problems are varied and are discussed in Chapter 3.

2.4.1 Evaluative Feedback and the Roles of $\boldsymbol{\mu}$

In most non-trivial applications, the set of possible genotypes imposed by an environment $\mathbf{E}$ is very large, and combined with the effects of $\mathbf{I}$, there may exist a large group of genotypes that show optimality, local or global. It should be noted that measurement of performance, $\boldsymbol{\mu}$ is not a concept that is only of consequence in GA. It is the approach towards it in GA that is different.

In methods like random-search, exploration is maximal with no exploitation of any knowledge arising from that exploration. On the contrary, hill-climbing methodologies are highly exploitative of knowledge but have little (random-restart hill-climbing) or no exploration. This leaves GA with a challenge: Optimise the balance between exploration of the domain space and exploitation of any evaluative feedback from environment **E** in the form of $\boldsymbol{\mu}$. Thus, GA must persistently identify and compare properties of genotypes and incorporate high-performing properties into new genotypes.

2.4.1.1 GENETIC CO-ADAPTATION

In GA, as in evolutionary systems[1], high-performing genotypes are generated from smaller components or properties. This then redefines the process of GA as one that searches for structures where alleles at various loci combine to improve the performance of their corresponding phenotypes. This phenomenon of different alleles of different functions merging together to augment phenotypic effects and improve survivability is called co-adaptation.

The structure arising from the co-adaptation of alleles is called a schema, $\xi \in \Xi^E$, where Ξ^E is the set of schemata representing the complete set of genetic structures $\mathbf{X}^E$. In a nutshell, a schema is a set of genotypes with common allelic attributes. To achieve that, a 'don't care' $*$ meta-character is often used to allow any alleles to match it.

Consider again Rosebrock's function (Eq. 2-6), and let the variables x_1 and x_2 be binary-encoded (or Gray-encoded). A random sampling across the domain space might yield genotypes

$$\begin{aligned} G_1 &= [100101011110 \quad 011010111001] \\ G_2 &= [000101111100 \quad 101010011110] \\ G_3 &= [111010111001 \quad 010010100111] \end{aligned}$$

Consider also two schemata

[1] A typical example will be the physical and functional combination of replicators into primitive cell-like structures, and henceforth into more complex structures like viruses, bacteria and multi-celled organism.

$$\begin{aligned}\xi_1 &= [1{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}{*} \quad 0{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}] \\ \xi_2 &= [0{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}{*} \quad 1{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}{*}]\end{aligned}$$

A genotype matches a schema if the value of its genes matches at all positions where the values are fixed, i.e., not denoted as $*$. In other words, a schema can also be viewed as a hyper-plane across the domain or a subset of $\mathbf{X}^{E}$. In the above example, G_1 and G_3 match ξ_1, while G_2 matches ξ_2. It is also obvious that a schema matches $\prod_r N_r^A$ genotypes, where N_r^A is the size of the set of alleles at position r, and r are positions where values are $*$. Also, a genotype with m genes matches in total 2^m schemata, regardless of the cardinality of the allelic set. Hence, in a GA where the number of genes are fixed at m, the number of schemata processed by a population of size p at generation, t is then between 2^m and $p \times 2^m$.

2.4.1.2 Schema Growth

Schemata, as a basis for associating attributes to high-performance, must therefore be able to sustain the rigours of genetic operators. On top of that, increasingly complex schemata, i.e., has fewer $*$, must be progressively discovered. Hence, a comparison of the performance of schemata processed by a population must be made.

The fitness of a schema ξ at generation t, $\mu(\xi,t)$ is usually taken to be the average fitness of p number of genotypes in the population at t that matches it. If the fitness of the i-th genotype in the set of genotypes matched by ξ is $\mu(G_i,t)$, then

$$\mu(\xi,t) = \frac{\sum_{i=1}^{p} \mu(G_i,t)}{p} \qquad 2\text{-}8$$

If GA are to exploit this inherent knowledge in the form of schemata in genotypes, then above-average performing schemata must receive an increasing number of genotypes, below-average performing schemata must receive decreasing number of genotypes, and an average performing schema must receive equal number of genotypes at the next generation. This behaviour is proven to exist in GA by the Schema Theorem and modelled mathematically as the Schema Growth Equation. A general formulation[1] of the Schema Growth Equation depends largely on two factors that are dependent on the type of GA employed:

1. The probability that a schema is sampled, and
2. The probability of survival[2] of a schema upon the action of a genetic operator.

This behaviour of a schema where the number of instances (in the form of genotypes) to be sampled is proportional to its fitness and unaffected by other schemata[3] is called implicit parallelism. It also results in an intrinsic process where schemata are ranked relatively to each other in terms of observed performance, and are continuously readjusted as new schemata emerge and instances of these schemata are evaluated. Therefore, GA rely on evaluative feedback and a measure of performance to progressively derive higher order schemata (increasing accurate representation of optimality) from low-order ones.

2.4.2 Fitness Evaluations and Probability Distributions

As mentioned previously, the fitness value of an individual, and hence the schemata it matches, is critical to the workings of GA. Without a proper measurement of performance, it cannot exploit useful and discard useless pieces of information. Hence, a good fitness function[4] is pivotal to the success because it serves as the only link between the underlying

[1] It should be noted that the Schema Theorem and its mathematical proof, the Schema Growth Equation, presented by Holland (1975) are specific to binary encoding, stochastic sampling mechanism and the one-point crossover. See Chapter 1 for recent developments in the Schema Theorem, and why it should be viewed cautiously.

[2] When formulating for a string-like representation, two attributes of a schema are commonly defined – *order* and *defining length*. The order is simply the number of fixed-value loci, i.e., those not with *, whilst defining length is the distance between the first and last fixed-value loci. The defining length is particularly important when formulating the probability of surviving a genetic operator that transforms a segment of genes. On the other hand, the order is useful when the probability is formulated for an operator that modifies a single gene in isolation.

[3] The reason it is unaffected is because many schemata can co-exist in a single genotype and concurrently across different genotypes, and can be processed without incurring additional overhead. On top of that, there are potentially a larger number of schemata sampled and processed than there are genotypes. Hence, the term 'parallel' used by Holland.

[4] There are a few misconceptions on GAs relating to fitness functions: (1) Fitness values must be positive. Depending on the sampling mechanism and scaling procedures (to be discussed below), the fitness values returned does not necessarily have to be positive. This misconception often arises because of illustrative examples in literature when placed in the context of a specific sampling mechanism, and (2) fitness functions must be modelled with positive gradients. Again, this is dependent on the sampling mechanism and scaling procedures as sampling may be readjusted to sample genotypes with lower fitness more than those with higher fitness. This is because of the generalisation that higher fitness equates to better performance.

low-order processes of GA systems and the environment.

In its simplest form, the function is taken simply as the objective function of the optimisation problem on which GA is employed. However, in applications to constrained optimisation problems, many methods have been suggested and discussions are presented later.

Once fitness values are obtained, the likelihood that a genotype contains good schemata may be represented by the probability of selection, $\rho_S \in \boldsymbol{\rho}(t)$. Methods of assigning ρ_S are usually derived from the field of statistics. The most commonly adopted form of calculating ρ_S requires fitness values to be positive. Given the fitness of the i-th genotype, G_i in population of size p as $\mu(G_i,t)$, its probability of selection is

$$\rho_S(G_i,t) = \frac{\mu(G_i,t)}{\mu(\mathbf{X}^E(t))} \qquad \text{2-9}$$

where $\mu(\overline{\mathbf{X}}^E(t)) = \sum_{i=1}^{p} \mu(G_i,t)$ is the total fitness of the population and $\mu(G_i,t) >= 0$.

2.4.3 Scaling of Fitness

The process of fitness scaling is often used in GA because of several factors:

1. *Higher fitness is better performance.* Based on this assumption, most GA are modelled to work on this basis. This implies that cases where minimisation problems are involved, resulting fitness values (taken almost always from objective functions) have to be inverted.
2. *Sampling works only on positive values.* As evident in Eq. 2-9, sampling methods that depend on this definition of probability of selection almost always need fitness values to be scaled to prevent negative values from occurring.
3. *Exploration versus exploitation must be optimised.* At any generation t, there may exist genotypes with such a high fitness value, $\mu(G_i,t) >> \bar{\mu}(\mathbf{X}^E(t))$, where $\bar{\mu}(\mathbf{X}^E(t))$ is the average population fitness, that they completely mask the performance of other genotypes. These super individuals will dominate the probability of selection thus leading to rapid convergence, possibly prematurely to local optima. Scaling of fitness, like sigma truncation, can be used to normalise this monopoly of high-fit genotypes when population standard deviation is particularly

high.

The factors stated above affect almost every GA, and although scaling of fitness is a method commonly used, it is also possible to modify sampling mechanisms and genetic operators to accommodate these factors. A summary of commonly used scaling functions is provided below, where the scaled fitness value, also called the augmented fitness, is denoted $\mu'(G_i,t)$:

1. *Linear Scaling.* This method is particularly useful in minimisation problems for handling negative fitness values. It simply involves applying an additional linear mapping to the fitness value. For minimisation problems, it is obvious that $a < 0$, and to prevent negative values from occurring, b must be large enough such that the smallest fitness value will at least map to a non-negative value.

$$\mu'(G_i,t) = a \cdot \mu(G_i,t) + b \qquad \text{2-10}$$

2. *Sigma Truncation.* This method involves mapping the fitness value to the scaled fitness using problem dependent information. The average population fitness is denoted by $\mu(\mathbf{X}^E(t))$, and the population fitness standard deviation by $\sigma(\mathbf{X}^E(t))$. Any negative scaled fitness values arising from the transformation are usually set to zero. The multiplier c is usually a value between 1 and 5.

$$\mu'(G_i,t) = \mu(G_i,t) + \bar{\mu}(X^E(t)) - \left(c \cdot \sigma(X^E(t))\right) \qquad \text{2-11}$$

3. *Power Law Scaling.* This method simply requires taking the fitness to some power of k.

$$\mu'(G_i,t) = \mu(G_i,t)^k \qquad \text{2-12}$$

It should be noted that most of the time, scaling of fitness is problem dependent and hence the above methods are merely generalised guidelines. It is also possible to dynamically update the scaling procedure or its parameters to reflect the general state of the system. With dynamic scaling parameters, the algorithm can be guided between an explorative or exploitative nature. It should be noted that scaling of fitness should not be done arbitrarily, but rather, as mentioned by Hancock (1994), to readjust direct values to more sensible ranges.

2.4.4 Constraint Handling

Most optimisation problems involve constraints, thus requiring fitness values to reflect the satisfaction of the constraints and the optimality of the evaluated genotypes. Generally, the approaches to handling constraints in GA can be classified into three distinct types – extinctive, penalty and repair methods.

The extinctive method, also known as the death penalty, is the simplest to implement and involves automatic termination of genotypes that do not satisfy all constraints. In other words, once a genotype does not satisfy all constraints, it is assigned the lowest possible fitness value making it impossible to further contribute to the search. In GA however, the contribution of weaker genotypes are equally as important as fitter genotypes because of the potentially good schemata that may simply be inhibited due to epistasis, and this leads to probably the sole criticism of this method. Then again, such an approach to constraint handling can be viewed as one that searches only from the safe regions.

A more popular approach, particularly to numerical optimisation problems, involves the use of the classic penalty function. The process usually entails transforming the objective function and constraints into a single monotonic fitness function. However, the more conventional approach is to provide two separate fitness functions, where one is used for fitness evaluation if a genotype satisfies all constraints and another if it does not,

$$f'(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \text{ satisfies,} \\ f(\mathbf{x}) + p(\mathbf{x}) & \text{otherwise.} \end{cases} \qquad 2\text{-}13$$

The penalty function, $p(\mathbf{x})$ assigns a reduction in fitness if any constraints are violated. The obvious shortcoming of this approach is the potential difficulty in deciding an appropriate $p(\mathbf{x})$. Therefore, there exist different generic methods for assigning $p(\mathbf{x})$ and this will be discussed further in the next chapter, when a direct approach to constraint handling is proposed.

The repair methods are even more problem-dependent as they involve 'repairing' genotypes until they satisfy the constraints. These methods are particularly useful but should only be considered in cases when constraint evaluations are computationally cheap. Otherwise, the overhead can be better utilised for additional fitness evaluation, as in the case of composite laminates design and optimisation presented later. Repair algorithms are usually implemented as genetic operators.

2.5 Genetic Operators, $\mathbf{\Omega}$

In the summary of the framework, a set of operators, $\mathbf{\Omega}$ based on the probability distribution $\mathbf{p}(t)$ performs low-level interactions. These interactions are responsible for preserving high-performing schemata and progressively explore and exploit knowledge inherent in these schemata. This behaviour is implicit and requires no additional processing overhead, and therefore, genetic operators are required to transfer information inherent in schemata to the next set of genotypes without explicit knowledge of it.

Each operator $\omega \in \mathbf{\Omega}$ operates on a transitional population, $\overline{\mathbf{X}}^{E}(t+1)$ and sequentially processes it until eventually the next population, $\mathbf{X}^{E}(t+1)$ is formed.

$$\mathbf{\Omega} : \mathbf{X}^{E}(t) \rightarrow \mathbf{X}^{E}(t+1) \qquad \text{2-14}$$

This then implies that the operators in $\mathbf{\Omega}$ are often, but not exclusively, implemented in a linear sequence. Following also from it, three important steps can be summarised as follows:

1. *Replicate*. This involves sampling and replicating genotypes from $\mathbf{X}^{E}(t)$ for genetic information processing to form $\overline{\mathbf{X}}^{E}(t+1)$.
2. *Interact*. At this stage, the candidates derived in step 1 undergo a series of information exchange and renewal or schema processing, hence constantly updating $\overline{\mathbf{X}}^{E}(t+1)$.
3. *Replace*. A new population $\mathbf{X}^{E}(t+1)$ is generated from $\overline{\mathbf{X}}^{E}(t+1)$ and $\mathbf{X}^{E}(t)$ based on some pre-defined strategy.

2.5.1 Replication

Genetic operators responsible for sampling a population to be further processed can be viewed as nothing more than replication operators or replicators. This process of replication is known as the selection mechanism. The fundamental role of replicators is to reproduce verbatim[1] the genetic blueprint of genotypes.

In order for the Schema Theorem to hold true, the selection mechanism must replicate genotypes based on their performance and hence on the probability of selection, ρ_S. However, replication does not introduce new information into the system. It simply reinforces good schemata. The amplification, at which schemata are reinforced, is known as the selection pressure. By indiscriminately reinforcing good schemata, particularly during the early stages of evolution, genetic diversity may be lost, the result of which may be premature convergence to a local optimum. This is particularly true in cases when there is noticeable epistasis. On the other hand, insufficient selection pressure may degenerate the system into one that is comparable to a random walk. Current research is still far from reaching a universal sampling mechanism that can be effectively applied to all problem domains.

2.5.1.1 Stochastic Methods

The roulette-wheel selection (Goldberg, 1989a) (RWS) is probably the most well-known of stochastic sampling methods employed in GA. The strategy proceeds by first constructing a biased roulette wheel where the number of slots equals the population size under consideration, p. The sizes of the slots are proportional to the probability of selection, ρ_S and each spin of the wheel constitutes the replication of a single genotype. The genotype replicated is based on the sector the marker lands. Consequently, genotypes with higher ρ_S will have a greater likelihood that the marker will end on the sector that represents it. Figure 2-1 provides a visualisation of RWS.

However, there are a few apparent drawbacks with RWS.

1. *Effects of Super Individuals.* RWS is unable to deal effectively with the overpowering effects of super individuals. Without some form of control, any super individuals present will most likely dominate next generation's population because of their extremely high probability of selection relative to those in the same

[1] This is not true in biological systems. Replicators are not perfect copying machines and therefore, mutations can occur during replication. These imperfections resulted in the first variations that then sparked the process of natural selection (Sterelny, 2001). This behaviour of imperfect replication is explicitly modelled in another evolutionary-based algorithm, Evolution Strategies (ES).

population leading to rapid convergence before the domain space is sufficiently explored.

2. *Stochastic Errors in Sampling.* Inevitably, RWS will introduce sampling errors or selection error and hence produce replicas of genotypes that do not correspond to their probability of selection. Many corrections have been proposed and hence, RWS is no longer the most successful of selection mechanisms but is still widely quoted, like here, for illustrative purposes.
3. *Fitness Landscape.* RWS does not perform well when fitness landscapes are particularly flat. This is because the distribution of probability of selection does not impose sufficient selection pressure to exploit potentially good schemata. The use of scaling functions, in particular the Power Law Scaling or Ranking procedures, are therefore often applied in such cases.

Many other selection mechanisms have been proposed, though they are mainly centred on the principles governing RWS. The remainder stochastic selection with replacement (RSS-R) uses an expected count parameter to gauge exactly the number of copies a genotype is sampled. This value is obtained by

$$count(G_i,t)=\frac{\mu(G_i,t)}{\bar{\mu}(\mathbf{X}^E(t))} \qquad 2\text{-}15$$

Stochastic errors in sampling or selection errors arise due to the rounding up or down of this parameter for discrete copies of genotypes. To counter this effect, RSS-R first assigns exactly the number of copies due to the integer component of the expected count. The genotypes then compete for any remaining places in the population based on their fractional components of their corresponding expected count. Any suitable sampling mechanism can be used for the competition of remaining places in the population. The effects of super individuals and flat fitness landscapes still remain as problems in RSS-R.

The stochastic universal selection (Baker, 1987) (SUS) is similar to RWS. The algorithm also first constructs a biased roulette wheel where the number of slots is equal to the size of the population being sampled. The size of the slots is proportional to the probability of selection of the corresponding genotypes.

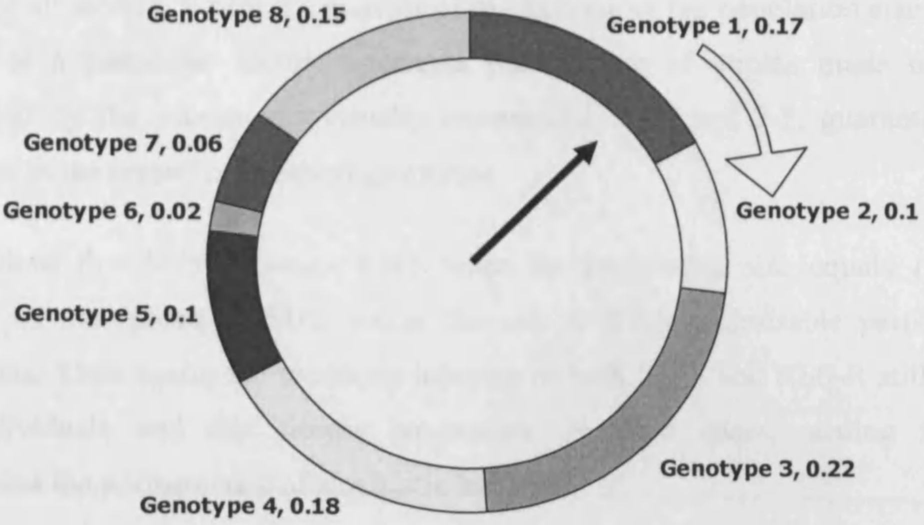

Figure 2-1 Visualisation of RWS with 8 genotypes in the population with corresponding probability of selection.

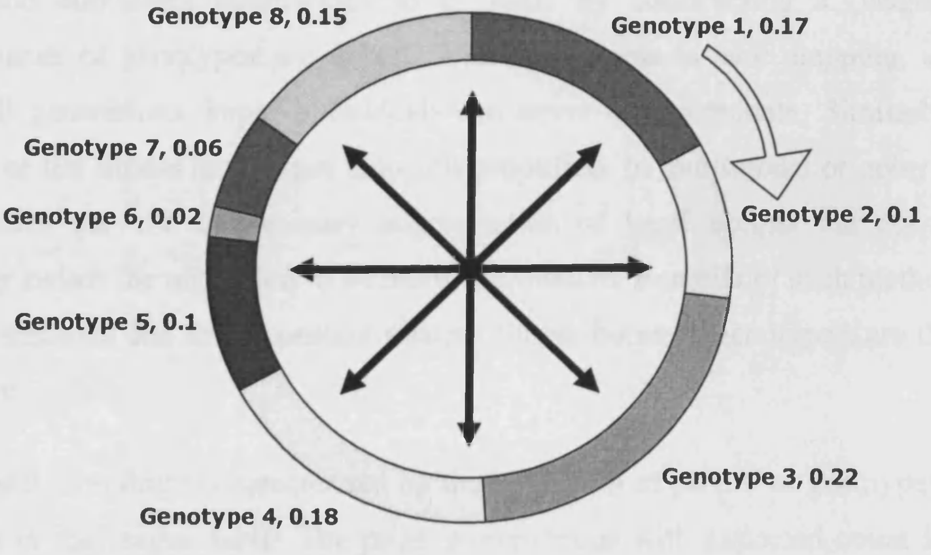

Figure 2-2 Visualisation of SUS, which is similar to RWS, except with the addition of equally spaced markers. In this case, 2 markers pointing at genotype 4 implies that 2 copies of it is made.

Unlike RWS that requires the number of spins equal to the population size, SUS proceeds with only a single spin. From the position at the end of this spin, identically spaced markers are placed all around, where the number of markers equal the population size. The number of markers at a particular sector represents the number of copies made of the genotype represented by the section, as visually represented in Figure 2-2, guaranteeing minimum deviations in the spread of sampled genotypes.

It is obvious that SUS becomes RWS when the population size equals 1. However, the accuracy of the spread in SUS makes the use of RWS undesirable particularly in large populations. Then again, the problems inherent in both RWS and RSS-R still exist in SUS – super-individuals and flat fitness landscapes. In such cases, scaling functions often complement the performance of stochastic methods.

2.5.1.2 STATIC AND OTHER METHODS

Stochastic methods that deal directly with raw (without scaling) fitness have particular vulnerabilities outlined in the previous section. In combating such effects, sampling mechanisms that do not directly include fitness factors have distinct advantages.

One such different approach is the ranking selection (RS). RS simply discards the fitness values and eliminates sensitivities to its scale by constructing a 'league table' where performances of genotypes are sorted. With this fitness-to-rank mapping, usually constant across all generations, super individuals can never over-dominate. Similarly, the selection pressure of flat fitness landscapes is locally amplified. In multimodal or noisy landscapes, the consequence may be unnecessary augmentation of local optima but conversely, it may implicitly switch the algorithm to be more exploitative. Benefits of such methods, also known as static methods due to its constant relative fitness between genotypes, are therefore, highly subjective.

Rank-based sampling is characterised by the allocation of prizes[1] to genotypes based on their positions in the league table. The prize, synonymous with expected count in RSS-R, is of course the number of copies of itself a genotype can replicate, which is usually allocated linearly or non-linearly. For a population of size p and ranking the genotypes in decreasing

[1] The use of the term *prize* here is to prevent the confusion arising from the reuse of the term 'fitness'.

fitness whereby the best genotype is situated at the top of the table at position 0 and the worst at position $p-1$, the prize payoff, $prize(G_i,t)$ of the i-th genotype in the table can then be allocated linearly as

$$prize(G_i,t) = k - \left[(k-1)\left(\frac{2i}{p-1}\right)\right] \qquad \text{2-16}$$

where the parameter k is an indication of the selection pressure and is $1 \le k \le 2$. The lower bound imposes no selection pressure by assigning each genotype a single copy. The upper bound, however, arises naturally due to the natural constraint that no genotype can receive a negative number of copies and maintaining the same population size after replication. A non-linear payoff can be a geometric progression

$$prize(G_i,t) = kr^i \qquad \text{2-17}$$

where $k > 1$, and the corresponding geometric series, where r can be suitably derived based on relationships between the total prize and the population size.

Another different approach to selection is the tournament selection (TS), where the idea of ranking is incorporated. It involves selecting and competing k number of random genotypes in a 'mini league table' scenario. The best genotype wins the tournament and contributes a copy, and the process is repeated until the required population size is reached. The value of k, also known as the tournament size, determines the selection pressure and is often taken as 2, or a binary tournament.

Although TS solves the super individual and flat fitness landscape problems, it reintroduces the selection error. Ways around this may be to use SUS to pre-select the genotypes of the tournaments. For a k-tournament with population size p, the total number of equally spaced markers will be $k \times p$. The genotypes of each tournament are spaced equally between each other, giving maximum genetic diversity with those of the next tournament in the next marker.

2.5.2 INTERACTION

There are generally two types of interaction operators – sexual and asexual. Sexual operators[1] require the participation of at least two genotypes and typically involve some form of genetic exchange to produce new genotypes or offspring, like the reproductive processes of natural adaptive systems. Sexual operators are necessary to exploit existing knowledge in the system. Educated guesses at the trajectory of the search come from exploiting the knowledge in the schemata. Thus, sexual operators are primarily responsible for this exchange of meta-information between genotypes.

With only stochastic replication and sexual operations, a finite population will only be able to explore regions that are already inherent in the schemata it represents. In addition, a finite population through this process will inevitably lose potentially good schemata. Therefore, there are genetic operators that will reinstate lost information and inject genetic diversity into the population. This process is usually provided by the mutation operator and belongs to a wider class of operators known as asexual operators[2] and they operate on genotypes independently.

2.5.2.1 CROSSOVERS AND SEXUAL OPERATORS

Sexual operators have been traditionally the main difference between GA and other evolutionary-based algorithms. The underlying belief, proven in the Schema Theorem, is that the trajectory of the search can be obtained successfully from the embedded knowledge in previous explored regions in the form of genotypes. This exchange of information between genotypes is implemented in a process called crossing-over, where its genetic operator is the crossover operator.

The canonical crossover is the simple crossover (Holland, 1975; Goldberg, 1989a), which is more commonly known as the one point crossover (1PC) and is often attributed only to GA using binary-Gray or string-like encoding. The process involves randomly selecting a common locus between two genotypes (mates, parents, etc.) and swapping the segment

[1] When the process involves only two genotypes, it is also known as a *binary operator* for obvious reasons. However, the more general term – sexual operator – is used here to indicate more than one genotype involved in the process.

[2] The asexual operator is also known as a *unary operator*, but the former is used for consistency.

demarcated by this locus and the end (or start) of the two genotypes to produce two offspring, as illustrated in Figure 2-3. Often, not every sampled genotype will undergo crossover and the probability that they will is known as the probability of crossover, p_C.

An extension of 1PC is the multi-point crossover (MPC). The difference in the process is to randomly selection m loci and alternatively swapping the segments between these points. The introduction of m crossover points is to combat one particular shortcoming of 1PC, whereby it cannot effectively combine two high-performing schemata in a single iteration. This drawback of 1PC, and hence its use as justification for the use of other crossovers should be viewed with scepticism because it arises from three assumptions:

1. A high-performing schema is discontinuous (fixed alleles with $*$ interspersed between),
2. Two high-performing schemata will only exist in a single generation, and
3. The combination of high-performing schemata will result in an even higher-performing schema, i.e., ignoring epistatic effects.

Discontinuous schemata will inevitably arise during the course of the search, and therefore the dependency on such schemata recombining to produce better ones is potentially inadequate and flawed. On top of that, it is possible to argue that the emergence of such higher-performing schemata particularly in the early stages of the search is detrimental as it may introduce the effect of super individuals. The performance of any schema is only as good as the performance history to which it can be compared and evaluated. Therefore, the efficiency of 1PC and indeed most other crossovers should not be measured against their ability to recombine high-performing schemata to produce better ones, but against their inherent ability to generate high-performing high-order continuous schemata in the presence of epistasis and discontinuous schemata.

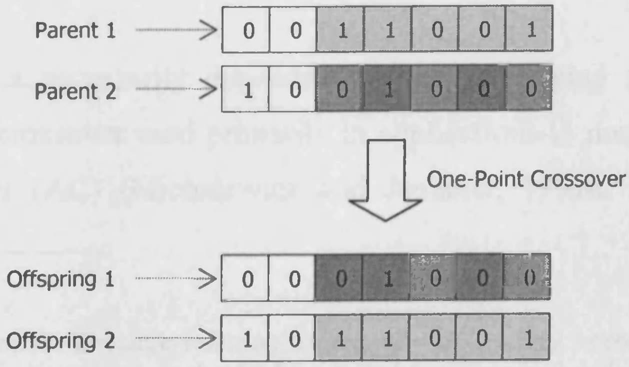

Figure 2-3 1PC on two parent genotypes, resulting in two offspring.

The preceding factors have resulted in a myriad of crossovers. One particular crossover intended to overcome the discontinuous schemata effect is the α-uniform crossover or simply uniform crossover (Syswerda, 1989) (UC). The operator involves progressively moving from one locus to the next, exchanging the genes at each locus based on a pre-defined probability that genes will swap. The most commonly used is the 0.5-uniform crossover, which implies that approximately half the genes between two mates will be exchanged. A feature of UC is that it is isolated from the problem of discontinuous schemata but then again, it potentially destroys continuous schemata.

The shuffle crossover (Caruna et al., 1989) (SC) is another method aimed at improving the robustness of crossovers. It proceeds by randomly shuffling the genes of the mates to a new position, performing some other crossover mechanism (like 1PC or MPC) before finally un-shuffled back to their original positions. Shuffling is aimed at reducing the probability that continuous schemata are disrupted during crossover and also to encourage the transmission of discontinuous schemata across to the offspring. This generalises the shuffling process into an additional operator rather than a true crossover because it can be made independent of the crossover mechanism that is eventually deployed.

The purpose of the shuffle operator is also present in another sexual operator – inversion operator[1] (Holland, 1975). Inversion proceeds similarly like the shuffle operator, except that random segments of genes are selected for inversion rather than shuffling, crossover performed, and then restored back to its original form. The process of inversion is to reduce the length of the discontinuous schemata and thus increase the likelihood that such schemata have a greater chance of being transferred across to the offspring. Despite the proof by Holland that inversion works if the resulting schemata has a shorter defining length before crossover, it has largely fallen out of use due to inability to reproduce the results empirically. Nevertheless, it has found its revival as a solution to tackling epistasis (Goldberg, 1989b). Inversion, like shuffling, can also be considered an additional process rather than an operator.

A typical crossover is necessarily dependent on the underlying genotypic representation. Hence, a specialised crossover used primarily in applications to numeric optimisations is the arithmetical crossover (AC) (Michalewicz and Janikow, 1991a, 1991b). It targets direct

[1] There has been confusion in literature referring to inversion as a unary operator. The actual definition by Holland requires that two structures (genotypes) be selected for inversion, and not one. Hence, it is a sexual (binary) operator.

encoding, particularly real-value representations like floating-point numbers.

The process involved is defined as a linear combination of two genes. Given two chromosomes from the parents, C_1^p and C_2^p, the corresponding chromosomes in the offspring C_1^o and C_2^o is usually given as

$$\begin{aligned} C_1^o &= aC_1^p + (1-a)C_2^p \\ C_2^o &= (1-a)C_1^p + aC_2^p \end{aligned} \qquad 2\text{-}18$$

where the parameter $a = [0..1]$ determines the linear distribution of the value of the genes. It is also not uncommon to apply AC only to a randomly selected number of genes rather than to the complete vector. A feature of AC is that while crossovers involving string-like representations are able to produce offspring that lie outside the numerical bounds enclosed by the parents, AC is unable to do so. The special case when $a = 0.5$ is known as the averaging crossover[1].

2.5.2.2 MUTATIONS AND ASEXUAL OPERATORS

Mutation, as an operator, aims to exploit the latency of high-performing schemata in an environment. A finite population coupled with the restriction of producing only a small number of offspring will lead to gene frequency error. Subsequent sampling, even without additional selective pressure, will draw the gene frequency towards a particular extremum, if no force drives it back to its original state. Since populations generally occur as independent events, it is impossible for a genotype to revert to its original state many generations ago. This phenomenon is known as genetic drift, and is observed in natural adaptive systems. Thus, mutation's main role is to prevent this drift towards any such extrema, although other alternatives such as introducing random immigrants into the population (Grefenstette, 1992) are also feasible.

[1] In evolutionary genetics, the concept of averaging characteristics and attributes has already been long disproved by Mendel in his theory of genetic inheritance. Nevertheless, variations of arithmetical crossover, which goes against Mendelian genetics, are still widely employed in evolutionary computation particularly in applications to numerical optimisation.

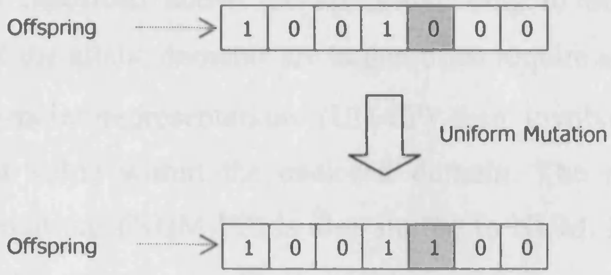

Figure 2-4 Uniform mutation and the flip from a 0 to 1 at the 5th locus.

The process of mutation, particularly in binary-Gray genotypic representations, is a simple one and involves only random bit flips[1] as illustrated in Figure 2-4. Not all genes will undergo mutation and hence a parameter p_M, called the probability of mutation, is used to govern the likelihood a gene will undergo mutation. This form of mutation is known as uniform mutation (UM), as p_M is held constant throughout the search process. Traditionally, p_M is usually assigned small values of around 0.001 and considered a 'background' operator[2]. However, it is now accepted that higher values are equally appropriate, but not arbitrarily too high to shadow the effect of selective pressure and degenerate the search into a random walk. In practical applications, genetic drift towards the later stages of the search is probably an advantage rather than a disadvantage due to its convergent nature. The constancy of p_M in uniform mutation makes this impossible.

Non-uniform mutation (NUM), as the name suggests, varies p_M by some predefined strategy. The most commonly used strategy would involve some deterministic decrease of p_M over time, $\lim_{t\to\infty} p_M(t) = 0$ (Fogarty, 1989). A slightly more complicated approach would be to encode p_M directly into the genotype and allow even it to undergo self-adaptation in order to arrive at an optimal mutation rate (Bäck, 1992). The process begins by randomly assigning appropriate values to p_M for each genotype at $t = 0$. During the next mutation process, p_M is mutated based on its own probability. The resulting new p_M will then be the probability that the genes will undergo mutation.

[1] There are two ways of implementing the flip. One is to force the flip from 0 to 1 and vice-versa. Another way is for equal probability between 0 and 1 to be assigned to the gene.

[2] Mutation and selection alone is highly successful for numerical optimisation in another form of evolutionary computation – Evolutionary Strategies. The argument as to whether mutation is a background operator also has its parallel in real evolutionary studies.

Both UM and NUM described above are applicable only to bit representations. Non-bit representations, where the allelic domains are larger, often require a smaller p_M. The uniform mutation for floating-point representations (UN-FP) then involves replacing a gene with another floating-point value within the enclosed domain. The non-uniform mutation for floating-point representations (NUM-FP) is also similar to NUM. If the domain of a gene is $[g^{UB}, g^{LB}]$ and it has the value g, then the mutated gene can be

$$g' = \begin{cases} g + \delta(t, g^{UB} - g) & \text{if } rand(0,1) = 0 \\ g + \delta(t, g^{LB} - g) & \text{otherwise.} \end{cases} \qquad \text{2-19}$$

where $\lim_{t \to \infty} \delta(t,x) = 0$, and $rand(0,1)$ is simply a random bit flip.

Most of the other asexual operators are mainly repair operators and therefore, are encoding and problem specific. Their use is particularly common in constrained optimisation problems where genotypes are often 'repaired' so that constraints are satisfied in problems such as TSP. However, the use of such repair mutations may be computationally expensive when objective and constraint evaluations are complex.

2.5.3 Replacement

One important genetic operator is to determine a strategy on how genotypes in the transitional population $\overline{\mathbf{X}}^E(t+1)$ and those in the original population $\mathbf{X}^E(t)$ are merged to create the population of the next generation $\mathbf{X}^E(t+1)$. This step in the plan is known as replacement or reinsertion.

The merger of parents and offspring to create the new population can be done in a variety of ways. There are generally two classes of replacement strategies – generational and steady-state. Traditionally and in the simplest GA, the generational strategy is often used where offspring unconditionally replace the entire population of parents.

It is commonly presumed, albeit erroneously, that population size using the generational replacement strategy has to remain constant and hence imposes a restriction on selection mechanisms to sample the number of genotypes equal to the population size. Most natural adaptive systems that tend to display a seasonal pattern in reproduction[1] often produce substantially more offspring than there are number of parents. Of course, most offspring die before they reproduce (predation, environmental effects, etc.) and hence maintain a stable but by no means constant population size. This can be suitably simulated in the generational strategy. However, there is an issue of efficiency pertaining to additional fitness evaluation costs for the additional genotypes.

The steady-state, or incremental, replacement strategy requires the selection and genetic operation of a considerably smaller number of offspring, typically one or two. These offspring are then evaluated and reinserted into the population. The process of selecting the genotypes that will be replaced can be done using any one of the criteria below:

1. *Parents*. Replace the parents responsible for the reproduction.
2. *Weakest*. Replace the weakest genotype in the population.
3. *Oldest*. Replace the oldest genotypes in the population. This then requires an additional 'age' counter to be tagged along each genotype.
4. *Random*. Randomly replacing any genotypes in the original population.

Using either the parents or random criteria above is approximately similar to the generational approach particularly when the magnitude in the number of offspring is comparable. However, an additional replacement criterion can also be imposed such that an offspring only truly replaces the selected genotype if it has a better fitness than the replacement candidate, or fitness above some fitness threshold (usually population average fitness).

The advantage of steady-state over generational strategy is in the number of sampled and processed offspring. In complex fitness evaluations, the lower processing cost of steady-state is attractive. The earlier insertion of generated genotypes back into the population also adds to a steadier and faster increase in selection pressure, particular fitness-based replacement, for successive populations.

From a different perspective, replacement strategies are akin to selection mechanism because they discourage (or encourage) weaker genotypes from further contribution to the search. Thus, it is also possible to merge both the original and transitional population and using some

[1] Unsurprisingly, the generational strategy finds its parallel in natural adaptive systems. In nature, reproduction in many species often takes place seasonally and many, if not all, parents eventually die and their offspring becomes the next generation.

selection mechanism (RWS, SUS, etc.) to sample genotypes for the new population. Such a strategy, particularly when using stochastic sampling mechanisms, often increases the average fitness population while reducing the differential and hence, gives a guarantee towards convergence[1]. A strategy that unconditionally preserves the best genotypes in the new population is said to exhibit elitism.

During replacement, it is also not uncommon to additionally insert new genotypes into the population. This introduction of random immigrants has the benefit of injecting genetic diversity and hence reduces genetic drift effects. Another approach is to penalise the reproduction of similar genotypes by replacing genotypes that are most similar with offspring. The latter technique is known as crowding.

2.6 Adaptive Plans, $\boldsymbol{\tau}$

Following the definition in the framework, the adaptive plan $\boldsymbol{\tau}$ is the cumulative strategy that determines the genetic processes involved in the evolution of genetic structures that individually represents particular points on the landscape.

$$\boldsymbol{\tau} : \boldsymbol{\mu} \times \mathbf{X}^{E}(t) \rightarrow \boldsymbol{\Omega} \qquad \text{2-20}$$

Traditionally, all genotypes in a population of every generation undergo the same genetic operators. For example, every population will be sampled, crossover and mutation performed, and replaced to generate the next population. Such an implementation is simple and has been widely reported for its success across different fields of research. However, it is also possible to implement operators in other ways such as alternating between performing only crossover or only mutation in a particular generation where the best genotypes produced are only replaced back into the population (steady-state).

Optimisation problems can be multimodal, such as in practical application of GA into design and optimisation of composite laminates (discussed later). Here, there are many candidate solutions that indicate global optimality and hence discovering them is equally as important as finding the optimum solution. The ability of GA to simultaneously process a set of solution

[1] Convergence also implies genetic drift, which may or may not be a desired effect particularly in early stages of the search.

vectors gives it a major advantage in such fields. The naive implementation will be to conduct several runs to locate these optima, but success is not guaranteed. There are generally two distinct groups of plans that are suitable for the discovery of multiple optima – parallel and niche plans.

2.6.1 Parallel and Distributed Implementations

An interesting approach to adaptive plans is to allow subsets of genotypes to undergo different operators. Such implementations employ parallel architecture and are known collectively as parallel GA (p-GA). With continuously decreasing computation overhead and the increased availability of distributed computing, the developments and applications of p-GA have been more thoroughly researched and benefit from the amenability of GA to parallel computation without requiring further modification. Such plans have been particularly successful in tackling multimodal optimisation problems as separate populations are allowed to evolve simultaneously with minimal interaction and hence, have a greater likelihood of converging into separate optima. Parallel implementations are also particularly attractive when computation costs for fitness evaluations are high.

The traditional approach to parallel implementation is the farmer-workers model, where a centralised single 'farmer' handles the core GA processes while the 'workers' perform fitness evaluations. This methodology often involves delegating function evaluations to separate physical processors. Each processor is then responsible for the handling of a single genotype, and is particularly attractive in solving highly complex and difficult problems where traditional optimisation and search techniques fail due to time complexities. In such cases, maximum utilisation of computation resources is attained by immediately reinserting evaluated genotypes back into the central population for further genetic processes. A downfall to this approach is the bias towards genotypes that evaluate faster, which may or may not be desirable. This modelling approach, known as panmictic, often forces genotypes to mate and compete with one another without isolation and usually involves only a single population and thus do not guarantee the discovery of multiple optima. Therefore, this implementation in itself is not strictly a parallel model per se.

A more accurate parallel model is the island model. It proceeds by allowing several populations[1] to evolve simultaneously and independently of each other most of the time. They are then allowed to interact intermittently by exchanging genotypes if migration is implemented. There is no restriction as to what lower-level GA model each population implements. Each island is then allowed to model any GA implementation. If and when populations do interact, there can be further distinguishing of 'geographical locations', where only islands close to one another may allow migration.

This is similar in context to actual evolutionary processes whereby geographical isolations often produce different organisms that are able to exploit the same environmental conditions. Such isolation is particularly promising in cases where domain spaces are unbounded or too large, and therefore each population is forced to evolve only to a constrained subspace of the domain proper. Without any or limited interactions between populations, the corresponding genotypes are restricted to the evolutionary model they are subjected to and are often referred to as structured. It is also possible to incorporate hybridisation into the model such that individual populations undergo additional domain specific optimisation.

Yet another approach is the diffusion model, where genotypes are distributed over some grid-like structure and allowed to interact only with its immediate neighbours. A generalisation of this model is if the genotypes are extended to include a population, which then becomes similar to an island model with migration.

2.6.2 CLUSTERS AND NICHE IDENTIFICATION

Before the wide deployment of parallel implementations of GA, techniques like crowding (Goldberg, 1989a) and fitness sharing (Goldberg and Richardson, 1987) were used for multimodal optimisations and are highly successful. In fitness sharing, genotypes compete for the same resources in a closed environment. This environment is typically bounded by relationship through a genotypic or phenotypic distance threshold, σ_{share}. By having to share the same resources, genotypes are weakened[2] and the resulting reduced fitness causes genotypes to cluster around different optima. However, this technique requires the

[1] There is also the use of the term *sub-population* to refer to this subset of genotypes in the overall GA process. However the term population here is used to immediately imply that there is no immediate link of genetic operators or processes between these groups of genotypes in the form of employing the same selection mechanism, crossover, etc.

[2] Weakened here points to the expected number of replicas in the sampled population due to fitness sharing.

determination of the difficult σ_{share} parameter, which in turn should be an estimation of the location and number of optima, both generally unknown parameters (Deb and Goldberg, 1989). Techniques that construct genotype clusters are known as niche induction techniques.

Another approach is through a sequential identification of niches (Beasley, et al., 1993), not unlike the approach discussed earlier. The difference lies in the additional process of suppressing the fitness values within certain radii of identified optima in previous runs. If the depression of the fitness within the region is not done properly, this may lead to additional optima that are not part of the landscape in the first place. On top of that, the additional storage requirements for tracking identified optima particularly in problems with high dimensionality can lead to a slowing down of the search process during actual implementation.

More recent developments in niche identification are centred on the use of some 'pattern recognition' algorithm, in particular clustering algorithms commonly associated with data analysis and mining. A particular interesting approach includes the use of a 'density-based' clustering algorithm (Ester et al., 1996; Ester et al., 1998), known as the GDBSCAN algorithm, in the automatic identification of clusters in any evolutionary-based algorithm (Streichert, 2003). The problem still lies in correctly assigning an appropriate distance parameter, σ_{dist}, which in numerical optimisations can simply be the Euclidian distance. However, in more complex problems such as those experienced later in applications to optimisation of composite laminates, it is unnatural to assume that design variables of differing nature and behaviour (thickness, fibre orientation, etc.) can be assigned an appropriate Euclidian distance.

2.6.3 EVOLUTIONARY ALGORITHMS

The approach of modelling natural adaptive systems is not unique to GA alone. There exists a general class of algorithms that model evolving systems and are known as evolutionary algorithms (EA). The similarities of such algorithms lie neither in their operators nor in their structures or representations (of solutions), but simply on the underlying notion of an adaptive system.

One particular class of EA that has attracted a lot of research, particularly in numerical optimisations, is Evolution Strategies (ES). It traditionally encompasses only a single individual as opposed to GA population, although extensions have now been made to include multi-member populations. Its main difference to GA is in the genetic representation, where a pair of floating-point vectors is used, $\boldsymbol{\upsilon} = (\mathbf{x}, \boldsymbol{\sigma})$. The vector $\mathbf{x}$ represents the solution vector of an optimisation problem, while $\boldsymbol{\sigma}$ is a vector of standard deviations. The only operator used is the mutation operator and is realised by replacing $\mathbf{x}$ at time t, $\mathbf{x}^t$ with

$$\mathbf{x}^{t+1} = \mathbf{x}^t + N(0, \boldsymbol{\sigma}) \qquad 2\text{-}21$$

where $N(0, \boldsymbol{\sigma})$ is a vector of independent random Gaussian numbers with zero mean and standard deviation $\boldsymbol{\sigma}$. The offspring, $\boldsymbol{\upsilon}^{t+1} = (\mathbf{x}^{t+1}, \boldsymbol{\sigma})$ replaces the parent if and only if its fitness is better. This traditional approach is now often referred to as the $(1+1)$-ES, and other variants now exists that show promising results in numerical optimisations.

Other algorithms such as Evolutionary Programming (EP) techniques were initially intended for the evolution of artificial intelligence. Its representation uses finite state machines. Genetic Programming (GP) instead of solving a problem by building an evolution program, searches for the best computer programs that solve it. Therefore, a population consists of a group of executable programs, traditionally represented with LISP (List Processing). Both methods also require some operators that allow interaction between their respective structures.

It can be seen that the objects encapsulated by the GA framework are mutual across other EA although the actual implementation may differ. In fact, there is a constant flow of ideas between EA and this has led to the development of more generic evolutionary algorithms, such as the use of multi-individual ES.

2.7 Closing Remarks

GA was not traditionally intended for applications in numerical and engineering optimisation, but provided a generalised framework for modelling adaptive systems. It is still actively being pursued as a suitable machine learning framework in the field of artificial intelligence.

Due to its flexibility and relative robustness in a broad range of problem domains, GA is now gaining considerable support and application in engineering and structural design and optimisation. The increasing availability of distributed computing and parallel architecture has made GA more attractive, especially in tackling previously intractable problems that were unmanageable with traditional methods due to high dimensionality and time complexity.

Since GA works on stochastic transition rules and a set of solution vectors, it has led to, understandably, the wrong assumption that the search processes involve long computation time and more unfortunately, GA have been branded as merely 'improved' random searches. In most non-trivial cases, the main overhead lies in the fitness evaluation, which is more an indication of the complexity of the problem domain rather than that of the GA itself.

A particular common criticism thrown at GA is the lack of heuristics in guiding the decision to specifying suitable values for the various control parameters such as population size, probability of crossover and mutation rates. However, such problems are not isolated to GA alone as most optimisation techniques equally require definition of parameters. Contrary to general perception, there are general guidelines for assigning appropriate values to the control parameters. Population size is always proportional to the complexity of the problem domain and almost always reflected in the number of genes in the genotypes. GA often employs the 'high-crossover low-mutation rate' rule as opposed to other EA like ES. The emphasis on the higher crossover rate is to stress that GA are 'self-learning' algorithms capable of planning a suitable trajectory across the problem landscape from a constantly evolving knowledge repository modelled as schemata, as opposed to 'discovery' algorithms that traverses the domain usually by identifying the best surrounding points. This 'self-learning' behaviour of GA also makes it more robust in dynamic environments.

GA does not guarantee that the optimum is found because they are particularly weak in high-precision problems, but nevertheless they are sufficiently reliable to ensure that near-optimal solutions are obtained. Indeed, this reliability in identifying optimal regions should prove attractive enough for future developments of hybrid methodologies, particularly with gradient-based methods, for use in more practical applications such as engineering design and optimisations. The only difficulty lies in the lack of a simple to deploy and reliable niche identification technique that can be easily incorporated into GA without incurring the need to tweak additional control parameters.

As GA and indeed other EA continue to mature, they have been used in more practical applications. The successes of their applications have been reported many times over across different fields, and the next chapter introduces a new efficient constraint handling technique for constrained optimisation problems.

3 CONSTRAINED OPTIMISATION WITH GENETIC ALGORITHMS

This chapter covers the development of a new method for the handling of constraints for Genetic Algorithms (GA). It starts by introducing the standard constrained optimisation formulation and discusses some of the alternatives for constraint handling commonly employed in GA. Then, new and efficient methods for constraint handling will be presented. Various tests performed to gauge their robustness are then discussed, together with full results of comparisons and conclusions made to the performances of other commonly employed methods.

3.1 INTRODUCTION

GA, as mentioned in the previous chapter, was initially developed as an adaptive system aimed mostly as search algorithms particularly for machine learning and artificial intelligence. Its lack of direct correlation with the behaviour of objective functions (continuous, discontinuous, etc.) is a major departure from traditional optimisation techniques as it relies mostly on a measurement of performance. This departure from direct objective or cost values is particularly beneficial in multimodal landscapes fraught with noises. The independency on gradient or other direct information gives GA an additional robustness and has therefore been widely used for solving optimisation problems. Due to its highly robust nature in problem solving, it has increasingly gained applications into diverse fields of studies particularly in practical engineering optimisation.

However, most if not all engineering optimisation problems have a set of constraints that define the feasibility or preference of solutions. There are generally two kinds of constraints – domain bounds and system constraints. Domain bounds restrict and define the range of values in which the solution vectors are valid. GA deals with such constraints directly in the genotypic representation and they are therefore not considered as constraints in subsequent discussions below. System constraints define a set of preferences for solutions. It is these constraints that are considered in GA satisfaction of a set of design loads for an engineering problem being a prime example. Considering only a single objective optimisation problem, the standard formulation,

$$
\begin{array}{lll}
\textbf{Optimise} & & \\
\qquad f(\mathbf{x}), & \mathbf{x} \in \mathbf{X}, & \\
\textbf{such that} & & \\
\qquad h_i(\mathbf{x}) = 0, & i = 1,2,\dots,m, & \\
\qquad g_i(\mathbf{x}) \leq 0, & i = m+1,\dots,n, & \text{3-1} \\
\textbf{where} & & \\
\qquad \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB}, & & \\
\qquad \mathbf{x} \in \mathbf{F} \subseteq \mathbf{S} & &
\end{array}
$$

The objective function $f(\mathbf{x})$ is defined on the search space $\mathbf{S}$, and the set $\mathbf{F} \subseteq \mathbf{S}$ defines the feasible regions of the search space. The search space $\mathbf{S}$ itself is a hyper-dimensional rectangle defined by the domain bounds, which are determined by their respective upper and lower bounds of the variables in consideration. The feasible region is bounded by a set of n system constraints defined by the set of equality $h_i(\mathbf{x})$ and inequality $g_i(\mathbf{x})$ constraints. All equality constraints are active at all points $\mathbf{x} \in \mathbf{F}$.

The functions $f(\mathbf{x})$, $h_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are not necessarily differentiable, but can instead be a mixture of complex unrelated types such as linear or nonlinear, continuous or discrete, and analytical or procedural. The above formulation is the representation for the general nonlinear programming problem. Arriving at the global optimum is usually possible only if the objective function and the set of constraints satisfy certain properties, such as convexity. Methods proposed for solving the general nonlinear programming problem can be generally divided into two groups – indirect and direct.

Indirect methods decompose the general nonlinear programming problem into one or more linear problems and solve it as a set of linear problems. Direct methods, however, tackle the problem head on by trying to derive a trajectory through the search space leading towards the global optimum. Traditional methods, like gradient-based methods, often transform the original problem into its unconstrained equivalent and use gradient knowledge to determine successive search points. Many indirect and direct methods have been widely documented elsewhere and are outside the scope of the current work, and hence will not be reported here.

Based on the above definitions of indirect and direct methods, GA can be classified as a direct method, although the trajectory is not instantly identifiable[1]. There exists also a wider group of methods that proceed via a heuristic search. Whether GA belongs to this wider group or to the former group is believed to be nothing more than semantics, but its search proceeds stochastically by constantly updating its projected trajectory through the search space using schemata as indicators of optimality.

3.2 Constraints in Genetic Algorithms – A Literature Review

In the standard formulation for the general optimisation problems presented in Eq. 3-1, there are two immediately distinct groups of constraints – equalities and inequalities. In methods such as linear programming, equality constraints are easily handled by exploiting the fact that the optimum, if it exists, lies at the surface of the convex set. The handling of inequalities, however, is less direct and requires the introduction of slack variables into the system. The system then proceeds by moving from one vertex to another around this surface.

On the other hand, GA has opposite characteristics to linear programming dealing excellently with inequality constraints but being rather weak when handling equalities. This is due to the stochastic nature of the underlying system. However, methods for handling equality constraints are available that reduce the whole optimisation problem to one that is mainly constrained by inequalities. One particularly simple method is to carefully eliminate equalities present in the set of constraints by variable substitution and producing a new set of constraints that are governed by inequalities, together with the domain bounds of the vector of variables. This method is only easy in the case of linear equalities and becomes increasingly complicated in the presence of nonlinearity. Nonlinear equalities are usually converted into inequalities by introducing accuracy parameters, where given a nonlinear equality $h_i^{\mathrm{NL}}(\mathbf{x})$, the resulting inequality is

$$-\delta \le h_i^{\mathrm{NL}}(\mathbf{x}) \le \delta \qquad 3\text{-}2$$

The accuracy parameter δ is normally sufficiently small so that the resulting solutions are as close to satisfying the constraints as possible.

[1] The trajectory is inherent in the sampling of the schemata, and is therefore not directly apparent in the search.

Another interesting, albeit more difficult, approach to handling equalities is to design specialised genetic operators. The main task of these operators will be to maintain satisfaction of equality constraints by limiting transformation of potential solutions, or genotypes, to the constrained but not necessarily fully feasible solution space. The reason that such operators are not expected to maintain full feasibility is the notion that there may be inequalities present as well, and these can be handled more readily. Furthermore, expecting full feasibility can be particularly difficult. Such methods are only suitable for systems whereby designing such operators is easy. One particular difficulty lies in the possibly large number of active[1] constraints at the optimum. The process of satisfying a large number of constraints using such operators is an optimisation process itself, and therefore developing a universal generalised operator applicable to a substantially wide range of problem types is probably an impossible task.

As mentioned previously, GA is well-suited to constrained problems with inequalities. As such, various approaches have been proposed to handle inequality constraints and they primarily fall into the traditional optimisation technique of using penalty functions. The concept of penalty function is centred on incurring an additional penalty to objective function values for non-feasible solutions. Without loss of generality, consider a typical minimisation problem, a solution vector $\mathbf{x}$ together with the original raw objective function $f(\mathbf{x})$

$$f'(\mathbf{x}) = f(\mathbf{x}) + p(\mathbf{x}) \qquad 3\text{-}3$$

The function $p(\mathbf{x})$ is zero if the solution vector is fully feasible with respect to the complete set of constraints in consideration, or some positive value otherwise. Generally, a set of penalty functions $p_i(\mathbf{x}) \in p(\mathbf{x}), i \leq n$ are considered, where the complete set of equality and inequality constraints are combined. Hence, the i-th penalty function provides a measurement of violation for the i-th constraint. This is typically done in the following manner, where the penalty of the i-th penalty function is

$$p_i(\mathbf{x}) = \begin{cases} |h_i(\mathbf{x})| & \text{if } 1 \leq k \leq m, \\ \max\{0, g_i(\mathbf{x})\} & \text{otherwise.} \end{cases} \qquad 3\text{-}4$$

[1] Active constraints imply constraints where the solutions lie at their boundary. Obviously, all equality constraints are active at any point in the feasible region.

The implementation details of various methods differ largely with respect to the penalty functions and their application to infeasible solutions. A review of constraint handling methodologies in evolutionary algorithms was reported by Michalewicz (1995), although a more critical (literally) perspective can found in a paper by Coello (1999c), which also covers other taxonomies.

Proposed constrained handling methodologies often require the definition of a certain number of system parameters. This is due to the need to assign the amount of constraints violation to a fitness value that is reflective of global optimality, which then gives rise to another problem: For almost all such approaches, there often exists an optimal set of values for the system parameters that guarantees the best results for a set of problems.

The search for this optimal set of parametric values is an optimisation problem on its own. More often than not, the complexities of such systems goes beyond trivial analyses, and are therefore usually reported with sets of 'best-performing' system parameter values that are often arrived at empirically. Such ad hoc approaches leave a yawning chasm to be bridged if such methods are to be applied successfully to real optimisation problems. The matter is slightly alleviated if the set is a small one as several trial-and-error runs can be made to arrive at intuitive rule-of-thumb values. However, the problem is compounded in cases where the number of such parameters depends entirely on the nature of the optimisation problem and computationally cheap and rapid sensitivity evaluations are impossible.

3.2.1 Death Sentence

The traditional approach to constraint handling in GA and indeed most other evolutionary-based algorithms has been to impose a direct penalty on infeasible genotypes by disregarding the amount of violation it has incurred. The method completely discards or minimises the contribution of infeasible genotypes from subsequent contribution, hence the term 'death sentence' or death penalty. This, though, can be done in a myriad of ways.

The easiest approach to achieving this will be to simply discard infeasible genotypes from the population and generating new ones until the required population size of genotypes is fully feasible. Such an approach, of course, breaks down if the proportion of the feasible space to the overall search is negligible.

Another more flexible and arguably better approach will be to allow infeasible solutions to coexist with feasible ones in the population by assigning an appropriate fitness. Contrary to common belief, it is not necessary for infeasible solutions to have fitness that is no better than that of the worst infeasible solution. Rather, it is sometimes more appropriate that infeasible solutions are merely never better than the best feasible solution. This allows sub-optimal infeasible genotypes to appear to be equally as good as feasible ones. However, such an approach is beginning to look more like a penalty-based approach. Nevertheless, the rationale behind this is that GA is equally likely to benefit from feasible as well as infeasible genotypes. Consigning infeasible genotypes to the death penalty implies directly discarding possibly beneficial schemata, which itself may bring detrimental effects. In optimisation, this has a direct implication in that the search is almost entirely directed from the feasible region rather than a two-prong approach from both the feasible and infeasible regions. Such an approach has a beneficial side-effect, which can be fully exploited by a later proposed method to handling constraints in GA.

3.2.2 Penalty Functions

Methods based on the traditional penalty function approach come in various shapes and sizes for GA and can be generally divided into two groups – static and dynamic penalties. The static penalty formulations discount the effects of GA time. Therefore, genotypes experience the same selection pressure regardless of the point in time as long as the amount of violation remains the same.

The opposite is true for dynamic penalty formulations whereby the penalties imposed on infeasible genotypes are dynamically increased as a function of time t where $\lim_{t\to\infty} p_i(\mathbf{x},t) = \infty$[1]. The reason for doing so is to first allow the GA to search through a combination of feasible and infeasible solutions, before progressively focusing the search and filtering onto regions with decreased violation. This leaves the use of such dynamic formulations with the task of balancing between exploration and exploitation of the search space. An over-zealous approach will undoubtedly force the search to be directed only from the feasible regions. Whether this is an intended effect and beneficial is highly subjective and problem dependent.

One approach to penalty functions (Powell and Skolnick, 1993), denoted M1, involves the

[1] The use of the infinite symbol is dependent on the context of the problem. It is denoted as such to imply a substantially large enough value so that infeasible solutions are almost likely to be discarded, as in the death sentence approach.

classical formulation with one notable exception, where fitness values are assigned according to the formulation

$$f'(\mathbf{x}) = f(\mathbf{x}) + r\sum_{i=1}^{n} p_i(\mathbf{x}) + \lambda(\mathbf{x},t) \qquad 3\text{-}5$$

In essence, the parameter r determines the selection pressure on infeasible genotypes. The exception, however, is in the $\lambda(\mathbf{x},t)$ component. This additional component forces a further comparison between feasible and infeasible genotypes by applying the additional heuristic rule: For any feasible genotype and any infeasible genotype, the feasible individual is always better than any infeasible one. In other words, assuming that higher fitness implies better optimality, infeasible individuals cannot have higher fitness than the worst of infeasible individuals. The difficulty in this approach lies in the formulation of the $\lambda(\mathbf{x},t)$ component. Often, it is implemented as a rule, and is updated as the search progresses and as more solution points are considered. The method is also sensitive to the correct value of the parameter r although it can be safeguarded against by a good formulation of $\lambda(\mathbf{x},t)$.

Another proposed penalty function approach (Schoenauer and Xanthakis, 1993), denoted M2, processes the constraints sequentially in a linear fashion. It proceeds with a random population of possibly both feasible and infeasible genotypes. A constraint counter k, used to trace the number of constraints satisfied, is introduced and initialised to 1 at the start. The population is evolved and the fitness assigned as

$$f'(\mathbf{x}) = p_k(\mathbf{x}) \qquad 3\text{-}6$$

It proceeds as such until a given percentage of the population, the flip threshold ϕ, is feasible for the k-th constraint. Once achieved, the constraint counter is incremented, where the $k \leftarrow k+1$-th constraint is now considered, and the fitness of genotypes in the current and subsequent populations evaluated as before. This is repeated until the flip threshold is again attained. If at any point a genotype does not satisfy any of the previous constraints considered, it is immediately dropped and eliminated from the population. The method proceeds in such a way until all constraints are eventually satisfied. Once fully satisfied, the genotypes are then assigned the fitness according to the raw objective function. By then, any genotypes not satisfying all the constraints are rejected.

It should be immediately apparent that this is not strictly a penalty-based approach because there is no standard formulation for assigning fitness values. The discontinuous treatment of constraints and eventual discarding of genotypes that do not satisfy earlier handled constraints often leads it to behave more like the death sentence. Therefore, this makes it highly sensitive to the size of the feasible region. It is also not known how the search will behave if the order of constraints handled is different. The method also needs to maintain diversity in the population rather than converge onto a specific region with each handled constraint, therefore requiring the deployment of some niche induction techniques. That having said, the method is particularly powerful in situations that makes it impossible to deploy a monotonic fitness function with penalty formulation.

Another method (Homaifar et al., 1994), denoted M3, employs a set of penalty coefficients for every constraint considered. First, a set of levels of violation L, for each constraint, is determined. Then an appropriate penalty coefficient R_{li} is defined for the l-th level of violation of the i-th constraint. The GA process begins with a random population of possibly both feasible and infeasible genotypes. Each genotype is then assigned fitness according to the formula

$$f'(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} R_{li} p_i^2(\mathbf{x}) \qquad 3\text{-}7$$

The weakness of this method clearly lies in the number of system parameters in L and R_{li}. In most non-trivial optimisation problems, it is not easy to generalise a set of suitable parameters that is sensitive to the search domain, let alone a universal set for all problem types. The sensitivities of the penalty coefficients are important as they ultimately decide the selection pressure of the search process. Failing to tune the coefficients to the nature of the problem in hand may leave the 'rewards' of infeasible solutions too attractive to resist, thus guiding the search towards a local optimum or even an infeasible solution.

One dynamic approach (Joines and Houck, 1994), denoted M4, includes the effects of varying timelines on the selection pressure of infeasible solutions. Dynamic, or non-stationary, penalties incorporate GA time into the penalty functions, and in this method, the fitness is assigned as

$$f'(\mathbf{x}) = f(\mathbf{x}) + (C + t)^{\alpha} \sum_{i=1}^{n} p_i^{\beta}(\mathbf{x}) \qquad 3\text{-}8$$

Three system parameters - C, α and β - controls the general selection pressure. Although suitable values reported indicate that $C = 0.5$, $\alpha = \beta = 2.0$, these control parameters still require sensitivity analysis to ensure stable search progress. The indiscriminate application of values to system parameters may lead to over-zealous penalising of infeasible solutions or insufficient pressure to drive the search towards optimal regions. If the former is true, the search may degenerate into a search from only the feasible region far too soon since the death sentence is inadvertently simulated.

In general, penalty approaches based on dynamic penalties can be formulated as

$$f'(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} p_i(\mathbf{x},t) \qquad 3\text{-}9$$

The above formulation for dynamic penalties allows the weightings of violation effects as a function of time t, where generally $\lim_{t\to\infty} p_i(\mathbf{x},t) = \infty$ is to ensure that the search develops eventually into a search for only feasible solutions. The consideration of different weights for different constraints may be necessary particularly in cases when certain constraints result in tight feasible regions, i.e., they are difficult to satisfy.

Another dynamic approach (Michalewicz and Attia, 1994), known as GENOCOP II or denoted M5 here, involves also the use of a system parameter τ, which is termed as the 'system temperature'. This, like many of the previously described approaches, determines the selection pressure on infeasible solutions. The process begins by first initialising the temperature to an initial value, $\tau = \tau_0$, and terminates once a so-called 'freezing' temperature $\tau = \tau_f$ has been reached. At any temperature τ, the fitness are assigned based on the formulation

$$f'(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2\tau} \sum_{i=1}^{n} p_k^2(\mathbf{x}) \qquad 3\text{-}10$$

Suggested starting and freezing temperatures are given as $\tau_0 = 1$ and $\tau_f = 1 \times 10^{-6}$, where the temperature decreases with each iteration by $\tau \leftarrow 0.1 \times \tau$.

It should be noted that the GENOCOP II model encapsulates a lower-level GA, the GENOCOP, where a single iteration implies a complete evolution of this underlying system through to convergence. It starts by first selecting a starting point that forms the basic template for the genotypes in GENOCOP subjected to a set of constraints. Specialised mutation operators are applied to introduce genetic diversity and the evolution of GENOCOP is performed until some termination criterion is met. The best genotype from the evolution becomes the template for the genotypes in the next iteration of the GENOCOP II model with a reduced temperature and a new set of constraints violated by the template genotype.

The complexity of this approach lies in the need to possibly perform several complete GA processes. Whether there is significant advantage of simpler models that are highly-tuned to the nature of the problems is inconclusive and unreported. Furthermore, the effects of varying the decrement function of the temperature apart from the linear function above have not been reported. On the other hand, the same report did mention that the temperature parameter is sensitive to problems, and thus requires additional evaluations to extract a suitable set of values to guarantee good results.

3.2.3 Multi-Objective Constraint Handling

There are also developments in using multi-objective concepts in handling of constraints in GA. All, however, are based on the same concept of redefining n constraints (equalities and inequalities alike) and the objective function into an $n+1$ multi-objective problem. Multi-objective optimisation procedures can then be applied on the vector of objectives, $f_0, f_1, \ldots, f_n$, where $f_1, \ldots, f_n$ are constraints, described in more detail below. This adoption also makes virtually all methods identical, except in the implementation details and approach towards the optimum.

Parmee and Purchase (1994) implemented a version of Vector Evaluated Genetic Algorithms[1] (Schaffer, 1985) (VEGA) for constraint handling. Each sub-population is made to handle a single objective (cost objective and constraints) to allow the GA to locate feasible regions in a highly constrained space. Instead of proceeding to use VEGA to explore the detected feasible regions, specialised operators were deployed to create variable-sized hyper-cubes around such regions to assist the GA in remaining within them. The report however did not go on to discover the global optimum, so it remains to show if optimal solutions can indeed be found.

[1] VEGA forces distinct sub-populations to be affected only by a single objective of a multi-objective problem, and thereby specialising genotypes. Crossover, however, did not differentiate between the various distinctive sub-populations. Heuristics such as inter-breeding and wealth/resources redistribution were developed. Studies showed that there is a tendency to converge towards solutions that were not optimal with respect to any objectives.

Furthermore, specialised operators prove a major limitation for more general usage.

Surry et al., (1995) included the above multi-objective definition, as well as Pareto-dominance (see below for formal definition) to handle constraints in the COMOGA method. Pareto-dominance was used to rank genotypic preferences of a population based on constraints satisfaction. However, selection is not based fully on this information. A fraction of the next generation, determined by a threshold parameter that is not dissimilar to the flip threshold ϕ of M2, will be selected based on dominance, while the remainder will rely on actual cost value. The threshold parameter is readjusted during run-time depending on preference for cost optimisation or constraints satisfaction. An important drawback of this approach lies in the increase in computational overhead as the number of constraints increases, which is a problem noticeable also in some penalty-based approaches.

At first glance, the method proposed by Camponogara and Talukdar (1997) appears to be similar to the one proposed here (see below). The method restates a constrained optimisation problem as a dual-objective problem, where the aim is to optimise the main cost function and minimise

$$\phi(\mathbf{x}) = \sum_{i=1}^{n} \max(0, p_i(\mathbf{x})) \qquad 3\text{-}11$$

The similarity ends here. Rather than proceed to use multi-objective tools to explore search space, non-dominated solutions are generated with respect to the two objectives. These are then used to define a search direction. Furthermore, no crossover is applied but search is projected along the defined direction, which limits the appeal for a more general GA application since impact of segments selected is unclear.

Jiménez and Verdegay (1999) proposed a method that, again on first glance, appears similar to the one proposed here (see below). The difference is in the min-max rules for genotypic preference. Comparisons are made between overall violations, by selecting the maximum violation values, and objective costs. However, constraints are non-dominated in the sense that the genotype with the lowest maximum violation is always preferred. If genotypes are both feasible, then costs become the basis of comparison. The drawback is in the lack of accuracy in constraint handling, since different classes of constraints may inevitably have wide ranges in violation values.

Coello (1999a; 1999b) proposed a population-based modelling VEGA to handle individual constraints and cost function instead, which implies $n+1$ sub-populations. A novel approach to fitness assignment is then made to drive the GA into progressively reducing constraint violations, until feasibility is attained. The primary drawback at the moment is in the lack of heuristics in the population size required for each sub-population. On top of that, the effects of a high number of constraints (with increasing number of sub-populations) on computation overhead may again be significant.

The above methods derived constraint handling methodologies from the discipline of multi-objective optimisation. Two common problems befall the above approaches – computational overhead, and introduction of specialised operators. Such issues will hopefully be tackled more easily in the proposal of a method that is also based on multi-objective principles, as presented below.

3.3 Feasibility of Genotypes: The Multi-Objective Approach

Part of the aim of this work is to develop a highly efficient constraint handling technique that minimises the number of system parameters as a consequence of the technique for reasons discussed earlier. On top of that, it has to be capable of easy deployment without requiring excessive reimplementation of the underlying technique. Such a requirement arose due to the difficulty of determining a different optimal set of system parameters for different problem sets often associated with other constraint handling techniques, particularly penalty-based approaches.

Most of the currently available techniques often work on the objective function itself, transforming it into a monotonic fitness function by the addition of a penalty component that penalises infeasible solutions as shown in Eq. 3-2. The difficulty arises in cases when the use of penalty components is nigh on impossible. One particular situation illustrating such a scenario is when the use of too small a penalty leaves the basins of attractions of infeasible regions too rewarding to resist, thus leading the search to converge onto infeasible solutions. Although a higher penalty may emphasise regions of optimality, it will do so at the expense of transforming it to isolated peaks of optima with steep and almost vertical slopes with flat surrounding regions. If the search does indeed find such peaks, it will probably accomplish it by virtue of the random nature (through mutation) of the process itself rather than through gradual improvements. Furthermore, the use of a single monotonic fitness function incorporated with penalties obviously transforms the original fitness landscape. Indiscriminate use of penalty functions without careful analysis may lead to unwanted behaviours such as introduction of multimodality, flatness, discontinuities, etc. Therefore, perhaps the better option is to isolate both the objective/cost component and the violation of constraints as they both were initially, rather than force a marriage of the two.

Another problem arises when objective functions and constraints are not necessarily discernible. As mentioned earlier, it can even be as complicated as procedural or analytical. The use of penalty methods in such situations is again virtually impossible, and the death sentence may leave out too much useful knowledge for any good. Such a situation may occur if constraints arise only by virtual of the satisfaction, i.e., requiring sequential treatments.

In any constrained optimisation problem, the main aim is to search for solutions that minimise constraints violation while optimising the objective function. The ideal situation is when full feasibility is achievable.

In the standard formulation of an optimisation problem, the whole system can be considered as the multi-criterion or multi-objective optimisation of $n+1$ criteria since the aim is to discover a set of solutions, if it exists, that minimises the violation of the set of n equality $h_i(\mathbf{x})$ and inequality $g_i(\mathbf{x})$ constraints and optimises the objective $f(\mathbf{x})$.

Multi-objective problems are often defined by a set of objective functions $f_0, f_1, \dots, f_n$ over the original search space $\mathbf{S}$, where

$$\textbf{Optimise } f_{\text{MOBJ}} = (f_0, f_1, \dots, f_n) \tag{3-12}$$

Solutions to multi-objective problems are usually not a unique singular solution, but a set of equally efficient, non-dominating, alternative solutions that are part of the Pareto-optimal set. The keyword here is dominance, and the Pareto-set is characterised by it. Consider two solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ in a multi-objective problem. The solution $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$ if its optimality (here assumed to be minimisation again without loss of generality) against each of the objective functions f_i is partially better than that of $\mathbf{x}_2$, and is defined formally as follows

$$\begin{array}{lll} & \forall i \in \{0,1,\dots,n\}: & f_i(\mathbf{x}_1) \le f_i(\mathbf{x}_2), \\ \text{and} & \exists j \in \{0,1,\dots,n\}: & f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{array} \tag{3-13}$$

A solution $\mathbf{x}_1$ is then said to be Pareto-optimal or non-dominated if there exists no other solution $\mathbf{x}_2$ that dominates it.

Consider again the constrained optimisation formulation from Eq. 3-1 and reformulated into a multi-objective problem formulation as

$$f_{\mathrm{MOBJ}} = (f(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}), g_{m+1}(\mathbf{x}), \ldots, g_n(\mathbf{x})) \qquad 3\text{-}14$$

This newer formulation gives a multi-objective formulation with $n+1$ criteria. The formulation can be rewritten as

$$f_{\mathrm{MOBJ}} = (f(\mathbf{x}), C_1(\mathbf{x}), \ldots, C_n(\mathbf{x})) \qquad 3\text{-}15$$

where $C_i(\mathbf{x})$ represents the full set of constraints, both equalities and inequalities, in consideration. It is hereby noted that a higher value of $C_i(\mathbf{x})$ here indicates higher violation[1], whereby $C_i(\mathbf{x}) = |h_i(\mathbf{x})|$ if $i \leq m$, and $C_i(\mathbf{x}) = \max\{0, g_i(\mathbf{x})\}$ otherwise. The standard constrained optimisation problem is now reduced to an unconstrained multi-objective problem, which makes it easier to adapt to GA as will be later shown. Closer inspection reveals that the above formulation can be further reduced into a two-objective problem by defining the subset of constraints a measurement of violation, f_{CON} , where

$$f_{\mathrm{CON}}(\mathbf{x}) = (C_1(\mathbf{x}), \ldots, C_n(\mathbf{x})) \qquad 3\text{-}16$$

which then produces the simplified form

$$f_{\mathrm{MOBJ}} = (f(\mathbf{x}), f_{\mathrm{CON}}(\mathbf{x})) \qquad 3\text{-}17$$

With the above simplified form, many issues must still be addressed with regards to handling f_{CON} and f_{MOBJ}, without which it will be unable to differentiate between the optimality of one solution and another. There exist many various methodologies drawn from multi-

[1] If the reverse is true, then the aim is to maximise the feasibility rather than to minimise the violation.

objective optimisation, and only those implemented in the proposed techniques will be discussed below. Nevertheless, there are two possible approaches to handling the constraints violation f_{CON} given above – priorities and weighting.

3.3.1 Lexicographical Comparison of Prioritised Constraints

The approach using priorities is based on a lexicographical comparison of dominance (Fonseca and Fleming, 1993, 1998a, 1998b). Priorities can be used to determine the sequence in which feasibility is maximised or violations minimised. The formal approach in multi-objective optimisation is to assign a priority value, i, which is typically an integer where higher values correspond to higher priorities of the individual objective function. In the case of GA, the underlying cost function can then be viewed as a universal measurement of violation, where f_{CON} is such that for two genotypes with phenotypic values $\mathbf{x}_1$ and $\mathbf{x}_2$

$$\begin{aligned} & f_{\text{CON}}(\mathbf{x}_1) < f_{\text{CON}}(\mathbf{x}_2) \Leftrightarrow \\ & \forall i \in \{1,2,\ldots,n\} : \exists j \in \{1,2,\ldots,n\}, \\ & C_i(\mathbf{x}_1) \le C_i(\mathbf{x}_2) \wedge C_j(\mathbf{x}_1) < C_j(\mathbf{x}_2) \end{aligned} \tag{3-18}$$

This approach is suggested for constraints that require sequential processing and are dependent, i.e., when one constraint arises only by virtue of the satisfaction of another[1].

For the general multi-objective problem, traditional search methods that deal with a single solution at any one point in time optimises the highest prioritised objective before proceeding to the next in the list of prioritised objectives. Adopting such an approach can be achieved easily as evident in the penalty-based approach M2 discussed earlier. However, it should be recalled that GA encapsulates a population of solutions at any one time, and thus sequentially processing the constraints can and probably should be regarded as a waste of resources (evolutionary time). Furthermore, it is not impossible for genotypes with several levels of violations to co-exist in a single population. The question, however, arises as how best to represent the feasibility of genotypes that has satisfied differing number of prioritised constraints.

[1] This is also possible in natural adaptive systems whereby organisms need to adapt to environmental pressures (constraints) of a different set due to their current adaptability and exploitation of a particular environmental niche.

An easy approach will be to sort genotypes based on the Pareto-dominance of feasibility. The resulting ranked population can then undergo some sort of rank-based (linear, quadratic, nonlinear, etc.) violation and fitness value assignment. This avenue has already been explored in the COMOGA approach (Surry et al., 1995) with known problems. As the number of constraints in the system increases, performing Pareto comparison can lead to serious computational overhead. The suggestion here, however, is to assign a universal violation value that implicitly perform Pareto comparison with minimal overhead.

The naive approach to a universal violation value will be to select several discrete levels of violation to represent the number of constraints satisfied, where feasible solutions will obviously have zero violation. However, such an approach leaves the need to determine the values that best represent such levels of violation. Similar to M3 (presented above), the number of parameters will be unwieldy with a large number of constraints. Hence, an alternative approach is required that ideally introduces zero system parameters (to reduce sensitivity to problems) and minimises computational overhead.

The suggestion here is to first normalise the range of violation of all considered constraints into a standard region $[0,U]$, where $U>0$. With n constraints in consideration, the universal measurement of violation is then bounded by $0 \le f_{\text{CON}}(\mathbf{x}) \le U$ and can be assigned as

$$f_{\text{CON}}(\mathbf{x}) = (i-1)\frac{U}{n} + C_i^U(\mathbf{x}) \qquad \text{3-19}$$

where $C_i^U(\mathbf{x})$ is the normalised violation of the i-th constraint such that $0 \le C_i^U(\mathbf{x}) \le U/n$. Here, the priority value i can as easily be used to indicate the i-th constraint in consideration. A noticeable feature of the above formulation is that the satisfaction of each constraint implies a shift in the upper bound of f_{CON} by U/n. Of course, the ideal situation is when genotypes have $f_{\text{CON}}=0$, which implies full feasibility, while anything higher will imply partial satisfaction or maximum violation when $f_{\text{CON}}=U$. This formulation also gives the added advantage of automatically inducing implicit feasibility ranking based on Pareto-dominance into the genotypes without having to perform additional sorting, thus saving precious resources. The previous statement should be immediately obvious since no two genotypes satisfying differing number of constraints can possibly share the same violation value.

Possible routes to normalisation of the constraints violation can be taken from fitness handling techniques in GA, particularly when dealing with minimisation problem. Again, the ranking method can prove useful and used to rank genotypes that satisfy the i-th constraint according to their true violations and assign a new (prize) value, $C_i^U(\mathbf{x})$ in the range $[0, U/n]$ based on individual performances in the resulting league table. The criticism of this method is the same as when applied to fitness – indiscriminate disregard for behaviour of true violation values in the resulting assigned values. Furthermore, difficulty arises when there exists only a single genotype in the sampled population that satisfies the i-th constraint, which then requires either assigning the highest or lowest prize to it.

Another approach to normalisation, which still maintains the behaviour of the amount of violation between genotypes, is to assign the normalised violation according to

$$C_i^U(\mathbf{x}) = \frac{C_i(\mathbf{x})}{k_i} \times \frac{U}{n} \qquad \text{3-20}$$

The parameter k_i can be drawn from many sources, where one is to initially assign a value large enough value such that $\max\{C_i(\mathbf{x})\} \le k_i$. However, this is only suitable if it is easily identifiable either empirically or through careful analysis. Then again, the approach now requires n parameters k_i.

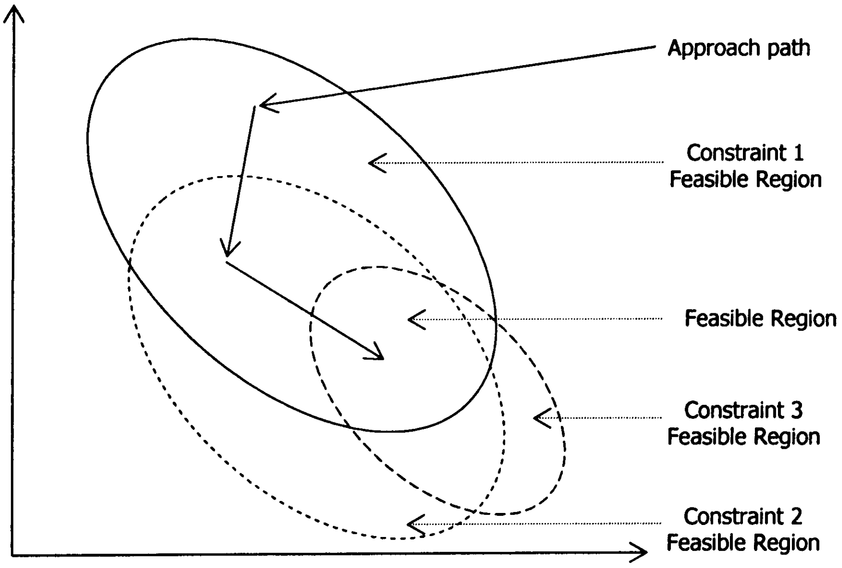

Figure 3-1 Directional approach path in prioritised constraints.

One solution is to let $k_i = \max\{C_i(\mathbf{x})\}$, which means that it can either be the largest violation observed in the current population or encountered so far in the whole evolution for the i-th considered constraint. There is, however, a noticeable shortcoming with this solution, and the problem is evident when the highest observed violation of a constraint is updated. Such an update usually requires existing genotypes with feasibility normalised on a previous k_i value to be re-evaluated. This is particularly true when variants of steady-state replacement strategies are used since a comparison between offspring and parent needs to be performed. The compromise, though, is to decide whether this re-evaluation gives additional advantage to the search or is computationally too expensive to perform, and just allow the algorithm to proceed with this discrepancy inherent in the population as noise.

The lexicographic approach to constraint handling is useful when sequential treatment of constraints is required, or a progressive satisfaction of constraints is desirable. However, this makes an assumption that there exist such priorities exist and are easily identifiable. Otherwise, they need to be artificially introduced[1]. Priorities, in essence, form a predictable path of approach towards feasible regions by progressing towards areas of increasing feasibility, as shown in Figure 3-1. However, it often makes little sense in assigning priorities arbitrarily. The alternative option will be to use the weighting approach to multi-objective optimisation, which is not unlike the penalty component of penalty-based approaches.

3.3.2 Sum Violation of Non-Prioritised Constraints

A commonly used approach for handling multiple objectives in multi-objective optimisation is to introduce a new objective function that is a weighted sum of individual objectives. In the context of constraint handling, it is the measurement of violation where

$$f_{\mathrm{CON}}(\mathbf{x}) = \sum_{i=1}^{n} a_i C_i(\mathbf{x}) \qquad a_i \in \Re^+ \qquad 3\text{-}21$$

The definitions for $C_i(\mathbf{x})$ are as before but positive weighting coefficients, a_i, are introduced. In this approach, the full set of n constraints is considered unlike the lexicographic approach. The aim is then to minimise f_{CON} until full feasibility is achieved.

[1] Artificial introducing of priorities in the handling of constraint may be beneficial in certain cases even when no natural sequential treatment is required. One such case is when constraints are particularly complex and expensive to perform. This avenue will be explored again later.

One limitation is that this approach can be sensitive to the setting of the weighting coefficients a_i, particularly when assigned arbitrarily. If weighting coefficients are not tuned to the problem, then preferential treatment of constraints will arise. This is because different constraints may have different violation levels and, unless precautionary steps are taken, two genotypes may share the same measurement of violation even though feasibility is different. In most cases, there exists no such preference of constraints and the unintentional simulation of it may even be disastrous. Such a situation may arise, for example, when there is a mixture of constraints where some restrict the feasibility space to a small negligible region. Another is when weights do not correctly scale the effects of violations (or are sensitive to the problem), as evident in the penalty-based approach M3. Another issue is of course the introduction again of n system parameters. Since the main aim is to introduce an effective constraint handling technique that minimises the effects of such parameters and reduces the number of system parameters (ideally to zero), a different approach is necessary.

The suggestion here is to completely discard the weighting coefficients, achievable by setting $a_i = 1, \forall i \in \{1,2,\ldots,n\}$. Without such parameters, hopefully the approach is more applicable to a wider class of constrained problems since sensitivities to them are reduced. Violations are again normalised to a unit space, as before in the lexicographical approach, which reduces the measurement of violation to

$$f_{\text{CON}}(\mathbf{x}) = \sum_{i=1}^{n} C_i^U(\mathbf{x}) \tag{3-22}$$

As before, the bound $0 \le f_{\text{CON}} \le U$ now holds. It is convenient to simply normalise $U = n$ which results in the violation bounds of individual constraints to $0 \le C_i^U(\mathbf{x}) \le 1$, which then reduces the normalisation formulation to

$$C_i^U(\mathbf{x}) = \frac{C_i(\mathbf{x})}{k_i} \tag{3-23}$$

The limitation in the sum-weighted approach is the lack of clear indication of the number of constraints satisfied, although crude approximations due to normalisation may be possible. The Pareto-ranking of feasibility in this case is, at best, a rough estimation.

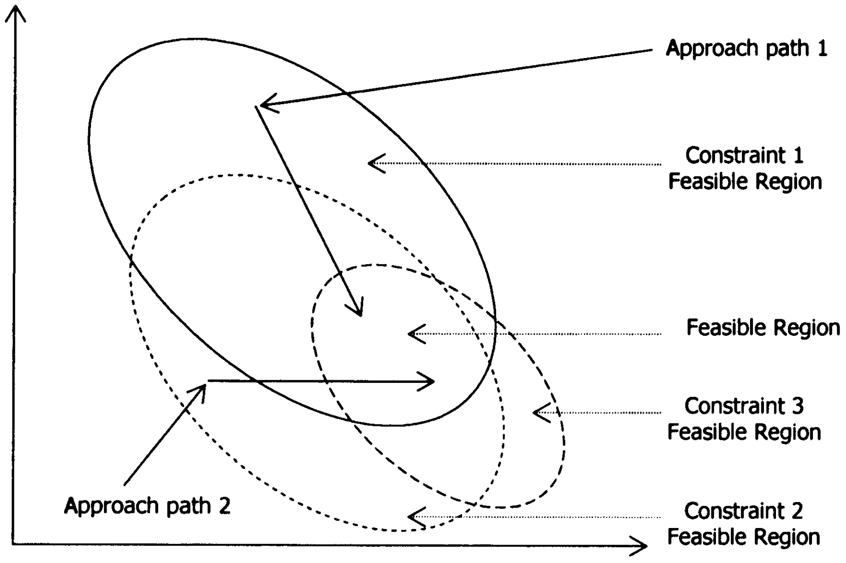
Approach path 1
Constraint 1 Feasible Region
Feasible Region
Constraint 3 Feasible Region
Approach path 2
Constraint 2 Feasible Region

Figure 3-2 Omnidirectional approach paths in non-prioritised constraints.

The benefit over the lexicographical model, though, is that the search is not forced to proceed via a predetermined path. This gives rise to a wider possibility of locating feasible regions from various perspectives and increasing detection of them, as shown in Figure 3-2. Nevertheless, there is also considerable expense in the approach, in terms of computation overhead, particularly if measurements of violations of individual constraints require complex analytical procedures. In such circumstances, the progressive approach taken by the lexicographical method is probably more suitable. However, no reports were found that considered the effects and tradeoffs between a directional (lexicographical) and omnidirectional (sum-weighted) approach towards feasible regions. An empirical study on a constrained numerical problem using both these methods will be presented later.

3.3.3 Fitness of Infeasible Genotypes

With the above suggested methods for obtaining a measurement of violation f_{CON}, there is still a need for assigning an appropriate fitness to genotypes. There are two general approaches – Pareto-ranking or penalty-assignment.

3.3.3.1 Pareto-Ranking of Genotypes

The Pareto-ranking approach is the more natural model since the constraints have been handled so far based on concepts drawn from multi-objective optimisation. Recall again that the re-formulated constrained optimisation problem is now

$$f_{\mathrm{MOBJ}} = (f(\mathbf{x}), f_{\mathrm{CON}}(\mathbf{x}))$$

This approach involves first comparing genotypes on the basis of their Pareto-dominance in terms of the measurement of violation f_{CON}, and only then by cost f if they share the same dominance. In essence, this means using the lexicographical approach and setting the priority of f_{CON} higher than that of f.

The benefit of this approach is that it leaves out the need to specifically formulate a function that represents the resulting multi-objective function f_{MOBJ}. However, only a restricted subset of GA sampling mechanisms is able to directly utilise and exploit this knowledge of dominance, tournament-based selection mechanism being one of them. It is also worth pointing out that with this approach, the search process usually separates into two general phases – constraint satisfaction and objective optimisation.

The search will most likely start by minimising the violation particularly when no feasible genotypes are present in the population. Only if more than one feasible solutions exists, i.e., when $f_{\mathrm{CON}} = 0$, will comparisons be made based on objective value and the system then begins to optimise cost. Whether this is beneficial is highly subjective. Problems are likely to occur when feasible regions are tight, sparse and disconnected. The consequence, unless some precautionary steps are taken, is that the search for the optimal solution will finally occur from the feasible region and converging onto local optima. Thus, the search is left incapable of deriving and maintaining highly-feasible near-optimal solutions, assuming of course that such solutions exist and search trajectories through such regions are possible.

One precautionary step may be to maintain within a population, a subset of high-performing infeasible solutions. In other words, feasibility now becomes an asymptotic event for these genotypes. The population then degenerates into two populations where one strives for feasible optimal cost while another aims for continuous minimisation of violation without ever being fully feasible. This solution, though, is not foolproof since problems may also arise if problems where the feasible regions are sparse and disconnected which then allows high-performing infeasible solutions to possibly cling on to a single region of minimum violation that leads to local optima.

A useful way out of this is to use niche induction techniques to maintain variability of such genotypes in the hope that clusters of genotypes will form around different locations. There is also no reason not to allow the size of this subset (number of genotypes) to vary dynamically as a function of evolutionary time to ensure that the final population of genotypes are feasible rather than a combination of both. Parallel implementation with island models may be more suitable here since then two separate populations can evolve independently of each other and only periodically some individuals, like the elitist group, come together on a third 'interaction' island and exchange information. This is not explored in this work but discussed further as a possible future development.

3.3.3.2 PENALTY-ASSIGNMENT OF GENOTYPES

If the broader group of selection mechanisms such as RWS and SUS are to be applicable, then inadvertently a tangible fitness value has to be derived based on the information of f_{CON} and f to produce a suitable f_{MOBJ}.

The easiest will be to still perform sorting of genotypes based on dominance as outlined in the Pareto-ranking approach. Then, the resulting sorted population of genotypes can have their corresponding multi-objective value f_{MOBJ} assigned based on their individual ranks in the resulting league table using some linear or nonlinear prize function. However, criticisms surrounding ranking methods again surface here.

To avoid such criticisms and other problems associated with ranking, then an approach not dissimilar from penalty-based approaches can be applied – penalty functions. Consider now in the cases of minimisation, Eq. 3-14 can now be reformulated as

$$f_{\text{MOBJ}} = \begin{cases} f(\mathbf{x}) & \text{if } f_{\text{CON}}(\mathbf{x}) = 0, \\ \delta + f_{\text{CON}}(\mathbf{x}) & \text{otherwise.} \end{cases} \qquad 3\text{-}24$$

The δ component is used to define how feasible and infeasible solutions are actually compared. In other words, it is an indication of selection pressure on infeasible genotypes and the same is actually true of all penalty-based techniques. If the traditional model whereby the best infeasible solution is no better than the worst performing feasible solution is subscribed, then δ can be the objective value of the worst performing feasible solutions. However, the problem arises when no feasible solutions have yet been encountered. It is here suggested that

δ is taken as the maximum observed feasible objective value in a population or throughout the run, or zero if no feasible solution exists yet

$$\delta = \begin{cases} \max\{f(\mathbf{x})\} & \text{if feasible exists,} \\ 0 & \text{otherwise.} \end{cases} \qquad 3\text{-}25$$

Such a model, although commonly used, leaves the algorithm setting a search path only from feasible regions. Therefore, the best use of the δ component is perhaps to formulate a function that allows high-performing infeasible solutions to be as likely to contribute to the search as high-performing feasible ones for reasons discussed earlier.

If the best observed combination of feasibility and cost is from the solution $\mathbf{x}^*$ and the corresponding objective values is $f(\mathbf{x}^*)$, one such function is

$$\delta = kf(\mathbf{x}^*) + \left|f(\mathbf{x}^*) - f(\mathbf{x})\right| \qquad 3\text{-}26$$

The parameter k is used to control the selection pressure of infeasible solutions. In actual fact, k can be completely dropped if a minimalist system is preferred because Eq. 3-24 can be reformulated as

$$f_{\mathrm{MOBJ}} = f(\mathbf{x}^*) + \left|f(\mathbf{x}^*) - f(\mathbf{x})\right| + f_{\mathrm{CON}}(\mathbf{x}) \qquad 3\text{-}27$$

The selection pressure is now independent of any control parameters but merely reliant on the measurement of violation f_{CON}, which arguably is a more accurate representation of the underlying problem when compared to ranking procedures. This, in turn, is then influenced by how f_{CON} is normalised to the unit space U. Provided it is normalised into a sufficiently small unit space, high-performing infeasible solutions will then be able to contribute longer in the search rather than weeded out early. This is because the larger the U, the higher the selection pressure on infeasible solutions and the smaller the likelihood of high-performing infeasible solutions competing on a more or less equal basis with feasible solutions of similar objective value. The above formulation is limited to a single system parameter, which is actually the unit space U. Taking it to the extreme, the technique can be devoid of any system parameter if no normalisation of violation is performed. Anyway, the advantage it provides is the use of the same function for both feasible and infeasible solutions, since f_{CON} disappears completely for feasible ones. The system is now reduced only to the optimisation, or in this case, minimisation, of the resulting new objective function f_{MOBJ}.

3.3.4 Partial Knowledge of Violation

In certain optimisation problems, the constraints are complicated as it could possibly involve either complex procedural analysis or some complicated sequence of evaluations in order to determine the feasibility of solutions. Such constraints usually involve expensive computational overhead and can be regarded as time-complex constraints, and are more noticeable in GA since a population of solutions has to be evaluated and maintained. While genetic operators like steady-state replacement provide a useful respite, it does not guarantee that the search will perform significantly less number of constraint evaluations before convergence. Therefore, a more robust approach for time-complex constraints can be a useful tool in the GA armoury.

The suggested method, here termed lazy evaluation, is straightforward and has probably been applied many times over. If such is the case, then it is reiterated and formalised here since it will be employed later on for practical optimisation. The argument is that for most practical search in constrained optimisation problems, the required solution is simply the best optimum solution with minimal, if not zero, violation. Without loss of generality, let the considered optimisation be a minimisation problem and recall also the earlier definition in the Pareto-ranking approach where the objective f is given a lower priority than f_{CON} when deciding the dominance between two solutions. Lazy evaluation reverses this once a feasible solution is found, giving f a higher priority than f_{CON} although still performing a lexicographic comparison. Therefore, for any new genotypes, constraint evaluations are only performed for those that are possible improvements over the current best known optimal feasible solution, i.e., when $f(\mathbf{x}) \leq f(\mathbf{x}^*)$ if $f_{\mathrm{CON}}(\mathbf{x}^*) = 0$. If the new solution shows improvement, obviously it replaces the best observed feasible solution as the new optimum. The reason that solutions that share the same cost, $f(\mathbf{x}) = f(\mathbf{x}^*)$, should undergo constraint evaluation is the possibility that there may exist a subset of infeasible solutions that share the same objective value. For those that do not need to undergo the full range of constraint evaluations, some 'fuzzy' amount of violation can be assigned.

It is obvious that the difficulty lies in deciding the so-called fuzzy measurement of violation.

In the worst case scenario, violation is assigned to solutions that are otherwise feasible and none is assigned to infeasible ones. The suggested solution here is to assign it some token value ε, where

$$f_{\mathrm{CON}}(\mathbf{x}) = \varepsilon$$

The ε parameter can be considered as a 'safety factor'. However, it is believed that in most cases, $\varepsilon = 0$ should suffice. In other words, only genotypes with $f(\mathbf{x}) \leq f(\mathbf{x}^*)$ will have the true measurement of violation, while others are assumed to be fully feasible. The rationale behind this statement is that this allows both sub-optimal feasible and infeasible solutions with similar cost to compete equally, giving rise to the possibility of developing a search trajectory through infeasible regions. Coupled with f_{CON} that has been tuned to a suitable unit space, this gives the algorithm a large selection of search paths from which to focus in onto the optimal solution or region.

Another suitable method for handling time-complex constraints is by introducing priorities into constraints, albeit artificially. Although most constrained optimisations do not require sequential treatment of constraints, artificial prioritisation may save considerable computation overhead. This is due to the fact that there is no further need to evaluate remaining constraints once one with a higher priority has been violated.

The simple option to introducing priorities will be to consider the constraints randomly. Nevertheless, such an approach leaves much to be desired particularly if a more structured logical approach is required. Therefore, the better option may be to organise the constraints based on their relative 'hardness' in decreasing priority. Hardness here implies the size of the resulting feasible space imposed by the constraint alone, where smaller size naturally equates to harder constraints. The use of a hardness or 'constrainedness' parameter, κ (Gent et al., 1999), may be a suitable measurement. For simplicity though, subsequent application in this work will only consider the random prioritisation with other techniques to be discussed in greater detail as possible future work.

Certain features stand out from the two approaches described above. It should be obvious that using priorities is particularly effective against performing unnecessary constraint evaluations just like lazy evaluations. However, as the search focuses in onto the feasible regions, most genotypes, assuming they are already feasible, will then undergo the full rigour of constraint evaluations. Therefore, artificial prioritisation is beneficial in reducing the number of evaluations only during the 'pre-feasibility' stage. On the other hand, lazy evaluation only becomes significant when some known feasible, not necessarily the optimum, solution has emerged. Thus, it is effective for 'post-feasibility' reduction in number of evaluations. The timeline in which both are uniquely effective then gives rise to an opportunity of integrating both into a singular technique, which may further provide significant discount of computation cost for time-complex constraints. Whether this is a beneficial marriage will be investigated further below when results of application in a constrained numerical optimisation problem are provided.

An important feature and probably of greater consequence is that both lazy evaluation and artificial prioritisation introduces partial knowledge of violation. In both cases, assumptions are made with regards to the true violation of a potential solution. The effect results in a transformation of the problem landscape, and a dynamic one. The reason it is dynamic is obvious: As more knowledge is gained from full scale evaluation, more details of the landscape are revealed. The question now remains as to how the search is affected by such 'partial' genotypes amongst fully-evaluated ones.

The intuitive response to it is that the search will compensate for this impartial knowledge by requiring a longer evolution time to convergence. In other words, more such impartial genotypes will have to be scanned before the true picture of the problem landscape reveals itself to the search. It then begs the question as to whether the end product will require as many constraint evaluations as one that undergoes full evaluation from the start. The crux of the problem is whether this partial knowledge of the landscape is beneficial or detrimental to the search, and at this point, it remains inconclusive and a full mathematical analysis is far from trivial. Nevertheless, an empirical investigation will be performed and results produced below, which should shed some light on such issues.

3.4 Case Study on Constrained Numerical Optimisation

A constraint handling technique drawn from the discipline of multi-objective optimisation has been proposed earlier together with several variations as potential improvements. In order to gauge the robustness of these methods, a case study was performed on a constrained numerical optimisation problem where results for the other penalty-based methods, namely M1 to M5, were readily available and comparisons may be made.

3.4.1 QUADRATIC TEST FUNCTION

The following test function (Michalewicz, 1995) was used to test the performance of the proposed technique and its variants. The quadratic test function (QTC) is to minimise a function

Minimise

$$f(\mathbf{x}) = \sum_{i=1}^{4} 5x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

subject to

$$\begin{aligned}
&2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0,\\
&2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0,\\
&2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0,\\
&-8x_1 + x_{10} \le 0,\\
&-8x_2 + x_{11} \le 0,\\
&-8x_3 + x_{12} \le 0,\\
&-2x_4 - x_5 + x_{10} \le 10,\\
&-2x_6 - x_7 + x_{11} \le 10,\\
&-2x_8 - x_9 + x_{12} \le 10,
\end{aligned} \qquad 3\text{-}28$$

with domain bounds

$$\begin{aligned}
&0.000 \le x_i \le 1.000, && i = 1,2,\ldots,9,13\\
&0.000 \le x_i \le 100.000, && i = 10,11,12
\end{aligned}$$

The QTC has 9 linear inequality constraints with a single global minimum with objective value of $f(\mathbf{x}^*) = -15$ at

$$\begin{aligned}
\mathbf{x}^* = (&1.000, 1.000, 1.000, 1.000, 1.000,\\
&1.000, 1.000, 1.000, 1.000, 3.000,\\
&3.000, 3.000, 1.000)
\end{aligned} \qquad 3\text{-}29$$

3.4.2 EXPERIMENTATION FRAMEWORK

Several experiments were carried out in order to cover the main technique and its other variants. This can be broadly divided into two groups where one consists of variants of Pareto-Ranking (PR) approach and another that are Penalty-Assigned (PA). Detailed descriptions of these two groups and their variations are provided further below.

A general purpose object-oriented GA computer program was written in C++ with 2551 lines of coding[1] to perform the above test problem and other design optimisation problems (presented in Chapters 5 and 6). All tests, including those from Chapters 5 and 6, were conducted on an Intel Pentium III 600MHz PC with 128 MB RAM.

All experiments were conducted using Gray-encoded strings, although experiments for the referenced results on methods Ml to M5 were performed on floating-point representation. The conscious decision was made because it was felt that binary-type representations are more suitable as they are more applicable to a wider range of problems and even to those that are non-numerical. The number of bits in individual chromosomes (a single variable x_i) reflects the precision presented in the domain bounds outlined. Although this resulted in genotypes with a high number of bits, it will also be a good test on the suitability of binary representations particularly in high-precision numerical problems. The required number of genes for each chromosome is as summarised in Table 3-1.

i	Precision	Number of Genes
1,2,...,9,13	3	10
10,11,12	3	17
Total Number of Genes		151

Table 3-1 Number of genes required for each design variable in QTC to reflect the number of significant figures.

The published result employed a population size of 70. However, this is deemed unsuitable for binary-type representations since there will be insufficient genetic variation in a population of that size. Therefore, the population size was assigned to be equal to the genotype string length of 151. Since the reported experiments ran for 5000 generations, this implies 350000 genotypes were evaluated at most. The experiments conducted here compensated for this upper bound by setting the maximum number of generations to evolve to respect this ratio of 350000:70. Non-integer numbers of generations were rounded down, i.e., maximum of 350000 evaluations is preserved.

As mentioned before, the PR approach produces no direct or tangible fitness value, which means that traditional sampling mechanisms break down here. In experiments where this is

[1] Number of lines is approximate since comment lines and line breaks were not excluded. However, it does not include the coding for case studies.

true, the binary tournament selection (B-TS) was used.

For PA though, the resulting fitness allows the use of traditional sampling mechanism, hence the stochastic universal sampling (SUS) was used in such cases. All sampling mechanisms were required to sample a transitional population with size equivalent to the original population. Since no genotypes are removed at any point in the search, this is a constant size of 151.

The genetic operator used for the recombination process is the α-uniform crossover (UC), with uniform mutation (UM) complementing it. The probability of crossover was set at 0.8 while the probability of gene swap was taken to be 0.3, i.e., 3 in 10 genes are exchanged between mates. Initial runs indicate that the 0.5-UC is unsuitable as it is too disruptive and tends to generate offspring that are too dissimilar to their parents. Several mutation rates were also considered but it was finally settled at 1 in 151 because a higher mutation rate was only necessary in cases where epistasis is noticeable (Bãck, 1993). In this case, the linear inequality constraints did not impose a noticeable effect. A summary of the adopted GA parameters are provided in Table 3-2.

GA Parameters	Value
Population Size	151
Probability of Crossover	0.8
Probability of Gene Swap	0.3
Probability of Mutation	1/151

Table 3-2 GA parameters and their corresponding values used in the QTC experimentation framework.

All conducted runs used the generational steady-state replacement strategy. It is only slightly different from the traditional steady-state in the sense that the transitional population size is identical in size to the preceding population rather than the typically smaller size associated with steady-state. This transitional population then undergoes recombination and evaluation. Then, the preceding parent population is merged with the transitional offspring population and the best performing individuals of up to population size are selected to be the new parent population, i.e., using a strong-replace-weak algorithm.

3.4.2.1 NON-PRIORITISED PARETO-RANKING

The first part of the experiment concentrated on employing only the PR approach, denoted GAPR, without additional features. No constraint priorities are implemented. Instead, the sum violation method is used thereby giving all constraints equal priority. Nevertheless, violations for individual constraints are normalised to the $[0,1]$ domain ($U = 9$), which should then give an approximate indication of the number of constraints satisfied by each genotype.

Normalisation is performed with respect to the maximum observed violation so far in the run. For any new observed greater violation, only subsequent genotypes will reflect this change although strictly, any changes should require re-evaluation of the entire population as well as the parent population. However, the additional computational overhead was deemed too expensive since the benefit of doing so was only noticeable in the very early stages of the run. The inconsistency was therefore considered as noise in the system.

3.4.2.2 NON-PRIORITISED PARETO-RANKING WITH LAZY EVALUATION

The second set furthered the GAPR approach by introducing the lazy evaluation feature, denoted GAPR-LZ. Once a feasible solution is present, only new solutions with improved objective value will undergo constraint evaluation. The main aim of this set of experiments was to investigate the reduction, if any, in the total number of constraint evaluations performed compared to GAPR. It was also believed that the partial knowledge of violation in genotypes may require significantly more generations of evolution before convergence. Therefore, the secondary aim was to investigate the trade-off between the number of constraints evaluated and the number of sampled genotypes. Generally, it was believed that significant advantages in terms of computational cost may be gained if constraints are significantly more time-complex compared to the underlying GA process and objective evaluation. Constraints violations, as before, are normalised to the $[0,1]$ domain. No prioritisations are considered because the full set is performed once an improvement in the optimal objective value is deemed possible.

3.4.2.3 PRIORITISED PARETO-RANKING

The third part is also based on the PR approach but introduces priorities, denoted GAPR-P, unlike the two set of experiments above. The constraints are simply prioritised based on the order presented in the outline of QTC (see Eq. 3-28). The algorithm proceeds from one constraint to another and continues to do so while the previous constraint is satisfied. Otherwise, it breaks from performing further constraint evaluation and assigns the violation of the remaining constraints as the maximum observed so far in the run for that constraint. The main aim of this part was to investigate the effect of prioritisation. Furthermore, the feasible knowledge of genotypes is also partial as a result of handling a set of time-complex constraints. Violations are again normalised to the $[0,1]$ domain, which means that no two genotypes with differing number of satisfied constraints can share the same amount of violation. This in turn means that dominance due to feasibility is internally ranked.

3.4.2.4 PRIORITISED PARETO-RANKING WITH LAZY EVALUATION

The fourth part implements the notions presented earlier that artificial prioritisation and lazy evaluations gives additional time-saving in constraint evaluations, denoted GAPR-PLZ. In other words, it combines the process of GAPR-P and GAPR-LZ. The main aim is therefore to study the effect of performing only partial evaluation whenever possible and thus simulating incomplete knowledge of the domain at all stages of the search. It should also be noted that even in the case of a potential improvement over the observed optimum, the candidate solution does not need to undergo the full constraint evaluation as a consequence of constraint prioritisation. Results should then be indicative of the relative robustness of the method in the presence (or lack of) violation knowledge. It should also show, when compared to GAPR, whether excessive trade-off in terms of evolution time and sampled genotypes is required.

3.4.2.5 NON-PRIORITISED PENALTY-ASSIGNED

The fifth and final part deals with the alternative PA approach denoted GAPA. Here, a true fitness value is assigned based on the measurement of violation f_{CON} as presented earlier in Eq. 3-22, and reproduced below

$$f_{\mathrm{MOBJ}} = f(\mathbf{x}^*) + \left| f(\mathbf{x}^*) - f(\mathbf{x}) \right| + f_{\mathrm{CON}}(\mathbf{x})$$

This allows the use of traditional sampling mechanisms, which in this case SUS is employed. The performance is compared primarily against GAPR. The additional need of formulating a fitness function and performing real fitness evaluations is an additional overhead over PR methods. Furthermore, it was believed that this approach requires fitness values to be scaled appropriately in order to be sensitive to the underlying problem. Violations are normalised to the $[0,1]$ domain. With this sensitivity, it was believed that this is the weakest approach overall due to the inclusion of a penalty component in the fitness function, which is also true in other penalty-based techniques in constraint handling.

3.4.3 Results and Discussion

Comparison results for penalty-based methods M1 to M5 were readily available and are reproduced in Table 3-3, for subsequent reference and discussion. It also combines the results obtained for the experimentation, namely for methods GAPR, GAPR-LZ, GAPR-P, GAPR-PLZ and GAPA. For the developed methods, ten independent runs were conducted for each. Each run had a random initial population, although the i-th run of every method is identical[1].

The proposed GAPR and its variants as well as a penalty-based approach GAPA all showed promising results, where all runs conducted managed to find the optimum objective value without difficulty. None of the methods in any run came up with solutions that were sub-optimal unlike M1, M2, M3 and M4, which, in their worst run, returned sub-optimal solutions even at the end of a relatively long evolution with respect to the experimentation results obtained. Even though M1 to M5 employed floating-point representation combined with specialised operators like Gaussian mutation, arithmetical crossover and heuristic crossover, they still did not outperform PA and PR with its variants apart from M5. Furthermore, M3 with its virtually countless number of system parameters even returned a non-feasible median solution.

Methods	Representation	$f(\mathbf{x}^*)$			Violation	CPU
		Best	Median	Worst		
M1	Floating-Point	-15.000	-15.000	-14.999	0,0,0	
M2		-15.000	-15.000	-14.998	0,0,0	
M3		-15.002	-15.002	-15.001	0,0,4	
M4		-15.000	-15.000	-14.999	0,0,0	
M5		-15.000	-15.000	-15.000	0,0,0	
GAPR	Gray-Encoding	-15.000	-15.000	-15.000		31.9
GAPR-LZ		-15.000	-15.000	-15.000		56.7
GAPR-P		-15.000	-15.000	-15.000		30.8
GAPR-PLZ		-15.000	-15.000	-15.000		55.6
GAPA		-15.000	-15.000	-15.000		40.3

Table 3-3 Results for 10 independent runs of penalty-based methods, M1 to M5, together with proposed methods and variants, GAPR to GAPA, on QTC. The violation values are a sequence of three-comma separated numbers indicating the number of violations with violation amount between 10.0 and 1.0, 1.0 and 0.1, and 0.1 and 0.0001 respectively at the median solutions. It is applicable only for M1 to M5. Source for results from M1 to M5: (Michalewicz, 1995). Average solution times per run are given in the CPU column where the values are in CPU seconds.

[1] For example, the initial populations of the 5-th and 6-th run of GAPR are different. However, the initial populations of the 5-th run of GAPR, GAPR-LZ, GAPR-P, GAPR-PLZ and GAPA are all the same.

It has been observed that recent trends in the application of GA into practical problems have called for the use of a genotypic representation natural to the underlying system, numerical optimisation being one of them favouring floating-point representation over smaller building blocks like binary-type representation. The consequence of this is the additional need of developing specialised operators that deal specifically with such encoding. One conclusion that can be drawn from the above results is that using small genetic blocks, in this case Gray-encoding, is capable of handling even high precision numerical problems with relative ease. Recall also that no specialised operators were developed and adopted for this experimentation, as it relies primarily on genetic operators that were already well understood and extensively studied. If such common operators can be deployed with promising results, it does justify the questioning of developing over-complicated GA variants for general problem solving. Granted that specialised operators are suitable and give additional robustness to the underlying GA in problem-specific application, simplicity is still by far the best virtue in the application of GA.

The only additional feature introduced in the experimentation is the use of a slightly different steady-state (SS) replacement strategy, as described earlier. An additional set of experiments comparing GAPR using generational-elitist (GE) and SS were performed. No runs using GE found the true optimum within the stipulated timeframe unlike SS. The search appears to have benefited from having the best individuals in the population consistently with SS as shown in Figure 3-3. One drawback found of SS from the experimentation is that selection pressure induced by the strong-replace-weak algorithm may have been too strong particularly towards the later stages of the search as improvements on the current observed optimal becomes harder. Standard deviation in the population objective value indicates that the search is at risk from losing all genetic diversity if no improvements can be found after a significant period of time. This did occur in several runs as shown in Figure 3-4.

However, it is believed that the search benefited from steady-state replacement because of the quadratic nature of the test case as well as an apparent lack of epistasis. Such a strong selection pressure must be applied with caution particularly in noisy multimodal environments as the search may be driven far too early into local optima. The resulting lack of genetic diversity often also implies inability to escape from them unless mutation or other operators are able to induce such jump from local traps. Then again, its performance over traditional generational-elitist models is far too convincing to ignore in this case.

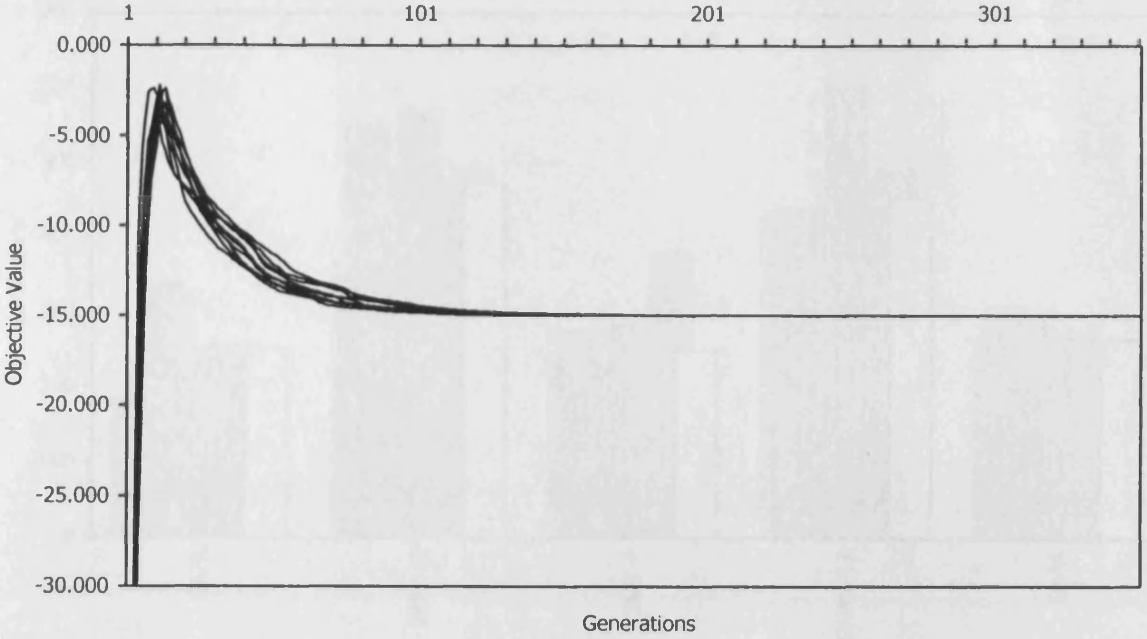

Figure 3-3 Convergence behaviour of all ten GAPR runs under steady-state replacement strategy with respect to the population average objective value. The same pattern is prevalent across all runs for all other GAPR variants as well as GAPA with the only difference being the convergence rate.

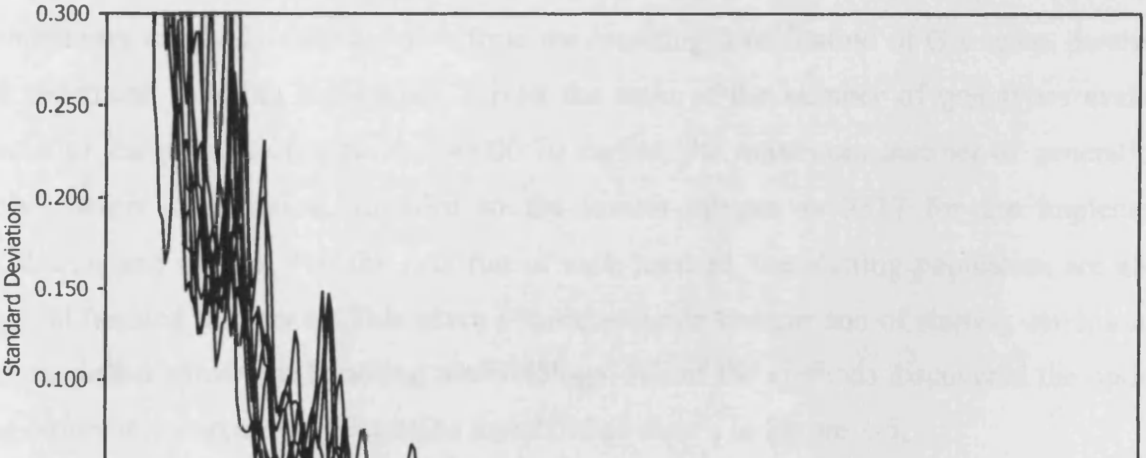

Figure 3-4 Population standard deviation in terms of objective value of all ten GAPR runs. It shows clearly that genetic diversity in later stages is minimal.

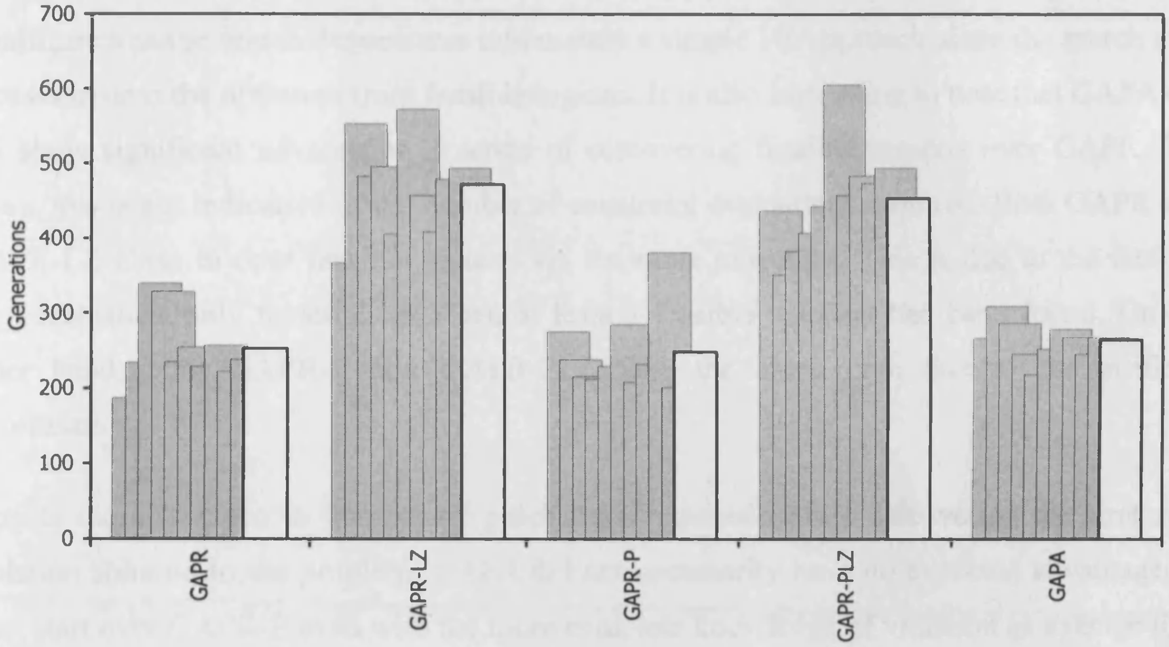

Figure 3-5 Number of generations required on discovering the optimum, i.e., the true optimum appears in the population for all runs and for all constraint handling methodologies. The non-shaded column indicates the average number of generations required for each method.

If robustness can be considered as the ease in which a technique solves a problem, then the problem was relatively easy to solve from the resulting combination of GA setup, parameters and constraint handling techniques. Given the ratio of the number of genotypes evaluated overall to the population size as 350000:70 earlier, the maximum number of generation to evolve before termination, rounded to the lowest integer, is 2317 for the implemented population size of 151. For the i-th run of each method, the starting population are always identical (seeded the same). This gives a more accurate comparison of starting conditions for each modelled constraint handling methodology. All of the methods discovered the optimum long before this maximum generation threshold as shown in Figure 3-5.

The difference in convergence rates for the different constraint handling algorithms are as expected. It is immediately noticeable from Figure 3-5 that those that have only partial knowledge of violations for the majority of the evolution process, namely GAPR-LZ and GAPR-PLZ, typically require almost double the evolution time. This, however, is not indicative of the number of constraint evaluations performed.

The effect of constraints prioritisation, implemented by GAPR-P, on convergence rate is however not obvious globally. This is not surprising since the effects of prioritisation are only more evident before a substantial number of feasible solutions start appearing as summarised in Figure 3-6. Upon appearance of feasible solutions, prioritisation effects no longer bear any significance as the search degenerates into mainly a simple PR approach since the search then focuses in onto the optimum from feasible regions. It is also interesting to note that GAPA did not show significant advantages in terms of discovering feasible regions over GAPR. Yet again, this is not indicative of the number of constraint evaluations required. Both GAPR and GAPR-LZ close in onto feasible regions via the same trajectory. This is due to the fact the lazy evaluation only materialises when at least a feasible solution has been found. On the other hand, both GAPR-P and GAPR-PLZ take the same path due to the artificial prioritisation.

Despite the difference in number of generations required before discovering the first zero violation solution to the problem, GAPR did not necessarily have an expected advantageous head start over GAPR-P even with the more complete knowledge of violation as average time required before discovering the true optimum is roughly equal. Figure 3-7 shows the difference in the knowledge of violation amounts between the best genotypes in GAPR and GAPR-P. Since the i-th run for each method begins with the same population, the individuals from GAPR represents the actual violation amount if individuals undergo the full set of constraint evaluations. It should also be noted that GAPR also represents GAPR-LZ at this stage just as GAPR-P represents GAPR-PLZ.

The next logical question is then to ask whether this average increase in evolution time with partial knowledge is justifiable. If prioritisation is to provide sufficient computational advantage in terms of overhead reduction, the number of evaluations should be significantly below the upper bound. The amount of reduction is, however, dependent on the complexity of the constraints themselves. The critical question as to whether prioritisation is beneficial cannot be answered definitively, since many other factors also influence the computational overhead such as genotypic representation and resulting number of genes (encoding/decoding functions), population size, complexities of genetic operators, etc. Furthermore, it is impossible, based on this isolated experimentation, to conclude for certain how constraint hardness affects the search for discovering feasible regions. Nevertheless, if the time required for performing constraint evaluations outstrips the computational cost of the underlying GA system, then prioritisation can be used to reduce the number of constraint evaluations.

Since GAPR undergo full constraint evaluations, its number of evaluations forms the upper bound as shown in Figure 3-8 for all methods. It is immediately obvious that there is substantial reduction in the early stages of the search before feasible solutions appear.

Thereafter, the number of evaluations increases almost linearly as more and more genotypes begin to stabilise into sub-optimal feasible regions. This virtually linear growth and almost parallel to the upper bound in the number of evaluations continues in such a manner for GAPR-P until convergence.

Lazy evaluation, on the other hand, is evident and beneficial only after feasible solutions start appearing. GAPR-LZ focuses in onto feasible regions in the same manner as GAPR since no constraints are prioritised, visible in the small section at the bottom left corner at approximately the 10-th generation in Figure 3-9. However, as more such sub-optimal feasible individuals appear, the algorithm only performs full constraint evaluations if they are potential improvements over the current observed optimum.

In the case of QTC, the majority of the search process is concentrated on this phase, hence allowing lazy evaluation to give an increased bonus in terms of constraint evaluations as the search progresses. However, just like prioritisation discussed earlier, it is unwise to blatantly assume that lazy evaluation gives definitive benefits since many other parameters also do have influences. Therefore, the trade-off between longer evolution time and evaluations should always be decided locally to the underlying optimisation problem itself. In the case of QTC, the best GAPR-LZ run gave about 50.8% savings over the best GAPR run in terms of number of constraint evaluations to discover the optimum. Figure 3-9 also shows that in all runs conducted for GAPR-LZ, the increase in evaluations is relatively stable across different starting conditions although the same cannot be said about the number of generations required till convergence.

Given the evidence that prioritisation affects primarily the early stages of the search before discovering the feasible regions, or 'pre-feasibility' in GAPR-P, and lazy evaluations thereafter, or 'post-feasibility' in GAPR-LZ, the natural question arises as to whether the marriage of the two in GAPR-PLZ can yield further significant reduction in computational costs of constraint evaluations. Prior to the experimentation, the conjecture was that savings can be made and the results have indicated likewise as shown in Figure 3-10 since there will be some occasional encounters with infeasible genotypes. Results also indicate that the integration works far better than even GAPR-LZ in isolation. This is not unexpected.

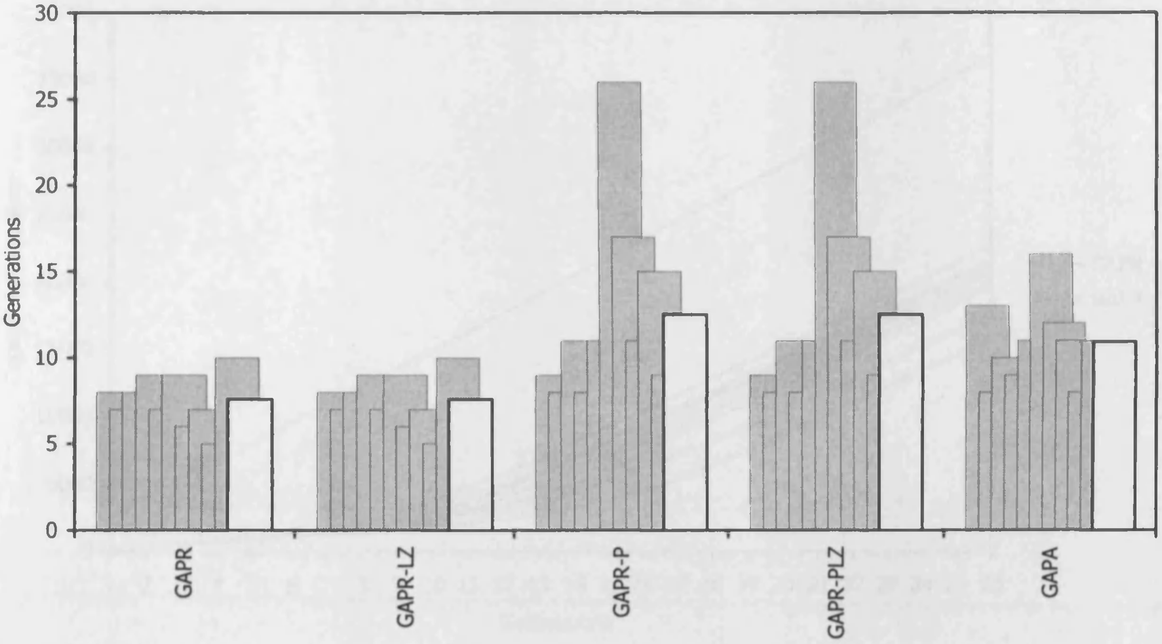

Figure 3-6 Number of generations required on discovering the first fully feasible (zero violation) solution for all runs and for all constraint handling methodologies. The shaded column indicates the average number of generations required for each method.

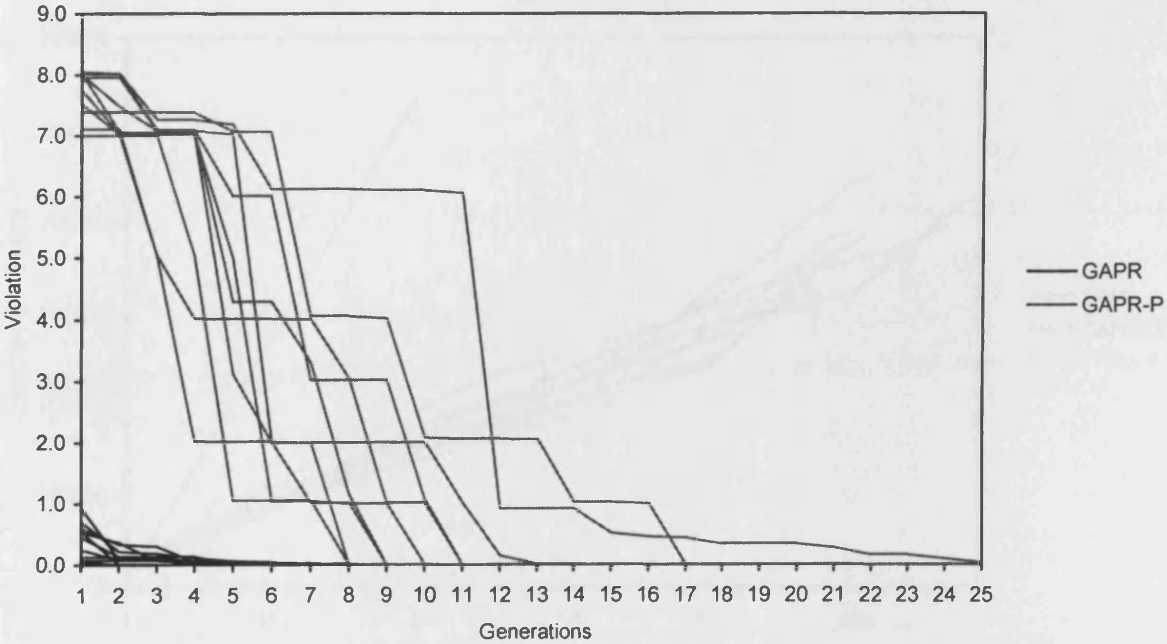

Figure 3-7 Normalised violation for the best genotype observed in every generation at the start of the evolution for all runs of methods GAPR and GAPR-P before the first feasible solution is discovered for the worst run in GAPR-P (on generation 26). With full constraint evaluations, genotypes from GAPR have more accurate knowledge of violation and can be seen to be clustered in the bottom left corner. On the contrary, individuals from GAPR-P do not undergo the full set of constraint evaluations.

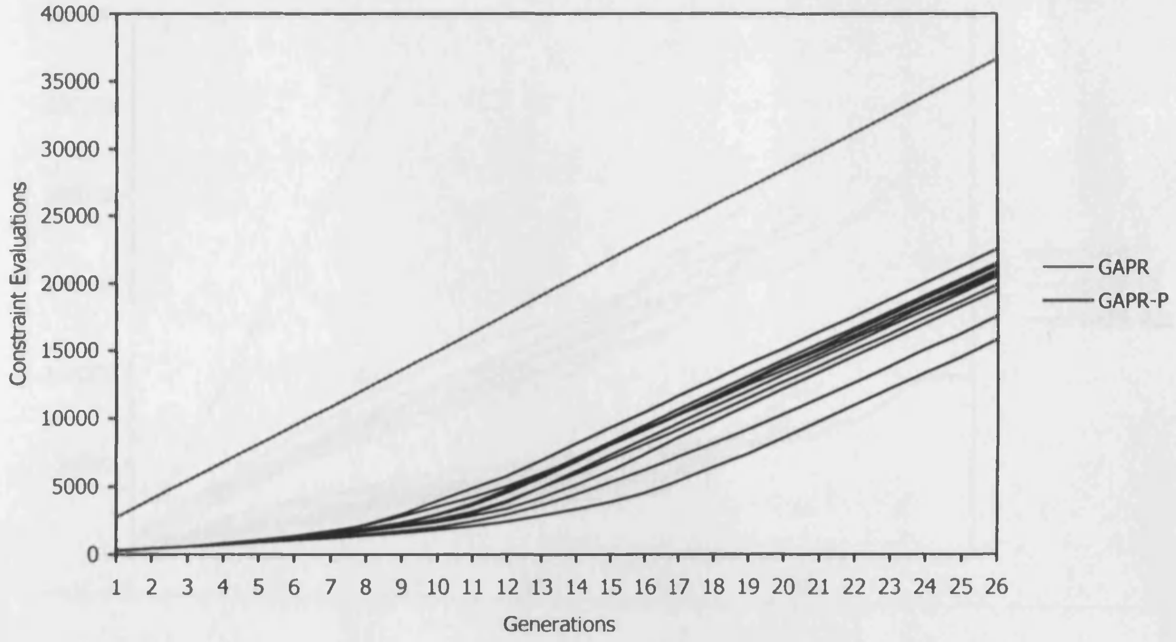

Figure 3-8 Increase in the total number of constraint evaluations over successive generations for all runs of GAPR and GAPR-P before the worst run in GAPR-P discovers a feasible solution (on generation 26). Since all individuals in GAPR undergo full constraint evaluations, its total number of evaluations forms the upper bound from which alternative constraint handling algorithms must improve.

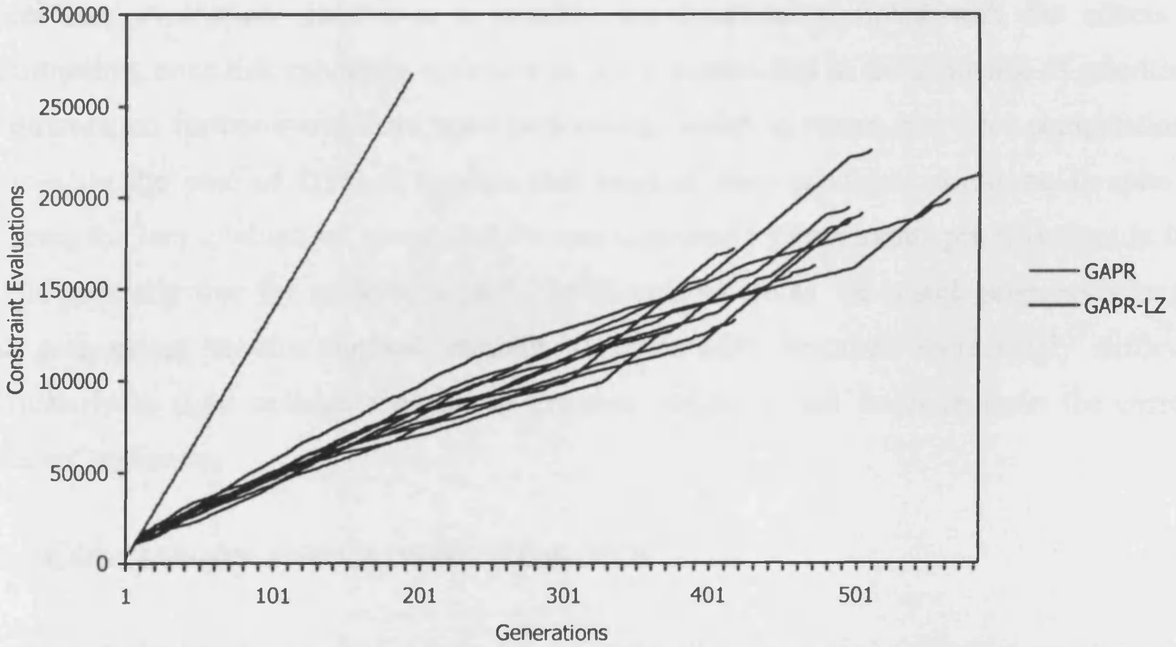

Figure 3-9 Increase in the total number of constraint evaluations over successive generations for all runs of GAPR and GAPR-LZ. The plots for GAPR-LZ are truncated at the point the search terminates.

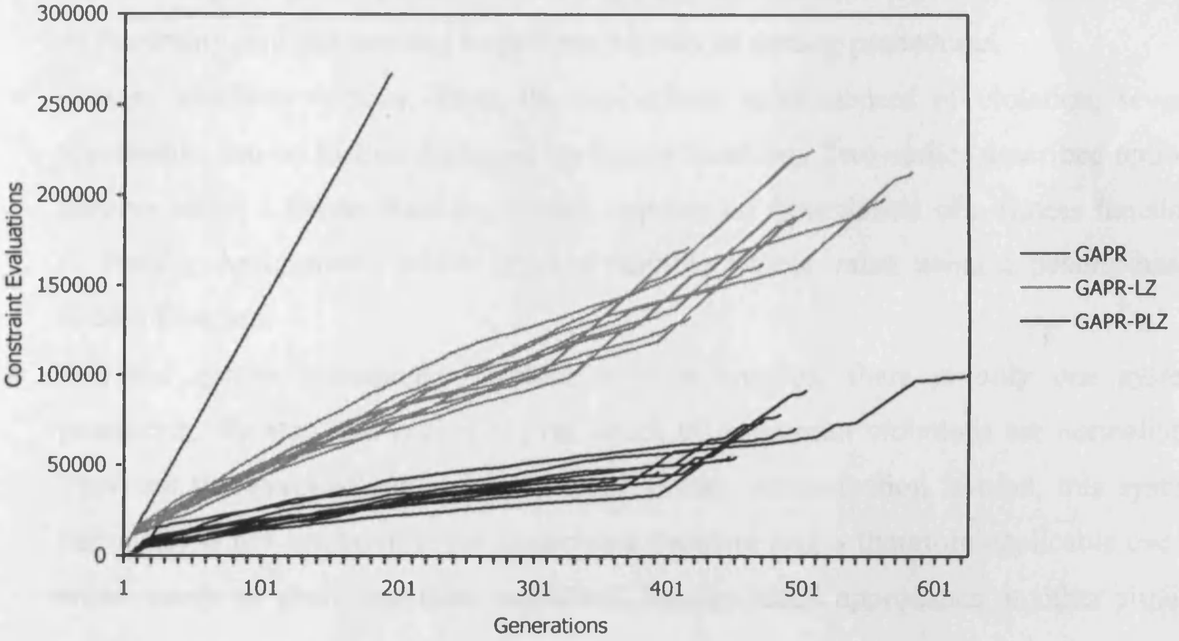

Figure 3-10 Savings in constraint evaluations through the integration of artificial prioritisation and lazy evaluations of constraints for QTC. Plots for GAPR-LZ and GAPR-PLZ are truncated at the point the search terminates due to convergence onto the true optimum.

Consider a genotype that has a cost better than the current observed feasible optimum, which forces lazy evaluation since it is a possible improvement. Coupled with the effects of prioritisation, once this candidate optimum fails any constraints in the sequence of prioritised constraints, no further evaluations need performing, which in return, provides computational savings. In the case of QTC, it appears that most of such candidate solutions, in spite of escaping the lazy evaluation 'sieve', fail the one simulated by constraints prioritisation. In fact, this is generally true for most optimisation problems where as the search progresses longer and gets closer to the optimal regions, it potentially becomes increasingly difficult, particularly in tight optimal regions, to discover solutions that improve upon the current observed optimum.

3.5 Conclusion and Closing Remarks

Most constraint handling techniques to date are primarily based on the classical penalty-based approaches. Therefore, an alternative technique drawn from inspirations in multi-objective optimisations was developed. This technique provides several useful features and can be summarised as show below:

- *Implicit ranking of feasibility*. By normalising violations of all constraints to a standard region, genotypes are ranked implicitly according to their dominance in terms of feasibility without needing to perform additional sorting procedures.

- *Fitness handling options.* From the normalised measurement of violation, several approaches can be further deployed for fitness handling. Two earlier described options involve either a Pareto-Ranking, which requires no formulation of a fitness function, or Penalty-Assignment, which gives a tangible fitness value using a penalty-based fitness function.
- *Minimal system parameters.* Unlike most techniques, there is only one system parameter, the standard region U, on which all constraint violations are normalised. Provided the Pareto-Ranking approach to fitness determination is used, this system parameter is not sensitive to the underlying problem and is therefore applicable over a wider range of problems than traditional penalty-based approaches or other similar multi-objective routines.
- *Reduced time-complex constraint evaluations.* Using either artificial prioritisation or lazy evaluation of constraint, the number of constraint evaluations can be reduced and is therefore useful for time-complex constraints where evaluations are computationally expensive relative to the deployed GA. The integration of both can even be used and has been shown empirically to provide significant advantage over any both techniques in isolation for the investigated test case.

In most general applications of GA, it is almost unnecessary to develop complex algorithms or operators when simpler traditional ones are equally successful. Before proceeding to investigate the use of uncomplicated GA in more practical engineering design and optimisation of composite laminates, the next chapter first covers the necessary groundwork for the analyses involved.

4 Analysis of Composite Laminates

This chapter provides a general overview of the design and analysis of composite laminates. It starts off by covering a brief introduction to composite materials particularly in the context of laminates commonly used in plate structures such as aircraft panels. Then an overview of available analysis methods is provided, before concentrating more on the approach that will be used as the analytical framework for subsequent work on the design and optimisation of composite plates and prismatic assemblies, which is an exact method based on the Wittrick-Williams algorithm. Finally, the algorithm is formalised into existing computer programs, namely VIPASA and VICONOPT, where the simpler model of the former is the main analysis tool featured in the optimisation studies conducted for this work.

4.1 Introduction

Composites, as the name implies, fuse several basic materials to galvanise the positive attributes evident when considered singly. The most commonly encountered composite in the field of building and structural engineering is concrete. Concrete, through a combination of cement, aggregates and other materials, is a material that is both capable of sustaining a wider load range than any of the constituent materials alone. The varying types of composite materials have allowed higher-performing lower-cost material to be used in many fields from bridges to aircraft structures.

Aircraft structures have benefited hugely from the development and advances in the design and analysis of composite materials. Modern jets are designed not only to incorporate the most efficient aerodynamic shapes for flights of all kinds (long range, supersonic, etc.), but they must also perform additional tasks and roles apart from the principles of flight. As the limits of flight are pushed further, so too must aircraft be able to sustain these new design loads as well as more severe environmental factors. The ability to engineer high strength, chemically stable and lightweight composite materials, particularly composite laminates, is such a material suitable for the use of aerospace applications. Nevertheless, resulting structures must still adhere to a serviceability of minimum maintenance over as long a period as possible while being economically viable. Minimum mass, in all applications, immediately implies a reduction in material cost. On the other hand, vehicles, especially aircraft, also benefit indirectly from reduction in mass in terms of reduced operating costs.

The design of such structures must be as close to the optimum as possible in terms of manufacturing technologies, design load capacity and requirements and production costs. Therefore, newer designs often incorporate the use of increasingly efficient materials and detailed analytical process. For example, minimisation of mass while maintaining ease of production is a common design goal in the design of composite laminates as competition increases for more efficient designs that are able to cope with more demanding performances with minimal weight penalty. As in the solution of many other engineering design problems, previously tedious time-consuming calculations are rapidly being processed by faster computers producing more efficient structures. The advent of such computing powers has also opened the doors to stochastic-based analytical tools, such as Genetic Algorithms, which previously involved impossible time scales.

4.2 Composite Laminates

As mentioned earlier, the most general definition of a composite material is evident in the use of the term *composite* itself. Composite materials incorporate two or more materials, or *constituents*, and typically aim to produce properties that cannot be achieved by the constituents alone. The use of the term 'properties' here is pretty broad, and covers not only engineering and material properties but economic cost, ease of manufacturing, etc.

Fibre-reinforced composites (FRC) are one such example of a class of composite material. The primary load-carrying constituents of FRC are the fibres[1] themselves since they are generally strong and stiff, in contrast to the binding and supporting agent, the *matrix*. The matrix[2] holds the fibres together in a continuous element serving also as a load-redistribution agent in the event of fibres failing under extreme loading conditions or environmental impacts. Typically, such composites are fully bonded together in thin layers to form *composite*

[1] Typical examples of materials used as fibres in FRC are glass, carbon and metals.

[2] Matrix materials are typically polymer-based made from long chains of organic molecules, although metal-based and other generic matrices are also possible. There are two general kinds of polymer-based matrices – thermoset and thermoplastic. Thermoset-based composites are soft and malleable whilst thermoplastic-based are harder and board-like. Typical thermoset matrices are like epoxy and polyester, whilst polyethylene is a thermoplastic example.

laminates. It should be noted that it is not a pre-requisite that the individual layers are themselves composites, but may also be singular materials such as metals. One such example of a composite is the humble plywood, which orientates the plies of wood according to the grain in order to maximise strength.

In fact, composite laminates of FRC work as illustrated in Figure 4-1, thus forming two main material axes 1-2 that are indicative of the directional nature of the fibres. This introduces directional dependency of material properties, thus becoming part of the broader class of *anisotropic* materials. At the other end of the spectrum, *isotropic* materials display no such dependency. As indicated in Figure 4-1, fibres are typically aligned continuously in a single direction, thus forming principal material axes resulting in two mutually perpendicular planes of symmetry in material properties, producing *orthotropic* behaviour. In such composites, properties such as material elastic stiffness are therefore defined along two axes – one parallel to the orientation of the fibres, and the other perpendicular to the orientation. An example of elastic properties of various FRC is as provided in Table 4-1.

FRC laminates are often described using a standard notation, the *stacking sequence*, which indicates how layers or lamina are arranged in the resulting laminate. The fibre orientation θ is clockwise positive from a reference axis, which is typically the set of axes describing the elastic properties of the laminate. The notation starts from the top of the laminate to the bottom n-th layer in the format $[\theta_1/\theta_2/.../\theta_n]$, which assumes a standard thickness t for each lamina. This is usually correct because current manufacturing practice often requires specification to be provided as integer components of a standard thickness[1]. In such a case, layers of the same orientation when grouped together are often specified as an integer value, which represents the number of layers of the standard lamina, such as $[0_2/\pm 45_2/90_2]$. It is also standard practice to produce *symmetric* laminates for reasons described later, which are those where the orientations are mirrored along the mid-plane. Symmetric laminates are usually presented in the same format with an additional subscript S to denote symmetry, such as $[0_2/\pm 45_2/90_2]_S$. Note that in this example, the laminate is also *balanced*, where all positive off-axis layers have a corresponding negative off-axis layer of identical thickness. The 'balanced and symmetric' properties of a laminate are highly desirable for reasons presented later.

[1] This requirement also has led to the research into the discrete optimisation of laminate thickness because the simplistic rounding-up technique often results in sub-optimal mass.

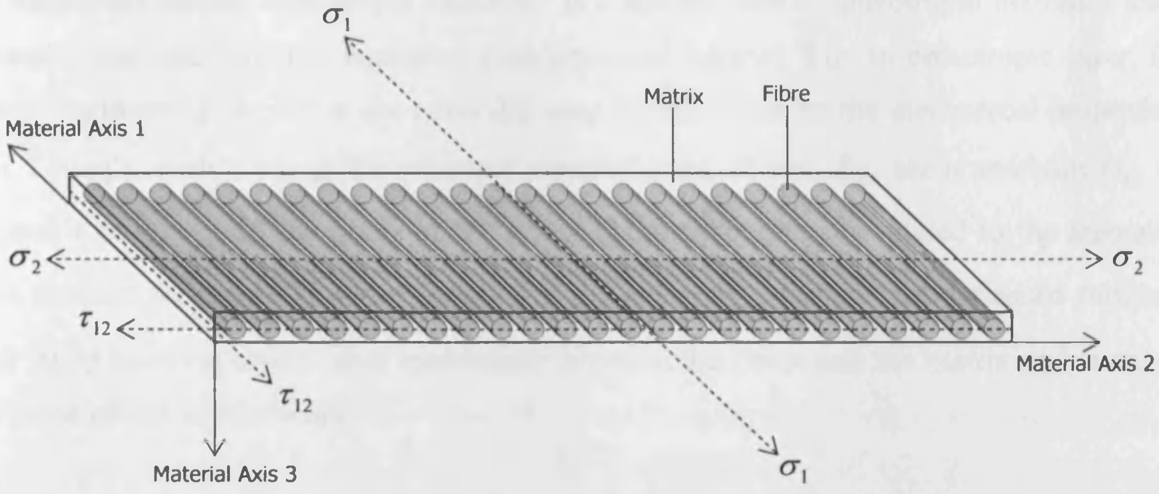

Figure 4-1 Constituents of a fibre-reinforced composite laminate and the main material axes together with stresses acting on it.

There are many issues that make the design and analysis of composite laminates complex and occasionally intractable, even when disregarding manufacturing complications. Therefore, the study of composite laminates is often divided between the macro and micro levels. The macro level, where properties of the individuals are known a priori, is the main focus of this work. The difficulty arises due to the need to address the resulting anisotropy of laminates with different fibre orientations. For any given stacking sequence of a laminate, the stress-strain relationships are required to model the various coupling mechanisms between the in-plane and out-of-plane deformations.

Material	Constituents		E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12}	V_f
	Fibre	Matrix					
AS4/3501-6	Carbon	Epoxy	142	10.3	7.2	0.27	0.63
7781/5245C	Glass	Epoxy	29.7	29.7	5.3	0.17	0.45
Kevlar 149/Epoxy	Aramid	Epoxy	87	5.5	2.2	0.34	0.60

Table 4-1 Properties of a selection of fibre-reinforced composite materials. The material properties are subscripted, where 1 indicates the primary axis (parallel to fibre) and 2 points to the secondary (perpendicular to fibre). Source: http://composite.about.com/

4.2.1 Stress-Strain Relationship of an Orthotropic Lamina

As mentioned earlier, orthotropic behaviour is a special case of anisotropic materials and is generally the case for FRC laminates (unidirectional lamina). For an orthotropic layer, four elastic engineering properties are generally used to fully describe the mechanical properties – two Young's moduli along the principal material axes, E_1 and E_2, shear modulus G_{12} and Poisson's ratio, ν_{12}, in the plane of the layer. These properties are related to the amount of fibre present, represented by the fibre volume fraction, V_f, but are not discussed further as their study involves micro-level interactions between the fibres and the matrix and is outside the scope of the current work.

The stress-strain relation for a three-dimensional anisotropic linear elastic medium, Hooke's Law, is typically expressed in the simplified matrix notation as

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} \tag{4-1}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ represents the vector of stress and strain components respectively, and $\mathbf{C}$ is known as the *material stiffness matrix*. The above relationship is reduced for a thin orthotropic material, a lamina, with principal material axes 1-2, with the assumption of *plane stress* ($\sigma_3 = 0$ and $\tau_{23} = \tau_{13} = 0$), as

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{4-2}$$

The above is often represented in the simpler matrix notation format as

$$\boldsymbol{\sigma} = \mathbf{Q}\boldsymbol{\varepsilon} \tag{4-3}$$

where $\mathbf{Q}$ is often referred to as the *reduced material stiffness matrix*. The reduced material stiffness coefficients, $Q_{ij} \in \mathbf{Q}$ where $i, j = 1,2,3$, are defined in terms of the four earlier described elastic engineering properties as

$$\begin{aligned} Q_{12} &= \frac{E_1}{1-\nu_{12}\nu_{21}}, \\ Q_{12} &= \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} = \frac{\nu_{21}E_1}{1-\nu_{12}\nu_{21}}, \\ Q_{22} &= \frac{E_2}{1-\nu_{12}\nu_{21}}, \\ Q_{33} &= G_{12} \end{aligned} \qquad \text{4-4}$$

The reciprocity relationship implies that $\nu_{12}E_2 = \nu_{21}E_1$. The stress-strain relationship of Eq. 4-3 therefore constitutes the basic state of an orthotropic lamina subject to forces in its corresponding principal material axes. It should also be noted that Eq. 4-4 can be specialised to represent isotropic materials by simply applying the relationship $\nu_{12} = \nu_{21}$, since $E_1 = E_2$. It should also be noted that for isotropic materials, the relationship $G_{12} = E_1/(2(1+\nu_{12}))$ holds.

Since orthotropic layers are generally part of a larger assembly of laminates and resulting structure, the principal material axes do not necessarily coincide with the laminate reference axes x, y and z, as shown in Figure 4-2 with the fibres orientated at an offset angle θ from the laminate reference axes. Transformation of the stress-strain relationship of Eq. 4-2 to the stress-strain relationship in the laminate reference axes will result in

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{13} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{23} \\ \overline{Q}_{13} & \overline{Q}_{23} & \overline{Q}_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \qquad \text{4-5}$$

where the coefficients of the *transformed reduced material stiffness matrix*, $\overline{Q}_{ij} \in \overline{\mathbf{Q}}$ where $i, j = 1,2,3$, are related to Q_{ij} by

$$\begin{aligned} \overline{Q}_{11} &= Q_{11}\cos^4\theta + 2(Q_{12}+2Q_{33})\cos^2\theta\sin^2\theta + Q_{22}\sin^4\theta, \\ \overline{Q}_{22} &= Q_{11}\sin^4\theta + 2(Q_{12}+2Q_{33})\sin^2\theta\cos^2\theta + Q_{22}\cos^4\theta, \\ \overline{Q}_{12} &= (Q_{11}+Q_{22}-4Q_{33})\sin^2\theta\cos^2\theta + Q_{12}(\sin^4\theta+\cos^4\theta), \\ \overline{Q}_{13} &= (Q_{11}-Q_{12}-2Q_{33})\sin\theta\cos^3\theta + (Q_{12}-Q_{22}+2Q_{33})\sin^3\theta\cos\theta, \\ \overline{Q}_{23} &= (Q_{11}-Q_{12}-2Q_{33})\sin^3\theta\cos\theta + (Q_{12}-Q_{22}+2Q_{33})\sin\theta\cos^3\theta, \\ \overline{Q}_{33} &= (Q_{11}+Q_{22}-2Q_{12}-2Q_{33})\sin^2\theta\cos^2\theta + Q_{33}(\sin^4\theta+\cos^4\theta) \end{aligned} \qquad \text{4-6}$$

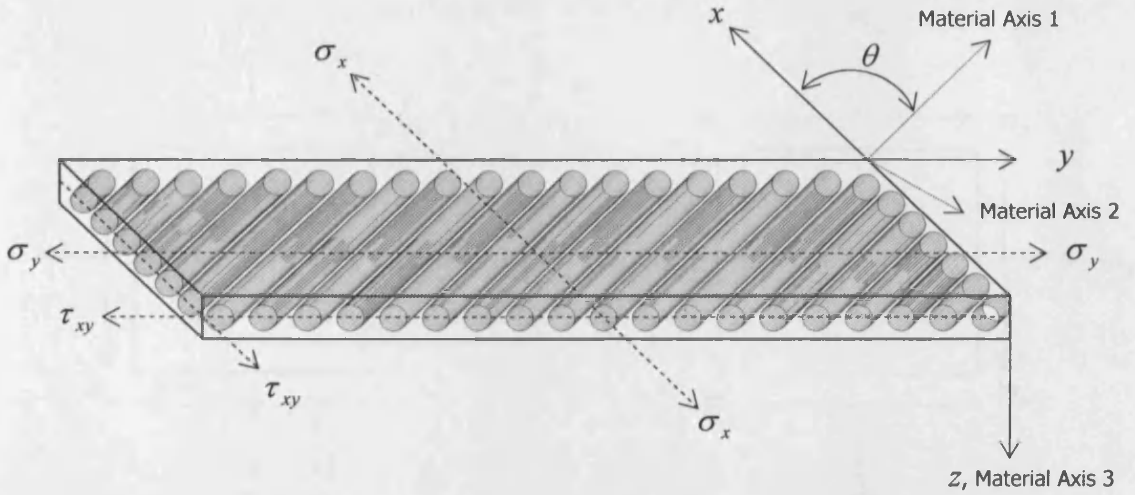

Figure 4-2 Lamina with an offset angle θ between its material axes and the governing laminate stresses with respect to the laminate reference axes.

4.2.2 Constitutive Relationship of a Laminate

In the setup of a laminate consisting of n layers, Eq. 4-5 defines the stress-strain relationship of the k-th lamina as shown in Figure 4-3. The assumption made is that layers in the laminate are bonded perfectly together with an infinitely thin bond line with continuous in-plane deformations across this line. The resultant forces and moments, or *stress couples*, acting on the laminate are obtained by performing a through-the-thickness integration of individual lamina stresses as a result of making the assumption of constant through-the-thickness strain distribution in case of in-plane loading and linear variation across the layers.

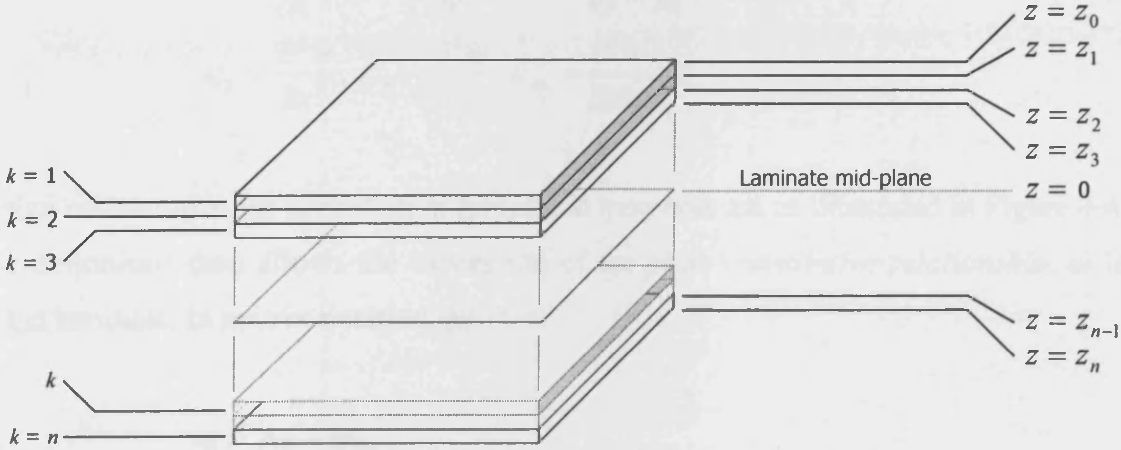

Figure 4-3 Lamina numbering system.

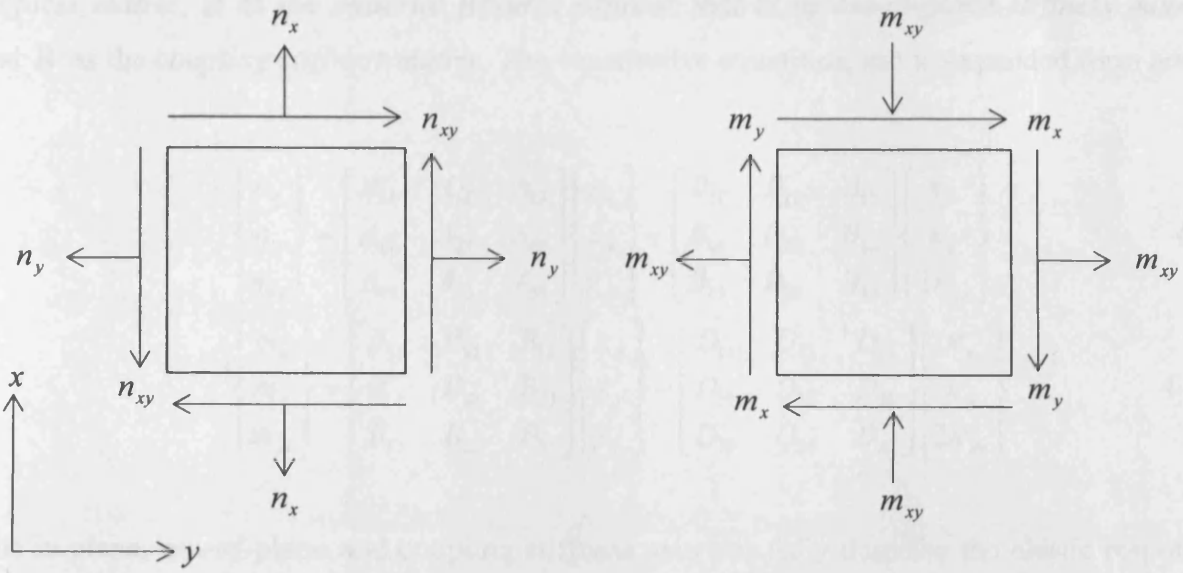

Figure 4-4 Sign conventions for membrane forces and bending moments.

The perturbation membrane forces per unit length, $\mathbf{n} = \{n_x\, n_y\, n_{xy}\}$, and perturbation bending and twisting moments per unit length, $\mathbf{m} = \{m_x\, m_y\, m_{xy}\}$, can be expressed in terms of the perturbation membrane strains, $\boldsymbol{\varepsilon} = \{\varepsilon_x\, \varepsilon_y\, \gamma_{xy}\}$, and perturbation curvatures and twist, $\boldsymbol{\kappa} = \{\kappa_x\, \kappa_y\, \kappa_{xy}\}$. These strains, curvatures and twist are related to the displacements u, v and w, parallel to the laminate reference axes x, y and z as

$$\begin{gathered}\varepsilon_x = \frac{\partial u}{\partial x},\quad \varepsilon_y = \frac{\partial v}{\partial y},\quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x},\\ \kappa_x = \frac{\partial^2 w}{\partial x^2},\ \kappa_y = \frac{\partial^2 w}{\partial y^2},\ \kappa_{xy} = \frac{\partial^2 w}{\partial x \partial y}\end{gathered} \qquad 4\text{-}7$$

The sign conventions for perturbation forces and moments are as illustrated in Figure 4-4. The above definitions then allows the expression of the *plate constitutive relationship*, or in this case the laminate, in matrix notation, as

$$\begin{aligned}\mathbf{n} &= \mathbf{A}\boldsymbol{\varepsilon} + \mathbf{B}\boldsymbol{\kappa},\\ \mathbf{m} &= \mathbf{B}\boldsymbol{\varepsilon} - \mathbf{D}\boldsymbol{\kappa}\end{aligned} \qquad 4\text{-}8$$

The matrix **A** is often referred to as the *extensional material stiffness matrix* or *in-plane* *stiffness matrix*, **D** as the *material flexural stiffness matrix* or *out-of-plane stiffness matrix*, and **B** as the *coupling stiffness matrix*. The constitutive equations, in the expanded form are

$$\begin{Bmatrix} n_x \\ n_y \\ n_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{13} & A_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ 2\kappa_{xy} \end{Bmatrix} \qquad 4\text{-}9$$

$$\begin{Bmatrix} m_x \\ m_y \\ m_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{13} & B_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ 2\kappa_{xy} \end{Bmatrix} \qquad 4\text{-}10$$

The in-plane, out-of-plane and coupling stiffness matrices fully describe the elastic response characteristics of a laminate when subjected to in-plane and bending loads, where the coefficients of the matrices, as a result of through-the-thickness integration of n layers in the laminate, are

$$\begin{aligned} A_{ij} &= \sum_{k=1}^{n} \left(\overline{Q}_{ij}\right)_k \left(z_k - z_{k-1}\right), \\ B_{ij} &= \sum_{k=1}^{n} \left(\overline{Q}_{ij}\right)_k \left(z_k^2 - z_{k-1}^2\right), \\ D_{ij} &= \sum_{k=1}^{n} \left(\overline{Q}_{ij}\right)_k \left(z_k^3 - z_{k-1}^3\right) \end{aligned} \qquad \text{where } i,j = 1,2,3 \qquad 4\text{-}11$$

The tradition of associating stiffness properties of a laminate with quantities modelled on classical elastic material properties is not possible for general laminates due to complex couplings between different deformation modes. The classical plate theory (CPT) is suitable only for thin plates with breadth-to-thickness ratio of 10 or more. Otherwise, analysis for thicker laminates is more accurate with the shear deformation plate theory (SDPT) (Cohen, 1979).

The *extension-twist coupling* terms B_{13} and B_{23} cause twisting deformation under in-plane loads even without twisting action. This is caused by off-axis layers that are not symmetric with respect to the laminate mid-plane. When laminates are restricted to symmetric sequences, the entire coupling stiffness matrix vanishes and thus eliminates the coupling between in-plane and out-of-plane responses. However, this still does not display classical material behaviour. The presence of non-zero *bending-twisting coupling* terms D_{13} and D_{23} will couple the moment resultants m_x and m_y with the twisting curvature κ_{xy}. These terms always exist in laminates with layers that have off-axis fibre orientations. Similarly, non-zero *shear-extension coupling* terms A_{13} and A_{23} induces shearing deformation under in-plane stress resultants n_x and n_y. Unlike bending-twisting coupling, this effect can be eliminated from the laminate by enforcing the rule whereby a layer with off-axis fibre orientations is balanced by an equivalent offset layer. In other words, a layer with positive off-axis fibre orientation is balanced by another layer of equal thickness with negative off-axis orientation and vice versa. Although there is no restriction needing such layers to be adjacent, doing so will result in smaller bending-twisting terms. Thus, the balanced and symmetric rules when applied will eliminate certain coupling behaviours and are therefore often included as constraints in the design of laminates. This behaviour is an important assumption and requirement in the analytical model used for design optimisation problems here.

4.3 Prismatic Plate Assemblies

Most practical uses of composite laminates are in complex geometrical shapes and forms. However, reliable analyses are available for the accurate modelling of structures that closely resemble flat rectangular prismatic plate assemblies. Prismatic plate assemblies here imply structures with panels of longitudinally invariant cross-sections. It is worth pointing out that such assemblies are not restricted to isotropic plates typically found in traditional plate construction, but includes the use of anisotropic panels as well such as composite laminates.

Different analytical models obviously make different assumptions of the macro-behaviour of the panels, but they are typically used to model the structure and find quantities that govern two commonly studied behaviours of prismatic thin-walled structures – buckling and vibration. This work concentrates on optimisation of assemblies with buckling stability and hence will be covered in greater detail than vibration. Two classes of analytical models are available for the analysis of prismatic plate assemblies – approximate and exact methods.

Two commonly used approximate methods are the finite element method (FEM) and the finite strip method (FSM). These two methods are commonly used in the analysis of structures such as floor slabs and walls. In the case of prismatic plate assemblies, they are used mainly to find the quantities associated with buckling and vibration phenomena – critical buckling loads and natural frequencies of non-damped free vibration.

The basis of FEM is to model a continuous structure using a discrete representation with a finite number of degrees of freedom. In other words, the structure is modelled by a geometrically similar representation divided into simpler and smaller discrete regions than the resulting element, thus producing a mesh-like model. Equilibrium conditions are then applied to each element producing a (typically large) system of simultaneous equations. One approximation in FEM is the interpolation between the resulting nodes to find displacements. Another approximation is that mass can be represented on nodal points by lumping surrounding mass together instead of a distribution across the continuous structure.

Accuracy of the FEM model is improved by using finer meshes. However, this cannot be *practically* increased indefinitely due to the limit determined by a convergence check performed by all FEM analysis and the possibility of numerical instability. Beyond this practical upper limit, accuracy stays virtually the same while computational overhead increases. FSM is an alternative to FEM, which reduces computational effort and costs but maintaining the accuracy provided by FEM.

An alternative approach allows the governing differential equations of the constituent members of the structure to be solved exactly by mathematical manipulation. The approach is based on the exact finite strip method introduced by Wittrick (1968a, 1968b). As in FEM and FSM methods, member stiffness matrices are combined into a global stiffness matrix, $\mathbf{K}$, representing the whole structure that relates a vector of perturbation forces $\mathbf{P}$ to corresponding displacements $\mathbf{D}$ through

$$\mathbf{KD} = \mathbf{P} \tag{4-12}$$

Unlike approximate methods like FEM, no approximations are made in representing the continuous structure because it involves an infinite number of degrees of freedom with a finite number of nodal freedoms, hence the 'exactness'. The eigenvalues of the structure, the critical buckling load factors or frequencies depending on the aim of the analysis, can be found by solving Eq. 4-12 when $\mathbf{P} = \mathbf{0}$, i.e.

$$\mathbf{KD} = \mathbf{0}$$

The penalty of using this exact approach, though, is that the elements of $\mathbf{K}$ becomes complicated transcendental functions of the corresponding load factor F or frequency ω. The high nonlinearity of $\mathbf{K}$ then implies that standard methods traditionally deployed to solve linear eigenvalue problems cannot be used. Although eigenvalues of the problem are found at points where $|\mathbf{K}| = 0$ and $\mathbf{D} = \mathbf{0}$, the irregularity of the relationship between $|\mathbf{K}|$ and F or ω

makes it impossible to detect the full range of eigenvalues with certainty. Fortunately, an algorithm developed by Wittrick and Williams (1971b) is able to guarantee the detection of all eigenvalues to any required accuracy. The following sub-sections covers the assumptions leading to the expedient use of this algorithm – the *Wittrick-Williams algorithm*. The algorithm itself will be covered in greater detail when the buckling phenomenon of prismatic plate assemblies is discussed.

4.3.1 Force-Displacement Relationship of a Component Plate

In CPT, four displacement components are considered, where the three displacements, u, v and w, are parallel along the plate reference axes x, y and z, and ψ is the rotation about the longitudinal x axis. These perturbation displacements are considered at a point (x, y) on the mid-plane of the plate. Figure 4-5 shows a component plate with the main reference axes, subjected to a basic force system. The force system comprises a uniform longitudinal compressive force per unit width, N_L, a uniform transverse compressive force per unit length, N_T and a uniform shear force per unit breadth, N_S. These are the forces that give rise to buckling instability in a buckling problem. On the other hand, a vibration analysis will regard them as forces that define the datum state of the structure about which small vibrations occur.

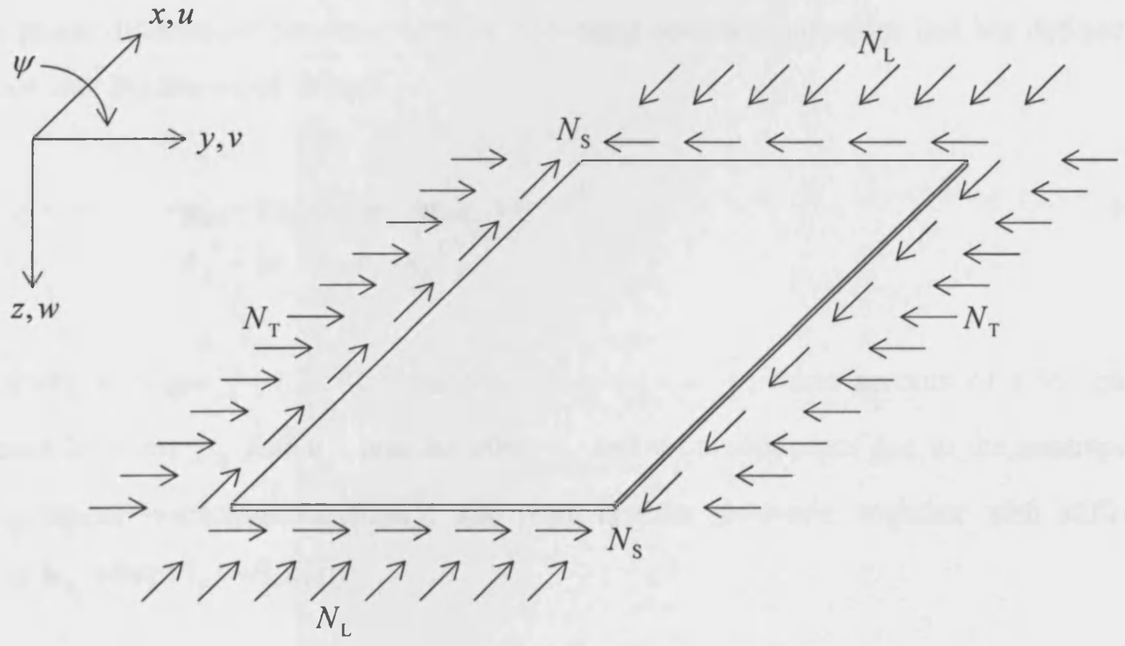

Figure 4-5 Basic force and axis system for a component plate in an assembly.

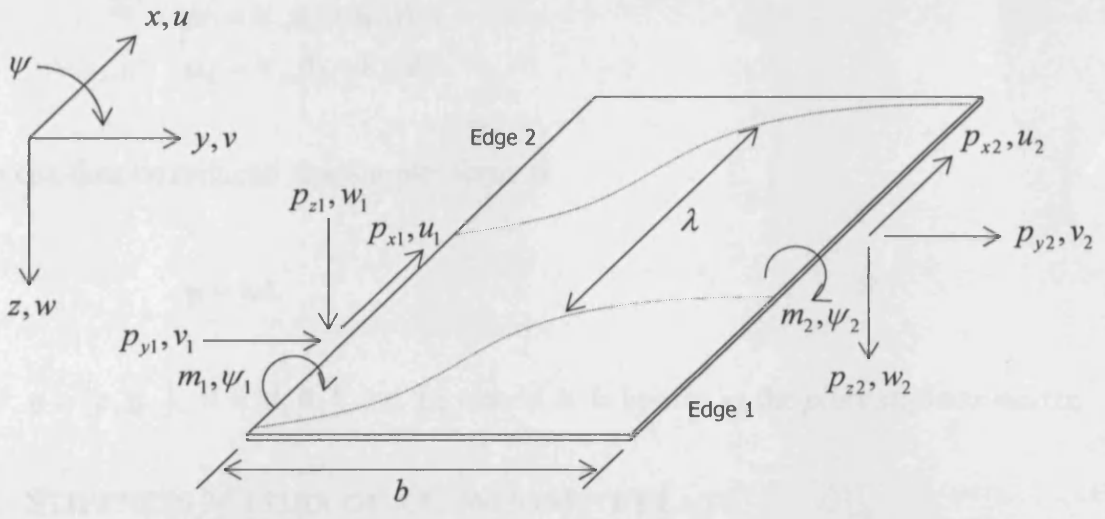

Figure 4-6 Perturbation edge forces, displacements and nodal lines of a component plate in an assembly. All edge forces and displacements are to be multiplied by exp(iπx/λ)cos(2πnt). The complex quantities allow for spatial phase differences between the edges.

The force system produces perturbation edge forces and moments during buckling or vibration, as illustrated in Figure 4-6, which vary in a sinusoidal mode in the longitudinal direction. Also illustrated in Figure 4-6, these perturbation edge forces and resulting displacements are to be multiplied by a factor of $\exp(i\pi x/\lambda)\cos 2\pi nt$, where λ is the half-wavelength of the mode, n is the frequency and t is time. In a buckling problem, $n = 0$. The magnitudes of the perturbation edge forces and displacements allows for the possibility of spatial phase differences between them by becoming complex quantities and are defined by the force and displacement vectors

$$\begin{aligned}\mathbf{p}_j &= \{m_j \ p_{zj} \ p_{yj} \ ip_{xj}\} \\ \mathbf{d}_j &= \{\psi_j \ w_j \ v_j \ iu_j\}\end{aligned} \qquad 4\text{-}13$$

respectively at edges $j = 1,2$. The complex factor i $(= \sqrt{-1})$ takes account of a 90° phase difference between p_{xj} and u_j, and the other $\mathbf{p}_j$ and $\mathbf{d}_j$ components due to the assumption of longitudinal sinusoidal variation[1]. The plate is then governed, together with stiffness matrices $\mathbf{k}_{ij}$ where $i, j = 1,2$, by

[1] This occurs even in an isotropic plate when no shear force is applied.

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{k}_{11}\mathbf{d}_1 + \mathbf{k}_{12}\mathbf{d}_2, \\ \mathbf{p}_2 &= \mathbf{k}_{21}\mathbf{d}_1 + \mathbf{k}_{22}\mathbf{d}_2\end{aligned} \qquad \text{4-14}$$

which can then be reduced to a simpler form as

$$\mathbf{p} = \mathbf{kd}, \qquad \text{4-15}$$

where $\mathbf{p} = \{\mathbf{p}_1\, \mathbf{p}_2\}$, $\mathbf{d} = \{\mathbf{d}_1\, \mathbf{d}_2\}$ and the matrix $\mathbf{k}$ is known as the *plate stiffness matrix.*

4.3.2 Stiffness Matrix of a Component Plate

Recall again the laminate constitutive relationship as shown in Eq. 4-9 and Eq. 4-10. The use of Wittrick's exact finite strip solution assumes a simpler behaviour of in-plane and out-of-plane responses in the plate constitutive relationship, given as

$$\begin{Bmatrix} n_x \\ n_y \\ n_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix} \qquad \text{4-16}$$

$$\begin{Bmatrix} m_x \\ m_y \\ m_{xy} \end{Bmatrix} = - \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ 2\kappa_{xy} \end{Bmatrix} \qquad \text{4-17}$$

The above relationship defines both the in-plane and out-of-plane elastic properties of a component plate in the overall prismatic plate assembly.

First, consider the fact that the coupling stiffness matrix $\mathbf{B}$ has been eliminated. This implies that as far as the membrane deformations are concerned, the bending forces in the component plate produce no membrane strains. Although it is possible for the general class of plates like FRC laminates to violate this, full decoupling between the in-plane and out-of-plane deformations can be forced by using the symmetric rule about the mid-plane as described earlier.

Next, it should also be noted that an assumption is further made with regards to the shear-extension coupling terms where $A_{13} = A_{23} = 0$. In other words, this assumes that the component plate displays only orthotropic properties along the principal elastic axes with regards to membrane deformations. Yet again, plates like FRC laminates can violate this assumption. However, a wide class of such composites do display orthotropic behaviour when they are fabricated using the balanced rule previously described. Nevertheless, such plates still exhibit full anisotropic bending properties and this is accounted for by the fully populated out-of-plane stiffness matrix $\mathbf{D}$.

The above two assumptions were made to allow for the derivation of explicit analytical expressions for the elements in the plate stiffness matrix $\mathbf{k}$. It is worthwhile to note that $\mathbf{k}$ is in general complex and Hermitian. The only special case where $\mathbf{k}$ is real and symmetrical is when the corresponding component plate displays full orthotropic bending behaviour and carries no shear loading, i.e., when $D_{13} = D_{23} = 0$ and $N_S = 0$. For FRC laminates, this is only achievable if all layers have fibres aligned along the plate's xy reference axes, known as *cross-ply laminates*, which enforces the restriction that fibre orientations can only be 0°, 90° or a combination of both.

The stiffness matrices $\mathbf{k}_{ij} \in \mathbf{k}$ can be expanded, with in-plane stiffness coefficients s_{NN}, s_{TT}, s_{NN}, f_{NN}, f_{TT} and f_{NT}, and out-of-plane coefficients s_{MM}, s_{QQ}, s_{MQ}, f_{MM}, f_{QQ} and f_{MQ}, as

$$\mathbf{k}_{11} = \begin{bmatrix} s_{MM} & -s_{MQ} & 0 & 0 \\ -\bar{s}_{MQ} & s_{QQ} & 0 & 0 \\ 0 & 0 & s_{NN} & -s_{NT} \\ 0 & 0 & -s_{NT} & s_{TT} \end{bmatrix}$$

$$\mathbf{k}_{22} = \begin{bmatrix} s_{MM} & \bar{s}_{MQ} & 0 & 0 \\ s_{MQ} & s_{QQ} & 0 & 0 \\ 0 & 0 & s_{NN} & s_{NT} \\ 0 & 0 & s_{NT} & s_{TT} \end{bmatrix} \qquad 4\text{-}18$$

$$\mathbf{k}_{12} = \begin{bmatrix} f_{MM} & f_{MQ} & 0 & 0 \\ -f_{MQ} & -f_{QQ} & 0 & 0 \\ 0 & 0 & -f_{NN} & -f_{NT} \\ 0 & 0 & f_{NT} & f_{TT} \end{bmatrix} = \bar{\mathbf{k}}_{21}^{\mathrm{T}}$$

where a bar denotes a complex conjugate, and superscript T denotes a matrix transpose. All coefficients are always real, except s_{MQ}, f_{MM}, f_{QQ} and f_{MQ}, which are in general complex, but become real too if $D_{13} = D_{23} = N_S = 0$, i.e., for isotropic or orthotropic plates without shear loading. The stiffness coefficients for anisotropic and isotropic plate under combined loading are complicated transcendental functions, and explicit analytic expressions for them can be found in the literature, e.g., see Wittrick and Williams (1974). A simplified version of the expressions for longitudinally compressed prismatic isotropic plate assemblies by Williams and Wright (1978) can be a useful introductory material, because the resulting stiffness matrix is real and symmetric.

It is worth pointing out that so far, the edge forces and displacements of a component plate have been referred to the middle lines of the two longitudinal edges and to a set of plate reference axes x, y and z, which are local to the plate. To construct the overall stiffness matrix of the resulting structure, transformation of the stiffness matrices of component plates must be performed to reflect the global edge references and an overall global reference axes.

4.3.3 Transformation of the Plate Stiffness Matrix

For reasons pointed out earlier, individual plate stiffness matrices must be updated to reflect a global system of reference before the overall stiffness matrix **K** can be constructed. Two types of transformation provide for this – eccentricity and rotational. Eccentricity transformation allows plates to be connected along a common longitudinal line at their respective edges, while rotational transformation caters for component plates with plate reference axes that do not coincide with the global reference axes.

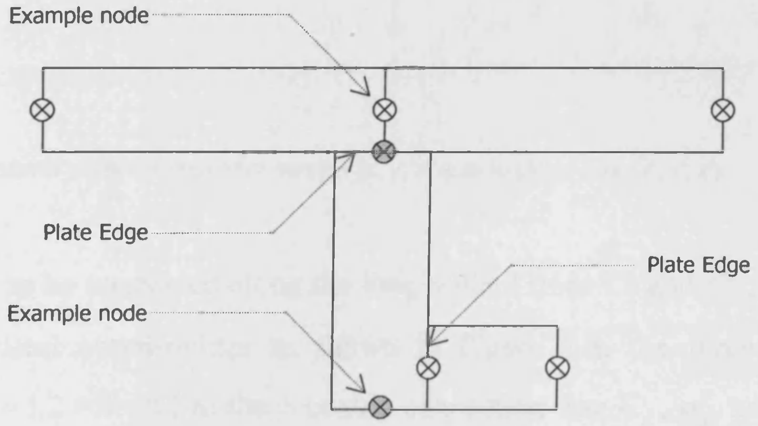

Figure 4-7 Connection modelling between component plates.

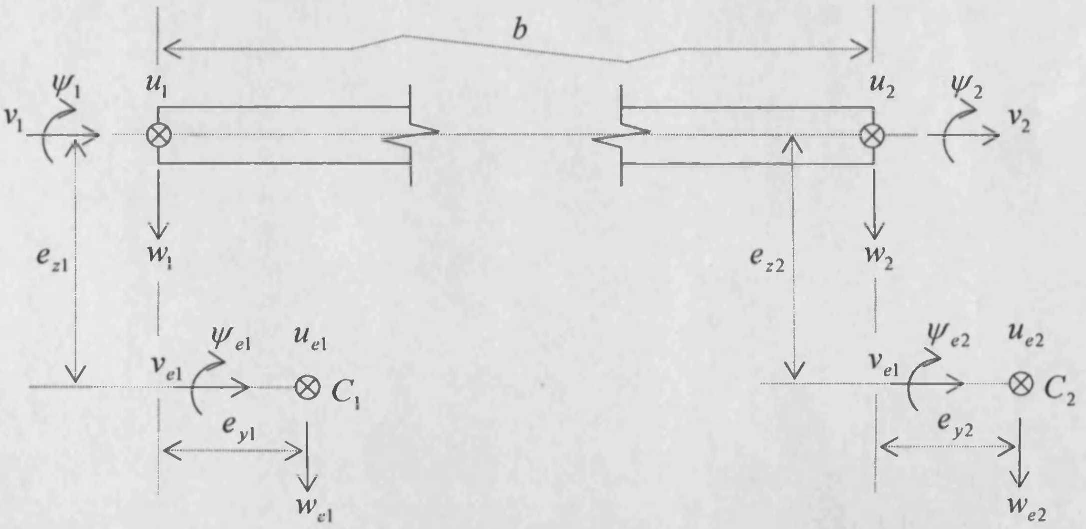

Figure 4-8 Plate eccentric edge connections together with corresponding dimensions and displacements of a component plate.

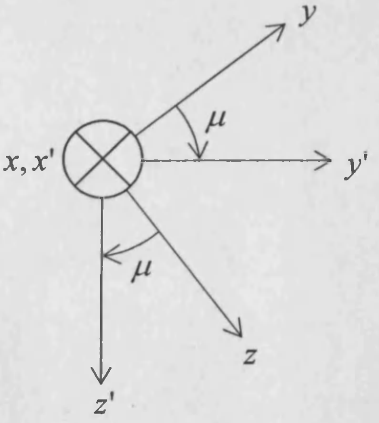

Figure 4-9 Rotational transformation from member (x, y, z) axes to global axes (x', y', z').

For a plate that is to be connected along the longitudinal lines C_1 and C_2, and corresponding transverse and lateral eccentricities as shown in Figure 4-8, the force and displacement vectors at edge $j = 1{,}2$ referred to the eccentric connection line C_j, $\mathbf{p}_{ej}$ and $\mathbf{d}_{ej}$ respectively, is given as

$$\begin{aligned} \mathbf{p}_{ej} &= \mathbf{T}_{ej}^{\mathrm{T}}\mathbf{p}_j, \\ \mathbf{d}_j &= \mathbf{T}_{ej}\mathbf{d}_{ej} \end{aligned} \qquad 4\text{-}19$$

where the eccentricity transformation matrix $\mathbf{T}_{ej}$ is

$$\mathbf{T}_{ej} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -e_{yj} & 1 & 0 & 0 \\ e_{zj} & 0 & 1 & 0 \\ 0 & -\frac{\pi e_{zj}}{\lambda} & -\frac{\pi e_{yj}}{\lambda} & 1 \end{bmatrix} \qquad 4\text{-}20$$

and $\mathbf{p}_j$ and $\mathbf{d}_j$ are the original force and displacement vectors of the component plate with respect to the plate reference axes. It should be pointed out that $\mathbf{T}_{ej}$ is not exact. Nevertheless, it is sufficiently accurate provided that eccentricities are small compared to the half-wavelength λ, which is typically the case for prismatic plate assemblies.

The force and displacement vectors $\mathbf{p}_{ej}$ and $\mathbf{d}_{ej}$ are then further transformed to reflect also the global reference axes x', y' and z' to produce corresponding vectors $\mathbf{p'}_j$ and $\mathbf{d'}_j$. The resulting relationship is formed by applying a rotational transformation matrix $\mathbf{T}_r$, where a rotation of μ about the longitudinal axis is required as shown in Figure 4-9, giving

$$\begin{aligned} \mathbf{p'}_j &= \mathbf{T}_r^{\mathrm{T}}\mathbf{p}_{ej}, \\ \mathbf{d}_{ej} &= \mathbf{T}_r\mathbf{d'}_j \end{aligned} \qquad 4\text{-}21$$

where the rotational transformation matrix $\mathbf{T}_r$ is

$$\mathbf{T}_r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\mu & \sin\mu & 0 \\ 0 & -\sin\mu & \cos\mu & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 4\text{-}22$$

Relationships from Eq. 4-19 and Eq. 4-21 then allows Eq. 4-14 to be updated to reflect these transformations to produce, for individual component plates, the new relationships

$$\begin{aligned} \mathbf{p'}_1 &= \mathbf{k'}_{11}\,\mathbf{d'}_1 + \mathbf{k'}_{12}\,\mathbf{d'}_2, \\ \mathbf{p'}_2 &= \mathbf{k'}_{21}\,\mathbf{d'}_1 + \mathbf{k'}_{22}\,\mathbf{d'}_2 \end{aligned} \qquad 4\text{-}23$$

where the transformed stiffness matrices $\mathbf{k}'_{ij}$, where $i,j=1,2$, are

$$\mathbf{k}'_{ij}=\mathbf{T}_r^{\mathrm{T}}\mathbf{T}_{ei}^{\mathrm{T}}\mathbf{k}_{ij}\mathbf{T}_{ej}\mathbf{T}_r \qquad 4\text{-}24$$

It should be noted that since $\mathbf{k}_{ij}=\overline{\mathbf{k}}_{ij}^{\mathrm{T}}$, it follows that $\mathbf{k}'_{ij}=\overline{\mathbf{k}}_{ij}^{\mathrm{T}}$, which then implies that the transformed plate stiffness matrix $\mathbf{k}'$, to which $\mathbf{k}$ from Eq. 4-15 transforms, is also generally Hermitian. Derivations for the transformation matrices can be found in Wittrick and Williams (1974).

With the stiffness matrices of all component plates transformed to reflect global reference axes and eccentric connection lines, the assembly of an overall stiffness matrix $\mathbf{K}$ for the structure can be done using standard techniques and solved for eigenvalues as described below.

4.3.4 Buckling and the Eigenvalue Problem

The overall stiffness matrix $\mathbf{K}$ of a prismatic plate structure relates the forces vector $\mathbf{P}$ and displacement vector $\mathbf{D}$. The force vector $\mathbf{P}\exp(i\pi x/\lambda)\cos 2\pi nt$ is the set of perturbation forces at the longitudinal connection line and external plate edges, which are distributed in a sinusoidal mode along the longitudinal axis with half-wavelength λ and $\mathbf{D}\exp(i\pi x/\lambda)\cos 2\pi nt$ being the corresponding vector of displacements.

Regardless of whether the problem in consideration is buckling ($n=0$) or vibration, the eigenvalue problem corresponds to $\mathbf{P}=\mathbf{0}$ such that solving for $\mathbf{KD}=\mathbf{0}$ gives the critical load buckling factors or frequencies for a specified λ. The non-trivial solution ($\mathbf{D}\neq\mathbf{0}$) is obtained only if $|\mathbf{K}|=0$. In fact, there is a complete and simple relationship between the buckling problem and the problem of the natural frequency of vibration of such structures (Wittrick and Williams, 1974). For a given value of half-wavelength λ, the i-th buckling stress σ_i and i-th natural frequency ω_i are related by the formula

$$\sigma_i=4\rho\lambda^2\omega_i^2 \qquad 4\text{-}25$$

where ρ is the density of the material. Furthermore, the modes of buckling and vibration are identical. The relationship therefore summarises to a simple fact that for every buckling problem, there is a corresponding vibration problem with a particular natural frequency.

In a buckling problem, interest centres mainly on the lowest eigenvalue or critical buckling load factor, which then becomes the governing critical buckling load of the overall structure. However, there may well be situations in which discovering other eigenvalues close to the lowest is important too. Nevertheless, several buckling modes can occur for any given structure although the tendency for any one to occur depends on many other factors such as the geometry, stiffness, loading conditions, etc.

4.3.4.1 WITTRICK-WILLIAMS ALGORITHM

It has been mentioned earlier that the elements of $\mathbf{K}$ are complicated transcendental functions of the corresponding load factor F or frequency ω. While standard methods for solving linear eigenvalue problems are available, the behaviour of $\mathbf{K}$ implies that such methods will fail. The hope of solving the eigenvalue problem here is provided by the Wittrick-Williams algorithm, a numerical iterative technique that guarantees the detection of all eigenvalues to a required accuracy. The Wittrick-Williams algorithm first found its use in vibration problems (Wittrick and Williams, 1971a, 1971b) before later extended into the general form here that also covers buckling (Wittrick and Williams, 1973), used in the VIPASA model (Wittrick and Williams, 1974). Many modifications have since been done, the most important being arguably the VICON analysis model, now found in VICONOPT (Williams et al., 1990a), that assumes a plate deflection mode that can be expressed as a Fourier series rather than the sinusoidal mode discussed so far (Williams and Anderson, 1983).

Since the concern of this work centres on buckling problems, the following description of the VIPASA model concentrates on finding the critical load factor F rather than the natural frequency ω. The problem of natural frequency, in vibration analysis, can be argued in the same fashion as a buckling problem by simply substituting ω for F.

The basis of the algorithm involves calculating the number of eigenvalues J (critical load factor F) that lie between zero, i.e., when $F = 0$, and some trial eigenvalue $F > 0$. This then implies that a change in J, ΔJ, between any two trial values gives the number of eigenvalues lying between these trial values.

For example, consider a general buckling problem with two trial values F_1 and F_2 such that $F_2 > F_1 > 0$. If J_1 represents the number of eigenvalues when $F = F_1$ and J_2 similarly represents the number of eigenvalues when $F = F_2$, then it can be concluded that the number of critical load factors lying between F_1 and F_2 is therefore $\Delta J = J_2 - J_1$. The interesting case here is when $\Delta J = 1$ has been established, which then allows determination of the appropriate m-th eigenvalue with certainty to any required accuracy. To converge onto the required eigenvalue, the iterative procedure of bisection (interval halving) can be used although more sophisticated techniques to speed up the convergence process are also possible, such as the multiple determinant parabolic interpolation method (Williams and Kennedy, 1988; Kennedy and Williams, 1992).

The Wittrick-Williams algorithm is formalised as

$$J = J_0 + s[\mathbf{K}] \tag{4-26}$$

where, as before, J is the number of positive eigenvalues less than some chosen positive trial eigenvalue F^*, J_0 is the number of eigenvalues less that F^* if the nodes of the structure are fully constrained, $\mathbf{K}$ is the overall stiffness matrix of the structure evaluated at the trial value F^*, and $s[\mathbf{K}]$ is known as the *sign count* of the matrix $\mathbf{K}$. Clearly, the overall stiffness matrix $\mathbf{K}$ is obtained by constructing the individual component member stiffness matrix $\mathbf{k}'$ as described earlier.

The sign count $s[\mathbf{K}]$ is derived from the Sturm sequence property of the generally Hermitian overall stiffness matrix $\mathbf{K}$. An easy method exists for computing this value. First, $\mathbf{K}$ is converted into its upper triangular form, $\mathbf{K}^{\Delta}$, by applying the usual form of Gaussian elimination without scaling or row interchanges and making all elements below the pivot zero. Then, $s[\mathbf{K}]$ is simply the number of negative elements on the leading diagonal of the upper-triangulated matrix $\mathbf{K}^{\Delta}$.

A component requiring more detailed explanation is the calculation of J_0. In actual fact, J_0 is the number of eigenvalues which would have been exceeded by F^* if all nodes of the structure were fully clamped, i.e., $\mathbf{D} = \mathbf{0}$. The structure therefore degenerates into a set of m individual component members (plates) with their corresponding longitudinal edges fully clamped. Thus, J_0 is equal to the sum of the total number of positive eigenvalues of the individual component plates with their edges fully clamped, J_{m} that lie below F^*, or formally as

$$J_0 = \sum_m J_m \qquad 4\text{-}27$$

Two factors contribute directly to J_m – in-plane and out-of-plane eigenvalues of the respective members. Analytical formulae are provided by Wittrick and Williams (1974) for direct evaluation of J_m.

Computational expediency for complex repetitive structures can be achieved by using sub-structuring, which replaces repeating portions of the structure with sub-structures. This reduces the order of $\mathbf{K}$ by only including stiffness terms relating to nodes of the substructures that are connected to the overall assembly, thus saving processing time. Despite allowing unlimited nesting levels of sub-structuring, the analysis is still exact because no additional approximations are introduced. Sub-structuring modifies the J_0 term from Eq. 4-27 to

$$J_0 = \sum_m J_m + \sum_s J_s \qquad 4\text{-}28$$

which adds a second summation over all specified substructures. The term J_S represents the total number of positive eigenvalues less than the chosen positive trial eigenvalue F^*, and is calculated in a similar manner as a complete structure on its own using Eq. 4-26 except with nodes that connect to the main parent structure modelled as fully clamped. It should also be noted that J_S is recursive in the sense of allowing additional sub-structures to be connected to it. A complete description of sub-structuring is provided by Wittrick and Williams (1974).

4.3.4.2 BOUNDARY AND SUPPORT CONDITIONS

The assumption that buckling or vibration occurring in a sinusoidal mode in the longitudinal direction is important and is accurate only for a subset of plate assemblies. Nevertheless, this assumption approximates closely to other cases too up to a certain degree. Given the case that computation of eigenvalues for a specified value of half-wavelength λ can now be performed accurately and with guarantee using the Wittrick-Williams algorithm, the resulting critical buckling load factors or natural frequencies are accurate provided several prerequisites are met.

Support and nodal conditions can be modelled directly in the overall stiffness matrix $\mathbf{K}$ itself without resorting to additional formulations. Recall that the on-going analysis model allows four degrees of freedom at every node aligned along longitudinal line supports, namely u, v, w and ψ for the three displacements along the global reference axes x', y', z' and rotation about x' respectively. Displacement components at nodes or boundary supports, including internal nodes of sub-structures, can therefore be modelled as elastically or rigidly restrained by adding appropriate stiffness to the leading diagonal of $\mathbf{K}$ (or $\mathbf{K}_S$ for sub-structures) which corresponds to the displacement components. Rigid restraints can be simulated by making the corresponding stiffness substantially large.

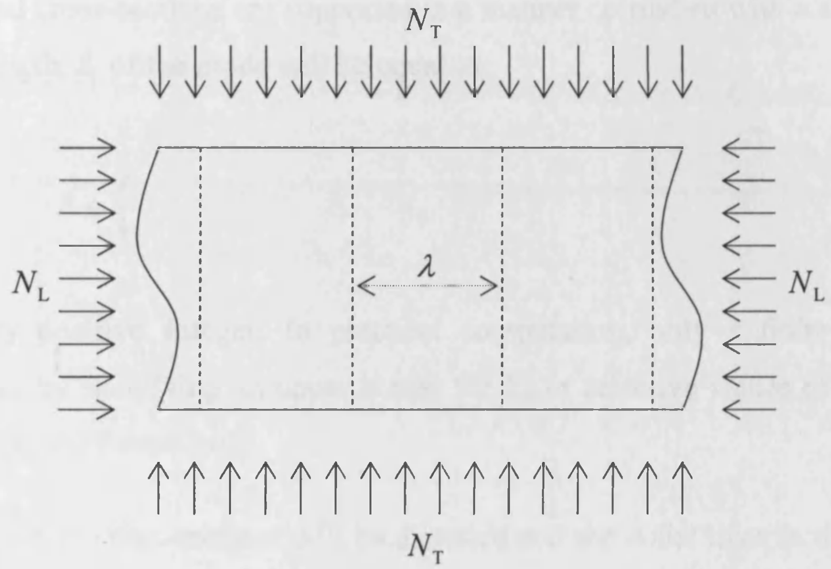

Figure 4-10 Straight nodal lines for isotropic plates or anisotropic plates without shear, compatible with simply support transverse ends.

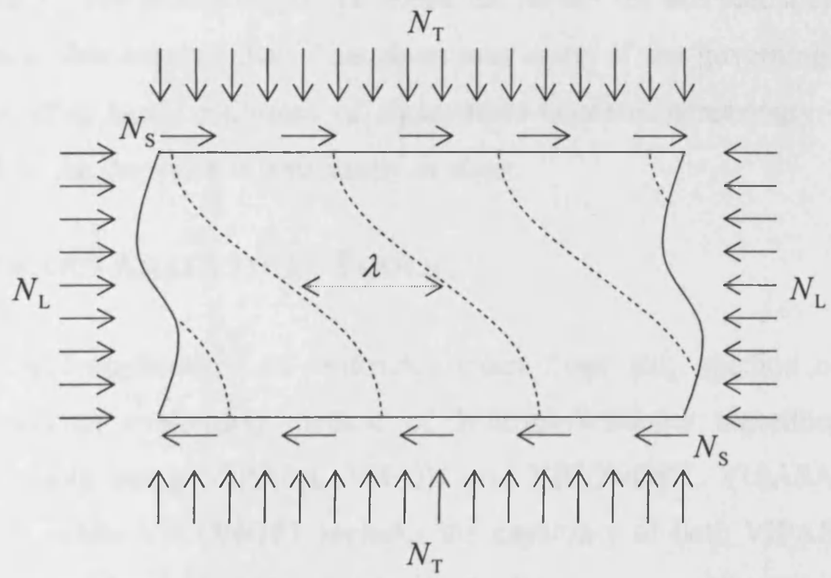

Figure 4-11 Skewed nodal lines as a result of anisotropy and/or shear loading, resulting in incompatibility with simply supported transverse ends assumption.

So far, there is no mention of supports in the transverse direction because they cannot be effectively simulated with the current model. The sinusoidal buckling or vibration mode assumption implies a deflection that is consistent with simply-supported end conditions, as shown in Figure 4-10, where lines of zero displacement are straight and perpendicular to the longitudinal panel axis and spaced λ half-wavelength apart. Although it is very likely that most plates do not conform to this behaviour, isotropic or orthotropic plates in the absence of shear loading are consistent with this model. Hence, analyses of assemblies with simply-supported end conditions are generally considered accurate. In such a structure of length ℓ, when its two end cross-sections are supported in a manner consistent with a sinusoidal mode, the half-wavelength λ of the mode will be equal to

$$\lambda = \frac{\ell}{k} \qquad 4\text{-}29$$

where k is any positive integer. In practical computation, only a finite range of λ is considered either by specifying an upper bound for k, or selective values of λ to which the relationship in Eq. 4-29 must hold.

In general, transverse cross-sections will be distorted and the nodal lines in the plates will be curved. Although the sinusoidal mode of half-wavelength λ still applies in the longitudinal direction, the nodal lines are no longer compatible with simply-supported end conditions, but are skewed as illustrated in Figure 4-11. This inconsistency results in evaluated eigenvalues as no longer accurate but estimates. Nevertheless, the results are still realistically accurate for λ that are considerable smaller than ℓ, such as may occur if the governing buckling mode is local. On the other hand, estimates of eigenvalues become increasingly conservative as λ approaches ℓ in the presence of anisotropy or shear.

4.4 Design and Analytical Tools

There are several applications of Wittrick's exact finite strip method combined with the efficient eigenvalue evaluation method of Wittrick-Williams algorithm, with the most successful arguably being VIPASA, VICON and VICONOPT. VIPASA and VICON are analytical tools, while VICONOPT includes the capability of both VIPASA and VICON as analysis options and provides additional design facilities.

4.4.1 VIPASA

The computer program VIPASA (Vibration and Instability of Plate Assemblies including Shear and Anisotropy) encompasses the analytical model for prismatic plate assemblies outlined so far and was developed by Wittrick and Williams (1974). The program provided major improvements over their two previous plate analysis programs – GASVIP (General Axial Stress Vibration and Instability Program) and VIPAL (Vibration and Instability Program with Axial Load).

GASVIP and VIPAL were used for the evaluation of eigenvalues of buckling or vibration problems of isotropic plate assemblies loaded with uniform longitudinal stress. VIPASA extended the capabilities of both these programs and catered for anisotropy, thus opening the possibility of including more sophisticated materials such as FRC laminates in prismatic plate assemblies. On top of that, more complicated loading systems were now modelled that included longitudinal, transverse and shear loading. The VIPASA analytical model assumes a sinusoidal mode of deflection along any longitudinal line and incorporates phase differences, thus accurately representing the effects of anisotropy and shear. The only other assumptions are those already inherent in the classical small deflection thin plate theory. Other similar programs present at the same time were one by Smith (1968), which includes also longitudinal beam elements, and BUCLASP2 (Viswanathan et al., 1973), which allows curved plates as well as longitudinal beam elements. Both, however, did not allow full anisotropy and shear in component plates, but BUCLASP2 included coupling effects between bending and mid-surface deformations.

There is, however, a limitation to the analytical model of VIPASA. As mentioned previously, in-plane shear loading on component plates resulted in skewed sinusoidal modes, which are not properly modelled by VIPASA, results of which are accurate only for isotropic or orthotropic plate assemblies with shear or in shorter half-wavelengths in the case of full anisotropy or presence of shear. Otherwise, produced results are conservative. With this restriction in the analytical model, VICON was developed as an alternative to a more accurate model.

4.4.2 VICON

The VICON (VIPASA with CONstraints) analytical model overcomes the conservatism of VIPASA in the case of full anisotropy or in the presence of shear load (Anderson et al., 1983).

One major improvement in VICON is the assumption that the deflection mode of a panel may be expressed as a Fourier series involving an appropriate set of half-wavelengths λ, as opposed to the sinusoidal mode of VIPASA.

Results from the VICON model provided are for an infinitely long panel with supports repeating at intervals of ℓ, so that the plate assembly and its supports form identical lengthwise bays of length ℓ. This allows an assembly of a finite length ℓ with simply supported transverse ends to be modelled to reasonable accuracy by representing the end conditions as a line of rigid point supports at $x = 0$ and restraining appropriate displacements[1].

The eigenvalues arising from the VICON analysis are obtained similarly to VIPASA with exception of an extension to the Wittrick-Williams algorithm (Williams and Anderson, 1983). A full derivation of the governing equations and extended stiffness matrix using Lagrangian multipliers can be found in Anderson et al. (1983). The VICON model was later adopted into the design and optimisation tool, VICONOPT.

4.4.3 VICONOPT

The computer program VICONOPT (VICON with OPTimisation) extends the analytical capability of VICON and provided an additional design optimisation feature (Williams et al., 1990a; Williams et al., 1990b; Butler and Williams, 1990, 1992; Williams et al., 1991; Kennedy et al., 1994).

VICONOPT builds on the two analytical models of VICON and VIPASA by the addition of a design optimisation feature. The design objective in VICONOPT is the constrained mass or cost minimisation of a prismatic plate assembly. Available design variables include plate widths, layer thicknesses and layer ply angles, which can be linked to other dimensions to become dependent variables to keep geometric consistency. The constraints are typically those particular to the design of plate assemblies and include buckling constraints provided by either the VIPASA or VICON analysis, material strength constraints defined using maximum allowable stress or strain for compression, tension and shear, or calculated using the Tsai-Wu criterion, and other configuration requirements.

[1] Since the analytical model of VICON assumes lengthwise repetition, the modelling of supports at one end of the assembly also implies the occurrence of the same support at the other end.

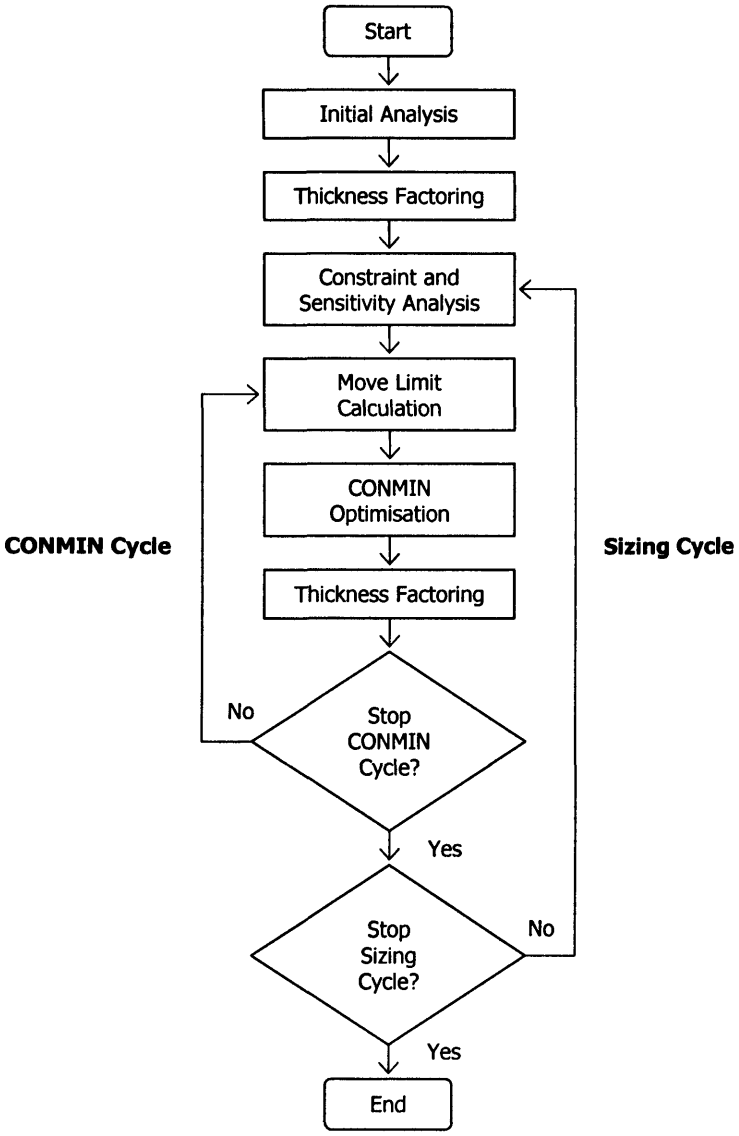

Figure 4-12 VICONOPT continuous design strategy.

The continuous design strategy in VICONOPT, upon initial analysis, enters a stabilisation step, where the thicknesses of the initial design are factored to a just feasible design. Then, it enters into an overall sizing cycle with an optimiser routine within the cycle. First, a constraint and sensitivity analysis is used to obtain gradient information about the sensitivities of eigenvalues to small changes in the design variables. Then, the upper and lower bounds of the design variables are calculated (move limit calculation) to prevent large and unreliable design moves of the optimiser. VICONOPT then proceeds to the linear CONMIN optimisation step (Vanderplaats, 1973; Vanderplaats and Moses, 1973), which is based on the method of feasible directions. Here, it converges onto a design which may or may not be stable. Hence, the cycle again enter the stabilisation step and the whole optimiser routine repeats itself until either the maximum number of user-defined cycles has been reached or the stabilised mass of the prismatic plate assembly is within a user-defined tolerance of the mass after previous stabilisation. New sizing cycles are performed if the mass has not converged or the maximum number of sizing cycles has not been reached. Figure 4-12 illustrates the VICONOPT design strategy.

4.5 Closing Remarks

The availability of accurate and even exact analytical models, such as the one outlined above, has opened up the opportunity for the design and optimisation of structures constructed from composite laminates, in particular FRC laminates. As computing power continues to increase, so too does the ability to include more design variables into the optimisation process and produce even more efficient composite laminates and assemblies. The design and optimisation of composite laminates falls broadly into two categories – micro-behaviour and macro-mechanics optimisations.

Micro-behaviour optimisation obviously deals with the interaction of the constituents, i.e., between fibres and matrix. Although approximate analytical methods are available to study such behaviours, most properties of composite laminates are still obtained empirically. Thus, optimisation at this level is often limited to experimental studies of materials.

Macro-mechanics, on the other hand, studies the mechanical responses of composite laminates on a structural level such as buckling, post-buckling and vibration. However, such behaviours in composite laminates are more complex than most traditional materials commonly used in plate structures such as metals and alloys, particularly due to the anisotropy. Analysis at this level has improved a great deal with developments of accurate and exact methods. Even if such methods are not the preferred choice, more general analytical methods such as the FEM can still be used to give reasonably accurate estimates of mechanical responses although there may be compromise in terms of computational overhead.

Finding an efficient composite laminate design subjected to macro-mechanical constraints whether for an individual plate or an assembly is highly sought after in many areas of application such as in aircraft manufacturing. Efficiency here deals not only with load carrying capacity but extends equally to weight and cost savings or any acceptable trade-off margin. Perhaps the most common design optimisation of composite laminates, and indeed the use of other materials, involves some kind of mass reduction. In most applications, mass often has an effect on performance and cost, and therefore becomes the primary objective function. However, the mass minimisation of composite laminates goes beyond the simplistic optimisation of lamina thickness.

Mass minimisation of composite laminates, in particular FRC laminates, can be affected by layer fibre orientation, physical dimensions, enforcement of balanced and symmetric rules, amongst many others. The high-dimensionality of the design optimisation problem when all these factors are considered concurrently often makes it impossible to evaluate. Coupled with computationally expensive evaluations of constraints, such as satisfaction of a design buckling load, designs of FRC laminates are often forced to be reduced to the design of a single laminate rather than a complete assembly. Nevertheless, successful studies into such high-dimensionality mass design problems have been reported by York et al. (1993) and Kennedy et al. (1999).

Although mass optimisation is a common design problem, load capacity maximisation of composite laminates is also useful particularly in cases where there is a restriction or limit in geometrical dimensions. Such an optimisation, in the case of FRC laminate, typically involves finding a configuration of lamina fibre orientations that satisfy the geometrical constraint that optimises the load capacity, and is also known as a stacking sequence optimisation. This is evident from the fact that different configurations result in different in-plane and out-of-plane stiffness matrices, which, in turn, affects the overall stiffness of the structure.

With the availability of more sophisticated and robust optimisation techniques such as genetic algorithms, previously impossible or expensive high-dimensional design optimisation problems become easier, even if the objective landscape (mass, cost, load capacity, etc.), when subjected to numerous constraints, is unpredictable. The following two chapters investigate such problems.

5 DESIGN OF LAMINATES

Following the description of the Genetic Algorithm (GA) framework in Chapter 2 and effective handling of constrained optimisation methods in GA presented in Chapter 3, this chapter covers the design and optimisation of composite laminates. It begins with a thorough description of the general design problem formulation of mass optimisation together with the GA optimisation process. Then, more detailed coverage of several case studies performed will be presented, complete with full results and discussion.

5.1 INTRODUCTION

The design of composite laminates and prismatic plate assemblies, particularly when relating to mass, is an important issue. In the simplest of correlations, reduction in mass can immediately be related to a reduction in material cost given the same material. While cost is an important economic constraint in certain cases, mass minimisation without consideration of cost is also useful, particularly in weight-critical applications. In such scenarios, with two materials of similar strength, obviously the one with lower resulting mass is preferable. An example of this is can be found between aluminium and a $[\pm 45/0/90]_S$ T300/5208 Graphite/Epoxy laminate, where their corresponding effective elastic moduli are both approximately $70\,GPa$ while their weight density is approximately $26\,kNm^{-3}$ and $15\,kNm^{-3}$ respectively (Gürdal et al., 1999).

Despite advantages that can be gained from specific tailoring of composite laminates, early usages are noticeably simplistic. Traditional materials, like metals, were simply replaced with fibre-reinforced composite (FRC) laminates that were quasi-isotropic, thereby suppressing any directional properties of the unidirectional lamina and maintaining, to a certain degree, the predictability of behaviours previously offered by isotropic materials. One such laminate is like the one featured above, which has equal percentages, in terms of mass, of unidirectional lamina aligned symmetrically about the laminate mid-plane.

It is naive to consider that mass of a laminate can be reduced only by the optimisation of the individual lamina thickness. Unlike the use of a single traditional isotropic material, FRC laminates can be tailored to produce different effective laminate material properties for different design requirements and loading cases. As seen in the analytical framework of composite laminates presented in Chapter 4, the layering and variation in fibre orientations clearly have an effect on the stiffness properties of the resulting laminate. This opens the possibility of designing specialised laminates that are not only able to perform the structural duties provided by traditional materials, but do so with greater efficacy.

In general, the challenge of designing composite laminates, hereafter refers specifically to FRC laminate unless noted otherwise, is to maximise the utility of their directional properties in terms of mass, cost, buckling load capacity, amongst many others. For most engineering applications, analysis that addresses fibre-matrix interface stresses is unrealistic and as such, such micro-behaviours are often ignored. What are of more concern are usually the restrictions imposed on composite laminate as a result of commonly observed failure modes such as buckling and vibration. The emphasis of the design optimisation process here is on continuous, constant thickness laminates under any combination of in-plane longitudinal, transverse and shear loads. It is, however, important to note that realistic structures often include discontinuities and other quirks such as bolt holes and thickness variations in ply drop-offs that produce other localised issues such as stress concentrations. One assumption here is, therefore, that laminates are continuous and have constant thickness.

Composite laminates, unlike their other traditional counterparts such as metals or even wood, require more sophisticated modelling techniques in order to predict macro-mechanical behaviour such as buckling or vibration. This is due, in part, to the generally anisotropic nature of such laminates and various other corresponding assumptions as outlined earlier. With such apparent difficulties in the analysis, it is therefore not unusual to find that designs of composite laminates often centre on the use of approximate modelling techniques such as the finite element method (FEM).

Fortunately, the availability of the exact finite strip method by Wittrick (1968a, 1968b) together with a numerical iterative technique for solving the resulting transcendental eigenvalue problem, the Wittrick-Williams algorithm (Wittrick and Williams, 1971b), enables fast analysis of plates and plate assemblies in buckling or vibration problems. Therefore, the analysis technique used here is modelled on the above VIPASA model because of its simplicity and exactness. Although concerns may arise in anisotropic plates or in the presence of shear, they are unwarranted as the results are still accurate, albeit conservative estimates.

The justification of using the VIPASA model is also in its speed, where the VICON model is inevitably slower due to half-wavelength coupling of stiffness matrices. Bearing in mind that since GA is used here as the optimisation tool and that it encapsulates a population rather than a single candidate solution at any one time, any computational savings are welcome indeed. Obviously, real-world applications should nevertheless use the more accurate model provided by VICON. Here, the emphasis is on evaluating the GA rather than on designing realistic aircraft structures.

A VIPASA analytical component was coded in C++ (approximately 2591 lines) to provide a more consistent platform for testing the design of composite laminates (also cases considered in Chapter 6). The only modification required to the GA component is to additionally code separate objective functions for each of the design cases considered.

5.2 Mass Minimisation Design Problem Formulation

The goal of the optimisation is to minimise the mass of a composite laminate subjected to two types of constraints – structural constraints and physical attributes. The former is provided to account for structural obligations such as load capacity, etc., while the latter is to account for commonly imposed restrictions or practices in the fabrication process, although it also has significance on mechanical responses. Generally, the mass minimisation design problem can be formulated, like a typical optimisation problem, as

$$\begin{array}{lll}
\textbf{Minimise} & & \\
\quad M(\mathbf{x}) & \mathbf{x} \in \mathbf{X} & \\
\textbf{subject to} & & \\
\quad CS_i(\mathbf{x}) \leq 0, & i = 1,2,\ldots,n_{CS}, & \\
\quad CP_j(\mathbf{x}) \leq 0 & j = 1,2,\ldots,n_{CP}. & \text{5-1} \\
\textbf{where} & & \\
\quad \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB} & &
\end{array}$$

The objective function, in this case, is the *mass function* $M(\mathbf{x})$ with corresponding variable set $\mathbf{x}$ that governs the mass of the laminate and implemented constraints. Note that the optimisation here is bounded, as represented by $\mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB}$. The structural constraints and restriction on physical attributes are accounted for by the set of constraints $CS_i(\mathbf{x})$ and $CP_j(\mathbf{x})$ respectively, giving a total of $n_{CS} + n_{CP}$ constraints.

The same design problem formulation is sufficiently general to encompass problems concerning prismatic assemblies, as will be covered in the next chapter. One of the modifications required will be the objective mass function $M(\mathbf{x})$, which will undoubtedly include more design variables to reflect the configuration of the assembly. The constraints, to be described below, will, in general, be also applicable to prismatic assemblies unless otherwise stated. It should be noted that the constraints covered below do not include the full set of constraints, because some are specific to the individual design problems.

5.2.1 Buckling Constraint

In the case studies conducted, only a single mechanical constraint is imposed on the overall system – buckling load capacity. All laminates are required to be designed to have a critical buckling load of more than a design threshold. It is worthwhile to note that a single design threshold does not necessarily directly imply a single loading condition. The buckling load capacity of each laminate is performed using the VIPASA model and thus laminates of candidate designs are expected to behave, in theory at least, in a manner consistent with any corresponding assumptions.

A naive implementation of this constraint will require the full evaluation to find the lowest critical buckling load, and may be necessary if a different analytical model is used. However, the VIPASA model uses the Wittrick-Williams algorithm that may be exploited in such design optimisations. Recall again that the algorithm returns the number of eigenvalues that lie between some lower bound and upper bound positive trial value. In this case, this information is more than sufficient to determine whether a candidate is a stable (feasible) design.

Consider a candidate laminate $\mathbf{x}$ with a set of loading conditions $\mathbf{N}_{\mathrm{DESIGN}}$ imposed and a set of $\boldsymbol{\lambda}$ half-wavelengths for every loading condition $\mathbf{N}_{\mathrm{LOAD}} \in \mathbf{N}_{\mathrm{DESIGN}}$, where $\mathbf{N}_{\mathrm{LOAD}}$ is a singular loading event of any combination of invariant in-plane longitudinal compressive load per unit width N_{L}, transverse compressive load per unit length N_{T} and shear load per unit width N_{S}, i.e., $\mathbf{N}_{\mathrm{LOAD}} = \{N_{\mathrm{L}}\ N_{\mathrm{T}}\ N_{\mathrm{S}}\}$. For any given loading condition $\mathbf{N}_{\mathrm{LOAD}}$ and half-wavelength $\lambda \in \boldsymbol{\lambda}$, the Wittrick-Williams algorithm returns the number of eigenvalues as the number of critical buckling loads *exceeded* at the trial value $\mathbf{N}_{\mathrm{LOAD}}$. Thus, the candidate laminate $\mathbf{x}$ is considered stable if, for all $\lambda \in \boldsymbol{\lambda}$ of every corresponding $\mathbf{N}_{\mathrm{LOAD}} \in \mathbf{N}_{\mathrm{DESIGN}}$, there exist no eigenvalues (critical buckling loads) less than the trial value $\mathbf{N}_{\mathrm{LOAD}}$. To formalise, a laminate is defined as stable in buckling if and only if

$$\forall \mathbf{N}_{\mathrm{LOAD}} \in \mathbf{N}_{\mathrm{DESIGN}}, \forall \lambda \in \boldsymbol{\lambda}, J(\mathbf{N}_{\mathrm{LOAD}}, \lambda) = 0 \qquad 5\text{-}2$$

where $J(\mathbf{N}_{\mathrm{LOAD}}, \lambda)$ is the number of eigenvalues less than the trial value $\mathbf{N}_{\mathrm{LOAD}}$ at a particular λ, as stated simply as J in the Wittrick-Williams algorithm of Eq. 4-26.

The set of half-wavelengths $\boldsymbol{\lambda}$ is determined typically by the laminate length ℓ, such that for all $\lambda \in \boldsymbol{\lambda}$, they divide equally into the laminate length by some integer value. All conducted test cases will be analysed for the first 10 half-wavelengths, i.e.,

$$\boldsymbol{\lambda} = \left\{ \lambda = \frac{\ell}{k} : k = 1,2,\ldots,10 \right\} \qquad 5\text{-}3$$

Since the VIPASA analytical model is used here, results at longer half-wavelengths ($\ell/3 \leq \lambda \leq \ell$) are conservative in cases of laminates with simply-supported ends in the presence of shear load or anisotropy.

The combination of definitions from Eq. 5-2 and 5-3 allows a prioritised lazy evaluation of constraints. For a design candidate, constraint evaluation is only performed if its objective (mass) is equal or better than the current known optimum, i.e., lazy evaluation. If lazy evaluation is performed, prioritisation of constraints implies that evaluation is terminated once a single $J(\mathbf{N}_{\mathrm{LOAD}}, \lambda)$ is non-zero, since there exist buckling load factors less than the trial load $\mathbf{N}_{\mathrm{LOAD}}$ at the particular half-wavelength λ, i.e., the structure is not safe.

5.2.2 Balanced and Symmetric Constraint

The constraints implementing the balanced and symmetric rules are necessary to model as close as possible the assumptions inherent in the VIPASA analytical model. They are the two other restrictions enforced on the physical attributes of a laminate. It is possible to design composite laminates using the VIPASA model without enforcing these limitations, but then the results obtained are no longer accurate. Furthermore, there are many practical reasons for the use of such configurations.

The rule of symmetric configurations of lamina will ensure that there is actual decoupling between in-plane and out-of-plane deformations. Symmetry is achieved by mirroring configurations about the mid-plane of the laminate. Consider a candidate laminate $\mathbf{x}$ with n layers, where the top and bottom surface of k-th layer can be defined using the laminate reference z axis with coordinate as z_{k-1} and z_k respectively and with corresponding fibre orientation θ_k. The constraint that the laminate is symmetric about the mid-plane ($z = 0$) is satisfied if and only if

$$\forall k \in \left\{ k \leq \frac{n}{2} : k \in \mathbf{Z}^+ \right\}, \theta_k = \theta_{n-k+1} \wedge |z_{k-1}| = |z_{n-k+1}| \qquad \text{5-4}$$

On the other hand, the rule of balanced configurations of lamina will ensure the elimination of shear-extension coupling, therefore ensuring that the laminate displays orthotropic behaviours along the principal elastic axes with regards to membrane deformations. Consider again the candidate laminate $\mathbf{x}$ as described above, where the k-th layer has a corresponding thickness $t_k = z_k - z_{k-1}$. The constraint that the laminate is balanced is satisfied if and only if

$$\forall \theta^* \neq \frac{i\pi}{2}, i \in \mathbf{Z}_0^+, \sum\nolimits_{k=1,\theta_k=\theta^*}^{n} t_k = \sum\nolimits_{k=1,\theta_k=-\theta^*}^{n} t_k \qquad \text{5-5}$$

5.2.3 Discrete Thickness Constraint

Many designs and optimisations of composite laminates and prismatic structures assumed continuous layer thicknesses, such as those conducted for benchmark stiffened aircraft wing panels by York et al., (1993). This is most often a result of the underlying optimiser, which is seldom unable to straddle discontinuous design variable values or cost functions. However, manufacturing considerations often require a laminate specification of integer (discrete) multiples of a standard lamina (ply) thickness, typically in the order of $0.1mm$ per lamina. Consider any general standard lamina thickness t_{L}, and a candidate laminate $\mathbf{x}$ with n layers, where the k-th layer has a corresponding thickness t_k. The constraint that the laminate is discrete is satisfied if and only if

$$\forall k \in \{k \leq n : k \in \mathbf{Z}^+\}, t_k \in \{dt_{\mathrm{L}} : d \in \mathbf{Z}_0^+\} \qquad \text{5-6}$$

where d represents the integer multiples of the standard lamina thickness.

The imposing of the discrete thickness constraint adds a further difficulty to the design process, transforming it to a discrete optimisation process, which cannot be readily solved using the common gradient-based approaches applicable to continuous optimisation problems.

Although naive techniques such as rounding-up to the nearest discrete value are possible, they often produce stable sub-optimal designs in terms of mass, which are most often not the global optima of the discrete problems. Another route to such a problem (Kennedy et al., 1999) is to deploy a two-step approach, where the first step is to apply the standard continuous design phase to obtain the global continuous optimum, and then using this as a reference point, deploy discrete optimisation methods to arrive at the global discrete optimum.

5.3 Mass Minimisation with Genetic Algorithms

Common design problems often consist of a priori knowledge of laminate configuration, with a typical configuration often being something along the line of $[0/\pm 45/90]_S$. The design strategy here takes this further by allowing the configuration to be designed as well. It has been mentioned earlier that specific tailoring of composites allows different stiffness properties, and thus can be more efficacious. This avenue is explored fully here rather than implement a fixed configuration where the design variables are merely the thickness of the laminate. The specific details will be covered later for the various laminate mass minimisation cases.

It is also not uncommon for designs to concentrate on continuous thickness, and, as mentioned before, this requires further rounding to discrete multiples of a standard ply thickness if manufacturing practices are taken into consideration. The design strategy developed here does not go through this continuous design process, but concentrates directly on the actual discrete optimisation process itself.

GA is used as the optimisation technique here. Developments, covered earlier, allow a more efficient handling of constraints. Despite the simplicity of the Wittrick-Williams algorithm, used to measure the buckling constraint satisfaction, it is still undoubtedly a computationally expensive algorithm as it involves setting up the individual layer stiffness matrices before forming the overall laminate stiffness matrix. The various generalised details of the underlying GA optimiser are as discussed below.

5.3.1 Genetic Structures

All design variables are coded using the binary representation. In most cases, the Gray-encoding will be used as it displays adjacency properties not found in the standard binary encoding. This allows the resulting genotypes to be set up in a standard way, and concentrating rather on the performance of the constraint handling technique and the optimisation problem.

Three general properties are usually part of the design variables vector – geometric dimension, fibre orientation and discrete thickness. Geometric dimensions are encoded in the standard way, as found normally in numerical optimisation problems, where a lower and upper bound coupled with a required precision becomes the determining factor for the number of bits required. Unless otherwise stated, all such geometric dimensions will be encoded in such a manner using Gray codes, where the specific details such as number of bits, etc., will be provided in their respective sections.

5.3.1.1 Representation of Fibre Orientation

What requires further explanation, though, is the encoding of fibre orientation. Four types of layers with different fibre orientations will be considered throughout this design optimisation - 0°, 45°, 90° and -45°. This requires a binary string of 2 bits, and the default encoding being Gray-encoding. Exploiting the adjacency property of Gray-encoding gives an added advantage in the representation of fibre orientations. The binary representations together with their corresponding ply orientations are as illustrated in Figure 5-1.

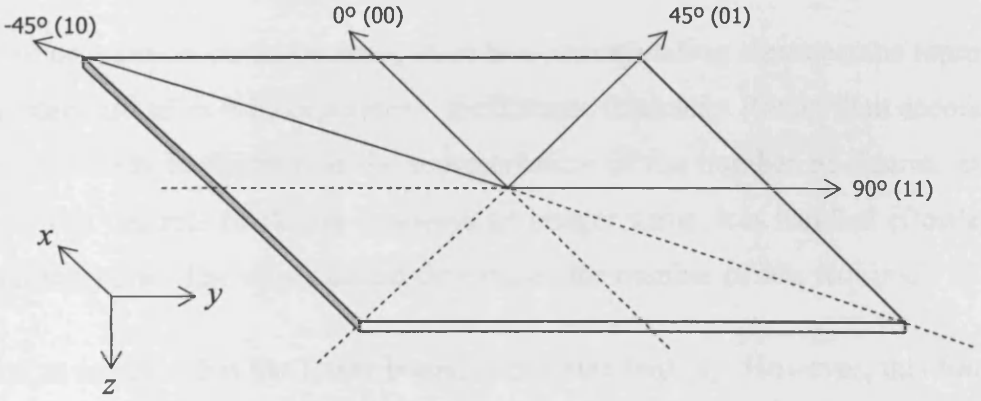

Figure 5-1 Fibre orientations in consideration and corresponding encoded chromosomal representation. Notice also that the adjacency property of Gray-encoding here ensures that a 1-bit mutation shifts the fibre orientation of a lamina by 45º.

In this case, the mutation of a single bit will imply a shift in the fibre orientation by $\pm 45^\circ$, where a clockwise rotation is considered positive. An off-axis layer, i.e., 45° and -45°, comes with a corresponding layer that satisfies the balanced constraint, hence $\pm 45^\circ$ and $\mp 45^\circ$ respectively. This restricts the configuration in a way that once an off-axis layer is encountered, the next layer must be a layer that will balance the configuration.

On the other hand, the symmetric constraint is more easily handled by only modelling half the configuration, either the top or bottom half. However, this introduces a natural assumption that there is an even number of layers in the laminate, which may or may not be desirable. For example, a random 10-layer symmetric configuration can be

$$S = (01)(00)(11)(11)$$

which represents the symmetric $[\pm 45/0/90/90]_S$. Symmetry can also be achieved if the configuration is $[\pm 45/0/90/90/90/0/\mp 45]$, which implies a saving of one layer because symmetry is about the mid-plane of the mid-layer. An optional *even-odd* bit can be introduced to force this behaviour, but is deemed here unnecessary. Furthermore, savings can only be made if the layers do not have off-axis fibre orientation, otherwise the balanced rule will be broken.

5.3.1.2 REPRESENTATION OF DISCRETE THICKNESS

For all fibre orientation chromosomes, there is a corresponding chromosome representing the number of standard plies it incorporates – the discrete thickness. Rather than encode the actual thickness, all which is required is the representation of the number of discrete standard ply thickness. Since discrete thickness is always an integer value, it is handled effortlessly by the binary representation. The upper bound determines the number of bits required.

It is natural to assume that the lower bound is one standard ply. However, this forces a lower bound on the number of layers in the laminate, which may or may not be desirable. One solution is to deploy a variable length genotype. Such a genotype is advantageous in the sense that it immediately reflects the laminate configuration. The downside, though, is the need to accommodate for the variability of the genotype during genetic operations.

It is the result of this that emphasised the benefit of allowing the lower bound for the discrete thickness to be zero rather than one, where the number of possible layers can be fully represented resulting in fixed-length genotypes. A zero thickness then simply implies a non-existent layer as opposed to the need of 'deleting' the layer from the genotype. Thus, the existence (or non-existence) of a layer can be obtained from simple genetic operators, such as crossover and mutation, without having the need to introduce specialised ones.

An unnecessary complication, however, will arise out of adopting the zero lower bound approach. There is a possibility, however small, that there is zero thickness for all layers resulting in a laminate of zero mass, which inevitably, cannot be handled by the analysis. In such a case, analysis is not performed but assigned objective and constraint violations, if any, identical to that of the worst observed genotype.

5.3.2 Measurement of Performance

The objective function here is the mass of the resulting structure. This depends entirely on the material properties and the geometric dimensions of each genotype as candidate solutions, and cannot be fully detailed until individual cases are covered.

The balanced and symmetric constraints have already been handled entirely by the genetic structures, which ensure that all genotypes will be feasible in this respect. The discrete thickness requirement is also implicitly handled by the genetic representation.

The buckling constraint is handled as a normal constraint with full evaluation required in determining the feasibility of the genotype. However, the number of constraint evaluations is high. Consider a single loading condition $\mathbf{N}_{\text{LOAD}}$ evaluated over 10 half-wavelengths[1] λ, which results in 10 constraints. A case with more design loads will become highly impossible and expensive. Therefore, constraints are prioritised and undergo lazy evaluation, as described earlier, to reduce the number of constraint evaluation. For all loading conditions $\mathbf{N}_{\text{LOAD}} \in \mathbf{N}_{\text{DESIGN}}$ and corresponding half-wavelengths $\lambda \in \boldsymbol{\lambda}$, constraint evaluations terminate once a single combination of $(\mathbf{N}_{\text{LOAD}}, \lambda)$ is violated.

In essence, the GAPR-PLZ (see Chapter 3) is used here since previous results show that the least number of constraint evaluations were performed. The lack of a tangible fitness value in

[1] The number of half-wavelengths considered here is excessive as the critical buckling mode will most likely occur at longer half-wavelengths for such a simple configuration. This choice is aimed primarily at stimulating the computational requirements with respect to constraint evaluations that would be needed for larger problems.

the Pareto-Ranking (PR) approach also restricts the set of selection mechanism that can be used. The PR approach is also able to implicitly determine superiority of a genotype over another despite the minimisation process, which, in traditional fitness formulation, will require a scaling of fitness.

5.3.3 GENETIC OPERATORS

The use of the PR approach to fitness handling allows little room for choice in selection mechanisms. In this case, the binary tournament selection is deployed. The sampling mechanism is expected to select the number of genotypes identical to the population size, i.e., a fixed size sampling.

The recombination operator used here is the α -uniform crossover complimented by the uniform mutation operator. The mutation rate or probability of mutation, in all cases, will be determined locally, while the probability of crossover is fixed at 0.8. The probability of gene swap α is 0.3. These genetic parameters were adopted from the previous test (QTC) presented in Chapter 3, since they have been successful.

The generational steady-state replacement strategy is used here. It is slightly different to the traditional steady-state replacement only in the sense that the number of sampled genotypes here is the actual population size. Typically, only a fraction of the population size is sampled when using steady-state replacement strategy. Upon completion of sampling, crossover, mutation, objective evaluation and constraint evaluation, the offspring (transitional) population is merged with the parent (original) population. The 50% top-performing genotypes then becomes the default original population.

5.3.4 ADAPTIVE PLAN

The adaptive plan here adopts the standard GA framework without additional modification. In other words, the model is deployed without prevention of in-breeding, incest, etc., and other such features. The overall GA optimisation framework is as illustrated in Figure 5-2, which is the GAPR-PLZ strategy developed (see Chapter 3). The GA cycle is intentionally left generic to indicate the flexibility of the constraint handling strategy, since the genetic structures, operators, etc., are independent.

5.4 Benchmark Laminate

The design of a benchmark composite laminate here is to obtain a lower bound reference to a design case that will be an eventual datum reference for subsequent mass minimisation of more complicated plates, where longitudinal voids will be introduced.

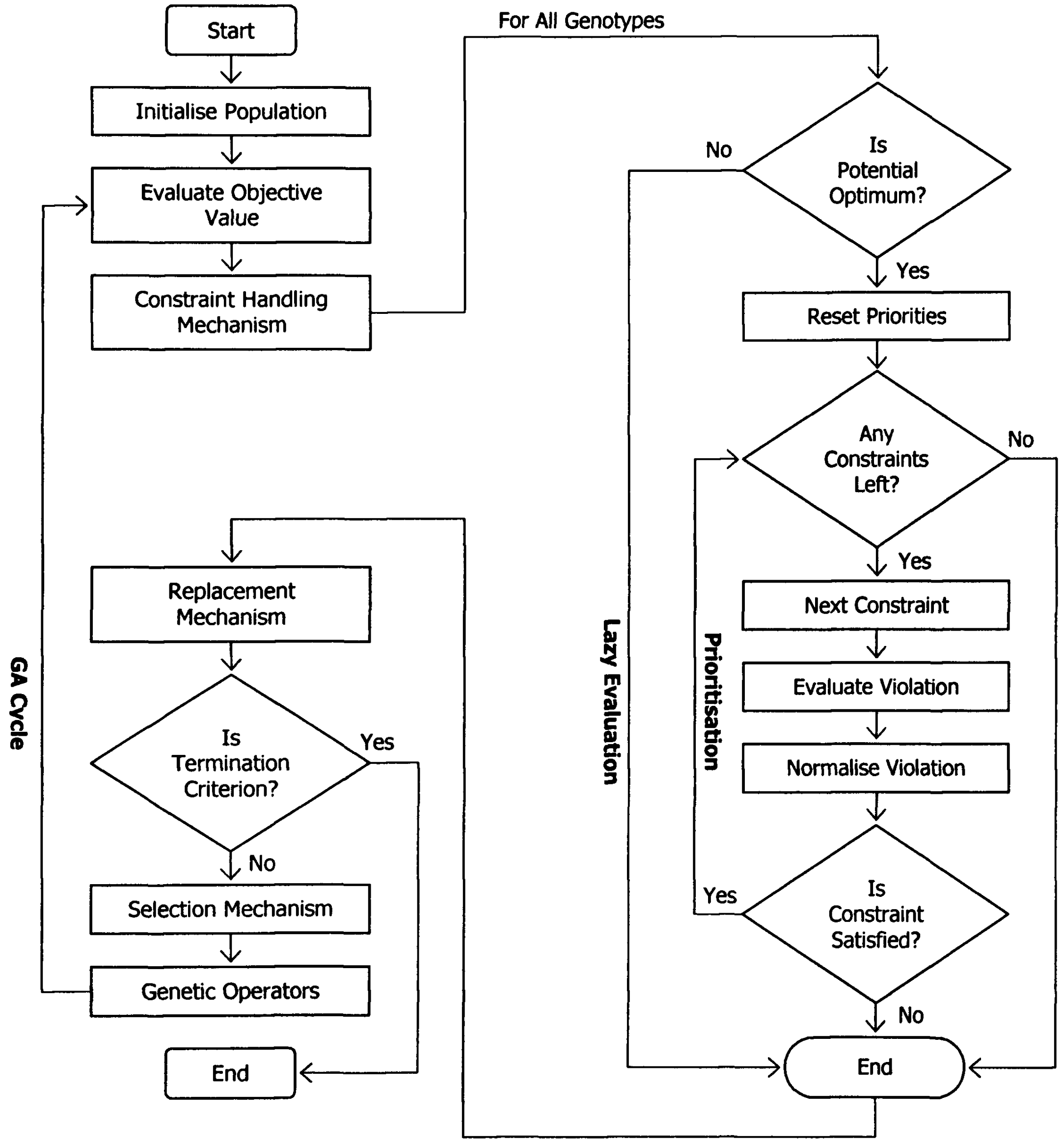

Figure 5-2 Developed GA optimiser framework with prioritisation and lazy evaluation of constraints.

As opposed to conventional design problem, the design variables will not be restricted to the discrete thickness of the layers with fixed configuration but extended to include finding the optimum sequence of fibre orientation in the hope of producing more effective laminates as shown in Figure 5-3.

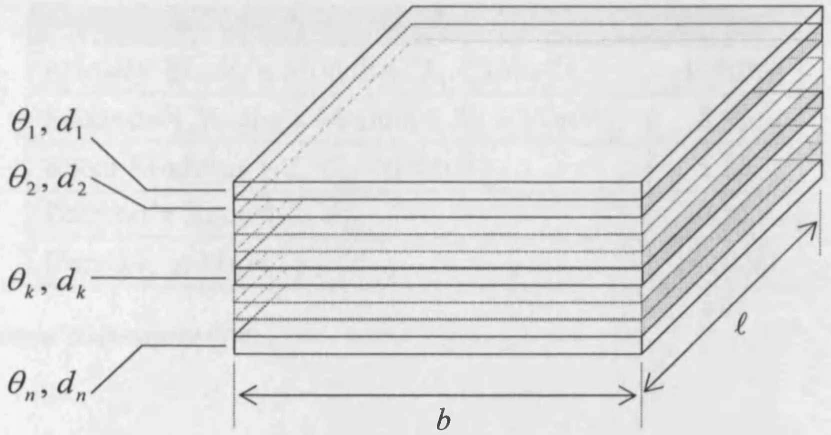

Figure 5-3 Design variables, layer fibre orientation and corresponding discrete thickness, of the benchmark laminate. The width and length of the laminate is fixed.

5.4.1 Problem Definition

With the design variables defined, as shown above, the objective mass function $M(\mathbf{x})$ can now be formulated. Let θ_k be the fibre orientation of the k-th layer with corresponding discrete thickness t_k, such that $t_k / t_\mathrm{L} = d_k$, where t_L is the standard ply thickness and d_k being the integer number of standard plies for the k-th layer . For a laminate consisting of n layers, the objective mass function $M(\mathbf{x})$ is therefore

$$M(\mathbf{x}) = t_L b \ell \rho \sum_{k=1}^{n} c_k \qquad 5\text{-}7$$

where $c_k = 2d_k$ for off-axis fibre orientation, i.e., $\theta_k \notin \{i\pi/2 : i \in \mathbf{Z}\}$, or $c_k = d_k$ otherwise, and ρ is the density of the material. The width and length, b and ℓ respectively, in this case, are both fixed at $0.2m$, thus producing a square plate. These dimensions are taken arbitrarily but are nonetheless common dimensions expected of a composite laminate. The standard ply thickness here is taken to be $0.0001m$. The material used is carbon fibre reinforced plastic (CFRP) with material elastic properties defined in Table 5-1.

Property	Value
Primary Young's Modulus, E_1 (GNm^{-2})	125.00
Secondary Young's Modulus, E_2 (GNm^{-2})	8.80
Shear Modulus 1-2, G_{12} (GNm^{-2})	5.30
Poisson's Ratio 1-2, ν_{12}	0.35
Density, ρ (kgm^{-3})	1620.00

Table 5-1 Material properties for CFRP.

5.4.2 Parameter Encoding

Only two sets of parameters are encoded into the genotypes – fibre orientation and discrete number of layers. Also, the genotypes only need to model half of the maximum number of layers because of the symmetry constraint.

The fibre orientation, as mentioned earlier, is represented by a 2-bit string. Modelling the complete laminate requires 12 bits. The discrete thickness, on the other hand, is modelled accurately by 10 bits per layer, thus requiring a total of 60 bits for the $n/2 = 6$ layers in the top half of the laminate. A genotype, therefore, has the following structure

$$G_{\text{BENCHMARK}} = \Big(\underset{\substack{\uparrow \\ \theta_1}}{01}\,\Big|\,\underset{\substack{\uparrow \\ d_1}}{1001000101}\Big)_1 \underset{\substack{\ldots \\ \ldots}}{\ldots} \Big(\underset{\substack{\uparrow \\ \theta_{\frac{n}{2}}}}{01}\,\Big|\,\underset{\substack{\uparrow \\ d_{\frac{n}{2}}}}{1001010010}\Big)_{\frac{n}{2}}$$

This implies a total of 72 bits per genotype, making the problem substantially difficult to tackle with the additional variation in stacking sequence. A population size equivalent to the string length is assumed without any fine-tuning procedures, and undergoes the operators as described earlier with a probability of mutation of 3 in 72 bits. Bits are forced to change during a mutation, i.e., $1 \leftrightarrow 0$, thereby guaranteeing that mutation does occur. The optimisation process terminates once all genotypes have the same mass, or after a maximum of 2000 generations.

5.4.3 Comparison Laminate

Before going into the main results, a comparison study was performed to determine the optimality of the discrete mass obtained from the main, rather more generalised, case. A preconfigured laminate, with general stacking sequence as shown in Figure 5-4, was designed such that the discrete thickness became the only set of design variables[1]. The design variables have the same discrete thickness bounds as the benchmark laminate, as shown in Table 5-2.

The optimisation of the comparison study was much easier as the fibre orientations of the individual layers are now not allowed to change even though they can be absent from the final configuration. In other words, the stacking sequence is fixed. The general case of the benchmark laminate should then at least be comparable in mass since the balanced and symmetry rule is also enforced. Each discrete thickness parameter requires, as before, a 10-bit string, thus totalling 60 bits. All GA parameters are as mentioned for the general benchmark problem except where the mutation rate is 2 bits in 60, with a population size also identical in magnitude to the overall genotype length.

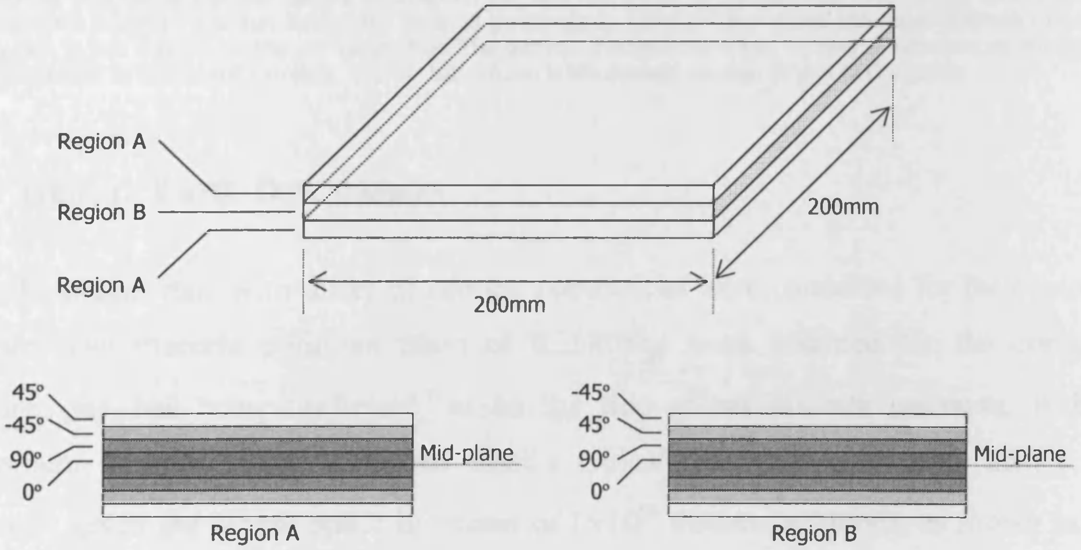

Figure 5-4 Comparison laminate for the benchmark problem. The stacking sequence is fixed unlike the benchmark problem.

[1] The reason for specifying such a laminate stacking sequence with two independent regions A and B is to cater for regions of longitudinal voids in design cases presented in section 5.5.

Region	Fibre	Lower Bound	Upper Bound	Variables
A	±45°	0	1023	d_1
	90°	0	1023	d_2
	0°	0	1023	d_3
B	±45°	0	1023	d_4
	90°	0	1023	d_5
	0°	0	1023	d_6

Table 5-2 Design variables for the comparison study and the bounds, which is the number of standard plies.

Region A			Region B			No. of Plies	Mass (kg)	CPU
±45°	90°	0°	±45°	90°	0°			
2	0	0	1	4	9	46	0.29808	3.036
2	0	0	1	7	6			
3	0	0	0	1	10			
3	0	0	0	6	5			
3	0	0	1	0	9			
3	0	0	3	4	1			
3	1	0	1	6	1			
x8	x4	x4	x4	x2	x2			

Table 5-3 Global discrete optima obtained on the best run out of 10 independent runs for the comparison laminate problem. The run terminated once all genotypes in the population share the same optimum mass, which in this case, is on the 29th generation. The last row indicates the actual number of occurrences of each orientation in the overall laminate, and the last column is the average solution time in CPU seconds.

5.4.4 Results and Discussion

Ten independent runs with different starting populations were conducted for the comparison laminate. The discrete optimum mass of 0.29808kg were obtained for the comparison laminate, and has been confirmed[1] to be the true global discrete optimum. It is also unsurprising that the results obtained from a typical run often gave more than a single optimum[2], given the search space in excess of 1×10^{18} discrete solutions, as shown in Table 5-3. Although no niche induction techniques were deployed, the population tends to shift from

[1] An 'exhaustive' search was performed to verify the optimum mass of the comparison laminate. Since the best known optimum is close to the lower bound, the search started from the lower bound and slowly increased the number of plies in each region. Once a 'just stable' configuration was found, then configurations with higher mass no longer need to undergo the full stability check.

[2] All the best genotypes (sharing the same mass) were reported. It does not imply that they were all maintained concurrently in a single generation, which would be virtually impossible unless niche induction techniques were introduced. It was observed, though, that there were occasionally at least 2 different genotypes that share the same optimum mass in a single generation.

one optimum to another. Nonetheless, all eventually settle onto a dominating solution at the final population before termination. It was also noticed that any additionally discovered global discrete optima tend to be similar to one another, which, in turn, implies that optima, crowded together into a region traversable by the genetic operators, tend also to be discovered.

The comparison laminate with fixed stacking sequence was extremely easy to solve for the underlying GA even though the upper bound was artificially forced to be high, making the optima clustered near the lower bound. Together with the constraint handling technique modelled on multi-objective optimisation principals, the GA was able to direct its search accurately despite the vast search space. Given this result, the benchmark laminate with its additional flexibility of varying stacking sequence should improve, or at least match, the global discrete optimum of the comparison laminate.

Ten independent runs, where each population is seeded differently, were carried out for the benchmark laminate. All runs were expected to terminate either when the 2000th generation was reached, or when all genotypes shared the same mass. Results indicate that the latter termination criterion was met long before the former, with all converging before the 300th generation as illustrated in Figure 5-5. Whether the final discrete mass returned is the global discrete optimum is not verified[1]. Nonetheless, all ten runs converged onto a mass that matches exactly the global discrete optimum mass of the comparison laminate - $0.29808kg$ with 46 standard plies. Hence, no improvements in mass were achieved although the minimum target was met.

As before in the comparison laminate, the problem was still handled effectively by the underlying GA and constraint handling technique despite the lack of a tangible fitness function, and having to rely on using sampling mechanisms that do not require tangible fitness values. In spite of the criticisms that are often thrown at selection methods such as the tournament selection used here, particularly selection error, results here indicate otherwise. Evidently, Figure 5-5 also shows that the problem takes longer to optimise. This is not unexpected with the additional complexity of the varying stacking sequence and a resulting larger search space exceeding 4×10^{21}. Nevertheless, the implemented GA package showed encouraging robustness, even in the presence of incomplete violation knowledge, as Figure

[1] Trial runs were conducted under the same GA parameters where they were allowed to run uninterrupted until the 2000th generation to determine whether any further improvement in mass is possible. None managed to locate a laminate with a lower mass.

5-6 shows. With the optimal region located near the lower bounds of the set of discrete thickness design variables, none of the runs began their search from infeasible regions.

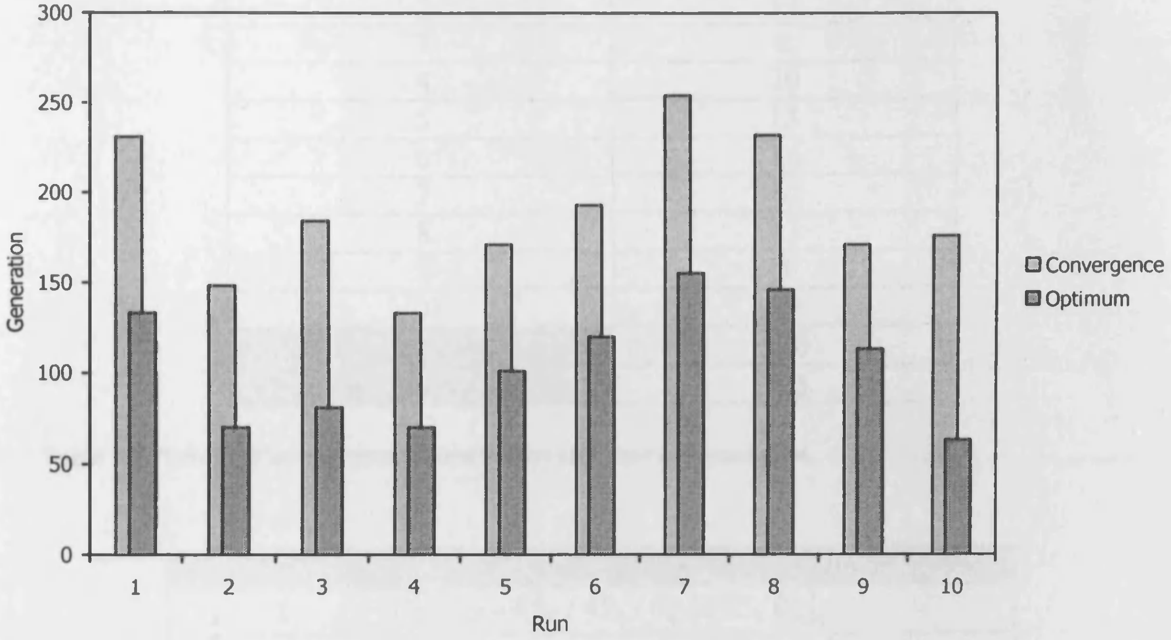

Figure 5-5 Number of generations required until convergence, i.e., termination criterion is met, and until the optimum mass, i.e., the final mass each run converges onto, is first discovered for the benchmark laminate problem.

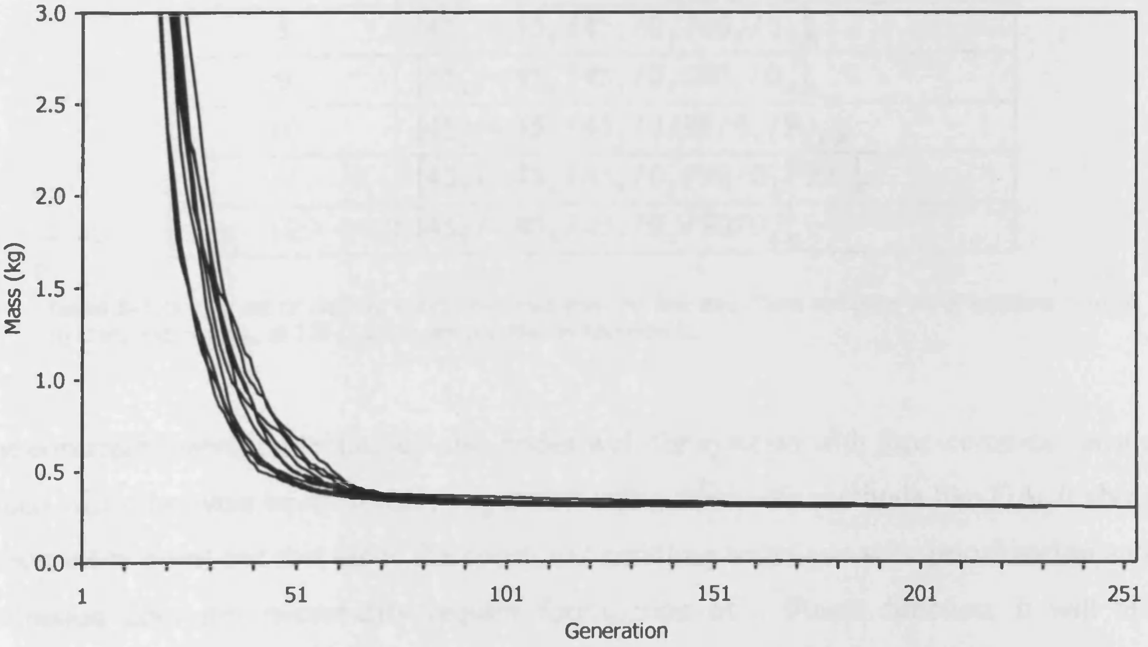

Figure 5-6 General convergence trend of all ten runs conducted for the benchmark laminate. The mass is the population average mass at every generation.

Run	No. of Optima Found
1	12
2	12
3	8
4	20
5	13
6	16
7	7
8	13
9	8
10	19
Total	**128**
Overall Unique Optima	**125**

Table 5-4 Number of unique optima found in each of the ten runs conducted.

Solution No.	Laminate Configuration
1	$[45_2/-45_6/45_4/0_4/\pm 45/0_5]_S$
2	$[45_2/-45_6/45_4/0_4/\mp 45/90_5]_S$
3	$[45_2/-45_6/45_4/0_5/90_6]_S$
4	$[45_2/-45_6/45_4/0_6/90/45_2/-45_2]_S$
5	$[45_2/-45_6/45_4/0_6/90_5]_S$
6	$[45_2/-45_7/45_5/0_4/90_5]_S$
7	$[45_3/-45_6/45_3/0/90_2/0_2/90_6]_S$
8	$[45_3/-45_6/45_3/0_2/90_2/0_7]_S$
9	$[45_3/-45_6/45_3/0_3/90_2/0_6]_S$
10	$[45_3/-45_6/45_3/0/90/0_3/90_6]_S$
11	$[45_3/-45_6/45_3/0_2/90/0_2/90_6]_S$
12	$[45_3/-45_6/45_3/0_2/90/0_8]_S$

Table 5-5 Unique set of discrete optima returned from the first run. These complete set of solutions from all 10 independent runs, all 128 of them, are provided in Appendix II.

The constraint handling technique also bodes well for systems with time-complex constraints, which will otherwise be extremely expensive using stochastic methods like GA. It should be important to point out that since the constraint handling technique with prioritisation and lazy evaluation does not necessarily require formulation of a fitness function, it will also be generally applicable to other stochastic methods that require comparison between two candidate solutions. Even in the current benchmark laminate problem with 10 half-wavelengths to be considered, full evaluation will then demand 10 full constraint evaluations for each genotype. If the problem is required to consider several more distinct loading conditions, then concurrent optimisation will become almost prohibitive.

Yet again, unsurprisingly, every single run returned several optima with all of them pointing to the same discrete mass of $0.29808kg$. While most traditional methods will return only a single solution, the ability of GA to detect multiple optima even without any niche induction techniques implemented is an added advantage. One possible use for these detected optima, in this case, is to probably perform secondary analysis upon completion of the GA process, with one being the candidate solution that gives the highest safety factor in terms of buckling load capacity. The number of optima found for each run is as summarised in Table 5-4, with details from the first run in Table 5-5.

5.5 LAMINATE WITH LONGITUDINAL VOIDS

Upon obtaining the optimum discrete mass of a benchmark laminate above, the natural question will then be to ask whether further mass savings are indeed possible for another laminate of similar dimensions under the same loading condition. The section here explores the possibility of introducing longitudinal voids in the mid-plane of the plate. Such an avenue has been previously explored by Williams and Ye (1992a, 1992b), and was shown to yield substantial mass savings when designing a composite laminate or prismatic plate assembly subjected to buckling requirements. However, there remain other issues associated with such a configuration, particularly the concern of reduced damage tolerance and other such adverse effects.

The problem here, in a certain sense, extends the earlier benchmark laminate problem. All layer thicknesses are still subjected to the discrete constraint, and the balance and symmetry constraints are also implemented. Thus, the problem is made significantly more difficult, not only in the sense of dimensionality, but in the nature of the problem itself. The general laminate with longitudinal voids in the mid-plane of the plate is as shown in Figure 5-7. The width of non-void regions are fixed at $4mm$[1] except at the edges, which are $2mm$, in order to model it as a standard repeating element. Therefore, each repeating element has a total width of $(200/v)mm$, where v is the total number of voids.

[1] Optimisation tends to make these non-void regions as small as possible. Therefore, the value here is chosen to give a realistic lower limit.

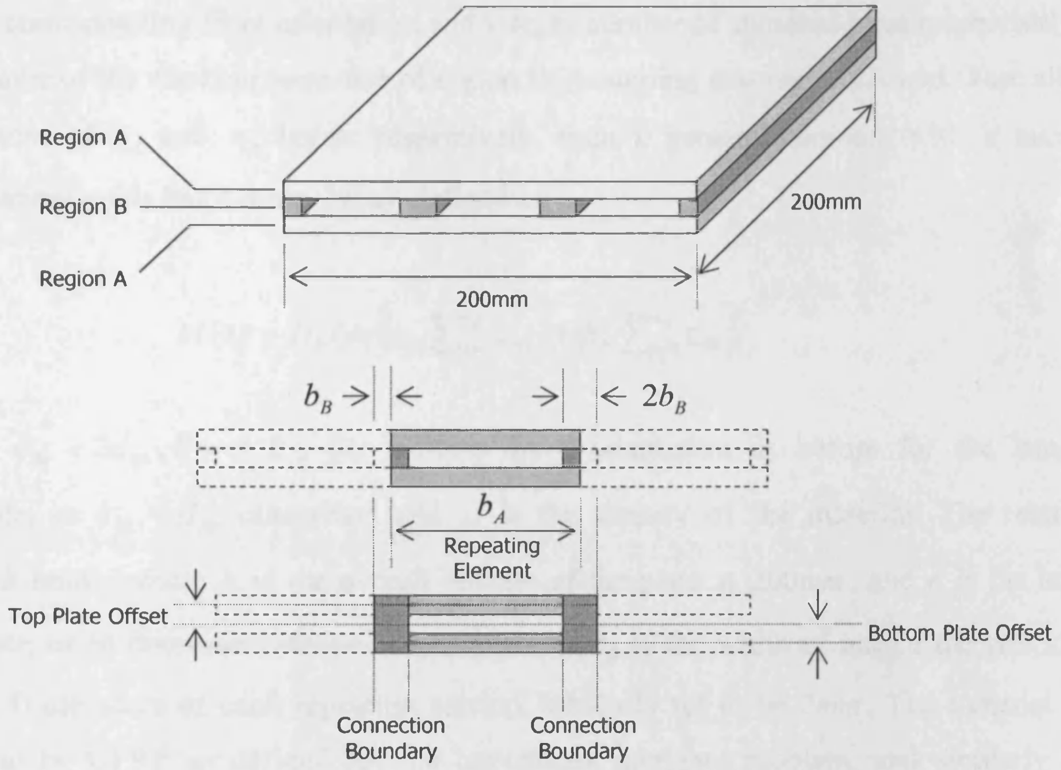

Figure 5-7 Typical laminate with 3 longitudinal voids. The problem can be simplified into the repeating elements as shown, where the number of voids determine the number of such elements. The modelling with the VIPASA model though is slightly different as shown in the bottom diagram.

The modelling using the VIPASA model is slightly different from the actual model, as shown also in Figure 5-7. The continuous top and bottom plates, region A, are allowed a set of design variables available to the benchmark laminate, θ_{Ak} and d_{Ak}, separate from the void boundary elements, region B, which also encompass all the design variables of the benchmark laminate, namely θ_{Bk} and d_{Bk}. The problem is therefore the design of the optimum stacking sequence, in terms of mass, of regions A and B. Although it is possible to allow the number of voids as well as other parameters to be design variables as well, the current problem defines a fixed number of voids.

5.5.1 Problem Definition

Three separate sets of studies were conducted for laminates with longitudinal voids. They each have a fixed number of longitudinal voids, namely 3, 4 and 5. As shown in Figure 5-8, a general objective mass function $M(\mathbf{x})$ can therefore be formulated.

Let θ_{Ak} and d_{Ak} be the corresponding fibre orientation and integer number of standard plies respectively of the k-th layer of the stacking sequence of region A. Similarly, let θ_{Bk} and d_{Bk}

be the corresponding fibre orientation and integer number of standard plies respectively of the k-th layer of the stacking sequence of region B. Assuming that regions A and B are allowed a maximum of n_A and n_B layers respectively, then a general laminate with v number of longitudinal voids has a mass $M(\mathbf{x})$ defined as

$$M(\mathbf{x}) = 2t_L \ell \rho v\left(\left(b_A \sum\nolimits_{k=1}^{n_A} c_{Ak}\right) + \left(b_B \sum\nolimits_{k=1}^{n_B} c_{Bk}\right)\right) \tag{5-8}$$

where $c_{Rk} = 2d_{Rk}, R = A, B$, for off-axis fibre orientation as before for the benchmark laminate, or $c_{Rk} = d_{Rk}$ otherwise, and ρ is the density of the material. The relationship $vb_A = b$ holds, where b is the overall breadth of the plate at $200mm$, and ℓ is the length of the plate, or in this case $200mm$. The parameter b_B is the width of one of the two identical region B elements of each repeating section, which is set to be $2mm$. The material here is taken to be CFRP as defined for the benchmark laminate problem, and similarly for the standard ply thickness.

Both regions are expected to have a maximum of 12 fibre orientation changes, $n_A = n_B = 12$, with each layer having a maximum of 1023 discrete plies, $0 \le d_{Rk} \le 1023$. In other words, both regions have the same makeup as the benchmark laminate in terms of design variables. The laminate is also expected to be stable in buckling under a single loading event of a uniform longitudinal compressive load of $100kN$, with no transverse or shear loading. This is to preserve the continuity of extending the previous problem to form a coherent process of design optimisation. The edge and support conditions are as before in the benchmark laminate.

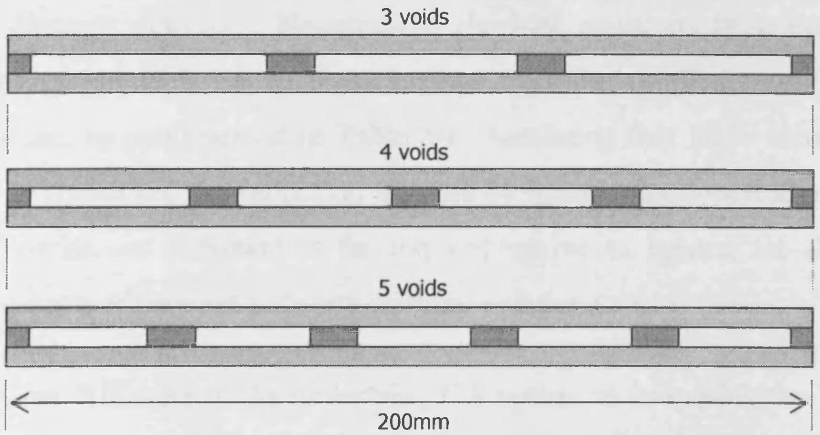

Figure 5-8 General setup of the 3 cases studied for laminate with longitudinal voids.

5.5.2 Parameter Encoding

The same two sets of parameters encoded for the benchmark laminate is also applicable here, and, as before, only half the stacking sequence of each region is modelled. Nevertheless, the presence of two regions up for optimisation implies that the number of bits is effectively doubled resulting in a genotype with 144 bits, and a problem in excess of 4×10^{43} discrete solutions. The resulting genotypic structure is then the double concatenation of the structure from the benchmark laminate

$$G_{\text{VOID}} = G_{\text{BENCHMARK}} \cdot G_{\text{BENCHMARK}}$$

The problem is noticeably more difficult than the benchmark problem, and combined with the complication of longitudinal voids, it is expected that the optimisation process will take a substantial number of generations to solve.

It is in this light and an indication from preliminary trial runs that the GA process is only subjected to the maximum generation termination criterion, where the search is allowed to proceed up to a maximum of 2000 generations. The population size is equivalent to the genotypic number of bits, i.e., 144, and undergoes the same mutation rate as the benchmark problem – 6 bits in 144. As before, bits are forced to flip during a mutation.

5.5.3 Results and Discussion

Each case was subjected to ten independent runs, where the i-th run of every case having the same starting condition. None of the best solutions for each was confirmed as to whether it was the global discrete optimum. Nonetheless, the best solutions from each case showed substantial improvement in terms of mass savings over the simplistic configuration of the benchmark laminate, as summarised in Table 5-6. Assuming that these solutions are indeed the global discrete optimum, the reduction in mass by introducing longitudinal voids in what is otherwise a continuous laminate is far too impressive to ignore, let alone the further possible improvements if they are only sub-optimal solutions.

Despite the apparent difficulty of the underlying GA optimiser in solving the 3 cases, in terms of all runs of each case converging onto the same final mass, it showed extreme robustness in detecting the optimal regions (considered here to be below the $0.29808kg$ level) as shown in Figure 5-9. The difficulty then arises due probably to the high-dimensionality of the problem themselves. Reduction in mass in the later stages, compared to the benchmark laminate, is no longer achievable only by the reduction of an arbitrary standard ply and change in fibre orientation, but in achieving a fine balance of buckling stability and minimum mass between regions A and B.

The best runs of all cases, i.e., those with the feasible lowest discrete mass, typically returned several candidate solutions with different genotypic structure. Despite this dissimilarity, the resulting decoded structure, the phenotype, are mainly identical to one another. This is not unexpected since it is possible for genotypes to have zero standard plies, and adjacent layers may be sharing the same fibre orientation. The resulting unique set of solutions, though, seem to indicate that genotypes for laminates with longitudinal voids respectively all ended as identical phenotypes, as shown in Table 5-7.

What is also of particular interest is that the solutions tend to be composed of mainly 90° for the continuous top and bottom layers, or A regions. On the other hand, the B regions, demarcating the longitudinal voids, have primarily 0°. Another particular property emerging from the best observed solutions indicate that the continuous A regions for the top and bottom layers have a small number of layers, in contrast to the B regions. This, in turn, causes the top and bottom layers to be spaced relatively far apart, as shown in Figure 5-10.

Case	Best Mass (kg)	Mass Savings (%)
Benchmark	0.29808	-
3 Longitudinal Voids	0.17315	41.9
4 Longitudinal Voids	0.13790	53.7
5 Longitudinal Voids	0.13349	55.2

Table 5-6 Best solutions, in terms of mass, returned for each of the 3 cases with longitudinal voids. The mass savings are relative to the benchmark laminate.

No. of Voids	Region A	Region B
3	$[\pm 45/90_3]_S$	$[0_{56}]_S$
4	$[90_3]_S$	$[0_{48}/-45_5/45_5]_S$
5	$[90_2]_S$	$[0_{63}]_S$

Table 5-7 Best solutions found from all runs of all cases of the laminate with longitudinal voids.

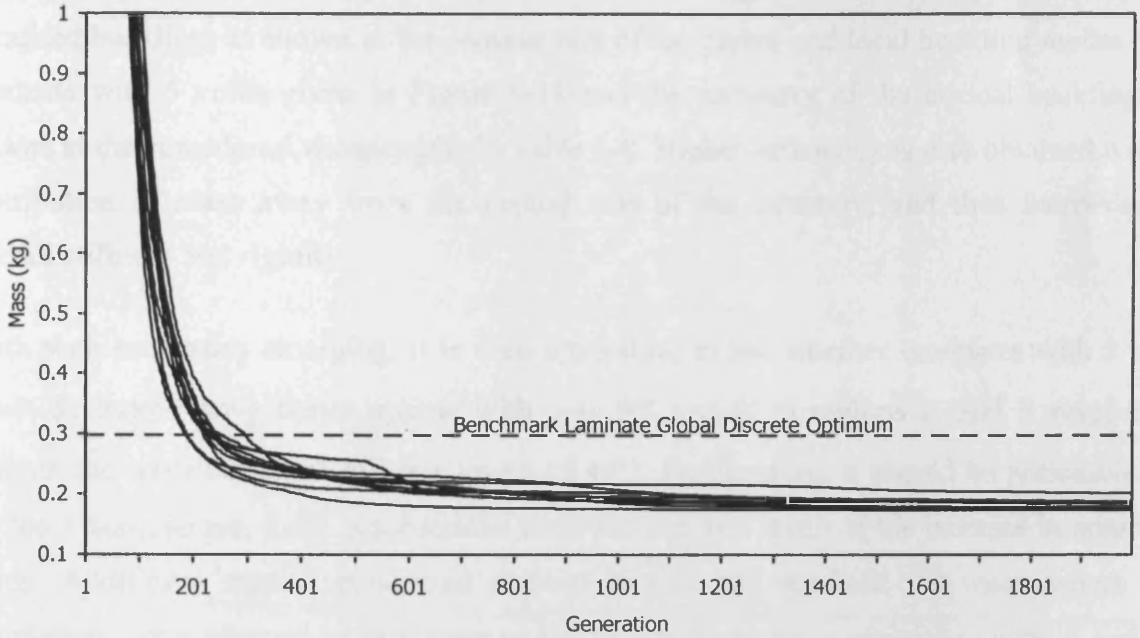

Figure 5-9 Population average mass of the ten runs from the case with 3 longitudinal voids. The same behaviour is also observed in the other two cases.

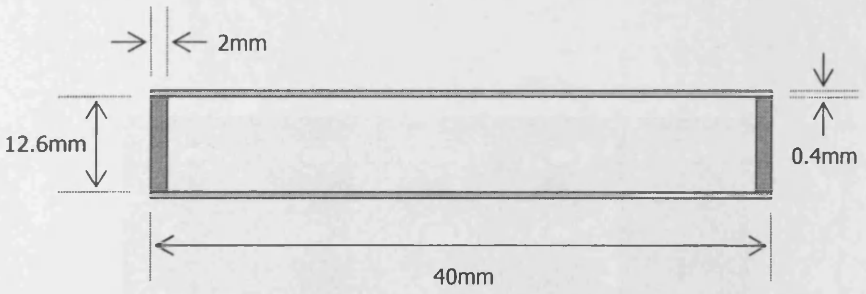

Figure 5-10 Scaled dimensions of a repeating element in the best solution of the laminate with 5 voids.

This behaviour can be attributed to the laminate configuration adjusting for Poisson's ratio, ν effects with 90° and 0° as the main load carrying component. Recall again that the laminate is loaded only with a single invariant uniform compressive longitudinal load of $100kN$. There may be other effects if additional loads are present. An almost guaranteed occurrence is the presence of transverse effects, while shear effects will only occur if there is non-orthotropic anisotropy[1] (since there is no shear loading). By having massive B regions, they effectively act as 'semi-rigid' supports spaced across the laminate. Thus, the overall buckling load is

[1] This conclusion can be drawn from the derivation of the stiffness coefficients leading to the constitutive equations of the individual plates (Wittrick and Williams, 1974).

increased. On the other hand, the top and bottom A regions have to cope primarily with localised buckling, as shown in the contour plot of the global and local buckling modes of the laminate with 5 voids given in Figure 5-11 and the summary of the critical buckling load factors at the considered wavelengths in Table 5-8. Higher efficiency is also obtained with the distribution of mass away from the neutral axis of the structure, and thus increasing the overall stiffness and rigidity.

With such properties emerging, it is then interesting to see whether laminates with 3 and 4 voids do indeed have better optima with only 90° and 0° in regions A and B respectively without the contribution of off-axis layers (±45°). Furthermore, it should be noticeable that, for the 3 cases above, there is substantial mass savings as a result of the increase in number of voids. Additional studies conducted showed that results obtained for cases where fibre orientations were allowed to vary were no worse than cases that restricted regions A and B to 90° and 0°, as shown in Table 5-9.

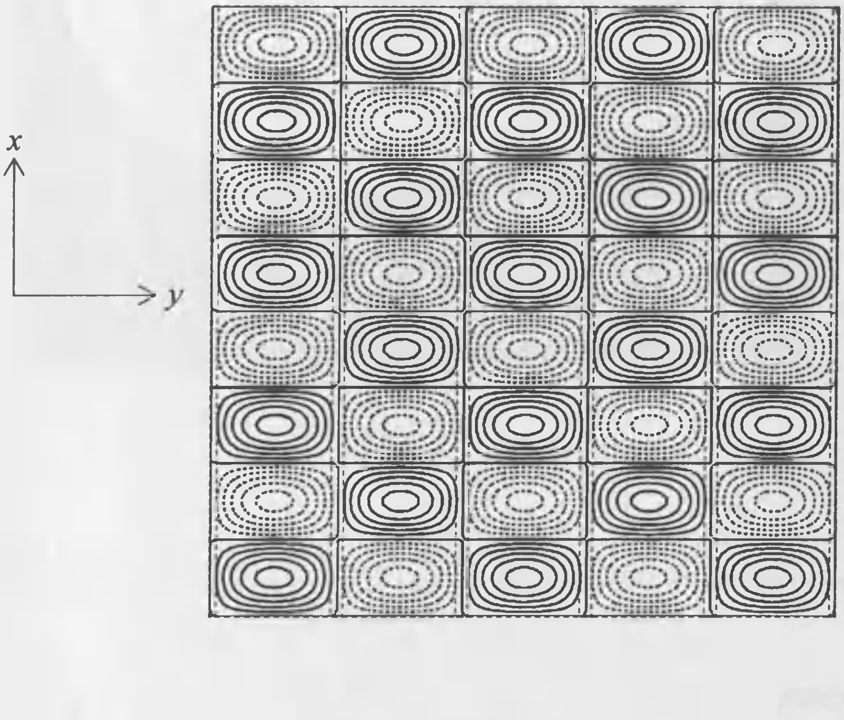

Figure 5-11 Contour plot of the governing critical buckling mode, which is local, for the laminate with 5 longitudinal voids. The nodal lines, parallel with at both transverse ends, indicate the accuracy of the VIPASA model in the absence of shear and anisotropy. See Appendix III for other modes of this panel.

Half-Wavelength λ	Critical Load Factor
ℓ	1.287440878
$\ell/7$	1.736238700
$\ell/8$	1.388920512
$\ell/9$	1.158034764
$\ell/10$	1.000327968

Table 5-8 Critical buckling load factor at several of the wavelengths considered for the laminate with 5 voids. The lowest critical buckling load is at the 10th wavelength, just satisfying the design loading.

No. of Voids	Region A	Region B	Mass (kg)	Mass Penalty (%)
3	$[90_5]_S$	$[0_{64}]_S$	0.17937	3.6
4	$[90_3]_S$	$[0_{72}]_S$	0.15241	10.5

Table 5-9 Optima for laminates with 3 and 4 voids if restricted only to 90° and 0° plies in regions A and B respectively. The mass penalty is compared to the best known optimum if stacking sequence is allowed to vary, as presented in Table 5-6.

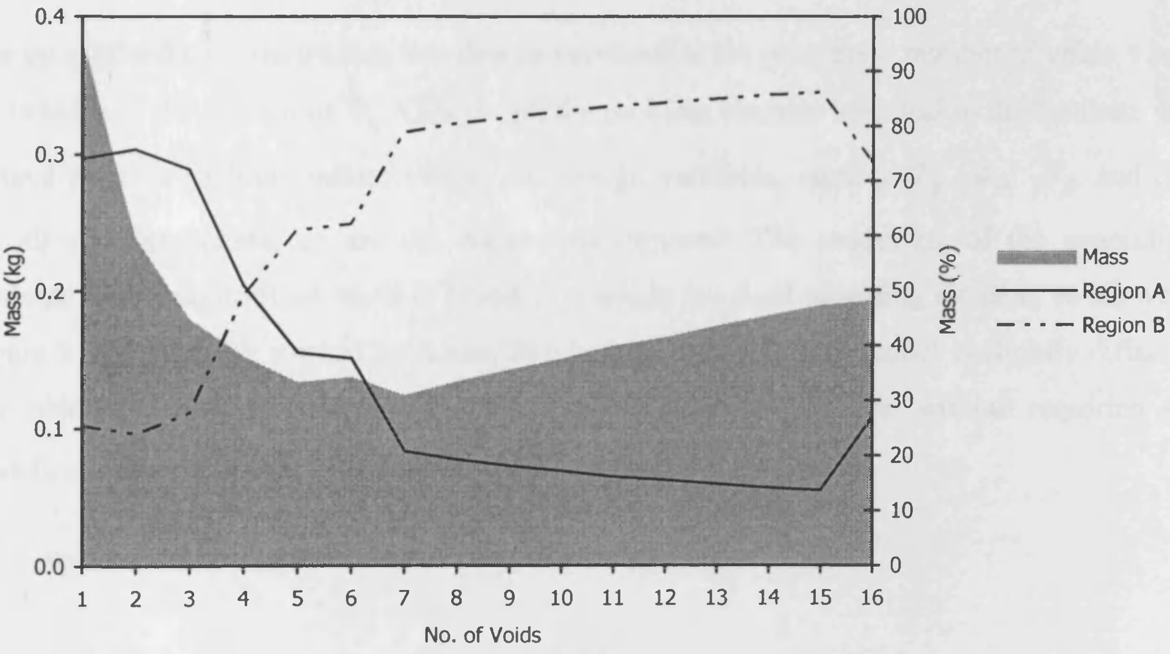

Figure 5-12 Overall global discrete optimum mass of laminates with different number of voids with regions A and B composing of only 90° and 0° plies respectively, and the distribution of mass between the regions.

However, results do show that there exist a corresponding optimum number of voids that results in the lowest global discrete optimum mass, as shown in Figure 5-12. Given the loading condition and geometrical dimensions of the laminate, the optimum number of voids is 7 resulting in a mass of $0.12390kg$. It can also be seen that as the number of voids increases, the distribution of mass is slowly biased towards B regions, and is so because the dimensions and distribution of the void regions make them less susceptible to localised buckling. However, it becomes more prone to overall buckling as the laminate becomes more uniform in the distribution of mass. It is worthwhile to note that region A of laminates with 7-15 voids took on the lower bound number of plies, i.e., $[90]_S$.

5.6 Generalised Laminate with Longitudinal Voids

Longitudinal voids have been shown to provide significant mass savings, although additional considerations must be taken into view in practical applications. Also, from results of laminates with a fixed number of voids, it is clear that the number of voids can clearly influence the optimum structural mass for buckling stability. This section proceeds to further formalise and investigate a generalised laminate with longitudinal voids in the hope of pushing the boundary of mass reduction.

5.6.1 Problem Definition

The generalised case introduces two design variables to the problem – number of voids v, and the breadth of the B regions b_B. Otherwise, the problem remains identical to the laminate with a fixed number of longitudinal voids. The design variables, namely θ_{Ak}, d_{Ak}, θ_{Bk} and d_{Bk}, are all still applicable, as are the constraints imposed. The modelling of the generalised laminate with longitudinal voids is based on a single standard repeating element, as shown in Figure 5-13, although it must be noted, like before, the analytical model is slightly different. The objective mass function $M(\mathbf{x})$ from Eq. 5-8 remains the same without requiring any modifications.

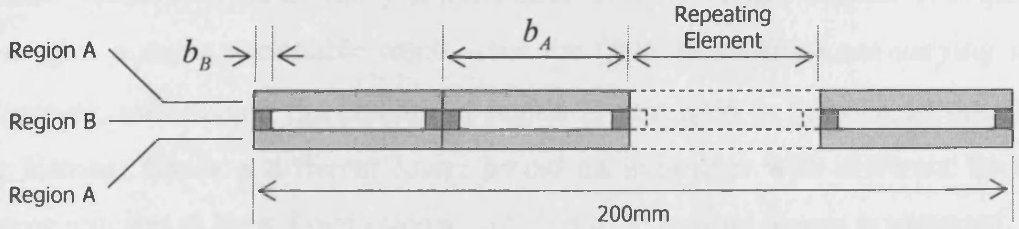

Figure 5-13 Schematic diagram of the generalised laminate with longitudinal voids.

5.6.2 Parameter Encoding

The number of voids parameter, v, is handled without any complications since it is an integer value bounded by the range $1 \le v \le 16$. This requires a 4-bit chromosome.

On the other hand, the encoding of the breadth of B regions, b_B requires clarification. Rather than a direct encoding of the design variable itself, the chromosome represents the fraction of the breadth to half the breadth of a repeating element. As shown in Figure 5-14, the physical constraint where the breadth of B regions cannot be more than half the breadth of the repeating elements has to be imposed. Otherwise, discrepancy in the modelling will arise. This naturally imposes the bounds on b_B such that

$$0 < b_B \le \frac{b}{2v} \tag{5-9}$$

where b in this case is, again, $200mm$. Since v is not constant, encoding b_B directly to a fixed bound will imply that for certain values of v, the values of b_B becomes illogical. Therefore, the decision was made to represent instead a normalised value, b'_B such that

$$b'_B = 2v\frac{b_B}{b} \tag{5-10}$$

In other words, b'_B represents the ratio of the breadths. The chromosome then encodes this value to a fixed range, where in this case, the bounds $0.10 \le b'_B \le 1.00$. One main reason for selecting a lower bound of 0.10 is that preliminary optimisation studies indicated that there is a tendency for this width to be reduced to as low a value as possible and taking the lower bound value. Therefore, the arbitrary lower bound of 10% of the breadth of region A was adopted to give a more reasonable result since the fibre orientations are varying in the *xy* plane. However, introducing the breadth of region B as a ratio to the overall breadth of the repeating element forces a different lower bound on laminates with different bounds. The chromosome requires at least 7 bits since a precision of 2 decimal places is expected.

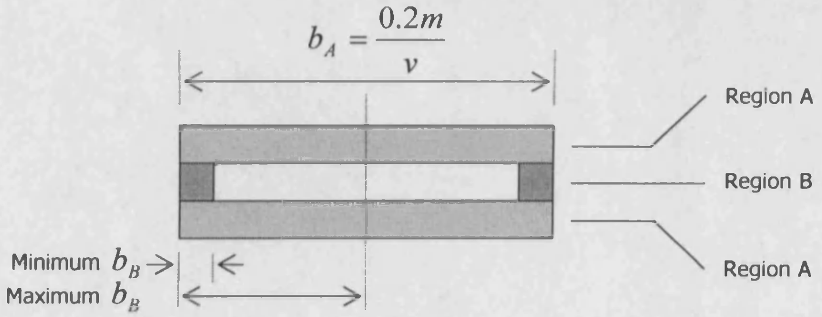

Figure 5-14 Constraints on the breadth of region B in a single repeating element.

The resulting genotypic structure is the extension of that from the laminate with a fixed number of voids plus the concatenation of the additional two design variables, resulting in a total of 155 bits. The problem has in excess of 4×10^{46} discrete potential solutions, although, admittedly, many genotypes points to the same phenotypic representation as before. Therefore, there is a certain degree of redundant representation in the overall encoding scheme.

As in the previous studies, the number of genotypes in the population reflects exactly the number of bits in a genotype, which is 156 and the additional genotype is to create an even number of individuals. Mutation is arbitrarily taken to be 8 bits in 155 and genes are forced to bit-flip. The same termination criterion is also adopted, i.e., a maximum of 2000 generations.

5.6.3 Results and Discussion

Ten independent runs were conducted where the starting conditions are seeded uniquely. It is important to note that the results obtained are not guaranteed to be the true global discrete optima of the problem. In fact, additional studies, discussed below, showed that they are not. Nevertheless, results again show substantial reduction in mass compared to the three cases of laminate with fixed number of voids. The best solution obtained indicates a mass of $0.107568kg$, where mass penalties from the best observed of previous studies are as shown in Table 5-10.

Despite the variability of genotypes sharing the best discrete mass, the redundancy of representation pointed to only a single configuration, which is $A = [90]_S$ and $B = [0_{49} / 90_{14}]_S$ with 8 longitudinal voids and the breadth of region B being $b_B = 0.00125m$, or 10% of half of

b_A, which is too small to be realistic. Hence, this justifies the reason for setting a lower bound of $4mm$ for the cases with fixed number of voids. As before, the main distribution of mass went into the B regions, where when compared to A regions, the distribution is in the ratio of $3.15:1$. Of course, this is not an indication of the stiffness distribution.

It should be noted that the number of plies in region A is effectively the lower bound imposed by the problem. In fact, additional studies restricting the stacking sequence of regions A and B to 90° and 0° shows that such is indeed the case for laminates with 8 to 16 voids. Results also confirmed that the best solutions obtained were highly optimal solutions as no laminates with lower mass were found, as shown in Figure 5-15.

Cases	No. of Voids	Mass (kg)	Mass Penalty (%)
Generalised	8	0.10757	-
Benchmark	0	0.29808	177.1
Fixed 3 Voids	3	0.17315	61.0
Fixed 4 Voids	4	0.13790	28.2
Fixed 5 Voids	5	0.13349	24.1

Table 5-10 Mass penalty incurred by the best observed feasible solution of other configurations of composite laminate compared to the best observed feasible solution from the generalised laminate with longitudinal voids.

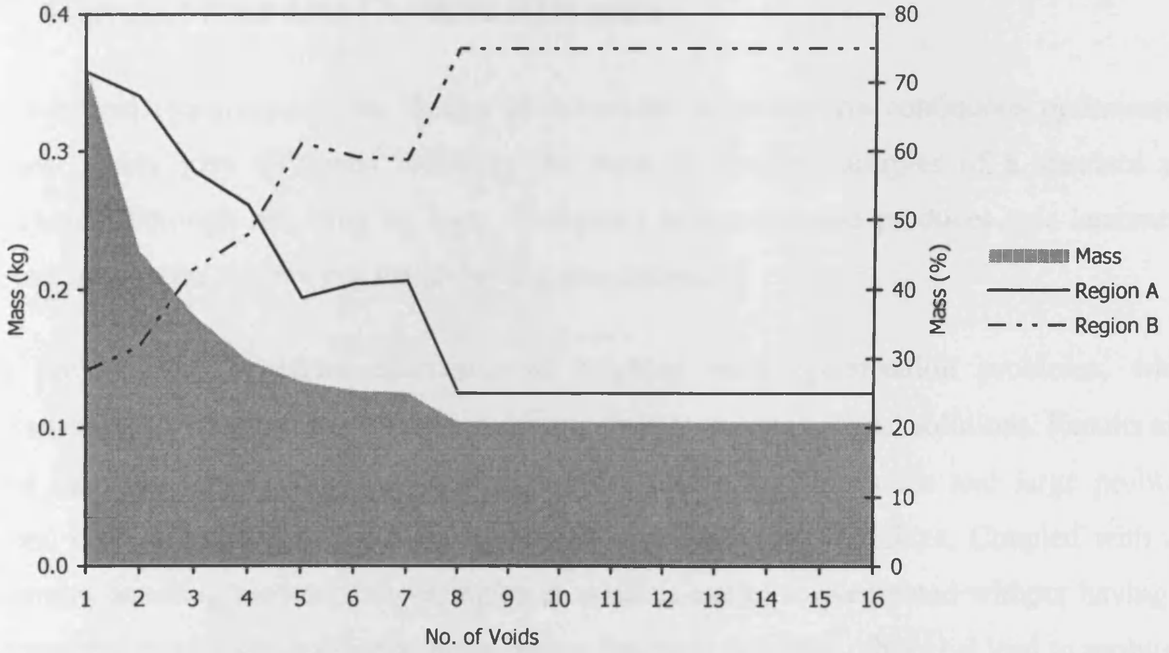

Figure 5-15 Overall global discrete optimum mass of laminate with different number of voids with regions A and B composing of only 90° and 0° respectively, and the distribution of mass between the regions. The breadth of region B is 10% of the half of the width of the repeating elements.

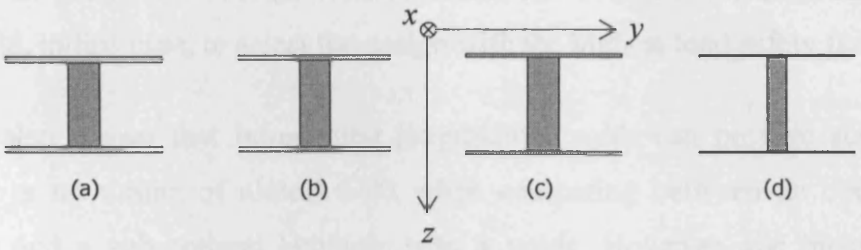

Figure 5-16 Scaled diagrams illustrating the breadth to thickness ratio of region B in the best obtained laminates with (a) 3, (b) 4, (c) 5 longitudinal voids, and the (d) generalised laminate with 8 voids.

One particular property emerging from laminates with longitudinal voids that has been previously overlooked, albeit intentionally, is the geometry of the B regions. As shown in Figure 5-16, the breadth to thickness ratio of the known optimum of the considered cases hardly justifies modelling them as a plate in the *xy* plane. In fact, the more accurate model should be as a plate in the *xz* plane.

5.7 Conclusions and Closing Remarks

For practical requirements, the design of composite laminates via continuous optimisation cannot readily give solutions satisfying the need of integer multiples of a standard ply thickness. Although rounding up layer thicknesses is possible and produces safe laminates, the resulting mass is often not the global discrete optimum.

GA provides an excellent alternative to handling such optimisation problems, while guaranteeing to a certain degree that results are optimal or near-optimal solutions. Results also show the robustness of GA in handling problems of high-dimensions and large problem spaces, thus allowing the inclusion of a wider range of design variables. Coupled with the constraint handling method, time-complex constraints can be implemented without having to undergo full evaluation and formulating fitness functions that may otherwise lead to problem-sensitivity.

Results from the benchmark laminate also show the ability of GA to detect multiple global optima even without the explicit use of niche induction techniques, granted that such optima were huddled close together and reachable using genetic operators imposed. Nevertheless, the ability of GA to discover multiple optima in a single design run is important, as it gives designers a wider selection of design candidates that can be further considered. For example, it may be useful, in this case, to select the design with the highest load safety factor.

Results have also shown that introducing longitudinal voids can provide substantial mass savings, up to a maximum of almost 64% when comparing between an optimal laminate without voids and a sub-optimal laminate with 8 voids. However, the inclusion of voids introduces other issues. Practical use of such laminates will inevitably have to consider additional detrimental effects particularly those relating to reduced damage tolerance and delamination. Nevertheless, results can also be interpreted as indicative of the minimum 'structural' mass required if such a configuration is used.

An important point that requires further emphasis is that the behaviour of plates with longitudinal voids is not captured by VIPASA. The plates, delineating sections of voids, are not thin plates that satisfy CPT assumptions. This is not a weakness of the VIPASA model, but rather a consequence of the structural modelling. Therefore, a more appropriate model of a box-section should be considered instead, which is presented in the next chapter.

Voids can be optionally filled with lightweight core materials that are intended to 'bulk up' the plate and introduce greater stiffness without excessive mass penalties. Such a design optimisation with core materials will be studied in the next chapter. Furthermore, optimal solutions also show that modelling of laminates with longitudinal voids is probably more accurate when treated as box-sections.

Despite the optimiser producing dimensions of webs (region B) that are not compatible with classical thin plate theory, which requires breadth-to-thickness ratio of individual plates to be more than or equal to a factor of 10, such a construction is still possible if precautions are taken. This statement is based on a panel concept known generally as the *orthogrid panel* (Niu, 1993), as shown in Figure 5-17, although variation in dimensions is applicable. Such panels are composed of narrow strips of composite and syntactic materials, modelled almost similarly to laminates with longitudinal voids as discussed above. Although orthogrid panels provide structural continuity in frame and stringer and the opportunity for low cost automated fabrication, they are difficult to splice and skin-grid bonding must be carefully checked. Furthermore, such panels are difficult to inspect for quality assurance and delamination damages.

Detailed studies conducted and presented above also showed that repeating sections can and should be optimised if greater mass reduction is desirable, as indicated by an optimum number of longitudinal voids in laminates. The conclusion that can be drawn is the same as that of other optimisation processes: The greater the number of design variables included in the optimisation, the likelier it is to discover solutions of better efficacy. The next chapter will move on to stiffened panel assemblies, which are used widely in a range of applications from aerospace to civil engineering construction.

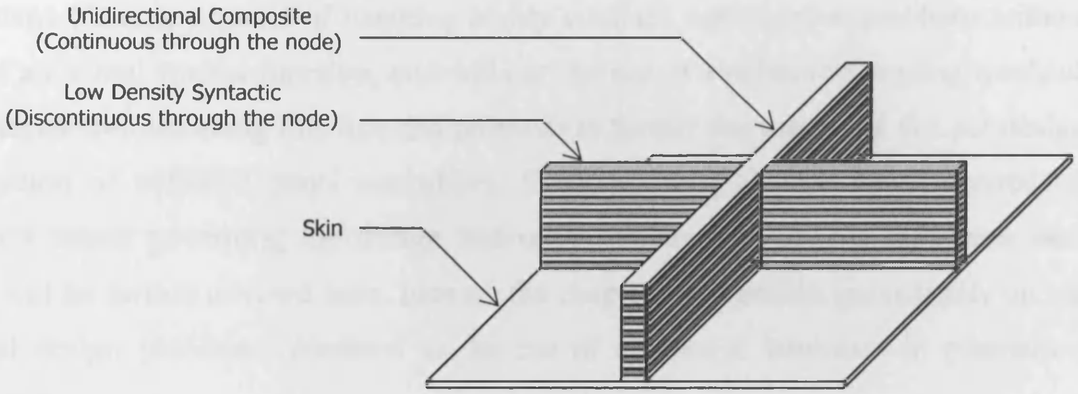
Unidirectional Composite
(Continuous through the node)
Low Density Syntactic
(Discontinuous through the node)
Skin

Figure 5-17 Orthogrid panel concept.

6 DESIGN OF STIFFENED PANEL ASSEMBLIES

The previous chapter has shown that the constraint handling technique proposed for Genetic Algorithms (GA) is capable of handling highly complex optimisation problems without the need of an actual fitness function, and without the use of stochastic sampling mechanisms. This chapter follows along this line and proceeds to further the use of the GA for design and optimisation of stiffened panel assemblies. Since previous chapters have covered all the necessary issues governing the design and optimisation of composite laminates, no such details will be further covered here. Instead, the chapter concentrates immediately on various practical design problems, common to the use of composite laminates in prismatic plate assemblies.

6.1 INTRODUCTION

Prismatic plate assemblies are structures with panels of longitudinally invariant cross-sections, such as those commonly used in the aerospace industry. With the advent of exact analytical methods and optimisation methodologies capable of handling vast problem spaces, it is now possible to explore difficult optimisation problems previously intractable with traditional methods with more realistic modelling of the structure.

Following results from the previous chapter, it has been shown that, despite apparently difficult optimisation problems of high-dimensionality and time-complex constraints, GA and the use of the proposed set of constraint handling were capable of traversing vast solution landscapes and home in onto regions of optimality. By including more design variables into the optimisation process and introducing additional features, the required structural mass of a simplistic laminate was reduced substantially by more than 50%. Nevertheless, results show that the resulting dimensions of a laminate with longitudinal voids were inconsistent with assumptions made by the underlying analytical model that is based on classical thin plate theory.

Ever since the advancement in the exact classical thin plate theory by Wittrick and Williams

(1974), more practical design applications have been studied and researched, such as those performed by a GARTEUR (Group for Aeronautical Research and Technology in Europe) Action Group on Structural Optimization (York et al., 1993) and by Kennedy et al., (1999). It should also be mentioned that such works used the more sophisticated VICON analytical model, as opposed to the more conservative VIPASA model. Although the latter model is used here, it does provide more conservative results in the design studies presented in this section.

It should be obvious by now that any analytical model, or indeed any design and optimisation problems, can easily substitute the model in use without the need for additional modification to the GA framework. Despite this conservatism of the VIPASA analytical model, which can affect the results to a significant degree, the same model is shared by all test cases conducted, and hence should not consider as invalidating the results and conclusions presented hereafter.

6.2 Box-Section Analogy of Laminate with Longitudinal Voids

As shown in the previous chapter, the introduction of longitudinal voids has been shown to provide substantial mass savings. However, it resulted in geometrical dimensions inconsistent with the assumptions in the case of classical thin plate theory, particularly in the 'web' regions. This section attempts to solve this discrepancy in the modelling by designing a box-section in compression. The dimensions and other design environment variables will be made to follow as closely as possible the laminate with longitudinal voids design problem, as described previously.

As before, all layer thicknesses have to conform to the discrete thickness constraints. This provides an advantage over continuous optimisation, in the sense that it does not require an additional round of optimisation process to arrive at the discrete design solution, which, for reasons described previously, may or may not be the actual global discrete optimum. The balance and symmetry constraints are applied without any modifications, and will have to be satisfied by all panels in the resulting structure.

The modelling of the box-section considered here is similar to that for laminates with longitudinal void elements. Repeating elements will be used to construct the overall resulting structure as shown in Figure 6-1. Consideration of skin and web offsets is also considered to provide a more accurate representation during production of such a structure. The support conditions for the box structure are as shown in Figure 6-2, which has a significant difference compared to the case of laminates with voids.

The skin and web are allocated a set of design variables similar to the laminate with longitudinal voids problem, where θ_{sk} and d_{sk} represent the fibre orientation and integer number of standard plies of the k-th layer, and a corresponding θ_{wk} and d_{wk} for the web. The problem is therefore the design of the optimum stacking sequence and discrete thicknesses in terms of mass subjected to a compressive design load.

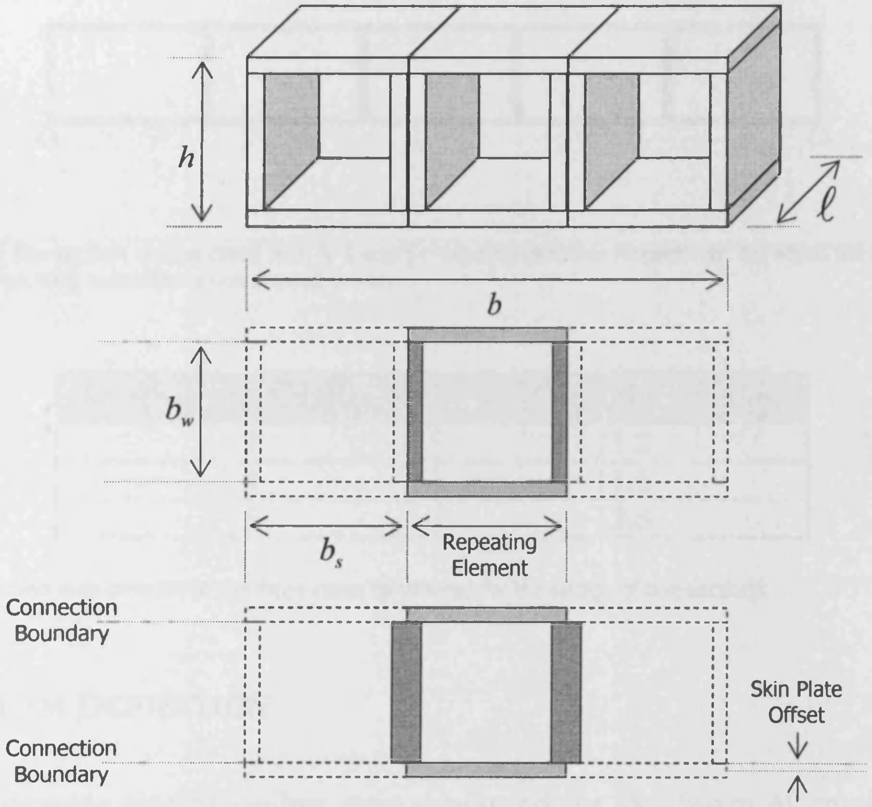

Figure 6-1 Typical box-section with 3 voids. The problem can be simplified into repeating elements as shown. The VIPASA analytical model is as shown in the bottom diagram where the skin plate offsets are taken into consideration.

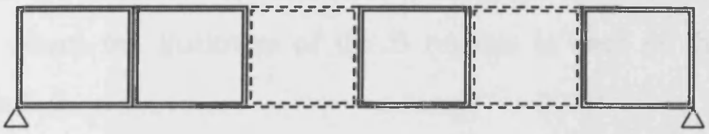

Figure 6-2 Support conditions for an arbitrary box-section. The simply-supported ends prevent displacement in the three principal directions but allow rotation about the longitudinal axis.

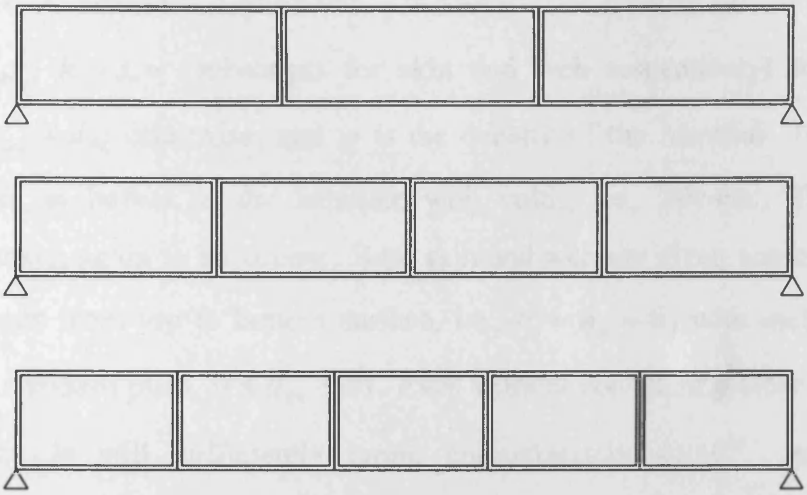

Figure 6-3 Box-section design cases with 3, 4 and 5 repeating elements respectively, for which the breadth of the overall resulting assembly is not allowed to vary.

Number of Elements, v	Web Breadth, b_w (mm)
3	11.2
4	11.6
5	12.6

Table 6-1 Fixed web breadth of the three cases considered for the design of box-sections.

6.2.1 Problem Definition

Firstly, three separate sets of studies were conducted for the design of compression box-section members. Each set considers a box-section with differing number of repeating elements, namely 3, 4 and 5, as shown in Figure 6-3. In this set, the breadth and depth of the individual elements are fixed, where the breadth of the skin is defined by the relationship, $vb_s = b = 200mm$, where v is the number of repeating elements. The breadth of the web b_w, on the other hand, is set to reflect the best known solutions from laminates with 3, 4 and 5 voids respectively, where the thickness of the B regions is used as the breadth and is as summarised in Table 6-1.

Following the above definitions, the objective mass function $M(\mathbf{x})$ can therefore be formulated, which is a straightforward process. Assuming that the skins and webs of the repeating elements are allowed a maximum of n_s and n_w layers respectively, a general box-section with v number of repeating elements, therefore, has a mass $M(\mathbf{x})$ of

$$M(\mathbf{x}) = 2t_L \ell \rho v \left(\left(b_s \sum_{k=1}^{n_s} c_{sk} \right) + \left(b_w \sum_{k=1}^{n_w} c_{wk} \right) \right) \qquad 6\text{-}1$$

where $c_{Rk} = 2d_{Rk}$, $R = s, w$ (subscripts for skin and web respectively) for off-axis fibre orientation, or $c_{Rk} = d_{Rk}$ otherwise, and ρ is the density of the material. The length of the section ℓ here is as before in the laminate with voids, i.e., $200mm$. The standard ply thickness, t_L is taken again to be $0.1mm$. Both skin and web are given a maximum of 6 fibre orientation changes from top to bottom surface, i.e., $n_s = n_w = 6$, with each layer having a maximum of 31 standard plies, $0 \le d_{Rk} \le 31$. Even without considering other design variables, the search space is still sufficiently large, approximately 4×10^{12} , to prevent from degeneration into a trivial optimisation problem.

The second set of studies concentrates on extending the above design problem by introducing the breadth of web as a design variable. The design bound imposed on the variable is $0.1mm \le b_w \le 204.8mm$, where precision up to 1 decimal place is expected. Again, three different cases will also be considered by restricting the number of elements to 3, 4 and 5.

The third and final set of studies incorporates the design variables from the second set and adds the number of elements v design variable. By doing so, it is hoped that an optimum number of repeating elements balanced by the stacking sequences, discrete thicknesses and breadth of the skin and web can be discovered, resulting in even greater mass reduction compared to the first and second set. The rationale behind this decision is due to the observations made from laminate with longitudinal voids, showing that there exist an optimum number of voids corresponding to a global discrete optimum mass. The bounds imposed on this additional design variable is $1 \le v \le 16$.

For all cases, the assembly is expected to be stable under a single loading event of uniform longitudinal compressive load of $100kN$, with no additional transverse or shear loading. A total of the first 10 half-wavelengths are considered, as before.

6.2.2 GA Properties and Setup

Generally, the genotypic representation for the fibre orientation and layer discrete thickness remains the same as that for the laminate with voids, presented in the Chapter 5. Fibre orientation and d_{Rk} requires a 2-bit and 5-bit chromosome respectively for each layer. With only half the number of layer changes (3 for skin and 3 for web) needs to be modelled, the resulting genotype has a total of 42 bits, which is also same genotype for the first set of studies since no additional design variables are included.

In the second set of studies though, the additional design variables of b_w are encoded in the standard Gray-encoding for floating-point values. The bounds imposed on the breadth of the web require at least a 11-bit chromosome in order to model also the required precision, resulting in a genotype with 53 bits.

The third and final set of studies simply concatenates the additional number of elements design variable chromosome into the genotypic structure from the second set. The bounds imposed on the number of elements is $1 \leq v \leq 16$, which requires an extra chromosome with 4 bits, thereby producing a genotype with 57 bits.

The Pareto-Ranking (PR) constraint handling technique will be used with artificial prioritisation and lazy evaluation, i.e., GAPR-PLZ. The model requires no extra formulation of a fitness function, but requires instead the use of sampling mechanisms that need no such fitness value. The fitness, in this case, is the measure of dominance first in terms of satisfaction of the buckling constraint, and then finally in terms of mass. The binary tournament selection is the selection mechanism used.

As before, the 0.3 α-uniform crossover recombination operator is deployed, complimented by the standard uniform mutation. The probability of crossover is fixed at 0.8 throughout the search process, while the mutation rate will be representative of the number of bits in the resulting genotype, taken here to be 1 bit-flip in 42 for all design sets and cases. The steady-state replacement strategy is used, where the top 50% performing individuals are selected to be the new population of the next generation. The GA plan used here is identical to that used in the design and optimisation of laminates, where the population size is the number of bits in the genotypes of individual cases rounded to the nearest even integer value.

6.2.3 Results and Discussion

For the first set where only the stacking sequence and discrete thicknesses of the skin and web are design variables, ten independent runs were conducted for each case and the termination criterion being either a maximum of 2000 generations or when all genotypes share the same mass. The starting populations for the i-th run were always identical. Table 6-2 gives the best solutions returned from the runs on the design of box-sections.

v	Mass (kg)	Mass Savings	Skin	Web
3	0.18600	37.6%	$[\pm 45/45_2/-45_2]_S$	$[90/0_6]_S$
4	0.15780	47.1%	$[\mp 45_2]_S$	$[0_9]_S$
5	0.14308	52.0%	$[90_3]_S$	$[0_8]_S$

Table 6-2 Best unique solutions found for box-sections with 3, 4 and 5 repeating elements respectively and their corresponding discrete mass. Mass savings are compared to the benchmark laminate in Chapter 4.

v	Skin ΔM	Web ΔM
3	83.6%	16.4%
4	65.7%	34.3%
5	54.3%	45.7%

Table 6-3 Distribution of mass between skin and web for the best known solution of the box-sections with fixed web breadth given earlier.

As expected, all showed substantial mass savings over the benchmark laminate spanning the same breadth and length and designed to the identical buckling constraint. This confirms the fact that even for plates spanning small distances, modelling additional features such as the box-section can provide substantial mass savings as a result of the redistribution of mass, and hence stiffness, to cope more efficiently with the overall and local buckling modes. Of course, deciding whether the mass saving justifies the additional fabrication complications and costs is another optimisation process altogether.

It should be apparent that, as in the results for laminates with voids (see Chapter 5), there is a trade-off between the distribution of mass in the skin and the web as the number of elements varies, as shown in Table 6-3. As the number of elements increases, the distance between the supports provided by webs along the top and bottom skin plates decreases and reduces the likelihood of overall buckling, and the structure begins to optimise for local modes. Additional studies showed that that such is the case for a box-section where the skins are composed of only 90° and the web of only 0° with fixed web breadth, as shown in Figure 6-4.

Despite the relatively similar dimensions between laminates with voids and box-sections with identical number of repeating elements, such as that shown in Figure 6-5, the necessary increase in weight of the latter can be attributed to a more accurate modelling and therefore a better analysis of the buckling stability of the structure. Nevertheless, the distribution of fibre orientations seems to indicate a similar pattern as that of laminates with voids. In all three cases, the majority of 0° plies are located in the web region with most of the $\pm 45^\circ$ and 90° in the skin. The same explanation made previously is thus applicable here too.

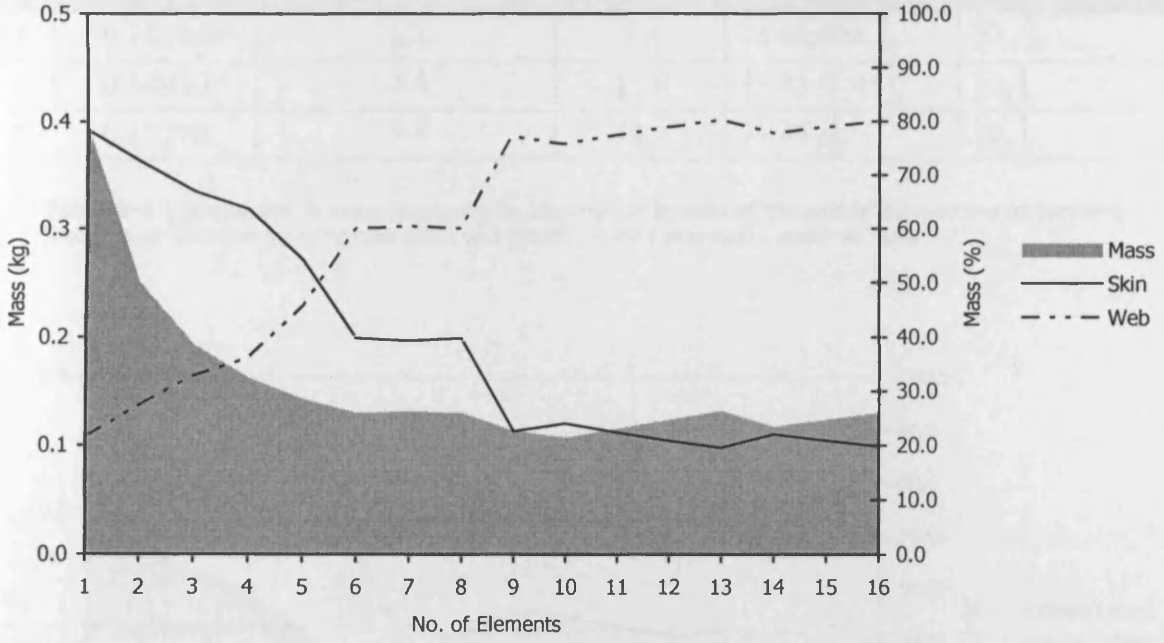

Figure 6-4 Distribution of mass between skin and web of a fixed-breadth box-section as number of elements increases with standard stacking sequence. Both skin and web were not allowed to exceed 126 plies, with skin restricted only to 90° plies and web only to 0°.

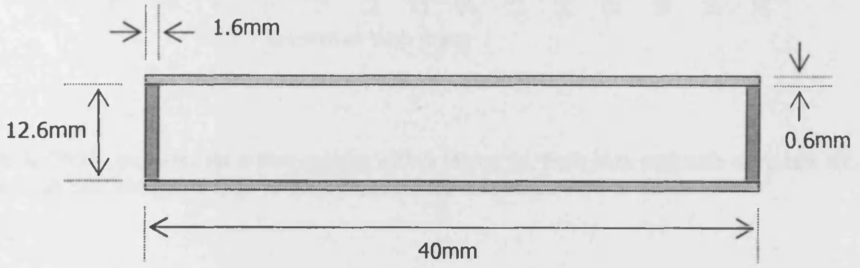

Figure 6-5 Scaled diagram illustrating the dimensional relationships of the best solution of a repeating element of the box-section with 5 elements.

However, the second set of experiments with varying web breadth did not show significant improvements in terms of mass, with the best found solutions changing little either in its overall stacking sequence or even web breadth, as summarised in Table 6-4, except for the box-section with 4 elements. Even for this latter case, mass savings were only around 5.4%. Nevertheless, what it confirms is that the breadths chosen initially for cases with fixed dimensions were clearly close to the optimum, which, in turn, seems to suggest that designing for laminates with longitudinal voids gave almost optimum dimensions. This is, of course, an unconfirmed generalisation that might not hold up for other cases.

v	Mass (kg)	Mass Savings (%)	b_w (mm)	Skin	Web
3	0.18384	1.2	9.3	$[\pm 45/90_3]_S$	$[0_{15}]_S$
4	0.14930	5.4	11.0	$[\pm 45/\mp 45]_S$	$[0_8]_S$
5	0.13770	3.8	10.5	$[\mp 45_2]_S$	$[0_5]_S$

Table 6-4 Improvement in mass discovered by allowing the breadth of the web of box-sections to vary and savings over the corresponding case when web breadth is fixed as provided earlier in Table 6-2.

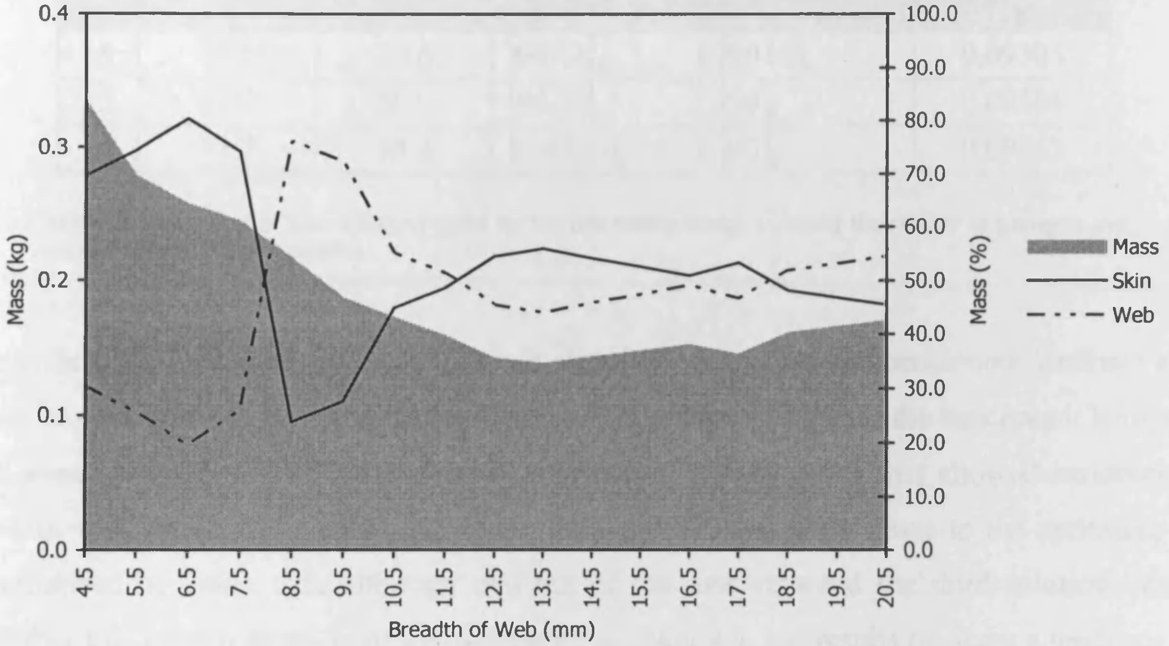

Figure 6-6 Optimum mass for a box-section with 5 elements. Both skin and web were not allowed to exceed 126 plies, with skin restricted only to 90° plies and web only to 0°.

Additional studies were performed on a box-section of 5 elements and restricting top and bottom skin plies to 90°, and web plies to 0°, summarised in Figure 6-6. Results indicate that unless care is taken in the selection of an appropriate value for the breadth of webs, mass penalties relative to the optimum breadth value can be excessive. Putting it in another way, the inclusion of more design variables into the design optimisation process can produce highly efficient solutions, previously difficult or even impossible with more traditional optimisation methodologies. In this second set of studies, 13 design variables were considered in the GA optimisation process, and it is probably fair to say that despite the large search space, the deployed optimiser was sufficiently capable of finding optimal or near optimal solutions.

The final set of studies included also the number of elements in the box-section as a design parameter. Results show impressive mass savings from a set of 10 runs conducted, where every run began with a unique population. The best solution found had only a weight of $0.09305kg$, which is the first solution ever found below the $0.1kg$ threshold for a structure of the given dimension, loading and buckling constraint. The solution is fairly simple, where the skin and web both have the setup $[90/0]_S$, with 15 repeating elements and a web breadth of $0.0106mm$. Therefore, each repeating element is almost square in dimensions of the structure, although it should be mentioned that the solution is not a confirmed optimum.

v	b_s (mm)	b_w (mm)	**Skin**	**Web**	**Mass** (kg)
15	13.3	10.6	$[90/0]_S$	$[90/0]_S$	0.09305
15	13.3	9.3	$[90/0]_S$	$[90]_S$	0.09584
14	14.3	10.4	$[\pm 45/0]_S$	$[90]_S$	0.09663

Table 6-5 Selection of best solutions found for the box-section design including the number of elements and web breadth as design variables.

Nevertheless, substantial mass savings can already be seen over the benchmark laminate and other box-section designs obtained earlier, by as much as 68.8% over the benchmark laminate and even as much as 32.4% over the box-section with 5 elements that allowed variation in breadth web. In fact, the runs also found other solutions that are close to the optimum, as summarised in Table 6-5, although two out of ten runs returned the third solution listed. Whether this is an indication of multimodality is unknown, but results do show a tendency to converge onto this solution in the ten runs conducted. Even if the landscape is noisy and fraught with local optima, this is unsurprising given the number of design variables up for optimisation.

Admittedly, the problem is fairly difficult to solve given the GA framework. It should, however, be said that there is no problem in detecting regions of high-optimality given the large search space, but difficulty arises hereafter. It is probably fair to say that most, if not all, GA frameworks suffer from the same problem in the sense of homing in onto actual optimum.

6.3 Discrete Thickness Design of Stiffened Panels

This section proceeds to a practical design problem now commonly known as the GARTEUR benchmark problem as set out by York et al., (1993). The design is that of the components of Dornier wing, which are stiffened panels, but have mainly been confined to a continuous optimisation problem. Thus, results presented required variables to be made discrete in order to comply further with common manufacturing practices. Two main plates were considered that consisted of the component panels in the wing with lowest and highest maximum principal in-plane stresses, denoted P2 and P8 respectively. However, only panel P8 will be considered here since modelling panel P2 involves only a change of geometry and loading conditions. Loading conditions for panel P8 are as given in Table 6-6, and all cases must be satisfied by the panel. Whereas the transverse and shear loads are constant and considered only to be carried in full by the skin, the longitudinal loads are redistributed between the skin and stiffeners so as to give a uniform longitudinal strain over the panel cross-section. This calculation, thus, has to be done for each candidate solution. The geometry of the panel is such that the breadth and length is $b = 0.610m$ and $\ell = 0.752m$ respectively.

Panel	Load Case	Stress Resultants (kNm^{-1})		
		Longitudinal	Transverse	Shear
P8	1	2221.10	378.40	-643.70
	2	2795.40	240.60	-187.70
	3	1303.30	268.70	-420.70

Table 6-6 Loading conditions considered for the GARTEUR benchmark stiffened panel design with the highest maximum in-plane stress resultants, P8, in a Dornier wing.

In the literature mentioned earlier, GARTEUR covered widely the set of panel design variables in a continuous optimisation problem, which will not be attempted again here. Inevitably, some of the design variables will overlap.

Although most structures justify the use of quasi-isotropic laminates, such an approach mitigates the primary advantage of directional properties of composites. Since no universal configuration can effectively satisfy the myriad loading conditions and stiffness requirements (buckling, static deformations, flutter, etc.), simultaneous optimisation of stacking sequence to determine the lamination geometry can be advantageous, albeit computationally expensive. Therefore, contrary to the design problem posed by GARTEUR, the stacking sequences here are allowed to vary. The fibre orientations are still restricted to the same set used in the preceding chapter, i.e., 0°, 90° 45° and -45°, and off-axis layers will be composed of two standard plies that seek to balance the laminate. All designs are immediately expected to satisfy the discrete thickness requirement, and therefore do not first undergo a continuous optimisation process. Stacking sequences in all component plates are also expected to be symmetric, with stiffeners dividing the skin into equal portions. Two stiffener types are considered in this design process – blade and flanged stiffeners.

6.3.1 Problem Definition

Two separate design sets were considered following the types of stiffeners considered. Each set was broken down further to accommodate differing number of stiffeners, as shown in Figure 6-7. The first set considers blade-stiffened panels, with 3, 4 and 5 number of stiffeners, and the panels denoted as I3, I4 and I5 respectively. Similarly for flanged-stiffened panels, three cases each with 3, 4 and 5 number of stiffeners were considered, with the panels denoted J3, J4 and J5 respectively.

The simply-supported longitudinal ends prevent in-plane and out-of-plane displacements, but allow rotation about the longitudinal axis. The difference is that the original GARTEUR optimisation prohibits only the out-of-plane and transverse in-plane skin displacements. Nevertheless, it was reported that allowing only rotation gave results with 3.8% lower mass. The analytical model of VIPASA also assumes simply supported ends, which may be violated in this case as a result of the presence of shear load. It should be noted that the GARTEUR work used the more accurate VICON analysis.

Stiffener heights b_w for the blade-stiffened panels were measured from the mid-plane surface of the skins, implying therefore that no plate offsets were considered. On the other hand, b_f for flanges of flanged-stiffened panels were also measured from the mid-plane surface of the webs. This connection is true for all considered cases, as shown in Figure 6-8, which is the same as that employed in the parametric studies conducted by GARTEUR.

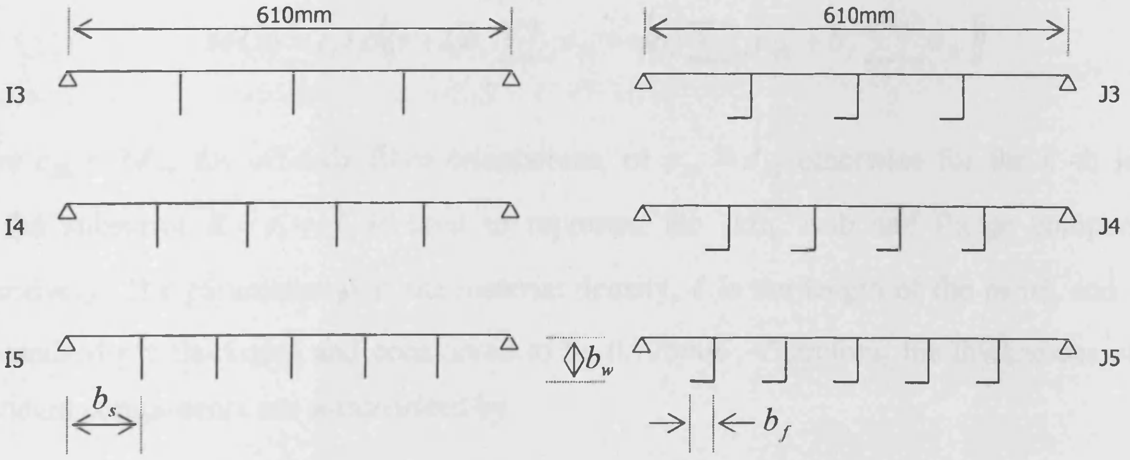

Figure 6-7 Three cases conducted for each set of stiffened panel assemblies.

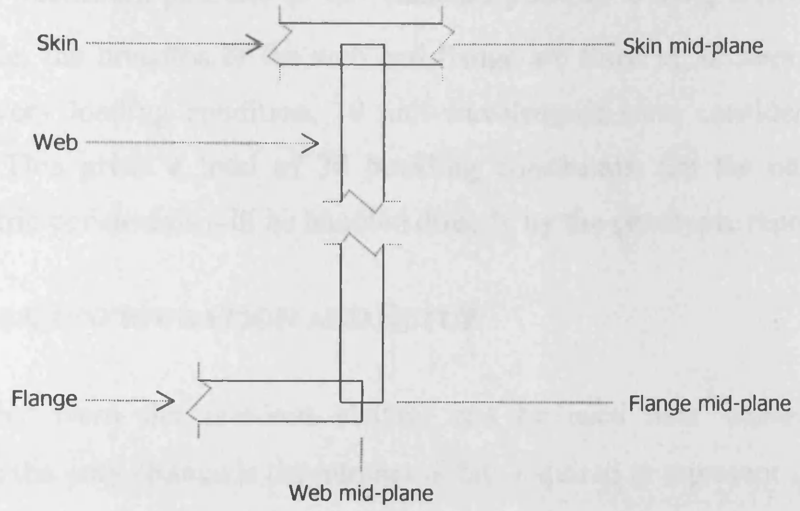

Figure 6-8 Components of a stiffened panel and connection boundaries considered in the design. The flange is not applicable in the case of blade-stiffened panels.

Given the above definitions, the objective mass function $M(\mathbf{x})$ can therefore be formulated. Without loss of generalisation, let v denote the number of stiffeners that divide the skin equally into equal portions with breadth b_s. Also, let b_w and b_f denote the breadth of the web and flange respectively, where in the case of blade-stiffened panels, $b_f = 0$. Assuming that the skin, web and flange have n_s, n_w and n_f number of layers respectively, the mass of a stiffened panel assembly is therefore

$$M(\mathbf{x}) = t_L \ell \rho \left((v+1) b_s \sum_{k=1}^{n_s} c_{sk} + v \left(b_w \sum_{k=1}^{n_w} c_{wk} + b_f \sum_{k=1}^{n_f} c_{fk} \right) \right) \tag{6-2}$$

where $c_{Rk} = 2d_{Rk}$ for off-axis fibre orientations, or $c_{Rk} = d_{Rk}$ otherwise for the k-th layer, and the subscript $R = s, w, f$ is used to represent the skin, web and flange components respectively. The parameter ρ is the material density, ℓ is the length of the panel, and t_L is the standard ply thickness and considered to be $0.125mm$. Therefore, the thicknesses of the individual components are summarised by

$$t_R = t_L \sum_{k=1}^{n_R} c_{Rk} \tag{6-3}$$

Each component is allowed a maximum of 6 fibre orientation changes, i.e., $1 \le n_R \le 6$, with each layer having a maximum possible of 127 standard plies, or $0 \le d_{Rk} \le 127$. In all cases and where applicable, the breadths of the web and flange are fixed at 32.5*mm* and 20.0*mm* respectively. For every loading condition, 10 half-wavelengths were considered, as in the preceding chapter. This gives a total of 30 buckling constraints. On the other hand, the balance and symmetric constraints will be handled directly by the genotypic representation.

6.3.2 Optimiser Configuration and Setup

The design variables from the previous chapter can be used here without significant alterations, although the only change is the number of bits required to represent d_{Rk}, which, in this case, is 7 bits[1]. Therefore, each layer is correctly modelled by two chromosomes, with one representing the ply angle θ_{Rk} and another the number of integer standard plies d_{Rk}. Each component is thus fully represented by (9×3) 27 bits, where the blade-stiffened panel has a genotype with 54 bits and the flange-stiffened panel with 81 bits.

As before, the Pareto-Ranking (PR) approach with prioritisation and lazy evaluation is used as the constraint handling technique without specifically formulating a true fitness function. Therefore, the binary tournament selection is again used here without modifications, with constraints violation being the most prioritised objective over objective cost, i.e., mass. This then implies that the primary aim of the underlying GA is to first discover feasible regions before optimising based on the actual objective mass function value.

There are 30 constraints considered here since there are 3 load cases with each having to be evaluated against 10 buckling modes. This gives a virtually insurmountable number of constraint evaluations without artificial prioritisation and lazy evaluation. Therefore, the constraints are prioritised according to the sequence of the loading cases as presented above, and beginning first at the lowest half-wavelength. The violation is, therefore, the number of eigenvalues exceeded, with constraint evaluation terminated once a genotype is unfeasible at a particular load case and buckling mode. As before, this method will not provide the GA optimiser with a true account of the number of exceeded eigenvalues across all combinations of load and wavelength, which again leads to partial knowledge of violation in most cases.

[1] The number of bits required to represent the ply angles is still 2. See previous chapter for details.

The population size follows the number of bits represented in the genotype of each case rounded to the highest even integer, with the sampling mechanism selecting this constant number of genotypes at every generation. The 0.3-uniform crossover is again used with probability of crossover at 0.8, complimented by the uniform mutation at a rate of 1 in 27 bits. The replacement strategy merges the sampled genotypes with the parent population and selects up to population size number of top performing individuals to be the new set of parents. The process repeats itself until termination criterion, which, in this case, is either a maximum generation of 2000, or when all genotypes share the same mass value, whichever is satisfied first.

6.3.3 Results and Discussion

There are a total of six independent cases, with three for blade-stiffened panels and three for flanged-stiffened panels, and each undergoing ten independent runs. The i-th run of each case of each panel type has an identical starting population. The best found solution for each of the six cases is as summarised in Table 6-7, although it should be noted that they are not confirmed optima.

Case	Mass (*kg*)	Configuration		Mass (%)	Thickness (*mm*)
I3	7.95511	S	$[90_6/-45_8/45_8/0_4]_S$	60.7	6.50
		W	$[45_4/-45_4/0_{64}]_S$	39.3	18.00
I4	8.34800	S	$[90_5/-45_7/45_7/0_6]_S$	55.6	6.25
		W	$[0_4/45_5/-45_5/0_{50}]_S$	44.4	16.00
I5	8.10663	S	$[90_5/-45_9/45_9]_S$	52.7	5.75
		W	$[45_3/-45_3/0_{47}]_S$	47.3	13.25
J3	6.64398	S	$[90_9/-45_7/45_7]_S$	64.3	5.75
		W	$[90_4/-45_7/45_7]_S$	11.8	4.50
		F	$[-45_5/45_5/0_{77}]_S$	23.9	21.75
J4	6.70641	S	$[90_7/-45_7/45_7/0]_S$	60.9	5.50
		W	$[-45_{10}/45_{10}]_S$	17.3	5.00
		F	$[-45_2/45_2/0_{56}]_S$	21.8	15.00
J5	6.83966	S	$[90_5/-45_4/45_4/0_9]_S$	59.8	5.50
		W	$[90/-45_8/45_8]_S$	18.0	4.25
		F	$[0_{50}]_S$	22.3	12.50

Table 6-7 Best solution found for each case of blade-stiffened panels (I3 – I5) and flange-stiffened panels (J3 – J5). The rows S, W and F stand for skin, web and flange respectively, showing their final stacking sequence, mass distribution in the overall assembly and overall thickness.

The GA optimiser found relatively efficient solutions to the six cases considered despite its relative naivety of true constraint violations throughout the run. Furthermore, only simple and well-studied genetic operators such as uniform crossover and uniform mutation were used. Despite no parameter tunings being performed prior to the conducted experimentations, results were highly robust and efficient. It is, therefore, possible that solutions, convergence rate, reliability, and other factors, can be expected to improve if more detailed fine-tuning of parameters were performed.

The underlying constraint handling technique, which does not allow the use of standard stochastic-based selection mechanism, forced the use of ranked-based sampling techniques and this led to the use of the tournament selection. The advantage here is the lack of a need to couple constraint violations to the objective function and, therefore, produce a proper fitness function, as is often the case in penalty-based approaches to constraint handling. The Pareto dominance comparison of violations as primary objective and mass as secondary objectives makes the problem less susceptible to sensitivity of control parameters often found in penalty-based approaches. This is obvious from results presented so far, which showed relative stability to the effects of the number of constraints, violation and objective values. It is in the localised comparison of prioritised values, as opposed to a global one-off comparison with penalty-based fitness functions, that more accurate representation of genotypic fitness can be made in constrained optimisation problems.

Attention should be drawn to the I4 case, where result obtained showed a mass that is higher than I5, which is believed that it is a sub-optimal solution. The population size of 54 genotypes for blade-stiffened panels is obviously too small, providing insufficient genetic diversity. Results are much more stable for flanged-stiffened panels with a population size of 82. However, the larger size and the need to model additional flanges contributed to an increase in computation time. Citing the case of 3 stiffeners, the average time to convergence for blade-stiffened panel I3 is approximately 114 CPU seconds[1]. The equivalent flanged-stiffened panel J3 required an increase of approximately 6.9 times in computation time. This is an indication of an increase in analytical complexity due the additional flange. The number of generations to convergence for J3 increased approximately 4.7 times from an average of 100 generations for I3.

[1] All runs were performed on an Intel Pentium III 600MHz PC with 128MB RAM.

Load Case	Half-Wavelength (mm)	Critical Load Factor
1	752.0	1.005148091
2	94.0	1.065880668
3	752.0	1.550084673

Table 6-8 Critical buckling load factor and governing half-wavelength for flange-stiffened panel J3 for each of the loading condition.

Allowing stacking sequences to vary in all components of stiffened panel assemblies has shown remarkable improvement in terms of mass. This statement is justified since all dimensions, and loading conditions, especially for blade-stiffened panels, models closely that used by the GARTEUR benchmark problems. The only variation is in the analysis of buckling constraint satisfaction, which, in the case of GARTEUR, the more accurate VICON model was used. Since shear loading was present in all three loading conditions, mass results were generally expected to be higher. In the case of I3, York et al. (1993) reported that using the VIPASA model gave masses that were uneconomical, namely an increase in mass by 17.5%. Discrete thickness results of the GARTEUR benchmark blade-stiffened panels, reported by Kennedy et al., (1999) and obtained by forcing the continuous optimum to undergo a two-stage discrete optimisation via a simple sequential rounding technique and then a combinatorial partial enumeration method, suffered less than a 1% mass penalty.

It is, of course, impossible to make a direct comparison with published results due to the fact that stacking sequences are allowed to vary and integer multiples of standard plies are enforced in the work presented here[1]. Nevertheless, assuming that the discrete thickness optimum using the VIPASA model suffers the same mass penalty as that of the VICON model, the discrete optimum mass of I3 using the former analysis should be approximately $9.3kg$. Results here indicate that savings by allowing stacking sequences to vary can be as much as 14.6%, if the above assumptions are correct and that the best results obtained for the I3 panel above is the global discrete optimum.

The conservatism of the VIPASA model can be clearly seen in a full analysis of the best solution for the J3 panel. Critical load factors for all load cases, given in Table 6-8, indicates that the critical buckling mode is the overall mode, ie., $\lambda = \ell$. Given the fact that the longer wavelength is the governing mode, the presence of shear loads has a significant effect on its

[1] Published results had stacking sequences fixed in cases that were directly comparable to those presented here and were continuous optimisation problems, which required a secondary round of optimisation to reach a global discrete optimum.

stability and producing skewed nodal lines, as plotted in Figure 6-9. In cases when the governing mode is local (e.g., $\lambda = \ell/8$ for load case two), the simply-supported ends are more accurately modelled, thereby increasing the accuracy of the load factor obtained. For example, additional analysis for load case 2 of the J3 panel using the more accurate VICON model gave a critical load factor of 1.067 as opposed to 1.066 for VIPASA, which is a difference of approximately 0.1%. In contrast, the factors for load case 1 of the corresponding panel, where a longer wavelength is the governing mode, are 1.156 and 1.005 for VICON and VIPASA, respectively. This implies that VIPASA, in this case, is 15.0% more conservative. Similarly for load case 3, the critical load factors of 1.856 and 1.551 for VICON and VIPASA analysis, respectively, amount to an estimate that is 19.7% more conservative by VIPASA. Nevertheless, all configurations listed in Table 6-7 have been verified to be stable when analysed with the VICON analysis.

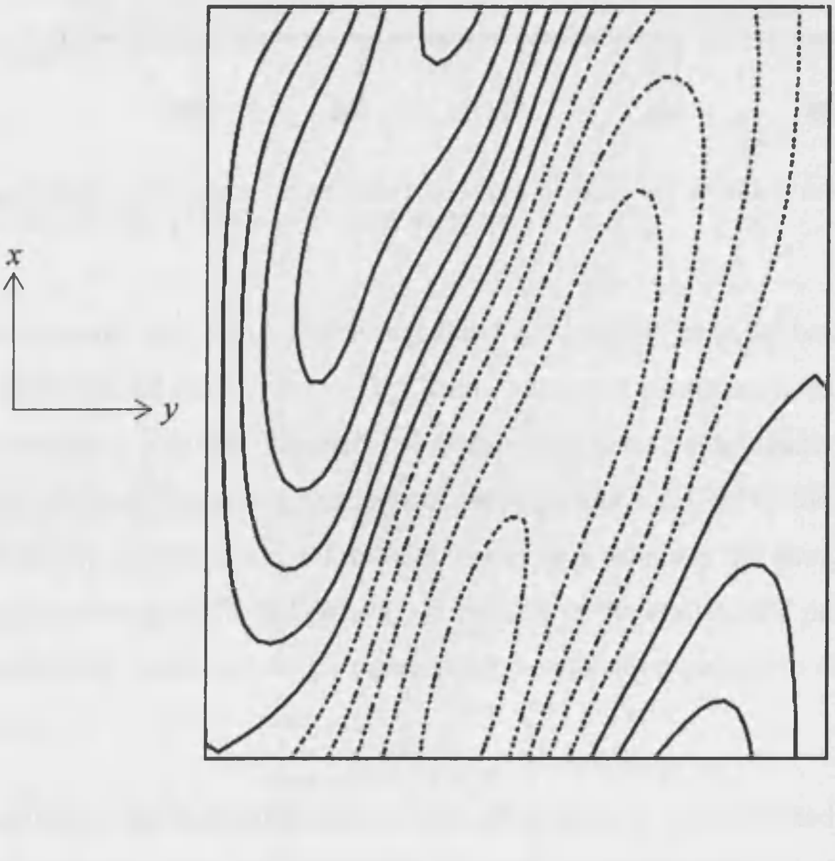

Figure 6-9 Skin contour plot of the governing buckling mode of the best solution for the J3 panel. It clearly shows the skewed nodal lines that violate the simply-supported transverse end conditions assumption. Isometric plots of the critical modes of all load cases for the J3 panel are provided in Appendix IV.

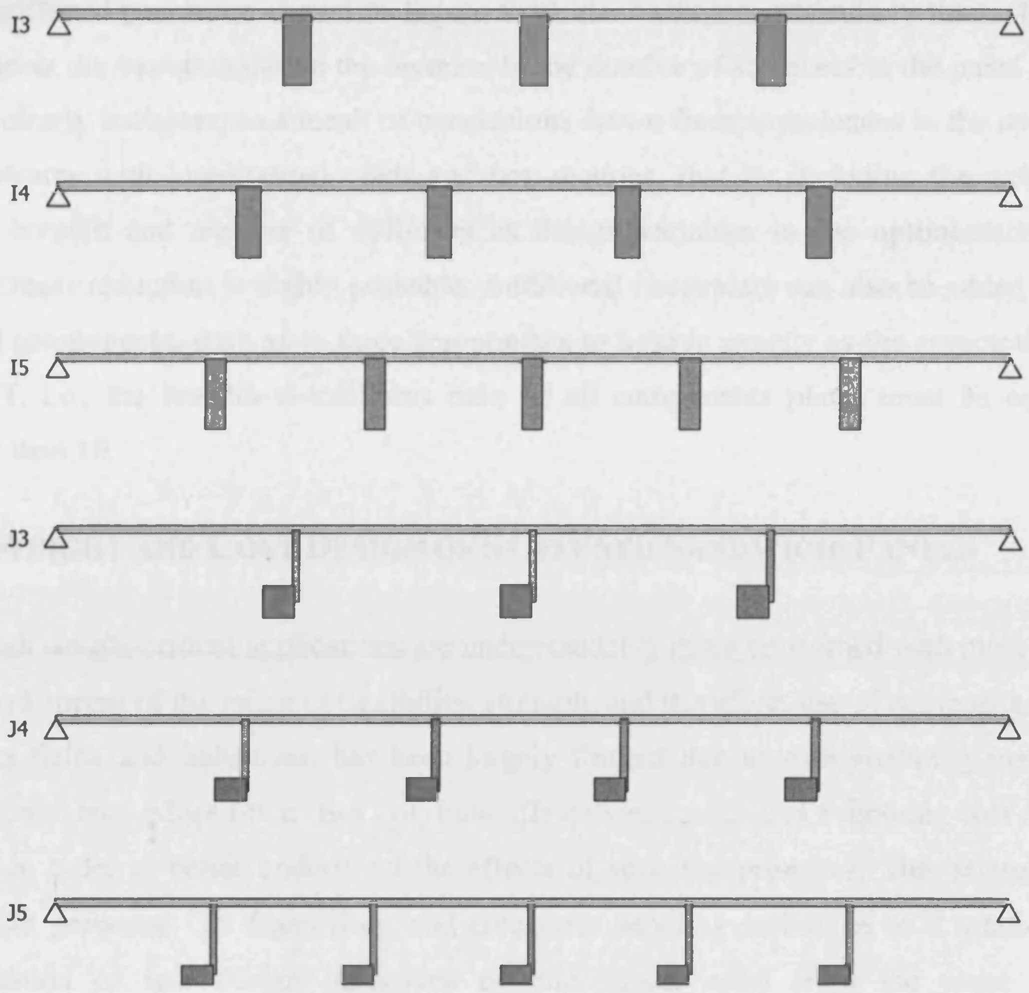

Figure 6-10 Scaled diagrams of the best solution found for each of the six cases of stiffened panels considered, together with longitudinal boundary support conditions.

The above results also show that flanged-stiffened panels give far better mass efficiency that their blade-stiffened counterparts. This behaviour is not unexpected, as the results have shown. Higher efficiency can be obtained by distributing mass away from the neutral axis of the structure, and thus increasing the overall stiffness and rigidity. In the blade-stiffened panels, this distribution is done most effectively in the web, whereas the flange is the more effective component in flange-stiffened panels. As evident in the results, the percentage of mass in the skin is relatively stable across all cases since the skin is expected to carry any transverse and shear loads.

The longitudinal compressive load, on the other hand, is redistributed across all components based on stiffness, thus the behaviour as seen. It is, therefore, not wrong to assume that allowing the web and flange breadth to vary will create more efficient structures in terms of mass subjected to buckling load constraints. However, fixing the breadths to constant values resulted in components, flanges in particular, that violate the assumptions of CPT. Even in blade-stiffened panels, as shown in Figure 6-10, the webs are particularly thick. This figure also shows the variation due to the increase in the number of stiffeners in the panel assembly, which clearly indicates, as a result of conclusions drawn from experiences in the optimisation of laminates with longitudinal voids and box-sections, that by including the web breadth, flange breadth and number of stiffeners as design variables in the optimisation process, further mass reduction is highly probable. Additional constraints can also be added to further restrict components, such as to force components to behave exactly as the assumptions made by CPT, i.e., the breadth-to-thickness ratio of all components plates must be equal to or greater than 10.

6.4 Weight and Cost Design of Stiffened Sandwich-Panels

Although weight-critical applications are understandably more concerned with mass reduction, the broad appeal of the inherent flexibility, strength, and therefore, use of composite laminates in other fields and industries, has been largely limited due to overwhelming material and fabrication costs. More often than not, tradeoffs between mass and economic cost are highly sought in order to better understand the effects of such compromises. This section aims to apply the proposed GA framework and constraint handling technique to a multi-objective optimisation of two design objectives of importance, apart from the usual structural requirements[1], in the application and use of composite laminates – weight and cost. Combined weight and cost optimisation has been already performed using VICONOPT, such as those presented by Edwards et al., (1998), which replaced the minimum mass objective with minimum cost or a combination of both. Another work by Grosset et al., (2001), performed a mass-cost multi-objective optimisation involving two materials on a single laminated plate.

The design optimisation here will take on a more common manufacturing approach used in the industry of composite laminates to increase stiffness, and hence structural load capacity, without excessive increase in mass. This approach, also known as *sandwich construction*, uses generally isotropic and lightweight materials as the core, sandwiched between two layers of composite laminates as skins, hence the term, as shown in Figure 6-11. Engineering theory can easily show that flexural stiffness of any uniform panel is proportional to the cube of its thickness.

[1] Structural requirements are almost always implemented as design constraints rather than design objectives.

The purpose of any core material is, therefore, mainly to increase the laminate flexural stiffness by 'thickening' with a low-density, low-cost structural material. Figure 6-11 also shows how such a component can be likened to a structural I-beam, with the skins acting like the analogical flanges of the I-beam, and the core as a shear web. It is with this principle that selection of core materials should ideally be based on shear strength and stiffness. Nevertheless, the core must equally be capable of accommodating sufficient compression without premature failure to prevent adverse effects, particularly those with thin skins, such as face wrinkling.

Perhaps the most generally used cores are synthetic polymer-based, with examples such as polyurethane, polyvinyl-chloride and polystyrene, ranging from a density of less than $30kgm^{-3}$ to more than $300kgm^{-3}$, although other materials such as wood (balsa and cedar) and even metals[1] have also been used. An important issue, regardless of core material, is the bonding between the composite laminate skins and the core[2]. Therefore, apart from plain block cores, honeycomb-structured cores can also be used, which allows small bonding fillets around cell walls.

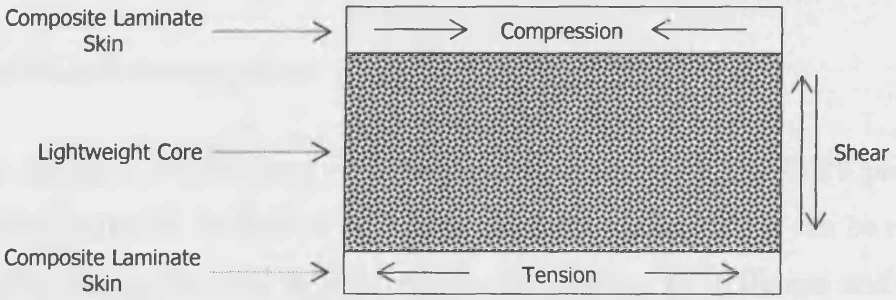

Figure 6-11 Typical cross-section of sandwich construction, and forces acting on an element of a panel in flexure.

[1] Metals, like aluminium, have also been used, although arguably, they function more than just as a lightweight core.

[2] Although low-density core materials arguably provide excellent increase in stiffness with little penalty to mass, the porosity also increases absorption of matrix resins.

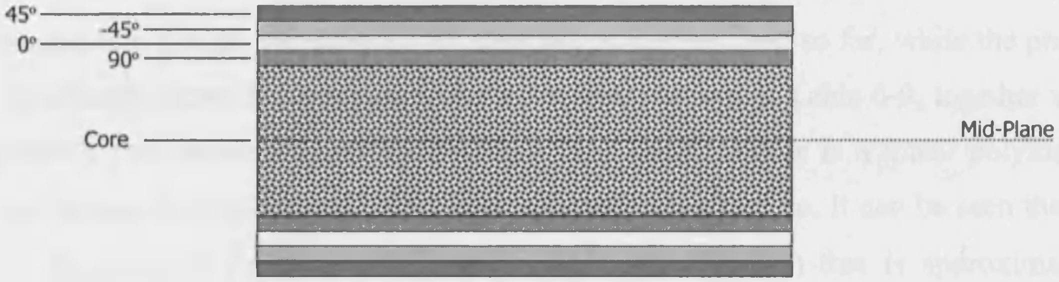

Figure 6-12 Stacking sequence of all component plates, i.e., the skin, web and flange of the stiffened panel.

Material	E_1	E_2	G_{12}	ν_{12}	ρ	Price
CFRP	125.00	8.80	5.30	0.35	1620	10
SAN Core	0.161	0.161	0.057	0.40	140	1

Table 6-9 Material properties considered for the plies and core, as well as the price factor. The price factor is considered as price per unit volume and should not be considered as absolute. It is used here primarily for design studies.

6.4.1 Problem Definition

The design problem concentrates on a single case of the GARTEUR P8 panel, namely the blade-stiffened panel I3, as illustrated in Figure 6-7 above, because it can be readily extended to other cases where the only difference is in the number of stiffeners and the addition of flanges in the case of flange-stiffened panels (J3 – J5). The dimensions and loading conditions are, therefore, the same. Contrary to the design problems studied so far, a fixed stacking sequence will be used here. For all skin and web components, the stacking sequence remains identical, as shown in Figure 6-12, illustrating also the design variables associated with the thickness of different fibre orientations and the core material.

As before, the thicknesses are expected to be discrete, where $t(\theta)_R = d(\theta)_R t_L$, hence the integer number of plies $d(\theta)_R$ are actually the design variables, and the subscript R is represented by s and w for skin and web respectively. The angles θ considered, as seen, are 0°, $\pm 45^\circ$, and 90°, with and additional C to represent the core. The discrete thickness of the standard lamina is here taken to be $0.125mm$, as in the GARTEUR benchmark problem presented earlier. The core thickness is expected to be accurate to $1.0mm$, which is also discrete although manufacturers do specify other standard thicknesses.

All laminates will use the standard CFRP material properties used so far, while the properties for the structural sandwich foam core material are summarised in Table 6-9, together with the corresponding price factors rather than actual price. The core here is a linear polymer foam based on Styrene Acrylo-Nitrile (SAN), with a closed-cell structure. It can be seen that in the principal direction, CFRP has a stiffness-to-weight ratio (E_1/ρ) that is approximately 61 times more than that of the core material. The price-to-weight factor will produce different solutions at different factors.

6.4.2 Weight-Cost Tradeoffs

Coupling both mass and economical cost objectives require the use of multi-objective optimisation concepts. As covered in Chapter 3, the interaction between objective functions creates tradeoffs, otherwise known as the Pareto set. In this case, the Pareto surface is constructed from the interaction of the objective mass function $M(\mathbf{x})$ and objective material cost function $E(\mathbf{x})$. The compromise between these two functions is controlled by a parameter a, or the trade-off factor, such that

$$F(\mathbf{x}) = aM(\mathbf{x}) + (1-a)E(\mathbf{x}), \qquad 0 \le a \le 1 \qquad \text{6-4}$$

The above method is based on the weighted sum of objectives. For any set of positive weights, the global optimum of $F(\mathbf{x})$ is always a non-Pareto dominated solution of the multi-objective problem. However, the reverse is not always true, where a non-dominated solution is not necessarily the global optimum, as illustrated in Figure 6-13. It shows concave and convex regions on the Pareto front, where non-dominated solutions on the concave front cannot be obtained since their sum-weighted objective value is sub-optimal. At best, therefore, the Pareto set obtained here is convex and does not guarantee global optima. Additionally, this approach is also particularly sensitive to the weights, i.e., a and $1-a$ in this case. Initial studies showed that the given bounds on the trade-off parameter gave acceptable convergence rates at all tradeoffs with the GA optimiser framework.

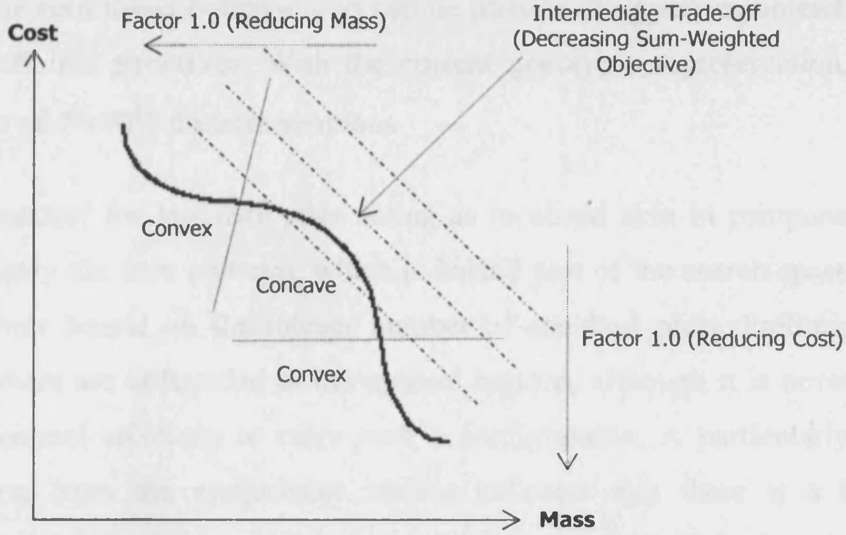
Cost
Factor 1.0 (Reducing Mass)
Intermediate Trade-Off (Decreasing Sum-Weighted Objective)
Convex
Concave
Convex
Factor 1.0 (Reducing Cost)
Mass

Figure 6-13 Graphical representation of a Pareto front with convex and concave regions, as well as the lines of equal sum-weighted objective value at some arbitrary trade-off factor.

There exist GA that construct the Pareto set in a single optimisation, such as the one proposed by Deb (2001). The simpler approach of steadily incrementing a from 0 to 1 will be adopted here, thereby requiring several independent GA runs. Obviously, the more intervals taken between 0 and 1 for a, the better the accuracy of the Pareto front, but for the design studies conducted here, the interval is taken to be 0.1. Additional intervals of 0.02 will be taken between 0.9 and 1.0 to provide better estimate of the transition between optimum mass and sub-optimal mass as a result of a compromise with material cost.

6.4.3 Optimiser Parameters and Configuration

Since the stacking sequences are fixed in all panel components, the design variables concentrate solely on the integer number of standard plies $d(\theta)_R$. Considering also the fact that the component plates are all balanced and symmetric, only half the set of variables of each panel are included in the genotype, i.e., $d(0)_s$, $d(\pm45)_s$, $d(90)_s$, $d(C)_s$, $d(0)_w$, $d(\pm45)_w$, $d(90)_w$, and $d(C)_w$, where each are individual chromosomes, and $d(C)_w$ represents the full thickness in millimetres of the foam core at each component panels since symmetry about the mid-plane of the core is implied. For each of these chromosomes, the design bounds[1] are $0 \le d(\theta)_R \le 127$, which requires precisely a 7-bit string, resulting in a

[1] The upper bound is adopted for ease of applying a binary-type encoding, where in this case, it clearly fits a 7-bit string.

total of 56 bits[1]. The zero lower bound allows certain plies to disappear completely, hopefully producing more efficient structures. With the current genotypic representation, the search space has in excess of 7×10^{16} discrete solutions.

It is actually impractical for laminate plies acting as localised skin in component plates to disappear leaving only the core material, which is indeed part of the search space here, since there is a zero lower bound on the integer number of standard plies. Preliminary studies showed that such fears are unfounded in the optimal regions, although it is nevertheless still possible for sub-optimal solutions to carry such a configuration. A particularly interesting behaviour emerging from the preliminary studies indicates that there is a tendency to distribute all the mass only in the stiffened panel assembly skin. By using the genotypic representation as outlined above, it is possible to reach such a solution, albeit with difficulty. With this in mind, the main studies concatenated an additional one-bit chromosome that allowed the genotype to 'switch' on or off the web stiffeners in the phenotypic representation. In other words, whenever this bit evaluates to Boolean true, stiffeners are added to the structure, otherwise, it is a simple plate spanning the breadth without stiffeners, as shown in Figure 6-14.

In the conducted studies, the effect of this chromosome has direct implication on the genotype too, which, if present, will force an additional mutation operator to mutate all genes corresponding to web design variables to the lower bound, as shown in Figure 6-15. In other words, the switch chromosome takes precedence over other chromosomes. Forcing the 'switch' mutation also ensures that future interactions between genotypes are done based on correct genetic information.

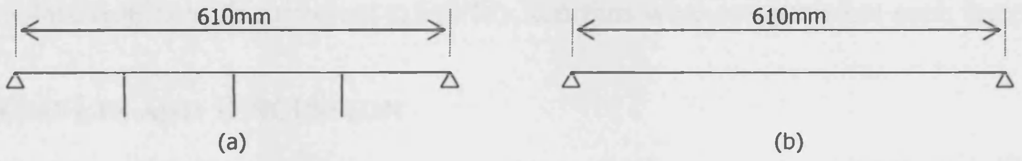

Figure 6-14 Two phenotypic representation depending on the additional on/off switch chromosome, where the bit evaluates to Boolean (a) true, or (b) false.

[1] This also implies that core thickness can range from $0mm$ to $127mm$.

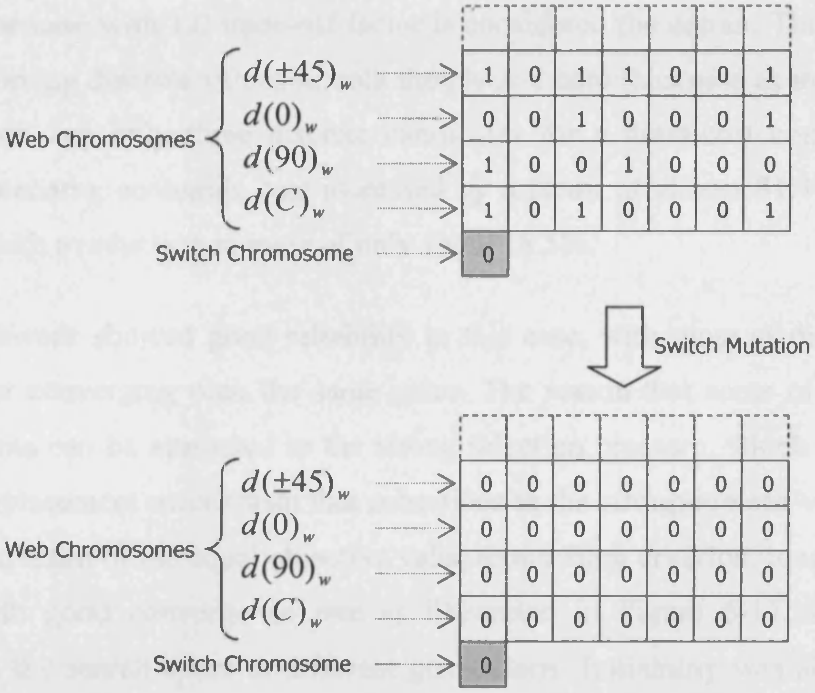

Figure 6-15 Effects of the switch chromosome evaluating to false on web chromosomes. Skin chromosomes are not shown in full but indicated by the boxes in dashed lines.

The population size is taken arbitrarily to be 100 and is sampled at this constant value across all generations. The recombination process involves the binary tournament selection with a uniform crossover at a probability of crossover at 0.9 and probability of gene swap at 0.3. On the other hand, each population undergoes uniform mutation at a rate of 4 bits in 57. The search is terminated either when the objective values of all genotypes are identical or the 2000th generation has been reached. The GA parameters were determined from results of trial runs.

The trade-off factor starts at 1.0 (minimum mass) and terminates at 0.0 (minimum cost), decreasing by 0.1 at each round, with additional intervals of 0.02 between 0.9 and 1.0. In order to obtain results with sufficient reliability, ten runs were conducted at each factor.

6.4.4 Results and Discussion

Results for the best solution found for the 15 tradeoffs are as given in Table 6-10. At all factors, none of the best solutions found had web stiffeners, and their corresponding design variables are therefore left out from the results.

It interestingly shows that compromise between the mass and economic cost flattens out after a factor at 0.8 and lower, as no improvement can be gained either way, as shown in Figure

6-16, where the case with 1.0 trade-off factor is considered the datum. This is believed to be the effect of forcing discrete values in both the ply and core thickness as integer multiples. In effect, therefore, are only three discrete candidates for a mass-cost compromise. For the lowest mass scenario, economic cost increased by a factor of almost 81.4% over the lowest cost solution with a reduction in mass of only about 18.5%.

The GA framework showed good reliability in this case, with most of the ten runs at each trade-off factor converging onto the same value. The reason that some of the runs failed to reach the optima can be attributed to the strong selection pressure, which is imposed by the steady-state replacement mechanism that subscribes to the strong-replace-weak rule. All runs terminated as a result of the equal objective value termination criterion, long before the 2000th generation with good convergence rate as illustrated in Figure 6-17, showing genotype distribution in the search space at different generations. Reliability was also improved over preliminary studies as a result of introducing the web on/off switch chromosome, which, in a single bit mutation, allowed genotypes to add or remove phenotypic features, i.e., the web stiffener. Otherwise, substantially longer evolution time will be required before the effect of not having web stiffeners is explicitly modelled by genotypic representation[1].

Factor	Mass (kg)	Cost	$d(\pm45)_S$	$d(0)_S$	$d(90)_S$	$t(C)_S$
1.00	3.23398	13.26618	3	0	0	33
0.98						
0.96						
0.94	3.37618	10.06432	2	0	0	41
0.92						
0.90						
0.80	3.96793	7.31200	1	0	0	56
0.70						
0.60						
0.50						
0.40						
0.30						
0.20						
0.10						
0.00						

Table 6-10 Best solutions found at various trade-off factors. The core thickness is in millimetres.

[1] It should be noted that this does not imply that solutions without stiffeners were almost impossible without the switch chromosome. In fact, preliminary studies without it were also successful in finding the same solutions.

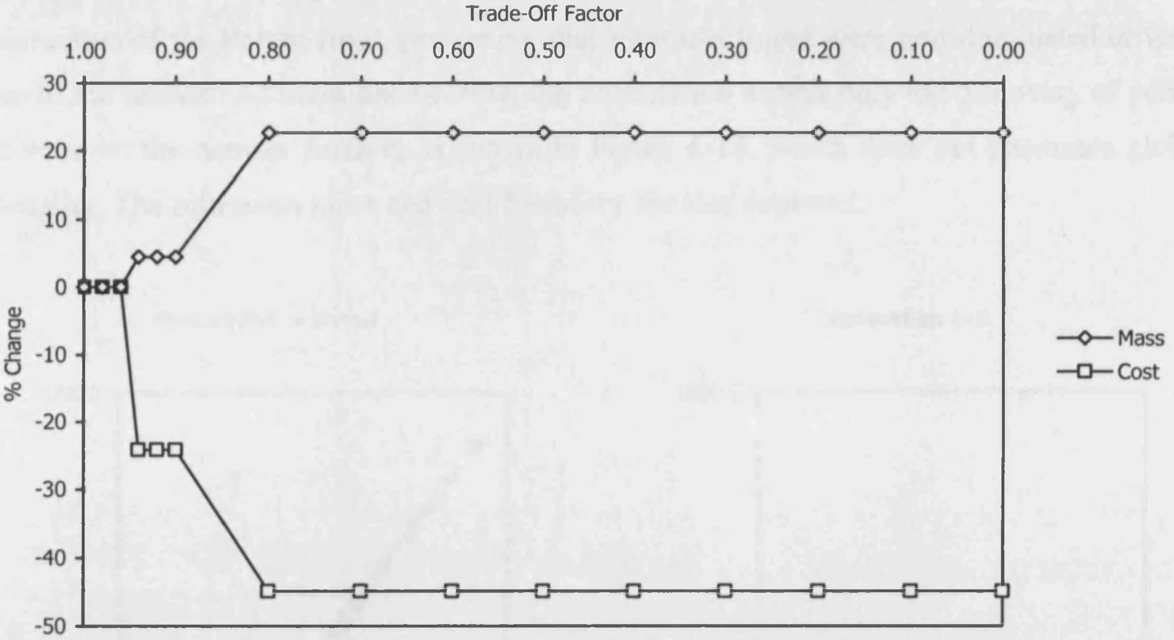

Figure 6-16 Percentage change in mass and cost relative to the minimum mass case (1.0 factor) with respect to trade-off factor.

Despite substantial mass savings over the GARTEUR I3 benchmark problem, presented in the previous section, of up to almost 59.3% with the lowest mass of about $3.23kg$, the skin panels were noticeably thick but still satisfy the assumptions of classical thin plate theory. Breadth-to-thickness ratios in the three discrete optima found are, from highest trade-off factor to lowest respectively, 17.7, 14.5 and 10.8. Although satisfying the traditional CPT assumption of breadth-to-thickness ratio, it should also be noted that the thick flexible core is highly deformable, and so may violate assumption of zero through-the-thickness shear deformation.

In all cases, thickness contribution came mainly from the foam core, which is as expected. The additional stiffness provided by lightweight core materials, by moving important stiffness contribution from composites further from the neutral axis, resulted in very thin skins enclosing the foam core. This, naturally, gives rise to concern about stress concentrations leading to micromechanical failures in the composite skins, such as face wrinkling. The absence of 0° or 90° plies is hardly surprising, as the thick foam core, with its isotropic properties, is advantageous also in in-plane longitudinal and transverse effects. The presence of minimal ± 45° plies is, however, indicative of the more efficient handling of shear effects. Nevertheless, additional good design practices such as the 10% design rule (Niu, 1993) should in essence be included to provide better practicality of such laminates.

The variation in weighting factors of the multi-objective mass and cost problem allows the construction of the Pareto front, presuming that solutions found were non-dominated or were close to the surface. As mentioned earlier, the formulation allows only the discovery of points that were on the convex surface, as shown in Figure 6-18, which does not guarantee global optimality. The minimum mass and cost boundary are also depicted.

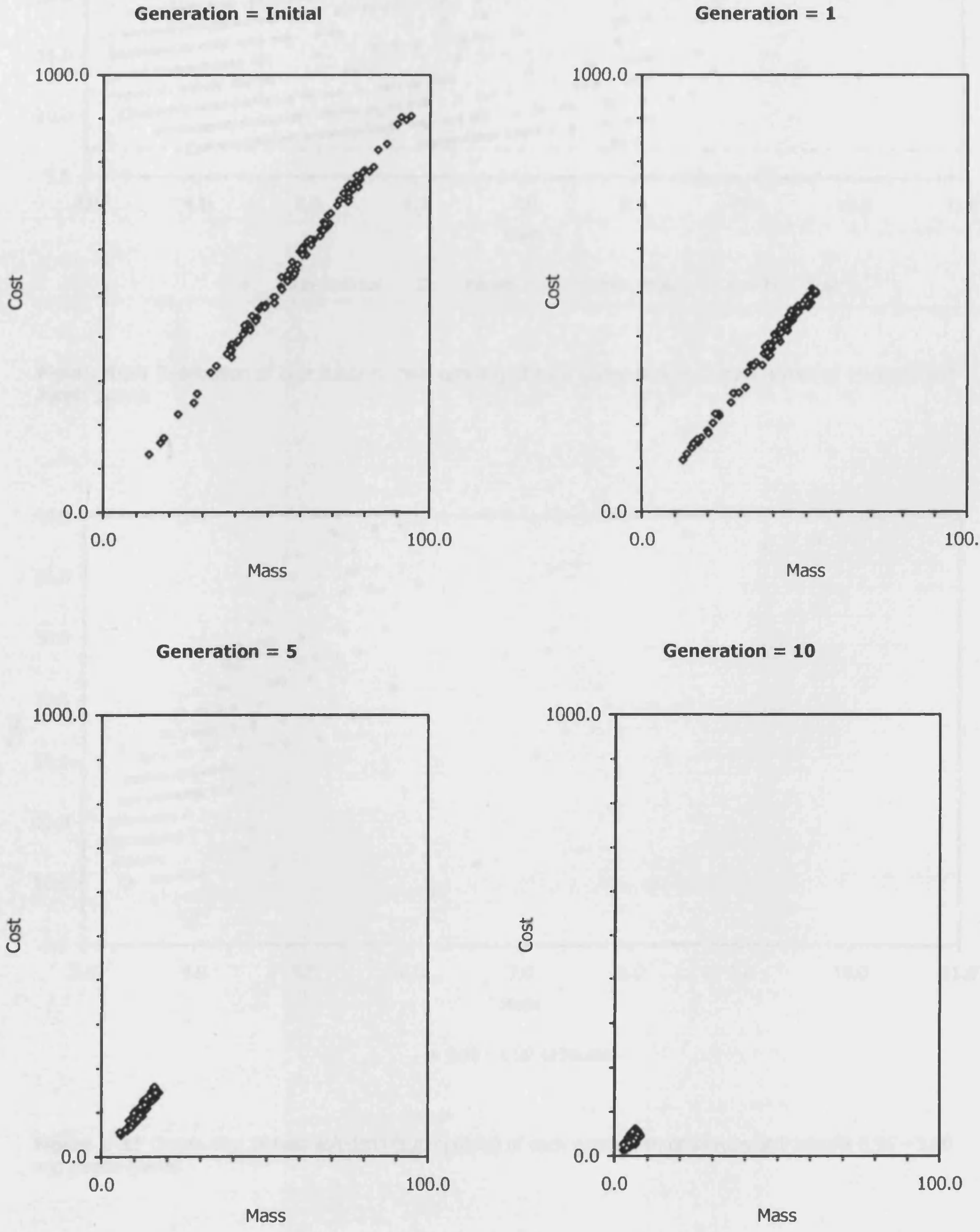

Figure 6-17 Distribution of genotypes in terms of mass and cost at selection generations.

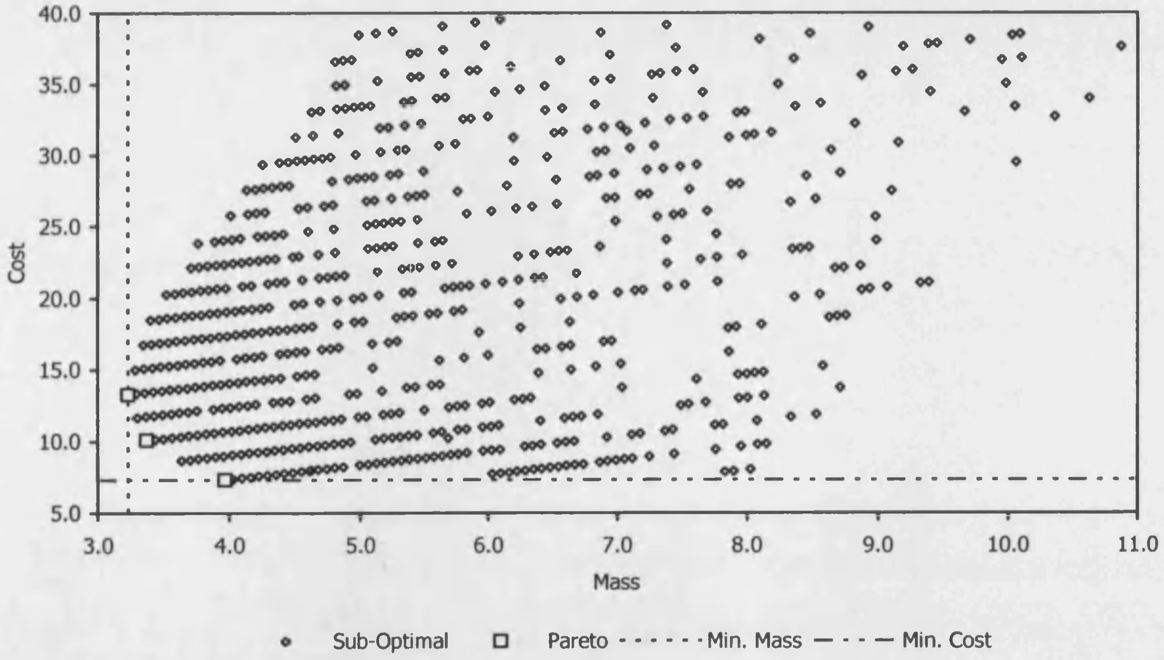

Figure 6-18 Distribution of best solutions (sub-optimal) of each generation of all runs across all tradeoffs and Pareto points.

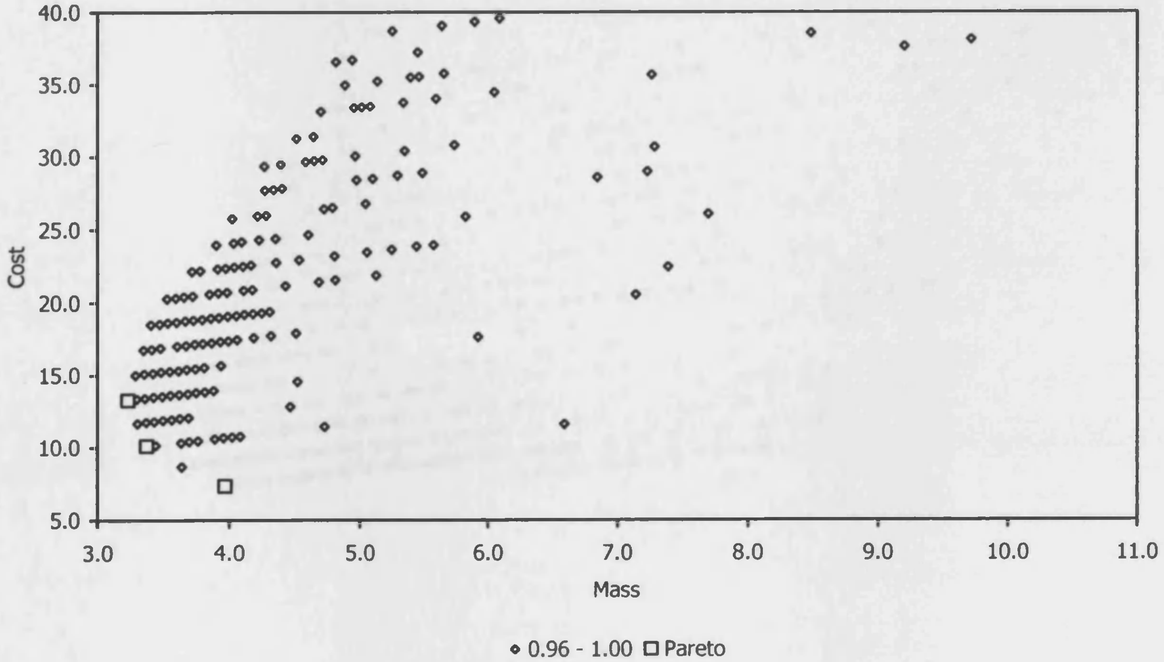

Figure 6-19 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.96 – 1.00 and Pareto points.

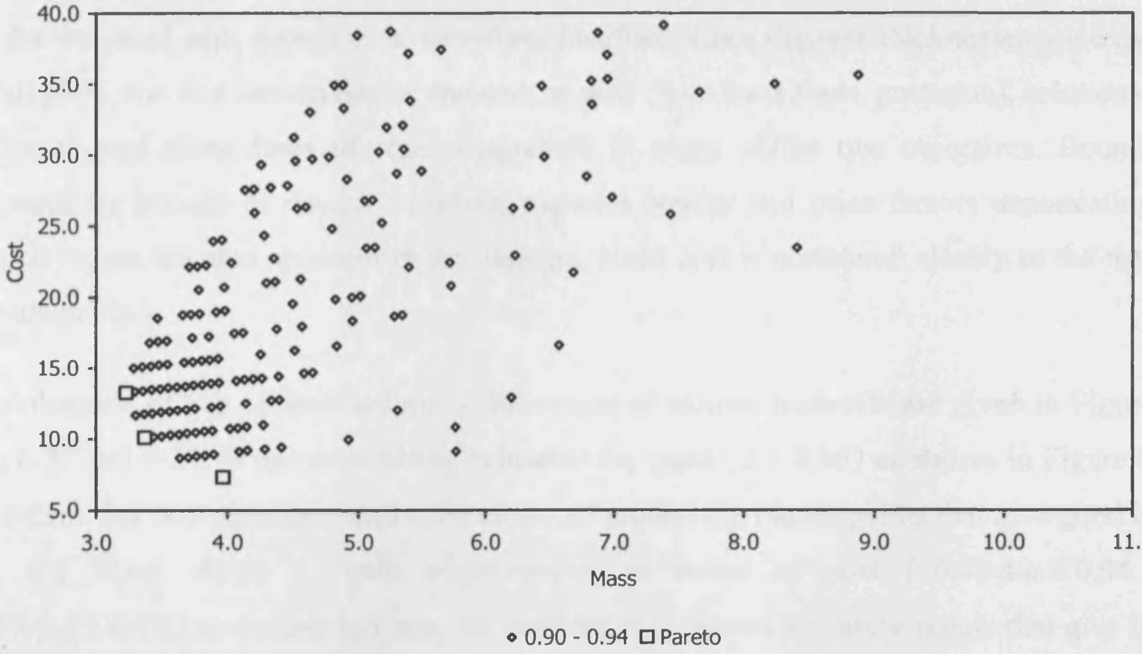

Figure 6-20 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.90 – 0.94 and Pareto points.

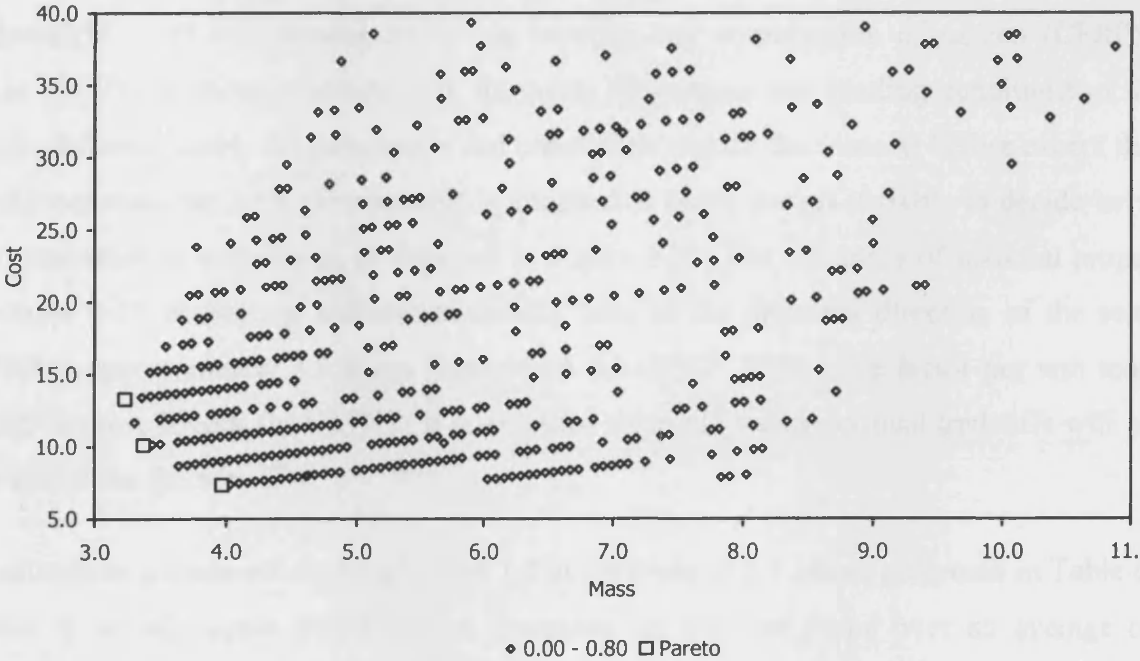

Figure 6-21 Distribution of best solutions (sub-optimal) of each generation of all runs at tradeoffs 0.00 – 0.80 and Pareto points.

The diagram, with available data, clearly shows that the surface is indeed convex and the use of the weighted sum approach is, therefore, justified. Since discrete thicknesses are expected of all plies, and to a certain degree the core as well (note the $1.0mm$ precision), solutions tend to be aligned along lines of equal magnitude in terms of the two objectives. Boundaries imposed by bounds of design variables, material density and price factors demarcating the search region are also apparent in the diagram, since cost is correlated closely to the mass of the assembly.

Distributions of sub-optimal solutions discovered at various tradeoffs are given in Figures 6-19, 6-20 and 6-21. When minimising primarily for mass ($a \geq 0.96$) as shown in Figure 6-19, it is clear that best solutions tend to be clustered around the Pareto points that give good mass. As the factor drops to gain improvement in terms of cost ($0.90 \leq a \leq 0.94$ and $0.00 \leq a \leq 0.80$) as oppose to mass, the preference is moved to Pareto points that give lower cost.

6.4.5 Variant Materials

Additional design runs were performed to obtain less extreme solutions (bulking with lightweight core) by allowing switching between any combination of carbon (CFRP) and glass (GFRP) reinforced epoxy, with the same dimensions and loading conditions of the I3 blade-stiffened panel. All parameters and constraints remain the same as before except that an additional material 1-bit chromosome is attached at every design variable to decide between both materials at each layer, as depicted in Figure 6-22. The summary of material properties in Table 6-11 shows the stiffness-to-density ratio in the principal direction of the selected CFRP is approximately 3.5 times higher than for GFRP. With price factor per unit mass of CFRP 8 times higher than GFRP, it is expected that only weight-critical tradeoffs will prefer the use of the former.

Results from a trade-off factor of 0.0 to 1.0 at intervals of 0.1 are as presented in Table 6-12. It has to be said again that solutions presented are the best found over an average of 10 independent runs at each interval without any guarantee of Pareto or objective optimality. As in the case with foam core, tradeoffs disappear after a certain threshold, where in this case, it is at a factor of 0.7 and lower. However, the web has a more important role to play here since an increase in stiffness cannot be effectively achieved by an increase in the skin alone.

The best solution, in terms of mass, expectedly utilised fully the stiffness-to-weight properties of CFRP, giving a mass of $9.1kg$. Once a trade-off with cost is required, the cheaper GFRP starts to bulk up the core in the skin and web at intermediate factors (0.9 and 0.8), effectively creating a sandwich type construction as seen when using strictly structural core materials. At the other extreme of lowest cost, where mass is of no concern, then obviously GFRP becomes preferable. Care should, of course, be taken when using mixed materials, noticeably at non-extrema tradeoffs. The two different fibres, in an otherwise homogeneous resin matrix, create hybrid laminates that may well experience internal thermal expansion effects.

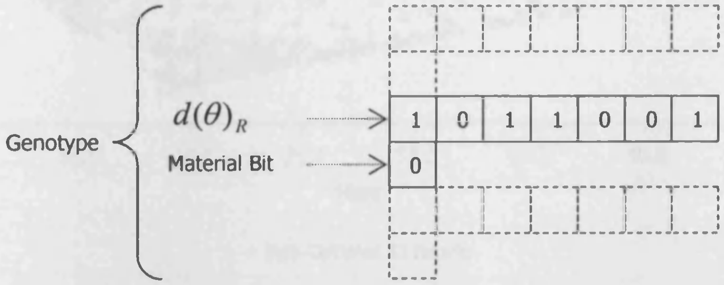

Figure 6-22 Concatenation of the material 1-bit chromosome to the genotype.

Material	E_1	E_2	G_{12}	ν_{12}	ρ	Price
CFRP	125.00	8.80	5.30	0.35	1620	8
GFRP	43.00	8.90	4.50	0.27	2000	1

Table 6-11 Material properties between a carbon and glass fibre reinforced epoxy.

Factor	Mass (kg)	Cost		Stacking Sequences
1.0	9.10787	72.86293	S	$[\underline{\mathbf{45}}_8/-\underline{\mathbf{45}}_8/\underline{\mathbf{0}}_{12}]_S$
			W	$[\underline{\mathbf{45}}_{10}/-\underline{\mathbf{45}}_{10}/\underline{\mathbf{0}}_{70}]_S$
0.9	10.81747	51.03081	S	$[\underline{\mathbf{45}}_6/-\underline{\mathbf{45}}_6/0_{20}/90_2]_S$
			W	$[\underline{\mathbf{45}}_5/-\underline{\mathbf{45}}_5/\underline{\mathbf{0}}_{71}/90_{0.5}]_S$
0.8	12.17615	43.54202	S	$[\underline{\mathbf{45}}_4/-\underline{\mathbf{45}}_4/0_{29}/90_{3.5}]_S$
			W	$[\underline{\mathbf{45}}_7/-\underline{\mathbf{45}}_7/\underline{\mathbf{0}}_{55}/90_{4.5}]_S$
0.7-0.0	18.90058	18.90058	S	$[45_{10}/-45_{10}/0_{29}]_S$
			W	$[45_{10}/-45_{10}/0_{123}]_S$

Table 6-12 Results for mass-cost tradeoff at various levels for a CFRP/GFRP optimisation. Plies in CFRP are in **bold**.

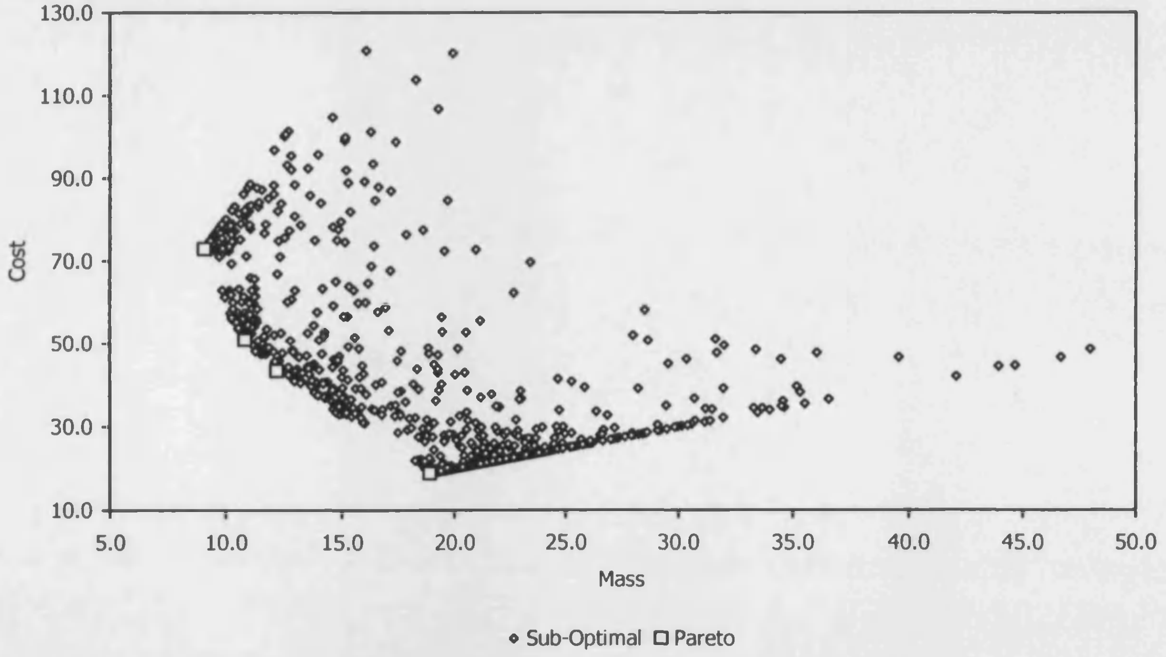

Figure 6-23 Spread of best individuals (sub-optimal) at each generation at all tradeoffs of all runs conducted for the CFRP/GFRP variant case and Pareto points.

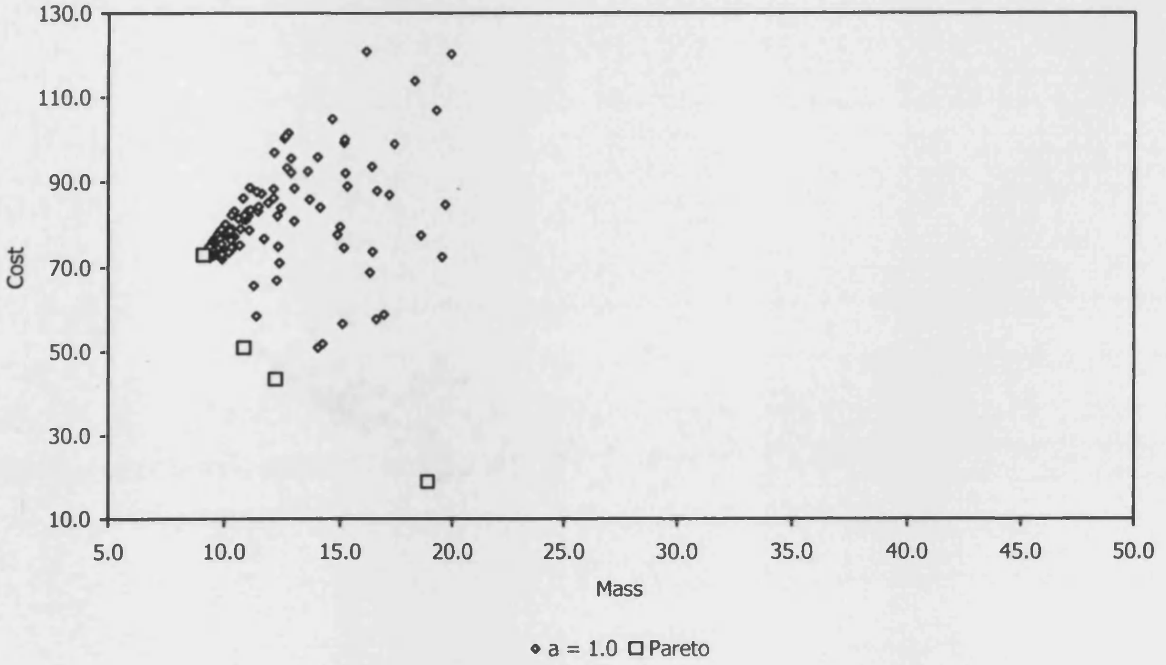

Figure 6-24 Spread of best individuals (sub-optimal) at each generation at factor 1.0 of all runs conducted for the CFRP/GFRP variant case and Pareto points.

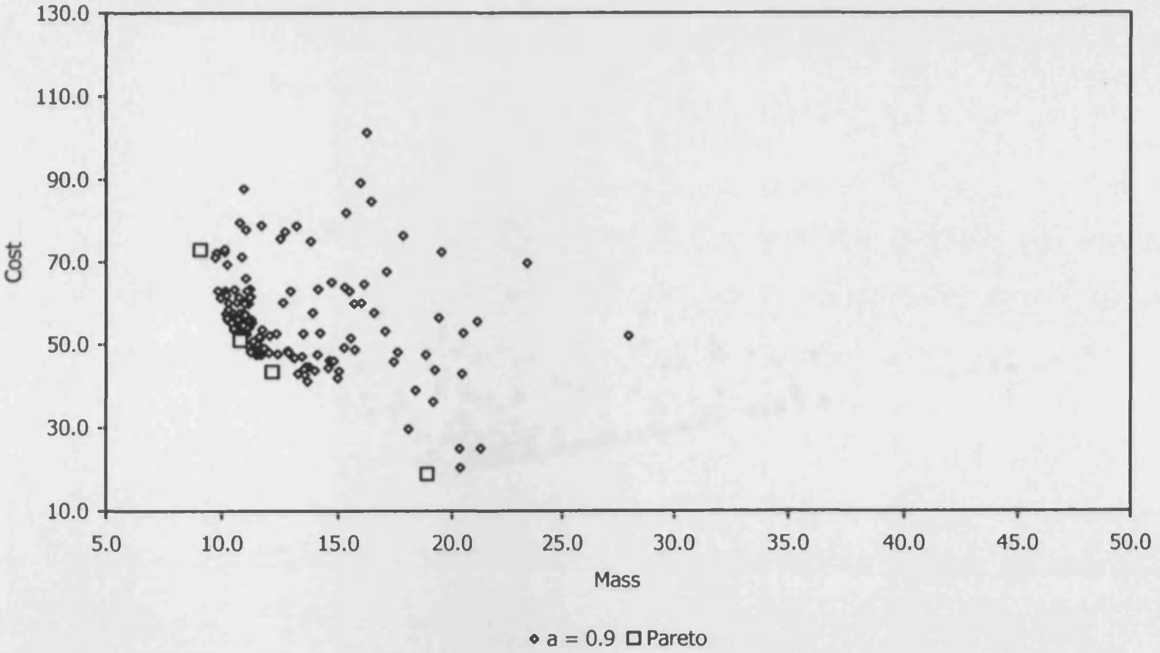

Figure 6-25 Spread of best individuals (sub-optimal) at each generation at factor 0.9 of all runs conducted for the CFRP/GFRP variant case and Pareto points.

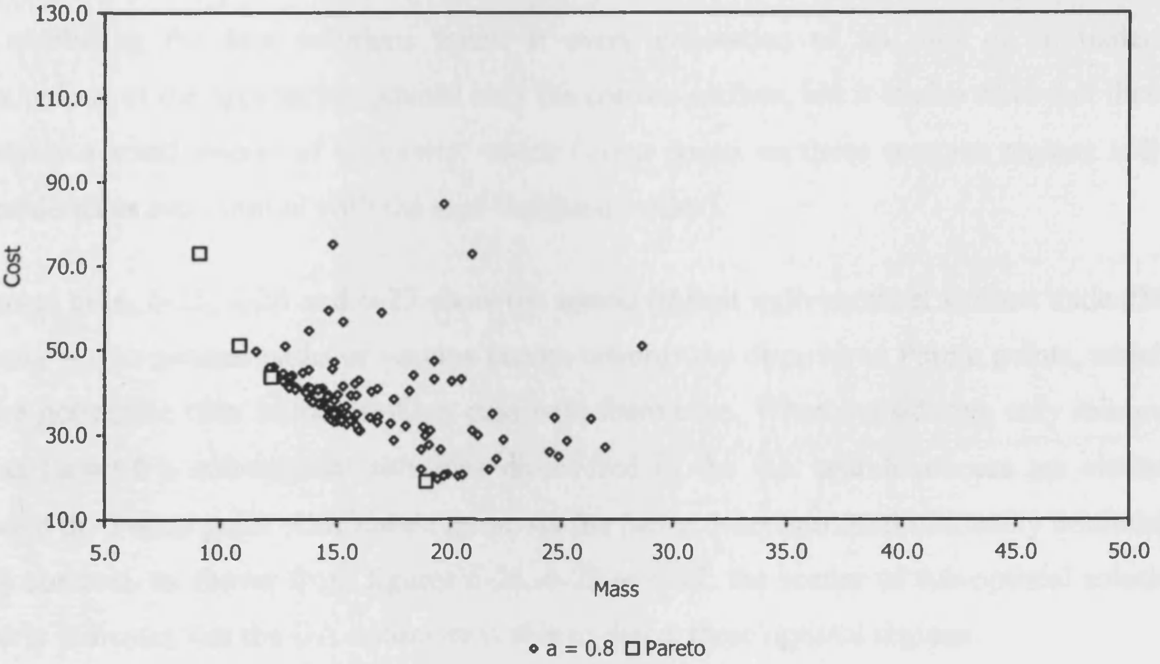

Figure 6-26 Spread of best individuals (sub-optimal) at each generation at factor 0.8 of all runs conducted for the CFRP/GFRP variant case and Pareto points.

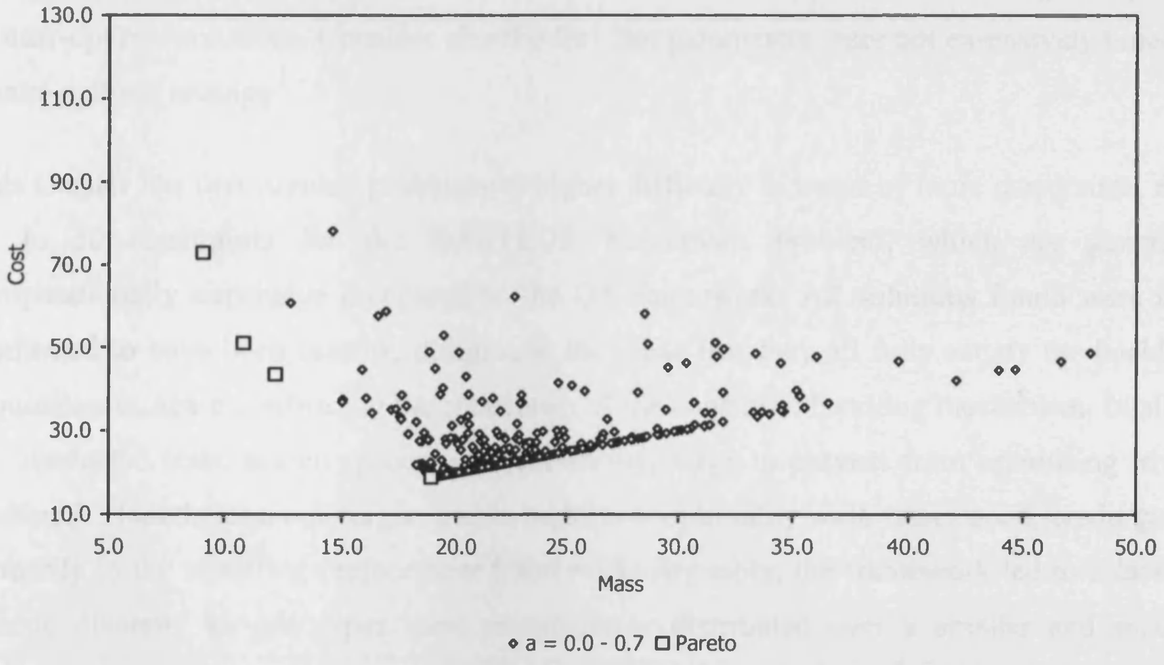

Figure 6-27 Spread of best individuals (sub-optimal) at each generation at factors 0.0 to 0.7 of all runs conducted for the CFRP/GFRP variant case and Pareto points.

Figure 6-23 above shows the Pareto surface obtained from the weighted sum approach as well as combining the best solutions found at every generation of all runs of all tradeoffs. Limitations of the approach produced only the convex surface, but it is also clear that there is possibly a small degree of concavity, which Pareto points on these concave regions will be considered as sub-optimal with the sum-weighted method.

Figures 6-24, 6-25, 6-26 and 6-27 show the spread of best individuals at various tradeoffs. It clearly shows general paths of various factors towards the discovered Pareto points, which is more noticeable than in the previous case with foam core. When considering only minimum mass ($a = 1.0$), sub-optimal solutions discovered in the GA search process are clustered around the Pareto point with lowest mass. As the factor decreases until ultimately minimising only for cost, as shown from figures 6-25, 6-26 to 6-27, the scatter of sub-optimal solutions clearly indicates that the GA optimiser is able to detect these optimal regions.

6.5 Conclusion and Closing Remarks

The lack of a fitness function and problems associated with the use of a non-stochastic sampling mechanism, the binary tournament selection, had no great effect in the various design cases considered. Even without the use of complicated genetic operators and hybrid genetic representation, the deployed GA was able to discover, to a reasonable degree, optimal or near-optimal solutions. Consider also the fact that parameters were not extensively tuned to ensure optimal settings.

This chapter has investigated problems of higher difficulty in terms of more constraints, e.g., up to 30 constraints for the GARTEUR benchmark problem, which are generally computationally expensive compared to the GA framework. All solutions found were also confirmed to have been feasible designs, in the sense that they all fully satisfy the buckling requirements, again confirming the robustness of the constraint handling mechanism. In all of the conducted tests, search spaces were excessively large to prevent from optimising trivial problems. Nonetheless, convergence onto regions of optimality were rather good, credit going primarily to the sampling-replacement framework. Arguably, the framework led to a lack of genetic diversity as genotypes were progressively distributed over a smaller and smaller region of the search space. From a positive point of view, this, coupled with a small mutation rate and small gene swap probability of the uniform mutation and uniform crossover respectively, created a more steady and progressive search across the landscape rather than arbitrary random jumps. The main disadvantage of this approach is the more apparent effects of genetic drift leading possibly to premature convergence (equal objective termination criterion) and falling into local optima.

Understandably, additional work needs to be done if the GA framework is to be successfully applied into design programs such as VICONOPT. Time efficiency and reliability of the GA leaves a lot to be desired. These, and other issues, will be covered in greater detail in the next and final chapter as potential for future improvements and developments.

7 Conclusions and Future Work

With results obtained using Genetic Algorithms (GA) on a wide range of composite laminates and prismatic plate assemblies design problems, this final chapter ascertains directions, and more importantly, practical guidelines on the future establishment of more robust GA tailored specifically to the optimisation of high-dimensional engineering problems.

7.1 Genetic Algorithms

GA, and indeed other evolutionary algorithms like evolutionary computation (EC) and genetic programming (GP), have broadened the scope of optimisation in engineering practice. Without the involvement of over-complicated and problem-specific operators, they are mainly simple to deploy heuristic methods. Self-discovered domain features, without the aid of additional data or any gradient-based knowledge, are used to produce and refine search trajectories internally leading to regions of optimality.

7.1.1 Low Cardinality Genetic Alphabets

One of the strengths of GA lies in its ability to incorporate any combination of coding alphabets into its genetic representation. In general, problems that involve mixed decision and design variables can be handled effectively, with a little bit of ingenuity, without having to resort to high-order alphabets such as integer and floating-point values. This statement is made in accordance with the Building Block Hypothesis (see Chapter 2 for definition, and Chapter 1 for issues associated with it) and the belief that complex structures (and behaviours) can be effectively adapted from seemingly unconnected small building blocks and simplistic interaction rules.

The question will inevitably be asked as to the applicability of the lowest common denominator that is the binary representation, to design and optimisation problems. Although by no means an extensive study of this issue, the various studies conducted have utilised none other than the binary representation from a constrained numerical problem to a multi-objective engineering optimisation of prismatic plate structures involving the mass-cost objective. It has to be admitted that the design variables considered in all cases involve primarily numerical variables. Nonetheless, decision variables have also been paired successfully with numerical design variables. The switch and material chromosomes (see Chapter 6) are prime examples here. This argument does not necessarily rest entirely on using only the binary representation, but urges, instead, the use of the simplest genetic representation possible, be it a ternary or any higher-order scheme, since a universal genetic alphabet is probably unlikely. For example, continuous numerical optimisation might probably benefit more from a floating-point representation rather than binary.

7.1.2 SEARCH CONSTRAINTS

Most, if not all, design and optimisation problems involve a certain number of constraints, be it bounds on design variables or system constraints. The former are almost certainly handled intrinsically by genetic representations and/or operators, but the latter require more careful considerations.

Traditionally, approaches to constraint handling in evolutionary-based algorithms are mainly restricted to test-and-discard, fitness penalties, and genetic repairs. While all such methods have strengths and weaknesses (see Chapter 3 for discussion), a more robust and problem-neutral approach is necessary to give GA a wider appeal and applicability. One of the main aims of this new approach is to reduce the number of tuning parameters, and hence, sensitivity, apparent in penalty-based methods, remove the specificity of repair algorithms, and increase exploitation of seemingly unworthy knowledge.

In Chapter 3, a new general-purpose constraint handling methodology was developed and presented, which borrows principles from multi-objective or multi-criterion optimisation. Constraints are handled as primary objectives to be optimised in an unconstrained multi-objective optimisation framework, whether in terms of minimisation of violation or maximisation of feasibility. Consistency is also maintained in the technique by using multi-objective methods for handling fitness assignment.

Another attractive feature of GA is the ability to address classes of problems that produce non-tangible fitness values. Despite the misnomer associated with measurement of fitness, GA only really requires information pertaining to relative preference of one genotype over another in order to generate internal ranking of performance by means of schema representation. On the simplest level, this implies a Boolean response of 'better than' or 'worse than'. Such is the principle modelled by the constraint handling methodology developed here, generating a relative comparison between two candidate solutions based first on feasibility before turning to objective optimality. With such a simple response, selection and replacement mechanisms are then able to rank genotypes accordingly.

The process of Pareto-ranking is potentially an expensive exercise as the number of constraints increases. Therefore, the new constraint handling method here improves efficiency using the simple process of normalisation. With all violations normalised to a standard region, the differences in the sum total violation between two or more genotypes is an immediate indication of the varying levels of feasibility. In other words, normalisation allows the GA to determine intrinsically, without additional ranking processes, the number of constraints satisfied by a genotype.

Studies have indicated that this knowledge of the Pareto-based dominance (see Chapters 5 and 6 for practical applications) is sufficient to gear the search towards optimal regions. The proposed constraint handling method also pushes the robustness of GA by providing only partial knowledge of fitness in the form of lazy evaluation and artificial prioritisation of constraints. The advantage gained is in terms of savings in computation overhead, which is particularly beneficial to constraints that would have otherwise incurred expensive computation and analysis. This approach models a progressive adaptation to increasing number of environmental constraints observed in natural systems. The more constraints genotypes have adapted to, the greater their complexity, and hence, preference. Even without specialised genotypic representation and genetic interactions, results show that the typical GA framework deployed is able to cope well with the apparent lack of complete fitness knowledge on high-dimensional problems with a large number of constraints. Furthermore, genotypes in a population are allowed to be at varying stages of feasibility.

7.1.2.1 DYNAMIC PRIORITISATION OF CONSTRAINTS – FUTURE WORK

An area requiring further developments is in the decision of prioritising constraints. Most of the time, there exist no such priorities in constraint handling. It is believed that the ordering of constraints will inevitably affect performance and convergence rates. This is due mainly to the delineation of the search boundaries into regions of feasibility and infeasibility. Assuming that the feasible region is continuous regardless of the number of constraints in effect, imposing each constraint independently will result in feasible regions of differing sizes, such as that shown in Figure 7-1, where the true feasibility is in the intersection of all constraints.

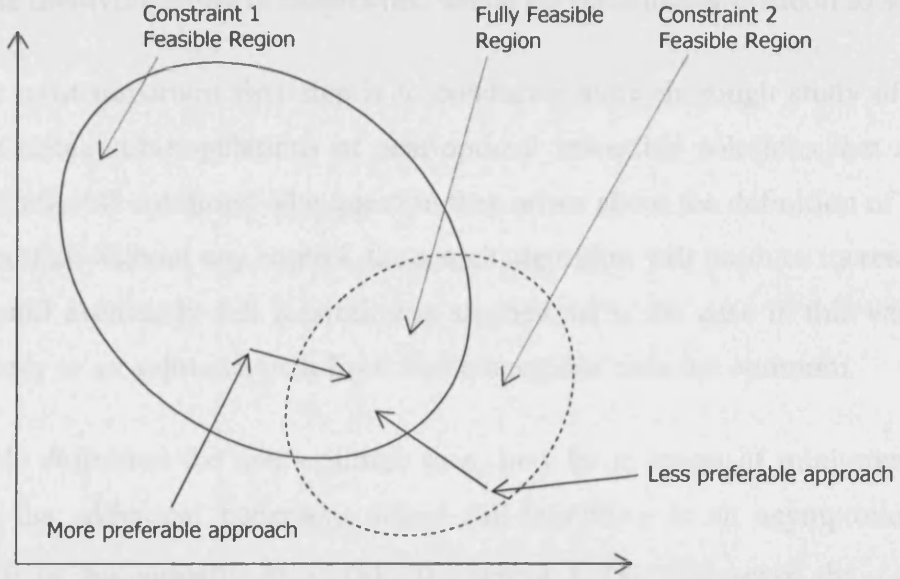

Figure 7-1 Feasible regions imposed by the first and second constraints, and the fully feasible region located at the intersection of both.

Future work can possibly concentrate on developing a method of determining, internally and dynamically in the GA framework, the difficulty imposed by constraints or constraint 'hardness'. Once hardness can be decided to a degree of certainty, the prioritisation can be reordered during run-time to produce a more progressive approach towards the true feasible region, also illustrated in Figure 7-1. In this manner, genotypes are expected to adapt more steadily towards regions of increasing feasibility without hovering too long in infeasible or semi-feasible regions. The effect is particularly apparent when feasible regions are almost negligible when compared to the overall search space in GA-difficult problems, with equality constraints being obvious examples. If tight constraints are at the top of the prioritised list of constraints, there is a tendency for genotypes to either be infeasible or simultaneously satisfying several constraints at once. Dynamic prioritisation is also believed to be able to also increase reliability in discovering optimal regions. One possibility is to progressively shift the approach paths to various feasible regions from various perspectives.

7.1.2.2 CLUSTERING FOR CONSTRAINTS SPECIATION – FUTURE WORK

Although the lazy evaluation of constraint violations is able to introduce infeasible genotypes into the population, masked as feasible individuals due to sub-optimal objective values, further work can be done to increase the robustness in discovering the optimum. The benefit in a mixture of differing level of feasibility is the opening up of extra paths from which to approach the optimum. It may also be suitable for seemingly intractable problems for GA, such as those involving equality constraints, which are notoriously difficult to solve with GA.

Perhaps the most important first step is to conduct a more thorough study of the effects of maintaining stable sub-populations of near-optimal infeasible solutions that are allowed to interact with feasible solutions. The question then arises about the definition of 'near-optimal'. It is expected that without any control, the search algorithm will produce increasingly feasible genotypes until eventually full feasibility is attained, as is the case in this work. This leads eventually only to an approach path from feasible regions onto the optimum.

One probable definition for near-optimal, then, may be in terms of minimum violation but introducing the additional constraint where full-feasibility is an asymptotic event, either $\sum_i C_i(\mathbf{x}) > 0$ or individually as $C_i(\mathbf{x}) > 0$, where $C_i(\mathbf{x})$ represents the i-th constraint. Coupled with suitably adapted clustering algorithms used for data mining and data clustering, such as one explored by Streichert et al., (2003), feasibility boundaries may be detected by forming 'species' of genotypes around such boundaries. Species then, inadvertently, specialise into performing a specific role, whether it is to minimise the violation of one constraint, several constraints, or objective cost reduction with full feasibilities, etc. In fact, such an approach may be more suitable using a parallel implementation with individual 'islands' enforcing different 'environmental' constraints and objectives. Inter-species interaction should then be encouraged to allow information exchange, hopefully producing better individuals with increasing adaptability to a range of constraints and objective(s).

7.1.3 Improving Search Resolution

During the course of this work, one of the main difficulties encountered by GA was the lack of ability in focusing onto an optimum. This can perhaps be attributed to the large search space as a result of the bounds imposed on the design variables. Large search spaces have also been the main reason for turning to other genetic encodings that are more exact representations of the problem involved.

To illustrate using binary encoding (Gray or standard), the number of bits in a chromosome can be said to be proportional to some extent to the precision required of the solution of a numerical design variable. As more design variables are included in the search, the difficulty of reaching a solution of high-precision becomes increasingly apparent. This is due largely to the almost improbable event of operators producing the true optimum solution. In studies conducted here, the responsibility fell mainly on two operators – uniform crossover and uniform mutation. It should be said that discovering the true optimum with the required precision is not impossible, but merely difficult with only random changes induced by genetic operators.

7.1.3.1 DIVIDE-AND-CONQUER – FUTURE WORK

Perhaps the most direct and simple approach to solving this resolution problem can be done via some form of 'divide-and-conquer' technique. As search spaces become excessively large, a single run will require prohibitive computation time in reaching the optimum for reasons stated earlier.

Literally dividing the search domain into smaller domain components can be a simple alternative. However, such an approach breaks down in high-dimensionality problems. Consider a problem with 5 design variables x_i, each with their respective lower and upper bounds, $x_i^{LB} \leq x_i \leq x_i^{UB}$. Assuming that, at the very least, the bounds are divided into non-overlapping lower and upper halves, represented by the new set of variables x_i^- and x_i^+ respectively. In order to cover all combinations, up to 32 set of separate studies must be conducted. Hence, the problem becomes intractable as more design variables are included.

Future work can, therefore, develop systems that are able to decide, to a reasonable degree of accuracy, the approximate location of potentially good, and hopefully, optimal solutions. This information can then be used by future generations of genotype to confirm or refute the fact. Such an approach may be successful since GA is able to detect optimal regions rather easily, but slows down considerably in fine-resolution situations.

The search starts off with a population of genotypes with insufficient number of bits (assuming binary representation) required to represent the precision expected in full. Rapid bursts of successive generations should be able to discover potentially good locations from which to expand resources on future searches. Once known, the bounds governing the bits are then updated to reflect these localised, potentially optimal, regions. Another possibility is to regenerate a new set of genotypes that are tailored specifically for such known regions. Again, due to the small number of bits, the search should again be able to detect rapidly even more optimal regions. The process can then be repeated until the required precision has been achieved.

Several precautionary steps should be taken. One is the possibility of missing global optima, particularly those that are tight and situated in an otherwise flat landscape, a steep optimal peak, for example. The resolution of the genotypes must also be able to detect substantial changes in objective values. Otherwise no decision can be made about optimal regions. Another is the probable failure in detecting optima in multimodal problems.

7.1.3.2 GENOTYPE CREEP – FUTURE WORK

Following from the fact that GA is useful as a higher level tool for detecting optimal regions, efficiency in focusing onto true optimal points should be increased. One possibility that can be explored is the use of 'empty' genes that are specialised according to the location of a genotype.

For example, consider, without loss of generality, a hypothetical 2-dimensional unconstrained minimisation problem.

Minimise

$$f(x_1,x_2) \qquad 7\text{-}1$$

where

$$0 \leq x_1, x_2 \leq 127$$

Consider also that the search space is represented by a two-chromosome binary-encoded genotype, where each chromosome has 7 bits (additional bits may be added to increase precision where necessary). A random genotype may then be

$$G = [0001010 \quad 0011000]$$

which represents the phenotype $x_1 = 10$ and $x_2 = 24$. Consider then that the last few bits of each chromosome are in fact 'empty' genes

$$G = [0001{*}{*}{*} \quad 0011{*}{*}{*}]$$

which is not unlike a schema. The genotype now represents the bounds $8 \leq x_1 \leq 15$ and $24 \leq x_2 \leq 31$, as shown in Figure 7-2. Depending on the number of empty genes to consider, the genotypes can move, or 'creep', across this restricted space to the best position that gives an optimum objective value locally. It is believed that such a creep operator will provide fine-grain tuning of genotypes compared to the usual crossover/mutation combination. It is worth stating that the usual genetic operators should still be involved to provide coarse-grain resolving of global optimal regions.

Although using GA to set a creep path to the local optimum will provide a consistent framework, it is believe that other search methodologies may also be useful here to create a hybrid search system. Techniques such as hill-climbing, binary partial enumeration or even exhaustive search of the creep region may be used, provided that such methods give excellent reliability with reasonable speed. It is important that the creep boundaries restrict a negligible space compared to the overall search domain. Otherwise, the localised search process may prove too difficult to be successful. With the schema analogy, the method can also be generalised further to allow empty genes to be in any loci the genotype, e.g.

$$G = [1*1**00 \quad 00***1*]$$

7.2 Composite Laminates

A simple GA framework complete with efficient constraint handling technique has shown good robustness in designing a range of different structures made up of composite laminates and other mixed materials. Studies range from simple plates to stiffened prismatic plate assemblies that allowed simultaneous design of stacking sequences and discrete thicknesses, as well as multi-objective mass-cost optimisation. However, the studies have still considered only a relatively small number of constraints that should ideally be enforced.

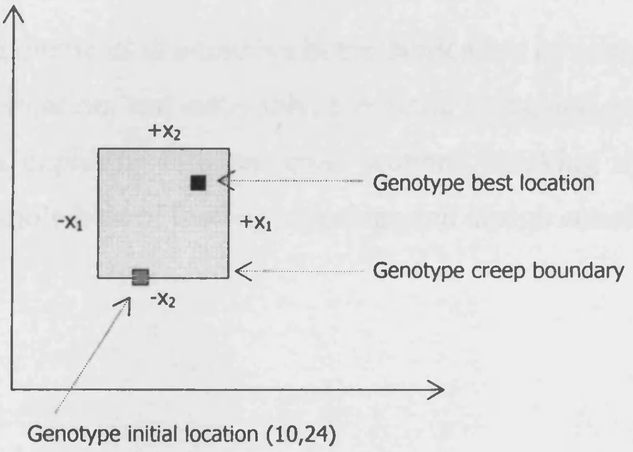

Figure 7-2 Boundary induced by the creep operator and movement from an initial to a locally optimal location.

7.2.1 Design Constraints and Objectives

Studies here have concentrated only on buckling. Future design and optimisation studies should include as many design constraints as possible, such as buckling, vibration, material strength, displacement amplitudes, good design practices, etc., to provide a more realistic simulation of real-world demands on composite-based structures.

Objectives do not necessary have to be mass alone, although it is arguably the most sought after. This is primarily due to the majority of the use of composite laminates falling into weight-critical applications, such as aerospace. Nonetheless, the directional specificity of composite laminates is much too beneficial to ignore, provided that issues such as cost can be resolved. Therefore, multi-objective optimisations involving mass and cost can be further explored. Perhaps the most direct way to cost-saving use of composite laminates is in the design of a more varied range of standard laminates, apart from the usual quasi-isotropic laminates $[\pm 45/90/0]_S$ or its variants.

Another avenue of exploration for lowering costs and mass can be in the variation of fibre density in standard plies of unidirectional layers, since cost and mass are also affected by the fibre volume fraction. The use of rule-of-mixtures may well be used to provide theoretical estimation[1] on the elastic properties of such configurations, and hence fibre/matrix volume fractions values that give optimal performance in terms of mass, cost or some intermediary compromise.

7.2.2 Shape optimisation

The use of repeating elements is attractive in the fabrication of composite-based structure as it reduces work on fabrication, and inevitably, economic costs, and ease of assembly. This leads to the possibility of exploring efficient cross-sections, stacking sequences, and thicknesses that are ideal for a whole host of loading situations and design constraints.

[1] The rule-of-mixtures gives, at best, rough estimates of material properties. Physical tests are needed to give accurate properties before such theoretical designs can be used.

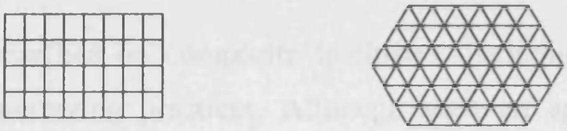

Figure 7-3 Variation in repeating geometrical shapes and resulting structural members.

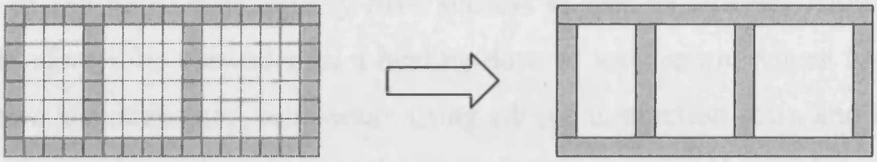

Figure 7-4 Topological grid with locations that require mass (shaded) and a possible interpretation of the resulting shape.

An interesting case can be the 2-dimensional repetition of elements with basic geometrical cross-sectional shapes, as shown in Figure 7-3. The number of repeating elements in the transverse and lateral directions obviously results in different types of structures. For example, the diagrams when assigned lengths substantially greater in magnitude in breadth or depth will result in long and slender structural members. Depending on the primary loading direction, these members will then be either beams or columns.

Topological optimisation of cross-sectional shapes also has potential for future developments. In the case of the design of laminates with longitudinal voids, the position and location of voids have been fixed in advance, with only the number and dimensions of these voids as design variables. Topological optimisation can potentially further the generalisation and investigate efficiency that can be gained from bespoke shapes.

It typically begins by setting up a system of grids, not unlike the repeating elements as shown in Figure 7-3, and search for optimal configurations where objective(s) are optimised, such as that shown in Figure 7-4. Care must be taken to ensure that continuity is observed so that resulting cross-sectional shapes are not composed of individual unconnected elements. It is believed that different shapes should result with different loading conditions and factors.

7.3 Closing Remarks

Most of the current researches on composite laminates, including this work, have rarely satisfied common manufacturing practices. Although such an approach is laudable as it immediately gives a general overview of design problems, it lacks the ability to drive further developments in production and fabrication.

On the other hand, GA has seen so many advances in all aspects, from genetic representation to specialised genetic operators, that it is becoming increasingly complicated. Of course, developments in any fields stem equally from success as well as failures. Although proof-by-analogy should always be viewed with a healthy dose of scepticism, nature has been able to develop complex structures and behaviours using simple interaction rules and small building blocks. It is perhaps wise to draw lessons from this in future approaches to GA.

REFERENCES

Almroth, B. O., and Brogan, F. A., (1978), *The STAGS Computer Code, NASA CR-2950.*

Anderson, M. S., Williams, F. W., and Wright, C. J., (1983), "Buckling and Vibration of any Prismatic Assembly of Shear and Compression Loaded Anisotropic Plates with an Arbitrary Supporting Structure", *International Journal of Mechanical Sciences*, Vol. 25, pp. 585-596.

Baburaj, V., and Kalyanaraman, V., (1993), "STIPSY – A Program for Stiffened Panel Synthesis", *Computers & Structures*, Vol. 48, No. 2, pp. 347-355.

Bäck, T., (1992), "Self-Adaptation in Genetic Algorithms", *Proceedings of the First European Conference on Artificial Intelligence*, MIT Press, pp. 263-271.

Bäck, T., (1993), "Optimal Mutation Rates in Genetic Search", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 2-8.

Bäck, T., (1994), "Evolutionary Algorithm: Comparison of Approaches", *Computing with Biological Metaphors*, Chapman and Hall, Cambridge, UK.

Baker, J. E., (1987), "Reducing Bias and Inefficiency in the Selection Algorithm", *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21.

Beasley, D., Bull, D. R., and Martin, R. R., (1993), "A Sequential Niche Technique for Multimodal Function Optimization", *Evolutionary Computation*, Vol. 1, No. 2, pp. 101-125.

Bertoni, A., and Dorigo, M., (1993), "Implicit Parallelism in Genetic Algorithms", *Artificial Intelligence*, Vol. 61, No. 2, pp. 307-314.

Bushnell, D., (1987a), "Theoretical Basis of the PANDA Computer Program for Preliminary Design of Stiffened Panels under Combined In-Plane Loads", *Computers & Structures*, Vol. 25, No. 4, pp. 469-605.

Bushnell, D., (1987b), "PANDA2- Program for Minimum Weight Design of a Stiffened, Composite, Locally Buckled Panels", *Computers & Structures*, Vol. 25, No. 4, pp. 469-605.

Bushnell, D., (1994), "Minimum Weight Design of a Stiffened Panel via PANDA2 and Evaluation of the Optimized Panel via STAGS", *Computer & Structures*, Vol. 50, No. 4, pp. 569-602.

Bushnell, D., Holmes, A. M. C., Flaggs, D. L., and McCormick, P. J., (1988), "Optimum Design, Fabrication and Test of Graphite-Epoxy, Curve Stiffened, Locally Buckled Panels Loaded in Axial Compression", *Buckling of Structures*, Elsevier, Amsterdam, pp. 61-131.

Butler, R., (1994), "Optimum Design and Evaluation of Stiffened Panels with Practical Loading", *Computers & Structures*, Vol. 52, No. 6, pp. 1107-1118.

Butler, R., (1995), "Optimum Design of Composite Stiffened Wing Panels – A Parametric Study", *Aeronautical Journal*, Vol. 99, No. 985, pp. 169-177.

Butler, R., and Williams, F. W., (1990), "Optimum Design Features of VICONOPT, an Exact Buckling Program for Prismatic Assemblies of Anisotropic Plates", *Proceedings of the 31st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 1289-1299.

Butler, R., and Williams, F. W., (1992), "Optimum Design Using VICONOPT, a Buckling and Strength Constraint Program for Prismatic Assemblies of Anisotropic Plates", *Computers & Structures*, Vol. 43, No. 4, pp. 699-708.

Callahan, K. J., and Weeks, G. E., (1992), "Optimum Design of Composite Laminates Using Genetic Algorithms", *Composite Engineering*, Vol. 2, Issue 3, pp. 149-160.

Camponogara, E., and Talukdar, S. N., (1997), "A Genetic Algorithm for Constrained and Multiobjective Optimization", *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, pp. 49-62.

Caruana, R. A., Eshelman, L. J., and Schaffer, J. D., (1989), "Representation and Hidden Bias II: Eliminating Defining Length Bias in Genetic Search via Shuffle Crossover", *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, pp. 750-755.

Coello, C. A. C., (1999a), "The Use of Multiobjective Optimization Technique to Handle Constraints", *Second International Symposium on Artificial Intelligence*, pp. 251-256.

Coello, C. A. C., (1999b), "Constraint-Handling through a Multiobjective Optimization Technique", *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO-99). Workshop Program*, pp. 46-53.

Coello, C. A. C., (1999c), "A Survey of Constraint Handling Techniques used with Evolutionary Algorithms", Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada.

Cohen, G. A., (1979), "FASOR – A Second Generation Shell of Revolution Code", *Computers & Structures*, Vol. 10, pp. 301-309.

Costa, L. A., Oliveira, P., Figueiredo, I. N., Roseiro, F., and Leal, R. P., (2000), "Structural Optimization of Laminated Plates with Genetic Algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pp. 621-627.

Crossley, W. A., and Laananen, D. H., (1996), "Genetic Algorithm-Based Optimal Design of Stiffened Composite Panels for Energy Absorption", *Proceedings of the American Helicopter Society 52nd Annual Forum*, pp. 1367-1376.

Davis, L., (Ed.), (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold.

Dawkins, R., (2003), *A Devil's Chaplain – Selected Essays*, Weidenfeld & Nicholson.

De Jong, K. A., and Spears, W. M. (1990), "An Analysis of the Interacting Roles of Population Size and Crossover", *Proceedings of the International Workshop Parallel Problem Solving from Nature*, pp. 38-47.

Deb, K., (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, UK Wiley.

Deb, K., and Goldberg, D. E., (1989), "An Investigation of Niche and Species Formation in Genetic Function Optimization", *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pp. 42-50.

Edwards, D. A., Williams, F. W., and Kennedy, D., (1998), "Cost Optimization of Stiffened Panels Using VICONOPT", *AIAA Journal*, Vol. 36, No. 2, pp. 267-272.

Eshelman, L. J., (1995), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., (1996), "A Density-Based Algorithm in Discovering Clusters in Large Spatial Databases with Noise", *Proceedings 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AIAA Press, pp. 226-231.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., (1998), "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications", *Data Mining and Knowledge Discovery*, Journal 2, Kluwer Academic Publishers, pp. 169-194.

Fogarty, T. C., (1989), "Varying the Probability of Mutation in Genetic Algorithm", *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pp. 104-109.

Fonseca, C. M., and Fleming, P. J., (1993), "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416-423.

Fonseca, C. M., and Fleming, P. J., (1998a), "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: A Unified Formulation", *IEEE Transactions on Systems, Man and Cybernetics, Part A: System and Humans*, Vol. 28, No. 1, pp. 26-37.

Fonseca, C. M., and Fleming, P. J., (1998b), "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part II: Application Example", *IEEE Transactions on Systems, Man and Cybernetics, Part A: System and Humans*, Vol. 28, No. 1, pp. 38-47.

Gent, I. P., MacIntyre, E., Prosser, P., and Walsh, T., (1996), "The Constrainedness of Search", *Proceedings of AAAI-96*, Morgan Kaufmann, pp. 246-252.

Goldberg, D. E., (1987), "Simple Genetic Algorithms and the Minimal, Deceptive Problem", *Genetic Algorithms and Simulated Annealing*, Morgan Kauffman, pp. 74-88.

Goldberg, D. E., (1989a), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

Goldberg, D. E., (1989b), "Messy Genetic Algorithms: Motivation, Analysis, and First Results", *Complex Systems*, Vol. 3, pp. 493-530.

Goldberg, D. E., (1990a), "Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale", *Complex Systems*, Vol. 4, pp. 415-444.

Goldberg, D. E., (1990b), "Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking", IlliGAL Technical Report No. 90001, University of Illinois.

Goldberg, D. E., (1991), Deb, K., and Korb, B., "Do No Worry, Be Messy", *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 24-30.

Goldberg, D. E., and Richardson, J., (1987), "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49.

Grefenstette, J. J., (1991), "Conditions for Implicit Parallelism", *Foundations of Genetic Algorithms*, Morgan Kaufmann, pp. 252-261.

Grefenstette, J. J., (1992), "Genetic Algorithms for Changing Environments", *Parallel Problem Solving from Nature, 2 – PPSN II*, pp. 137-144.

Grefenstette, J. J., and Baker, J. E., (1989), "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism", *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 20-27.

Grosset, L., Venkataraman, S., and Hafta, R. T., (2001), "Genetic Optimization of Two-Material Composite Laminates", *Proceedings 16th Annual Technical Conference American Society for Composites*, Virginia Tech.

Grosset, L., Venkataraman, S., and Haftka, R. T., (2002), "Probability-Based Genetic Algorithms for Composite Laminate Optimization", *Proceedings 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.*

Gürdal, Z., Haftka, R. T., Nagendra, S., (1994), "Genetic Algorithms for the Design of Laminated Composite Panel", *SAMPE Journal*, Vol. 30, Issue 3, pp. 29-35.

Gürdal, Z., Haftka, R. T., and Nagendra, S., (1999), *Design and Optimization of Laminated Composite Materials*, John Wiley & Sons.

Hancock, P. J. B., (1994), "Genetic Search Strategies in Multicriterion Optimal Design", *Evolutionary Computing, AISB Workshop*, Vol. 865 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 80-94.

Harrison, P. N., LeRiche, R., and Haftka, R. T., (1996), "Design of Stiffened Composite Panels by Genetic Algorithm and Response Surface Approximations", *AIAA Paper 95-1163-CP.*

Hesser, J., and Männer, R., (1991), "Towards an Optimal Mutation Probability in Genetic Algorithms", *Parallel Problem Solving from Nature, 1st Workshop, PPSN I*, pp. 23-32.

Holland, J. H., (1975), *Adaptation in Natural and Artificial Systems – An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, University of Michigan Press.

Holland, J. H., (1998), *Emergence – From Chaos to Order*, Oxford University Press.

Homaifar, A., Lai, S. H.-Y, and Qi, X., (1994), "Constrained Optimization via Genetic Algorithms", *Simulation*, Vol. 62, pp. 242-254.

Jiménez, F., and Verdegay, J. L., (1999), "Evolutionary Techniques for Constrained Optimization Problems", *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99)*, Springer-Verlag.

Johnson, S., (2001), *Emergence*, Scribner.

Joines, J. A., and Houck, C. R., (1994), "On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GAs", *Proceedings of the First IEEE International Conference on Evolutionary Computation*, Vol. 1, pp. 579-584.

Kennedy, D., Ioannidis, G., and Featherston, C. A., (2004), "Discrete Optimum Design of Composite Plates Including Longitudinal Voids", *Proceedings of 6th World Congress on Computational Mechanics*, Paper M636, pp. 1-10.

Kennedy, D., Ong, T. J., O'Leary, O. J., and Williams, F. W., (1999), "Practical Optimisation of Aerospace Panels", *Proceedings of 1st ASMO UK/ISSMO Conference*, pp. 217-224.

Kennedy, D., and Williams, F. W., (1992), "More Efficient Use of Determinants to Solve Transcendental Structural Eigenvalue Problems Reliably", *Computers & Structures*, Vol. 41, pp. 973-979.

Kosigo, N., Watson, L. T., Gürdal, Z., and Haftka, R. T., (1994), "Genetic Algorithms with Local Improvement for Composite Laminate Design", *Structural Optimization*, Vol. 7, Issue 4, pp. 207-218.

LeRiche, R., and Gaudin, J., (1998), "Design of Dimensionally Stable Composites by Evolutionary Optimization", *Composite Structures*, Vol. 41, No. 2, pp. 97-111.

LeRiche, R., and Haftka, R. T., (1993), "Optimization of Laminate Stacking Sequence for Buckling Load Maximisation by Genetic Algorithms", *AIAA Journal*, Vol. 31, No. 5, pp. 951-956.

Liu, B., Haftka, R. T., Akgun M. A., Todoroki, A., (1998), "Permutation Genetic Algorithm for Stacking Sequence Design of Composite Laminates", *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.*

Michalewicz, Z., (1995), "Genetic Algorithms, Numerical Optimization and Constraints", *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 98-108.

Michalewicz, Z., and Attia, N., "Evolutionary Optimization of Constrained Problems", *Parallel Problem Solving from Nature, 1st Workshop, PPSN I*, pp. 98-108.

Michalewicz, Z., and Janikow, C., (1991a), "Genetic Algorithms for Numerical Optimization", *Statistics and Computing*, Vol. 1, No. 1, 1991.

Michalewicz, Z., and Janikow, C., (1991b), "Handling Constraints in Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 151-157.

Mitchell, M., (1996), *An Introduction to Genetic Algorithms – Complex Adaptive System Series*, Bradford Publishing.

Mitchell, M., Forrest, S., and Holland, J. H., (1991), "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance", *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, MIT Press.

Mühlenbein, H., (1991), "Evolution in Space and Time – the Parallel Genetic Algorithm", *Foundations of Genetic Algorithms*, Morgan Kauffman, pp. 316-338.

Nagendra, S., Haftka, R. T., and Gürdal, Z., (1993) "Design of a Blade Stiffened Composite Panel by a Genetic Algorithm", *Proceedings of the 34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials*, pp. 2418-2436.

Nagendra, S., Jestin, D., Gürdal, Z., Haftka, R. T., and Watson, L. T., (1996), "Improved Genetic Algorithm for the Design of Stiffened Composite Panels", *Computers & Structures*, Vol. 58, pp. 543-555.

Niu, M. C. Y., (1993), *Composite Airframe Structures*, Conmilit Press.

Parmee, I. C., and Purchase, G., (1994), "The Development of a Directed Genetic Search Technique for Heavily Constrained Spaces", *Proceedings of Adaptive Computing in Engineering Design and Control, PEDC*, University of Plymouth.

Peters, F. J., (1979), "On the Implementation of an Application Oriented Software System", *International Journal for Numerical Methods in Engineering*, Vol. 14, No. 10, pp. 1477-1497.

Potgeiter, E., and Stander N., (1998), "The Genetic Algorithm Applied to Stiffness Maximization of Laminated Plates: Review and Comparison", *Structural Optimization*, Vol. 15, pp. 221-229.

Powell, D., and Skolnick, M. M., (1993), "Using Genetic Algorithms in Engineering Design Optimization with Non-Linear Constraints", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 424-430.

Sargent, P. M., Ige, D. O., and Ball, R. N., (1995), "Design of Laminate Composite Lay-Ups using Genetic Algorithm", *Engineering with Computers*, Vol. 11, pp. 59-69.

Schaffer, J. D., (1985), "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100.

Schoenauer, M., and Xanthakis, S., (1993), "Constrained GA Optimization", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 573-380.

Smith, C. S., (1968), "Bending, Buckling and Vibration of Orthotropic Plate-Beam Structures", *Journal of Ship Research*, Vol. 12, pp. 249-268.

Spears, W. M., and De Jong, K. E., (1990), "An Analysis of Multi-Point Crossover", *Proceedings of the Foundations of Genetic Algorithms Workshop*, pp. 301-315.

Spears, W. M., and De Jong, K. E., (1991), "On the Virtues of Parameterized Uniform Crossover", *Proceedings of the International Conference on Genetic Algorithms*, pp. 230-236.

Stephens, C. R., and Waelbroeck, H., (1999), "Schemata Evolution and Building Blocks", *Evolutionary Computation*, Vol. 7, No. 2, pp. 109-124.

Sterelny, K., (2001), *Dawkins vs. Gould – Survival of the Fittest*, Icon Books.

Streichert, F., Stein, G., Ulmer, H., and Zell, A., (2003), "A Clustering Based Niching EA for Multimodal Search Spaces", *Evolutionary Artificielle 6th International Conference (EA-2003)*, Springer, pp. 293-304.

Stroud, W. J., and Agranoff, N., (1977), "Minimum-Mass Design of Filamentary Composite Panels under Combined Loads: Design Procedure based on a Rigorous Buckling Analysis", *NASA-TN D-8417*.

Surry, P. D., Radcliffe, N. J., and Boyd, I. D., (1995), "A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks: The COMOGA Method", *Evolutionary Computing, AISB Workshop, Lecture Notes in Computer Science*, Springer, pp. 166-180.

Syswerda, G., (1989), "Uniform Crossover in Genetic Algorithms", *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pp. 2-9.

Todoroki, A., and Haftka, R. T., (1997), "Stacking Sequence Matching by a Two-Stage Genetic Algorithm with Consanguineous Initial Population", *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Part 2 (of 4), pp. 1297-1302.

Vanderplaats, G. N., (1973), "CONMIN: A FORTRAN Program for Constrained Function Minimization", *Nasa Technical Memorandum*, TMX-62282, Ames Research Centre.

Vanderplaats, G. N. and Moses, F., (1973), "Structural Optimization by Methods of Feasible Directions", *Computer & Structures*, Vol. 3, pp. 739-755.

Viswanathan, A. V., Tamekuni, M., and Tripp, L. L., (1973), "Elastic Stability of Biaxially Loaded Longitudinally Stiffened Composite Structures", *AIAA Journal*, Vol. 11, pp. 1553-1559.

Vose, M. D., "Generalizing the Notion of Schema in Genetic Algorithms", *Artificial Intelligence*, Vol. 50, No. 3, pp. 385-396.

Whetsone, W. D., (1980), "EISI-EAL: Engineering Analysis Language", *Proceedings of the 2nd Conference in Civil Engineering*, ASCE, pp. 276-285.

Whitley, D., Starkweather, T., and Shaner, D., (1991), "The Travelling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination", Davis, L., (Ed.), , *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, pp. 350-372.

Williams, F. W., and Anderson, M. S., (1983), "Incorporation of Lagrangian Multipliers into an Algorithm for finding Exact Natural Frequencies or Critical Buckling Loads", *International Journal of Mechanical Sciences*, Vol. 8, pp. 579-584.

Williams, F. W., Anderson, M. S., Kennedy, D., Butler, R., and Aston, G., (1990a), "User Manual for VICONOPT: An Exact Analysis and Optimum Design Program Covering the Buckling and Vibration of Prismatic Assemblies of Flat In-Plane Loaded, Anisotropic Plates, with Approximations for Discrete Supports and Transverse Stiffeners", *NASA Contractor Report 181966.*

Williams, F. W., and Kennedy, D., (1988), "Reliable Use of Determinants to Solve Non-Linear Structural Eigenvalue Problems Efficiently", *International Journal for Numerical Methods in Engineering*, Vol. 26, pp. 1825-1841.

Williams, F. W., Kennedy, D., and Anderson, M. S., (1990b), "Analysis Features of VICONOPT, an Exact Buckling and Vibration Program for Prismatic Assemblies of Anisotropic Plates", *Proceedings of the 31st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pp. 920-929.

Williams, F. W., and Wright, C. J., (1978), "A Compact Computer Program for Calculating Buckling Stresses and Natural Frequencies of Vibration of Prismatic Plate Assemblies", *International Journal for Numerical Methods in Engineering*, Vol. 12, pp. 1429-1456.

Williams, F. W., and Ye, J., (1992a), "Optimum Mass Design of Laminated Plates with Longitudinal Mid-Surface Voids", *Computers & Structures*, Vol. 43, pp. 265-272.

Williams, F. W., and Ye, J., (1992b), "Optimum Mass Design of Prismatic Assemblies of Plates with Longitudinal Voids", *Computers & Structures*, Vol. 44, pp. 557-565.

Williams, P. W. L., Williams, F. W., and Kennedy, D., (1996), "Inclusions of Transverse Shear Deformation in Optimum Design of Aircraft Wing Panels", *AIAA Journal*, Vol. 34, pp. 2456-2458.

Wittrick, W. H., (1968a), "A Unified Approach to the Initial Buckling of Stiffened Panels in Compression", *Aeronautical Quarterly*, Vol. 19, pp. 265-283.

Wittrick, W. H., (1968b), "General Sinusoidal Stiffness Matrices for Buckling and Vibration Analyses of Thin Flat-Walled Structures", *International Journal of Mechanical Sciences*, Vol. 10, pp. 949-966.

Wittrick, W. H., and Williams, F. W., (1971a), "Natural Vibrations of Thin, Prismatic, Flat-Walled Structures", *Proceedings of the International Union of Theoretical and Applied Mechanics Symposium on High Speed Computing of Elastic Structures*, University of Liege.

Wittrick, W. H., and Williams, F. W., (1971b), "A General Algorithm for Computing Natural Frequencies of Elastic Structures", *Quarterly Journal of Mechanics and Applied Mathematics*, Vol. 24, pp. 263-284.

Wittrick, W. H., and Williams, F. W., (1973), "An Algorithm for Computing Critical Buckling Loads of Elastic Structures", *Journal of Structural Mechanics*, Vol. 1, pp. 497-518.

Wittrick, W. H., and Williams, F. W., (1974), "Buckling and Vibration of Anisotropic or Isotropic Plate Assemblies under Combined Loadings", *International Journal of Mechanical Science*, Vol. 16, pp. 209-239.

York, C. B., Williams, F. W., and Kennedy, D., (1993), "A Parametric Study of Optimum Designs for Benchmark Stiffened Wing Panels", *Composites Engineering*, Vol. 3, No. 7-8, pp. 619-632.

APPENDIX I: GRAY CODE CONVERSIONS

The conversion procedures from a standard binary code $b = \langle b_1\ b_2\ ...\ b_i\ ...\ b_n \rangle$ to the binary-reflected Gray code $g = \langle g_1\ g_2\ ...\ g_i\ ...\ g_n \rangle$ and vice-versa are as given in Figure AI-1. Figure AI-2 lists the first non-negative integers and its binary and Gray code equivalents.

```
Binary-to-Gray
Begin
          $g_1 = b_1$
          for $i = 2$ to $n$, do
                    $g_i = b_{i-1}\ XOR\ b_i$
End


Gray-to-Binary
Begin
          $v = g_1$
          $b_1 = v$
          for $i = 2$ to $n$, do
                    if $g_i = 1$ then $v = NOT\ v$
                    $b_i = v$
End
```

Figure AI-1 Conversion procedures for Binary-to-Gray and vice-versa.

Decimal	Standard Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Figure AI-2 Equivalent decimal, standard binary and Gray codes.

APPENDIX II: UNIQUE SOLUTIONS TO THE BENCHMARK PLATE

Figure AII-1 shows all 128 unique optimum solutions found using GA, as outlined in Chapter 5. Results are a combination of 10 independent runs, and are presented in two columns for compactness.

$[45_2/-45_6/45_4/0_4/45/-45/0_5]_S$
$[45_2/-45_6/45_4/0_4/-45/45/90_5]_S$
$[45_2/-45_6/45_4/0_5/90_6]_S$
$[45_2/-45_6/45_4/0_6/90/45_2/-45_2]_S$
$[45_2/-45_6/45_4/0_6/90_5]_S$
$[45_2/-45_7/45_5/0_4/90_5]_S$
$[45_3/-45_6/45_3/0/90_2/0_2/90_6]_S$
$[45_3/-45_6/45_3/0_2/90_2/0_7]_S$
$[45_3/-45_6/45_3/0_3/90_2/0_6]_S$
$[-45_3/45_6/-45_3/0/90/0_3/90_6]_S$
$[-45_3/45_6/-45_3/0_2/90/0_2/90_6]_S$
$[-45_3/45_6/-45_3/0_2/90/0_8]_S$
$[45_8/-45_8/0_7]_S$
$[-45_3/45_4/-45_5/45_4/0_7]_S$
$[-45_3/45_5/-45_2/0_{10}/90_3]_S$
$[-45_3/45_5/-45_2/0_{11}/90_2]_S$
$[-45_3/45_5/-45_2/0_{13}]_S$
$[-45_3/45_6/-45_3/0_5/90/0/-45_2/45_2]_S$
$[-45_3/45_6/-45_3/0_6/90_5]_S$
$[-45_3/45_3/90/-45_4/45_5/-45/0_4/90_2]_S$
$[-45_3/45_3/90/-45_5/45_5/0_4/90_2]_S$
$[-45_6/45_8/-45_2/0_7]_S$
$[-45_6/45_6/90_2/-45_2/45_2/0_5]_S$
$[-45_7/45_9/-45_2/0_5]_S$
$[45/-45_4/45_4/-45/0_7/90_4/45/-45]_S$
$[45_2/-45_5/45_3/0_8/90_5]_S$
$[45_3/-45_5/45_2/0_{11}/-45/45]_S$
$[45_3/-45_5/45_2/0_{12}/90]_S$
$[45_3/-45_9/45_6/0_4/90]_S$
$[-45_8/45_8/0_6/90]_S$
$[-45_9/45_{11}/-45_2/90]_S$
$[-45_9/45_9/90_5]_S$
$[45_4/-45_4/0_3/-45_6/45_6]_S$
$[45_5/-45_6/45_2/-45/0_9]_S$
$[45_5/-45_6/45/0/90/0_9]_S$

$[45_5/-45_6/45/0_{11}]_S$
$[45_5/-45_6/45/90/0_{10}]_S$
$[45_5/-45_6/45/90_2/0_9]_S$
$[45_5/-45_7/45_2/0_2/90/0_6]_S$
$[45_5/-45_7/45_2/0_9]_S$
$[45_5/-45_8/45_3/90/0_6]_S$
$[45_5/-45_5/0/45/-45/0_2/90_2/0_6]_S$
$[45_5/-45_5/0/-45/45/0_{10}]_S$
$[45_6/-45_6/0_{11}]_S$
$[45_6/-45_6/0_2/90_3/0_6]_S$
$[45_6/-45_6/0_2/90_9]_S$
$[45_6/-45_6/90/0_{10}]_S$
$[45_6/-45_6/90_2/0/-45/45/0_6]_S$
$[45_7/-45_9/45_2/0_5]_S$
$[45_7/-45_7/0_2/90/0_6]_S$
$[45_7/-45_7/90_9]_S$
$[45_8/-45_8/0_7]_S$
$[45_6/-45_6/90_{11}]_S$
$[-45_2/45_5/-45_3/0_2/90_{11}]_S$
$[-45_3/45_5/-45_2/90_{11}/45/-45]_S$
$[-45_3/45_6/-45_3/0_2/90_9]_S$
$[-45_3/45_6/-45_3/90_{11}]_S$
$[-45_3/45_7/-45_4/90_9]_S$
$[-45_3/45_3/0/45_3/-45_3/90_8/-45/45]_S$
$[-45_4/45_6/-45_2/90_{11}]_S$
$[-45_4/45_6/-45_2/90_9/45/-45]_S$
$[-45_4/45_4/90_2/45_2/-45_2/90_9]_S$
$[-45_5/45_7/-45_2/90_9]_S$
$[-45_6/45_6/90_{11}]_S$
$[-45_7/45_7/90_9]_S$
$[-45_{11}/45_{11}/90]_S$
$[-45_3/45_3/90_2/45_5/-45_5/0_3/45/-45]_S$
$[-45_4/45_6/-45_2/0_3/-45_3/45_4/-45]_S$
$[-45_4/45_6/-45_2/0_3/-45_3/45_3/90_2]_S$
$[-45_4/45_6/-45_2/0_4/-45_3/45_3/90]_S$

$[-45_4/45_7/-45_3/0_4/90_3/-45/45]_S$

$[-45_4/45_7/-45_3/0_4/90_5]_S$

$[-45_4/45_8/-45_4/0_3/90_4]_S$

$[-45_4/45_8/-45_4/0_6/90]_S$

$[-45_4/45_{11}/-45_7/0]_S$

$[-45_4/45_4/0_2/-45_6/45_6/90]_S$

$[-45_4/45_4/90_2/45_4/-45_4/0_3/45/-45]_S$

$[-45_4/45_4/90_2/45_4/-45_4/0_3/-45/45]_S$

$[-45_4/45_4/90_2/45_4/-45_4/0_5]_S$

$[-45_6/45_9/-45_3/0_3/45/-45]_S$

$[-45_7/45_{11}/-45_4/90]_S$

$[45_3/-45_{11}/45_8/90]_S$

$[-45_{11}/45_{11}/90]_S$

$[-45_4/45_4/0/-45_6/45_7/-45]_S$

$[-45_5/45_5/0/-45_6/45_6]_S$

$[-45_6/45_6/90_2/-45/45/90_5/-45/45]_S$

$[90/-45_{10}/45_{10}/90_2]_S$

$[90/-45_{11}/45_{11}]_S$

$[-45_6/45_7/-45/0_4/90_5]_S$

$[-45_6/45_7/-45/90_9]_S$

$[-45_6/45_8/-45_2/90/0_4/90_2]_S$

$[-45_6/45_8/-45_2/90_7]_S$

$[-45_6/45_9/-45_3/90_5]_S$

$[-45_6/45_6/90/45_4/-45_4/90_2]_S$

$[-45_6/45_6/90/-45_4/45_4/90_2]_S$

$[-45_6/45_6/90_2/45_4/-45_4/90]_S$

$[-45_7/45_7/90/-45_3/45_4/-45]_S$

$[-45_7/45_7/90/-45_3/45_3/90_2]_S$

$[-45_7/45_7/90_7/-45/45]_S$

$[-45_8/45_8/90_5/45/-45]_S$

$[-45_8/45_8/90_7]_S$

$[45/-45_4/45_8/-45_5/0_5]_S$

$[45/-45_4/45_9/-45_6/0/90_2]_S$

$[45_6/-45_8/45_3/-45/0_5]_S$

$[45_6/-45_9/45_3/0_5]_S$

$[-45_3/45_9/-45_6/0_5]_S$

$[0/-45_3/45_8/-45_5/0_6]_S$

$[0/-45_3/45_9/-45_8/45_2]_S$

$[90/-45_2/45_7/-45_5/0_8]_S$

$[45_4/-45_5/45/0_3/45_5/-45_5]_S$

$[45_4/-45_4/0_3/-45_6/45_6]_S$

$[45_5/-45_6/45/0/45_3/-45_5/45_2]_S$

$[45_5/-45_7/45_2/0/45_3/-45_3/0_2]_S$

$[-45_3/45_4/-45_2/45/0_3/-45_5/45_5]_S$

$[-45_3/45_5/-45_2/0_3/-45_5/45_5]_S$

$[-45_3/45_6/-45_3/90_3/0_8]_S$

$[-45_3/45_7/-45_5/45/0_7]_S$

$[-45_3/45_7/-45_4/0/45_3/-45_3/0_2]_S$

$[-45_3/45_7/-45_4/0_3/90_6]_S$

$[-45_3/45_7/-45_4/0_9]_S$

$[-45_3/45_7/-45_4/90/0/45/-45/0_5]_S$

$[-45_3/45_7/-45_4/90/0_8]_S$

$[-45_3/45_7/-45_4/90_2/0_7]_S$

$[-45_3/45_7/-45_4/90_3/0_6]_S$

$[-45_3/45_8/-45_5/0_3/-45_2/45_2]_S$

$[-45_3/45_8/-45_5/0_7]_S$

$[-45_4/45_8/-45_5/45/0_5]_S$

$[-45_7/45_7/0/90_8]_S$

Figure AII-1 List of 128 unique optima found for benchmark laminate problem in Chapter 5.

Appendix III: Buckling Modes for Laminate with 5 Longitudinal Voids

Figure AIII-1 shows the two other contour plots of the non-critical buckling modes of the laminate with 5 longitudinal voids, as outlined in Chapter 5. The first shows the overall buckling mode, while the second is a local mode, both non-critical.

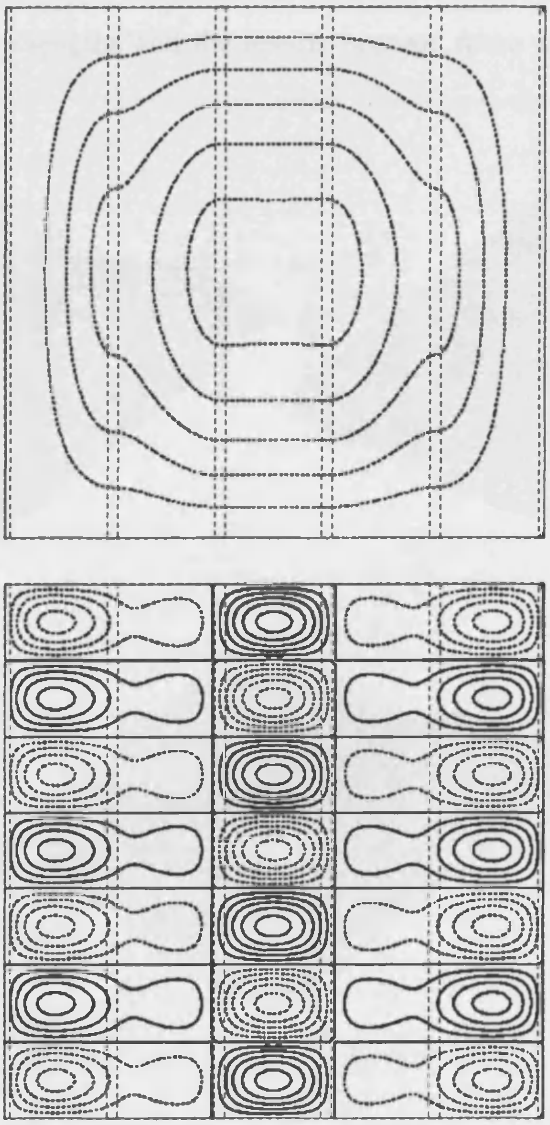

Figure AIII-1 Contour plots of non-critical buckling modes for laminate with 5 longitudinal voids problem in Chapter 5.

APPENDIX IV: ISOMETRIC PLOTS OF J3 PANEL BUCKLING

Figure AIV-1 shows the three isometric plots of the critical buckling modes of the best found solution for the J3 panel at load cases 1 to 3 respectively from top to bottom, as given in Table 6-6. It is obvious that the governing modes are global for the first and last load case. The problem with using the VIPASA analytical model is the skewed nodal lines, which makes the simply-supported transverse ends inaccurate. In the second load case, however, the critical buckling is at shorter wavelengths and the results become more accurate since there is less skew.

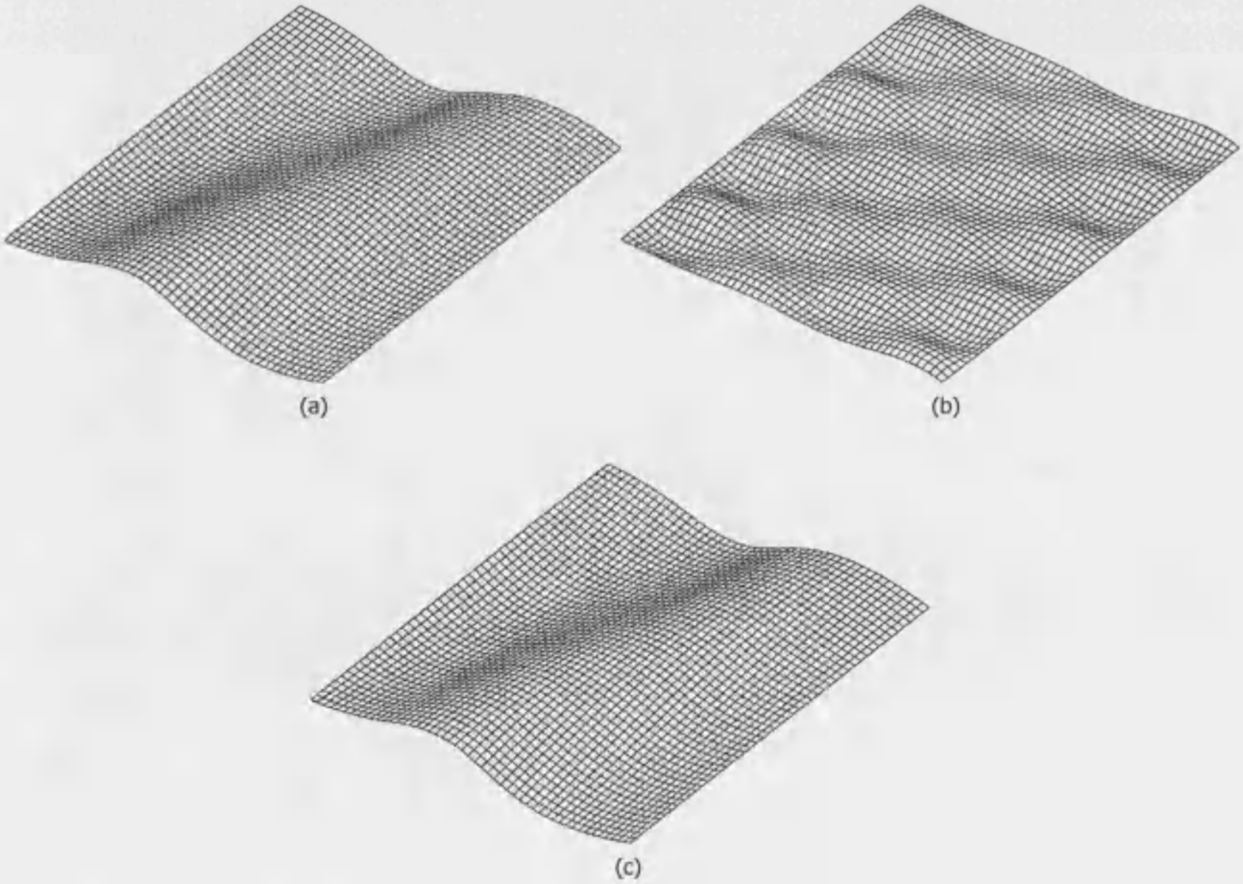

Figure AIV-1 Isometric plots of the critical buckling modes of the best found solution for J3 panel. The plots are for load case (a) 1, (b), 2, and (c) 3 from Table 6-6.