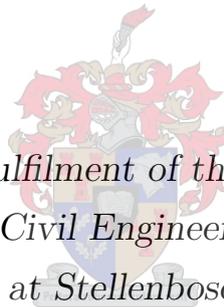


UNIVERSITEIT • STELLENBOSCH • UNIVERSITY

An Approach to Multi-objective Life Cycle Cost Optimization of Wind Turbine Tower Structures

by

Hagen Wolfgang Horsthemke



*Thesis presented in fulfilment of the requirements for the
degree of Master in Civil Engineering in the Faculty of
Engineering at Stellenbosch University*

Department of Civil Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisor: Mr. Etienne van der Klashorst

December 2013

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature:

H.W. Horsthemke

Date:

Copyright © 2013 Stellenbosch University
All rights reserved.

Abstract

An Approach to Multi-objective Life Cycle Cost Optimization of Wind Turbine Tower Structures

H.W. Horsthemke

*Department of Civil Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (Civil Engineering)

December 2013

Support tower structures of Wind Energy Conversion Systems (WECS) are major cost items and by means of integrated design and optimization, the Life-Cycle Cost (LCC) can be reduced substantially. In this thesis, Horizontal Axis Wind Turbine (HAWTs) tower structures are investigated by means of a technique or tool that can benefit in decision making related situations to reduce the LCC of such WECS support towers from inception to disposal.

Often, during the conceptual design phase a certain level of uncertainty or fuzziness exists and plays a role. The central focus in this project is on lattice type towers; however an account on tapered, tubular monopole towers is given as well. The problem is identified to be of a multi-objective nature, where a variety of criteria or objectives that are identified play a role in the possible reduction of the total LCC of the structure. The study also entails the delineation and discussion of the factors and components that affect the LCC of a steel structure. The decision maker has control over only a few of these factors and components as identified, and these can be formulated by means of an objective to be

minimized (or maximized in several other cases). Some of the objectives are incommensurable and others are commensurable with each other. In other words, several of these objectives either ‘compete’ or don’t ‘compete’ against each other, respectively. The investigation resulted in the development of a multi-objective LCC optimization using the λ -formulation (or min-max formulation) as the objective aggregating approach for the four objectives identified (varied during analysis for sensitivity checks). The objectives are user-defined in terms of membership functions that grade the degree of membership from total acceptance to total rejection by means of boundary values. This formulation is Non-Pareto based and the decision maker obtains the best trade-off or best compromise solution. The detailed discussion around these objectives is included in the literature study. The objectives in the multi-objective study are weight, cost, perimeter and nodal deflections, and a weighting of the objectives is possible but this is excluded from this study.

A Genetic Algorithm (GA), coded in MATLAB, is implemented as the optimization tool or technique. The algorithm uses a quadratic penalty function approach and a natively written Finite Element Analysis (FEA) tool is used for the response model in the fitness evaluation process, where the performance for stability, capacity and overall deflections of an individual in the population is quantified. A GA has the advantage that it operates on an entire population of individuals using basic principles such as genetics, crossover, mutation, selection and survival of the fittest from biology and Darwinian principles. GAs are very robust and effective global search methods that can be applied to most fields of study. GAs have previously been effectively applied in structural, single objective optimization (structural weight) problems. The GA is adopted and modified and verified with results on academic problems obtained from literature. Satisfactory performance was observed, although room for improvement is identified.

A case study on a full scale model is performed, using circular hollow sections and equal leg angle sections. These are commonly used steel profiles for lattice type towers. The results obtained are as expected. The structural mass was used as a measure to compare the results. A heavier structure is obtained using the equal leg angle sections compared to the CHS structure with a difference of up to 20% in weight. The best compromise solutions are feasible and near optimal, given the conditions of the equally weighted objectives in this study. The membership function definition and boundary value determination still remains a key issue when using fuzzy logic to incorporate the preference information of the decision maker.

Uittreksel

An Approach to Multi-objective Life Cycle Cost Optimization of Wind Turbine Tower Structures

(“ ’n Benadering tot Multikriteria Lewensiklus Kosteoptimering van Windturbinetorings”)

H.W. Horsthemke

*Departement Siviele Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (Siviele Ingenieurswese)

Desember 2013

Toringstrukture van windturbines is belangrike kostekomponente van ‘n windkragopwekking stelsel. Deur middel van geïntegreerde ontwerp en optimalisering kan die lewensiklus-koste aansienlik verminder word. In hierdie tesis word horisontale-as windturbinetoringstrukture ondersoek. Deur middel van ‘n tegniek of hulpmiddel wat kan baat vind by besluitneming situasies, word die lewensiklus-koste van sodanige windturbine ondersteuning torings vanaf voorgebruik-fase tot lewenseinde-fase verminder.

Dikwels, tydens die konseptuele ontwerp-fase, speel ‘n sekere vlak van onsekerheid of verwarring ook ‘n rol. Die sentrale fokus in hierdie projek is op staal vakwerk tipe torings gelê. ‘n Vereenvoudigde ontleeding van buisvormige torings is ook benader. Die probleem is van multikriteria aard, waar ‘n verskeidenheid van kriterie of doelwitte geïdentifiseer was. Hulle speel ‘n rol in die moontlike vermindering van die totale lewensiklus-koste van die struktuur. Die studie behels ook die bespreking en afbakening van die faktore en komponente wat die lewensiklus-koste van ‘n staal struktuur bepaal. Die besluitnemer het

slegs beheer oor sekere van hierdie faktore en komponente, en hierdie word deur middel van 'n saamgevoegde doel-funksie gedefinieer wat dan geminimeer word. Sommige van die doel-funksies kompeteer met mekaar en sommige kompeteer nie met mekaar nie. Die ondersoek het gelei tot die ontwikkeling van 'n multikriteria lewensiklus-koste optimalisering met behulp van die λ -formulering (of min-max formulering). Hierdie is 'n tegniek wat die kriterie in vorm van 'n verteenwoordigende doel-funksie saamvoeg. Daar is vier doelwitte wat geïdentifiseer was. Die gebruiker definieer spesiale, lineêre doel-funksies wat van totale aanvaarding tot totale verwerping streek. Dit word deur middel van randwaardes gedoen. Hierdie formulering is nie Pareto gebaseer nie, en die besluitnemer verkry die 'best trade-off' of die beste kompromis oplossing. Die gedetailleerde bespreking rondom hierdie doelwitte is in die literatuurstudie ingesluit. Die doelwitte wat in die multikriteria studie gebruik word is gewig, koste, omtrek van die snitprofiel en strukturêle defleksie. 'n Gewig kan aan elke kriterium toegeken word, maar dit word van hierdie studie uitgesluit.

'n Genetiese algoritme (GA), geïmplementeer in MATLAB, word as die optimalisering instrument en tegniek gebruik. Die algoritme gebruik 'n kwadratiese 'straf-funksie' en 'n MATLAB Eindige Element Analise (EEA) word gebruik vir die gedragsmodel in die 'fiksheid' evalueringsproses. Die prestasie vir stabiliteit, kapasiteit en algehele verlegging van 'n individu in die GA bevolking word daardeur gekwantifiseer. 'n GA het die voordeel, dat dit met 'n hele bevolking van individue werk. Dit is gebaseer op beginsels van genetika en Darwin se beginsels. GAs is baie stabiel en ook effektiewe globale soek metodes wat van toepassing in verskillende studierigtings is. GAs is al effektief toegepas in strukturêle optimalisering (veral strukturêle gewig optimalisering). Die GA in hierdie studie was aangepas en die gedrag en prestasie is bevestig met resultate van akademiese probleme uit die literatuur. Bevredigende prestasie is waargeneem, maar ruimte vir verbetering is ook geïdentifiseer.

'n Gevallestudie oor 'n grootskaal model is uitgevoer, en die gebruik van ronde holprofiel en gelykbenige hoekprofiel is uitgevoer. Dit is algemeen gebruikte staalprofiel vir vakwerk tipe torings. Die resultate wat verkry is, is soos verwag. Die strukturêle massa is gebruik as 'n maatstaf om die resultate te vergelyk. 'n Swaarder struktuur is die resultaat wanneer gelykbenige hoekprofiel gebruik word in vergelyking met die ronde holprofiel struktuur. 'n Verskil tot 20% in gewig is waargeneem. Die beste kompromis oplossing is haalbaar en naby-optimaal, gegewe die omstandighede van die gelyk geweegde doel-funksies in hierdie studie. Die doel-funksie definisie, die voorkeur van die besluitnemer en die bepaling van die randwaardes bly steeds 'n belangrike kwessie by die gebruik van hierdie benadering.

Acknowledgements

I would like to express my sincere gratitude to the following people and organizations:

- My study leader, Mr. Etienne van der Klashorst for the long hours of interesting discussions, the valuable insight, support and guidance throughout my postgraduate studies.
- Professor P. E. Dunaiski, whose academic and personal excellence inspired me to pursue my postgraduate studies in structural engineering.
- The CDSS for providing financial support.
- All staff members in the department for their excellent support on all matters and the opportunities granted to develop myself.
- All fellow students and friends in the office for the good times that are to be remembered at Stellenbosch.
- My family and especially my parents who provided me with the educational foundation that allowed me to attain this goal.
- Wolfgang and Angela Schaeffler for proofreading this thesis and also for their support on all matters.
- My girlfriend, Christiane Schaeffler, who encouraged and supported me during all the hours towards the completion of this task.

This thesis is typeset using open source software: the L^AT_EX document markup language for the T_EX typesetting system.

Contents

Declaration	i
Abstract	ii
Uittreksel	iv
Acknowledgements	vi
Contents	vii
List of Figures	xii
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
1.1 Research introduction and focus	1
1.2 Overall aim of the research	2
1.3 Preliminary study	3
1.4 Specific scope and limitations	4
1.5 Methodology overview	5
1.6 Framework of the report	5
2 Literature Review	8
2.1 Introduction	8
2.2 A background to wind energy	8
2.3 Advantages of wind energy	10
2.4 Disadvantages of wind energy	11
2.5 Basic characteristics of wind turbines	15
2.6 An overview of Life-Cycle Cost	19

2.7	Optimization of Wind Energy Conversion Systems	22
2.8	Mathematical optimization methods	30
2.8.1	Introduction	30
2.8.2	Single objective optimization	31
2.8.3	Multi-objective optimization	32
2.8.4	Gradient based optimization	34
2.8.5	Non-gradient based optimization	35
2.8.6	Life-cycle cost optimization using reliability theory	36
2.8.7	Fuzzy discrete multi-objective life-cycle cost optimization using GAs	38
2.8.8	Genetic Algorithms as a tool in LCC optimization	44
2.8.8.1	Introduction	44
2.8.8.2	Encoding	45
2.8.8.3	Crossover	46
2.8.8.4	Mutation	48
2.8.8.5	Selection	49
2.8.8.6	Elitism	51
2.8.8.7	Fitness function	51
2.8.8.8	Population size	52
2.8.8.9	Termination conditions	53
2.8.8.10	Penalty methods	53
2.8.8.11	Test functions	55
2.8.8.12	The search space	56
2.9	Life-cycle cost formulations	57
2.9.1	Introduction	57
2.9.2	Life-cycle cost components	57
2.9.2.1	Initial cost	59
2.9.2.2	Operating cost	67
2.9.2.3	Maintenance cost	68
2.9.2.4	Inspection cost	68
2.9.2.5	Repair cost	68
2.9.2.6	Probable failure cost	68
2.9.2.7	Decommissioning cost	69
2.9.3	Life-cycle cost factors	69
2.9.3.1	Cost of rolled sections	71
2.9.3.2	Number of different section types	72
2.9.3.3	Weight of rolled sections	73
2.9.3.4	Perimeter of rolled sections	74

2.9.3.5	Number of connections	75
2.9.3.6	Geographic location of the project	75
2.9.3.7	Maintenance policy of the structure	76
2.9.3.8	Anticipated life of the structure	77
2.9.3.9	Discount rate of the currency	77
2.9.3.10	Use of the structure	78
2.9.3.11	Importance of the structure	80
2.10	Chapter conclusion	81
3	Research Methodology	83
3.1	Introduction	83
3.2	Analysis methodology	83
3.3	Analysis overview	84
3.4	Limitations of the research	87
4	Monopole Type Tower LCC Optimization	89
4.1	Introduction	89
4.2	Modeling objectives	90
4.3	Modeling variables	90
4.4	Modeling constraints	91
4.5	Monopole tower optimization results	94
4.6	Chapter conclusion	96
5	Lattice Type Tower LCC Optimization	97
5.1	Introduction	97
5.2	Modeling objectives	98
5.2.1	Critical structural deflections	98
5.3	Modeling variables	99
5.4	Modeling constraints	100
5.4.1	Axial compression resistance	100
5.4.2	Axial tension resistance	103
5.5	The FEM Solver	105
5.6	An example using MS Excel	108
5.6.1	Part 1: Fuzzy multi-objective cost optimization with two objectives	109
5.6.2	Part 2: Fuzzy multi-objective cost optimization with four objectives	112
5.7	25-Bar truss tower model	115
5.8	212-Bar truss tower model	123
5.8.1	212-Bar truss tower single objective optimization	128

5.8.2	212-Bar truss tower multi-objective LCC optimization	132
5.9	Result discussion	138
5.10	Chapter conclusion	139
6	Conclusion and Future Perspectives	142
6.1	General conclusion	142
6.2	Monopole type tower LCC optimization conclusion	142
6.3	Lattice type tower LCC optimization conclusion	143
6.4	Future perspectives and recommendations	146
	List of References	147
	Appendices	151
	Appendix A: MATLAB Code	152
	MATLAB code for Chapter 5	152
	GA_Controller.m	152
	solver_FEMtruss.m	156
	GA_Crossover.m	158
	GA_Elitism.m	160
	GA_Mutation.m	160
	GA_SelectionRouletteWheel.m	161
	fitness_Truss.m	161
	create_FitnessGraph.m	164
	create_TrussDiagram.m	165
	displacement_Truss.m	169
	perimeter_Truss.m	170
	weight_Truss.m	170
	getFEMtruss_geometry.m	171
	getFEMtruss_sectionProperties.m	179
	gtrans.m	186
	assem.m	186
	bar3e.m	186
	bar3s.m	187
	coordxtr.m	187
	extract.m	188
	solveq.m	188
	eldisp3.m	189

eldraw3.m	192
pltstyle.m	193
MATLAB code for Chapter 4	194
monopole.m	194
obj.m	198
ineq1.m	198
ineq2.m	200
ineq3.m	200
ineq4.m	201
Appendix B: PWT and Cutting Times Tables	202
Post weld treatments time	202
Plate cutting time for welding	202
Appendix C: Wind Load Calculations on Wind Turbine Towers	204
Wind load calculations	204
Appendix D: FEM Solver verification check	206
PROKON 25-Bar Model Results	206

List of Figures

2.1	World total installed wind power capacity. Year 2010 is forecasted.	10
2.2	Coal fired power station compared to the horizontal axis wind turbine (HAWT).	11
2.3	Simplified representation of wind power extraction through turbine.	12
2.4	Simplified cubic relationship between wind speed and power.	14
2.5	Typical actual power curve plot for a 2 MW HAWT.	14
2.6	Estinnes Wind Farm, Belgium, 2010. One month before completion.	15
2.7	Savonius VAWT and Darrieus VAWT compared to modern HAWT.	15
2.8	Various types of HAWT tower structures	16
2.9	Relationship between tower height and rotor diameter	17
2.10	Relationship between tower height and rated power output	18
2.11	Schematic representation of the external components of a HAWT with a tubular tower	18
2.12	Schematic representation of the external components of a HAWT with a lattice tower	19
2.13	Successive main stages for life-cycle costing	20
2.14	Relationship between reduction opportunities and committed life-cycle cost (UITP, 2002).	21
2.15	Three classes of structural optimization.	23
2.16	Representation of the Pareto set of optimum results.	32
2.17	Classification of gradient and non-gradient based optimization methods with some commonly used examples.	34
2.18	Multi-objective methods classification.	43
2.19	Representation of binary encoding and the comparison to genetics.	47
2.20	Weighted Roulette Wheel for selection example in Table 2.4.	50
2.21	Breakdown of the various cost components (1-7) and factors (a-k)	70
2.22	Membership function for the minimum cost of material	72
2.23	Membership function for the minimum material weight	74
2.24	Membership function for the minimum perimeter	75

2.25	Decision tree diagram for O&M decisions.	77
2.26	Annual average CPI history of South Africa.	78
2.27	Schematic representation of accidental actions vs. time on a structure	79
2.28	Schematic representation of variable actions vs. time on a structure	79
2.29	Schematic representation of the reliability of a structure with passing of time.	81
3.1	The general scheme of an optimization approach to design.	85
4.1	Section view of a monopole tower segment	91
4.2	Graphical optimization result of a bottom monopole tower section for $M_u = 1698.2 \text{ kNm}$, $C_u = 64.5 \text{ kN}$ and $V_u = 79.6 \text{ kN}$	95
4.3	Magnified view of graphical optimization result of a bottom monopole tower section for $M_u = 1698.2 \text{ kNm}$, $C_u = 64.5 \text{ kN}$ and $V_u = 79.6 \text{ kN}$	95
5.1	Illustration of a bar (truss) element in three dimensional configuration	106
5.2	Representation of the loading configuration of (a) a two node tension bar element (Tie) and (b) a two node compression bar element (Strut)	107
5.3	A four-bar space truss structure used to verify optimization results	111
5.4	Fuzzy cost optimization results for a 4-bar truss structure with two objectives	112
5.5	General information and nodal positions	113
5.6	Member description and Finite Element Analysis	113
5.7	Fuzzy membership values and results	114
5.8	Fuzzy cost optimization results for a 4-bar truss structure with multiple objectives	114
5.9	25-bar space truss	116
5.10	Fuzzy discrete optimization flowchart including the applicable m-functions.	119
5.11	25-bar space truss fitness and mass graph	120
5.12	25-bar space truss FEM model	122
5.13	25-bar space truss fittest design	122
5.14	Complete schematic representation of the 212-bar wind turbine lattice tower and system	124
5.15	Schematic representation of the isolated 212-bar wind turbine lattice tower	125
5.16	212-bar lattice tower fitness and mass graph for weight as single objective	128
5.17	212-bar lattice tower fittest design for weight as single objective	130
5.18	212-bar lattice tower fitness and mass graph for multi-objective optimization using circular hollow sections	135
5.19	212-bar lattice tower fittest design for multi-objective optimization using circular hollow sections	135

5.20	212-bar lattice tower fitness and mass graph for multi-objective optimization using equal leg angle sections	137
5.21	212-bar lattice tower fittest design for multi-objective optimization using equal leg angle sections	138
5.22	Comparison of the masses of hollow and open steel sections under compression in relation to the load	141
D.1	Undeformed (left) and deformed (right) 25-Bar tower PROKON model used to verify displacement results from natively developed MATLAB FEM-Solver that is implemented in the multi-objective optimization	206
D.2	25-Bar FEM model from the multi-objective optimization yields same results as PROKON analysis	207

List of Tables

2.1	Electricity generation cost comparison for various types of technologies in 2009.	11
2.2	Typical wind turbine parameters	17
2.3	Section list of the first CHS with parameters	46
2.4	Roulette Wheel Selection example	50
2.5	CESA Basic consulting fees for an example category in 2011	60
5.1	Maximum deflections at serviceability	99
5.2	Model constraints and material parameters	110
5.3	Results of four-bar truss example - Part 1	111
5.4	Results of Four-bar truss example - Part 2	113
5.5	25-bar truss tower nodal coordinates	117
5.6	25-bar truss tower benchmark section list	117
5.7	25-bar truss tower benchmark nodal load values	117
5.8	25-bar truss tower benchmark nodal supports	118
5.9	25-bar truss tower benchmark run results	120
5.10	Sample FEM Output	121
5.11	212-bar truss tower nodal coordinates 1 to 40	126
5.12	212-bar truss tower nodal coordinates 41 to 68	127
5.13	212-bar truss tower nodal supports	127
5.14	212-bar lattice tower run results, using the benchmark problem sections	129
5.15	212-bar truss tower section list for multi-objective investigation using circular hollow sections	133
5.16	212-bar lattice tower multi-objective optimization maximum and minimum objective values	134
5.17	212-bar lattice tower multi-objective optimization run results using circular hollow sections	134
5.18	212-bar truss tower section list for multi-objective investigation using equal leg angle sections	136

5.19	212-bar lattice tower multi-objective optimization run results using equal leg angle sections	137
B.1	Time needed for different PWT techniques.	202
B.2	Cutting time of plates for 1 <i>mm</i> length, T_{CP} (<i>min/mm</i>) in the function of weld size a_w (<i>mm</i>) for longitudinal fillet welds, T-, V- and $\frac{1}{2}$ V-butt welds. . . .	202
B.3	Cutting time of plates for 1 <i>mm</i> length, T_{CP} (<i>min/mm</i>) in the function of weld size a_w (<i>mm</i>) for fillet, longitudinal X- and K-butt welds.	203
D.4	PROKON Nodal Point Displacements at SLS	208
D.5	PROKON Statistical Data of 25-Bar Tower	208

Nomenclature

Acronyms and Abbreviations

AISC	American Institute of Steel Construction
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASD	Allowable Stress Design
CDSS	Center for the Development of Steel Structures
CESA	Consulting Engineers of South Africa
CHS	Circular Hollow Section
CPI	Consumer Price Index
CPIX	Consumer Price Index
DOF	Degree of Freedom
DNA	Deoxyribonucleic acid
DNV/RISØ	Det Norske Veritas Risø National Laboratory
DS	Danish Standard
EA	Evolutionary Algorithms
FEA	Finite Element Analysis
FEM	Finite Element Method
FORM	First Order Reliability Method
GA	Genetic Algorithm
GMAW-M	Gas Metal Arc Welding with Mixed Gases
GUI	Graphical User Interface
HAWT	Horizontal Axis Wind Turbine
HSFG	High Strength Friction Grip
IEC	International Electrotechnical Commission
LCC	Life-Cycle Cost
LCCM	Life-Cycle Cost Management
LRFD	Load Resistance Factor Design
MATLAB	Matrix Laboratory

MS	Microsoft®
O&M	Operation and Maintenance
PSO	Particle Swarm Optimization
PV	Present Value
PWT	Post-welding treatment
RC	Reliability Category
RWS	Roulette Wheel Selection
SABS	South African Bureau of Standards
SAISC	South African Institute of Steel Construction
SANRAL	South African National Road Agency Limited
SANS	South African National Standard
SHM	Structural Health Monitoring
SLS	Serviceability Limit State
SORM	Second Order Reliability Method
TIG	Tungsten Inert Gas method
TCO	Total Cost of Ownership
UIT	Ultrasonic Impact Treatment
ULS	Ultimate Limit State
VAWT	Vertical Axis Wind Turbine
WECS	Wind Energy Conversion System

Latin Letter Variables

A	swept area; cross-sectional area
A	curve fitting constant
A_{ef}	effective cross-sectional area of a member
A_g	gross cross-sectional area of a member
A_{gv}	gross area in shear for block-shear failure
A_{ne}	effective net area of a member
A'_{ne}	reduced effective net area of a member
A_{nt}	net area in tension for block failure
A_{nv}	net area in shear for block-shear failure
A_s	surface area; shear area
\mathbf{a}	global displacement vector
a_{sp}	constant time value per unit area
a_w	leg size of the weld
B	curve fitting constant

b	width of cross-section
C	cost; technology dependent welding parameter; curve fitting constant
c_1, c_2	trust parameters
C_b	material cost of bolts/pins
c_b	unit or item cost of bolt or pin
C_{CP}	cost of cutting
C_d	cost of drilling holes
C_e	erection cost of the structure
C_{el}	critical Euler force
C_f	fabrication cost of structural members
C_F	expected cost of failure
C_f	unit cost of fabrication
C_{HDG}	hot-dip galvanizing cost
c_{HDG}	unit or item cost of hot-dip galvanizing
C_I	initial cost
C_m	material cost of structural members
c_m	material unit volume cost
C_r	factored axial compressive resistance of a strut/structural element
C_T	total cost
C_t	transportation cost of structural members
C_u	axial compressive force under ultimate loads
C_w	warping torsional constant
$\tilde{C}(\tilde{y})$	fuzzy function for material cost
C_x, C_y, C_z	directional cosines
\tilde{D}	fuzzy domain of variables
d	diameter
$d(\mathbf{S})$	decision rule
$\tilde{d}(\tilde{y})$	fuzzy function for displacement
E	energy/kinetic energy; modulus of elasticity/Young's modulus
e	geometrical imperfection; eccentricity; Euler's constant $e = 2.71828\dots$
\mathbf{e}^p	property matrix
\mathbf{F}	system force vector
f	average calculated axial compressive stress
f_{ad}	design stress due to axial force
f_{bd}	design stress due to bending moment
f_{cr}	critical compressive stress

f_e	Euler elastic critical buckling stress
f_{el}	Euler elastic critical buckling stress
f_{ex}	elastic buckling stress about the strong axis
f_{ey}	elastic buckling stress about the weak axis
f_{eyz}	elastic lateral-torsional buckling stress
f_{ez}	elastic torsional buckling stress
f_u	ultimate strength of steel
$f(\mathbf{x})$	objective function in a single objective optimization problem
$\mathbf{f}(\mathbf{x})$	objective function vector in a multi-objective problem
f_y	yield stress
g	gravitational acceleration constant, $g = 9.81 \text{ m/s}^2$
G	shear Modulus
$g(\mathbf{x})$	inequality constraint in an optimization formulation
g_j^{lower}	lower boundary constraint of constraint function
g_j^{upper}	upper boundary constraint of constraint function
H	building height; storey height; tower height
h	height of cross-section
$h(\mathbf{x})$	equality constraint in an optimization formulation
I	moment of inertia
i	iteration counter number; discount rate of the currency
J	St. Venant torsional constant
\mathbf{K}	global Stiffness Matrix
\mathbf{K}_e	element stiffness matrix
k	buckling coefficient for compressive elements
KL	effective length
l	subscript for lower boundary value; counter
L_c	cutting length
L_i	length of the structural element
L_w	welding length
m	mass
M_u	applied moment under ultimate loads
\mathbf{N}	element force vector
N	axial (normal) force; counter
n	exponent in fabrication time equation; counter
n	exponent
n_f	number of times a section has been assigned earlier

n_s	number of spot welds
O	element stress vector
P	power; point load; force; counter
p^i	fittest solution found by particle i
P_F	probability of failure
p_i^q	fittest solution found at iteration point q
P_{rated}	rated power of wind turbine
$\tilde{P}(\tilde{y})$	fuzzy function for total section perimeter length
Q	system reaction vector
Q	counter
q	iteration counter
R	shell structure radius
r	rotor radius; radius of gyration; random number
r_1, r_2	randomly generated numbers
r_x, r_y	radii of gyration about principal x - and y - axis respectively
S	a search vector
S	total sum of population; random outcome of a decision
s	running sum of population
S_p	fuzzy set of discrete standard shapes
t	time; thickness
T	production time
T_{CG}	total cutting and grinding time
T_{CP}	time of plate cutting and edge grinding
t_f	flange thickness
T_r	factored axial tensile resistance of a tie/structural element
T_s	time of welding per spot weld
T_{sp}	surface preparation time
T_w	real welding time
t_w	web thickness
$\tilde{T}(\tilde{y})$	fuzzy function for the number of different section types
u	subscript for upper boundary value
v	wind velocity; velocity
V	volume of structural member
v_1	cut-in wind speed
v_2	rated power wind speed
v_3	cut-out wind speed

V_r	factored shear resistance of a structural member/element
V_u	shear force under ultimate loads
w_i	objective function weight
w	inertia of a particle in optimization
W	flat width-to-thickness ratio
W	total weight
W_{lim}	limiting width-to-thickness ratio for fully effective compression elements
$\tilde{W}(\tilde{y})$	fuzzy function for the total weight
\mathbf{X}	random variable
X	global X - axis label
\mathbf{x}	design variable vector in a multi-objective problem
\mathbf{x}^*	optimal design variable vector
x^0	initial/starting design variable
x_0, y_0	principal coordinates of the shear center from centroid
x_i	design variable in a single objective problem
x_l^{lower}	lower boundary constraint on design variable
x_l^{upper}	upper boundary constraint on design variable
x_q^i	position of design solution at iteration point q
Y	fuzzy set of variables in Z ; global Y - axis label
\tilde{y}	fuzzy variable
y_n	time period
Z	arbitrary set of variables; global Z - axis label
z	arbitrary variable in fuzzy set Z ; local z -axis label; decision variable

Greek Letter Variables

α	step size in the direction of an incremental search; scaling factor
α^*	optimum step size in the direction of an incremental search
β	reliability level; shape (element) penalty factor
β_t	target reliability level
γ	partial factor; exponential function for choosing a candidate shape
Δe	additional increment
δ	nodal deflection; difference
ε	reduction factor
ε_a	reduction factor
ε_b	reduction factor
Θ	shell forming difficulty factor; angle

Θ_{dc}	difficulty factor for material cutting
Θ_{ds}	difficulty factor for surface preparation
Θ_{dw}	welding difficulty factor
κ	number of structural elements to be assembled
λ	non-dimensional slenderness ratio for structural compression elements
λ	overall satisfaction parameter
λ	Lagrange multiplier
λ_a	relative slenderness ratio for local shell buckling
λ_r	relative slenderness ratio for global stability
μ_C	fuzzy membership function for the total section cost
$\mu_{\tilde{D}}$	membership function in a fuzzy domain \tilde{D}
μ_d	fuzzy membership function for the representative deflection
μ_P	fuzzy membership function for the total section perimeter
μ_T	fuzzy membership function for the number of section types
μ_W	fuzzy membership function for the total section weight
μ_i	shell segment forming or production exponent
μ_P	membership function
$\mu_y(z)$	membership function that grades of membership of z in Y
∇	partial derivative operator
ν	Poisson's ratio
ρ	air density; material density
σ	normal stress
Φ	normal distribution function
ϕ	angle between two members; material factor
ϕ	cumulative normal distribution function

Chapter 1

Introduction

1.1 Research introduction and focus

Support tower structures are major cost items for any type of Wind Energy Conversion System (WECS). The optimization of such components through integrated design of the entire system is a powerful way of reducing the LCC and therefore increasing the appeal to investors and government utilities while maintaining a particular, predefined level of reliability and robustness. This research aims at downwind configured Horizontal Axis Wind Turbine (HAWT) support towers. For downwind HAWTs, the only difference is that the rotor is placed on the leeward side of the tower and less costly structural components are practical as these can therefore have a lower stiffness and cost; also larger deflections are tolerable.

Recent research work has mainly focussed on optimum power output economics and levelized cost of energy optimization. Little literature is available that deals with the problem of structural design with regard to LCC optimization. Therefore, this field requires further investigation. Life cycle costing is a methodology for calculating the whole cost of a system from inception to disposal. Whatever the system, a LCC analysis technique will be similar in most cases; the major items of cost can be defined through the life time of the system. The major items may be further subdivided and sub-categorized until the cost of each element can be defined as a good estimate by a mathematical equation (the ideal case). This is not always possible and often uncertainty of possibility or fuzzyness plays a role and techniques of fuzzy logic can be implemented. The elements of cost will then be added together to yield the total discounted or present value (PV) cost for each item and thus a LCC for the system through its full life. Optimization techniques for determining an optimal solution that is dependent on a fixed set of variables will be considered and

implemented in a case study as described in this thesis.

1.2 Overall aim of the research

The overall aim of this thesis is to establish design guidance for wind turbine support towers, directly or indirectly, in order to reduce (optimize) costs associated with future support tower structures for the South African industry. Below is an extract from the recent Center for the Development of Steel Structures (CDSS) Steering Committee Report by Retief (2012). The intent is to link this research project with the main idea of this extract.

“[...] to study the potential for reducing the Life Cycle Cost (LCC) of steel structures. A current research will attempt to optimize the weight of truss [lattice] structures. This is a well-established field of study and it will be used as basis for future work on the reliability based optimization of similar structures.”

According to Adeli and Sarma (2006), research on cost optimization of large structures can encourage the use of the optimization approach in the structural steel design practice for two reasons: Firstly, it provides a more realistic way of modeling structural steel design; and secondly, from previous research, when considering both the minimum weight design and the minimum cost design (material and connection cost) the consensus is that cost optimization can result in additional cost savings in order of 7% to 26% compared to weight optimization only.

As outlined in the beginning of the following chapter, the wind energy sector is very likely to expand significantly worldwide and also here in South Africa. Support structures contribute a significant proportion to the total cost and LCC for a WECS. Reduction of costs (especially in terms of LCC) of WECS support tower structures make wind energy a more attractive investment that is sustainable and economically feasible. In this thesis, the approach to multi-objective LCC optimization related to wind turbine support structures may be reduced to the following aspects and goals, given in brief form:

- Assessing the available options for structural optimization in the design phase that influence the LCC.

- Adapt and implement current, available design guidelines in order to find a simplified way to optimize designs for LCC optimization using computer tools and algorithms, by putting emphasis on practicality, economy and computational effectiveness.
- Investigate possible support structures to be suitable for South African conditions.

1.3 Preliminary study

A WECS consists of various components and many factors influence optimal solutions. Broadly said, optimization concerns finding the best solution from a set of feasible solutions subject to predefined constraints. Many optimization techniques and methods can be found in the literature which have been used in previous research. Optimization methods, or rather mathematical optimization methods are a powerful addition to enhance structural design and research needs to underline and substantiate this. A short discussion of several optimization aspects of WECS components is given below:

1. Turbine blade optimization: Various research has been conducted in this field and a large amount of literature is available.
2. Generator optimization: These are already near optimal (depends on technology though).
3. Wind farm layout optimization: Mostly applied to offshore types only and integer programming is applied.
4. Topography optimization.
5. Multiple-sites optimization.
6. Transmission line and system optimization.
7. Storage (battery systems) optimization.
8. Structural optimization using Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Fuzzy Logic and many other mathematical optimization methods have been applied previously. Generally, the minimum weight of a structure was considered as the objective function. It is claimed, that this does not necessarily coincide with the optimum (minimum) LCC and motivates further investigation and research.

Modeling and optimization in general, including the assessment thereof is quite complex and requires expertise knowledge. There is room for an expertise pool and models and the tracking of evolving technologies and methods.

1.4 Specific scope and limitations

This section provides the reader with initial information on the specific scope and limitations of the research. The focus and specific scope of this study entails the investigation and exploration of a multi-objective fuzzy logic based mathematical optimization approach of on-shore lattice type wind turbine towers by means of a genetic algorithm (GA). The result is to arrive at and identify the region of the global optimum, given the multi-objective nature of the problem.

The problem is of multi-objective nature, thus a variety of objectives are defined and ‘fuzzified’, which means that these are graded in terms of a membership function from total acceptance to total rejection. The objectives are therefore normalized and this eludes the user or designer to consider and study actual, real-world cost values in the optimization. Actual cost are generally very volatile, unpredictable and not constant over time. In this report, a benchmark study and several verification checks are presented to validate and support the functionality and applicability of the adapted algorithm. The drawback with the fuzzy multi-objective optimization in this study entails the exclusion of relative weighting or importance factors for the objectives. Prevailing economic and local conditions affect these considerably and in this study equal weights are assumed by default. The author also limited the LCC optimization study to the presentation of the fittest design result given the rather simplified preference input information. A cost value analysis and assessment, however, is not included.

Additional limitations and simplifications concern the type of analysis used for the search: A static, linear FEM model is employed in the fitness evaluation process of the individuals in the population over the iterations of the GA. Only axially loaded truss elements are considered and the connections are assumed to have at least the same capacity away from the corresponding connection of the members.

A further limitation concerns the loading during the multi-objective optimization of the full scale case study. Simplified, arbitrary nodal point loads are assumed for the study of the full scale case study. The focus of this work is on the multi-objective optimization approach discussed in the subsequent sections of this work and can be used to evaluate

different aspects of LCC due to the fact that it enables one to find trade-off solutions in terms of initial construction cost and eventual maintenance cost. It is clear that the models for calculating each of these costs may be refined extensively but the approach in this thesis shows the methodology using simplified assumptions. The author does not include the actual cost data of the fittest designs obtained from the optimization runs respectively. It is deemed sufficient to present the best compromise solution according to the λ -formulation obtained from the search. The mass of fittest designs obtained as the best compromise solution in the the multi-objective optimization is used as a means of measure to determine the relative performance of the fittest designs and also to compare the results.

1.5 Methodology overview

The methodology followed in this research is presented in chapter 3 in more detail. This section includes a brief and broad overview of the methodology of the contribution in this study:

- To present general methodologies and approaches that are followed when considering LCC optimization and LCC assessment of steel structures with regard to all the life-cycle phases of the structure. Emphasis is put on the discussion of factors and components that carry the potential and capacity for LCC reduction.
- The investigation and expansion of a search technique to solve a well-defined multi-objective optimization problem.
- Apply design constraints and design guidelines to solve the optimization problem.
- To test, validate and compare the results obtained from the optimization in order to draw conclusions and make recommendations with specific regard to lattice type wind turbine tower steel structures.
- Perform a simplified optimization for a monopole tubular tower segment.

1.6 Framework of the report

In chapter 2, the study commences with a brief background on wind energy in general. Thereafter, a discussion on LCC in general follows and a description on some significant concepts around it. An extensive literature study is presented to give the reader the

contextual information on optimization methods and the associated theory that is required to comprehend the problem better. The first steps concern the delineation and discussion of the options, components and factors that affect the LCC of a steel structure over the useful life-cycle. The focus is then shifted to the search technique and optimization methodology in general and the associated formulations. Thereafter, a discussion and description of the multi-objective approach is presented together with the functionality of GAs. Thereafter, a detailed discussion of the fabrication and manufacturing cost follows, but it is emphasized that the focus remains with the discrete multi-objective optimization due to the identified nature of the problem.

In chapter 3, the methodology followed in the subsequent chapters and the specific limitations are outlined and discussed in more detail.

Chapter 4 includes a simplified, two variable graphical cost optimization approach on a tubular, prismatic monopole tower to illustrate the development and formulation of a search space and the visualization of the typical objective functions and constraints.

The central theme of this study concerns the discrete multi-objective optimization and certain limitations, simplifications and assumptions are made. Chapter 5 commences with the multi-objective handling approach (or objective aggregation approach) together with the constraints for axially loaded members. This is illustrated using a 4-bar space truss structure with a single design variable using MS Excel. This makes the visualization and representation of the methodology that is to be extended, easier. Additionally, a single objective (mass) optimization benchmark problem is used to validate and test the algorithm used. The first benchmark problem deals with a 25-bar truss tower structure with eight design variables and is treated as a single objective optimization to compare the benchmark results.

After satisfactory performance and results the objective aggregating approach was expanded to incorporate multiple objectives and a full-scale 212-bar truss tower model is investigated. Initial runs were treated as a single objective optimization using the same section list as in the 25-bar benchmark problem together with the constraints, variables and objectives as described in the respective section. Further, the optimization was expanded to include real-world steel sections from the *South African Steel Construction Handbook* by SAISC. Two case study models are compared considering the same conditions for a section list of circular hollow sections and equal leg angle sections. The constraints, variables and objectives are as described in the corresponding subsections.

Finally, a result comparison and discussion follows together with certain conclusions in chapter 6. The conclusion remains slightly open and highlights the potential and future developments towards LCC optimization using a similar approach. Substantial modifications to the described approach are recommended and need to be identified. Examples include the formulation of different constraint handling approaches and the incorporation of relative objective importance or weighting.

Chapter 2

Literature Review

2.1 Introduction

This chapter covers a comprehensive and extensive literature review around the central theme of Life-Cycle Cost (LCC) optimization of small, medium and standard utility-scale wind turbine support towers. According to Sarma and Adeli (2002), LCC is the total cost of a structure throughout its intentional service period. The lifetime can be different for a variety of structures, also depending on the primary use. This total cost generally involves initial costs which also include preliminary cost for design, planning and construction. Additional to this, cost of operation, maintenance, repair, and eventually decommissioning or dismantling costs of the structure are included. The most economical or optimal design is therefore a function of all of the above aspects. A structure with low initial cost but high operation and maintenance cost is therefore not necessarily the most economical one, as the low initial costs can be deceptive. The principal goal of LCC optimization entails the process of minimizing a total LCC function in the ideal case.

First, a brief background on wind energy is provided and thereafter on life-cycle costing and optimization. In this chapter, structural LCC optimization techniques are discussed comprehensively. Furthermore, the LCC case study model is outlined that includes the proposed optimization algorithm.

2.2 A background to wind energy

Wind energy has been used for thousands of years for various applications that include the extraction of water from underground wells, grinding grain, wind powered ships and numerous other uses. All these examples exemplify the extraction of power from wind.

A recent headline from an online source gives a good initial idea of where wind power, especially in South Africa, might be heading:

“Wind is now the cheapest form of renewable electricity generation, with an average price of 89c a kilowatt hour compared to 97c per kWh for Eskom’s new coal-fired power stations”. This was said by Roger Price, chief executive officer of Windlab, an international wind energy company that is investing in wind energy in South Africa. “The costs are unlikely to go up because, unlike coal, there are no input costs as wind is free”, Price said.

(Source: <http://www.windlab.com>[October 2012])

Over the course of the past few decades, together with the understanding of mechanics and machines, the use of wind energy has evolved considerably. When it comes to Wind Energy Conversion Systems (WECS), the generation of electricity has become the most widely used application (Jain, 2010).

Due to the global oil crisis in the 1970s, an increased interest in wind energy emerged. However, already in the 1980s a decreasing trend in alternative energy resources took place. Investments continued in Europe and up to now, Europe (especially with Denmark as the forerunner) is the leader in terms of technological advances in the wind energy industry. Wind turbines are under development that have a rated power output of up to 10 MW, enough to supply electrical power to 3000 modern, middle-income households (Jain, 2010).

Two main categories of wind turbines are available: Turbines that have rotors spinning around their horizontal axis are known as Horizontal Axis Wind Turbines (HAWT) and those that spin around their vertical axis are termed Vertical Axis Wind Turbines (VAWT).

The *World Wind Energy Report 2009* by The World Wind Energy Association (WWEA, 2009) presented some inspiring data in terms of recent, global trends in this field. Figure 2.1 illustrates the total installed capacity of wind power available worldwide. By inspection, one can see that the total installed capacity follows a non-linear growth pattern during the last ten years.

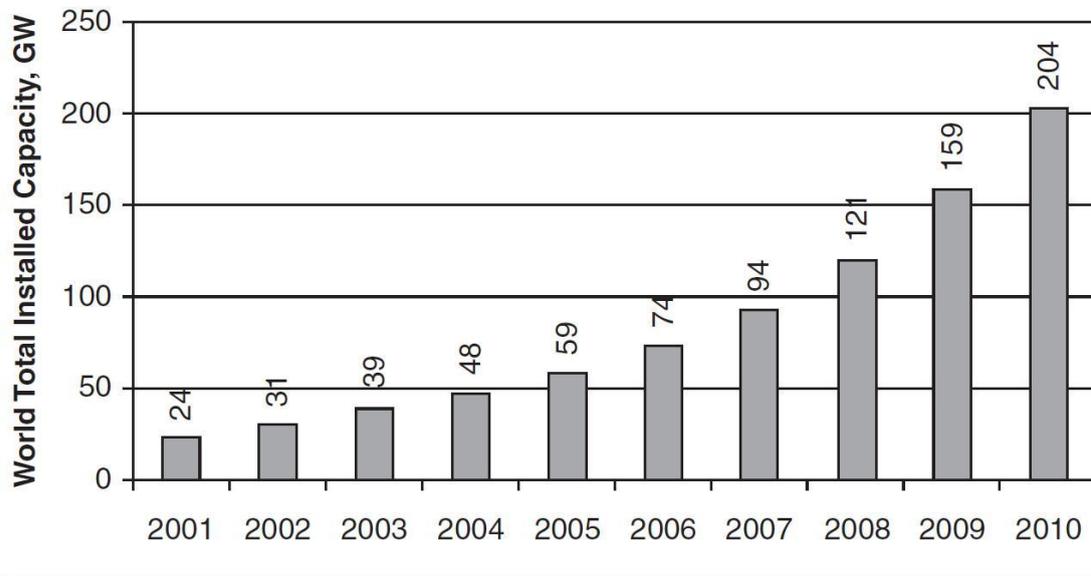


Figure 2.1: World total installed wind power capacity. Year 2010 is forecasted. (Jain, 2010)

The results presented in Figure 2.1 provide information on a global basis, while the aim of this research is to focus on South African conditions and circumstances. The next section discusses a broad outline on the advantages and disadvantages of wind energy.

2.3 Advantages of wind energy

The benefits of wind energy are primarily related to environmental factors, sustainability and possibly lower long term cost due to zero fuel requirements (Jain, 2010). Wind energy production causes no emissions at all. Figure 2.2 compares two conventional methods used for power generation. Both are examples from South Africa. On the left is the Arnot Coal Power Station in Middleburg, Mpumalanga, while on the right are the HAWTs at Klipheuwel, Western Cape. The answer to the question regarding which is the more sustainable and environmental friendly option, is clear.

The fact that wind energy is renewable and readily available makes it a very sustainable option and it should therefore be considered as an alternative option before the other, conventional types of power generation are envisaged. However, not all locations are suitable for effective wind energy generation. Another advantage is that capital cost, operation and maintenance cost for onshore wind projects are comparable to coal fired projects, due to the fact that no fuel costs are involved (Jain, 2010). Table 2.1 compares the respective costs for various categories of different technologies for electricity generation



Figure 2.2: Coal fired power station compared to the horizontal axis wind turbine (HAWT).

during 2009.

Table 2.1: Electricity generation cost comparison for various types of technologies in 2009 (Jain, 2010).

Technology	Installed Cost (€/kW)	O&M Cost (€/kW)	Total Cost (€/kW)	Fuel Price (€/MWh)
Gas-fired	635-875	19-30	655-905	27
Coal-fired	1300-2325	30-60	1330-2390	18
Nuclear	1950-3400	80-96	2030-3500	3.6-5.5
Onshore wind	1300-1500	33-50	1335-1550	n/a
Offshore wind	3000	70	3070-3100	n/a

Additionally, wind energy is among the cheapest sources of renewable energy and occupies relatively little space, as mentioned. Other examples of renewable energy sources are wave and tidal energy, photovoltaic solar energy and geothermal energy. Furthermore, farmers and landowners can make use of wind energy facilities to generate an additional passive income, either by selling electricity to the local or national grid or by land lease payments, while still occupying the surrounding land effectively.

2.4 Disadvantages of wind energy

Despite the advantages, wind energy is not a cure-all. The core disadvantages of wind energy are resource variability, high initial investment costs and environmental impacts (Jain, 2010). It is also to a large extent a question of taste, opinion and belief how the public perceives wind turbines. Plenty of studies have been made, including surveys

and questionnaires. The general trend according to The Danish Wind Energy Association (DWIA, 2002) is that people living near existing wind turbine farms or single turbines are generally more in favor of the wind power alternative than people living in cities.

Whether wind energy can be effectively converted into electricity depends highly on the availability and variability of the wind conditions. As a rule of thumb, wind energy is only economically viable if annual average wind speeds above 6.5–7.0 m/s prevail (Jain, 2010). However, most coastal areas have this characteristic. The variability in wind climate conditions makes this source of energy fairly unreliable. When no wind blows, other sources of energy need to be in place. Below is a simple mathematical deduction that explains the importance concerning the interrelationship between power output and wind velocity. For ideal conditions, the mass of air from which the energy is extracted is cylindrical, and thus energy (or kinetic energy) is then defined by equation 2.4.1. Note that the cross-sectional area A of the cylinder representing the wind turbine rotor is referred to as the swept area of the rotor.

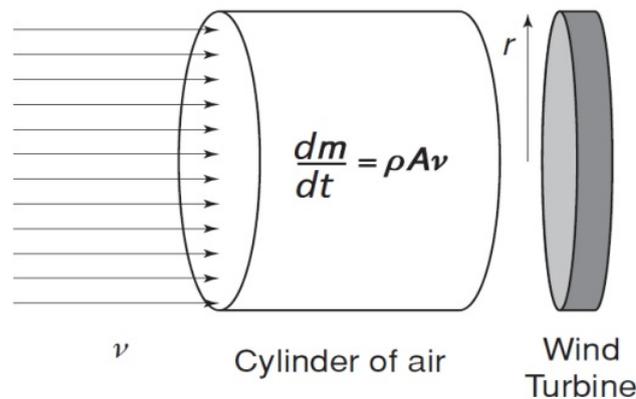


Figure 2.3: Simplified representation of wind power extraction through turbine.

$$E = \frac{1}{2}mv^2 \quad (2.4.1)$$

$$P = \frac{dE}{dt} \quad (2.4.2)$$

$$P = \frac{1}{2} \frac{dm}{dt} v^2, \text{ thus } P = \frac{1}{2} \rho A v^3 \quad (2.4.3)$$

Substituting for area A in equation 2.4.3 reduces to:

$$P = \frac{1}{2}\rho\pi r^2 v^3 \quad (2.4.4)$$

In equations 2.4.1 to 2.4.4, E is the kinetic energy of the moving air mass m , v is the wind velocity, ρ is the air density and P is the ideal power extracted from the moving mass of air. r is the radius of the rotor, as illustrated in Figure 2.3. From equation 2.4.4 it can be observed that the power-velocity relationship is a cubic function.

Figure 2.4 shows a typical, simplified and idealized power-velocity relationship (for a radius of unity, $r = 1$ and air density of 1.225 kg/m^3). The purpose here is to illustrate the importance of high wind speeds that are required for cost-effective power generation and return on investment. For example: by doubling the annual average wind speed, the increase in power extraction is eight-fold. The graph in Figure 2.4 is only a simplified representation, and all of the energy from the wind passing through the wind turbine can never be absorbed. In the case of real fluid flow, some energy is lost in overcoming viscous drag, vortices and wake turbulence near and on the blade surfaces. These losses resemble the effect that determines the aerodynamic efficiency. Betz's law states, that the maximum possible value of aerodynamic efficiency that can be achieved is $\frac{16}{27} = 0.5926$ or 59.26% of the power curve, as shown in Figure 2.4, (Jain, 2010).

However, the actual power curve for a typical wind turbine is different to the ideal case and also the reduced curve due to Betz's law. A certain minimum wind velocity is necessary to initiate the rotational movement of the rotor and therefore the power absorbed from the wind starts at that minimum velocity. The cubic portion of the curve shown in Figure 2.5 is slightly shifted to the right and power production starts around 3 m/s approximately.

At the operating speed, which should be close to the optimal (most efficient) speed of the generator, the curve's gradient starts to decrease and eventually forms a plateau where the constant, rated power output value is reached (in Figure 2.5 this corresponds to around $v_2 = 15 \text{ m/s}$). At around $v_3 = 25 \text{ m/s}$, the cut-out wind speed is reached and power generation is terminated. Various mechanisms can be implemented to achieve the shutdown of the turbine. The curve's shape may also look slightly different depending on whether the WECS is stall regulated or pitch controlled. Further variations may also depend on variable speed and fixed speed generators.

Another disadvantage of wind power is that the initial investment costs are high if compared to other conventional methods. Usually, wind farms are located far from a popula-

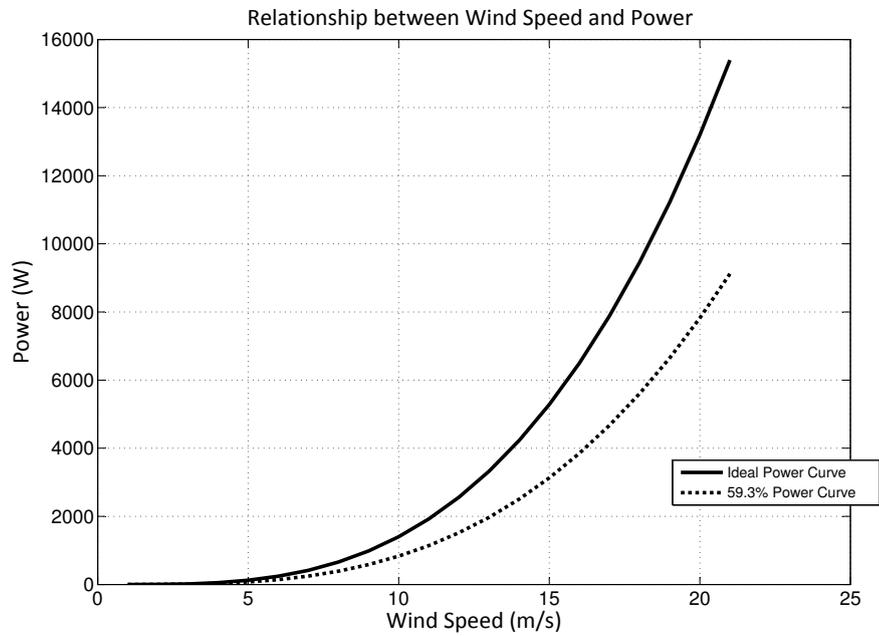


Figure 2.4: Simplified cubic relationship between wind speed and power.

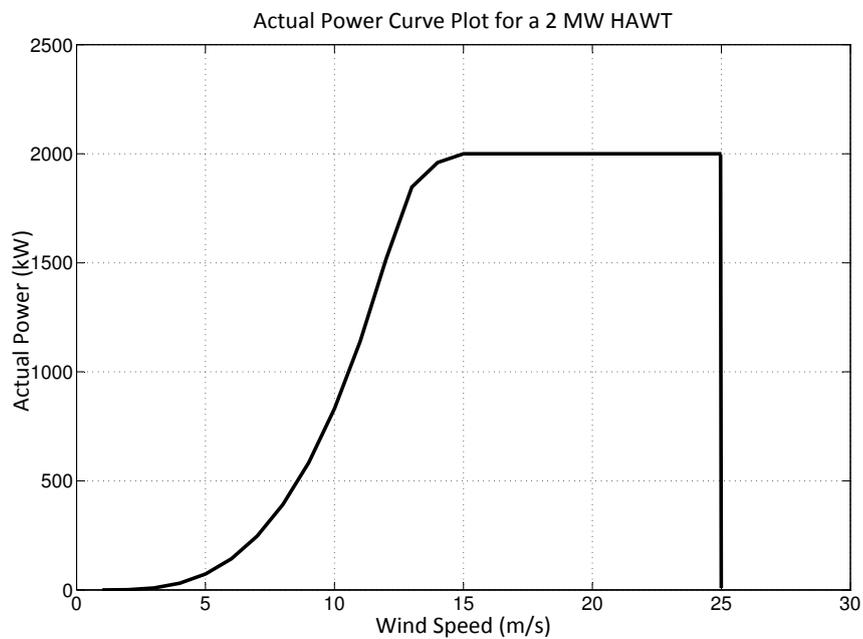


Figure 2.5: Typical actual power curve plot for a 2 MW HAWT.

tion center and high capacity, large and expensive transmission lines need to be in place for supply (Jain, 2010).

The environmental matters that have been discovered include the danger to wildlife like birds and bats. However, the aesthetics are of much higher concern as the scenery is disturbed by wind turbine structures. This is often referred to as scenic pollution. Figure 2.6 demonstrates an example of scenic pollution of the Estinnes Wind Farm in Belgium, while still under construction. Lattice type support towers are sometimes referred to as less attractive due to aesthetics and wider base requirements.



Figure 2.6: Estinnes Wind Farm, Belgium, 2010. One month before completion.

2.5 Basic characteristics of wind turbines

The two major commercial wind turbine categories that have been identified and defined thus far are the HAWT and VAWT. The fundamental difference between these two is the orientation of the axis about which the rotor spins. Figure 2.7 illustrates the difference between a HAWT and two different VAWTs. This research covers the HAWT support

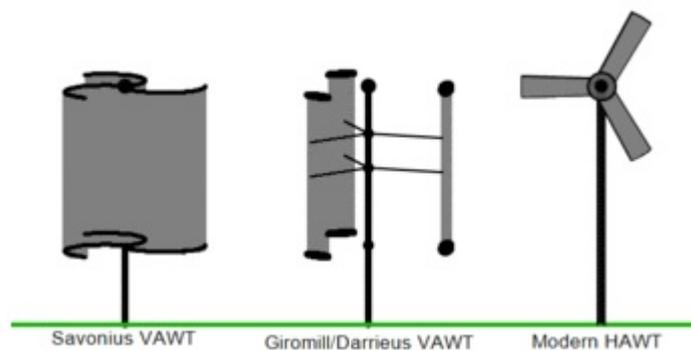


Figure 2.7: Savonius VAWT and Darrieus VAWT compared to modern HAWT.

towers for standard utility-scale systems only, as HAWTs constitute the most common type of wind turbines available on the market. Propeller-type, three bladed rotors dominate the range of commercial wind turbines nowadays according to DNV/RISØ (2002).

This rotor type is also referred to as the Danish Concept rotor. For this type of wind turbine, three main categories of support towers have been identified. These are the tubular type, guy-wired tubular type and the lattice type. The preferred use of tubular steel towers over lattice towers is generally justified due to reduced visual and landscape impacts (Jain, 2010). Figure 2.8 is an illustration of the various support towers compared next to each other. According to Uys *et al.* (2006), the most suitable load-carrying structure for a wind turbine is a welded steel shell tower. It can be constructed as a tower composed of stacked cylindrical and non-prismatic shell segments. However, one might question its suitability for South African circumstances with regard to technical expertise required for manufacturing tubular towers, other manufacturing constraints and logistics. It is interesting to see how a lattice tower performs in this regard, which is also one objective of this research project. The 2010 FIFA World Cup with the erection of expensive stadia and also the recent construction of giant coal power station components (exhaust gas flumes) however show that South Africa can develop the capacity to facilitate the construction and erection of such structures. From left to right, the various tower types are:



Figure 2.8: Various types of HAWT tower structures

Tubular steel tower, tubular concrete tower, lattice tower, three-legged tower, guy-wired pole tower.

Table 2.2 below compares different parameters of typical wind turbines with each other. These parameters are the tower height, rotor diameter and the rated power output.

Table 2.2: Typical wind turbine parameters (DNV/RISØ, 2002)

Tower Height (m)	Rotor Diameter (m)	Rated Power (kW)
22	21	55
31	30	225
35	35	450
37	41	500
44	43	600
50	48	750
50	54	1000
60	58	1500
67	74	2000
85	115	5000

Figures 2.9 and 2.10 show the data from Table 2.2 graphically. Figure 2.9 relates the tower height of various HAWT to their rotor diameters. Figure 2.10 relates the tower height to the power output of various HAWT.

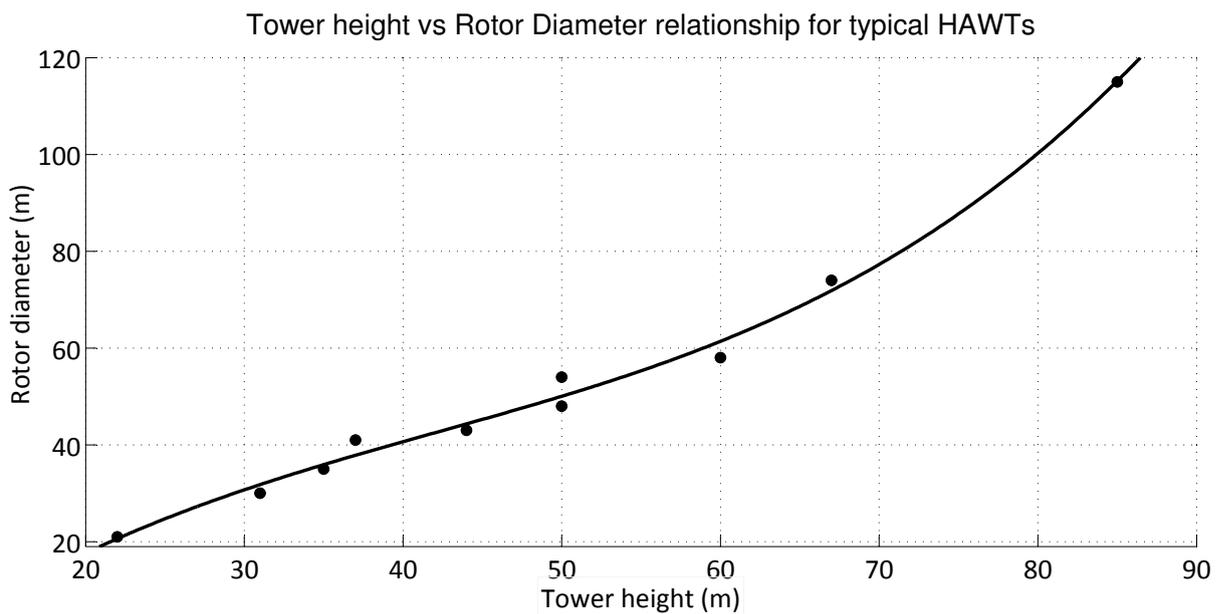
**Figure 2.9:** Relationship between tower height and rotor diameter

Figure 2.11 below is an illustration of the external components of a HAWT. In this report, reference will be made to the various components as depicted.

A short description of the different external components is given below:

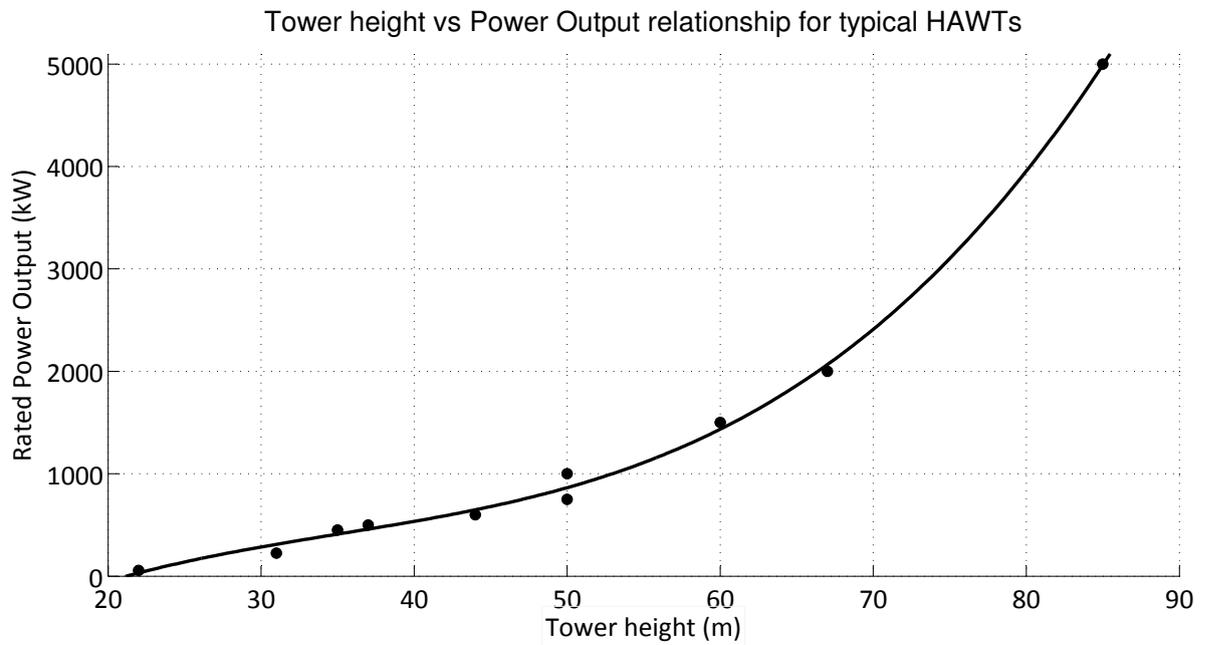


Figure 2.10: Relationship between tower height and rated power output

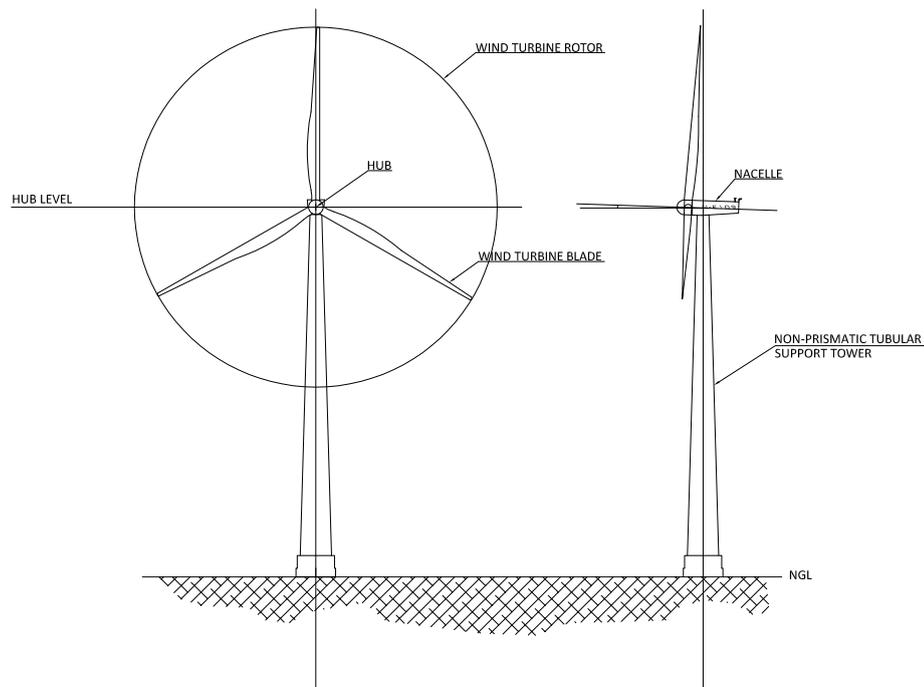


Figure 2.11: Schematic representation of the external components of a HAWT with a tubular tower

- Rotor: The rotor consists of the hub and the blades of the wind turbine system. In the case of a pitch-controlled wind turbine, the pitch adjusting mechanism is

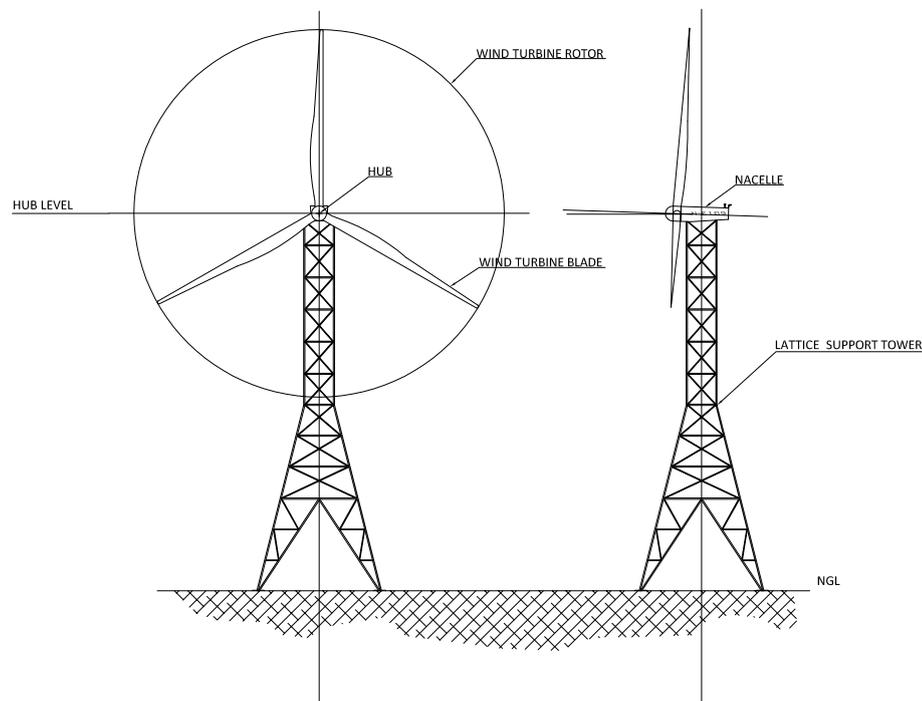


Figure 2.12: Schematic representation of the external components of a HAWT with a lattice tower

included in the hub. The rotor drives the generator located inside the nacelle.

- **Nacelle:** The nacelle is the housing of the various mechanical and electrical components and contains most of the control systems and also the generator. Apart from the generator the nacelle might also house the following: a reduction gearbox, a cooling system, the driveshaft and access facilities for maintenance staff.
- **Tower:** The tower or support tower, as mentioned, supports the wind turbine assembly (rotor and nacelle). It allows for the increase in height to reach improved wind conditions for more optimal, effective and efficient power generation.
- **Foundation:** The foundation supports the structure/system in its entirety and forms the interface between the support tower and the soil and ensures effective transfer of the loads from the structure to the soil.

2.6 An overview of Life-Cycle Cost

In engineering, models are used to represent real world situations. Such engineering models are, generally speaking, a simplification with the intent of being the best possible approximation. Modeling is often performed with the aid of computers during analysis or optimization (Pahl, 2012).

In an informative paper on LCC optimization by the International Association of Public Transport (UITP, 2002) a worthy account on a study of the general concept of LCC optimization is provided. One objective of the study entails Life-Cycle Cost Management (LCCM).

LCCM, in general, entails two main concepts: The life-cycle of the installation (or structure in this context) and the total cost of ownership (TCO). Furthermore, LCC is the total cost of the installation over its useful life and includes the total acquisition cost, development and design costs, operational (maintenance and repair) costs and decommissioning costs. Figure 2.13 gives an overview of the main stages that contribute to the total LCC of a system, or more particularly, a physical asset.

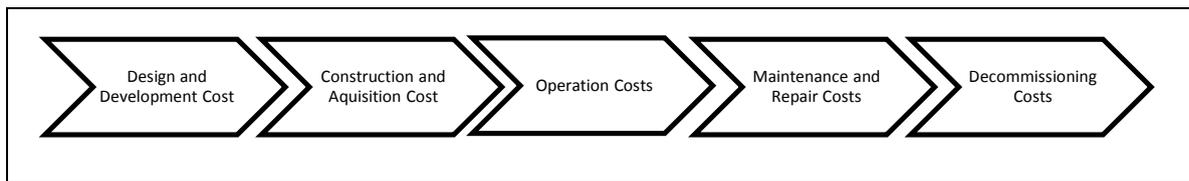


Figure 2.13: Successive main stages for life-cycle costing

A key objective of any asset, project or construction should be an efficient life-cycle cost management approach. LCCM is a set of tasks and processes that focus on reducing the cost components at all stages during the life of an asset.

In the publication of UITP (2002), a field of interest that is covered considers the total life-cycle cost evolution. A representation regarding the relationship between committed cost and life-cycle cost is provided for the life-cycle stages in Figure 2.14. The relationship between committed and life-cycle cost can be described as inversely proportional.

The design and technology of a structure on its own is not sufficient in determining the cost, or rather, life-cycle cost of it. The cost of design and inspection can in most cases be well predicted. These are often proportional to the weight of the structure. The majority of structural optimization papers generally deal with the minimization of the weight of the structure (Adeli and Sarma, 2006). Cost and production time data have been gathered and studied from different companies and manufacturers from all over the world. One has to consider the differences between labor costs at different locations. This generally has a large or sometimes the biggest effect on the structure, even if the technology is the same (Farkas and Jarmai, 2008). Weight often constitutes a significant part of the cost but the minimization of cost is the final objective for optimum use of

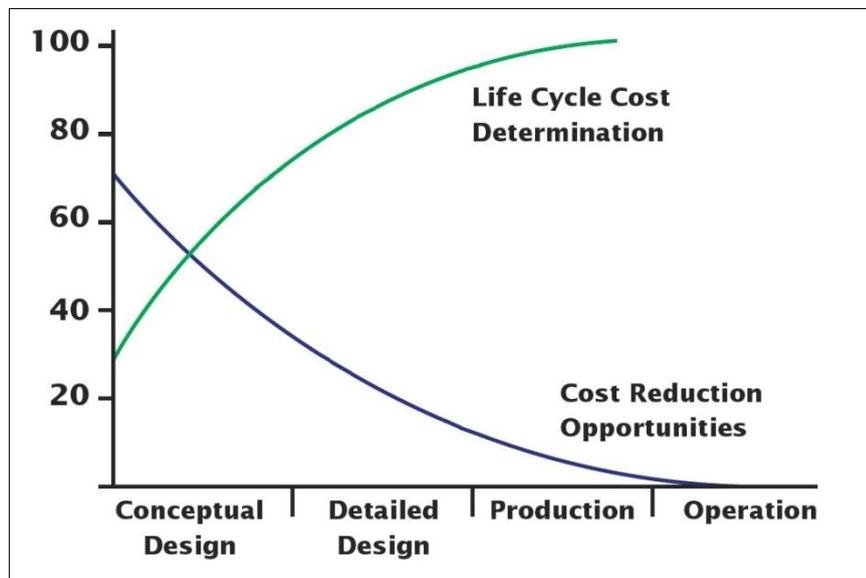


Figure 2.14: Relationship between reduction opportunities and committed life-cycle cost (UITP, 2002).

available resources (Adeli and Sarma, 2006). Concrete structures, for example, are made from more than one material and any optimization approach needs to be formulated uniquely or differently. For example, different structural concrete members have different reinforcement areas and number of ties (links). Also, mix designs are based on strength requirements and differ from shear walls through to foundations and beams. For the optimization of steel structures, the problem can be formulated as a weight minimization. A minimum weight design, as stated before, may not be a minimum cost design as other cost components apart from materials are influential as well.

Up to the 1990s, little research work had been published on the optimization of the overall or LCC cost for three-dimensional steel structures subjected to the constraints of the common codes of practice. Preferably, the optimization problem should be formulated in terms of the LCC, which includes the cost of materials, fabrication, erection, maintenance and dismantling the structure at the end of its life-cycle (Adeli and Sarma, 2006).

Adeli and Sarma (2006) recommend that considerations of the total cost as well as Life-Cycle Cost should be the prime focus of structural optimization in future.

2.7 Optimization of Wind Energy Conversion Systems

Various literature is available on component-based optimization of WECS. Examples of such components include the rotor and blades, generators, power transmission utilities, storage devices (batteries), structural support towers, foundations, mechanical bearings and shafts, stiffeners, and structural connections to mention some.

A wide and large variety of literature is available on turbine blade design optimization and aerodynamic optimization (Wind Energy Optimization Summit, 2009). According to proceedings from the *Wind Energy Optimization Summit, Hamburg, 2009* generators have been optimized extensively and near optimal solutions for generators have been achieved already, subject to the type of technology used.

Further, optimization on wind farm layouts have been made, taking into account a variety of factors such as the aerodynamic vortex shedding effects of upwind WECS on performance and structural integrity. Integer programming has been applied and the scope concerns mostly offshore wind projects. For onshore projects, optimization of topographic factors and multiple-site options has been carried out.

According to Gencturk and Elnashai (2012), structural optimization problems may be divided into three classes, namely sizing, shape and topology optimization. Figure 2.15 has been obtained from the work by Appelo (2012), *Structural Optimization via Genetic Algorithm*.

With sizing optimization, the location and number of structural elements are known and fixed. Generally the individual elements are optimized by minimizing an appropriate objective function that describes the properties of such elements, subject to prescribed constraints. Shape optimization concerns the contour of the boundary of a structural domain, but no new boundaries are formed. Topology optimization is the most general type as both size and location of structural members are determined and the formation of a new boundary is allowed. Furthermore, classification can be made based on the number of objectives, namely either single- or multi-objective.

The total weight of a steel structure is often chosen as a single objective in structural optimization (which also yields a unique optimal solution), while the performance is often based on several requirements of a design code, vibration amplitudes or similar criteria. Such requirements are introduced as constraints, either as inequality or equality

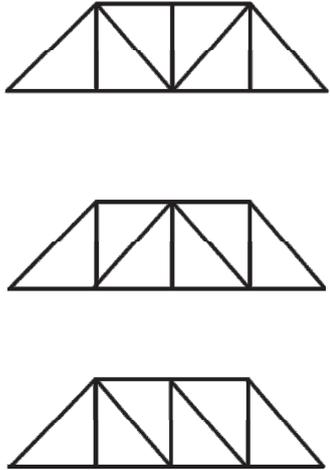
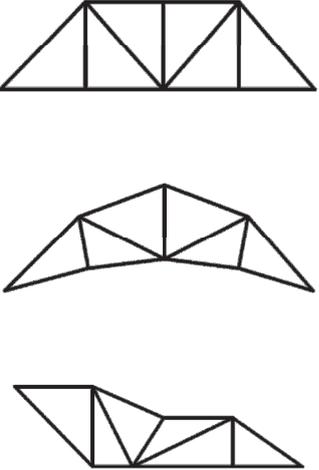
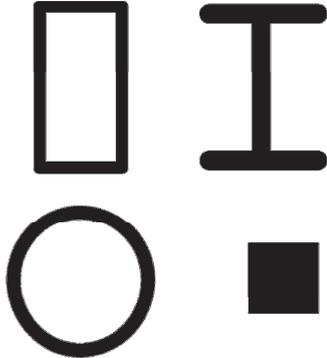
Topology Optimisation (Connectivity)	Shape Optimisation (Nodes)	Size Optimisation (Elements)
		

Figure 2.15: Three classes of structural optimization (Appelo, 2012).

constraints.

Gencturk and Elnashai (2012) also states that for LCC optimization multiple objectives are commonly used, such as structural performance and various associated costs. Such performance criteria have been established for buildings subject to seismicity. The performance measure considered is inter-storey drift and is found by using a static, non-linear pushover analysis. The exceedance of a certain threshold value has been related to a certain damage cost. However, the number of objectives is not the only difference between LCC and initial structural design optimization problems. The former might require the use of probabilistic formulations to evaluate the occurrence of an event (e.g. damage, corrosion or collapse during an extreme event) at the various limit states.

Uys *et al.* (2006) proposed an approach by formulating a set of cost functions for the design and fabrication process of a non-prismatic, ring-stiffened welded tubular tower. Wind loads are calculated according to Eurocode 1 Parts 2-4. Further considerations include constraints such as shell buckling and local buckling of the ring stiffeners. The approach from DNV/RISØ (2002) was applied to formulate these constraints. Rosen-

brock's search method was applied and the proposed design optimization provides for material and manufacturing cost minimization and could be directly incorporated into the LCC optimization process of non-prismatic, tubular towers that are primarily loaded by bending due to dynamic loads in their work. Uys *et al.* (2006) also state that successful application has been made to welded beams, tubular trusses, frames and shells to name a few.

In the paper *Life-cycle cost optimization of structures* by Sarma and Adeli (2002) the concept around the three major costs during the lifetime of a structure are mentioned. This includes the initial design/build cost, the operating cost and maintenance cost. Furthermore it is stated, that differences in design life and anticipated (actual) life of a structure can occur. Sometimes a structure is occupied or used much longer than designed for, and necessary measures to accommodate an extended lifetime need to be deliberated. It seems obvious, but also the type of material used for construction directly and indirectly influences the life-cycle cost of a structure due to repairs and maintenance. Consider a concrete element for example: Concrete may lose its strength after time with the formation of drying shrinkage cracks, creep effects or reinforcement corrosion in extreme environments and thus may require more periodic maintenance and repair.

The same paper by Sarma and Adeli (2002) includes the fuzzy multi-criteria discrete optimization model to obtain a life-cycle optimum. The reader should take note that the terminology multi-criteria and multi-objective are analogous in this thesis. The code was programmed in the *C/C++* language and the model was based on Allowable Stress Design (ASD) and Load Resistance Factor Design (LRFD) from AISC design codes. The methodology presented provides a logical way for the designer to consider the best design for the life-cycle of the structure and can be extended to include additional parameters relevant to the life-cycle design of a structure.

The work by Adeli and Sarma (2006) *Cost Optimization of Structures* includes chapters and sections on life-cycle cost optimization approaches on a variety of concrete and steel structures. The methods that are covered on steel structures are deterministic cost optimization (the majority of publications on optimization in this field make use of this method), cost optimization using the reliability theory and fuzzy logic based cost optimization. The sections on deterministic optimization cover specific cases like beams and plate girders, trusses, plane frames, industrial buildings, guyed towers and steel transmission poles to name a few. Difficulties and challenges of cost minimization are owed to the definition of the cost function, uncertainties and the fuzziness involved in determin-

ing the cost parameters. The book also includes a chronological review of papers on the cost optimization of concrete and steel structures from archived journals. The three cost optimization methods will be outlined in the following paragraphs.

Deterministic-based cost optimization can be described as follows. In terms of steel structures, a general cost function C_T can be defined as in equation 2.7.1, but does not cover the effects of maintenance, repair and dismantling during the life cycle of a structure.

$$C_T = C_m + C_f + C_t + C_e \quad (2.7.1)$$

where C_m is the material cost of structural members, C_f is the fabrication cost, C_t is the transportation cost and C_e is the cost of erection.

Certain deterministic behavioral constraints are generally considered. Examples for such constraints are allowable stresses or forces, deflections and inter-storey drifts.

Additional aspects in deterministic-based optimization are due to engineering practice demands. The members of a frame or truss type structure are generally grouped. The members of each group share the same design variables and this ‘linking’ of design variables results in a trade-off between the use of more material and the need for symmetry and uniformity of structures for practical considerations (Papadrakakis *et al.*, 2005). This can deviate significantly from a real, attainable global optimum that would be impractical.

Due to manufacturing limitations, design variables cannot be considered continuous and should be treated as discrete, since cross-sections belong to a certain set as provided by manufacturers (Papadrakakis *et al.*, 2005). The concept of fuzzy discrete multi-criteria optimization seems practical in this regard and is described in subsection 2.8.7.

The designer is forced to look at a good compromise solution that is also subject to conflicting requirements. This is generally referred to as a multi-objective or a multi-criteria optimization problem. Further details are discussed in the subsequent sections.

When only the material cost of structural members are included, the cost function can be presented as proportional to the volume of the structure (Adeli and Sarma, 2006). Such a problem is similar to a weight optimization problem. This simplification assumes that the various hot-rolled or cold-formed steel profiles that are usually used as struts, ties, beams, columns or beam-columns have the same unit price, which is actually not always

the case.

The work by Farkas and Jarmai (2008) *Design Optimization of Metal Structures* covers substantial aspects of structural optimization, with the focus on cost optimization in the initial phase of a structure. Precise formulations and expressions regarding cost functions for different structures, elements and materials are included and described. The cost functions include typical design and structural variables that carry the potential for effective optimization. Cost function examples provided are based on material, assembly, welding, fabrication, preparation, cutting, edge grinding, forming of shells and fabrication sequence. A primary conclusion of this work is that the fabrication details and costs play an important role in the optimum cost design of welded steel structures. It is concluded that cost optimization design can be more economical than weight optimization when considering only material and fabrication costs. However, little research has been done in this field, but the authors recommend the work of Klansek and Kravanja (2007), Jalkanen (2007) and Timar *et al.* (2003) on this topic.

Additional to the above, an investigation into optimization was made by Farkas and Jarmai (2008) on a ring-stiffened tubular and non-prismatic wind turbine support tower. The work covers similar and also some of the identical aspects as covered by Uys *et al.* (2006), due to the overlapping involvement of one of the authors. The investigation is made on a 1 MW wind turbine located in Greece and a thorough description regarding the optimization technique and methods, the objective function (cost), the design constraints (local and global buckling, fatigue, ultimate limit state for example) and some other checks are provided.

In a further section of the book, an investigation regarding a tubular lattice wind turbine support tower is presented. Similar, but adjusted formulations are provided for this type of structure.

The paper titled *Incorporation of Life-Cycle Models in determining Optimal Wind Energy Infrastructural Provision* by Cleary *et al.* (2012) emphasizes the reasons and opinions that at present support the large scale development of onshore and offshore wind energy systems. The authors state that the European Wind Initiative's main objective is to maintain technology leadership in both onshore and offshore wind energy by making it the most competitive energy sources by 2020 and 2030 respectively. This underlines and supports the research activities in the field of LCC optimization, also within the South African context.

From review of the available literature it seems that the present tendency is towards larger wind turbines in the range of more than 10 MW. Difficulties related to manufacture and transportation arise with the bottom sections of tubular, non-prismatic steel towers that have a hub height of about 90 m (and higher). These can no longer be transported by ordinary road transport due to bridge clearance limits and road width limits.

Cleary *et al.* (2012) also include details on support tower materials and considerations. The predominant design of wind turbine towers worldwide are tubular steel towers primarily due to ease in design and installation. The drawback with these is the increasing cost in steel prices, manufacturing constraints and logistical/transportation challenges as well as vibrational issues with towers higher than 85 m (fatigue and serviceability limit state related). Concrete and hybrid towers, a combination of steel and concrete, are becoming a viable, if not an optimal solution for tall wind turbine towers. Concrete tower solutions are being used onshore by at least three wind turbine manufacturers. However, offshore concrete tower uses have not been realized yet (Cleary *et al.*, 2012). With the ever increasing turbine sizes, the need to optimize wind turbine structures is vital to reduce the cost of wind energy. Cleary *et al.* (2012) proposed a multi-objective harmony search optimization algorithm to optimize the set of objective functions subject to a set of constraints. The result is a vector within the design space that optimizes a set of objectives and satisfies this set of constraints. More details on multi-objective optimization techniques and formulations of objectives, variables and constraints can be found in subsection 2.8.3.

Harmony search is a new meta-heuristic technique and is inspired by the natural musical performance process that occurs when a musician searches for a better state of harmony. Fewer mathematical requirements are needed and they can be easily adapted for solving various kinds of optimization problems. The methodology that was followed considers two objective functions, namely (1) the ratio of the cost to produce energy to the amount of energy that is produced. The cost formulation includes the life-cycle net present value cost of electricity generation, namely capital cost, maintenance and operation cost and decommissioning cost. The second objective function, (2) seeks to minimize the emission intensity of electricity production and is given as the ratio of the life-cycle emissions of generation to the electricity produced. The life-cycle emissions include capital, maintenance, operational and decommissioning related emissions. In other words, an emission life-cycle assessment is developed.

The paper *Structural Design Optimization of Wind Turbine Towers* by Negm and Maalawi

(2000) discusses several structural optimization models for a typical wind turbine tower structure. The techniques discussed generally do not directly consider the cost optimality criteria, but rather other structural response and behavioral characteristics. The central focus of the paper concerns optimization considering the dynamic response of a tubular wind turbine steel tower. Nevertheless, important aspects that relate to cost are covered. In the study by Negm and Maalawi (2000), the basic aspects concerning design optimization of the combined tower/rotor structure is also investigated. A simplified approach is made by assuming the rotor/nacelle part to be a rigid non-rotating lumped mass placed at the top of the tower. The optimization problem is formulated as a non-linear mathematical programming problem that is solved by the interior penalty function technique. The author explains that the normal mode method is applied to obtain the forced response for different excitation types and it is claimed that a global optimum is attainable from the discretized model.

A simplified approach is applied and the wind turbine tower is considered to be built from uniform, prismatic, tubular segments or modules. The effective design variables are the height of each segment, the cross-sectional area and the radius of gyration. The isolated tower dynamics include the formulation of the applied loads and the kinematic analysis. The set of governing dynamical equations of motion are formulated in an appropriate non-dimensional form.

Several basic assumptions that are described in the work by Negm and Maalawi (2000) in formulating the optimization problem are listed. Three principal phases must be considered when formulating an optimization problem, and are given below:

- Defining the system objectives and the measuring thereof.
- Selecting design variables and parameters.
- Defining the design constraints.

Their tower model is represented by an equivalent long, slender cantilever beam built from segments (modules) having different, uniform cross-sectional parameters. The inertia parameters of the nacelle/rotor are approximated and assumed rigid. The construction material, steel in this case, behaves linearly elastic, isotropic and is homogeneous. The Euler-Bernoulli beam theory is used to predict deflections, and any secondary effects such as shear and axial deformations are neglected. Aerodynamic forces are also restricted to profile drag forces and a two dimensional steady flow model is assumed. Non-structural

mass was not optimized and structural analysis was limited to the case of bending of the tower perpendicular to the plane of the rotor disk.

The set of objectives for the tower design included the following. Negm and Maalawi (2000) discuss the following design objectives related to the optimal tower design: Light-weight design, high stiffness, high stiffness/mass ratio and design for minimum vibration. The design for minimum vibration levels is extensively discussed in the paper and forms the largest portion of the paper. Minimization of the overall vibration level is one of the most cost-effective solutions for a successful wind turbine design. This also favors other important design objectives such as long fatigue life, high stability and low noise level (Negm and Maalawi, 2000). Two different criteria for measuring vibration reduction are described, namely the frequency placement and maximum frequency criterion. The first criterion is achieved by separating the natural frequencies of the structure from the excitation frequencies to avoid resonance effects. The second concerns the maximum frequency placement of the system natural frequencies, by applying the strategy of maximizing a weighted-sum of the system modal frequencies. Higher frequencies cause a reduction in both the steady-state and transient responses of the tower (Negm and Maalawi, 2000).

The developed models in the paper by Negm and Maalawi (2000) have been successfully applied to an existing 100 kW (small) wind turbine. The maximization of a weighted-sum of the system natural frequencies proved to be the most representative objective function which directly reflects the major design goals and ensures a balanced improvement in both the stiffness and mass of the tower. A global optimum can be attained and the appropriate non-dimensionalization of parameters has led to a naturally scaled model, which eliminates the need for scaling of the design variables and therefore errors paired with it. Apart from this, the authors also state that much computation time is saved from the exact optimization process. Time that is required by finite element methods and other discretized approximate methods.

In the following section, an overview regarding newer, mathematical optimization methods and approaches will be given. It is essential to discuss the different types of techniques, methods and viewpoints as the theory plays a role in this study.

2.8 Mathematical optimization methods

2.8.1 Introduction

There are various mathematical optimization methods that can be applied to real world problems. With personal computers, that have become very powerful and affordable, numerical optimization problems are left for the computer to perform effectively.

Most classical mathematical optimization methods require auxiliary information such as derivatives of an objective function to find optima. Many of these methods require the search space to be continuous and also an equation to model the problem which is often impossible to formulate, especially in the case of discrete variables. Finding a local optimum is relatively easy, but finding a global or near global optimum is more challenging (Reynolds, 2009). A simple technique would be to evaluate the objectives at each available point, but this is highly inefficient. In structural engineering, optimization problems are generally highly constrained and non-linear. An unconstrained optimization would simply and trivially result in a structure with the smallest members available, with no consideration given to the structure to support itself or to carry loads.

For real world structural engineering problems, a suitable optimization technique should have the following characteristics:

1. Have the capacity to be easily extended to optimize real structural designs and not only formulated benchmark problems.
2. Solve optimization problems with discrete variables.
3. Attempt to find the global optimum or near global optimum.
4. Require a minimum of auxiliary information, such as function derivatives.

A number of tools are available and algorithms can be implemented accordingly. Reynolds (2009) states, that generally traditional mathematical programming techniques require auxiliary information such as the first derivative and are therefore not very suitable. MATLAB for example (a contraction of Matrix Laboratory) is a cross-platform numerical and technical computing environment written in *C* and is a proprietary commercial product of The Mathworks, Inc. It offers a so-called Optimization Toolbox that contains a library of functions that aid in optimization. A branch of applied numerical analysis called Global Optimization has been developed. Within the branch of Global Optimization in

general, there are a number of search techniques which are divided into the categories: deterministic, stochastic and heuristic.

As outlined before, the focus of this research lies on optimization of LCC of wind turbine support towers. The LCC philosophy is valid for all phases; thus from initial design to decommissioning. However, the structural design optimization also forms part of the LCC minimization and will therefore be investigated comprehensively. In the structural optimization process, it is of utmost importance that the engineer knows the behavior of the structure well. This includes the stresses, vibrations, stability and deformations to name a few. In order to find an optimum solution or any solution the question regarding what is an optimum solution, arises. According to Farkas and Jarmai (2008), such a solution is the outcome from a well-defined search towards the best possible alternative from a set of feasible solutions.

Often, engineers or designers need to choose a certain method or technique available and a further question regarding the suitability arises. It can generally be said, that none of the algorithms that have been developed is superior, but rather that they all have their advantages and disadvantages. There are a vast number of optimization methods available for single objective optimization. Some techniques are non-gradient based, others are gradient based. The gradient based techniques are generally either of first or second order (Beale, 1988). Non-gradient based optimization techniques such as Evolutionary and Ant-colony optimization are additional techniques that previously have been proven applicable in the field of structural optimization. The Evolutionary and Ant-colony techniques are heuristic types that are based on a mutation and crossover and stochastic philosophy respectively (Farkas and Jarmai, 2008).

2.8.2 Single objective optimization

The formulation of a single objective, non-linear optimization problem is as follows:

$$\text{minimize } f(\mathbf{x}) = f\{x_1, x_2, \dots, x_n\} \quad (2.8.1)$$

$$\text{subject to } g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, P \quad (2.8.2)$$

$$\text{and } h_i(\mathbf{x}) = 0, \quad i = P + 1, \dots, P + M \quad (2.8.3)$$

As mentioned, for the general case $f(\mathbf{x})$ is a multivariable, non-linear objective function. $g_j(\mathbf{x})$ and $h_i(\mathbf{x})$ are non-linear inequality and equality constraints. The general drawback with any optimization is paired with the search process for the global optimum. Often the

algorithm that is implemented gets ‘trapped’ in a local minimum and an unpredictable and wrong solution can result (Pahl, 2012).

2.8.3 Multi-objective optimization

With multi-objective optimization two or more, sometimes conflicting or incommensurable objective functions are to be optimized simultaneously (Farkas and Jarmai, 2008). In general, the solution to the multi-objective problem is a set of points that are the best trade-off between the various objective functions. This set is referred to as the Pareto optimal set of points. At this point, one has to bear in mind that the solution is not unique, but a set of optima, that all satisfy the optimality criteria. The general consensus among engineers and mathematicians is that the Pareto optimal set may contain information that can help the designer to make a decision and thus arrive at better trade off solutions (Cleary *et al.*, 2012). It is also stated, that a feasible design point is said to be Pareto optimal if no other feasible design can improve some of the objectives without being unfavorable to others. Refer to Figure 2.16 which illustrates the concept of the Pareto optimal set. The Pareto set or front is defined as the set of solutions which are such that no improvement can be obtained with one objective without deteriorating at least one of the other objectives. It is also termed the Pareto set of non-dominated solutions and assists considerably in decision making processes. In Figure 2.16 the Pareto front for the LCC is the entire set of values in the figure that satisfy the definition criteria and becomes useful when it comes to multi-objective optimization problems with economic considerations.

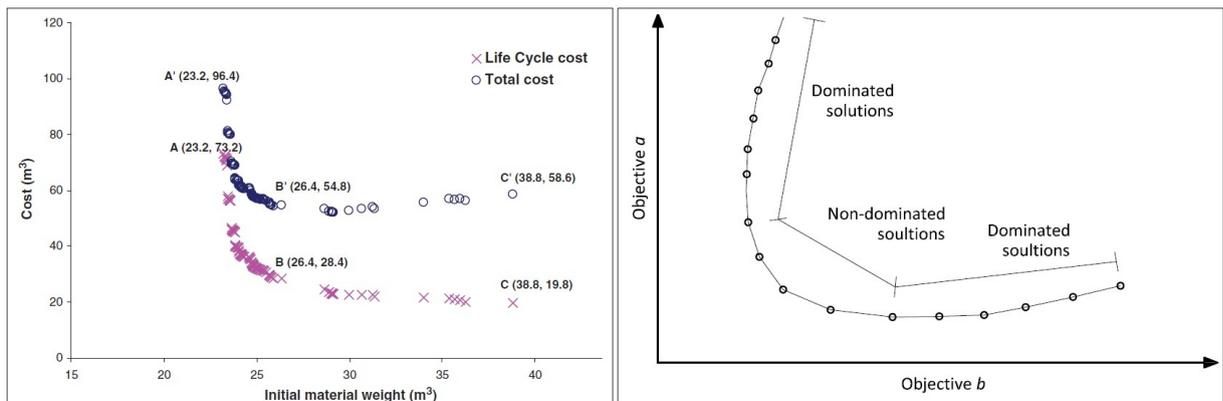


Figure 2.16: Representation of the Pareto set of optimum results (Cleary *et al.*, 2012).

The general multi-objective optimization problem can be set out as follows:

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\}^T \quad (2.8.4)$$

$$\text{subject to } g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, P \quad (2.8.5)$$

$$\text{and } h_i(\mathbf{x}) = 0, \quad i = P, \dots, P + Q \quad (2.8.6)$$

$$x_l^{\text{lower}} \leq x_l \leq x_l^{\text{upper}}, \quad l = 1, \dots, N \quad (2.8.7)$$

where $\mathbf{x} = x_1, \dots, x_N$, is the design vector with N variables defined in an n -dimensional space and $\mathbf{f}(\mathbf{x})$ is the objective function vector with M independent functions. $g_j(\mathbf{x})$ and $h_i(\mathbf{x})$ are the inequality and equality constraints, respectively. The upper and lower constraints on the vector components of \mathbf{x} , namely x_l^{lower} and x_l^{upper} are called the boundary constraints (Cleary *et al.*, 2012). The number of such boundary vector components can deviate from the number of the respective constraints.

One way to generate a Pareto optimal set is to make use of a weighted objective approach. Farkas and Jarmai (2008) state that a global objective function can be defined as a weighted sum of the values resulting from the various objective functions in a certain problem. Furthermore it is said, that population-based problems, such as Evolutionary Algorithm problems or Ant/Swarm Colony Algorithm problems can be used without defining a combined function.

The linear weighting methods described by Farkas and Jarmai (2008) are outlined below. Pure weighting is the adding of the objective functions together using different weighting coefficients for each. In other words, the multi-criteria optimization problem is transformed to a scalar and a single function of the following form is created:

$$f(\mathbf{x}) = \sum_{i=1}^M w_i f_i(\mathbf{x}) \quad (2.8.8)$$

where

$$w_i \geq 0 \text{ and } \sum_{i=1}^M w_i = 1 \quad (2.8.9)$$

By varying the i^{th} weight (w_i) it is possible to generate a set of Pareto optimum solutions for the problem definition in equation 2.8.4. The values of the weights w_i can be adjusted according to the importance of each criterion. Every combination of the various weights results in a single Pareto optimal solution. Performing a set of optimization processes with different weighting factors, a set of Pareto optimal solutions can be generated (Papadarakakis *et al.*, 2005).

2.8.4 Gradient based optimization

Optimization methods are generally divided into gradient based and non-gradient based methods. Figure 2.17 illustrates the classification of these optimization methods, together with some commonly used examples.

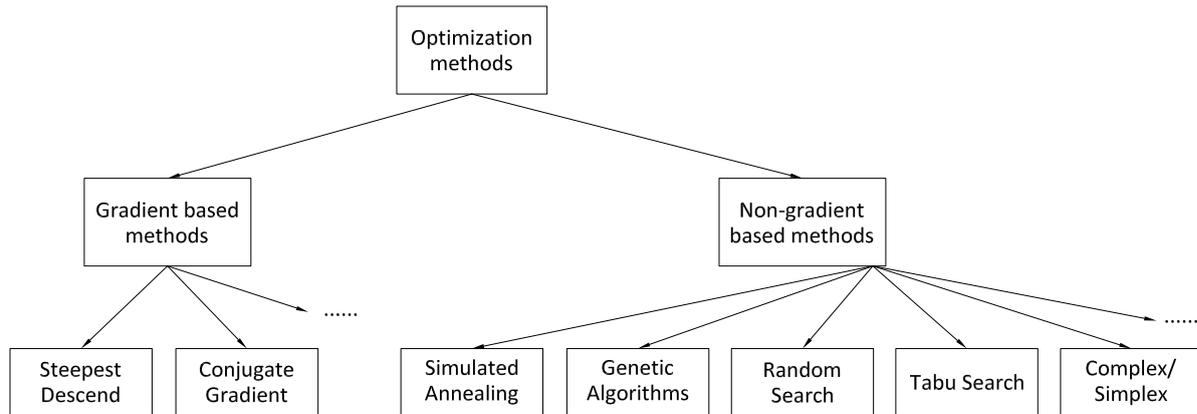


Figure 2.17: Classification of gradient and non-gradient based optimization methods with some commonly used examples (Anderson, 2001).

According to Cencelli (2006), gradient-based optimization methods are practical in the search for an optimum solution by defining a search vector to determine the direction of the most feasible location of the optimum. The principles from fundamental calculus theory hold, as the gradient of a function indicates whether a function value grows or diminishes. The search vector is thus, similarly, defined as a partial derivative of the objective function. Pahl (2012) comments on useful techniques that aid in the search for the global optimum. Techniques include linear programming, the method of steepest descend and the conjugate gradient method. The latter two are methods that entail the use of a search vector, as mentioned before. Direct computing is generally preferred, but often objective functions in engineering are not linear or quadratic and the analytical form of an objective function is not well defined. The answer to this is a numerical extension of the previous methods; although several difficulties are paired with this. With numerical methods it is often impossible to distinguish between local and global minima. Also, convergence and accuracy are not guaranteed.

With the objective and various constraints defined, the search of the optimum can commence and the general method of defining a search vector will be explained. The search vector \mathbf{S} indicates the direction of steepest descent and is defined as follows (Cencelli, 2006):

$$\mathbf{S} = -\nabla f(\mathbf{x}) = \begin{Bmatrix} \frac{f(x+\partial x_1)-f(x)}{\partial x_1} \\ \vdots \\ \frac{f(x+\partial x_N)-f(x)}{\partial x_N} \end{Bmatrix} \quad (2.8.10)$$

The search vector \mathbf{S} , here defined as the vector in the steepest descent method (for illustration), will most probably find a local minimum first along this linear, incremental search path. A step of size α will be taken in this direction from a starting design variable x^0 . The progression of the design variable towards an optimum solution can be expressed as follows:

$$x^q = x^{(q-1)} + \alpha^* S^q \quad (2.8.11)$$

where q is the iteration number and α^* is the optimum step size.

From calculus theory it is known that the optimum occurs where the derivatives are zero. As shown by Cencelli (2006), this is the Kuhn-Tucker condition that indicates when an optimization search towards a global minimum is complete. Once a local minimum is reached, a new search is initiated and the search process starts again as the Kuhn-Tucker condition is not satisfied yet. The Kuhn-Tucker condition can be expressed as follows:

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^P \lambda_j \nabla g_j(x^*) + \sum_{m+i}^{P+Q} \lambda_{m+i} \nabla h_i(x^*) = \mathbf{0} \quad (2.8.12)$$

λ is known as the Lagrange multiplier. If a constraint is not violated, then the corresponding Lagrange multiplier is zero. If this is not the case, the multiplier assumes the initial value and the Kuhn-Tucker condition is penalized according to the degree set by the Lagrange multiplier (Cencelli, 2006). If all constraints are satisfied, the solution converges to the global optimum.

2.8.5 Non-gradient based optimization

Non-gradient based optimization methods often imitate natural phenomena such as evolutionary characteristics or swarm behavior that is common with schools of fish or insect swarms. The two general terms used to describe these methods are Evolutionary/Genetic Algorithms (EA or GA) and Particle Swarm Optimization (PSO).

Upon initialization of the optimization algorithm, the particles or variables are randomly distributed over the solution space (Cencelli, 2006). The following expression describes the phenomenon of a school of fish adapting to its path or a system of ants in their search for food.

$$x_{q+1}^i = x_q^i + v_{q+1}^i \Delta t \quad (2.8.13)$$

x^i is the position of the design solution, q is the iteration counter and v^i is the velocity vector and Δt is the increment size. The velocity vector for this example is defined as follows:

$$v_{q+1}^i = wv_q^i + c_1 r_1 \frac{p^i - x_q^i}{\Delta t} + c_2 r_2 \frac{p_q^g - x_q^i}{\Delta t} \quad (2.8.14)$$

r_1 and r_2 are randomly generated numbers between 0 and 1. p^i is the fittest solution found by particle i and p_q^g is the fittest position or solution found by the combined swarm at iteration point q . w is the inertia of a particle, i.e. its resistance to movement and direction changes while c_1 and c_2 are trust parameters that are allocated after each iteration. Cencelli (2006) further mentions, that large values of inertia result in a more global behavior of the particle, while smaller inertia values inspire local behavior. The trust parameters c_1 and c_2 indicate the confidence of a particle in itself and the swarm respectively. Self confidence levels and smaller values of inertia therefore tend to result in local convergence behavior and the opposite holds for global convergence. The trust parameters could be related to a pheromone level parameter in ant colony optimization. Pheromone is the ‘substance’ that each ant leaves on its track as a marker during the search for food. The pheromone levels are updated after each step or each iteration in the algorithm, depending on whether it is an Ant System or Ant Colony System (Pahl, 2012). In optimization the pheromone trail influences the stochastic processes by which the ants construct new individuals in the next generation (Pahl, 2012). A different level in pheromone in turn influences the trust or confidence level accordingly.

2.8.6 Life-cycle cost optimization using reliability theory

Cost optimization and specifically LCC optimization has mostly been based on a deterministic approach in the literature (Adeli and Sarma, 2006). Reliability theory includes the uncertainties in the calculation and determination of the design resistances and ac-

tions on a structure. Deterministic optimization generally limits the optimization for a predetermined set of actions only.

With reliability theory, these are considered to be random variables. The safety, often expressed as a reliability or safety index β , is related to the probability of the action (a random variable) exceeding the structural capacity (also a random variable). An attempt is often made to consider different modes of failure and different loading scenarios simultaneously and this can be well incorporated with the reliability-based optimization. Instead of direct integration of the probability density functions to determine the probability of failure P_F , alternatives such as the First Order Reliability Method (FORM) or Second Order Reliability Method (SORM) can be effectively applied.

Adeli and Sarma (2006) state the fact that the major drawback with the reliability-based optimization approach is the computation of the probability of failure. This cannot be done consistently due to the lack of statistical information concerning the random variables.

The reliability factor in the optimization can be considered either directly or indirectly (Adeli and Sarma, 2006). With the direct approach, the reliability factor is included in the objective function. With the direct approach, the total cost C_T is calculated by summing the initial cost C_I , a function of the design variables (discussed in the subsequent sections) and the expected failure cost C_F multiplied by a probability of failure P_F . The probability of failure is also considered a function of the design variables and equation 2.8.15 represents the above description mathematically:

$$C_T = C_I + P_F C_F \quad (2.8.15)$$

subject to the following:

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, P \quad (2.8.16)$$

$$h_i(\mathbf{x}) = 0, \quad i = P, \dots, P + Q \quad (2.8.17)$$

where P and Q are the number of inequality and equality constraints respectively. The expected failure cost includes the cost related to the failure of the structure. Examples include aspects such as damage, casualties, litigation and replacement cost, to name a

few. The second term in equation 2.8.15 is the quantifiable risk (in monetary terms) of the actions on the structure exceeding the structure's capacity.

With the indirect method, the objective function only comprises of the initial cost C_I and the reliability term is considered indirectly in the form of a constraint in addition to the design constraints in expressions 2.8.16 and 2.8.17, such as

$$P_F \leq P_{F \text{ allowable}} \quad (2.8.18)$$

Therefore, this approach converts a deterministic optimization procedure into a reliability-based optimization by adding a constraint (Adeli and Sarma, 2006). Sørensen (2007) claims that with probabilistic life-cycle cost optimization of WECS (and structures), the probability of loss of human lives is often negligible for wind turbine systems, especially for offshore types. In these cases it can be relevant not to include the constraint and thus a purely monetary optimization is obtained. However, this is not exactly applicable to onshore types and this aspect is project specific and falls beyond the scope of this work.

2.8.7 Fuzzy discrete multi-objective life-cycle cost optimization using GAs

In Adeli and Sarma (2006), the concept of fuzzy discrete optimization using genetic algorithms (GAs) is introduced, discussed and also applied to frame structures and lattice type truss structures. The concept is based on the theory of fuzzy sets that was developed by Zadeh (1965). Its application to the field of structural engineering was introduced by Brown and Yao (1983).

The fundamental concept of reliability-based optimization is based on the theory of probability, while fuzzy optimization on the other hand deals with the theory of possibility. The latter is a more recent theory and therefore encourages further investigation and application to modern structural engineering practice.

With probability theory, variables or events happen at a random fashion in nature on a statistical basis, while the possibility of fuzzy variables is based on nonstatistical variables. Numerical values in probabilistic optimization are defined by probability density functions, while fuzzy values are defined by membership functions (Adeli and Sarma, 2006). Adeli and Sarma (2006) also report the development of a fuzzy controlled genetic search algorithm (GA) for the minimum weight and shape optimization of steel trusses by

coupling a heuristic fuzzy rule-based system with the GA without using the constraints of any commonly used design code.

The evaluation of structural resistance and the evaluation of the loads acting on a certain structure are regarded as the two major sources of uncertainty, ambiguity or fuzzyness. However, with cost optimization, the evaluation and formulation of the cost function forms the third leg (Adeli and Sarma, 2006).

Consider any set Z of variables. Y is a fuzzy set for any set Z and is characterized by a membership function $\mu_Y(z)$ which grades each point in Z with a value between zero and unity, i.e. on an interval $[0,1]$. The grade of membership of z in Y is described by this membership function. If the value of $\mu_Y(z)$ is close to unity, the grade of membership of Z in Y is high and the opposite case applies when its value is close to zero. Thus, a fuzzy set Y can be defined by the following expression.

$$Y = \{z, \mu_Y(z)\} | z \in Z \quad (2.8.19)$$

Fuzzy set theory has been used to model uncertainties in decision making situations. The membership functions of a fuzzy set are used to develop a transition from total rejection to total acceptance. By treating the constraints and objectives in a GA as fuzzy variables, the chance of obtaining the global optimum will be increased. The reason for this is that if a candidate solution does not satisfy a certain constraint strictly and violates that constraint only slightly, it might be discarded by the genetic search and it might miss the potential global optimum in the vicinity of the candidate solution.

Several papers have been published on fuzzy optimization of structures and the trend is in the direction of structural weight optimization. Only a few papers deal with cost optimization of structures. In most papers the cost function for the fuzzy cost optimization is expressed as the sum of the initial cost and the cost of maintenance and failure. This cost summation formulation is, broadly said, somewhat similar to the reliability-based optimization formulation, however without inclusion of the probability of failure factor.

Wang and Wang (1985a) have published several papers on this topic. In Wang and Wang (1985a), a simplified fuzzy optimization procedure is presented. It is termed the α -level cut method and considers the fuzzyness in the constraints and uses a non-fuzzy cost function. However, the amount of fuzzyness for the constraints is limited to preselected ranges. The problem is transformed to ordinary non-fuzzy single or multi-objective optimization

with lower and upper bound limits as described for multi-objective optimization in equation 2.8.7. These limits are a function of α and the problem can be easily solved using software such as MATLAB and even MS Excel (with Macros). However, the disadvantage is due to the somewhat arbitrary selection of values for α (Adeli and Sarma, 2006).

The approach described above has been applied by Adeli and Sarma (2006). A three-bar truss and a two-storey frame have been considered and investigated in a study (Adeli and Sarma, 2006).

Kelesoglu and Ulker (2005) used fuzzy set theory to perform a multi-objective optimization of a space truss. Take note that in this work, the terms ‘multi-criteria’ and ‘multi-objective’ are analogous. The authors used two objectives, namely the weight of the structure and the deflection. An approach, called the λ -formulation, as introduced by Rao *et al.* (1992) was applied in the study. The problem can be easily expanded to accommodate additional objectives for initial cost, repair cost, maintenance cost and other LCC related parameters.

Equality constraints are generally not included in the fuzzy formulation, because these need to be satisfied exactly.

The strategy of multi-objective fuzzy optimization using a GA was adapted as described under 2.8.3. The formulation on the technique described in 2.8.3 above will be briefly repeated here for reference purposes. Take note that the constraints are written in terms of the constraint function values and not in terms of the lower and upper vector components this time:

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\}^T \quad (2.8.20)$$

subject to

$$g_l^{\text{lower}} \leq g_l(\mathbf{x}) \leq g_l^{\text{upper}}, \quad l = 1, \dots, N \quad (2.8.21)$$

where \mathbf{x} is the design vector, $\mathbf{f}(\mathbf{x})$ is the vector of the objective functions and $g_j(\mathbf{x})$ is the j^{th} constraint function. The degree of membership must be $\mu_{g_j}(\mathbf{x}) > 0$ (Kelesoglu and Ulker, 2005).

The description of the λ -formulation from Rao *et al.* (1992) will be discussed in the following paragraph and is very useful from the point of view that an overall compromise optimal design can be achieved.

When fuzzyness is considered in both the constraints and the objective function, the design variable vector \mathbf{x} is obtained from a fuzzy domain \tilde{D} so that the membership function $\mu_{\tilde{D}}$ for the fuzzy domain can be obtained as from the intersection of the fuzzy membership functions for both, namely the constraints and the objective functions (Adeli and Sarma, 2006). The general procedure is explained in the following paragraphs.

Refer to the relationship in equation 2.8.22 below.

$$\mu_{\tilde{D}} = \mu_f(\mathbf{x}) \cap [\cap \mu_{g_i}(\mathbf{x})] \quad (2.8.22)$$

where $\mu_f(\mathbf{x})$ and $\mu_{g_i}(\mathbf{x})$ are the membership functions for the objective functions and the i^{th} inequality design constraint, respectively. From this fuzzy domain, the optimum solution \mathbf{x}^* for the design variable \mathbf{x} can be found using the min-max method by Bellman and Zadeh (1970).

$$\lambda = \mu_{\tilde{D}}(\mathbf{x}^*) = \text{maximize } \mu_{\tilde{D}}(\mathbf{x}) \quad (2.8.23)$$

where

$$\mu_{\tilde{D}}(\mathbf{x}) = \min[\mu_f(\mathbf{x}), \mu_{g_i}(\mathbf{x})] \quad (2.8.24)$$

This max-min procedure can be solved by maximizing the λ parameter using a GA.

Rao *et al.* (1992) states, as implied in equation 2.8.24, that a design vector \mathbf{x} can be considered feasible if $\mu_D(\mathbf{x}) > 0$. The following computational procedure is proposed by Rao *et al.* (1992). The solution to the multi-objective optimization problem according to the λ -formulation can be found by (1) finding the solutions of the various objective functions, (2) then determine the best and worst solution of each objective function. (3) The solutions obtained thus far are used as boundaries of the fuzzy range in the optimization problem and (4) the resulting problem then is solved (Rao *et al.*, 1992).

λ is maximized using a GA subject to the following conditions:

$$\lambda \leq \mu_f(\mathbf{x}) \quad (2.8.25)$$

$$\lambda \leq \mu_{g_i}^{upper}(\mathbf{x}) \quad i = 1, 2, \dots, N \quad (2.8.26)$$

$$\lambda \leq \mu_{g_i}^{lower}(\mathbf{x}) \quad i = 1, 2, \dots, N \quad (2.8.27)$$

$$0 \leq \lambda \leq 1.0 \quad (2.8.28)$$

where the symbols used are as discussed previously. $\mu_{g_i}^{lower}(\mathbf{x})$ and $\mu_{g_i}^{upper}(\mathbf{x})$ are the membership functions for the lower and upper bounds of the N inequality constraints $g_i(\mathbf{x})$

respectively. $\mu_{f_i}(\mathbf{x})$ is the membership function of the i^{th} objective function. Also note that the equality constraints have not been included as they have to be satisfied exactly. The counter term Q in equation 2.8.7 in section 2.8.3 therefore becomes zero for fuzzy optimization. N is the total number of boundary constraints under consideration.

The membership functions discussed and included in the expressions 2.8.25 to 2.8.27 above are constructed as follows.

$$\mu_{f_i}(\mathbf{x}) = \begin{cases} 0, & \text{if } f_i(\mathbf{x}) > f_i^{max} \\ \frac{-f_i(\mathbf{x}) + f_i^{max}}{f_i^{max} - f_i^{min}}, & \text{if } f_i^{min} < f_i(\mathbf{x}) \leq f_i^{max}, \quad i = 1, 2, \dots, k. \\ 1, & \text{if } f_i(\mathbf{x}) \leq f_i^{min} \end{cases} \quad (2.8.29)$$

The fuzzy constraints can be stated as $g_j^{(l)} - \Delta g_j^{(l)} \leq g_j(\mathbf{x}) \leq g_j^{(u)} + \Delta g_j^{(u)}$. The boundaries of the j^{th} constraint are moved by the distance values of $\Delta g_j^{(l)}$ and $\Delta g_j^{(u)}$ respectively. Thus the membership functions of the lower and upper constraints are defined as shown below in expressions 2.8.30 and 2.8.31 respectively.

$$\mu_{g_j^{(l)}}(\mathbf{x}) = \begin{cases} 1, & \text{if } g_j(\mathbf{x}) \geq g_j^{(l)} \\ \frac{g_j(\mathbf{x}) - g_j^{(l)} + \Delta g_j^{(l)}}{\Delta g_j^{(l)}}, & \text{if } g_j^{(l)} - \Delta g_j^{(l)} < g_j(\mathbf{x}) < g_j^{(l)} \\ 0, & \text{if } g_j(\mathbf{x}) \leq g_j^{(l)} - \Delta g_j^{(l)} \end{cases} \quad (2.8.30)$$

$$\mu_{g_j^{(u)}}(\mathbf{x}) = \begin{cases} 1, & \text{if } g_j(\mathbf{x}^*) \leq g_j^{(u)} \\ \frac{-g_j(\mathbf{x}) + g_j^{(u)} + \Delta g_j^{(u)}}{\Delta g_j^{(u)}}, & \text{if } g_j^{(u)} < g_j(\mathbf{x}) < g_j^{(u)} + \Delta g_j^{(u)} \\ 0, & \text{if } g_j(\mathbf{x}) \geq g_j^{(u)} + \Delta g_j^{(u)} \end{cases} \quad (2.8.31)$$

Adeli and Sarma (2006) state that only small scale academic cost optimization examples have been considered and discussed in the literature and limited application examples to moderate or larger structures are available. An example that consists of only four axial compression elements, and therefore only a single design variable, will be described and explained in chapter 5, section 5.6 to verify the calculated results with those obtained in the original paper. The theory of fuzzy discrete multi-criteria optimization using a GA will be applied to a full scale WECS lattice tower in chapter 5.

A multi-objective optimization problem can be handled in four different ways depending on when the decision maker articulates his/her preference on the different objectives,

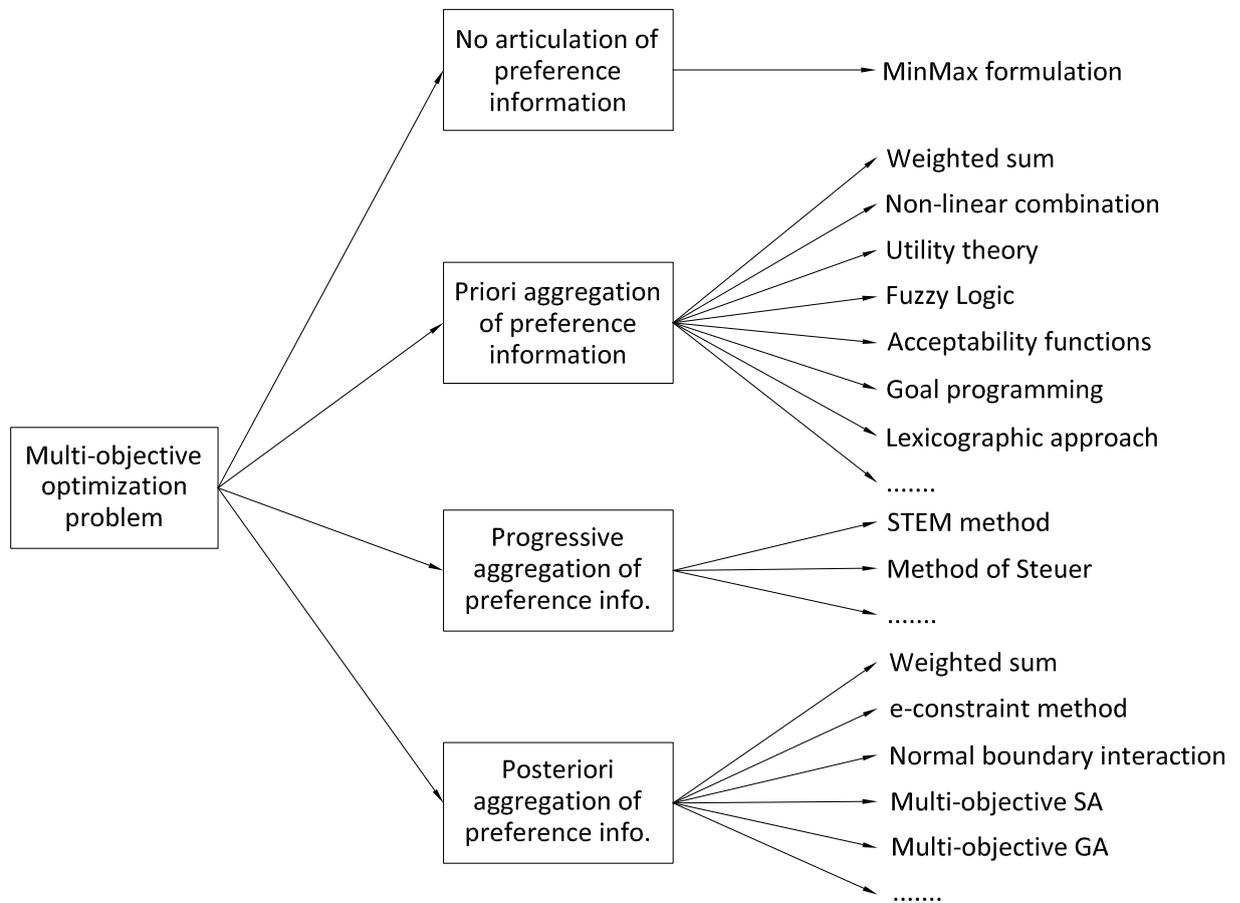


Figure 2.18: Multi-objective methods classification (Anderson, 2001).

namely (1) never, (2) before, (3) during or (4) after the optimization. Figure 2.18, extracted from Anderson (2001), is not a complete illustration of all available techniques but remains a good framework of the most common methods.

2.8.8 Genetic Algorithms as a tool in LCC optimization

2.8.8.1 Introduction

Evolution is the process by which a species improves certain characteristics over generations. The offspring inherit genetic qualities from their parents with reproduction. In biology, the genetic material is stored in chromosomes in a specie's cell nuclei as deoxyribonucleic acid (DNA). The DNA consists of a sequence of molecules, or rather bases, that are attached to a double helix of long polymers which provide its structure. In biological terms, each base can be one of four types, namely either Guanine, Adenine, Thymine or Cytosine (conventionally labeled G, A, T and C respectively) and are stored in pairs of these four types. At a simplified level, DNA can be thought of as a string which encodes genetic information using a long string consisting of many base pairs.

During reproduction, the offspring genes are created from the DNA of its parents through a combination of genetic crossover and mutation. During crossover, the genetic material of the parents is combined and mutation incorporates a random variation into the genes of the offspring. This creates a variation in the gene pool by new genetic features in the genes of the offspring. This random element of change to the DNA may or may not improve the fitness of an individual. If the resulting change from crossover and mutation does improve the fitness of an individual, survival chances are increased and reproduction becomes more likely. The DNA is then passed on to its offspring and the cycle continues. The selection process, in the form of 'survival of the fittest' ensures that genetic qualities which improve the current fitness are more likely to survive to the next generation.

Genetic Algorithms form a subset of evolutionary strategies and are inspired by the evolutionary biology and Darwinian principles as described above. GAs are a heuristic global search technique that has been chosen for investigation in this work because GAs do not require any auxiliary knowledge on derivatives and are suited to solve problems involving discrete variables. Also, the technique satisfies the four requirements stated in subsection 2.8.1. They are also computationally simple, relatively easy to implement and exceptionally robust and powerful in their search for global optima (Reynolds, 2009). Genetic Algorithms do not work on a point to point basis, instead they work on an entire population simultaneously.

Evolution operates blindly using elementary operations to manipulate the genetic material of a chromosome to create exceptionally complicated life. GAs operate on a similar principle, using simple encoding and reproduction processes. The simplest basis and easiest to understand, is binary encoding. There are a number of other encodings used

with GAs, for example integer encoding and many more, but with discrete structural optimization binary encoding seemed to be the most practical and logical approach.

GAs require a scalar fitness to work, which means that for multi-objective optimization problems, a scalarization of the objective vectors needs to be performed (Coello Coello and Christiansen, 2007). The λ -formulation, as introduced in subsection 2.8.7 makes provision for this.

This research will not focus in much depth on the development of GAs, but touches on the general and important aspects and concepts of it in the subsequent subsections where the basic operation of a GA is outlined.

2.8.8.2 Encoding

The search space needs to be encoded in a useful way for a GA to function. In the light of structural optimization, each total string (refer to Figure 2.19 in the following section) represents a possible unique solution to the problem. Substrings also need to be easily decoded to evaluate the fitness values.

There are a number of schemes that can be used to provide a genetic representation of solutions in a problem domain (Reynolds, 2009). In this work, abstract binary encoding has been used.

With binary encoding, a string consists of binary 1 or 0 bits and is a relatively flexible scheme. It is easy to represent an integer number in binary notation (i.e. 11010 for 26) or a character in binary notation (i.e. 01000001 for A in 8-bit ASCII). To decode a binary string into the original values, the string is simply parsed according to the number of bits used to represent each value and converted (Reynolds, 2009). Binary encoding has the advantage of simplifying the genetic operations as crossover and mutation need to be developed in such a way that they only handle binary bit strings.

The GA in this work makes use of binary encoding for the following reasons:

- From a programming point of view, implementation is relatively easy and straight forward.
- It is more obvious for the layperson during demonstration.
- It is domain independent and can be quickly adjusted and expanded.

For problems with discrete variables, such as a lattice tower structures from steel sections, a practical method is to construct a table of cross-sectional values and encode a position in the table. For this work it can be assumed that there are 32 sections in Table 2.3 below. Only the first five sections are listed for convenience.

Table 2.3: Section list of the first CHS with parameters

Section number	Designation	A (x 10 ³ mm ²)	I (x 10 ⁶ mm ⁴)	r (mm)
1	32 x 2.0	0.188	0.021299	10.63
2	32 x 2.5	0.232	0.025385	10.47
3	34.1 x 2.0	0.202	0.0261518	11.38
4	34.1 x 2.5	0.248	0.031261	11.22
5	34.1 x 3.0	0.337	0.040013	10.9
⋮	⋮	⋮	⋮	⋮

Consider the first two substrings for illustration purposes in Figure 2.19 in the following section. An example of encoding discrete variables is given below. To encode 32 possibilities in binary, 5 bits per substring are required. To obtain a section in the table for the first element, parse the first five bits as binary and add 1.

Substring 1: 10011

Substring 2: 01001

e.g. substring 1: 10011 = decimal 19 + 1 = Section 20; or substring 2: 01001 = decimal 9 + 1 = Section 10

To obtain the respective values for each section, a look-up pointer for section 20 and 10 respectively is used to obtain the values.

2.8.8.3 Crossover

Crossover, or recombination in biological terms is a process where two chromosomes fuse at a random point and split at the joint to exchange their DNA by swapping the genetic material beyond the joint. This results in two new chromosomes that differ from the parents. In biological terms, this occurs during meiosis, the process by which cells divide (Reynolds, 2009).

GAs implement an analogous operation, where pairs of chromosomes are selected at random from the mating pool and for each pair of strings, a random crossover point is selected. The randomization mechanism applies to simple GAs, while more advanced GAs make use of adaptive mutation and crossover mechanisms. The probability of crossover and mutation are adaptively changed during the genetic process. Some algorithms also make use of multiple crossover points to reduce the positional bias (Coley, 1999). However, in this work only a single point crossover will be used. The mating pool for the next generation is selected from the current generation with the aid of the objective evaluation of the fitness of the members of that generation (Reynolds, 2009).

Consider the individual in Figure 2.19. It consists of many chromosomes (substrings) that are made up of genes (bits). In the example shown in the figure, the strings are each made up of 5 bits. 5 bits are required to encode 32 possibilities in binary. (binary 00000 to 11111 = decimal 0 to 31 = 32 possibilities). This approach was adapted in the lattice type wind turbine support tower in chapter 5.

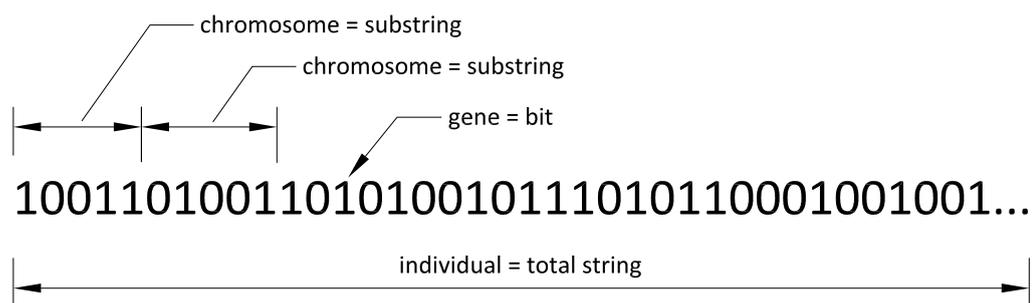


Figure 2.19: Representation of binary encoding and the comparison to genetics.

With reference to Figure 2.19, consider the individual that consists of say 75 bits, and another arbitrary individual of also 75 bits. They need to be crossed over and their length $l = 75$ for both.

Individual 1: 1001101001101...

Individual 2: 0100111010111...

The crossover point is determined by selecting a random number between 1 and $l - 1$, say 3 for this example. The crossover point is denoted by the pipe symbol(|).

Individual 1: 100|1101001101...

Individual 2: 010|0111010111...

The new individuals are as follows:

Individual 12: 100|0111010111...

Individual 21: 010|1101001101...

The new individuals replace their parents in the population and through a combination of crossover and selection as each generation passes, the representative fitness of a population trends upwards as selection favors the fitter individuals. The processes of selection and crossover are not necessarily sufficient to let the algorithm converge to a global or near global optimum and there will come a point where no further improvement in the representative fitness is possible due to a homogeneous population (Reynolds, 2009).

A mechanism needs to be implemented to generate novel genetic features. It can be possible that one or several very fit individuals existed in the initial population and these may predominate the further generations and thus crossover alone becomes ineffective to produce sufficient change. Crossing a pair of similar chromosomes may have very little effect. If both parents are the same, it will certainly have no effect at all during crossover since the identical bits are simply swapped (Reynolds, 2009).

This limitation in a GA is overcome by implementing an additional mechanism for random change, namely mutation. The following subsection will briefly deal with the aspects of mutation and the relevance in GAs.

2.8.8.4 Mutation

The limitation of crossover and selection is overcome by mutation and the process entails alterations to the bit sequence of a substring. In biology, alterations to the gene sequence of an organism are caused by a number of factors such as errors during cell division and external factors such as radiation. The seemingly accidental nature of mutation can provide new genetic traits or qualities that improve the fitness or survival or are lost when no such improvement is achieved (Reynolds, 2009).

Mutation is required in GAs, although crossover and selection effectively recombine existing genetic traits, potentially useful bits are not necessarily protected and might be lost. Apart from useful bits being lost, improvements to novel features cannot be introduced computationally other than by mutation. Reynolds (2009) states that mutation in a GA is generally implemented as a probability of inverting a random binary bit in the total string sequence. With binary encoding a bit can be inverted from 0 to 1 and vice versa.

However, mutation itself does not provide an advantage over a random search, but together

with crossover and selection a genetic diversity is maintained within the mating pool and thus a homogeneous population will be avoided. Reynolds (2009) also states, that mutation plays a secondary role in a GA.

2.8.8.5 Selection

The fitness of each generation is evaluated using a fitness function. In this work, the λ -formulation (or max-min approach) is a meaningful way to represent the fitness for the best trade-off in a multi-objective optimization, as explained before. The optimization is driven by the fitness value, however, a selection process needs to be in place to select the population for the next generation from the current generation.

There are generally two operators for selection in GAs. They fall into two categories, namely stochastic and deterministic selection (Reynolds, 2009).

One of the most commonly used selection methods is fitness proportional selection or Roulette Wheel Selection (RWS). It is a stochastic method which ensures that while fitter individuals are being selected, weaker individuals also stand the chance of being selected. RWS provides a useful analogy for fitness proportional selection. Each selection is independent of others and is consequently analogous to a random throw on a roulette wheel (Reynolds, 2009).

The entire population is sorted in descending order of absolute fitness and the fitness of all individuals in the population is summed to S and a random number r between 0 and S is generated. It is important to note that the fitness values used need to be absolute values. The algorithm iterates through the population, summing the fitness as a running total s . If s is larger than r , the individual number is returned. This is repeated until termination. The procedure is explained using a simple example with a population of six individuals.

The data from Table 2.4 is presented graphically in Figure 2.20. The sum of the fitness values is 2.27 and the relative percentage of the sum for each individual is calculated and shown in the table.

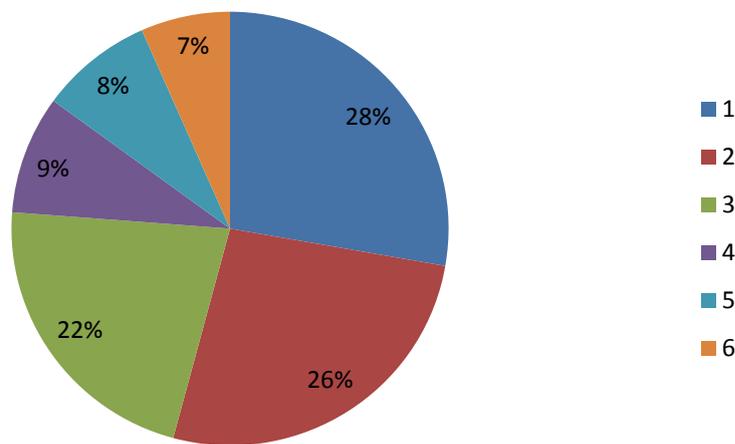
Individual 1 roughly has 28% stake in the total sum of fitness. Similarly, individual 6 roughly has 7% stake, but there is still a chance that it will be selected for reproduction. Generally said, individuals that have a higher fitness will reproduce more often.

An individual is selected by generating a random number between 0 and the total fitness,

Table 2.4: Roulette Wheel Selection example

Individual	Fitness	Cumulative Fitness	% of Total
1	0.63	0.63	27.75
2	0.60	1.23	26.43
3	0.50	1.73	22.03
4	0.20	1.93	8.81
5	0.19	2.12	8.37
6	0.15	2.27	6.61
Total	2.27	-	100.0

Roulette Wheel Selection

**Figure 2.20:** Weighted Roulette Wheel for selection example in Table 2.4.

equal to 2.27 for illustration in this case. The RWS algorithm then iteratively sums the fitness of the population (running total) until the running total is larger than the random number generated. Say, the random number generated was 2.0, the individual selected would be number 5 (the running total at this point is 2.12 and for the first time larger than the random number 2.0).

Deterministic selection methods generally only allow the fittest members of a population to survive. With such methods, only a certain percentage of the fittest individuals are selected, and thus only the fittest members will survive to the next generation and there is no randomness included. The result would be population stagnation and the population might become homogeneous, thus causing premature convergence of the search and no optimum or close to optimum solution will be found.

Elitism is not a selection method, but compliments other GA operators such as crossover and mutation. This ensures that the fittest members of a population survive to the next generation. Further details and reasons are explained in the following subsection.

2.8.8.6 Elitism

During the crossover process, it is possible that the fittest members of the population could be lost. This happens for example when the fittest substring is crossed over with the least fittest one. The consequence is that we have no guarantee that the global optimum or any other fit individuals will survive. Implementing elitism ensures that the optimization is not set back and the fittest members are not lost. Generally, with elitism applied to a GA, mutation may be applied to all members of the population but the fittest ones. Therefore, the elites do not undergo mutation and this has also been chosen as the default option in this study. In other words, the fittest individuals are preserved from one generation to the next (Reynolds, 2009).

Not allowing the mutation of elites can result in (1) that the fittest solution is preserved from harmful mutation, ensuring no loss of fitness and a smooth optimization is the result or (2) that the fittest member is prevented from beneficial mutation and this results that a potential improvement in the total fitness is missed and the optimization does not converge properly.

Reynolds (2009) states that it is not clear how to determine whether to prevent mutation of elites or not. Given the stochastic nature of GAs, the mutation of elites may cause an improvement, a detrimental effect or no effect at all.

2.8.8.7 Fitness function

The fitness function in a GA has the role of providing a measure of quality of a solution, in relation to other members in the population. In the previous sections, the λ -formulation (or min-max procedure) as a representative fitness function has been formulated and described in depth.

The algorithm evaluates the fitness of each member in the population. The fittest solutions in a particular generation are selected to form the genetic basis of the succeeding generation. The fitness function drives the GA and influences the speed, effectiveness and efficiency of the algorithm (Reynolds, 2009).

The majority of the computational demand from a GA is attributed to the fitness function

evaluation. It is important to ensure that the evaluation procedure of the fitness function can be performed relatively fast. A slow fitness function evaluation has a cumulative effect on the speed of the optimization. In order to optimize multiple objectives of a structural system, a measure of relative performance is required. For structural systems such as lattice towers, a structural analysis of the system must be undertaken to evaluate the fitness of an individual in terms of deflection, stress and buckling for example.

A problem specific FEM model needs to be developed for the fitness function evaluation in MATLAB.

2.8.8.8 Population size

Genetic algorithms are implicitly parallel, i.e. they work from a large number of points simultaneously. These points represent a population, and when the population size is increased, then it should follow that the efficacy of the optimization increases as well. The population needs to be an even number due to the genetic operations. Simply said, the population is ranked (sorted according to the corresponding fitness) and the best 50% of the individuals are selected to go forward to the next generation. Crossover is then performed on all individuals as described in subsection 2.8.8.3.

Increasing the population size while the other GA parameters are kept constant, the effect of the population size on the optimization is readily apparent (Reynolds, 2009). A low population size reduces the time it takes for the algorithm, but the algorithm rarely finds a solution.

By increasing the population size, the average duration of the algorithm increases. However, as expected, this is attributed to the larger number of genetic operations that need to be performed. For each generation, the fitness of each member in the population is evaluated by the fitness function (i.e. by means of a FEA). Additionally, operations such as crossover on a larger population also increase the computation time. Therefore, if the population size is increased, the fitness function generally remains the main factor, and the fitness is evaluated a proportionally larger number of times per generation (Reynolds, 2009).

Reynolds (2009) reports that with a population size of 100, 19 out of 20 runs resulted in a solution represented by the fittest individual that had a maximum fitness. This does not necessarily imply that the optimum solution was found. It is therefore possible that a set of consecutive runs results in an optimum in all runs.

2.8.8.9 Termination conditions

As with any optimization problem, the unknown is the optimum solution or set of solutions (Reynolds, 2009). However, it is not possible to simply terminate the search once this optimum has been found. How can one determine whether ‘the optimum’ has been found? This is generally based on some subjective factors. The GA that was developed and investigated in this work makes use of two termination conditions.

- Number of generations: After a number of generations have elapsed, the optimization is terminated. However, no guarantee exists that an optimum solution has been found.
- Convergence: If, after a number of generations, no improvement in the fitness value of the fittest solution is made, the optimization is terminated. Also, no guarantee exists that a solution has been found.

Another termination condition could be a limit on computational time, especially where access to super-computers is charged by time.

2.8.8.10 Penalty methods

Penalty methods are used to constrain the search space in order to obtain feasible solutions. GAs are generally performing best as unconstrained optimization techniques, but by incorporating penalty methods this is changed and the GA can be used to solve highly non-linear, constrained optimization problems.

The constraints split the search space into feasible and infeasible parts and can be seen as a region in the search space where no fitness is allocated (Appelo, 2012). Various penalty methods exist in the literature and have their advantages and disadvantages. In most engineering applications, the problem is constrained and needs to be transformed to an unconstrained problem for the GA. A practical approach seems to assign a zero fitness when a constraint is violated during the search process. In a highly constrained search space, this can cause the algorithm to lose valuable information regarding the fitness, as the GA evaluates the direction of the search based on previous knowledge on the fitness. This can result in premature convergence of the algorithm or in no convergence at all.

A penalty function reduces the fitness function value when a constraint is violated. However, for proper convergence of the search, a penalty function shall not disrupt the equilibrium of exploration and exploitation (Appelo, 2012).

Penalty methods are generally divided in three groups, namely barrier methods, partial penalty methods and global penalty methods. With barrier methods, the necessary condition is that the search is initialized in the feasible region and never enters the infeasible range. Partial penalty methods are formulated as such that the penalty is only applicable close to the feasibility margin. The third group of penalty functions consider the whole search space, and this includes the infeasible region as well.

Several penalty methods have been identified in the literature and some are listed below:

- Death penalty
- Dynamic penalty
- Annealing penalty
- Static penalty
- Adaptive penalty

Several additional penalty approaches exist, but will not be discussed in further detail in this work. For a more detailed summary and formulation concerning penalty methods, refer to the work by Appelo (2012).

In standard form, the optimization penalty formulation is implemented using either the additive or multiplicative approach. The additive penalty approach is given in equation 2.8.32 as follows:

$$f_i(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible} \\ f(\mathbf{x}) + \psi(\mathbf{x}), & \text{otherwise} \end{cases} \quad (2.8.32)$$

where $\psi(\mathbf{x})$ is the penalty factor and $f(\mathbf{x})$ is any objective function. The formulation is applicable to single and multi-objective formulation, where the objective function represents a scalar value that is a function of the design variable vector \mathbf{x} . For the additive case, when no constraints are violated, $\psi(\mathbf{x}) = 0$.

The multiplicative penalty approach is as given in equation 2.8.33.

$$f_i(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible} \\ f(\mathbf{x})\psi(\mathbf{x}), & \text{otherwise} \end{cases} \quad (2.8.33)$$

where $\psi(\mathbf{x})$ is the penalty factor and $f(\mathbf{x})$ is any objective function. The formulation is also applicable to single and multi-objective formulation, where the objective function

represents a scalar value that is a function of the design variable vector \mathbf{x} . For the multiplicative case, when no constraints are violated, $\psi(\mathbf{x}) = 1$.

The additive method is sometimes preferred above the multiplicative method due to improved behavior.

GAs are generally used to solve unconstrained optimization problems as mentioned, and are transformed from a constrained to an unconstrained problem by means of a penalty method. In this work, a quadratic penalty function is used. Some authors recommend different exponents to be used with various motivations respectively. The exponent generally imposes a higher penalty on larger constraint violations.

Constraints can be formulated as soft or hard constraints. A hard (or absolute) constraint generally affects the equilibrium of exploration of the search space. By applying the penalty method, a constraint is ‘softened’. In this work, the penalty for truss element slenderness, yielding and buckling is formulated as a ratio that decreases the fitness of an individual that violates the constraints.

Below is the pseudocode of the penalty approach used in this thesis. The multiplicative approach resulted in satisfactory performance. Take note that the fitness values are analogous to the penalty factors. For further details regarding the fitness determination and calculation, please refer to section 5.4.1 in chapter 5.

```
fitness = min(costFitness, weightFitness, displacementFitness,
perimeterFitness)*bucklingFitness*yieldingFitness*
slendernessFitnessC*slendernessFitnessT
```

2.8.8.11 Test functions

In this section, a brief overview is given on test functions concerning GAs. Test functions are essentially artificial landscapes that can be used to measure and analyze the performance of a GA. The output received is then used to rectify and adjust the internal functionality of an algorithm. This internal functionality is unique to most problems and can therefore be adjusted. Test functions are of less significance to real-world problems (Coley, 1999).

Examples of such test functions include De Jong’s function, Rastrigrin’s function and Griewank’s function to name a few.

2.8.8.12 The search space

The number of possible solutions that describe the so called optimum or best solution that is being searched is quite large. An example might be trying to find the best values for a set of adjustable design variables that, when described as a mathematical model, maximize a certain objective function. Say, these parameters are a and b . The variables a , b and the objective value are plotted on the x -, y - and z -axis respectively. The plot would be the representation of a search space for the problem. However, more complex and real problems with more than two design variables, the situation becomes harder to visualize. Nevertheless, the concept of a search space remains valid even for more than three dimensions. This remains true, as long as there is some measure of distance between solutions and there can be a measure of fitness assigned to the problem (Coley, 1999). Fitter solutions will then occupy the peaks within this fitness landscape and less fit solutions the valleys and intermediate areas.

Fitness landscapes often have very complex topographies, even when considering simple problems. The highest peak (or fitness) is generally associated with the global optimum, and the lesser peaks as the local optima. Broadly said, the goal of most search techniques is the realization and identification of the global optimum (Coley, 1999). For problems such as life-cycle cost optimization of structure, including elements of uncertainty and fuzzyness, it might be adequate to find and identify a large number of highly fit, yet distant and distinct designs.

The motivation for using GAs has not become so apparent yet. The efficiency of a GA in traversing the search space will be demonstrated in the following paragraphs. Considering the well known concepts, namely combinations and permutations. We often use the word ‘combination’ loosely in everyday life. In mathematical terms, the following holds: If the order of, say variables, does not matter, it is a combination. If the order does matter it is a permutation. For each of the two cases, there are generally two further scenarios, namely where repetition is allowed and where repetition is not allowed. In the following example, we have a Permutation with repetition.

Consider we have a truss tower structure with 10 design variables, i.e. the optimum solution (or any possible design) would be a structure with at least 10 different structural steel elements. Say, to evaluate the performance and fitness (stress, buckling and deflection considered) by means of a Finite Element Analysis (FEA) takes 1s. This, however, might be slightly overestimated but is only used for demonstration purposes. We can choose discrete steel sections from a prescribed list of 10 different sections. In structural design

the order does matter, as a different order would result in a new structure. The number of permutations is determined using equation 2.8.34. The time it would take to evaluate all possible designs for the global optimum would be 10^{10} seconds and equals 317 years.

$$P(n, r) = n^r \quad (2.8.34)$$

where n is the number to chose from and r of the n are chosen. Repetition is allowed and the order does matter.

2.9 Life-cycle cost formulations

2.9.1 Introduction

In section 2.7 an initial account and outline on the primary contributing factors and components of LCC during the lifetime of a WECS support structure was given. As previously outlined, a more detailed formulation and explanation of the concept follows. Adeli and Sarma (2006) defines, as mentioned, the Life-Cycle Cost as the total cost of a structure during its lifetime. This includes all the cost components during all phases of its lifetime. Such cost components are affected by a range of factors that vary from project to project. Also note that it is important to distinguish between cost components and cost factors. The factors merely affect the components to a certain extend. It is appropriate to only consider those factors (and also investigate these) about which the designer has control. However, the effects of all will be briefly discussed in this thesis. Taking a simple example: The structural engineer or designer has no direct control over the economic situation and the inflation rate in a country. This factor does, however, influence the LCC of a structure significantly.

2.9.2 Life-cycle cost components

Seven main cost components have been identified by Adeli and Sarma (2006) during the phases of the life of a steel structure. The life-cycle cost of a steel structure can be considered as the sum of these cost components. Recall the schematic representation of the phases of a structure from section 2.6, Figure 2.13. Some of these cost components may be representative in more than one stage as described earlier. The components are as follows:

1. Initial cost, this includes nine different costs including the material. These are
 - 1.1 planning and design cost

- 1.2 material cost
- 1.3 fabrication cost
- 1.4 transportation cost
- 1.5 handling and storage cost
- 1.6 erection cost
- 1.7 tools and machinery cost
- 1.8 site preparation and foundation cost and
- 1.9 initial maintenance cost
2. Operating cost
3. Maintenance cost, such as painting or galvanizing of exposed members of a steel structure.
4. Inspection cost
5. Repair cost
6. Probable failure cost
7. Decommissioning cost

The life-cycle cost of a steel structure can be formulated by equation 2.9.1, (Adeli and Sarma, 2006).

$$\begin{aligned}
 C_{Lifecycle} = & C_{Initial} + \sum \frac{1}{(1+i)^{y_{n1}}} C_{Maintenance} + \sum \frac{1}{(1+i)^{y_{n2}}} C_{Inspection} \\
 & + \sum \frac{1}{(1+i)^{y_{n3}}} C_{Repair} + \sum \frac{1}{(1+i)^{y_{n4}}} C_{Operating} + \sum \frac{1}{(1+i)^{y_{n5}}} C_{Failure} \\
 & + \sum \frac{1}{(1+i)^{y_{n6}}} C_{Decommission}
 \end{aligned} \tag{2.9.1}$$

where $C_{Lifecycle}$, $C_{Initial}$, $C_{Maintenance}$, $C_{Inspection}$, C_{Repair} , $C_{Operating}$, $C_{Failure}$ and $C_{Decommission}$ are the total life-cycle, initial, maintenance, inspection, repair, operating, failure and decommissioning cost of a steel structure, respectively. i is the discount rate of money and the subscripted y variables, y_{n1} , y_{n2} , y_{n3} , y_{n4} , y_{n5} and y_{n6} are the years when each of the costs incur. The summation symbol Σ denotes the summation of all possible cost within the category (Adeli and Sarma, 2006).

The equation 2.9.1 is based on the concept of present worth. The present value is a function of the discount rate i and the period y_n . The discount rate in turn is dependent on the interest rate and inflation rate of the currency. This rate is not constant

and also varies considerably from place to place and will be discussed in more detail in section 2.9.3. However, actual cost data needed in the life-cycle optimization of a structure using the approach as presented in equation 2.9.1 is virtually non-existent in the literature. It is therefore currently not feasible to optimize the life-cycle cost of a structure effectively using equation 2.9.1, as information is scant and is based on insufficient statistical data and assumptions (Adeli and Sarma, 2006). Alternative methods are required to perform life-cycle cost optimization. In chapter eight of Adeli and Sarma (2006), the authors discuss some useful methods based on other techniques that can be implemented. Wilson *et al.* (1997) discuss a decision support tool or system for analyzing the life-cycle cost of bridge deck designs. Some research presented includes probabilistic methods for life-cycle cost optimization of structures such as by Frangopol *et al.* (1997). However, also the use of probabilistic methods is limited due to statistical data often being unattainable (Adeli and Sarma, 2006). The authors present a model for LCC optimization based on fuzzy logic with the objective to formalize the life-cycle design process with active input from the designer. The example in chapter eight of their work considers only two type of cost that incur over the life of an exposed steel structure, namely initial cost and maintenance cost in the form of painting. This method can be expanded to incorporate multiple objectives and will be discussed later.

In a section on fuzzy discrete multi-criteria optimization, a description on different life-cycle cost optimization models is given. It entails formalizing the life-cycle design process by considering a fuzzy discrete multi-criteria life-cycle cost optimization model. Firstly, the cost components listed above will be discussed in the subsequent subsections.

2.9.2.1 Initial cost

In formulating and determining the cost of a structure at any stage during its lifetime, material costs and fabrication costs are taken into account during the initial phases of a structure. Farkas and Jarmai (2008) formulates a cost function that includes the cost of material, assembly, welding, surface preparation, cutting, painting, edge grinding and shell forming for the initial phase of a steel structure. The fabrication sequence is taken into account as well. The authors also refer to the work as mentioned in section 2.7. The cost function formulation is considered to be quite realistic and applicable even in different countries assuming that the technology applied is standard. For example, the cost function per unit length weld by standard GMAW-M (Gas metal arc weld with mixed gases) welding makes provision for a variety of factors. Examples are labor cost (skill), gas price, weld size, plate thickness and some more. Other items such as transportation cost, civil and road works cost and installation and erection cost have been adopted from Sagol

(2010).

Planning and design cost: Planning and design cost, seen from a structural engineering point of view, depend greatly on the project as such. For WECS, this cost depends on a variety of different factors. Probably, the designer will not be wishing to optimize this cost for a client, but rather maximize profit output for the expertise and consulting work that will be delivered to a client. A regulatory body, such as the institution of Consulting Engineers of South Africa (CESA), have established guidelines and predetermined rates that may be charged to a client based on the total cost of works. Therefore, further investigation is excluded from the scope of this research and the planning and design cost component will be assumed to make up a fraction of the other, total initial cost. Please refer to Table 2.5 which contains actual design and planning cost calculation information for civil and structural engineering projects specifically. Take note that the cost of the works in the table are an example for a certain category and are valid for 2011 and exclude any levies or other similar cost.

Table 2.5: CESA Basic consulting fees for an example category in 2011

Cost of the Works (R excluding VAT)	Percentage for the basic fee (%)
For the first R 800,000.00	12.5%
For the next R 2,600,000.00	10.0%
For the next R 9,700,000.00	7.5%
For the balance	5.5%

Material cost: Farkas and Jarmai (2008) discuss the cost of material in their work on *Cost Optimization of Metal Structures*. For non-alloy structural steel types, the material cost determination is probably one of the most simple forms. Equation 2.9.2 represents the general case for material cost. When several different materials are used it is possible to use different material unit cost factors. Farkas and Jarmai (2008) use the alphabetic letter K (or k) to represent the cost function or a cost factor. In this work, the cost function is represented by a C or c for consistency.

The cost of materials such as pins, bolts and similar parts, should also be included in the formulation for total initial material cost. $C_b = \sum_i c_{bi}$, where c_{bi} is the i^{th} bolt, pin or part item cost.

$$C_m = \rho \sum_i c_{mi} V_i + C_b \quad (2.9.2)$$

where C_m is the total material cost, c_m is the corresponding material unit volume cost, V is the volume of the structure or structural element and ρ is the material density.

However, different commercially available structural steel sections have different prices. The values vary for different section types and are not necessarily directly related to mass. This of course, only applies to commercially available sections. The material cost is calculated in a slightly different way, where the section-specific unit length cost is considered. Equation 2.9.3 can be used to calculate the material cost when differences in section prices have a significant impact on the total material cost. For example: A cold formed Circular Hollow Section (CHS) and an equal leg angle section of similar cross-sectional area have the same mass per unit length but differ significantly in cost. For this case, equation 2.9.3 is proposed as an alternative.

$$C_m = \sum_i c_{mi} L_i + C_b \quad (2.9.3)$$

where C_m is the total material cost, c_{mi} is the i^{th} section unit length cost, and L_i is the length of the i^{th} section.

Fabrication cost in general: The fabrication cost of a certain structure is complex to quantify and is often a function of multiple factors. However, Farkas and Jarmai (2008) formulate an approximation to calculate the production time required to perform a task. The fabrication related unit cost is generally dependent on labor cost, consumables and technology and varies from place to place. For the fabrication process, tools and machinery cost are treated separately and will be discussed under the subsequent section below.

It can be assumed that the fabrication cost factor is constant for each manufacturer. It is also possible to apply different factors in equation 2.9.4 simultaneously.

$$C_f = c_f \sum_i T_i \quad (2.9.4)$$

where C_f is the total fabrication cost, c_f fabrication cost per unit time, and T_i is the production time of the i^{th} fabrication sequence. The production times are discussed in the following paragraphs and may have various subscripts related to the processes involved.

During the fabrication process of structural members, connections or built-up sections, the majority of time is owed to preparation, assembly, tacking, time of welding, deslagging, chipping and painting (Farkas and Jarmai, 2008).

A formula to approximate the times related to surface preparation, assembly and tacking

can be calculated with equation 2.9.5.

$$T_{w1} = C_1 \Theta_{dw} \sqrt{\kappa \rho V} \quad (2.9.5)$$

where C_1 is a parameter that depends on the welding technology, Θ_{dw} is the difficulty factor, κ is the number of structural elements to be assembled. The difficulty factor expresses the complexity of the structure. The complexity varies with each structure. Examples are differences between planar and spatial structures. Stiffened shell sections and lattice towers are generally considered spatial structure types and girders or trusses can be considered planar during the manufacturing process. Farkas and Jarmai (2008) recommend the range of values for Θ_{dw} to be between 1.0 and 4.0.

Furthermore, real welding time can be calculated using equation 2.9.6.

$$T_{w2} = \sum_i C_{2i} a_{wi}^2 L_{wi} \quad (2.9.6)$$

where T_{w2} is the total, real welding time, C_{2i} is a constant for the welding technology applied, a_{wi} is the leg size of the weld and L_{wi} is the weld length. C_{2i} not only accounts for the welding technology, but also for vertical, overhead and normal welding in downhand position.

Farkas and Jarmai (2008) also refers to additional fabrication actions that need to be accounted for. Examples include changing the electrode, chipping and deslagging. The time of such actions can be calculated by equation 2.9.7.

$$T_{w3} = \sqrt{\Theta_{dw}} \sum_i C_{3i} a_{wi}^2 L_{wi} \quad (2.9.7)$$

where T_{w3} is the total additional fabrication time, C_{3i} is a constant for the different welding technology. However, Farkas and Jarmai (2008) refer to the work of Ott and Hubka (1985), where it is proposed that $C_3 = (0.20 \text{ to } 0.40)C_2$. On average this gives $C_3 = 0.30C_2$. a_{wi} and L_{wi} are as defined for equation 2.9.6. Furthermore it is stated, that the factor $\sqrt{\Theta_{dw}}$ can be neglected for equation 2.9.7 and only needs to be considered for determining T_{w1} .

Therefore, the modified equation for T_{w3} when neglecting $\sqrt{\Theta_{dw}}$ becomes:

$$T_{w3} = 0.3 \sum_i C_{2i} a_{wi}^2 L_{wi} \quad (2.9.8)$$

Farkas and Jarmai (2008) provide a generalized form for calculating T_{w2} and T_{w3} based on actual experimental and theoretical investigations obtained from manufacturers all over

the world and from software calculations. In the generalized formula, the power to a_w is n and is in some cases close to the value of 2. In general form, the calculation of times for real welding time and additional fabrication actions is given by equation 2.9.9.

$$T_{w2} + T_{w3} = 1.3 \sum_i C_{2i} a_{wi}^n L_{wi} \quad (2.9.9)$$

The time calculation for arc-spot welding and post-welding treatments are also provided in Farkas and Jarmai (2008). Equation 2.9.10 and 2.9.11 can be used to determine the time required for arc-spot welding and post-welding treatments respectively.

$$T_{w4} = n_S T_S \quad (2.9.10)$$

$$T_{PWT} = T_0 L_t \quad (2.9.11)$$

where n_S is the number of spots, T_S is the time of welding per spot and electrode transfer from one spot to the next. T_S depends on the degree of automation, welding equipment and technology used. For determining the total post-welding treatment (PWT) time, T_0 is the specific time for PWT as given in Appendix B.

Farkas and Jarmai (2008) also refer to total time calculations for the flattening of plates, but this item will be omitted here. The forming process of plates into shell elements will be discussed here, as well as the associated time calculation formulation. Tubular wind turbine towers are fabricated from plates that are formed into shells and are welded together into tower sections. This allows for easier and more efficient handling and transport. Forming of plates to shell elements greatly depends on the shape of the shell. However, only non-prismatic cylindrical sections will be considered here. This is applicable to the tubular, non-prismatic monopole tower types. Each segment of the shell structure will have a different curvature and therefore also a different radius R_i . Thus the total time is calculated using equation 2.9.12, which is approximated as the sum of the single segment production times.

$$T_{F0} = \sum_i \Theta e^{\mu_i} \quad (2.9.12)$$

where

$$\mu_i = 6.8582513 - 4.527217t_i^{-0.5} + 0.009541996(2R_i)^{0.5} \quad (2.9.13)$$

For conical sections t_i is the shell segment thickness, R_i is the radius. The fabrication difficulty value is constant for this configuration and is taken as $\Theta = 3.0$. It is valid for the ranges of $R_i \leq 1500 \text{ mm}$ and $t_i \leq 30 \text{ mm}$.

Surface preparation time forms part of the fabrication process and ensures adequate conditions for proper welds and painting. Surface preparation includes techniques such as

sanding and cleaning away cutting or mill debris. Equation 2.9.14 yields the time required for surface preparation and is a function of the surface area A_s .

$$T_{SP} = \Theta_{ds} a_{sp} A_s \quad (2.9.14)$$

where a_{sp} is a constant time value per unit area and can be determined experimentally. Farkas and Jarmai (2008) consider an average value for $a_{sp} = 3 \times 10^6 \text{ min/mm}^2$. Θ_{ds} is the difficulty parameter.

Plate cutting and edge grinding times can be determined using equation 2.9.15. Different technologies commercially used include Acetylene, Stabilized gas mix and Propane cutting at normal or high speeds (Farkas and Jarmai, 2008). The total time is function of the plate or element thickness t_i and cutting length L_{ci} .

$$T_{CP} = \sum_i C_{CPi} t_i^n L_{ci} \quad (2.9.15)$$

where n is an exponent that comes from curve fitting. The time values are included in Appendix B.

Farkas and Jarmai (2008) also refer to the hand cutting and machine grinding times of tubular strut ends. The following formula is proposed in calculating the total time required.

$$T_{CG} = \Theta_{dc} \sum_i \frac{2\pi d_i}{\sin \phi} (4.54 + 0.4229 t_i^2) \quad (2.9.16)$$

where d_i is the diameter of the strut in m , the thickness t_i is in mm and ϕ is the angle between the two members (for example the chord and brace in a truss structure) to be connected. Θ_{dc} is the difficulty factor and is taken as $\Theta_{dc} = 3.0$.

Drilling of bolt holes depends on the diameter of the bolt, the steel grade and the plate thickness. The cost of drilling holes can be calculated by multiplying the cost per hole C_{di} with the number of bolts n_i , (Farkas and Jarmai, 2008).

$$C_d = \sum_i C_{di} n_i \quad (2.9.17)$$

Hot-dip galvanizing (HDG) is a process where the fabricated steel is dipped into a molten zinc pool in a bath or kettle at an elevated temperature of about $450 \text{ }^\circ\text{C}$. The steel metallurgically reacts with the molten zinc in an electrochemical process and forms a zinc-alloy coating that provides excellent corrosion resistance to the steel at its surface. The pricing for HDG is generally related to the mass of the steel to be covered or galvanized.

The cost of HDG can therefore be calculated using equation 2.9.18.

$$C_{HDG} = c_{HDG} \sum_i^N m_i \quad (2.9.18)$$

where C_{HDG} is the cost for hot-dip galvanizing N structural steel members, each of mass m_i at a unit cost of c_{HDG} per unit mass. The designer should consider certain geometric limitations owed to the HDG process regarding the handling, lifting and the possible basin or kettle sizes and aspect ratios.

The total fabrication cost is given by equation 2.9.4 above and can therefore be modified to include the two additional terms related to the other processes not mentioned by Farkas and Jarmai (2008), namely hot-dip galvanizing cost and cost related to the drilling of holes.

Equation 2.9.4 then becomes:

$$C_f = c_f \sum_i T_i + C_{HDG} + C_d \quad (2.9.19)$$

Transportation cost: The support tower sections are generally, apart from the wind turbine blades, the most large and bulky components of a wind turbine system and may account for the biggest portion of transport cost. The tower structure is also in many cases the heaviest component and contributes to approximately 66% of the system's material mass, (AWEA, 2011). Other components are lighter and mostly smaller in size.

Transportation is generally done by road, by ship or by rail. Costs are generally calculated by either mass, volume or distance covered. Different billing methods exist for different logistics companies and road freight, for example, is generally billed per freight distance. This means, the cost is only calculated for a certain distance over which the goods are conveyed.

Abnormal load transport permit cost may also incurred and need to be carefully deliberated.

Bridge and road clearance limits are not directly related to cost, but certain cost can arise indirectly resulting from waiting time and detours at obstacles. The minimum vertical road clearance under existing bridge structures in South Africa is 4.9 m, (SANRAL).

Equation 2.9.20 has been obtained from Sagol (2010) and gives an approximation on transportation cost to the actual construction site of the structure based on the rated

power P_{rated} of the wind turbine. The expression has been obtained from curve fitting and is a quadratic function. Parameters A , B and C need to be determined from actual data. The cost values change over time and depend on the mode of transport, fuel cost and labor cost and this formulation is not very practical.

$$C_{transport} = A P_{rated}^2 + B P_{rated} + C \quad (2.9.20)$$

However, a simplified approach related to cost per unit distance is proposed and could be applicable to South African conditions as well.

$$C_{transport} = \sum_i c_{ti} d_i \quad (2.9.21)$$

where c_{ti} is the rate per unit distance [R/km] and d_i is the i_{th} transport distance.

Handling and storage cost: The handling and storage cost of fabricated structural steel elements of a structure are assumed to be included in the respective cost factors in the fabrication process. Other types of handling cost that relate to transport of any component shall be treated accordingly and further details fall outside the scope of this work. Such types of cost are important and need to be considered in any project, but the designer does not have direct control over these cost.

Erection cost: The erection cost are often directly related to the structural system. However, in some instances the designer can develop a system and mechanism that can aid in the erection process, for example a hinge mechanism that allows the assembly of the structure on the ground and the final erection is performed towards completion.

It can be assumed that the assembling cost and erection cost are included in the fabrication cost factors. Erection cost can also depend on the equipment required to erect a certain type of structure. Examples of such equipment include the following: scaffolding, ladders, tools, cranes, trucks, temporary supports and structures, administration cost and site establishment, water, electricity, fuel for machinery, tools and labor cost to name a few.

Tools and machinery cost: Tool and machinery cost are similarly, as mentioned under handling and storage and erection cost, assumed to be included in the fabrication cost of the structure. Additional constraints also determine and influence these cost, such as construction worker and tool operation skills or expert availability in certain scenarios.

Site preparation and foundation cost: The costs related to site preparation and the foundation depend on the geographical location of the project and also the type of structure. A detailed investigation is excluded from the scope of this work. However, a

significant aspect was identified concerning the foundations and the so-called foundation ‘footprint’ of lattice type towers compared to tubular, monopole towers. Lattice type towers have a wider overall base, but their foundations occupy less volume and are generally smaller. There is, however also a difference related to associated construction and material cost.

Initial maintenance cost: An expression for the initial painting time is given in equation 2.9.22.

$$T_{cp} = \sum_i C_{CPi} t_i^n L_{ci} \quad (2.9.22)$$

If galvanizing is considered, this factor becomes zero and galvanizing cost will be included in the fabrication section of the structure as discussed earlier. Also refer to section 2.9.3.7, where a decision tree illustrates how design decisions affect total life-cycle cost.

2.9.2.2 Operating cost

According to Adeli and Sarma (2006), the geographic location also influences the operating cost of a structure. For buildings this may be in the form of heating and air-conditioning for example. However, for structures such as wind turbine support towers, the operating cost is treated as maintenance or periodic maintenance cost.

The integration of structural health monitoring (SHM) into life-cycle management strategies enables wind turbine manufacturers, operators and owners to precisely schedule maintenance work and inspections. Such a structural health monitoring system installation contributes towards the operating cost but can reduce unnecessary maintenance and inspection cost for a wind turbine support structure. A study by the Department of Civil and Environmental Engineering at Stanford University, USA, together with Ruhr-University Bochum, Germany, presented a paper on an integrated approach towards life-cycle management of wind turbines. It was found that such reductions in maintenance cost and inspection cost are possible but also other performance related measurements proved to be advantageous.

Non-structure related operation costs include the project management and administration costs, the project insurance costs, public liability insurance costs, safety costs, cost of monitoring the project, and the business rates and taxes. For the structure to fulfil its fundamental task in supporting the nacelle and rotor, no additional operating cost are associated with it under normal operation. However, maintenance cost have been identified to have a major impact on the total LCC cost and are discussed in the subsequent and corresponding section(s).

2.9.2.3 Maintenance cost

The general approach is described by considering only one representative type of cost incurred over the life of the structure, namely preventative maintenance in the form of periodic painting of an exposed steel structure to avoid corrosion or rusting. Thus, in this context, maintenance is considered in the form of painting only. The only control the design engineer will have is a reduction in the overall exposed surface area of the steel structure (total perimeter length). Furthermore, maintenance and repair cost of structures in a difficult terrain are often expensive in terms of access and availability of skilled and unskilled labor. Unscheduled reactive maintenance can be more costly to conduct than scheduled maintenance.

2.9.2.4 Inspection cost

According to Sørensen (2007), it is widely accepted that cost optimal structural system designs, inspection plans and maintenance strategies are determined on the basis of preposterior analysis from classical decision theory. In general, parameters defining the decision variables are divided into variables related to the structural design and variables related to inspection plans and maintenance actions. Inspection costs are dependent on the inspection parameters such as number of inspections, the time intervals between inspections and the inspection qualities or methods (Sørensen, 2007). Rational decisions on design, inspection and maintenance are obtained using preposterior analysis from Bayesian decision analysis according to Sørensen (2007).

2.9.2.5 Repair cost

The repair cost are event specific and have a large uncertainty involved. The repair cost is the replacement value of the structure or the costs that incur to restore the structure to at least the condition before a certain damage occurrence.

2.9.2.6 Probable failure cost

In reliability-based design the loads and resistances of a structure are considered as random variables, and the safety is related to a probability of exceeding the structural capacity given a certain load and its parameters. The approach towards reliability-based design includes the uncertainties in both the loads and the resistances. The total LCC is affected indirectly by the probable failure cost. This is done by adding the product of the probability of failure and the total expected failure cost for a certain failure mode. However, according to Adeli and Sarma (2006), the expected failure cost and the probability of failure cannot be calculated with any measure of certainty due to insufficient statistical

data. Sometimes, the cost of failure is assumed to be known and taken as proportional to the initial material cost of the structure. The criterion of minimum expected failure cost is equivalent to minimization of weight with an allowable probability of failure.

Adeli and Sarma (2006) state that the probable failure cost is also a function of the geographic location. As an example, the probability of failure of a structure in an active earthquake zone increases significantly.

2.9.2.7 Decommissioning cost

Steel generally is 100% recyclable and life-cycle assessment (LCA) studies on WECS have shown that there is an energy payback time of three to five months of operation (World Steel Association, 2012). Thus, this is the time it takes for the energy savings of a project to equal the total energy expenditure since inception. Steel is infinitely recyclable and has a limited environmental impact and the recovery of the material at the end of its useful life additionally helps to recover cost due to the material being returned to the steelmaking process (World Steel Association, 2012).

Near the end-of-life phase, alternative solutions can be implemented to extend the life of wind farms. For example, 116 wind turbines in Germany with a total rated capacity of 56 MW were dismantled and replaced by 80 WECS with a total, rated capacity of 183 MW. More wind turbines are expected to reach re-powering age in the near future (World Steel Association, 2012).

Seeing that steel is used in most of the key components of wind turbines also allows the wind energy industry to meet technical requirements of turbines and climate change demands at the same time.

In general, from a climate change and sustainability viewpoint it is important to take into account the life-cycle of products. Generally, permit applications concerning development plans and future site usage require future management strategies of the site or area. From this it is apparent that reuse and recycling are of significance (World Steel Association, 2012).

2.9.3 Life-cycle cost factors

The eleven main factors that influence the life-cycle cost components are listed and described below. The approach is adopted from Adeli and Sarma (2006) and a breakdown of these factors and the allocation is presented in a tree diagram in Figure 2.21.

- (a) Cost of the rolled sections used for initial construction of the structure
- (b) Number of different section types used in the structure
- (c) Weight of rolled sections used in the structure
- (d) Perimeter of rolled sections in the structure
- (e) Number of connections
- (f) Geographic location of the project site
- (g) Maintenance policy of the structure
- (h) Anticipated life of the structure
- (i) Discount rate of the currency
- (j) Use of the structure
- (k) Importance of the structure

The summarized form is shown in the tree diagram and the components are specified in more detail in Figure 2.21. Take note that some of the factors are present under more than one cost component.

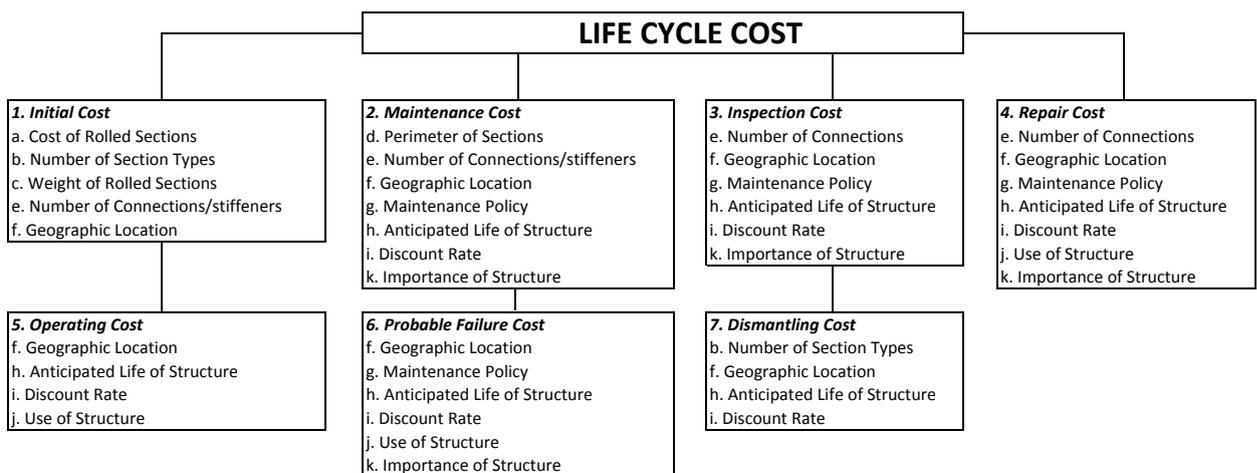


Figure 2.21: Breakdown of the various cost components (1-7) and factors (a-k)

Adeli and Sarma (2006) consider only six factors where the designer will have control when considering the LCC optimization model of a steel structure. These six factors are cost

of the rolled sections, the number of different section types, weight of the rolled sections, perimeter of the rolled sections, number of connections and the geographic location of the structure. The six factors, as listed above, will be outlined and discussed in the respective sections that follow. The criterion of minimizing the number of section types can conflict with the minimum material cost and minimum material weight criteria. Also, the latter two criteria often conform and may not compete in the optimization step which makes their inclusion trivial. For structural steel, the cost is sometimes a direct function of the weight. This can be solved by introducing a weighting coefficient that assigns a relative importance to each criterion. The sum of the weighting coefficients is one (Adeli and Sarma, 2006).

Adeli and Sarma (2006) define four fuzzy functions that are used in the discrete multi-objective (the authors refer to the term multi-criteria) cost optimization model. These will also be used and referred to in chapter 3. These are the material cost of the structure $\tilde{C}(\tilde{y})$, the number of different section types $\tilde{T}(\tilde{y})$, the weight of the structure $\tilde{W}(\tilde{y})$ and the total perimeter of the sections of the structure $\tilde{P}(\tilde{y})$. No membership functions are defined for the factors relating to the number of connections and the geographic location of a structure.

The fuzzy functions and variables are identified by the tilde sign (\sim) on the top. The objective of this four-criteria optimization model is to minimize the functions $\tilde{C}(\tilde{y})$, $\tilde{T}(\tilde{y})$, $\tilde{W}(\tilde{y})$ and $\tilde{P}(\tilde{y})$. Three of these fuzzy functions can be expressed explicitly in terms of the fuzzy variables \tilde{y} ; they are explained and given below.

Most of the subsequent sections are discussed in Adeli and Sarma (2006) on pages 104 - 114.

2.9.3.1 Cost of rolled sections

The fuzzy cost function for the cost of the rolled sections can be expressed as in equation 2.9.23.

$$\tilde{C}(\tilde{y}) = \sum_{i=1}^{N_t} l_i c_i \tilde{y}_{C_i}, \quad \tilde{y}_{C_i} \in S_{C_i} \quad (2.9.23)$$

Here, S_{C_i} is the fuzzy set of discrete standard shapes for the i^{th} design variable x_i that corresponds to the minimum cost objective. N_t is the number of initial section types, l_i is the total length of the members linked to the design variable x_i . c_i is the unit volume cost of the discrete shape or profile. The cross-sectional area of the discrete standard shape is the fuzzy variable \tilde{y}_{C_i} and the maximum membership function corresponds to the

minimum cost criterion and belongs to the fuzzy set S_{C_i} .

A linear membership function is defined by Adeli and Sarma (2006) and is in the form as given in expression 2.8.29. It is graphically shown in Figure 2.22. The equation for the membership function for minimum cost is given as:

$$\mu_{C_j} = 1 - \frac{c_j - c_{j \min}}{c_{j \max} - c_{j \min}}, \quad j = 1, 2, \dots, N_i \quad (2.9.24)$$

The material cost per unit length for the j^{th} candidate shape is denoted by c_j and the minimum and maximum values are denoted by $c_{j \min}$ and $c_{j \max}$ respectively.

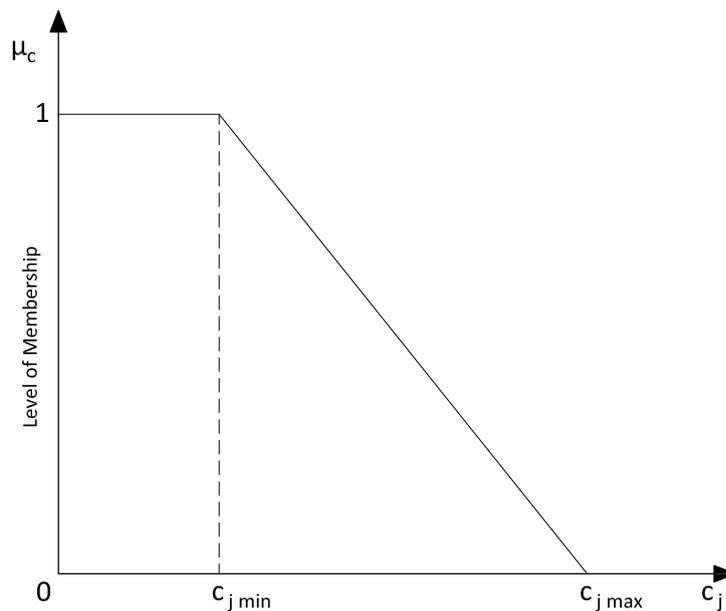


Figure 2.22: Membership function for the minimum cost of material

2.9.3.2 Number of different section types

For the design of high-rise building structures and tower structures, the individual members are generally designed from the bottom to the top (Adeli and Sarma, 2006). This objective entails the minimization of the number of different section types without affecting the material cost and weight of a structure. The approach by Adeli and Sarma (2006) is as follows: A record of the already selected standard shapes is kept such that the same shape is selected again for subsequent members to be designed. A scheme used by Adeli and Sarma (2006) to assign membership values to the candidate sections for minimizing

the number of section types is given in the following form:

$$\mu_{T_j} = \begin{cases} \beta\mu_{C_j}, & \text{if the section is never selected} \\ \gamma\beta\mu_{C_j}, & \text{if the section is selected earlier and } \beta\mu_{C_j} \leq \frac{1.0}{\gamma} \\ 1.0, & \text{if the section is selected earlier and } \beta\mu_{C_j} > \frac{1.0}{\gamma} \end{cases} \quad (2.9.25)$$

where $\gamma = 1 + e^{n_f}/\alpha$ and n_f is the number of times a section has been used earlier. β is a factor that is used to penalize a shape if it has not been used before and α is the scaling factor. The penalty and scaling factor are chosen in such a way, such that the value of the exponential expression for γ becomes useful for choosing a candidate shape only if the same shape has been used at least four times for another structural member already (Adeli and Sarma, 2006). Values from parametric studies for the penalty and scaling factor have been found by Adeli and Sarma (2006) to be suitable. The suitable values are $\beta = 0.2$ and $\alpha = 10$.

2.9.3.3 Weight of rolled sections

Similar to the fuzzy cost function above, the fuzzy function for weight can be expressed as in equation 2.9.26.

$$\tilde{W}(\tilde{y}) = \rho \sum_{i=1}^{N_t} l_i \tilde{y}_{W_i}, \quad \tilde{y}_{W_i} \in S_{W_i} \quad (2.9.26)$$

Here, S_{W_i} is the fuzzy set of discrete standard shapes for the i^{th} design variable x_i that corresponds to the minimum weight objective. N_t and l_i are as described for the fuzzy cost function above and ρ is the density per unit volume. The cross-sectional area of the discrete standard shape is the fuzzy variable \tilde{y}_{W_i} and the maximum membership function corresponds to the minimum weight criterion and belongs to the fuzzy set S_{W_i} .

Similar to the cost, a linear membership function is defined as for the weight and corresponds to the formulation as in expression 2.8.29. A graphical illustration is given in Figure 2.23. The equation for the membership function for minimum weight is given as:

$$\mu_{W_j} = 1 - \frac{y_j - y_{j \min}}{y_{j \max} - y_{j \min}}, \quad j = 1, 2, \dots, N_i \quad (2.9.27)$$

The minimum and maximum values are denoted by $y_{j \min}$ and $y_{j \max}$ respectively.

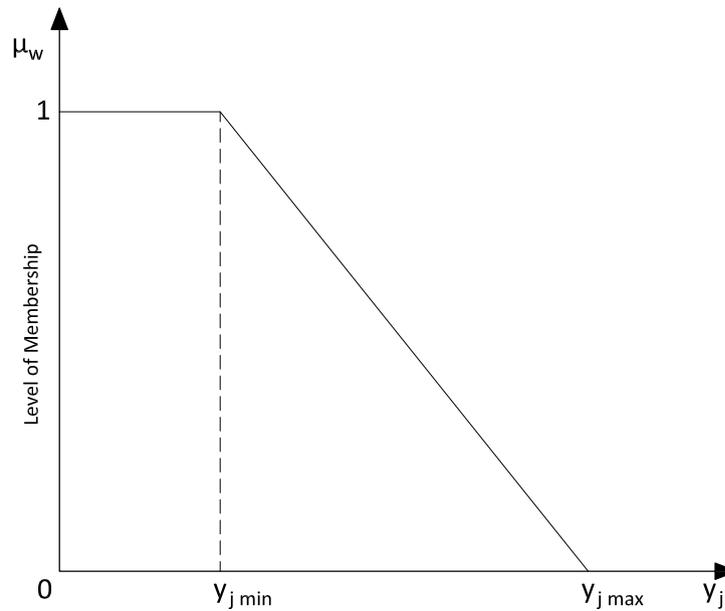


Figure 2.23: Membership function for the minimum material weight

2.9.3.4 Perimeter of rolled sections

The fuzzy function for the perimeter of the rolled sections can be expressed as in equation 2.9.28.

$$\tilde{P}(\tilde{y}) = \sum_{i=1}^{N_t} l_i \tilde{y}_{p_i}, \quad \tilde{y}_{p_i} \in S_{p_i} \quad (2.9.28)$$

Similarly, S_{p_i} is the fuzzy set of discrete standard shapes for the i^{th} design variable x_i that corresponds to the minimum cost objective. N_t is the number of initial section types, l_i is the total length of the members linked to the design variable x_i . The cross-sectional perimeter of the discrete standard shape is the fuzzy variable \tilde{y}_{p_i} and the maximum membership function corresponds to the minimum cost criterion and belongs to the fuzzy set S_{p_i} .

A linear membership function is defined by Adeli and Sarma (2006) and is in the form as given in expression 2.8.29. A graphical illustration is given in Figure 2.24. The equation for the membership function for minimum cost is given as:

$$\mu_{P_j} = 1 - \frac{p_j - p_{j \min}}{p_{j \max} - p_{j \min}}, \quad j = 1, 2, \dots, N_i \quad (2.9.29)$$

The minimum and maximum values are denoted by $p_{j \min}$ and $p_{j \max}$ respectively.

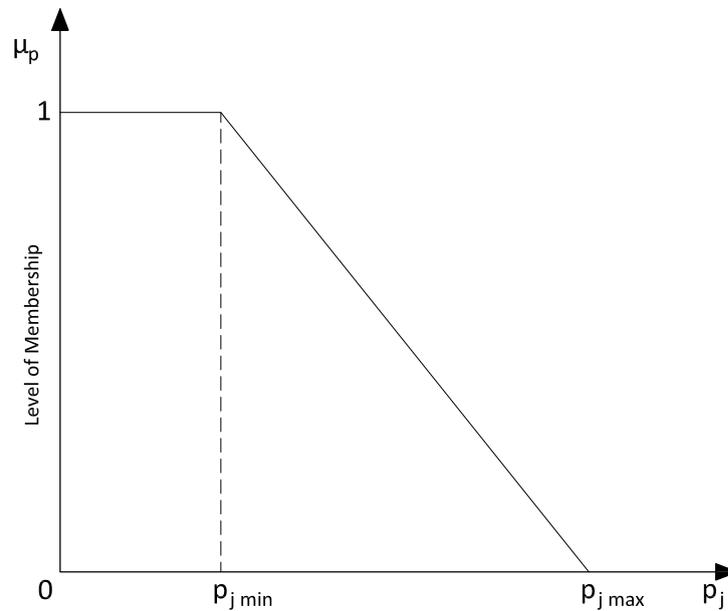


Figure 2.24: Membership function for the minimum perimeter

2.9.3.5 Number of connections

The number of connections in a structure adversely affect the cost of fabrication, labor, connection materials and erection (Adeli and Sarma, 2006). During the planning stage and configuration design of a structure, an engineer or architect decides upon the number and location of connections. The designer has little or no control over the number of connections once a configuration has been selected (Adeli and Sarma, 2006). Such a connection configuration could be an entire optimization problem on its own and is excluded from the scope of this research study.

The designer can, however, limit the number of different types of connections and thereby reduce the fabrication cost. It is important to note that this factor is dependent on the number of section types used in the structure. It seems plausible and practical, that the optimum for this objective is achieved when the number of section types is reduced to an optimum. However, no direct proof exists that supports this proposition.

2.9.3.6 Geographic location of the project

The geographic location of a structure influences LCC cost directly and indirectly. Adeli and Sarma (2006) states that this factor is one of the five major influential ones that are discussed in their work. Direct influences may be paired with aspects such as variations in wind action, seismic hazards and similar extreme actions. The geographic location can affect the transportation cost, handling and storage cost, erection cost, fabrication cost

and the site preparation and foundation cost. At a location where surplus skilled and unskilled labor force are available results in a less costly situation compared to a location where labor force is expensive and scarce. Transportation cost of rolled sections and construction material from steel mills to the fabrication work shop and ultimately to the construction site are also affected by the geographic location. Often, as mentioned under the transportation cost components, limitations to the cargo exist in terms of weight and size.

2.9.3.7 Maintenance policy of the structure

A poor maintenance policy often leads to early failure and a conservative maintenance policy on the contrary, may result in excessive cost (Adeli and Sarma, 2006). Operating and maintenance (O&M) cost of wind turbines form a significant component of the relative production of electricity from wind. In some cases this could be as high 25% for onshore systems, implying that corrective and preventative maintenance strategies and requirements are increasing. The O&M cost are very important for the profitability of a WECS. Initial investment costs can be determined and estimated well and are also known beforehand to a great extent, whereas O&M cost are accumulated and considered over the entire life-cycle and therefore are very uncertain (Sørensen, 2007). Current strategies are mainly based on experiences from other types of electricity generation utilities and Sørensen (2007) includes the following in the paper *Structural reliability aspects in design of wind turbines*.

A rational way of deciding on actions related to O&M is to use a risk-based approach based on pre-posterior Bayesian decision theory (Sørensen, 2007). See the decision tree in Figure 2.25 that has been obtained from the work of Sørensen (2007). The decision tree for O&M decisions has the following nodes and properties:

- Initial design: decision on design variables, constraints and objectives. The design variables are denoted \mathbf{z} .
- Sequential decisions or observations on: inspection and condition monitoring \mathbf{i} , observation of result of inspection/monitoring, represented by a random outcome \mathbf{S} and the decision on maintenance/repair that is based on observed inspection/monitoring result and represented by the decision rule $d(\mathbf{S})$.
- Observation of random realization of an event (e.g. extreme events) and is represented by random variables \mathbf{X}

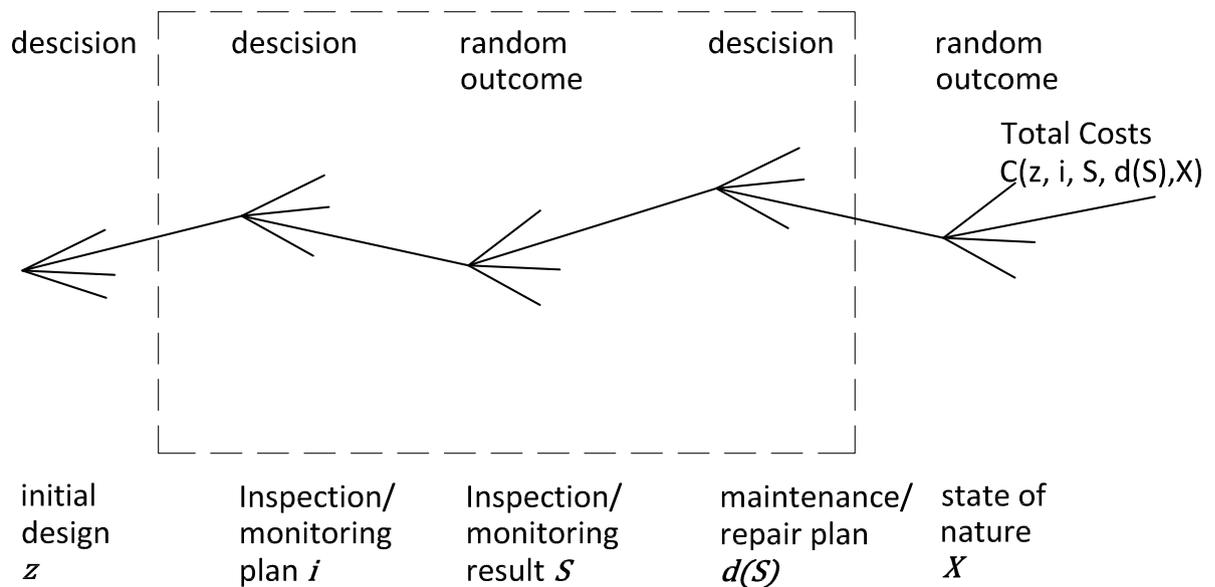


Figure 2.25: Decision tree diagram for O&M decisions (Sørensen, 2007).

2.9.3.8 Anticipated life of the structure

The design life (or economic life) of most structures is often in the range of 30 to 50 years. Wind turbines are currently designed for an economic lifetime of 20 years. However, actions are considered to have a return period of 50 years. Generally, as one expects, this differentiates from the actual life expectancy or anticipated life time of a structure. Often this value is much higher and it may be 60 to 85 years and even more (Adeli and Sarma, 2006).

The anticipated life of a structure plays an important role in the life-cycle design, but some factors may influence the anticipated life of a structure. Examples are natural hazards (extreme events), man-made catastrophes and obsolescence. Anticipated life time as well as repair and maintenance also depend on the materials used in the structure. Corrosion can occur in steel structures, cracks can form in concrete elements due to shrinkage strains and frequent weather changes.

2.9.3.9 Discount rate of the currency

The discount rate of money considerably affects the LCC of most assets, as it is related to the inflation rate or time value of money. One notices the time value of money by the ever increasing prices one has to pay for a certain amount of goods. Generally said, in most countries an inflation rate prevails, while some can also have a deflation. These values are often determined on a monthly basis and annual averages are calculated. The

inflation rate values change in a random fashion with time and are unpredictable in the long run.

A common measure for inflation is referred to as the Consumer Price Index (CPI or CPIX). CPI values of South Africa's economy for several years are shown in Figure 2.26. The data has been obtained online from www.inflation.eu and is shown graphically below. However,

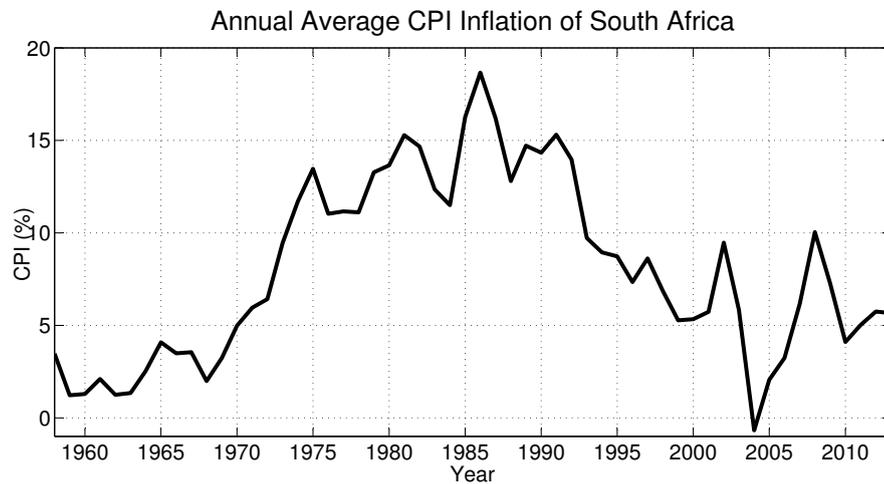


Figure 2.26: Annual average CPI history of South Africa. [Online: www.inflation.eu]

deflation can also occur and the discount rate becomes a negative value and the money actually increases in value or buying power over time. Further details are excluded and fall beyond the scope of this work. The discount rate of the currency in a country is a factor that carries the potential to affect the LCC assessment and optimization process. Adeli and Sarma (2006) exclude this factor in their LCC optimization, because the designer has no control over this factor, although it needs to be carefully deliberated.

2.9.3.10 Use of the structure

The selected design situation for a structure has to be estimated and selected as sufficiently severe and realistic as possible. This ensures that conditions can be reasonably accounted for during the use of the structure (Jacobsohn, 2009). SANS 10160 - Part 1:2011 distinguishes between unidentified and identified design situations, where identified design situations may be assessed and analyzed using classical structural analysis. For unidentified actions robustness requirements are included in the code. Design for robustness requirements can allow for a reduction in damage that a structure may experience under accidental actions. The risk that is associated with such an event is generally treated as a probability of occurrence and resulting consequences.

Accidental actions or loads, usually of a short duration, may occur on average less than once during the design life time of a structure (Jacobsohn, 2009). Generally, such actions have much larger magnitudes than the conventional variable and permanent loads considered for the SLS and ULS. Variable actions occur more frequently during the life time of a structure. Figure 2.27 is a diagrammatic representation of an accidental action over the time of a structure and Figure 2.28 a representation of variable or imposed loads over the life time of a structure. Permanent or self-weight actions are generally constant over time. Accidental loads may have a magnitude that can cause partial or total collapse of the structure. Such a load may be caused by an extreme event such as an explosion, impact, earthquake or wind and can cause the loss of an element (column, beam or connection). The behavior of the structure in such an event determines its level of robustness and is a function of the use of the structure.

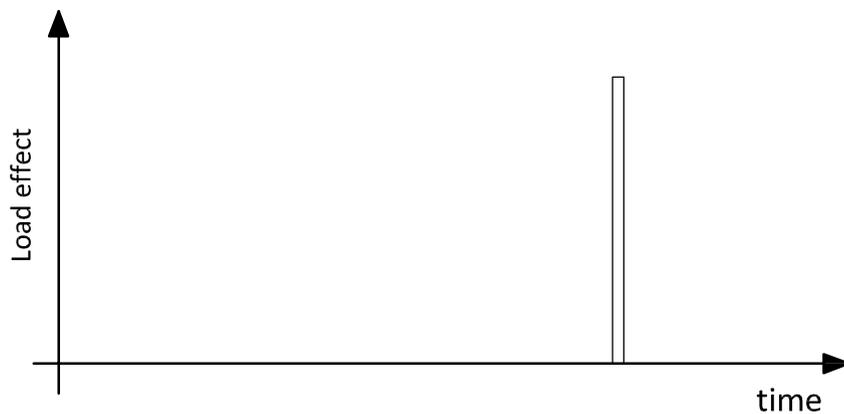


Figure 2.27: Schematic representation of accidental actions vs. time on a structure (Vrouwen-velder *et al.*, 2005).

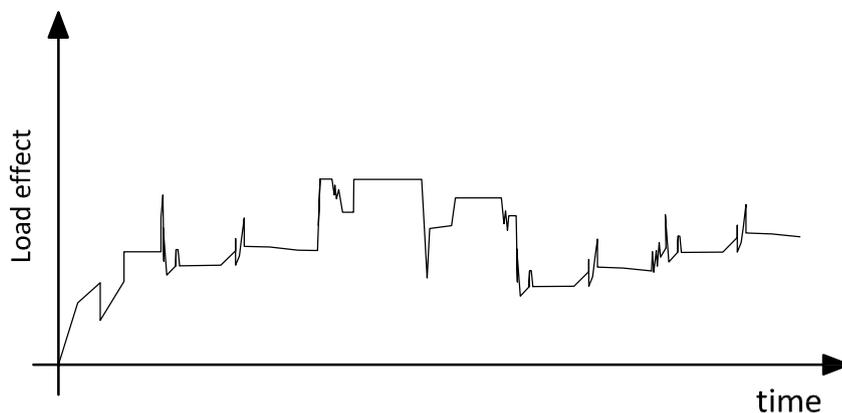


Figure 2.28: Schematic representation of variable actions vs. time on a structure (Vrouwen-velder *et al.*, 2005).

2.9.3.11 Importance of the structure

In terms of structural life-cycle design, the strength of a structural element or system (reliability) of a structure is dependent on time and also on the importance of a structure. SANS 10160 - Part 1:2011 (partially based on the Eurocode) as well as the *Guidelines for Design of Wind Turbines* by DNV/Riso (2002) make both use of reliability classification for a range of different structures. The level of reliability is referred to as the independent variable in the probability function, β_t . It is also sometimes referred to as the safety index. The probability function ϕ used in SANS 10160 - Part 1:2011 is the cumulative normal distribution function. Therefore the notional probability of failure at a certain reliability level or even at the target reliability is calculated with equation 2.9.30.

$$p_f = \phi(-\beta_t) \quad (2.9.30)$$

Table A.1 in SANS 10160 - Part 1:2011 provides various values of levels of reliability for the four different reliability classes, RC1 to RC4. The reliability classes are a function of the structure's use and the consequence associated with its failure mode. The designer should ensure the correct category choice prior to the design. For this project, the design of a typical structure would fall into reliability class 2, RC2. This corresponds to a reliability index value of $\beta_t = 3.0$. Refer to Figure 2.29, which is a pictorial representation of the reliability level over time and the effects of repair or maintenance. In this work, the first four factors are considered in the multi-objective optimization model for the design of a lattice truss type tower of a WECS.

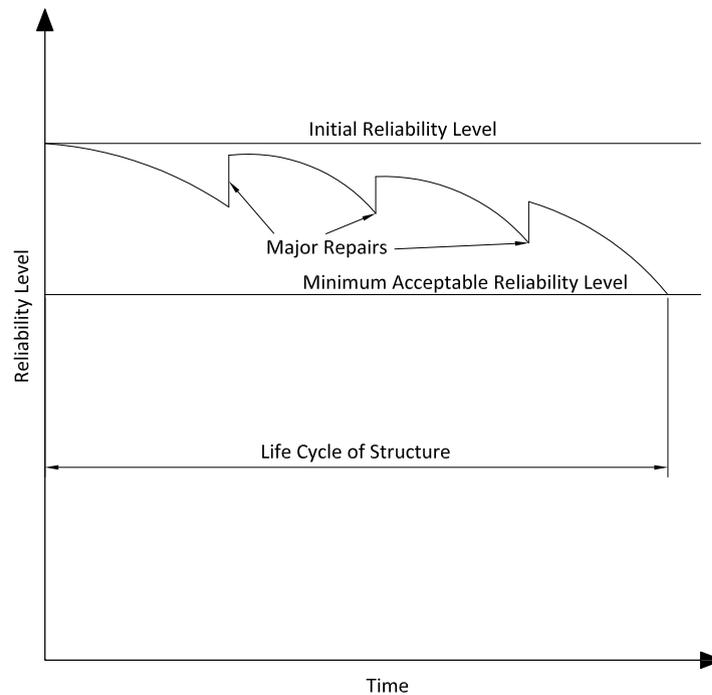


Figure 2.29: Schematic representation of the reliability of a structure with passing of time (Adeli and Sarma, 2006).

2.10 Chapter conclusion

The literature review chapter briefly discusses the concept of Life-Cycle Cost Management (LCCM). LCCM concerns a set of predefined tasks and processes that focus on the continuous reduction of the total cost during all stages of the life-cycle of a structure. Thereafter, the literature study includes an outline and brief discussions of various research approaches from various researchers in this field. From the initial literature consulted, the author can draw the following main conclusions:

- The initial phases of the life-cycle of a structure form a crucial part in the evolution of the future LCC of any structure.
- The LCC optimization of a WECS support tower structure or any steel structure can be effectively formulated as a multi-objective problem.
- The λ -formulation is a suitable objective aggregating approach for decision making situations and to determine the best trade-off solution.
- Carrying out a LCC analysis or assessment by determining and calculating the present value cost seems impractical. A simplified approach in the paper by Sarma

and Adeli (2002) considers the selection of sections that satisfy these criteria: (1) Select discrete commercially available sections with the lowest cost, (2) select discrete commercially available sections with the lightest weight, (3) select minimum number of different types of discrete commercially available sections, and (4) select discrete commercially available sections with minimum total perimeter length. The third objective is not applicable when a structure consists of only one section type. The nodal deflection can be treated as an objective as well.

- A non-gradient based search technique is required due to the non-linear nature of structural optimization problems. A GA satisfies this criterion and is relatively easy to implement and understand. Implementing a GA also satisfies the four requirements listed in subsection 2.8.1.
- Only certain factors affect the LCC components of structures. Six factors are identified by Adeli and Sarma (2006) and are listed below for convenience, although a detailed discussion and description has been included in the literature review.
 - Cost of the rolled sections
 - Number of different section types
 - Weight of the rolled sections
 - Perimeter of the rolled sections
 - Number of connections
 - Geographic location of the structure site

The investigation of the fuzzy discrete multi-objective optimization is a suitable method as discussed in the previous two chapters.

Anderson (2001) also states, that the λ -formulation (min-max formulation) can be used together with other techniques in multi-objective optimization. Such an approach was chosen, where the min-max formulation and fuzzy logic theory are combined.

This approach is non-deterministic; thus the actual cost values are not calculated in the fuzzy discrete multi criteria LCC optimization and this may not be in favor of the engineering practice but rather results in a relative trade-off solution. Manufacturing limitations cause design variables to be discrete, rather than continuous, as all cross-sections belong to a certain discrete set as provided by manufacturers.

Chapter 3

Research Methodology

3.1 Introduction

This chapter's purpose is to present the reader with the overall research methodology followed for the life-cycle cost optimization study that is described in this thesis and applied to a benchmark problem and a case study. Furthermore, the chapter gives a broad outline on the data and information required for investigation and analysis, as well as the limitations to this research.

The methodology described will focus on the attempt to obtain a near optimum tower structure given a selection of multiple objectives and the flexible and adjustable input of the design engineer or decision maker. For the purpose of this research, the same weight is assigned to all objectives considered (by default). This, by definition represents a near optimum solution that is Pareto optimal. Generating solutions from various different weights should result in points that all lie on a Pareto front. If a solution that converged properly using the methodology as set out in this work is not exactly Pareto optimal, then it should at least be in the feasible range in the proximity of the Pareto front.

The design engineer also chooses the upper and lower boundaries for the objectives as well as the constraints that are acceptable for the design, as the optimization is very sensitive to these parameters. Further details are included in the following chapters.

3.2 Analysis methodology

In this work, a genetic algorithm written and programmed in MATLAB is adopted from Reynolds (2009) and is developed to aid in possible decision making processes during

the structural design process of primarily lattice tower type wind turbine support towers. This technique is relatively universal and can be applied to a variety of structures and also in different fields of study such as operations research, scheduling and other decision making problems. There are various optimization approaches or methods that have proven to be suitable. Often, no optimization method is necessarily superior to another, all have their advantages and disadvantages.

For this work, the multi-objective formulation as discussed by Adeli and Sarma (2006) is adopted. The authors identified a range of factors and cost components that strongly affect and contribute to the total LCC of a steel structure. Furthermore, Adeli and Sarma (2006) also accentuate six of these factors that adversely affect the LCC of a steel structure, over which the designer has control.

In other words, an optimization technique needs to be developed and investigated, where the input from the designer has influence on a reduced total LCC.

After a critical evaluation of Adeli and Sarma (2006), four factors are defined that need to be considered in a multi-objective LCC optimization, namely weight $\tilde{W}(\tilde{y})$, cost $\tilde{C}(\tilde{y})$, perimeter $\tilde{P}(\tilde{y})$ and deflection $\tilde{d}(\tilde{y})$. Some of the objectives are commensurable and others are incommensurable. For example, the deflection objective ‘competes’ against the weight objective, while cost and weight are objectives that generally do not ‘compete’ against each other.

The designer could assign a preferred weight w_i to each objective. Adeli and Sarma (2006) propose linear fuzzy membership functions to rate the degree of membership from total rejection 0 to total acceptance 1 along a set domain. A GA with a penalty approach is used to find the best compromise solution as discussed in the previous chapter and the development tool used is MATLAB. The core of the algorithm is adopted from the work by Reynolds (2009). The unconstrained optimization problem is converted to a constrained problem by means of penalty methods.

Maalawi (2010) defines a scheme for the general optimization approach. Note that this is a general approach for any optimization method.

3.3 Analysis overview

The optimization approach in this work is divided into four parts. The first approach concerns the graphical optimization due to the nature of the design variables and objec-

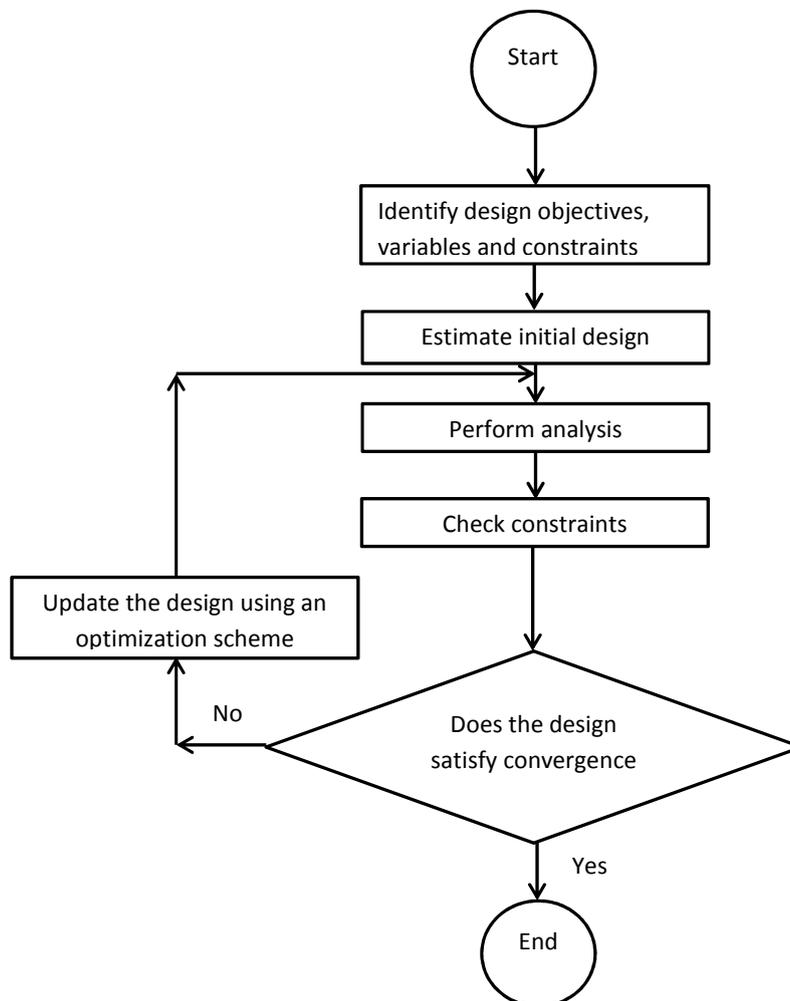


Figure 3.1: The general scheme of an optimization approach to design (Maalawi, 2010).

tives of a monopole type tower segment. All the required code as *m*-files and functions are included in Appendix A. Thereafter follows an illustrative example performed in MS-Excel to gain a better understanding of the formulation and methodology and two further multi-objective LCC optimization studies are performed using MATLAB together with a GA that is adopted from the work by Reynolds (2009). Below is the outline of the following chapters.

1. Chapter 4: Investigation into a simplified monopole tower section optimization using

the guidelines from DNV/RISØ (2002). Only a loaded segment or section of a tower structure is considered and no multi-objective LCC using a GA is done due to time constraints and the optimization challenges identified concerning lattice type towers.

2. Chapter 5: Investigation of a four bar space truss to verify the results obtained in the paper by Kelesoglu and Ulker (2005). The system has only a single design variable (one group of elements). A MS Excel spreadsheet is used to develop a simple Finite Element Analysis (FEA) tool together with the data solver add-in. The first part of the investigation deals with two objectives only, namely weight and displacement, as in the paper by Kelesoglu and Ulker (2005) for both continuous and discrete variables. The second part of the investigation considers all four objectives, weight, cost, displacement and perimeter of the rolled sections. No design constraints from any design code are used. The only constraints used in this part are the upper and lower boundary values of the design variable, the cross-sectional area.
3. Chapter 5: Investigation and verification of a benchmark problem using MATLAB, a three dimensional 25 bar space truss structure with eight (8) design variables. The work by Appelo (2012) is used as a reference for geometric properties, material properties, nodal loads and also for result comparison. A prescribed set of discrete sections is used and the problem is treated as a single objective optimization type. The objective considered is weight of the structure and the algorithm performance is tested.
4. Chapter 5: Investigation and verification of a utility scale type WECS tower structure using MATLAB, a three dimensional 212 bar lattice tower structure with 15 design variables. For the first part, the same set of discrete sections is used as in the previous 25 bar problem and it is a multi-objective optimization type. The four objectives considered are weight, cost, deflection and perimeter. The investigation is repeated for a different set of discrete rolled circular hollow sections (CHS) and equal leg angle sections obtained from the tables in The Red Book by the South African Institute for Steel Construction (SAISC). The constraints imposed on the circular hollow sections and equal leg angle sections are obtained from SANS 10162 - Part 1:2005. These include slenderness limits, axial tensile capacity (yielding) and axial compressive capacity (critical buckling).

3.4 Limitations of the research

As with any research project a few limitations exist. The limitations include assumptions and simplifications that are generally determined a-priori and recommendations and further investigations are based on the work that has been studied. In this work, a range of design constraints are implemented in the multi-objective sizing optimization model. The constraints are only based on the the following from the SANS 10162 - Part 1:2005 design code: slenderness limits, axial tensile (yielding) capacity and axial compressive capacity (critical buckling).

Additional to the above criteria, this project only covers the wind turbine types that will be used for electrical energy generation i.e. where wind power is converted from the kinetic energy of the wind to electrical energy. No off-shore located wind turbines, which are subject to wave loading and other extreme loads, will be considered.

Only truss elements with translational nodal degrees of freedom are considered in this work. For the lattice tower, the model can be relatively easily expanded and changed to beam elements that also have rotational degrees of freedom, if required. The upper and lower constraints of the objectives, such as maximum and minimum deflection depend on the design variables, i.e. the maximum or minimum section size. The values need to be evaluated analytically using the largest and smallest possible section size for all elements with the aid of a FEA, as there is a chance that the GA could possibly select the largest section for all elements, so the smallest deflection possible corresponds to this scenario respectively. The case study was performed using CHS and equal leg angle sections that are at least class 3 compression members according to clause 11.2, Table 3 in SANS 10162 - Part 1:2005. Additional constraints that are not considered in this study could be related to the frequency of vibration for a variety of modes, or even the fatigue life of welded members and connections. The structural element forces are determined by the FEM solver, and in this investigation the own weight of the structure was neglected. This certainly affects the real-life behavior of the structure significantly and care must be taken when interpreting the results.

Another assumption that has been made in this study concerns the connections at the nodes of the space structures studied. The connections are assumed to be designed such that they have at least the tensile or compressive capacity of the member away from the connection.

However, for the 212-bar lattice tower, arbitrary nodal loads have been used in the case

study. This simplification was chosen as the input is flexible and can be easily adjusted as required.

For this study, only one load combination was considered during the optimization. The inclusion of other load cases complicates the problem and this requires further investigation. With WECS supporting structures, symmetry is a definite advantage when it comes to optimization, because the loading is symmetric as well. A different load distribution in the truss elements from a different load combination would result in a different result.

The GA adopted and developed in this work is functional from the MATLAB command line, and a graphical user interface (GUI) was not built. The advantage of a GUI is to make it a more user friendly front-end and to easily alter GA input parameters for sensitivity studies and explorative analysis.

Chapter 4

Monopole Type Tower LCC Optimization

4.1 Introduction

Towers for large wind turbines are either tubular steel towers, lattice towers or concrete towers, or a mixed combination thereof. Approximately 90% of all wind turbine towers are tubular steel towers (Karpát, 2013). Karpát (2013) presents a study in which a cost optimization on a tubular wind turbine monopole tower is performed by means of a Particle Swarm Optimization (PSO) algorithm with four design variables. The objective function used by Karpát (2013) is formulated as a direct cost function using the cost formulations from Farkas and Jarmai (2008) as described in Chapter 2 in this thesis. The objective function is expressed as a single objective, multi-variable function that is constructed as a cost function from the different variables.

In chapter 5 fuzzy discrete multi-objective LCC optimization is applied as a useful tool towards decision making in the initial design stages of a lattice type support tower. The approach for a monopole type tower is similar. However, the author decided to make use of graphical optimization techniques as there are only two design variables identified for segments of such towers, namely the segment diameter d and thickness t . In contrast, the lattice tower case study in chapter 5 has 15 design variables.

The following sections deal with the determination and definition of the objectives, variables and constraints that are applicable to a LCC optimization model of a monopole type tower. Take note, that a similar cost optimization case study on a 1 MW wind turbine tower was studied by Uys *et al.* (2006). In this work, more emphasis is placed on the LCC

optimization approach as proposed by Adeli and Sarma (2006) for lattice type towers as an alternative to the widely used mono-pole type towers.

4.2 Modeling objectives

The objectives that are required in the multi-objective optimization of a monopole type tower are as discussed in the literature chapter and are comprehensively demonstrated and described in the subsequent chapter. In the work by Karpat (2013), the objective is expressed as a total cost function with the variables as listed in section 4.3.

In this work the LCC optimization is simplified using a graphical optimization approach for an arbitrary, unstiffened wind turbine segment.

4.3 Modeling variables

In the work by Karpat (2013), the following design variables are considered for the numerical optimization of a tubular tower segment. For this study, the ring stiffener variables h_r and t_r are not included because the segment is assumed to be unstiffened. A graphical optimization approach, as described in Venkatamaran (2002) is followed and therefore only two design variables are desirable. With more design variables, the search space dimensions increase and visualization thereof becomes difficult. This approach was chosen to get a better and improved understanding of the graphical search space with only two variables.

- Height of the ring stiffener, h_r .
- Thickness of the ring stiffener, t_r .
- Wall thickness of the shell, t .
- Diameter of the shell, d .

The definition of design variables and parameters is of utmost importance in formulating an optimization model. The variables used for the graphical optimization are reduced, based on a simplification. As outlined in the introduction, only a segment of a monopole tower is considered at a time and the variables for such a segment are the diameter d and the thickness t . Outside and inside diameter, d_{outer} and d_{inner} respectively, are alternatives for expressing these two variables since $t = \frac{1}{2}(d_{outer} - d_{inner})$.

Each segment of a tower is certainly subject to different load magnitudes and therefore an optimum tower solution would have many different segments. An additional, possible variable in the optimization of tower segments is the number of segments to be used, the segment height or the number of ring stiffeners.

Figure 4.1 shows a typical section through a monopole type tower. Also note that any transverse stiffeners, access ladders and equipment are not included in the optimization approach. The representation simply shows a simplified structural configuration of the tower section.

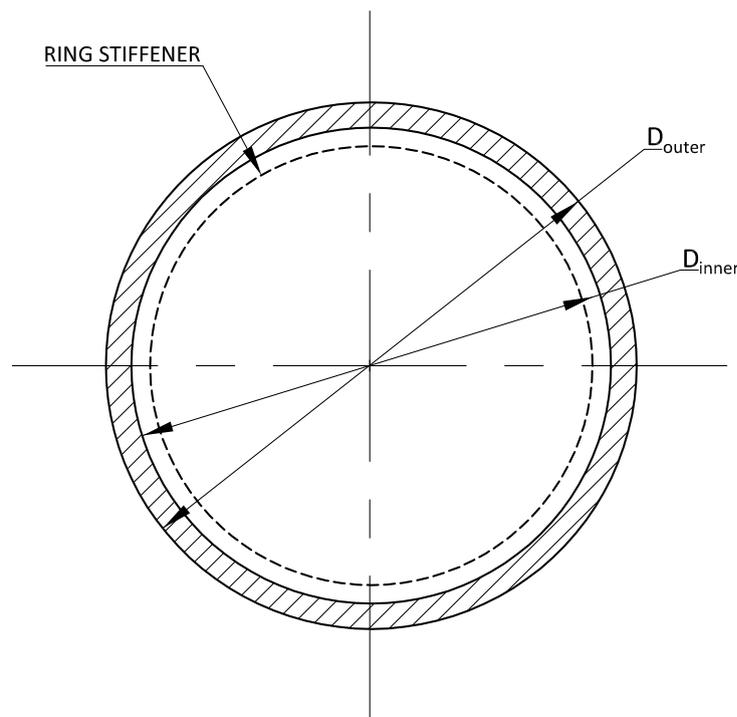


Figure 4.1: Section view of a monopole tower segment

4.4 Modeling constraints

As with the lattice tower, problems in engineering and especially in structural engineering are generally heavily constrained and non-linear. The capacity of the shell structure of a tubular monopole tower can govern the design. Specific constraints are identified and those that need to be considered are listed below:

- Width-to-diameter ratio limits.

- Slenderness ratio limits on tower segments.
- Global buckling resistance of tower sections.
- Local buckling resistance of tower sections.
- Tower deflections due to code limitations (SANS 10162-1 and IEC 61400).

According to DNV/RISØ (2002), the buckling strength of the tubular tower usually governs the tower design as far as the shell thickness is concerned. The analysis for the buckling strength of the tower can be done by using the approach described in Annex D of the Danish standard DS449 together with DS412 or other recognized standards. The method as presented in the Danish standard for tubular tower sections is implemented. The design calculations from the Danish standards are obtained from DNV/RISØ (2002) and no direct reference will be made to DS449 and DS412.

The calculation procedure is described in the following paragraphs and the graphical optimization is plotted in Figure 4.2.

The design stresses due to the axial force C_u and bending moment M_u are given in equations 4.4.1 and 4.4.2 respectively.

$$f_{ad} = \frac{C_u}{2\pi R t} \quad (4.4.1)$$

$$f_{bd} = \frac{M_u}{\pi R^2 t} \quad (4.4.2)$$

where R is the radius of the shell section and t is the shell thickness. A reduction factor ε , a function of the shell radius and thickness, is calculated as given in equation 4.4.5.

$$\varepsilon_a = \frac{0.83}{\sqrt{1 + 0.01 \frac{R}{t}}} \quad (4.4.3)$$

$$\varepsilon_b = 0.1887 + 0.8113\varepsilon_a \quad (4.4.4)$$

$$\varepsilon = \frac{\varepsilon_a f_{ad} + \varepsilon_b f_{bd}}{f_{ad} + f_{bd}} \quad (4.4.5)$$

The critical compressive stress is calculated according to principles from elasticity theory, as shown in equation 4.4.6.

$$f_{el} = \frac{E}{\frac{R}{t} \sqrt{3(1 - \nu^2)}} \quad (4.4.6)$$

For local buckling, the relative slenderness ratio is calculated as follows:

$$\lambda_a = \sqrt{\frac{f_y}{\varepsilon f_{el}}} \quad (4.4.7)$$

According to DNV/RISØ (2002) and the Danish codes, the critical compressive stress is a function of the relative slenderness ratio for local buckling. For the range where $\lambda_a \leq 0.3$, the critical compressive stress f_{cr} is calculated using equation 4.4.8.

$$f_{cr} = f_y \quad (4.4.8)$$

For the range where $0.3 \leq \lambda_a \leq 1.0$, the critical compressive stress f_{cr} is expressed as in equation 4.4.9

$$f_{cr} = (1.5 - 0.913\sqrt{\lambda_a})f_y \quad (4.4.9)$$

When the tower height, $H \leq 1.42R\sqrt{\frac{R}{t}}$, then the critical compressive stress f_{cr} is determined using equation 4.4.8.

The check for global buckling stability also needs to be verified. DNV/RISØ (2002) includes the approach for the global buckling capacity calculations. The Euler force for a cantilever beam is given by expression 4.4.10

$$C_{el} = \frac{\pi^2 EI}{L_e^2} = \frac{\pi^2 EI}{(KL)^2} \quad (4.4.10)$$

For the tubular tower structure, with $KL = L_e = 2H$ and $I = \pi R^3 t$, the Euler force becomes:

$$C_{el} = \frac{\pi^2 E \pi R^3 t}{4H^2} \quad (4.4.11)$$

For global stability, the relative slenderness ratio is determined using equation 4.4.12.

$$\lambda_r = \sqrt{\frac{f_{cr}}{\left(\frac{C_{el}}{2\pi R t}\right)}} \quad (4.4.12)$$

Additionally, DNV/RISØ (2002) makes provision for geometrical imperfection. The guideline document distinguishes between cold-formed welded towers and hot-rolled welded

towers. For cold formed welded towers, the equivalent geometrical imperfection e can be calculated using equation 4.4.13 and for hot-rolled welded towers e is calculated using equation 4.4.14. When $\lambda_r \leq 0.2$, then $e = 0$.

$$e = 0.49(\lambda_r - 0.2)k \quad (4.4.13)$$

$$e = 0.34(\lambda_r - 0.2)k \quad (4.4.14)$$

If $e > 0.002H$, an additional increment Δe is to be added to e . The core radius k of a tube is determined by $k = \frac{R}{2}$. The following inequality interaction equation needs to be fulfilled:

$$\frac{C_u}{2\pi R t} + \frac{C_{el}}{C_{el} - C_u} \frac{M_u + C_u e}{\pi R^2 t} < f_{cr} \quad (4.4.15)$$

Other constraints, such as flange end connections also form part of the design process and need to be accounted for. However, the design of such components fall beyond the scope of this work and design details can be found in DNV/RISØ (2002).

4.5 Monopole tower optimization results

A possible solution for the optimization problem with two variables, being inner and outer section diameter, can be presented as depicted in Figure 4.2. Loads are obtained from an existing design for a 24 m tower (confidential data). The graphical representation is not ideal and the optimum point in this figure is described as the best solution in the feasible range, marked with the square pointer. The feasible range is distinguished from the infeasible range by both the equality and inequality constraints. The red lines in Figure 4.2 and Figure 4.3 represent the constraint boundaries as previously described. Figure 4.3 shows a ‘zoomed’ view into the figure, and the optimum is observed as the solution ($d_{outer} = X_1 = 1.740$ m, $d_{inner} = X_2 = 1.645$ m) where the minimum objective value (green contours) is realized. In this example, it is taken as the total material cost which is minimized. The blue lines are the 10% values of the constraints (red lines) and are used to find the feasible region easier. The green contours represent the material cost and are used to get a better picture of the feasible and infeasible regions.

An actual cost estimate, c taken as R/kg , is used to plot the contours. c is taken as $R16.00/kg$ (approximate average cost of structural steel in February 2013), which is a reasonable estimate for initial material cost.

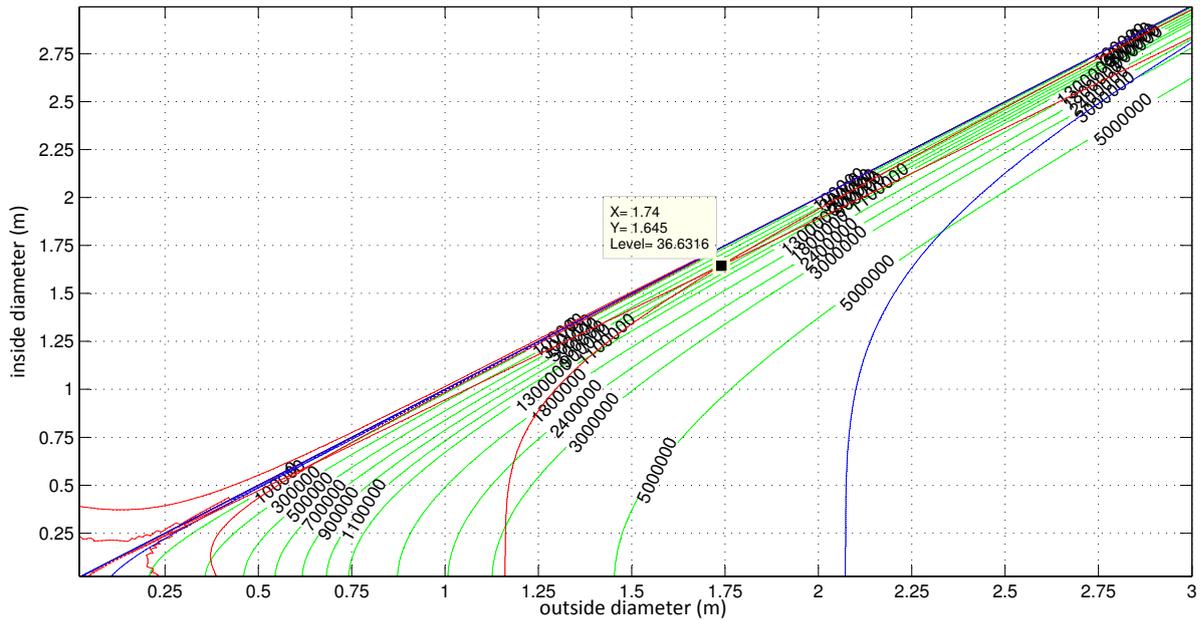


Figure 4.2: Graphical optimization result of a bottom monopole tower section for $M_u = 1698.2$ kNm , $C_u = 64.5$ kN and $V_u = 79.6$ kN

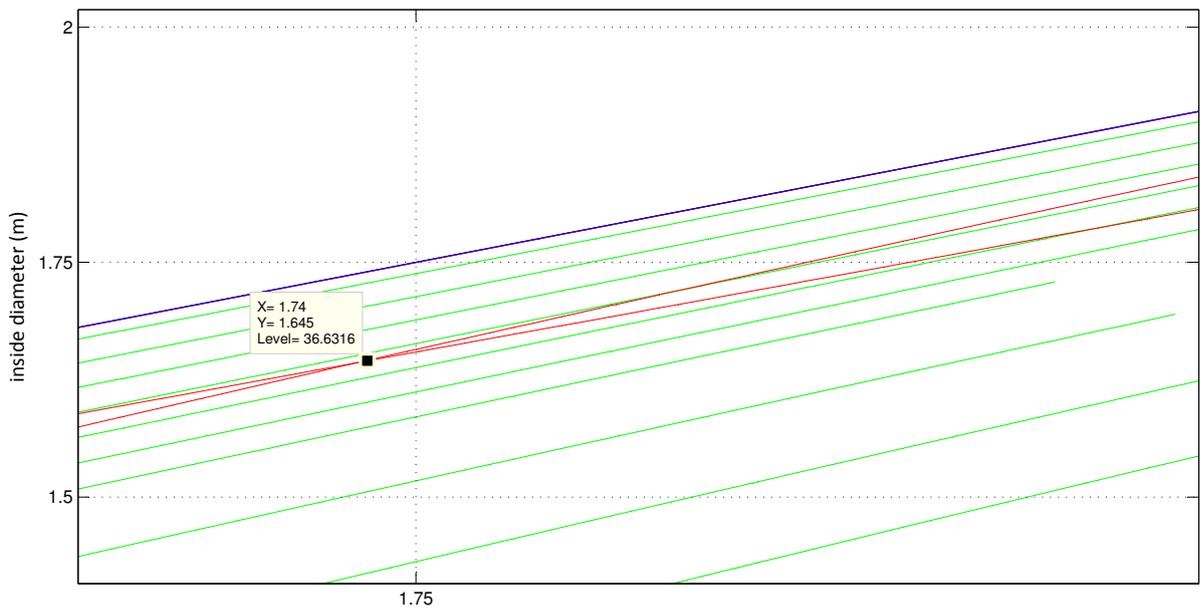


Figure 4.3: Magnified view of graphical optimization result of a bottom monopole tower section for $M_u = 1698.2$ kNm , $C_u = 64.5$ kN and $V_u = 79.6$ kN

The MATLAB code used in the implementation of this problem is included in Appendix A and is based on the design requirements described and discussed in section 4.4. The formulation of the constraint m-files, objective function and boundary values are included as well.

4.6 Chapter conclusion

From the graphical optimization approach with two variables, a relatively complex optimization problem can be simplified and the visualization helps to get a good picture of typical constraints, objective function contours and the feasible and infeasible regions. Such an approach is not necessarily suitable when it comes to real-world problems, as only two design variables can be considered at a time. Real-world structural optimization problems have a much larger search space and more design variables. The optimum solution is obtained by identifying the lowest possible objective value (cost contours) along the boundary of the feasible region. Considering the fact that most constraints are formulated as inequality constraints, the optimum generally lies on such a constraint boundary intersection where the global optimum, as defined, is realized.

Chapter 5

Lattice Type Tower LCC Optimization

5.1 Introduction

Fuzzy discrete multi-objective LCC optimization can be applied to model uncertainties in decision making situations as discussed in chapter 2. Intuitively, for a certain element group in a truss type structure, a design engineer might choose a section with the smallest possible cost (and/or lowest weight) that satisfies a set of constraints and regard this as the optimum. With the help of a case study, it will be demonstrated how misleading such an approach can be when dealing with multiple objectives related to LCC optimization. The λ -formulation is a practical approach to find the best compromise solution for such a discrete multi-objective optimization.

The subsequent sections deal with the determination and definition of the objectives, variables and constraints (lower and upper boundary constraints). A case study model is presented that applies the concepts of the λ -formulation to the two-objective problem as presented in the work by Kelesoglu and Ulker (2005). However, in this study the four-bar truss model is adjusted slightly to cater for a discrete set of sections that have varying cross-sectional areas. In contrast, Kelesoglu and Ulker (2005) assume the design variable (cross-sectional area) to be continuous. In a further subsection, the same problem is extended to a four-objective type as discussed by Adeli and Sarma (2006). In both of the four-bar truss models, the objective \tilde{T} (number of section types) is omitted because only one single element group is used and therefore only one design variable is present.

In section 5.8, a 24 m WECS tower LCC optimization is presented using the principles

discussed previously. The tower model is based on a scaled version from the work by Farkas and Jarmai (2008). The topology and shape are predetermined. The nodal and element positions within the structure, therefore the structural geometry is considered fixed, and numerous combinations exist that can result in an optimum solution. The aim with such a LCC optimization process is not to find such a global optimum; but rather to aid in decision making processes and to provide the design engineer with the best compromise solution for a set of predefined objectives, variables and constraints that can be varied.

5.2 Modeling objectives

The objectives that will be considered in the multi-objective optimization in this chapter are listed below and are discussed in chapter 2, section 2.9. An account on structural deflections is given under subsection 5.2.1.

- Weight of the rolled sections.
- Cost of the rolled sections.
- Perimeter of the rolled sections.
- Nodal deflections (alternatively inter-storey deflections).

5.2.1 Critical structural deflections

Tower deflections due to code limitations need to be accounted for when determining the constraint boundaries and for design in general. In the multi-objective optimization the structural deflection is treated as an objective, while for the single objective optimization, the deflection is treated as a constraint. However, the fitness value is affected similarly, and for more details, refer to the formulation by Rao *et al.* (1992) that is described and discussed in chapter 2, section 2.8.7. The limitations on critical deflections of WECS are discussed in detail in IEC 61400 - Part 1:2005, *Wind Turbines- Part 1: Design requirements*. SANS 10162 - Part 1:2005, Annex D includes a table with deflection limits for various types of structures and structural configurations. The values in Table 5.1 have been obtained from Table D.1 of the code. H is the corresponding building or storey height of the structure or structural component.

IEC 61400 - Part 1:2005, clause 7.6.5 states the following relevant information: It shall be verified that no deflections occur that affect the structural integrity under the respective design conditions that are stipulated in Table 2 of the code. Table 2 is a summary of

Table 5.1: Maximum deflections at serviceability (SANS 10162-1, Annex D, Table D.1)

Type of Structure	Deflection type	Design load	Application	Deflection/inter-storey limit
All other buildings	Lateral	Wind	Sway, due to all effects	$H/180$

corresponding limit states that shall be considered for a variety of design situations that are a function of the wind conditions and other external conditions. Examples of the design situations considered are start-up, power production, faulty power production, parked turbine, transport, assembly and maintenance (repair) to name a few.

The code makes provision for two limit states only, namely the ultimate and fatigue limit state.

Under clause 7.6.5 in IEC 61400 - Part 1:2005, emphasis is put on the design verification concerning the mechanical interference between blade and tower. The code specifies that the maximum elastic deflection in the unfavorable direction shall be determined for the load cases in Table 2 using the characteristic loads. Particular attention shall be given to geometrical eccentricities and imperfections as well as uncertainties. The elastic deflection shall be added to the un-deflected position in the most unfavorable direction and the resulting position compared to the requirement for non-interference (IEC 61400 - Part 1:2005).

Direct dynamic deflection analysis may also be used in determining the interference effects between blade and tower.

5.3 Modeling variables

The definition of design variables and parameters is of great importance in formulating an optimization problem. Design variables of a wind turbine include layout parameters as well as cross-sectional area variables. Tower variables include type (truss or tubular), height, cross-sectional area and material of construction. In this work shape optimization is investigated and the representative design variable in the optimization is the design vector of cross-sectional areas.

5.4 Modeling constraints

In fuzzy multi-objective optimization, the constraints are generally introduced to the model in terms of boundary constraints as discussed in chapter 2. The capacity at the ultimate limit state of the members in a lattice structure can govern these boundary constraints, especially the lower constraint value. Axial force resistance of compression elements is calculated according to SANS 10162 - Part 1:2005: *Limit-state design of hot-rolled steelwork*, clause 13.3. The tension resistance is calculated according to clause 13.2 and 13.10 and 13.11. Information regarding the calculation of a member's compressive and tensile resistance value is included in subsection 5.4.1 and 5.4.2 of this work, respectively. Nodal deflection limits may also govern the design variables for the serviceability limit state and a response model in the form of a Finite Element Analysis (FEA) is required.

5.4.1 Axial compression resistance

In this work, only at least class 3 axial compression elements have been used and class 4 sections were omitted in the optimization. Class 4 compression elements can be accommodated relatively easy by means of few adjustments and alterations. SANS 10162 - Part 1:2005, clause 10.4.2.1 states that the slenderness of elements in axial compression shall be limited as given by equation 5.4.1.

$$\frac{KL}{r} \leq 200 \quad (5.4.1)$$

where KL is the effective length of the member and r is the radius of gyration for buckling about the weak axis of the section.

SANS 10162 - Part 1:2005, clause 13.3 makes provision for determining the axial compressive resistance of an element using the double exponential formula in equation 5.4.2.

$$C_r = \phi A_g f_y (1 + \lambda^{2n})^{\frac{-1}{n}} \quad (5.4.2)$$

where λ is the non-dimensional slenderness ratio and is determined using equation 5.4.3.

$$\lambda = \sqrt{\frac{f_y}{f_e}} \quad (5.4.3)$$

Reference to different buckling failure modes (flexural, torsional or torsional-flexural) is made in clause 13.3 of the code. It accounts for doubly symmetric and singly symmetric

steel sections that are classified according to the description in clause 11 of the code. For class 1,2 and 3 (a) doubly symmetric (or axi-symmetric), hot-rolled sections, f_e should be taken as the lesser of f_{ex} , f_{ey} and f_{ez} . When (b) singly symmetric sections are considered, f_e shall be taken as the lesser of f_{ex} and f_{eyz} . With (c) asymmetric sections, f_e is the smallest root of expression 5.4.4. The values are calculated as stipulated in SANS 10162 - Part 1:2005, clause 13.3.2 of the code and are given below for reference purposes.

$$(f_e - f_{ex})(f_e - f_{ey})(f_e - f_{ez}) - f_e^2(f_e - f_{ey})\left(\frac{x_o}{\bar{r}_o}\right)^2 - f_e^2(f_e - f_{ex})\left(\frac{y_o}{\bar{r}_o}\right)^2 = 0 \quad (5.4.4)$$

$$f_{ex} = \frac{\pi^2 E}{\left(\frac{K_x L_x}{r_x}\right)^2} \quad (5.4.5)$$

$$f_{ey} = \frac{\pi^2 E}{\left(\frac{K_y L_y}{r_y}\right)^2} \quad (5.4.6)$$

$$f_{ez} = \left(\frac{\pi^2 E C_w}{K_z^2 L_z^2} + GJ\right) \frac{1}{A \bar{r}_o^2} \quad (5.4.7)$$

$$f_{eyz} = \frac{f_{ey} + f_{ez}}{2\Omega} \left(1 - \sqrt{1 - \frac{4f_{ey}f_{ez}\Omega}{(f_{ey} + f_{ez})^2}}\right) \quad (5.4.8)$$

$$\Omega = 1 - \left(\frac{x_o^2 + y_o^2}{\bar{r}_o^2}\right) \quad (5.4.9)$$

$$\bar{r}_o^2 = x_o^2 + y_o^2 + r_x^2 + r_y^2 \quad (5.4.10)$$

where E and G are the elastic and shear modulus of the material, respectively. C_w is the warping constant of the section, J is the St. Venant torsional constant, A is the cross-sectional gross area of the section and the effective length $K_z L_z$ is the effective length of the torsionally unrestrained portion of the element. x_o and y_o are the principal coordinates of the shear center of the section with respect to the centroid position. r_x and r_y are the radii of gyration about the x and y -axis respectively. It is important to note that, as

stipulated in SANS 10162 - Part 1:2005, clause 13.3.2 for singly symmetric sections, the y -axis needs to be taken as the axis of symmetry.

The factored compressive resistance C_r of a compression member that is classified as a class 4 section according to clause 11, is determined according to clause 13.3.3 of the code. Such members are generally sensitive to local buckling failure before global buckling occurs. When $W = \frac{b}{t} > W_{lim}$, the gross cross-sectional area needs to be reduced to an effective area A_{ef} . This effective area shall be determined with reduced element widths to result in an equivalent W of a class 3 section. The reduced element width b is obtained using equation 5.4.11. When $W \leq W_{lim}$, this gross area reduction of the class 4 section is not necessary.

$$b = 0.95t \sqrt{\frac{kE}{f}} \left(1 - \frac{0.208}{W} \sqrt{\frac{kE}{f}} \right) \quad (5.4.11)$$

W_{lim} is determined as follows:

$$W_{lim} = 0.644 \sqrt{\frac{kE}{f}} \quad (5.4.12)$$

where $k = 4.0$ for elements of a member supported along both longitudinal edges and $k = 0.43$ for elements of a member that are supported along one longitudinal edge. The average axial compressive stress f developed in the member is calculated using equation 5.4.13:

$$f = \frac{C_u}{\phi A} \leq f_y \quad (5.4.13)$$

As described in the literature review sections, the constraints such as the buckling fitness and slenderness fitness, are introduced as quadratic penalties to ‘punish’ the fitness of an individual in the search towards the optimum. For compression elements, the algorithm pseudocode is given below.

```
isBuckled = N <= -Cr
isSlenderC = L/rv > 200

numElementsBuckled = sum(isBuckled)
numElementsSlenderC = sum(isSlenderC)
```

```

numElementsBuckledPenalty = (1-numElementsBuckled/212)^2
numElementsSlenderPenaltyC = (1-numElementsSlenderC/212)^2

bucklingFitness = numElementsBuckledPenalty
slendernessFitnessC = numElementsSlenderPenaltyC

```

5.4.2 Axial tension resistance

SANS 10162 - Part 1:2005, clause 10.4.2.2 states that the slenderness of elements in axial tension shall be limited as given by equation 5.4.14.

$$\frac{KL}{r} \leq 300 \quad (5.4.14)$$

where KL is the effective length of the member and r is the radius of gyration about the weak axis of the section.

SANS 10162 - Part 1:2005, clause 13.2 makes provision for different tension related failure modes of axially loaded tension elements. The tensile resistance can be calculated as follows, where the tensile resistance becomes the smallest of the values obtained in equations 5.4.15, 5.4.16 and 5.4.17. The latter equation makes provision for the effects due to shear lag that may develop near the connection at load transfer and the reduced effective net area is denoted as A'_{ne} .

$$T_r = \phi A_g f_y \quad (5.4.15)$$

$$T_r = 0.85 \phi A_{ne} f_u \quad (5.4.16)$$

$$T_r = 0.85 \phi A'_{ne} f_u \quad (5.4.17)$$

For tension elements with pinned connections, equation 5.4.18 applies.

$$T_r = 0.75 \phi A_{ne} f_y \quad (5.4.18)$$

The calculated values of A_{ne} and A'_{ne} need to be altered according to the specifications for certain cases as stipulated in SANS 10162 - Part 1:2005, clause 12.3. Different net and effective net areas apply to different connection configurations. Examples of differences

could be the number of rows of transverse fasteners (bolts or pins) and various welding configurations.

At ends of tension members and elements such as angle cleats, gussets and single plate connections where block failure modes such as tensile fracture and shear yielding or fracture may govern, the tensile resistance of the tensile element shall be determined using expressions 5.4.19 and 5.4.20.

$$T_r + V_r = \phi A_{nt} f_u + 0.60 \phi A_{gv} f_y \quad (5.4.19)$$

or

$$T_r + V_r = \phi A_{nt} f_u + 0.60 \phi A_{nv} f_u \quad (5.4.20)$$

where A_{nt} is the net area in tension for block failure, A_{nv} is the net area in shear for block failure and A_{gv} is the gross area in shear for block failure.

The shear resistance of fasteners, welds and pins, as well as bearing resistances at fastener positions will not be included in this subsection. However, reference is made to SANS 10162 - Part 1:2005, clause 13.10 for bearing resistances, clause 13.12 for bolts in shear and tension and, clause 13.13 for welds.

As described in the literature review part, the constraints such as the yielding fitness and slenderness fitness, are introduced as quadratic penalties to ‘punish’ the fitness of an individual in the search towards the optimum. For tension elements, the algorithm pseudocode is given below.

```

isYielded = 0 >= yieldStress
isSlenderT = L/rv > 300

numElementsYielded = sum(isYielded)
numElementsSlenderT = sum(isSlenderT)

numElementsYieldedPenalty = (1-numElementsYielded/212)^2
numElementsSlenderPenaltyT = (1-numElementsSlenderT/212)^2

yieldingFitness = numElementsYieldedPenalty
slendernessFitnessT = numElementsSlenderPenaltyT

```

5.5 The FEM Solver

Finite Element Methods (FEM) are used as an effective and efficient tool in structural analysis. In short, a Finite Element Analysis (FEA) is a numerical technique that solves structural integral and differential equations for an entire system. Such a system consists of a number of finite elements whose behavior is individually formulated in mathematical and physical terms. Hand calculations may result in a very cumbersome task and computers are considered as useful tools. During the past few decades, personal computers available on the market improved significantly in performance and also became relatively affordable. A FEA can be applied to almost any field and also nonhomogeneous and anisotropic materials can be modeled. Techniques have been formulated to also accommodate material and geometric non-linearity. These are of an iterative or incremental nature, where step-wise adjustments are made to the stiffness \mathbf{K} and/or loads \mathbf{F} during analysis. The stiffness and load of a non-linear system become a function of the displacement \mathbf{a} . Examples include the Newton-Raphson and Arc-length methods (Cook *et al.*, 2002).

The mathematical formulations for various finite elements are described in detail in Cook *et al.* (2002). For space truss type structures with slender elements (large deflections) a second-order structural analysis is generally recommended due to P- Δ effects and thus stability becomes a concern. A simple linear, static analysis will be assumed to be adequate for LCC optimization. This supports the simplification to the optimization of LCC in terms of decision making and does not replace any structural analysis for second-order effects and stability.

The general equation for a linear, static FEA is given in equation 5.5.1.

$$\mathbf{K} \mathbf{a} = \mathbf{F} \quad (5.5.1)$$

For a truss element in three-dimensional configuration, the local stiffness matrix \mathbf{K}_e is given as.

$$\mathbf{K}_e = \left(\frac{EA}{L} \right) \begin{bmatrix} C_x^2 & C_x C_y & C_x C_z & -C_x^2 & -C_x C_y & -C_x C_z \\ & C_y^2 & C_y C_z & -C_x C_y & -C_y^2 & -C_y C_z \\ & & C_z^2 & -C_x C_z & -C_y C_z & -C_z^2 \\ & & & C_x^2 & C_x C_y & C_x C_z \\ & & & & C_y^2 & C_y C_z \\ & & & & & C_z^2 \end{bmatrix} \quad (5.5.2)$$

symmetric

where the cosines are defined as follows:

$$C_x = \frac{x_2 - x_1}{L}; C_y = \frac{y_2 - y_1}{L}; C_z = \frac{z_2 - z_1}{L} \quad (5.5.3)$$

A basis transformation is made in order to obtain the global stiffness matrix \mathbf{K} , global vector of equivalent nodal forces \mathbf{F} and vector of displacements \mathbf{a} (Ferreira, 2009).

Figure 5.1 is an illustration of a typical, pinned bar (or truss) element in a three dimensional configuration. Take note of the difference between the local (lowercase letters) and the global (uppercase letters) coordinate system. The element has two nodes with each three global, orthogonal degrees of freedom.

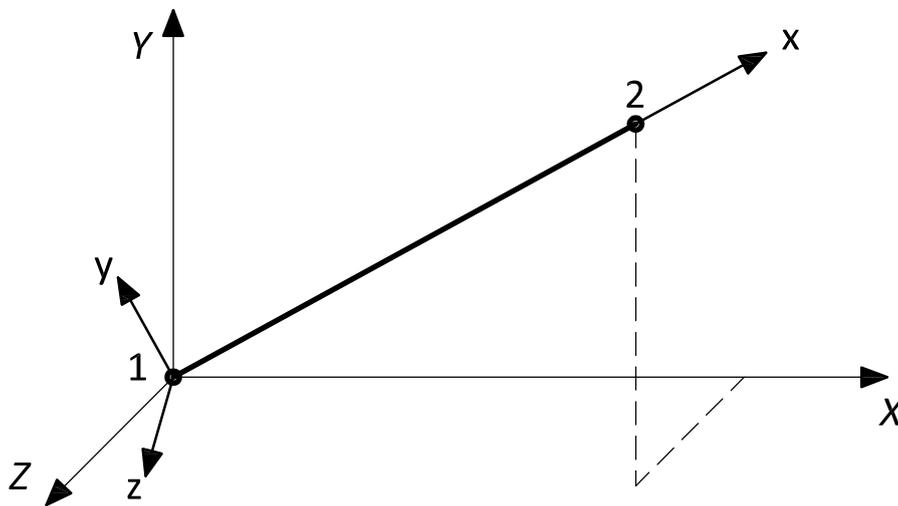


Figure 5.1: Illustration of a bar (truss) element in three dimensional configuration

Figure 5.2 is a schematic representation of a bar element (a) subject to a tensile axial force T_u and (b) under compressive axial force C_u . For different load cases and load combinations, a certain truss element in a structure can act as both, a compression element and as tension element consequently.

A structural analysis is required to inform the objective numerical evaluation made of an individual by the fitness function in a GA. The section on fitness functions, section 2.8.8.7, contains additional details. An interfaceable and compatible FEM program is required for the genetic search algorithm. The use of a FEM package with an Application Programming Interface (API) is often preferred, but the use of a natively written MATLAB tool was chosen as this also simplifies programming a lot. However, several

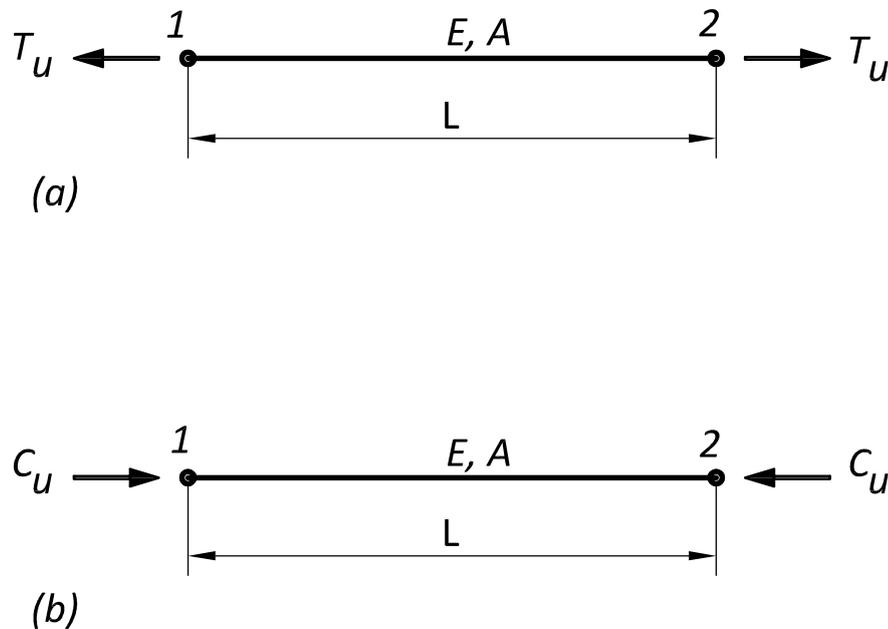


Figure 5.2: Representation of the loading configuration of (a) a two node tension bar element (Tie) and (b) a two node compression bar element (Strut)

limitations exist concerning the range of applications and the problem scope. The API of structural analysis software such as Strand7 would have been an alternative option. Using a MATLAB solution, however, eliminates outside dependencies on other software, as only MATLAB is required.

The FEM solver developed and used in this work has been used to provide the fitness function with the structural analysis data of an individual. The FEM solver used for the 25-bar truss and the 212-bar truss was extensively modified from the example in Reynolds (2009).

The FEM solver developed and adopted in this work is based on linear elastic behavior. Although a second order analysis is mandatory when the structural analysis concerns long and slender elements where large deflections can occur, it is regarded as adequate for decision making. A second order analysis is carried when it comes to the final design or verification model.

The main input to the solver is the array of element cross-sectional areas \mathbf{A} . A property matrix $\mathbf{e}^p = [\mathbf{E} \ \mathbf{A}]$ is created from a combination of the elastic modulus E and sectional area A_i .

The solver returns the following:

- The global displacement vector \mathbf{a}
- A reaction vector \mathbf{Q}
- An element force vector \mathbf{N}
- An element stress vector \mathbf{O}
- A total weight value W

The fitness function calls the FEM solver and the returned variables are used to evaluate the fitness function by means of the λ -formulation described earlier.

5.6 An example using MS Excel

In this section, a four-bar space truss structure is used to study and verify the methodology that will be applied to a larger model in the subsequent sections of this work by comparing the calculated results to those obtained by the max-min or λ -formulation in the paper *Multi-Objective Fuzzy Optimization of Space Trusses by MS-Excel* by Kelesoglu and Ulker (2005).

The authors apply the λ -formulation as described in chapter 2 of this work. By referring to 2.9.3 in chapter 2, the four factors that will be considered in the fuzzy multi-criteria optimization model for the LCC optimization of a lattice type truss tower are, as mentioned earlier, the cost, the weight, the perimeter of the rolled sections and the maximum nodal deflection. The expressions for the membership functions are presented in the chapter and section as mentioned above. The fuzzy sets of candidate discrete shapes that correspond to the aforementioned objectives or criteria are S_{C_i} , S_{W_i} , S_{d_i} and S_{P_i} respectively. The optimum multi-criteria solution is found by using the λ -formulation that is based on the max-min procedure by Bellman and Zadeh (1970). A fuzzy set, S_{D_i} is defined as the intersection of the four fuzzy sets as follows:

$$S_{D_j}(\tilde{y}_{ij}) = S_{C_j}(\tilde{y}_{ij}) \cap S_{W_j}(\tilde{y}_{ij}) \cap S_{d_j}(\tilde{y}_{ij}) \cap S_{P_j}(\tilde{y}_{ij}), \quad j = 1, 2, \dots, N_i \quad (5.6.1)$$

The membership of this intersection fuzzy set S_{D_j} is equated to the minimum values of the four membership functions described above and this minimum value is given in expression 5.6.2.

$$\mu_{D_j}(\tilde{y}_{ij}) = \min[\mu_C(\tilde{y}_{ij}); \mu_W(\tilde{y}_{ij}); \mu_d(\tilde{y}_{ij}); \mu_P(\tilde{y}_{ij})], \quad j = 1, 2, \dots, N_i \quad (5.6.2)$$

The fuzzy membership functions in expression 5.6.2 represent the four criteria as discussed. The commercially available discrete section that corresponds to the maximum membership function $\lambda = \mu_{D_{(best)}}$, of the N_i selected candidate shapes, is the best compromise section.

$$\lambda = \mu_{D_{(best)}} = \max[\mu_{D_j}(\tilde{y}_{ij})] \quad i = 1, 2, \dots, N_t \quad (5.6.3)$$

This step is referred to as defuzzification in the work by Adeli and Sarma (2006). The concept is further explained through the example that is presented in the following paragraphs and also in Figure 5.3.

For simplification of this study or rather verification model, steel bar elements with a constant cross-sectional area are used. The weight of the structure is ignored in the calculation of the axial forces and deflections that develop in the structure. A point load $P = 15 \text{ kN}$ is applied at node 5 in the direction of the global y-axis. The flexural properties do not affect the internal stresses or structural displacement, as it is assumed to be a perfectly pinned space truss structure with bar elements. The load can be assumed to be applied under serviceability limit state conditions.

The fuzzy multi-objective discrete cost optimization was performed in MS Excel using only the two objectives as presented in the work by Kelesoglu and Ulker (2005), namely the weight of the structure and the deflection at node 1. Further details are included in subsection 5.6.1. The problem of the four-bar truss structure was then expanded to also include the additional objectives, such as total section perimeter and the total cost of the rolled sections. The corresponding details are found in subsection 5.6.2.

5.6.1 Part 1: Fuzzy multi-objective cost optimization with two objectives

Objectives: The displacement and material cost (linear function of the material volume in this case) are the only objective functions used in this example for illustration pur-

poses and are the same as used by Kelesoglu and Ulker (2005). A simple FEA has been implemented in MS Excel to determine the nodal deflection at node 1 for each discrete cross-sectional area of the four-bar truss structure. The procedure that is described in section 5.4 has been applied.

Variables: The cross-sectional area is the only design variable chosen for this example. However, the cross-sectional area is a discrete variable and not continuous over a certain range. Six arbitrary, discrete candidate sections are chosen that have the following cross-sectional areas: 259.94 mm^2 , 280 mm^2 , 320 mm^2 , 380 mm^2 , 480 mm^2 and 600 mm^2 .

Constraints: The cross-sectional area has been constrained by two boundary values, a lower A_l and an upper A_u constraint.

The additional parameters used and obtained from the paper by Kelesoglu and Ulker (2005), are given below in Table 5.2:

Table 5.2: Model constraints and material parameters

Parameter	Value
Material	S355JR Steel
E	210 GPa
P	15 kN
A_l	259.94 mm^2
A_u	600.00 mm^2

Figure 5.3 is a schematic representation of the four-bar truss model and all dimensions and loads are shown.

The optimum solution \mathbf{x}^* , which is Pareto optimal (Papadrakakis *et al.*, 2005), is found by applying the concept of the λ -formulation as described in chapter 2. This optimum is the intersection of the membership functions of the objective functions and the constraints (Rao *et al.*, 1992). λ is maximized subject to the following conditions:

$$\lambda \leq \mu_f(\mathbf{x}) \quad (5.6.4)$$

$$\lambda \leq \mu_{g_i}^{upper}(\mathbf{x}) \quad i = 1, 2, \dots, N \quad (5.6.5)$$

$$\lambda \leq \mu_{g_i}^{lower}(\mathbf{x}) \quad i = 1, 2, \dots, N \quad (5.6.6)$$

$$0 \leq \lambda \leq 1.0 \quad (5.6.7)$$

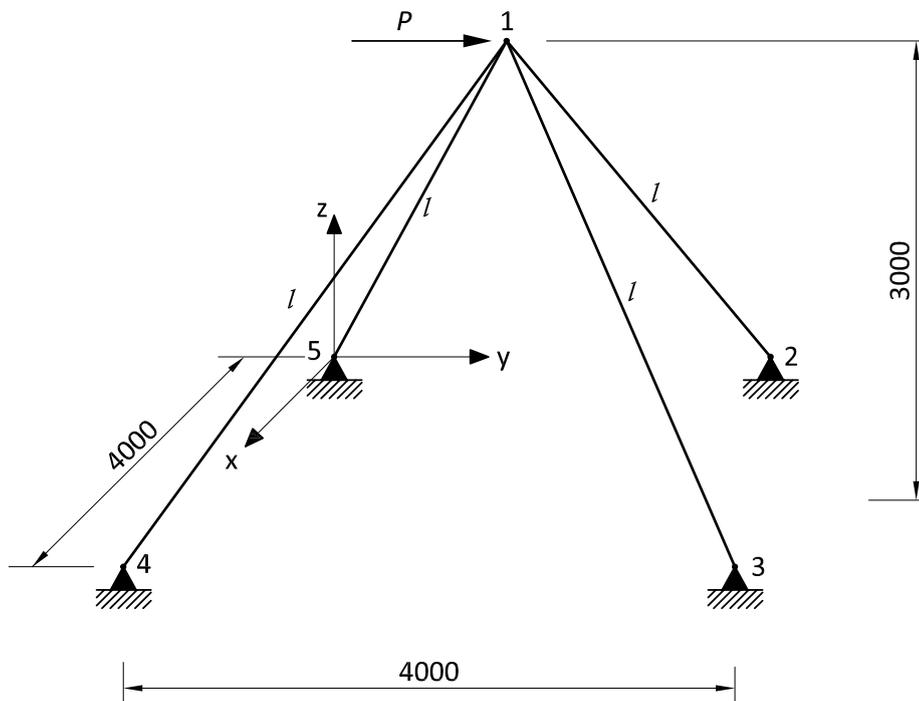


Figure 5.3: A four-bar space truss structure used to verify optimization results

The results obtained for the two-objective cost optimization are given in Table 5.3 and are graphically shown in Figure 5.4.

Table 5.3: Results of four-bar truss example - Part 1

Design variable	λ	A (mm^2)	δ (mm)	Cost (R)
Continuous	0.6012	394.92	0.792	1022.58
Discrete	0.557	380.00	0.823	983.94

For the case where the continuous design variables are considered, the same results as in the work by Kelesoglu and Ulker (2005) are obtained. For the discrete section case, the nearest possible section is the $A = 380 \text{ mm}^2$ and represents the best compromise solution of the bi-objective, single design variable optimization. $\lambda = 0.557$, which is lower than $\lambda = 0.6012$ for the continuous design variable case.

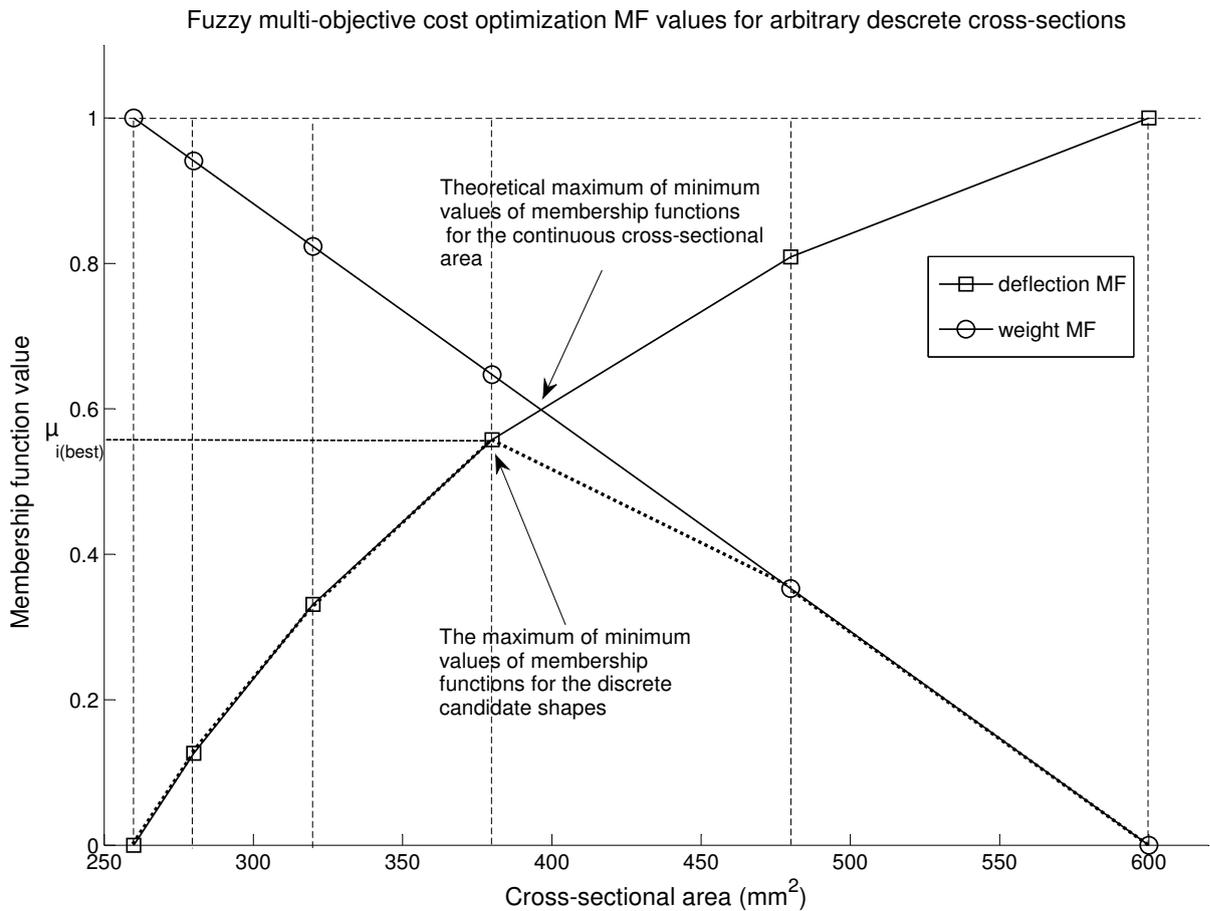


Figure 5.4: Fuzzy cost optimization results for a 4-bar truss structure with two objectives

5.6.2 Part 2: Fuzzy multi-objective cost optimization with four objectives

Objectives: The displacement, material cost, material weight and perimeter are the objective functions used in this second part of the example.

Variables: The single cross-sectional area is the only design variable chosen for this example. However, the cross-sectional area is a discrete variable and not continuous over a certain range. Again, six arbitrary, discrete candidate sections are chosen that have the following cross-sectional areas: 259.94 mm^2 , 280 mm^2 , 320 mm^2 , 380 mm^2 , 480 mm^2 and 600 mm^2 .

Constraints: The cross-sectional area has been constrained by two boundary values, a lower A_l and an upper A_u constraint. The additional parameters are the same as for the two-objective optimization problem and are given in Table 5.2. Figures 5.5 to 5.8 include

the details to the problem described above.

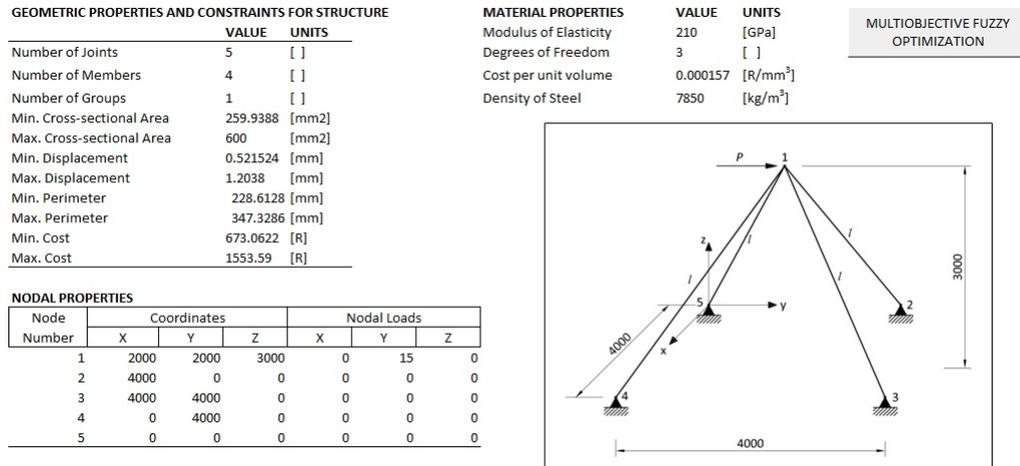


Figure 5.5: General information and nodal positions

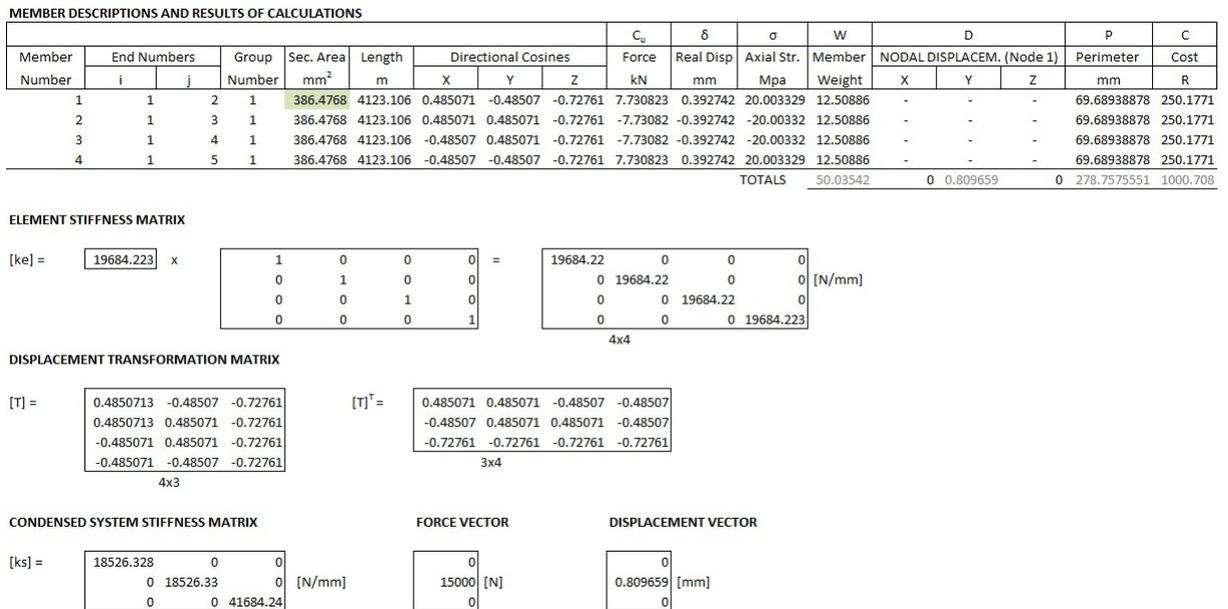


Figure 5.6: Member description and Finite Element Analysis

The results obtained for the four-objective cost optimization are given in Table 5.4.

Table 5.4: Results of Four-bar truss example - Part 2

Contin./Discr.	Area	λ	Area (mm ²)	δ (mm)	Weight (kg)	Perimeter (mm)	Cost (R)
Continuous		0.5776	386.48	0.8097	50.034	278.75	1000.68
Discrete		0.5575	380.00	0.8235	49.197	276.41	983.94

FUZZY MEMBERSHIP FUNCTION VALUES

$\max \min \lambda$
0.577607

STRUCTURAL WEIGHT OBJECTIVE MEMBERSHIP FUNCTION (a function of vol)

$Weight_{max}$	$Weight_{min}$	λ_{weight}
77.67931	33.653111	0.627896

STRUCTURAL DISPLACEMENT OBJECTIVE MEMBERSHIP FUNCTION

$Disp_{max}$	$Disp_{min}$	λ_{disp}
1.2038	0.5215238	0.577686

TOTAL PERIMETER OBJECTIVE MEMBERSHIP FUNCTION

$Perim_{max}$	$Perim_{min}$	λ_{perim}
347.328602	228.61277	0.577607

TOTAL MATERIAL COST OBJECTIVE MEMBERSHIP FUNCTION

$Cost_{max}$	$Cost_{min}$	λ_{cost}
1553.59	673.06222	0.627898

NUMBER OF SECTION TYPES

not applicable, only one element group.

Figure 5.7: Fuzzy membership values and results

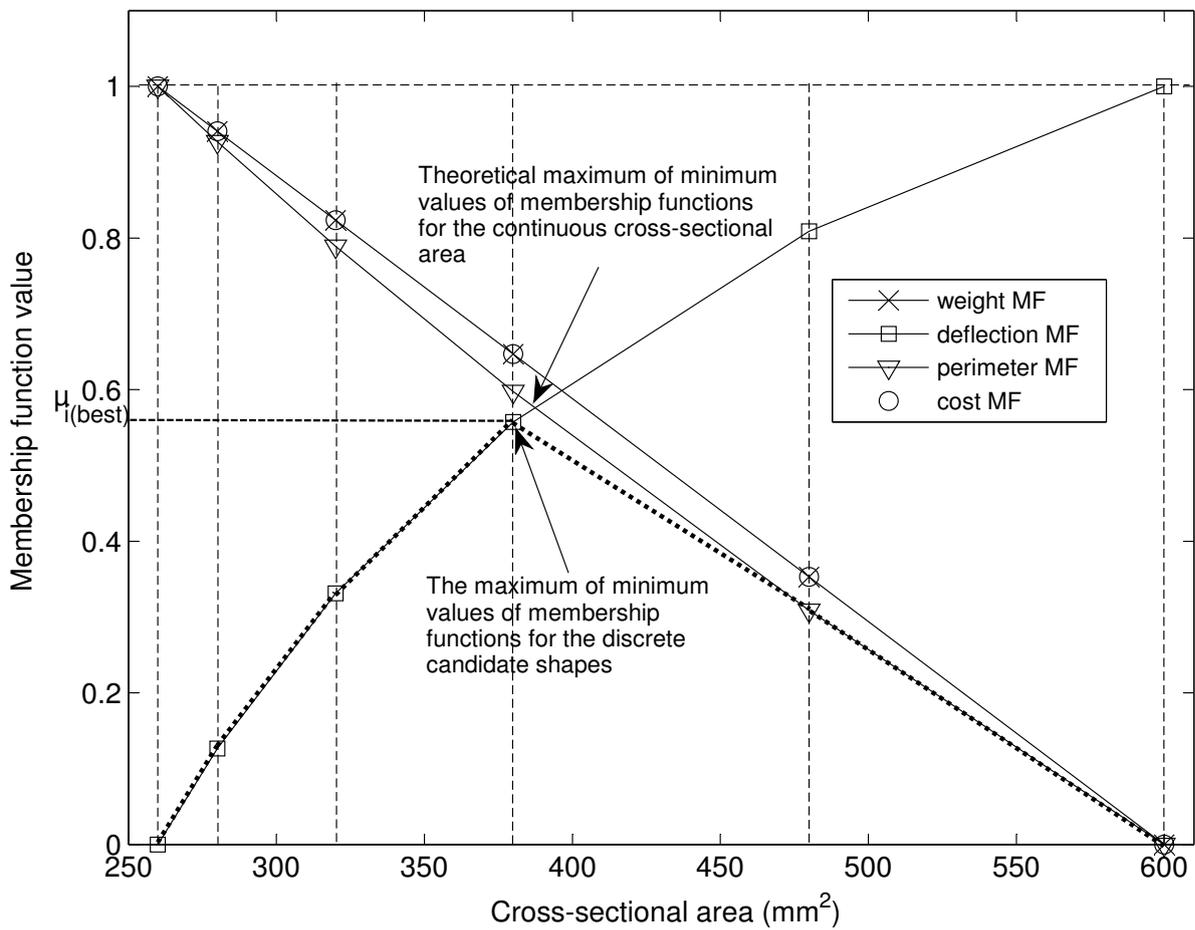


Figure 5.8: Fuzzy cost optimization results for a 4-bar truss structure with multiple objectives

The results obtained for the four-objective problem are similar to the bi-objective case. The multiple objectives caused the overall satisfaction factor λ for the continuous case to be significantly lower than before. Remember that the problem is unconstrained for buckling and yielding, and simply the upper and lower bound on the discrete design variable exist. The upper and lower boundary values of the fuzzy membership functions are the preference values from the decision maker. The values are obtained from Kelesoglu and Ulker (2005), where applicable and the others are calculated based on the minimum and maximum possible values.

5.7 25-Bar truss tower model

Real steel truss structures predominantly consist of standard hot-rolled steel sections that can be bought from steel mills and producers nationwide and that are listed in the *South African Steel Construction Handbook* from SAISC. Therefore, to attain the goal of optimizing multiple objectives of a truss structure, we must make use of discrete design variables as illustrated in the previous section.

The 25-bar truss tower is a famous and suitable model used for benchmark tests in a number of research papers and publications on structural optimization using GAs such as Erbatur *et al.* (2000), Coello *et al.* (1994), Togan and Daloglu (2008) and Appelo (2012). The 25-bar truss tower is converted from the original imperial units into metric SI units. The material used in this benchmark test is aluminium, with an elastic modulus of $E = 68.948 \text{ GPa}$ and material density of $\rho = 2768.0 \text{ kg/m}^3$.

The members were grouped by means of an index array or vector. The index array overwrites the position index of the member by assigning a user defined value to it. This allows the user to group certain elements together for loading symmetry and structural geometric symmetry. The index array for the 25-bar truss tower is given below and the values represent the eight design variables of the entire structure.

$$\mathbf{indexArray} = [1 \ 2 \ 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 3 \ 4 \ 4 \ 5 \ 5 \ 6 \ 6 \ 6 \ 6 \ 7 \ 7 \ 7 \ 7 \ 8 \ 8 \ 8 \ 8]^T$$

Objectives: The objectives for the 25-bar truss benchmark problem are the same as discussed in the first part of the four-bar example. It is widely used to validate the accuracy and functionality of a genetic algorithm by means of a benchmark problem. Weight is the only objective that was considered in the literature for this specific case. The maximum allowable nodal displacement is treated as a constraint in the single objective runs. A multi-objective optimization of the nature similar as discussed for the four bar

truss was not found in the literature. A schematic representation of the problem is shown in Figure 5.9.

Variables: The cross-sectional area vector. Each component of the vector represents one of the eight design variables.

Constraints: The constraints used for this benchmark problem are only based on maximum average axial stress and maximum nodal deflection. The deflection constraint used is $d_{max} \leq 8.89 \text{ mm}$ and the stress constraint $-275.8 \text{ MPa} \leq \sigma_i \leq +275.8 \text{ MPa}$, for $i = 1, 2, \dots, 8$.

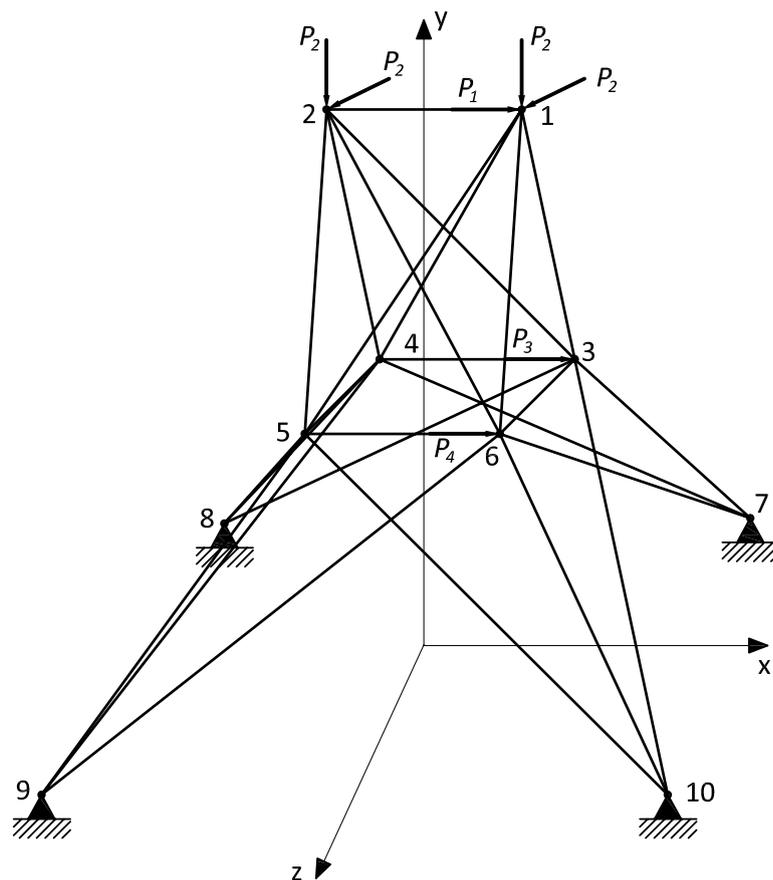


Figure 5.9: 25-bar space truss

The nodal coordinates of the 25-bar truss tower are given in Table 5.5.

Table 5.6 includes all the cross-sectional area values of the available discrete sections. Additional cross-sectional properties of the section are not available and concerning the benchmark problem, the stress constraint is based on the average normal or axial stress

Table 5.5: 25-bar truss tower nodal coordinates

Node number	X-Coord. (m)	Y-Coord. (m)	Z-Coord. (m)
1	0.9525	5.080	0.0
2	-0.9525	5.080	0.0
3	0.9525	2.540	-0.9525
4	-0.9525	2.540	-0.9525
5	-0.9525	2.540	0.9525
6	0.9525	2.540	0.9525
7	2.540	0.0	-2.540
8	-2.540	0.0	-2.540
9	-2.540	0.0	2.540
10	2.540	0.0	2.540

across the cross-section only and hence, the additional cross-sectional geometric properties do not play a role. This would not be the case when considering the critical buckling capacity of an element subject to axial compression.

Table 5.6: 25-bar truss tower benchmark section list

Section area list				
Area (mm^2)				
64.516	129.032	193.548	258.064	322.58
387.10	451.60	516.13	580.64	645.20
709.70	774.20	838.70	903.20	967.74
1032.26	1096.77	1161.29	1225.8	1290.32
1354.84	1419.35	1483.87	1548.38	1612.9
1677.42	1806.45	1935.48	2064.51	2193.54

For the benchmark problem a standard set of nodal loads is used. The force vectors are applied at the nodes as shown in Figure 5.9. The magnitudes of the forces are given in Table 5.7. The nodal support details are shown in Table 5.8. The upper case coordinates X, Y and Z in Table 5.8 represent fixed translational degrees of freedom at the respective nodes.

Table 5.7: 25-bar truss tower benchmark nodal load values

Nodal loads			
P_1 (kN)	P_2 (kN)	P_3 (kN)	P_4 (kN)
4.54	44.5	2.225	2.67

Table 5.8: 25-bar truss tower benchmark nodal supports

Nodal supports			
Node 7	Node 8	Node 9	Node 10
<i>XYZ</i>	<i>XYZ</i>	<i>XYZ</i>	<i>XYZ</i>

It should be noted that GAs follows stochastic processes and consequently there is an element of uncertainty involved in the results. However, general trends should hold true giving similar results if an identical test is performed. Say, a GA optimization is run 20 times with a particular set of parameters and the optimum solution is reached five times, it remains reasonable to expect that when the GA is repeatedly run for 20 times, the near optimum solution would be reached an approximately similar number of times.

The fuzzy discrete optimization (for the multi- and single-objective cases) is performed using MATLAB. The methodology as described in the Literature Review chapter is used and the optimization algorithm flowchart is given in Figure 5.10. The details in the figure give the reader a better overview of the m-functions implemented in the fuzzy discrete single and multi-objective optimization approaches described and discussed in the subsequent subsections.

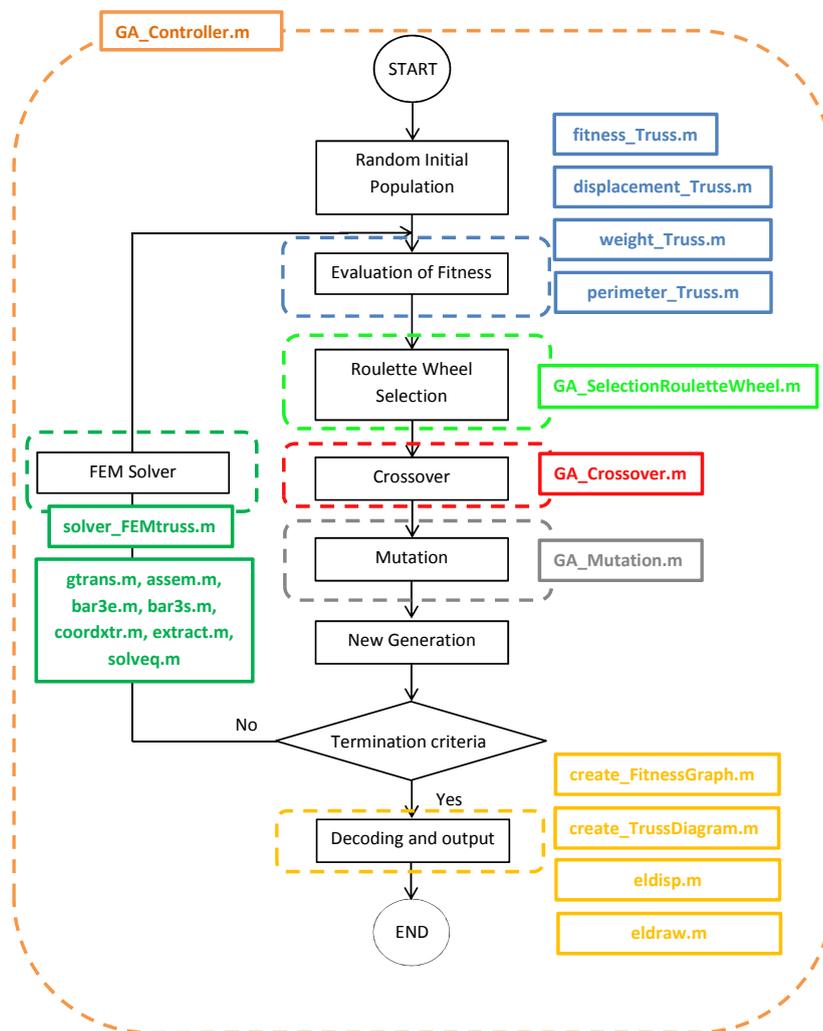


Figure 5.10: Fuzzy discrete optimization flowchart including the applicable m-functions.

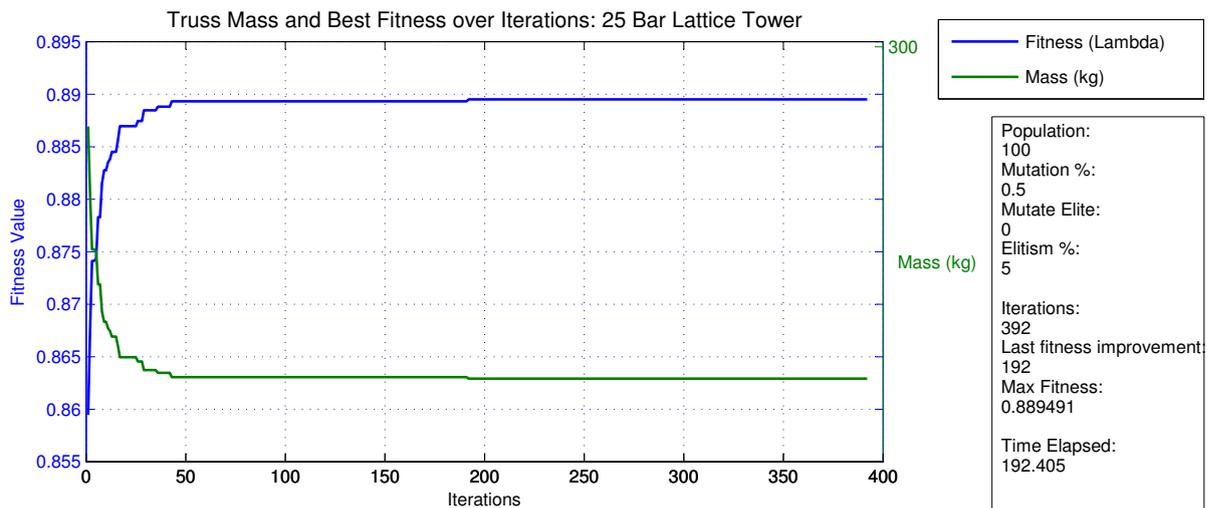
The results of the fuzzy discrete single-objective optimization for the 25-bar truss tower benchmark problem are shown in Table 5.9. The best (or fittest) result was obtained after 392 iterations and took 192 seconds. The GA parameters used are as follows: Population size of 100, mutation rate of 0.5%, elitism of 5% and mutation of elites disabled. The termination condition for the benchmark problem is, that after 200 iterations of no improvement in fitness, the algorithm ends and a near optimal solution is found given that no premature convergence took place and also given that convergence has taken place.

The results obtained are satisfactory and the representative fitness value λ equals 0.8895. Also take note that the representative fitness obtained is for the single objective problem (weight as the only objective). The optimum weight obtained is 221 kg and compared very well to the benchmark results in Appelo (2012).

Table 5.9: 25-bar truss tower benchmark run results

Run results	
Time (s)	192.41
Population	100
Iteration	392
A1	64.52 mm^2
A2	387.1 mm^2
A3	2193.54 mm^2
A4	64.52 mm^2
A5	1096.77 mm^2
A6	580.64 mm^2
A7	322.6 mm^2
A8	2193.54 mm^2

Figure 5.11 graphically shows the best fitness value and the structure's total mass over the total number of iterations until the termination condition is reached. Also take note of the details in the right column of the figure.

**Figure 5.11:** 25-bar space truss fitness and mass graph

As outlined before, the FEM solver returns the performance values for the algorithm to evaluate the fitness. Among these performance values are element forces and associated stresses, nodal deflections, reactions and structural weight. Figure 5.12 shows the FEM model as a MATLAB plot. The displacement in the figure is the representative nodal deflection in mm at the node with the maximum absolute deflection in the structure relative to the undeformed structure.

Table 5.10 presents the sample FEM solver output for the fittest design obtained for the 25-bar benchmark problem. Take note, that the element stresses are far below the constraint value. This is due to the nodal deflection constraint that acts as a limiting factor. The algorithm assigns the sections in such a way to attain the deflection limit (stiffness) simultaneously with the minimum total structural weight. The maximum axial element stress will be the limiting factor when a large enough deflection constraint value is used.

Table 5.10: Sample FEM Output

Element	N	Group	N_{max}	$A_{assigned}$	σ	Check
	(kN)		(kN)	(mm ²)	(MPa)	
1	2.9616	1	2.9616	64.52	45.90	< 275.8 MPa
2	-12.268	2	12.268	387.1	32.19	< 275.8 MPa
3	-8.3785					
4	7.1146					
5	11.1249					
6	-77.0913	3	80.2744	2193.54	36.60	< 275.8 MPa
7	-80.2744					
8	30.5007					
9	33.7827					
10	-0.3702	4	0.3702	64.52	5.74	< 275.8 MPa
11	-0.327					
12	-35.0956	5	35.0956	1096.77	32.00	< 275.8 MPa
13	15.3155					
14	-17.5017	6	18.8123	580.64	32.40	< 275.8 MPa
15	9.3887					
16	-18.8123					
17	8.0362					
18	3.0692	7	8.3472	322.6	25.87	< 275.8 MPa
19	4.0959					
20	-7.6817					
21	-8.3472					
22	-85.8611	8	93.3412	2193.54	42.55	< 275.8 MPa
23	44.4151					
24	36.4527					
25	-93.3412					

The fittest design obtained by the algorithm after termination is plotted in Figure 5.13, giving the user a clear visual result of the fittest design. The design variables are labeled accordingly in the figure. In some instances, the same color has been assigned to several design variables. To overcome the difficulty of distinguishing these, an element line thick-

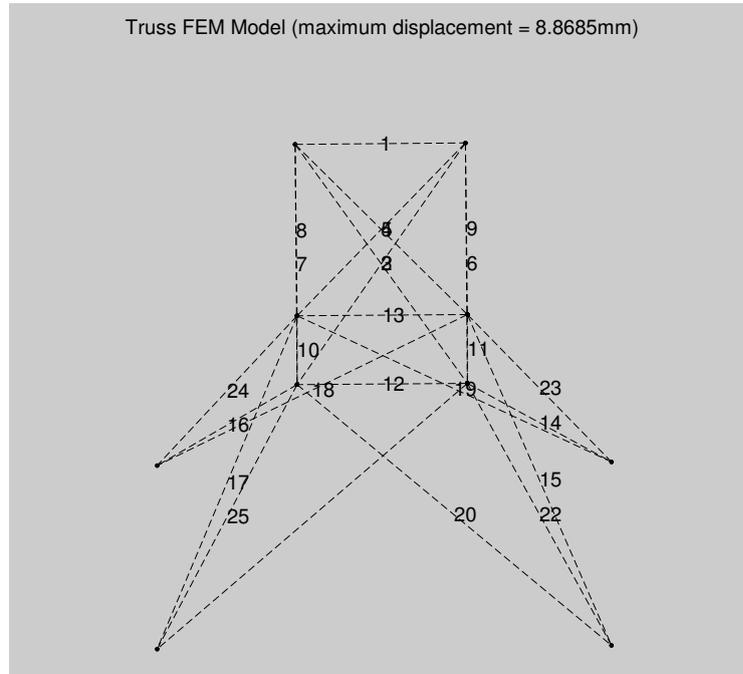


Figure 5.12: 25-bar space truss FEM model

ness scaling function is implemented to let the elements with a larger cross-sectional area of the same color appear with larger thickness.

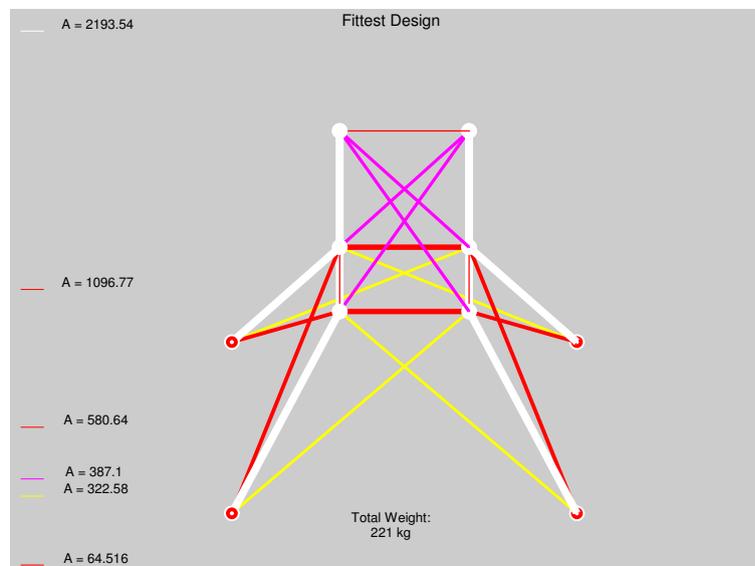


Figure 5.13: 25-bar space truss fittest design

5.8 212-Bar truss tower model

The 212-bar lattice tower investigated in this section has 15 design variables. The design variables are generally larger for real life structures, however, to demonstrate the functionality of the studied and proposed multi-objective LCC optimization strategy, 15 design variables are adequate and reasonable as well as satisfactory results have been obtained. The results obtained in this study were checked and fall within the feasible domain with no constraint violations. This check was done by carrying out a second order structural analysis, as well as a stability/buckling analysis in the PROKON structural analysis software package. A second order analysis takes the P- Δ effects due to the deformation of slender, structural elements or systems into consideration. The P- Δ effect is the increased load effect due to the relative large deformation and is solved incrementally.

Figure 5.14 shows a schematic representation of a complete WECS with a lattice type tower. The isolated support tower structure with its major dimensions and measurements is shown in Figure 5.15, has a total height of 24 *m* and consists of 212 truss elements with a total of 68 nodes and 204 degrees of freedom.

Table 5.11 includes all the nodal coordinates for nodes 1 to 40 and Table 5.12 contains the nodal coordinates for nodes 41 to 68. All coordinates are given in meters.

For the optimization of the 212-bar lattice tower, a simple quasi-static loading is imposed on the structure. The actual nodal loads of the rotor on the tower can be determined using the specifications in IEC 61400 - Part 1:2005 for a certain geographical location's wind characteristics, associated wind load cases and load combinations. Refer to Appendix C for the details on wind load calculations on a wind turbine tower structure. However, for this investigation only a pair of horizontal force vectors is applied to the uppermost nodes of the tower. For a more realistic optimization, the detailed approach on various load cases from IEC 61400 - Part 1:2005 should be included in the design process which falls beyond the scope of this research. The following load cases are required for structural design by IEC 61400 - Part 1:2005 under normal design situations and appropriate extreme external conditions, fault design situations and appropriate external conditions and transportation, installation and maintenance design situations and appropriate external conditions:

- Gravitational and inertial loads.
- Aerodynamic loads.

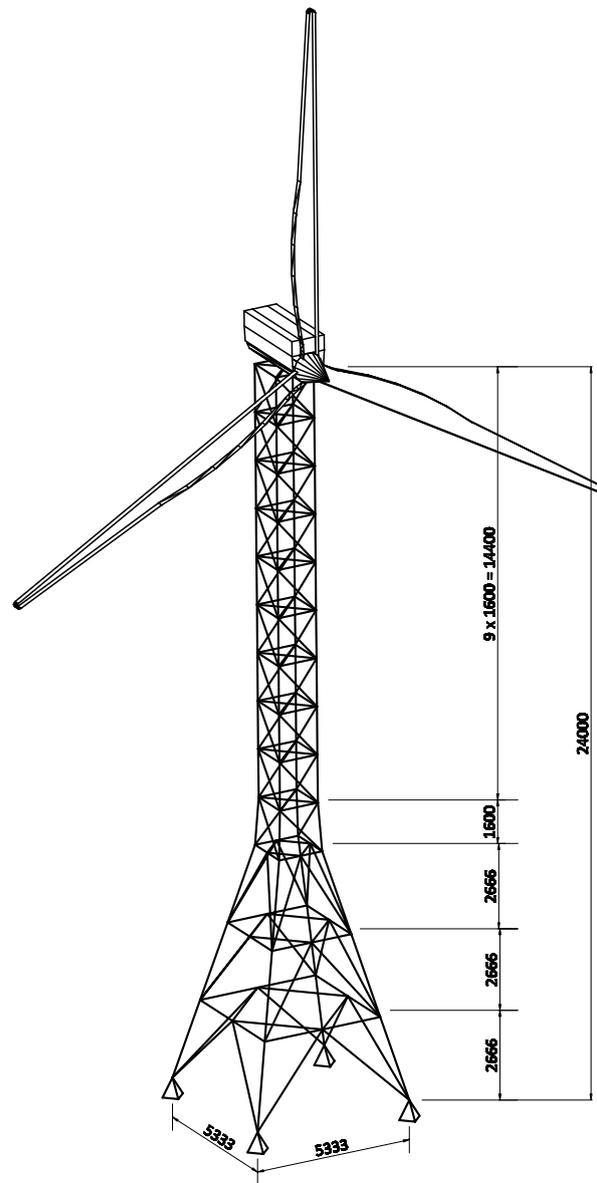


Figure 5.14: Complete schematic representation of the 212-bar wind turbine lattice tower and system

- Actuation loads.
- Other loads such as wake, ice and impact loads.

Table 2 in IEC 61400 - Part 1:2005 provides detailed guidance on the different design load cases to be considered. Clause 7.5 in the document also describes a set of possibly relevant loads that need to be identified and taken into account.

For this investigation, an arbitrary horizontal nodal load of $P = 10 \text{ kN}$ is applied at

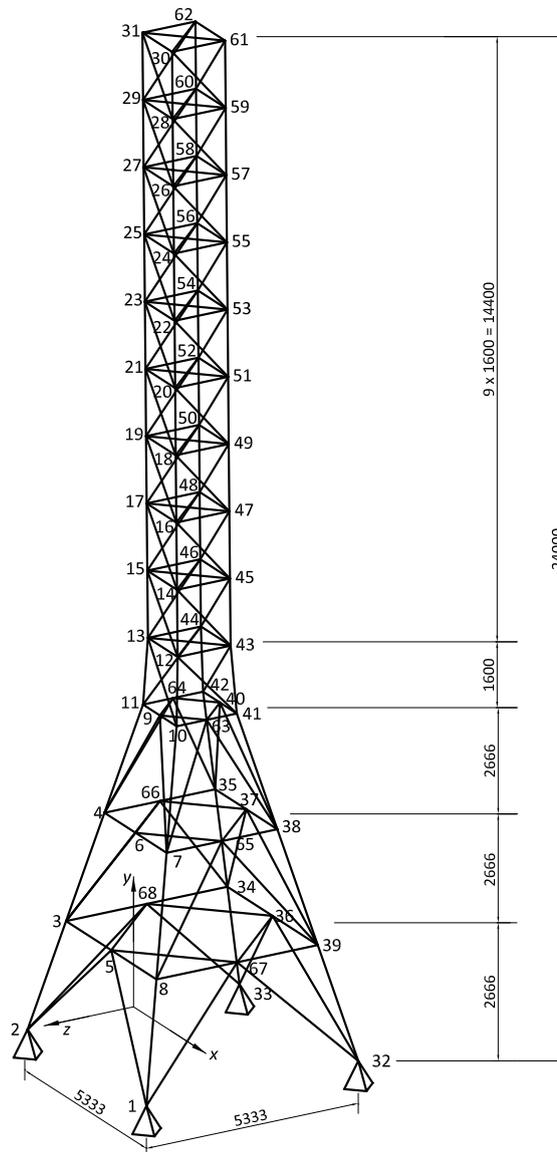


Figure 5.15: Schematic representation of the isolated 212-bar wind turbine lattice tower

nodes 31 and 64 each in the global X -direction at the top of the structure. The focus of this investigation entails the study and application of a multi-objective life-cycle cost optimization, rather than the detailed and exact determination of the design loads and respective load cases.

The nodal supports are similar to the benchmark problem. Translational restraints are provided at each of the four nodes in Table 5.13.

Table 5.11: 212-bar truss tower nodal coordinates 1 to 40

Node number	X-Coord. (m)	Y-Coord. (m)	Z-Coord. (m)
1	5.3333334	0.0	2.6666668
2	0.0	0.0	2.6666668
3	0.6370081	2.6666666	2.029658767
4	1.2740162	5.3333333	1.392650667
5	2.6666667	2.6666666	2.029658767
6	2.6666667	5.3333333	1.392650667
7	4.0593172	5.3333333	1.392650667
8	4.6963253	2.6666666	2.029658767
9	2.6666667	8.0	0.755642567
10	3.4223091	8.0	0.755642567
11	1.9110243	8.0	0.755642567
12	3.3333333	9.6	0.666666667
13	2.0	9.6	0.666666667
14	3.3333333	11.2	0.666666667
15	2.0	11.2	0.666666667
16	3.3333333	12.8	0.666666667
17	2.0	12.8	0.666666667
18	3.3333333	14.4	0.666666667
19	2.0	14.4	0.666666667
20	3.3333333	16.0	0.666666667
21	2.0	16.0	0.666666667
22	3.3333333	17.6	0.666666667
23	2.0	17.6	0.666666667
24	3.3333333	19.2	0.666666667
25	2.0	19.2	0.666666667
26	3.3333333	20.8	0.666666667
27	2.0	20.8	0.666666667
28	3.3333333	22.4	0.666666667
29	2.0	22.4	0.666666667
30	3.3333333	24.0	0.666666667
31	2.0	24	0.666666667
32	5.3333334	0.0	-2.6666668
33	0.0	0.0	-2.6666668
34	0.6370081	2.6666666	-2.029658767
35	1.2740162	5.3333333	-1.392650667
36	2.6666667	2.6666666	-2.029658767
37	2.6666667	5.3333333	-1.392650667
38	4.0593172	5.3333333	-1.392650667
39	4.6963253	2.6666666	-2.029658767
40	2.6666667	8.0	-0.755642567

Table 5.12: 212-bar truss tower nodal coordinates 41 to 68

Node number	X-Coord. (m)	Y-Coord. (m)	Z-Coord. (m)
41	3.4223091	8.0	-0.755642567
42	1.9110243	8.0	-0.755642567
43	3.3333333	9.6	-0.666666667
44	2.0	9.6	-0.666666667
45	3.3333333	11.2	-0.666666667
46	2.0	11.2	-0.666666667
47	3.3333333	12.8	-0.666666667
48	2.0	12.8	-0.666666667
49	3.3333333	14.4	-0.666666667
50	2.0	14.4	-0.666666667
51	3.3333333	16.0	-0.666666667
52	2.0	16.0	-0.666666667
53	3.3333333	17.6	-0.666666667
54	2.0	17.6	-0.666666667
55	3.3333333	19.2	-0.666666667
56	2.0	19.2	-0.666666667
57	3.3333333	20.8	-0.666666667
58	2.0	20.8	-0.666666667
59	3.3333333	22.4	-0.666666667
60	2.0	22.4	-0.666666667
61	3.3333333	24.0	-0.666666667
62	2.0	24	-0.666666667
63	3.4223091	8.0	0.0
64	1.9110243	8.0	0.0
65	4.0593172	5.3333333	0.0
66	1.2740162	5.3333333	0.0
67	4.6963253	2.6666666	0.0
68	0.6370081	2.6666666	0.0

Table 5.13: 212-bar truss tower nodal supports

Nodal supports			
Node 1	Node 2	Node 32	Node 33
XYZ	XYZ	XYZ	XYZ

5.8.1 212-Bar truss tower single objective optimization

To verify the multi-objective optimization, the 212-bar tower was optimized as a single objective problem first, with the objective being structural weight. The same section list as for the 25-bar benchmark problem is used initially (see Table 5.6). The maximum allowable nodal deflection of the tower is calculated using SANS 10162 - Part 1:2005. The code specifies a limit equal to $H/180$. For $H = 24.0\text{ m}$, the maximum nodal deflection $\delta_{max} = 133.33\text{ mm}$.

Objectives: The objective for the 212-bar truss tower single objective optimization problem is the structural weight. It is the only objective for this specific case and the maximum allowable nodal displacement is treated as a constraint in the single objective runs.

Variables: The cross-sectional area vector. Each component of the vector represents one of the 15 design variables.

Constraints: The constraints used for this full scale case study problem are only based on the requirements from SANS 10162 - Part 1:2005. The constraints considered in the optimization are maximum allowable nodal deflection, tension resistance, compression resistance and slenderness limits.

For more detailed results on the design variable vector for the single objective optimization, refer to Table 5.14, figures 5.16 and 5.17.

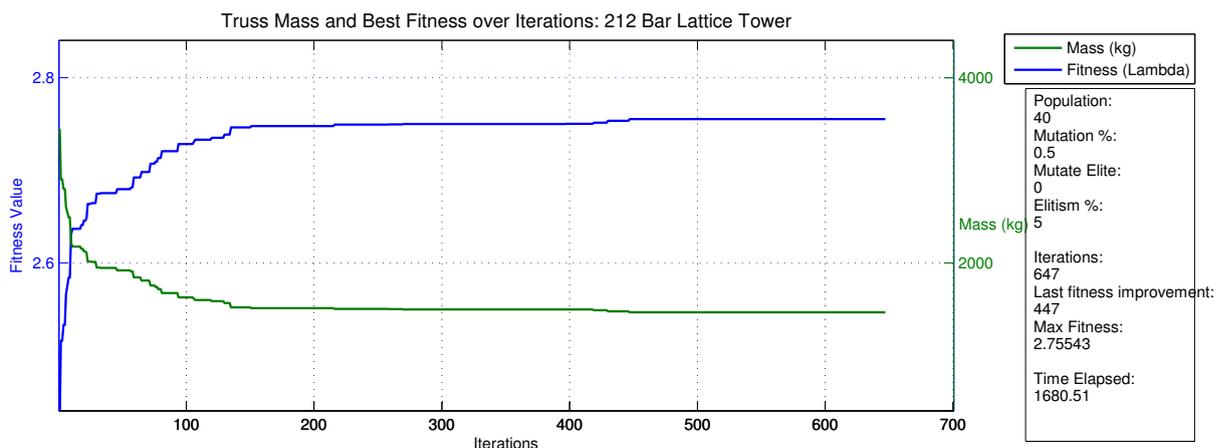


Figure 5.16: 212-bar lattice tower fitness and mass graph for weight as single objective

Table 5.14: 212-bar lattice tower run results, using the benchmark problem sections

Run results	
Time (s)	1680.51
Population	40
Iterations	647
λ	n/a
A1	1548.38 mm^2
A2	1161.29 mm^2
A3	193.55 mm^2
A4	129.03 mm^2
A5	258.06 mm^2
A6	64.52 mm^2
A7	645.2 mm^2
A8	451.6 mm^2
A9	64.52 mm^2
A10	64.52 mm^2
A11	64.52 mm^2
A12	129.03 mm^2
A13	64.52 mm^2
A14	193.55 mm^2
A15	64.52 mm^2

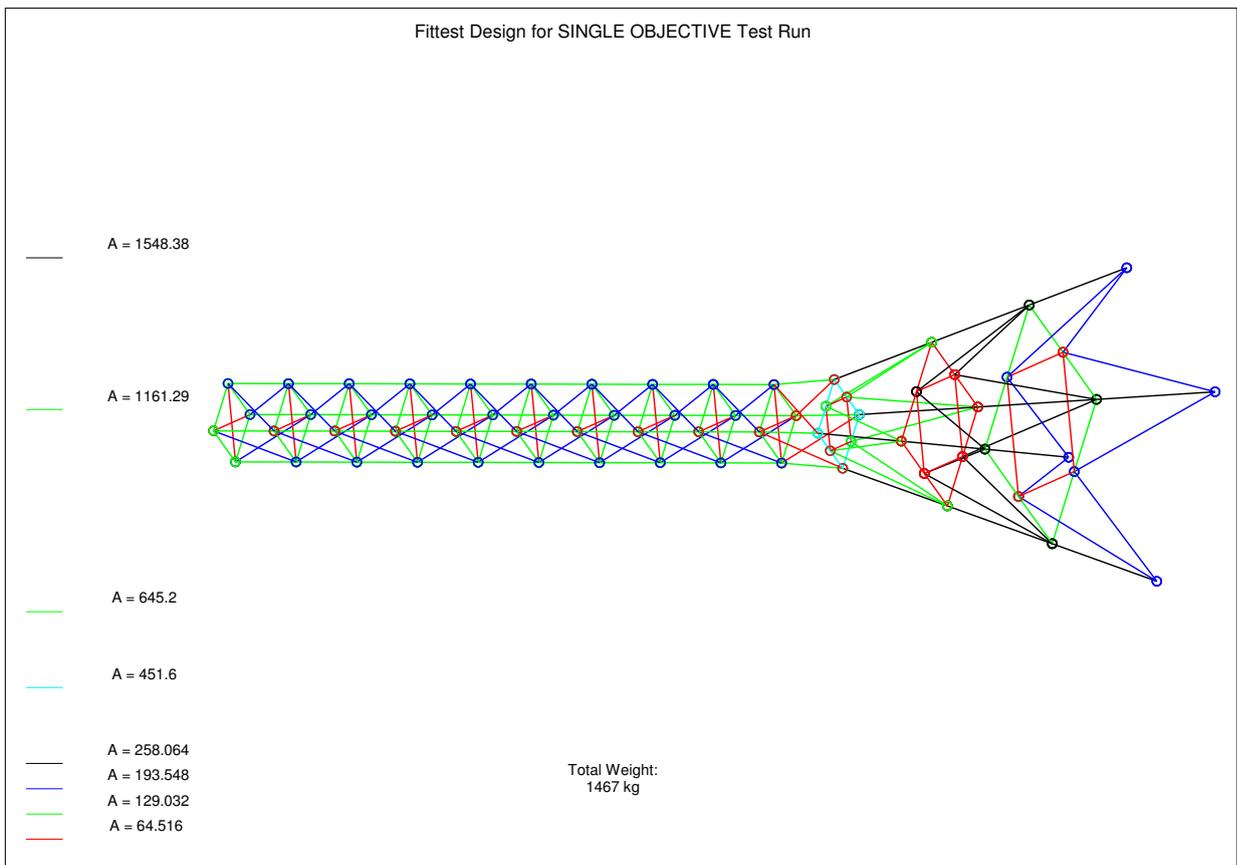


Figure 5.17: 212-bar lattice tower fittest design for weight as single objective

The single objective optimization result is expected to be an optimum result being a design that reaches a deflection close to this limitation value. The maximum nodal deflection obtained and calculated by the FEM solver for the tower equals 133.165 mm , as expected. The structural weight for the corresponding fittest design equals 1467 kg .

5.8.2 212-Bar truss tower multi-objective LCC optimization

Table 5.15 contains the geometric properties and values of the circular hollow section (CHS) list that the algorithm has available for selection during the optimization. The CHS are obtained from Table 2.34 in *South African Steel Construction Handbook* by SAISC. The optimization is repeated for hot-rolled equal leg angle sections that have been obtained from Table 2.13 in the *South African Steel Construction Handbook*.

Objectives: The objectives for the 212-bar truss tower multi-objective LCC optimization problem are the four objectives as discussed in second part of the four-bar example. The objectives are structural weight W_j , cost of the rolled section C_j , total exposed perimeter of the sections P_j and the maximum nodal deflection d_j . Note that the deflection is treated as an objective and not as a constraint in the multi-objective approach.

Variables: The cross-sectional area vector. Each component of the vector represents one of the 15 design variables.

Constraints: The constraints used for this full scale case study problem are based on the requirements from SANS 10162 - Part 1:2005. These are the same as described in section 5.8.1. The constraints considered in the optimization are tension resistance, compression resistance and the slenderness limits.

The same code specifications from SANS 10162 - Part 1:2005, as described in section 5.8.1, are applicable to the maximum allowable deflection of the multi-objective optimization. $\delta_{max} = 133.33 \text{ mm}$. The multi-objective optimization result with equally important objective weights (as assumed throughout this study) is expected to be a trade-off or best compromise among the four objectives listed above.

For the optimization the user needs to specify the upper and lower limits for the fuzzy objective functions. See Table 5.16 for the details of these values. The values are discussed and explained subsequently. The material cost C_j and weight W_j objectives are commensurable objectives, i.e. they generally do not compete in terms of fitness and due to the fact that one section type is considered for the entire structure. P_j and d_j are the perimeter and deflection objectives respectively. The upper and lower limits represent the input of the decision maker/engineer in terms of the degree of rejection and acceptance as shown in Table 5.16.

When, for example, a 'mixed' section list with both circular hollow sections and equal leg angles is used in the optimization, then these objectives are still commensurable but one

Table 5.15: 212-bar truss tower section list for multi-objective investigation using circular hollow sections

Area section list				
Description	A_g	I	r	D
	($\times 10^3 \text{ mm}^2$)	($\times 10^6 \text{ mm}^4$)	(mm)	(mm)
32.0 \times 2.0	0.1885	0.0213	10.6301	32.0
34.1 \times 2.0	0.2019	0.0262	11.3817	34.1
38.0 \times 2.0	0.2268	0.0371	12.7828	38.0
42.8 \times 2.0	0.2563	0.0535	14.4423	42.8
48.4 \times 2.0	0.2915	0.0786	16.4201	48.4
60.3 \times 2.0	0.3665	0.1561	20.6349	60.3
60.3 \times 2.5	0.4540	0.1899	20.4545	60.3
63.5 \times 2.0	0.3864	0.1829	21.7550	63.5
63.5 \times 2.5	0.4791	0.2232	21.5849	63.5
76.2 \times 2.5	0.5788	0.3935	26.0719	76.2
76.2 \times 3.0	0.6899	0.4629	25.9018	76.2
88.9 \times 2.5	0.6786	0.6337	30.5598	88.9
88.9 \times 3.0	0.8096	0.7476	30.3888	88.9
101.6 \times 2.5	0.7783	0.9561	35.0483	101.6
101.6 \times 3.0	0.9293	1.1304	34.8765	101.6
114.3 \times 2.5	0.8781	1.3726	39.5372	114.3
114.3 \times 3.0	1.0490	1.6255	39.3648	114.3
127.0 \times 2.5	0.9778	1.8953	44.0263	127.0
127.0 \times 3.0	1.1687	2.2475	43.8534	127.0
139.7 \times 3.0	1.2884	3.0109	48.3424	139.7
139.7 \times 3.5	1.4976	3.4749	48.1699	139.7
152.4 \times 3.0	1.4081	3.9301	52.8315	152.4
165.1 \times 3.0	1.5278	5.0197	57.3208	165.1
177.8 \times 3.5	1.9165	7.2811	61.6368	177.8
193.7 \times 3.5	2.0914	9.4603	67.2572	193.7
219.1 \times 3.5	2.3696	13.7589	76.2008	219.1
273.1 \times 6.0	5.0347	44.9213	94.4579	273.1
323.9 \times 6.0	5.9923	75.7247	112.4146	323.9
355.6 \times 6.0	6.5898	100.7055	123.6205	355.6
406.4 \times 8.0	10.0129	198.7389	140.8841	406.4
457.0 \times 8.0	11.2846	284.4636	158.7707	457.0
508.0 \times 8.0	12.5664	392.7996	176.7993	508.0

is a function of the other. The reason is as follows: the material cost per kg for equal leg angles and CHS differ. A separate and independent investigation would be required.

For more detailed results on the design variable vector for the multi-objective optimization, refer to Table 5.17, figures 5.18 and 5.19.

Table 5.16: 212-bar lattice tower multi-objective optimization maximum and minimum objective values

Objective	Minimum	Maximum
C_j	commensurable with weight obj.	commensurable with weight obj.
W_j	460.0 <i>kg</i>	10960.0 <i>kg</i>
P_j	21.2 <i>m</i>	169.6 <i>m</i>
d_j	60.0 <i>mm</i>	133.33 <i>mm</i>

Table 5.17: 212-bar lattice tower multi-objective optimization run results using circular hollow sections

Run results	
Time (s)	3315.46
Population	40
Iterations	1754
λ	0.7869
A1	219.1 × 3.5
A2	177.8 × 3.5
A3	60.3 × 2.0
A4	42.8 × 2.0
A5	127.0 × 2.5
A6	63.5 × 2.5
A7	139.7 × 3.0
A8	114.3 × 2.5
A9	32.0 × 2.0
A10	32.0 × 2.0
A11	32.0 × 2.0
A12	42.8 × 2.0
A13	32.0 × 2.0
A14	63.5 × 2.5
A15	32.0 × 2.0

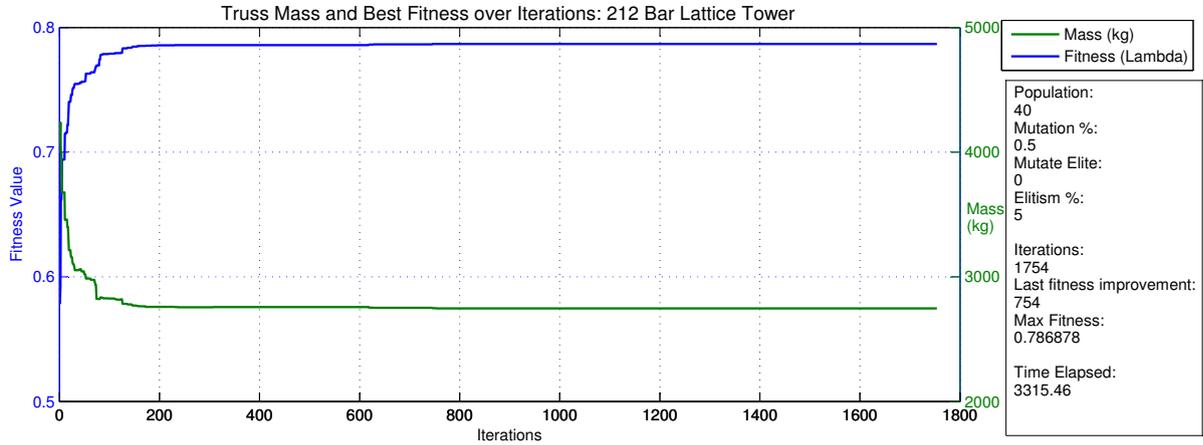


Figure 5.18: 212-bar lattice tower fitness and mass graph for multi-objective optimization using circular hollow sections

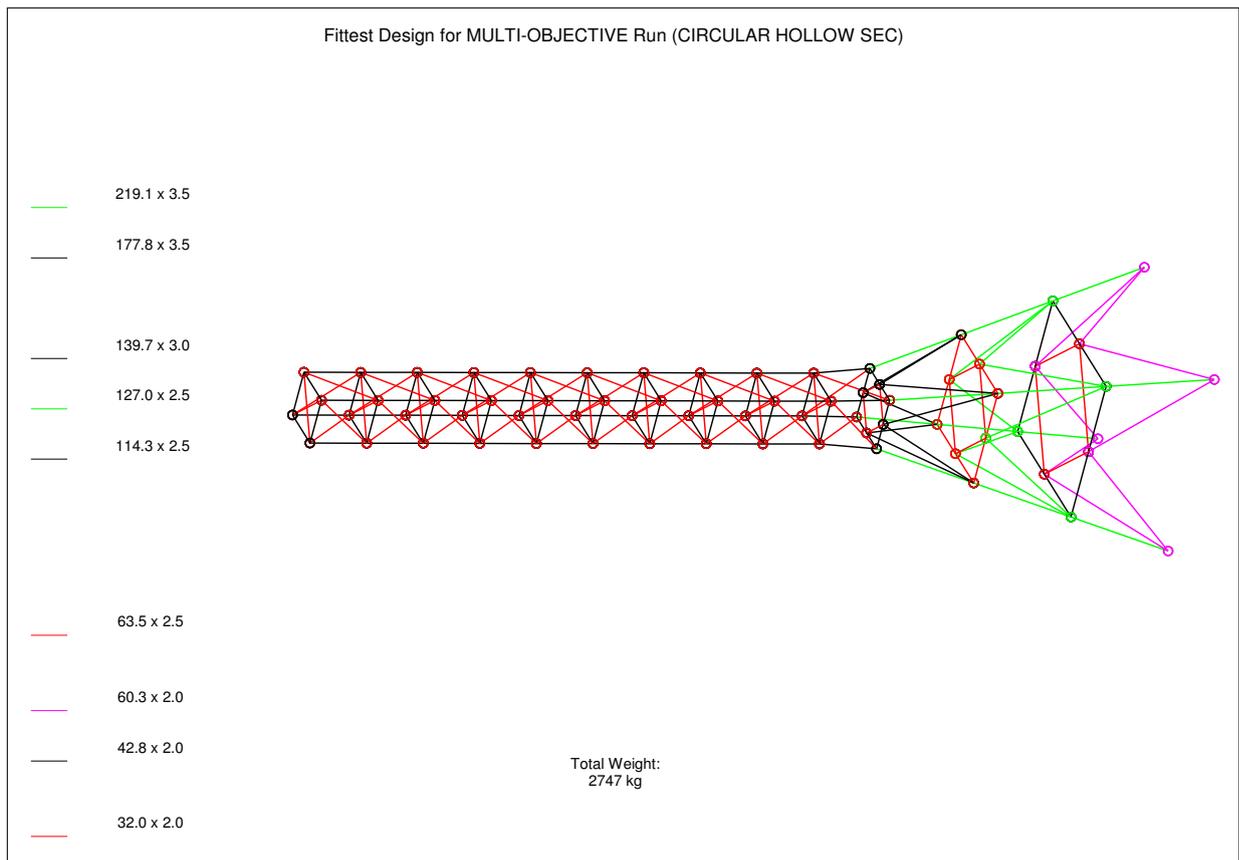


Figure 5.19: 212-bar lattice tower fittest design for multi-objective optimization using circular hollow sections

Table 5.18 contains the geometric properties and values of the equal leg angle section list that the algorithm has available for selection during the second multi-objective optimization investigation. The sections are also an extract from the *South African Steel Construction Handbook* by SAISC.

Table 5.18: 212-bar truss tower section list for multi-objective investigation using equal leg angle sections

Area section list					
Description	A_g	r_u	r_v	J	a_y
	$(\times 10^3 \text{ mm}^2)$	(mm)	(mm)	$(\times 10^3 \text{ mm}^4)$	(mm)
25×25×3	0.142	9.43	4.83	0.476	7.21
25×25×5	0.226	9.14	4.80	1.980	7.98
30×30×3	0.174	11.30	5.81	0.635	8.35
30×30×5	0.278	11.10	5.75	2.580	9.18
40×40×4	0.308	15.20	7.77	1.920	11.20
40×40×5	0.379	15.10	7.73	3.560	11.60
40×40×6	0.448	14.90	7.70	5.920	12.00
45×45×5	0.430	17.00	8.71	4.170	12.80
50×50×5	0.480	19.00	9.73	4.580	14.00
50×50×6	0.569	18.90	9.68	7.620	14.50
50×50×8	0.741	18.60	9.63	17.000	15.20
60×60×6	0.691	22.90	11.70	9.360	16.90
60×60×8	0.903	22.60	11.60	21.000	17.70
60×60×10	1.110	22.30	11.60	39.200	18.50
70×70×8	1.060	26.60	13.60	25.000	20.10
70×70×10	1.310	26.30	13.50	46.800	20.90
80×80×8	1.230	30.60	15.60	29.100	22.60
80×80×10	1.510	30.30	15.50	54.500	23.40
80×80×12	1.790	30.00	15.50	91.200	24.10
90×90×10	1.710	34.30	17.50	62.400	25.80
90×90×12	2.030	34.00	17.40	104.000	26.60
100×100×10	1.920	38.30	19.50	70.300	28.20
100×100×12	2.270	38.00	19.40	118.000	29.00
100×100×15	2.790	37.50	19.30	221.000	30.20
120×120×12	2.750	46.00	23.50	143.000	34.00
120×120×15	3.390	45.60	23.30	269.000	35.10
150×150×15	4.300	57.60	29.30	347.000	42.50
150×150×18	5.100	57.10	29.20	584.000	43.70
200×200×20	7.630	77.00	39.20	1070.000	56.80
200×200×24	9.060	76.40	39.00	1800.000	58.40

For more detailed results on the design variable vector for the multi-objective optimization,

refer to Table 5.19, figures 5.20 and 5.21.

Table 5.19: 212-bar lattice tower multi-objective optimization run results using equal leg angle sections

Run results	
Time (s)	1636.80
Population	40
Iterations	1014
λ	0.910
A1	$120 \times 120 \times 12$
A2	$80 \times 80 \times 10$
A3	$40 \times 40 \times 5$
A4	$30 \times 30 \times 5$
A5	$30 \times 30 \times 5$
A6	$25 \times 25 \times 5$
A7	$80 \times 80 \times 8$
A8	$60 \times 60 \times 6$
A9	$40 \times 40 \times 4$
A10	$25 \times 25 \times 3$
A11	$40 \times 40 \times 6$
A12	$60 \times 60 \times 6$
A13	$30 \times 30 \times 5$
A14	$80 \times 80 \times 12$
A15	$30 \times 30 \times 3$

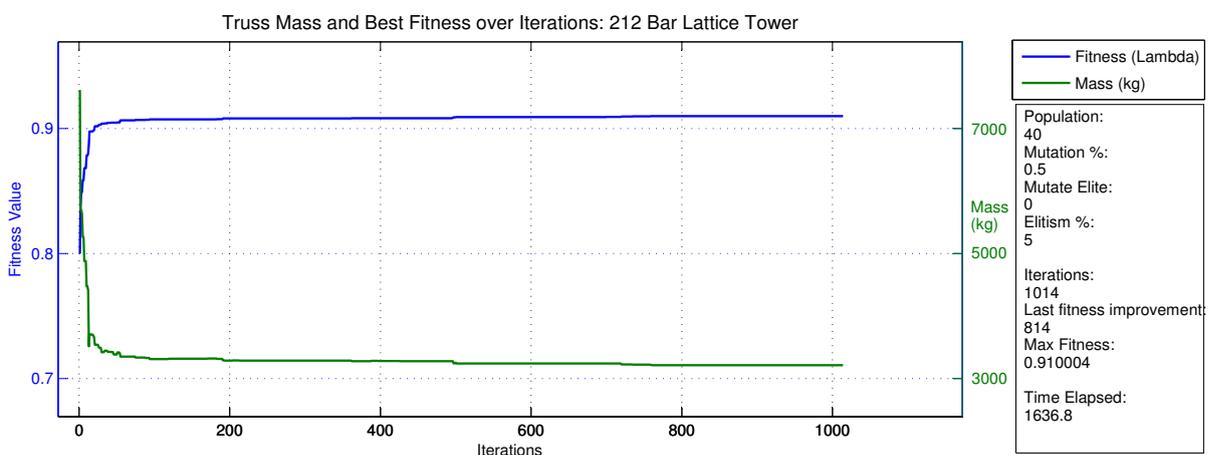


Figure 5.20: 212-bar lattice tower fitness and mass graph for multi-objective optimization using equal leg angle sections

The single objective optimization result is expected to be an optimum result being a design that reaches a deflection close to the constraint limit of 133.33 mm . The maximum nodal deflection obtained and calculated by the FEM solver for the tower equals 133.165 mm , and satisfies this requirement. The structural weight for the corresponding fittest design equals 1467 kg .

The maximum nodal deflection obtained and calculated by the FEM solver for the tower consisting of circular hollow sections equals 65.617 mm . The result obtained is as expected and not in the immediate vicinity of the boundary values supplied in Table 5.16. The result obtained represents a compromise solution or trade-off between all the equally important objectives. The structural weight for the corresponding fittest design equals 2747 kg and the overall satisfaction parameter $\lambda = 0.7869$.

The maximum nodal deflection obtained and calculated by the FEM solver for the tower consisting of equal leg angle sections equals 81.17 mm . The result obtained is as expected and not in the immediate vicinity of the boundary values. The maximum and minimum fuzzy objective values are used throughout as given in Table 5.16. Again, the result obtained with equal angle sections represents a compromise solution or trade-off between all the equally important objectives. The structural weight for the corresponding fittest design equals 3298 kg and the overall satisfaction parameter $\lambda = 0.910$.

5.10 Chapter conclusion

From the investigation it can be concluded that a well defined multi-objective formulation can be useful in decision making situations when it comes to reduction of the factors that influence and affect the LCC components of a structure that has multiple design variables. However, it needs to be pointed out that GAs are not necessarily an ultimate solution finder as they can be computationally expensive and relatively time consuming. Their ability to deal with problems of the nature described in this thesis is satisfactory and can be effectively applied to real world problems. The drawback concerning extremely high sensitivity of the algorithm on the value of the objective membership function's upper and lower constraint values is not very suitable. The fittest design obtained is the best possible result from the search subject to the input from the decision maker. λ therefore represents the 'level of conformity' given the membership function constraint values or preference values. Given this criterion, the tower model from equal leg angle sections has a higher 'level of conformity' than the tower model from CHS, given the specific preference values in Table 5.16 used for both cases. An investigation into the determination of realistic and

applicable membership function upper and lower constraint values is required and limited scope and time did not allow for this in this study.

The structural mass obtained for the equal leg angle fittest design is substantially higher than that for the CHS fittest design. The CHS fittest design is obtained after 3315.46 seconds and the equal leg angle fittest design after 1636.8 seconds. GAs make use of stochastic processes, as mentioned, and a difference in computation time can be expected due to the inherent randomness involved. The difference between these two fittest designs is 551 *kg*. This corresponds to a difference of about 20% in weight. It should be noted that the optimization evaluates the other objectives according to the min-max formulation, where the lowest membership value of all objectives represents the fitness at that time during the search and given the same information from Table 5.16, a better fitness is obtained subject to these specific values. Choosing a different set of upper and lower membership function constraints (preference values) may result in a totally different result.

CHS are the most suitable structural elements when it comes to compressive capacity. The drawback is the difficulty associated with the connection design and fabrication. Connections for angle sections are easier to manufacture and less costly when compared to CHS. In general, structures from CHS are aesthetically more appealing. However, there exist various other steel profiles, or combinations of profiles (starred-angles for example), that can be used in lattice type tower designs. Figure 5.22 is a graphical representation of various hollow and open steel sections under compression loads for a constant buckling length of $KL = 3 m$. It can be observed that the CHS perform the best, having the highest compressive capacity due to the geometric properties. On the vertical axis in the figure, the applied load to gross sectional area ratio is shown, the horizontal axis represents the mass per unit length. The contour values represent the capacity for the constant buckling length and is a function of the geometric properties of the various sections.

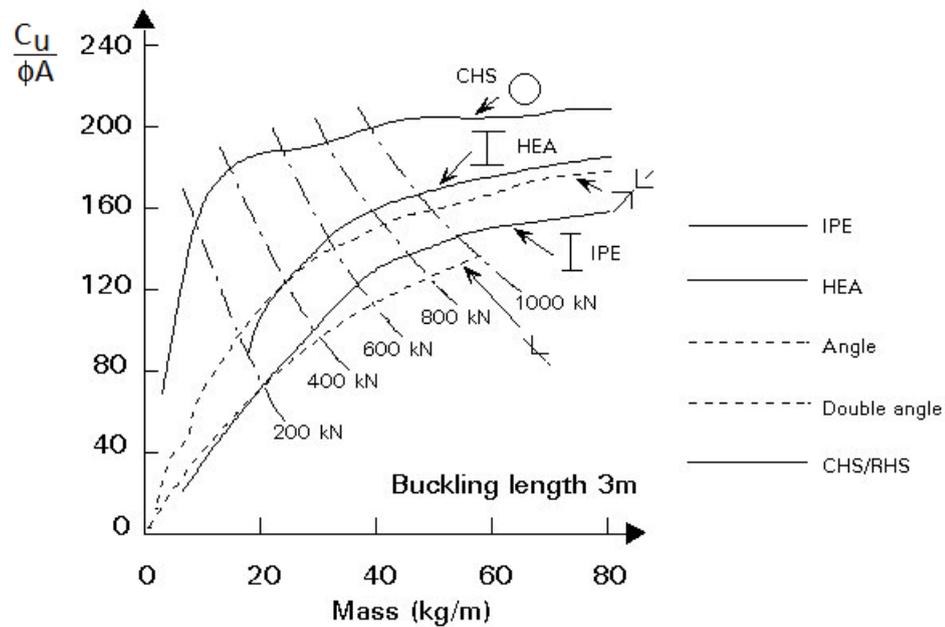


Figure 5.22: Comparison of the masses of hollow and open steel sections under compression in relation to the load (Wardenier, 2001).

However, it needs to be emphasized that the handling, cutting and fabrication of connections for CHS is far more expensive in terms of cost than those for angle sections. The increased cost for connections eliminate the cost benefits of the reduced material weight when using CHS.

Chapter 6

Conclusion and Future Perspectives

6.1 General conclusion

The optimization of a structure is the inverse of more conventional structural design approaches. When we consider a tower structure, the conventional approach would be to design the structure, analyze it for a certain set of load cases and combinations thereof and redesign it when a constraint is violated. In this thesis, an attempt was made to partly investigate the theory, applicability and development of a multi-objective life cycle cost optimization using a genetic algorithm. The work done shows that Genetic Algorithms are a robust and efficient method for optimization given the ability to relatively quickly identify optimum or near optimum solutions to a problem.

6.2 Monopole type tower LCC optimization conclusion

The monopole tower cost optimization is based on the material cost associated with a section of a tower segment only, given a set of loads and constraints as described in chapter 4. Various simplifications are assumed and the focus of this work is on the lattice type tower structure.

Given the loads, constraints and two variables, a clear contour plot is the result from which the optimum solution can be read-off relatively easily. Thus, the single objective graphical optimization with two variables is a good technique to solve such a problem effectively.

6.3 Lattice type tower LCC optimization conclusion

- The applications for numerical optimization have increased dramatically with the improved computational capabilities. This is of great value and has the potential to improve the value in designs. The concept of value plays a central role in decision making theory, namely the measure of what is good or desirable about a design.
- A necessary condition for a candidate optimum solution to a multi-objective problem is that it is not dominated, i.e. that it is Pareto optimal.
- GAs are very robust and can handle almost all types of fitness landscapes and also even a mixture of real and discrete parameters, but are generally associated with a high computational cost.
- There is no direct answer to which optimization method is the best for a given problem. It is generally a matter of opinion and depends on the nature of the problem and available software.
- To optimize a structural system, the method selected must be easily applied to various different design problems, use a minimum of auxiliary design information and attempt to find the global optimum using discrete variables. GAs satisfy these criteria and are robust and powerful techniques.
- Because GAs use the basic principles from evolutionary theory in biology, such as survival of the fittest, genetic crossover and mutation to name a few, they can be regarded as generic and can be applied to a variety of problems. The general and most challenging part is the formulation and implementation of a fitness function. The min-max or λ -formulation proved to be a useful technique for a multi-objective problem.
- Multi-objective optimization is doubtlessly a very important research topic for engineers, not just because of the multi-objective nature of real-world problems, but also owing to many more unresolved questions.
- In multi-objective optimization, there is no universally accepted definition of the ‘optimum’ such as in single objective optimization. This makes result comparison of different methods difficult.
- The decision of what the ‘best’ or ‘optimum’ solution or answer relies on the decision maker. Hence, optimize means, finding such a solution which would represent the values of all objective functions that are acceptable to the decision maker or engineer.

- Evolutionary Algorithms, such as a GA are desirable to solve multi-objective optimization problems because they work on an entire population of solutions and this could allow the user to find an entire set of Pareto optimal solutions in a single run. Recall that a Pareto optimal set of solutions is a set of so called non-inferior or non-dominated solutions.
- The min-max or λ -formulation forms part of an objective aggregating approach and by using a representative scalar fitness, it is avoided that one objective dominates the other when compared to the multiplicative or additive approaches. The combination of objectives into one scalar fitness is not only the simplest, but also one of the most efficient procedures, because no intermediate and further interaction with the decision maker is required, and if the GA converges, the result will be at least sub-optimum in most cases.
- In multi-objective optimization, it is common practice to treat the constraints as objectives and to convert the constrained problem into an unconstrained one by means of a penalty approach.
- Steel lattice type tower solutions received the most attention in this research, referring specifically to the approach of a multi-objective LCC optimization using a genetic algorithm. Such structures were popular in the past and may see a revival. When it comes to very high towers, lattice concepts are very suitable. The world's tallest wind turbine tower, the Fuhrländer Laasow 2.5 MW turbine has a 90 m diameter rotor and reaches a height of 160 m. By using mainly standard steel solutions, lattice towers compare well with other tower concepts when looking at life-cycle cost. Higher steel grades can be used to achieve lighter and taller towers. For example, upgrading the steel of a wind turbine tower structure from grade S355 to S500 can result in a weight saving of 30 %. Even with a cost increase of 20 – 25 % per unit weight, the balance is still positive due to 30 % material savings. Additional savings may result from lower transport and construction cost (World Steel Association, 2012). Higher steel grades are generally less ductile and this raises the need to investigate fatigue related and fracture failure modes that are dangerous and can happen spontaneously.
- The optimization was carried out for single load case or combination. An investigation into a possible way to incorporate multiple load cases in optimization is required and little or no literature is available. Element grouping or structural symmetry is advantageous to overcome certain aspects of different load cases and combinations but alternative approaches need to be deliberated.

- Weighting of different objectives gives very different results. The importance the objectives remains with the design engineer or decision maker. One possible approach would be to assess the current or present value cost of the LCC factors and weigh the respective objectives according to their respective ratios.
- Another important factor in LCC optimization is also the consideration of different technologies and construction alternatives, for example a hybrid structure. At a specific geographical location a certain so called optimum design may not be optimum anymore when considered at another location.
- The optimum results obtained were also very sensitive to the type of section used in the section list for the algorithm. A good example was demonstrated using the two different types of CHS and equal leg angles. The CHS solution obtained had a lower weight than the corresponding equal leg angle model. This can be attributed to the difference in the geometric properties of the sections and therefore their behavior and capacity in the structure. Refer to Figure 5.22 that illustrates the relationship between the compressive capacity for various cross-sections under a certain compressive load. Take note of the relative good performance of the CHS when compared to other sections with the same cross-sectional area.
- Concluding, the main issue which arises when considering fuzzy logic as an objective aggregating approach, is the incorporation of the users or decision maker's preference information. The definition of an appropriate membership function for LCC optimization has been followed as proposed by Adeli and Sarma (2006) and has proven effective to manage the uncertainties implied in the multi-objective decision making. This is underlined by Coello Coello *et al.* (2007), where the authors state that the membership function definition still remains a key issue when using fuzzy logic to incorporate the preference information of the decision maker.

Optimization problems with many design variables or where many potential solutions exist, the size of the search space increases rapidly. A small increase in the length of the bit string leads to an exponential growth in size of the search space. The 25-bar problem has eight design variables and therefore a bit string length of 32 to represent the eight variables with a choice of 30 sections. To illustrate this, consider a 50% increase in bit string length to 48 bits, perhaps for more design variables or a larger list of cross-sections, would result in a 1024% increase in the search space.

6.4 Future perspectives and recommendations

- Investigation into hybrid GA search techniques for improved search as it is difficult for a simple GA to still improve its fitness after many iterations. Although a near optimum is often found at that stage, chances of finding the true optimum can be improved. In other words, hybrid methods enhance the convergence capacity of a GA.
- Incorporate a second order analysis for stability and P- Δ effects to measure the performance or fitness of a structure more accurately.
- Implement other objective aggregating methods used in multi-objective optimization and compare the effects and perform sensitivity analysis concerning the variation in the parameters and deviation. Different methods used in optimization have different advantages and features and it is therefore attractive to also consider hybrid methods.
- Perform a life-cycle cost assessment on a real world structure and determine the contributions of various cost components to calibrate possible input parameters from the decision maker.
- The environmental impact of a lattice tower should be assessed in terms of the ‘Global warming Potential’, measured in CO₂e and in ‘Primary Energy Consumption’ for comparison to different material types and technologies.
- Develop and extend the exploration capacity of the multi-objective optimization tool to make it more universal. For example, the development of FEM software that has built-in features to solve single objective and multi-objective optimization problems.
- Include and formulate vibrational constraints such as mode shape frequency limits and ranges in the constraint formulations for both lattice and tubular (or monopole type towers).
- Connections form an important aspect when it comes to lattice towers. Often, High Strength Friction Grip (HSFG) bolts are necessary to accommodate vibrations and the dynamic behavior of such a tower structure. An investigation into the effect on the LCC using HSFG bolts or alternative connections can be investigated.
- A reliability-based design optimization with multiple objectives is also recommended, as many modern and recent design codes are based on the principles of reliability that accommodate uncertainties in the load effects and resistances.

List of References

- Adeli, H. and Sarma, K. (2006). *Cost Optimization of Structures*. John Wiley & Sons,Ltd., Chichester.
- Anderson, J. (2001). *A survey of multi-objective optimization in engineering design*. 581 83 Linköping, Sweden.
- Appelo, S. (2012). *Structural Optimization via Genetic Algorithm*. Masters Thesis, University of Stellenbosch, Stellenbosch.
- AWEA (2011). *Wind Energy Industry Manufacturing Supplier Handbook*. AWEA - BlueGreen Alliance - GLWN, 1501 M Street, Suite 1000, Washington DC, 20005.
- Beale, E. (1988). *Introduction to optimization*. John Wiley & Sons, Chichester, West Sussex.
- Bellman, R. and Zadeh, L. (1970). Decision-making in a fuzzy environment. *Journal of Management Science*, , no. 123(10), pp. 141–164.
- Brown, C. and Yao, J. (1983). Fuzzy sets and structural engineering. *Journal of Structural Engineering*, , no. 109(5), pp. 1408–1418.
- Cencelli, N. (2006). *Aeodynamic Optimization of a small scale Wind Turbine Blade for low Wind Speed Conditions*. Masters Thesis, University of Stellenbosch, Stellenbosch.
- Cleary, B., Duffy, A. and O’Connor, A. (2012). *Incorporation of Life-Cycle Models in determining Optimal Wind Energy Infrastructural Provision*. Dublin.
- Coello, C., Rudnick, M. and Christiansen, A. (1994). Using genetic algorithms for optimal design of trusses. *IEEE*, pp. 88–94.
- Coello Coello, C. and Christiansen, A. (2007). Multi-objective optimization of trusses using genetic algorithms.
- Coello Coello, C., Lamont, G. and Van Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-objective Problems*. 2nd edn. Springer, New York, USA.

- Coley, D. (1999). *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing Co. Pty. Ltd, Singapore.
- Cook, R., Malkus, D., Plesha, M. and Witt, R. (2002). *Concepts and Applications of Finite Element Analysis*. 4th edn. John Wiley & Sons Inc, New York.
- DNV/RISØ (2002). *Guidelines for Design of Wind Turbines*. Det Norske Veritas, Wind Energy Department and Risø National Laboratory, Copenhagen, 2nd edn.
- DWIA (2002). *Danish Wind Energy Association*. [Online]. <http://www.windpower.org/en/knowledge/statistics.html>. Retrieved 6th October 2012.
- Erbatur, F., Hasancebi, O., Tütüncü, I. and Kilic, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Computers and Structures*, pp. 209–224.
- Farkas, J. and Jarmai, K. (2008). *Design and Optimization of Metal Structures*. Horwood Publishing Limited. ISBN 978-1-904275-29-9.
- Ferreira, A. (2009). *MATLAB Codes for Finite Element Analysis-Solids and Structures*. Springer, Porto, Portugal.
- Frangopol, D., Lin, K. and Estes, A. (1997). Life-cycle cost design of deteriorating structures. *Journal of Structural Engineering*, , no. 123(10), pp. 1390–1401.
- Gencturk, B. and Elnashai, A. (2012). Life-cycle cost considerations in seismic design optimization of structures. *IGI Global*, , no. 1, pp. 1–22.
- IEC 61400 - Part 1:2005 (2005). *Wind Turbines- Part 1: Design requirements*. International Electrotechnical Commission, P. O. Box 131, CH-1211 Geneva 20, Switzerland.
- Jacobsohn, M. (2009). *Assessment of requirements for accidental design situations and robustness in the design of steel structures*. Masters Thesis, University of Stellenbosch, Stellenbosch.
- Jain, P. (2010). *Wind Energy Engineering*. McGraw Hill, New York.
- Jalkanen, J. (2007). *Optimum design of Steel Structures*. PhD Thesis, Tampere University of Technology, Korkeakoulunkatu 10, 33720 Tampere, Finland.
- Karpat, F. (2013). A virtual tool for minimum cost design of a wind turbine tower with ring stiffeners. *MDPI Energies*, , no. 6, pp. 3822–3840.
- Kelesoglu, O. and Ulker, M. (2005). Multi-objective fuzzy optimization of space trusses by MS-Excel. *Advances in Engineering Software*, , no. 36, pp. 549–553.
- Klansek, U. and Kravanja, S. (2007). *Cost optimization of composite I beam floor system*. American Journal of Applied Sciences, Technical Report.

- Maalawi, K. (2010). *Special Issues on Design Optimization of Wind Turbine Structures*. Cairo, Egypt.
- Negm, H.M. and Maalawi, K. (2000). Structural design optimization of wind turbine towers. *Computers and Structures*, , no. 74, pp. 649–666.
- Ott, H. and Hubka, V. (1985). Vorausberechnung der Herstellkosten von Schweisskonstruktionen (Fabrication cost calculation of welded structures). pp. 478–487. Proc.Int. Conference on Engineering Design ICED, Hamburg.
- Pahl, P. (2012). *Fundamentals of Technical Optimization*. Unpublished course notes, Institute of Structural Engineering, Department of Civil Engineering, University of Stellenbosch.
- Papadrakakis, M., Lagaros, N. and Plevris, V. (2005). Design optimization of steel structures considering uncertainties. *Engineering Structures*, , no. 27, pp. 1408–1418.
- Rao, S., Sundararaju, K., Parkash, G. and Balakrishna, C. (1992). Multi-objective fuzzy optimization techniques for engineering design. *Computers and Structures*, , no. 42, pp. 37–44.
- Retief, J. (2012). *CDSS-PSS Research Programme Report to the Steering Committee 2011*. University of Stellenbosch.
- Reynolds, G. (2009). *Genetic Optimization of Structural Systems*. Bachelor Thesis. [Online]. Retrieved 10th May 2013, University of Glasgow, <http://www.gavinreynolds.co.uk/wp-uploads/2009/03/GReynoldsUoGThesis2009.pdf>.
- Sagol, E. (2010). *Site Specific Design Optimization of Horizontal Axis Wind Turbine Based on Minimum Cost of Energy*. Masters Thesis, Middle East Technical University.
- SANRAL (2002). *Code of Procedure for the Planning and Design of Highway and Road Structures in South Africa*. SANRAL, P. O. Box 415, Pretoria 0001, South Africa.
- SANS 10160 - Part 1:2011 (2011). *Basis of structural design and actions for buildings and industrial structures: Basis of structural design*. SABS, Private bag X1, Pretoria 0001, South Africa.
- SANS 10162 - Part 1:2005 (2005). *The structural use of steel: Limit-state Design of hot-rolled steelwork*. SABS, Private bag X1, Pretoria 0001, South Africa.
- Sarma, K. and Adeli, H. (2002). Life-cycle cost optimization of steel structures. *International Journal for Numerical Methods in Engineering*, , no. 55, pp. 1451 – 1462.
- Sørensen, J. (2007). Structural reliability aspects in design of wind turbines. *Aspects of Structural Reliability*, pp. 91–100.

- Togan, V. and Daloglu, A. (2008). An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers and Structures*, pp. 1204–1218.
- UITP (2002). *International Association of Public Transport*. [Online]. <http://www.uitp.org/mos/corebrief/LCCOptimisation-EN.pdf-en.pdf>. Retrieved 4th January 2013.
- Uys, P., Farkas, J., Jarmai, K. and van Tonder, F. (2006). Optimization of a steel tower for a wind turbine structure. *Engineering Structures*, , no. 29, pp. 1337 – 1342.
- Venkatamaran, P. (2002). *Applied optimization with MATLAB Programming*. John Wiley & Sons,Ltd., New York, USA.
- Vrouwenvelder, A., Stieffel, U. and Harding, G. (2005). *Eurocode 1, Part 1 - 7, Accidental actions: Background document, first draft*. Unpublished.
- Wang, G. and Wang, W. (1985a). Fuzzy optimum design of structures. *Engineering Optimization*, , no. 8, pp. 291–300.
- Wardenier, J. (2001). *Hollow sections in structural applications*. CIDECT Free Educational Copy, Delft University of Technology. [Online]. Retrieved 5th June 2013, <http://www.cidect.org/en/Publications/HPPraxis.php>.
- Wilson, J., Wagaman, D., Veshosky, C., Shi, C., Adury, P. and Beidleman, A. (1997). Life-cycle engineering of bridges. *Microcomputers in Civil Engineering*, , no. 12(6), pp. 445–452.
- World Steel Association, 2012 (2012). *Steel Solutions in the Green Economy - Wind Turbines*. Brussels, Belgium.
- WWEA (2009). *World Wind Energy Association*. [Online]. <http://www.wwindea.org/home/images/stories/worldwindenergyreport2009.pdf>. Retrieved 5th September 2011.
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, , no. 8(3), pp. 338–353.

Appendices

Appendix A: MATLAB Code

MATLAB code for Chapter 5

Multi-objective LCC optimization genetic algorithm tool written in MATLAB used for the investigation in this thesis is given below:

GA_Controller.m

```

%% GA Controller: Generic procedural code representing the
   standard Genetic algorithm.
function GA_Controller(problemType, maxNumIterations,
   populationSize, mutationPercent, elitismPercent, mutateElite,
   options)
% 1. Generate initial population
% 2. Loop until completion (as determined by this controller)
5 % a) Evaluate fitness of the population members
% b) Select fittest members of the population to reproduce
% c) Give birth of offspring via genetic operations to form new
   pop.
% 3. Display and save results.
%Example: GA_Controller('212 Bar Lattice Tower', 20000, 40, 0.005,
   0.05, 0)
10 % warning('off','all');
% warning;
if (exist('options', 'var') == 0)
options = [];
end
15
%Ensuring that the population size is a multiple of 2.
if (mod(populationSize,2) ~= 0)
%this will only be seen if using GA_Controller directly and the
   user specifies an
%odd population size
20 disp('The population size must be a multiple of 2.')
return;
end;
%% Set fitness function based on problem type

```

```

switch problemType
25 case '212 Bar Lattice Tower'
    trussWeight = @weight_Truss;
    trussPerimeter = @perimeter_Truss;
    trussDisplacement = @diplacement_Truss;
    fitnessFunction = @fitness_Truss;
30 sectionProperties = @getFEMtruss_sectionProperties;
    structureDiagram = @create_TrussDiagram;
    saveResultsAs = '212BarLatticeTwr';
    %bitstringLength comes from 5 bits per design variable (i.e. cross
        -sectional area),
    %15 elements, therefore 5x15 = 75 char bitstring
35 bitstringLength = 75;
    disp('Solving 212 element truss problem type.')
    otherwise
    disp('Unknown problem type')
    error('Unknown problem type','Error');
40 return
end
%setFigures();
%% Generate initial random population
%A matrix of size populationSize by bitstringLength with values
    between 0 and 1
45 populationArray = logical(floor(rand(populationSize,
    bitstringLength) * (2)));
    %populationArray = randint(populationSize, bitstringLength, [0 1])
    ;
    %% Begin loop, set termination conditions, allowing for early
        termination
    numIterations = 0;
    continueLoop = true;
50 countSinceLastFitnessImprovement = 0;
    iterationBestFitnessArray = [];
    tic; %start the clock
    while continueLoop == true
    % Increment the iteration number (there is no increment operator
        ...)
55 numIterations = numIterations + 1;
    % disp('Iteration:')
    % disp(numIterations)
    % disp('Population Array:')
    % disp(populationArray)
60 fitnessArray = zeros(populationSize,2, 'double');
    %% For each member/row of the population array, evaluate fitness
    for popNum=1:populationSize
    individual = populationArray(popNum, :);
    fitnessArray(popNum) = popNum;
65 fitnessArray(popNum,2) = fitnessFunction(individual, popNum,
        options);

```

```

end;
%% GA Operations
%sort the population array by the fitness value
[sortedFitnessArray, sorter] = sort(fitnessArray(:, 2), 'descend')
;
70 populationArray = populationArray(sorter,:);
% disp('Sorted Population:')
% disp(populationArray)
fittestIndividual = populationArray(1,:);
bestFitness = sortedFitnessArray(1,:);
75 %Store the fittest individual if better than previous fitnesses
weight = trussWeight(fittestIndividual, popNum, options);
if (bestFitness >= max(iterationBestFitnessArray))
fittestIndividualOfRun = fittestIndividual;
end
80 disp('Fitness Array:')
disp(fitnessArray(:,2))
disp('Sorted Fitness Array:')
disp(sortedFitnessArray)
iterationWeight(numIterations,:) = weight;
85 iterationBestFitnessArray(numIterations,:) = bestFitness;
iterationAverageFitnessArray(numIterations,:) = mean(
    sortedFitnessArray);
newPopRemaining = populationSize;
%% Elitism
%Elitism is specified as a percentage (0.0 to 1.0) of the
    population
90 %Get x fittest members and add straight to new population.
%The number selected via elitism must be multiples of 2.
if elitismPercent ~= 0
elitism = ceil(elitismPercent * populationSize/2)*2;
[newPopulationArray, newPopRemaining] = GA_Elitism(elitism,
    populationArray, bitstringLength, newPopRemaining);
95 else
elitism = 0;
newPopulationArray = populationArray;
end;

100 %% Selection/Survival of the Fittest
% Using Roulette Wheel Selection
% Calculate sum of all chromosome fitnesses
totalFitness = sum(sortedFitnessArray);
% Select the next population (minus the number selected through
    via
105 % elitism). Those selected via elitism can still be selected again
    by
% the roulette wheel
selectedPopNumArray = zeros(newPopRemaining,1, 'double');
for i=1:newPopRemaining

```

```

[selectedPopNum] = GA_SelectionRouletteWheel(sortedFitnessArray,
    totalFitness,populationSize);
110 selectedPopNumArray(i,:) = selectedPopNum;
end;
selectedPopulationArray = populationArray(selectedPopNumArray,:);
%% Crossover
%A random chance to get crossed over, and a random point within
    each
115 %crossed over pair for crossover
numCrossoverPairs = newPopRemaining/2;
disp('Crossover Pairs:')
disp(numCrossoverPairs)
% Do the crossover
120 [newPopulationArray] = GA_Crossover(numCrossoverPairs,
    bitStringLength,selectedPopulationArray, newPopulationArray);
%% Mutation
[newPopulationArray] = GA_Mutation(mutationPercent, elitism,
    mutateElite,populationSize, bitStringLength, newPopulationArray
);
%% New Population
populationArray = newPopulationArray;
125 %% Termination conditions
if (numIterations == maxNumIterations), continueLoop = false; end
if (numIterations > 1) && max(iterationBestFitnessArray) ==
    iterationBestFitnessArray(numIterations-1)
    %if the best fitness has not improved then increment the count
    countSinceLastFitnessImprovement =
        countSinceLastFitnessImprovement + 1;
130 elseif (numIterations > 1) && max(iterationBestFitnessArray) ~=
        iterationBestFitnessArray(numIterations-1)
        %if the best fitness has improved, reset the count
        countSinceLastFitnessImprovement = 0;
end;
if countSinceLastFitnessImprovement == 500, continueLoop = false;
    end
135 end;
executionTime = toc;
disp('Num Iterations:')
disp(numIterations)
disp('Num Iterations since last fitness improvement:')
140 disp(countSinceLastFitnessImprovement)
%% Diagrams
%Fitness diagram
gaParameters = [populationSize; mutationPercent; elitismPercent;
    mutateElite];
results = [numIterations; countSinceLastFitnessImprovement; max(
    iterationBestFitnessArray); executionTime];
145 create_FitnessGraph(iterationBestFitnessArray,
    iterationAverageFitnessArray,iterationWeight,gaParameters,

```

```

    results, problemType)
if (max(iterationBestFitnessArray)~=0)
%Structure diagrams
fittestSolution = sectionProperties(fittestIndividualOfRun,
    options);
structureDiagram(fittestSolution, options);
150 end
%% Result saving
%No UNIX/Epoch time in MATLAB...
epochStart = [1970 01 01 0 0 0];
now = clock();
155 %UNIX Time is a signed 64bit integer...
runTime = int64(floor(etime(now, epochStart)));
%Save results to csv form, in specified results dir if provided or
    into
%current working directory otherwise (i.e. if resultsDir var is
    blank)
%save_Results(runTime, gaParameters, results, saveResultsAs,
    resultDir, options);
160 %Save figures to png images, in specified results dir if provided
    or into
%current working directory otherwise (i.e. if resultsDir var is
    blank)
%saveFigures(runTime, 'png', saveResultsAs, resultDir);

```

solver_FEMtruss.m

```

%% FEM Solver
function [a,Q,N,O,W,L,E]=solver_FEMtruss(elementArray, popNum,
    options)
% return a - Displacements (m)
% return Q - Reactions (kN)
5 % return N - Normal forces (kN)
% return O - Stresses (kN/m^2)
% return W - Weight (kg)
% return L - Length of elements (m)
% return E - Youngs Modulus (kN/m2)
10 A = cell2mat(elementArray(:,1));
%STEEL Young's Modulus is 210GPa = 210e9 N/m2 = 210e6 kN/m2
E = 210e6;
% STEEL Density is 7850 kg/m3 => Weight = Density * Volume
D = 7850;
15 [Edof, Ex, Ey, Ez, indexVec]= getFEMtruss_geometry();

K=zeros(204);
%Force in kN
20 %options Array contains nodal forces in [] format

```

```

f=zeros(204,1);
if (isempty(options))
%no options provided therefore use defaults
%f(89) = -14; % Y-dir at node 30
25 f(91) = 10; % X-dir at node 31
%f(92) = -14; % Y-dir at node 31
%f(182) = -14; % Y-dir at node 61
f(184) = 10; % X-dir at node 62
%f(185) = -14; % Y-dir at node 62
30 else
%options provided therefore use specified loads
%f(4) = options(1)*-1; f(8) = options(2)*-1;
end

35 for i=1:212
%passing ep values for each element (E is constant, A is variable)
%for grouping, element index mapping vector needs to be defined
ep=[E A(indexVec(i))];
Ke=bar3e(Ex(i,:),Ey(i,:),Ez(i,:),ep);
40 K=assem(Edof(i,:),K,Ke);

boundaryConditions = [1 0 0;2 0 0;3 0 0;4 0 0;5 0 0;6 0 0;94 0 0;
95 0 0; 96 0 0;97 0 0;98 0 0;99 0 0];
[a,Q]=solveq(K,f,boundaryConditions);
45 Ed=extract(Edof,a);
% Preallocating the N array, else it will grow incrementally...
N=zeros(212,1, 'double');
for i=1:212
50 ep=[E A(indexVec(i))];
N(i,:)=bar3s(Ex(i,:),Ey(i,:),Ez(i,:),ep,Ed(i,:)); % (kN)
end

A_ = zeros(212,1, 'double');
55 for i = 1:212
A_(i,1) = A(indexVec(i));
end;
O = N./A_; %kN/m^2

60 %Length = sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2)
L = sqrt((Ex(:,2) - Ex(:,1)).^2 + (Ey(:,2) - Ey(:,1)).^2 + (Ez
(:,2) - Ez(:,1)).^2); % (m)
%Weight = Density * Volume = Density * Length * Area
W = D.*A_.*L; % (kg)

65 if (popNum == 1)
%Draw the displacements if this is the first x members of the
population
figure(1), clf, axis('equal'), hold on, axis off

```

```

maxDisp = num2str((max(abs(a))*1000)); %mm
title(['Lattice Tower FEM Model (maximum displacement = ', maxDisp
, 'mm), Scaling Factor of 10'], 'FontSize', 10);
70 %draw original and annotate with element numbers
plotpar=[3 1 0];
elnum=Edof(:,1);
eldraw3(Ex,Ey,Ez,plotpar,elnum);
%draw deformed
75 %plotpar=[1 2 1];
%eldisp3(' ',Ex,Ey,Ez,' ',Ed,3,plotpar, 10,' ');
end;

```

GA_Crossover.m

```

%% GA_Crossover: Implementation of genetic crossover.
% Select x crossover pairs from the remaining population (or all
  if no
% elitism), for each pair of parents select a crossover point and
  swap the bits
function [newPopulationArray] = GA_Crossover(numCrossoverPairs,
  bitstringLength,selectedPopulationArray, newPopulationArray)
5 newPopRemaining = numCrossoverPairs*2;
availableParents = (1:newPopRemaining)';
for pairNum = 1:numCrossoverPairs
%disp('Crossover Pair:')
%disp (pairNum)
10 %% Parent numbers
%get the parent numbers, ensuring no identical numbers are
  selected
parentNums(1,:) = floor(rand * (newPopRemaining)) + 1;
%parentNums(1,:) = randint(1,1,[1 newPopRemaining]);
j = floor(rand * (newPopRemaining)) + 1;
15 %j = randint(1,1,[1 newPopRemaining]);
while j == parentNums(1)
j = floor(rand * (newPopRemaining)) + 1;
%j = randint(1,1,[1 newPopRemaining]);
end;
20 parentNums(2,:) = j;
%disp('Parent Nums:')
%disp (parentNums)
%% Crossover point
%Crossover point is between 0 (i.e. parents swap all bits) to
25 %bitStringLength i.e. nothing is swapped
crossoverPoint = floor(rand * (bitstringLength+1));
%crossoverPoint = randint(1,1,[0 bitstringLength]);
%disp('Crossover Point:')
%disp (crossoverPoint)
30 parents = zeros(2, bitstringLength);

```

```

%% Extracting parents
for i = 1:2
parents(i, :) = selectedPopulationArray(parentNums(i), :);
end;
35 % remove the two parents from the population array after they have
    been copied out
    ...
availableParents(parentNums(1),:) = [];
%ensure the correct row is removed, i.e. if the first row removed
%was before the second row then remove the second row num - 1
40 if parentNums(1) < parentNums(2)
availableParents(parentNums(2)-1,:) = [];
else
availableParents(parentNums(2),:) = [];
end;
45 selectedPopulationArray = selectedPopulationArray(availableParents
    , :);
newPopRemaining = newPopRemaining - 2;
availableParents = (1:newPopRemaining)';
%disp('Parents:')
%disp (parents)
50 %% Do the crossover
switch crossoverPoint
case {0, bitstringLength}
%if the crossover point is 0 or the bitstringLength just
%copy the parents through as the children. The order
55 %makes no difference i.e. in the case of 0 where the
%parents should really be switched, but doesn't matter
children = parents;
otherwise
%Take the first crossoverPoint bits and transfer straight to
    children
60 children = parents(:,1:crossoverPoint);
%Swap the remaining bits
children(1,crossoverPoint:bitstringLength) = parents(2,
    crossoverPoint:bitstringLength);
children(2,crossoverPoint:bitstringLength) = parents(1,
    crossoverPoint:bitstringLength);
end;
65 %disp('Children:')
%disp(children)
%disp('====='')
%append children to the new population array
newPopulationArray = logical(vertcat(newPopulationArray,children))
    ;
70 end;
end

```

GA_Elitism.m

```

%% GA_Elitism: Selects top x number individuals from the
   population array to preserve for the next generation.
function [newPopulationArray, newPopRemaining] = GA_Elitism(
    elitism, populationArray, bitStringLength, newPopRemaining)
%create new population array. false() = logical(zeros())
newPopulationArray = false(elitism, bitStringLength);
5 for eliteNum = 1:elitism
newPopulationArray(eliteNum,:) = populationArray(eliteNum,:);
end;
newPopRemaining = newPopRemaining - elitism;
%disp (newPopulationArray)
10 end

```

GA_Mutation.m

```

%% GA_Mutation: Randomly mutate (i.e. invert) a percentage of bits
   in the new population array.
% Introduces new novel genetic features into the population and
   maintains genetic diversity.
function [newPopulationArray] = GA_Mutation(mutationPercent,
    elitism, mutateElite, populationSize, bitStringLength,
    newPopulationArray)
%Calculate the number of bits to mutate
5 mutateNumBits = ceil(mutationPercent*populationSize*
    bitStringLength);
%Set mutation limits
%If mutateElite is off, stop mutation of the Elites by setting the
%random limits to 1 + the number of elites
if mutateElite == 0
10 mutateRowStart = 1 + elitism;
else
mutateRowStart = 1;
end;
%Select mutateNumBits bits at random and invert value
15 for i = 1:mutateNumBits
mutateRow = floor(rand * (populationSize+1-mutateRowStart)) +
    mutateRowStart;
%mutateRow = randint(1,1,[mutateRowStart bitStringLength]);
mutateCol = floor(rand * (bitStringLength))+1;
%mutateCol = randint(1,1,[1 populationSize]);
20 currentBitValue = newPopulationArray(mutateRow, mutateCol);
switch currentBitValue
case 0
newBitValue = 1;
case 1
25 newBitValue = 0;
end;

```

```

newPopulationArray(mutateRow, mutateCol) = newBitValue;
end;
end

```

GA_SelectionRouletteWheel.m

```

%% GA_SelectionRouletteWheel: Uses the roulette wheel selection
   paradigm to select individuals from population
% The greater the fitness the greater the proportion of the "wheel
   " an individual is given therefore greater chance of being
   selected.
% However, does not exclude lower fitness individuals from being
   selected i.e. they have a lower chance of being selected
function [popNum] = GA_SelectionRouletteWheel(sortedFitnessArray,
   totalFitness, populationSize)
5 r = rand*totalFitness;
s = 0;
for popNum=1:populationSize
%Go through the population and incrementally sum fitnesses
%until equal to or greater than r
10 s = s + sortedFitnessArray(popNum, :);
if (s >= r)
%return this population number
return
end;
15 end;
end

```

fitness_Truss.m

```

%% fitnessTenBarTruss: Returns an objective fitness value for the
   supplied individualtruss design
function [fitness] = fitness_Truss(individual, popNum, options)
elementArray = getFEMtruss_sectionProperties(individual);
%Solver_FEMtruss solves for and returns:
5 %a: Displacement (including boundary values) (m)
%Q: Reaction force vector (kN)
%N: Element forces (kN)
%O: Stresses (kN/m^2)
%W: Weight (kg)
10 %Additionally, for convenience, it returns the following constants
   :
%L: Element Length (m)
%E: Youngs Modulus (currently a scalar) (kN/m2)
[a,Q,N,O,W,L,E]=solver_FEMtruss(elementArray, popNum, options);
perimeter = perimeter_Truss(individual, popNum, options);
15 [Edof, Ex, Ey, Ez, indexVec]= getFEMtruss_geometry();

```

```

%% SANS 10162-1 Buckling Load check
% K=1.0
A = cell2mat(elementArray(:,1));
20 ru = cell2mat(elementArray(:,2));
rv = cell2mat(elementArray(:,3));
J = cell2mat(elementArray(:,4));
yo = cell2mat(elementArray(:,7));

25 A_ = zeros(212,1, 'double');
ru_ = zeros(212,1, 'double');
rv_ = zeros(212,1, 'double');
J_ = zeros(212,1, 'double');
yo_ = zeros(212,1, 'double');
30 xo_ = zeros(212,1, 'double');

for i=1:212
A_(i,1) = (A(indexVec(i)));
ru_(i,1) = (ru(indexVec(i)));
35 rv_(i,1) = (rv(indexVec(i)));
J_(i,1) = (J(indexVec(i)));
yo_(i,1) = (yo(indexVec(i)));
end;
phi = 0.9;
40 G = 77e6;
fex = ((pi^2*E)*(1/(L./rv_(:,1)).^2))';
fey = ((pi^2*E)*(1/(L./ru_(:,1)).^2))';
ro2 = xo_.^2 + yo_.^2 + ru_(:,1).^2 + rv_(:,1).^2;
omega = 1 - ((0 + yo_.^2)./ro2);
45 fez = (G*J_)/(A_.*ro2);
feyz = ((fey + fez)/(2.*omega)).*(1 - sqrt(1 - (4.*fey.*fez.*
omega)/(fey + fez).^2));
fe1 = min(fex, fey); % Use this 'min' method for element by
element comparison
fe2 = min(fez, feyz);
fe = min(fe1, fe2);
50 fy = 355e3;
lambda = sqrt(fy/fe);
Cr = phi.*A_.*(1 + lambda'.^(2.86)).^(-1/1.34);
%Using Relational operators, to perform an element by element
comparison of
%the two arrays. Returns a logical array of the same size, with
elements set to logical 1 (true)
55 %where the relation is true, and elements set to logical 0 (false)
where it is not.
isBuckled = N <= -Cr;
isSlenderC = L./rv_ > 200;
numElementsBuckled = sum(isBuckled);
numElementsSlenderC = sum(isSlenderC);

```

```

60 numElementsBuckledPenalty = (1-numElementsBuckled/212)^2;
numElementsSlenderPenaltyC = (1-numElementsSlenderC/212)^2;
bucklingRatio=zeros(212,1, 'double');
for i=1:212
elementBucklingRatio = abs(N(i,:))/Cr(i,:);
65 if (elementBucklingRatio > 1)
elementBucklingRatio = 1;
end;
bucklingRatio(i) = elementBucklingRatio;
end;
70 elementFitness = (1-bucklingRatio).^2;
bucklingFitness = numElementsBuckledPenalty;
slendernessFitnessC = numElementsSlenderPenaltyC;
%% Yield Stress constraint
% Typical structural steel yield stress is 355MPa = 355e6 N/m2 =
  355e3 kN/m2
75 yieldStress = phi*355e3;
isYielded = 0 >= yieldStress;
isSlenderT = L./rv_ > 300;
numElementsYielded = sum(isYielded);
numElementsSlenderT = sum(isSlenderT);
80 numElementsYieldedPenalty = (1-numElementsYielded/212)^2;
numElementsSlenderPenaltyT = (1-numElementsSlenderT/212)^2;
yieldingRatio = zeros(212,1, 'double');
for i=1:212
elementYieldingRatio = abs(0(i,:))/yieldStress;
85 if (elementYieldingRatio > 1)
elementYieldingRatio = 1;
end;
yieldingRatio(i) = elementYieldingRatio;
end;
90 elementFitness = (1-yieldingRatio).^2;
yieldingFitness = numElementsYieldedPenalty;
slendernessFitnessT = numElementsSlenderPenaltyT;
%% Displacement objective
%Displacements over MAX limit penalised severely.
95 %Displacements between MIN and MAX limits penalised.
%Displacements under MIN limit OK.
maxDisplacementLimit = 133.33e-3;
minDisplacementLimit = 60.0e-3;
maxDisplacement = max(abs(a));
100 if (maxDisplacement >= maxDisplacementLimit)
displacementFitness = 0.0;
elseif (maxDisplacement > minDisplacementLimit && maxDisplacement
  < maxDisplacementLimit)
displacementFitness = (maxDisplacementLimit - maxDisplacement)/(
  maxDisplacementLimit - minDisplacementLimit);
else
105 displacementFitness = 1.0;

```

```

end;

%% Perimeter objective
%Perimeter over MAX limit penalised severely.
110 %Perimeter between MIN and MAX limits penalised.
    %Perimeter under MIN limit OK.
    maxPerimeterLimit = 169.6;
    minPerimeterLimit = 21.2;
    if (perimeter >= maxPerimeterLimit)
115 perimeterFitness = 0.0;
    elseif (perimeter > minPerimeterLimit && perimeter <
        maxPerimeterLimit)
        perimeterFitness = (maxPerimeterLimit - perimeter)/(
            maxPerimeterLimit - minPerimeterLimit);
    else
120 perimeterFitness = 1.0;
    end;

%% Weight/mass objective
weight = sum(W);
weightMax = 10960.0;
125 weightMin = 460.0;
    if (weight >= weightMax)
        weightFitness = 0.0;
    elseif (weight > weightMin && weight < weightMax)
        weightFitness = (weightMax - weight)/(weightMax - weightMin);
130 else
        weightFitness = 1.0;
    end;

%% Fitness calculation
135 fitness = min(weightFitness, min(displacementFitness,
    perimeterFitness))*bucklingFitness*yieldingFitness*
    slendernessFitnessC*slendernessFitnessT;
end

```

create_FitnessGraph.m

```

%% create_FitnessGraph: Creates a fitness over iterations graph
function create_FitnessGraph(iterationBestFitnessArray,
    iterationAverageFitnessArray, iterationWeight, gaParameters,
    results, problemType)
%% Create the fitness graph
% Create figure
5 figure2 = figure(2);
    clf;
    % Create axes

```

```

axes1 = axes('Parent',figure2,'YGrid','on','XGrid','on','Position'
    ,[0.11 0.10 0.70 0.80]);
box('on');
10 hold('all');
numIterations = results(1);
x = (1:numIterations);
% Create plot
plotyy(x, iterationBestFitnessArray(:,1),x, iterationWeight(:,1));
15 %plot(x, iterationAverageFitnessArray(:,1),'-k');
titleText = ['Weight, Best & Average Fitness over Iterations: ',
    problemType];
% Create title
title(titleText,'FontSize',12);
% Create xlabel
20 xlabel('Iterations');
% Create ylabel
ylabel('Fitness Value');
% Create light
light('Parent',axes1,'Position',[-0.03 1.0 0.001]);
25 populationSize = gaParameters(1);
mutationPercent = gaParameters(2)*100;
elitismPercent = gaParameters(3)*100;
mutateElite = gaParameters(4);
countSinceLastFitnessImprovement = results(2);
30 lastFitnessImprovement = numIterations -
    countSinceLastFitnessImprovement;
maxFitness = results(3);
executionTime = results(4);
% Create textbox
solutionDesc = {'Population:', populationSize, 'Mutation %:',
    mutationPercent, 'Mutate Elite:', mutateElite,...
35 'Elitism %:', elitismPercent, ', ', 'Iterations:',numIterations,
    'Last fitness improvement:',...
    lastFitnessImprovement, 'Max Fitness:', maxFitness, ', ', 'Time
    Elapsed:', executionTime};
annotation(figure2,'textbox',[0.82 0.15 0.17 0.70],'String',
    solutionDesc, 'FitBoxToText','off');
end

```

create_TrussDiagram.m

```

%% create_TrussDiagram
function create_TrussDiagram(fittestSolution, options)
[Edof, Ex, Ey, Ez, indexVec]= getFEMtruss_geometry();
% Create figure
5 figure3 = figure(3); clf, axis('equal'), hold on, axis off
sectionDescs = char(fittestSolution(:,5))
sectionNums = cell2mat(fittestSolution(:,6))

```

```
% get weight
[a,Q,N,O,W]=solver_FEMtruss(fittestSolution, 0, options);
10 totalWeight = [num2str(round(sum(W))) ' kg'];
uniqueSectionNums = unique(sectionNums);
numSectionsUsed = numel(uniqueSectionNums);
for i=1:numSectionsUsed
j = uniqueSectionNums(i);
15 y1 = 0.02 + 0.03*(j-1);
y2 = y1;
switch j
case 1
Colour = [1 0 0];
20 case 2
Colour = [0 1 0];
case 3
Colour = [0 0 1];
case 4
25 Colour = [0 0 0];
case 5
Colour = [1 1 0];
case 6
Colour = [1 0 1];
30 case 7
Colour = [0 1 1];
case 8
Colour = [1 1 1];
case 9
35 Colour = [1 0 0];
case 10
Colour = [0 1 0];
case 11
Colour = [0 0 1];
40 case 12
Colour = [0 0 0];
case 13
Colour = [1 1 0];
case 14
45 Colour = [1 0 1];
case 15
Colour = [0 1 1];
case 16
Colour = [1 1 1];
50 case 17
Colour = [1 0 0];
case 18
Colour = [0 1 0];
case 19
55 Colour = [0 0 1];
case 20
```

```

Colour = [0 0 0];
case 21
Colour = [1 1 0];
60 case 22
Colour = [1 0 1];
case 23
Colour = [0 1 1];
case 24
65 Colour = [1 1 1];
case 25
Colour = [1 0 0];
case 26
Colour = [0 1 0];
70 case 27
Colour = [0 0 1];
case 28
Colour = [0 0 0];
case 29
75 Colour = [1 1 0];
case 30
Colour = [1 0 1];
case 31
Colour = [0 1 1];
80 case 32
Colour = [1 1 1];
end;
% Create line
annotation(figure3,'line',[0.02 0.05],[y1 y2],'Color',Colour);
85 end;
% Create Title textboxes
annotation(figure3,'textbox',[0 0.88 1.0 0.11],...
'String',{'Fittest Design'},...
'HorizontalAlignment','center',...
90 'FontSize',12,...
'FitBoxToText','off',...
'LineStyle','none');
annotation(figure3,'textbox',[0 0.01 1.0 0.11],...
'String',{'Total Weight:', totalWeight},...
95 'HorizontalAlignment','center',...
'FontSize',10,...
'FitBoxToText','off',...
'LineStyle','none');
for i=1:numSectionsUsed
100 j = uniqueSectionNums(i);
level = 0.02 + 0.03*(j-1);
sectionDescRow = find(sectionNums ==j, 1, 'first');
sectionDesc = sectionDescs(sectionDescRow,:);
% Create textbox
105 annotation(figure3,'textbox',[0.02 level 0.2 0.03],...

```

```

'String',{sectionDesc},...
'HorizontalAlignment','center',...
'FitBoxToText','off',...
'LineStyle','none');
110 end;
elementDiameters = cell2mat(fittestSolution(:,4));
elementDiameters_ = zeros(212,1,'double');
for i = 212
elementDiameters_(i,1) = elementDiameters(indexVec(i));
115 end;
%Find the minimum element diameter
minElementDiameter = min(elementDiameters_);
numElements=212;
x=Ex';
120 y=Ey';
z=Ez';
for i=1:numElements
%based the line width of the element
lineWidth = 1;%elementDiameters_(i)/minElementDiameter
125 elementNumber = cell2mat(fittestSolution(indexVec(i),6));
xe = x(:,i); ye = y(:,i);ze = z(:,i);
switch elementNumber
case 1
plotStyle = '-ro';
130 case 2
plotStyle = '-go';
case 3
plotStyle = '-bo';
case 4
135 plotStyle = '-ko';
case 5
plotStyle = '-yo';
case 6
plotStyle = '-mo';
140 case 7
plotStyle = '-co';
case 8
plotStyle = '-wo';
case 9
145 plotStyle = '-ro';
case 10
plotStyle = '-go';
case 11
plotStyle = '-bo';
150 case 12
plotStyle = '-ko';
case 13
plotStyle = '-yo';
case 14

```

```
155 plotStyle = '-mo';
    case 15
    plotStyle = '-co';
    case 16
    plotStyle = '-wo';
160 case 17
    plotStyle = '-ro';
    case 18
    plotStyle = '-go';
    case 19
165 plotStyle = '-bo';
    case 20
    plotStyle = '-ko';
    case 21
    plotStyle = '-yo';
170 case 22
    plotStyle = '-mo';
    case 23
    plotStyle = '-co';
    case 24
175 plotStyle = '-wo';
    case 25
    plotStyle = '-ro';
    case 26
    plotStyle = '-go';
180 case 27
    plotStyle = '-bo';
    case 28
    plotStyle = '-ko';
    case 29
185 plotStyle = '-yo';
    case 30
    plotStyle = '-mo';
    case 31
    plotStyle = '-co';
190 case 32
    plotStyle = '-wo';
    end;
    plot3(xe,ye,ze,plotStyle,'LineWidth',lineWidth)
    end
```

displacement_Truss.m

```
%% displacement_Truss: Returns a total perimeter value for the
entire supplied individual truss design
function [displacement] = displacement_Truss(individual, popNum,
    options)
elementArray = getFEMtruss_sectionProperties(individual);
```

```

%Solver_FEMtruss solves for and returns:
5 %a: Displacement (including boundary values) (m)
%Q: Reaction force vector (kN)
%N: Element forces (kN)
%O: Stresses (kN/m^2)
%W: Weight (kg)
10 %Additionally, for convenience, it returns the following constants
    :
%L: Element Length (m)
%E: Youngs Modulus (currently a scalar) (kN/m2)
[a,Q,N,O,W,L,E]=solver_FEMtruss(elementArray, popNum, options);
displacement = max(abs(a))*1000;

```

perimeter_Truss.m

```

%% perimeter_Truss: Returns a total perimeter value for the entire
    supplied individual truss design
function [perimeter] = perimeter_Truss(individual, popNum, options
    )
elementArray = getFEMtruss_sectionProperties(individual);
[Edof, Ex, Ey, Ez, indexVec]= getFEMtruss_geometry();
5 %Solver_FEMtruss solves for and returns:
%a: Displacement (including boundary values) (m)
%Q: Reaction force vector (kN)
%N: Element forces (kN)
%O: Stresses (kN/m^2)
10 %W: Weight (kg)
%Additionally, for convenience, it returns the following constants
    :
%L: Element Length (m)
%E: Youngs Modulus (currently a scalar) (kN/m2)
width = cell2mat(elementArray(:,8));
15 width_ = zeros(212,1, 'double');
for i=1:212
width_(i)= width(indexVec(i));
end;
secPer = 4*width_;
20 perimeter = sum(secPer);

```

weight_Truss.m

```

%% weight_Truss: Returns a weight value for the entire supplied
    individual truss design
function [weight] = weight_Truss(individual, popNum, options)
elementArray = getFEMtruss_sectionProperties(individual);
%Solver_FEMtruss solves for and returns:
5 %a: Displacement (including boundary values) (m)

```

```

%Q: Reaction force vector (kN)
%N: Element forces (kN)
%O: Stresses (kN/m^2)
%W: Weight (kg)
10 %Additionally, for convenience, it returns the following constants
    :
    %L: Element Length (m)
    %E: Youngs Modulus (currently a scalar) (kN/m2)
[a,Q,N,O,W,L,E]=solver_FEMtruss(elementArray, popNum, options);
weight = abs(sum(W));

```

getFEMtruss_geometry.m

```

%% getFEMtruss_geometry: Static function providing Geometry
function [Edof, Ex, Ey, Ez, indexVec]=getFEMtruss_geometry()
%This static data has been moved to an include to allow for access
%to geometry data without global variables or duplicating it in
code
5 %Topology matrix Edof
%----- Num A_Dof1 A_Dof2 A_Dof3 B_Dof1 B_Dof2 B_Dof3
Edof= [1  1  2  3  22  23  24  ;
2  19  20  21  22  23  24  ;
3  19  20  21  28  29  30  ;
10 4  28  29  30  34  35  36  ;
5  34  35  36  40  41  42  ;
6  40  41  42  46  47  48  ;
7  46  47  48  52  53  54  ;
8  52  53  54  58  59  60  ;
15 9  58  59  60  64  65  66  ;
10 10 64  65  66  70  71  72  ;
11 70  71  72  76  77  78  ;
12 76  77  78  82  83  84  ;
13 82  83  84  88  89  90  ;
20 14 94  95  96  115 116 117  ;
15 15 112 113 114 115 116 117  ;
16 112 113 114 121 122 123  ;
17 121 122 123 127 128 129  ;
18 127 128 129 133 134 135  ;
25 19 133 134 135 139 140 141  ;
20 20 139 140 141 145 146 147  ;
21 145 146 147 151 152 153  ;
22 151 152 153 157 158 159  ;
23 157 158 159 163 164 165  ;
30 24 163 164 165 169 170 171  ;
25 25 169 170 171 175 176 177  ;
26 175 176 177 181 182 183  ;
27 97  98  99  100 101 102  ;
28 100 101 102 103 104 105  ;

```

```
35 29 103 104 105 124 125 126 ;
30 124 125 126 130 131 132 ;
31 130 131 132 136 137 138 ;
32 136 137 138 142 143 144 ;
33 142 143 144 148 149 150 ;
40 34 148 149 150 154 155 156 ;
35 154 155 156 160 161 162 ;
36 160 161 162 166 167 168 ;
37 166 167 168 172 173 174 ;
38 172 173 174 178 179 180 ;
45 39 178 179 180 184 185 186 ;
40 4 5 6 7 8 9 ;
41 7 8 9 10 11 12 ;
42 10 11 12 31 32 33 ;
43 31 32 33 37 38 39 ;
50 44 37 38 39 43 44 45 ;
45 43 44 45 49 50 51 ;
46 49 50 51 55 56 57 ;
47 55 56 57 61 62 63 ;
48 61 62 63 67 68 69 ;
55 49 67 68 69 73 74 75 ;
50 73 74 75 79 80 81 ;
51 79 80 81 85 86 87 ;
52 85 86 87 91 92 93 ;
53 94 95 96 106 107 108 ;
60 54 97 98 99 106 107 108 ;
55 100 101 102 106 107 108 ;
56 106 107 108 115 116 117 ;
57 109 110 111 115 116 117 ;
58 100 101 102 109 110 111 ;
65 59 103 104 105 109 110 111 ;
60 109 110 111 112 113 114 ;
61 112 113 114 118 119 120 ;
62 103 104 105 118 119 120 ;
63 118 119 120 124 125 126 ;
70 64 118 119 120 121 122 123 ;
65 97 98 99 202 203 204 ;
66 4 5 6 202 203 204 ;
67 7 8 9 202 203 204 ;
68 100 101 102 202 203 204 ;
75 69 7 8 9 196 197 198 ;
70 100 101 102 196 197 198 ;
71 10 11 12 196 197 198 ;
72 103 104 105 190 191 192 ;
73 10 11 12 190 191 192 ;
80 74 124 125 126 190 191 192 ;
75 31 32 33 190 191 192 ;
76 4 5 6 13 14 15 ;
77 1 2 3 13 14 15 ;
```

```
78 7 8 9 13 14 15 ;
85 79 13 14 15 22 23 24 ;
80 16 17 18 19 20 21 ;
81 7 8 9 16 17 18 ;
82 16 17 18 22 23 24 ;
83 10 11 12 25 26 27 ;
90 84 19 20 21 25 26 27 ;
85 25 26 27 31 32 33 ;
86 25 26 27 28 29 30 ;
87 1 2 3 199 200 201 ;
88 94 95 96 199 200 201 ;
95 89 115 116 117 199 200 201 ;
90 22 23 24 199 200 201 ;
91 19 20 21 193 194 195 ;
92 112 113 114 193 194 195 ;
93 22 23 24 193 194 195 ;
100 94 115 116 117 193 194 195 ;
95 19 20 21 187 188 189 ;
96 112 113 114 187 188 189 ;
97 28 29 30 187 188 189 ;
98 121 122 123 187 188 189 ;
105 99 16 17 18 193 194 195 ;
100 16 17 18 196 197 198 ;
101 109 110 111 196 197 198 ;
102 109 110 111 193 194 195 ;
103 25 26 27 190 191 192 ;
110 104 118 119 120 190 191 192 ;
105 34 35 36 130 131 132 ;
106 37 38 39 127 128 129 ;
107 43 44 45 133 134 135 ;
108 40 41 42 136 137 138 ;
115 109 49 50 51 139 140 141 ;
110 46 47 48 142 143 144 ;
111 55 56 57 145 146 147 ;
112 52 53 54 148 149 150 ;
113 61 62 63 151 152 153 ;
120 114 58 59 60 154 155 156 ;
115 67 68 69 157 158 159 ;
116 64 65 66 160 161 162 ;
117 73 74 75 163 164 165 ;
118 70 71 72 166 167 168 ;
125 119 79 80 81 169 170 171 ;
120 76 77 78 172 173 174 ;
121 85 86 87 175 176 177 ;
122 82 83 84 178 179 180 ;
123 91 92 93 181 182 183 ;
130 124 88 89 90 91 92 93 ;
125 88 89 90 184 185 186 ;
126 91 92 93 184 185 186 ;
```

```
127 181 182 183 184 185 186 ;
128 88 89 90 181 182 183 ;
135 129 82 83 84 85 86 87 ;
130 85 86 87 178 179 180 ;
131 175 176 177 178 179 180 ;
132 82 83 84 175 176 177 ;
133 76 77 78 79 80 81 ;
140 134 79 80 81 172 173 174 ;
135 169 170 171 172 173 174 ;
136 76 77 78 169 170 171 ;
137 70 71 72 73 74 75 ;
138 73 74 75 166 167 168 ;
145 139 163 164 165 166 167 168 ;
140 70 71 72 163 164 165 ;
141 64 65 66 67 68 69 ;
142 67 68 69 160 161 162 ;
143 157 158 159 160 161 162 ;
150 144 64 65 66 157 158 159 ;
145 58 59 60 61 62 63 ;
146 61 62 63 154 155 156 ;
147 151 152 153 154 155 156 ;
148 58 59 60 151 152 153 ;
155 149 52 53 54 55 56 57 ;
150 55 56 57 148 149 150 ;
151 145 146 147 148 149 150 ;
152 52 53 54 145 146 147 ;
153 46 47 48 49 50 51 ;
160 154 49 50 51 142 143 144 ;
155 139 140 141 142 143 144 ;
156 46 47 48 139 140 141 ;
157 40 41 42 43 44 45 ;
158 43 44 45 136 137 138 ;
165 159 133 134 135 136 137 138 ;
160 40 41 42 133 134 135 ;
161 34 35 36 37 38 39 ;
162 37 38 39 130 131 132 ;
163 127 128 129 130 131 132 ;
170 164 34 35 36 127 128 129 ;
165 28 29 30 127 128 129 ;
166 121 122 123 130 131 132 ;
167 37 38 39 124 125 126 ;
168 31 32 33 34 35 36 ;
175 169 34 35 36 133 134 135 ;
170 127 128 129 136 137 138 ;
171 43 44 45 130 131 132 ;
172 37 38 39 40 41 42 ;
173 40 41 42 139 140 141 ;
180 174 133 134 135 142 143 144 ;
175 49 50 51 136 137 138 ;
```

```

176 43 44 45 46 47 48 ;
177 46 47 48 145 146 147 ;
178 139 140 141 148 149 150 ;
185 179 55 56 57 142 143 144 ;
180 49 50 51 52 53 54 ;
181 52 53 54 151 152 153 ;
182 145 146 147 154 155 156 ;
183 61 62 63 148 149 150 ;
190 184 55 56 57 58 59 60 ;
185 58 59 60 157 158 159 ;
186 151 152 153 160 161 162 ;
187 67 68 69 154 155 156 ;
188 61 62 63 64 65 66 ;
195 189 64 65 66 163 164 165 ;
190 157 158 159 166 167 168 ;
191 73 74 75 160 161 162 ;
192 67 68 69 70 71 72 ;
193 70 71 72 169 170 171 ;
200 194 163 164 165 172 173 174 ;
195 79 80 81 166 167 168 ;
196 73 74 75 76 77 78 ;
197 76 77 78 175 176 177 ;
198 169 170 171 178 179 180 ;
205 199 85 86 87 172 173 174 ;
200 79 80 81 82 83 84 ;
201 82 83 84 181 182 183 ;
202 175 176 177 184 185 186 ;
203 91 92 93 178 179 180 ;
210 204 85 86 87 88 89 90 ;
205 13 14 15 202 203 204 ;
206 13 14 15 199 200 201 ;
207 106 107 108 199 200 201 ;
208 106 107 108 202 203 204 ;
215 209 25 26 27 187 188 189 ;
210 118 119 120 187 188 189 ;
211 10 11 12 16 17 18 ;
212 103 104 105 196 197 198 ;];

% Element coordinates
220 % Using a global coordinate matrix, a global topology matrix and
      coordxtr
% to get Ex, Ey and Ez
Coord= ...
[5.3333334          0      2.6666668      %Node 1
          0          0      2.6666668      %Node 2
225 0.6370081 2.6666666 2.029658767      %Node 3
1.2740162 5.3333333 1.392650667      %Node 4
2.6666667 2.6666666 2.029658767      %Node 5
2.6666667 5.3333333 1.392650667      %Node 6
4.0593172 5.3333333 1.392650667      %Node 7

```

```

230 4.6963253 2.6666666 2.029658767 %Node 8
    2.6666667      8 0.755642567 %Node 9
    3.4223091      8 0.755642567 %Node 10
    1.9110243      8 0.755642567 %Node 11
    3.3333333     9.6 0.666666667 %Node 12
235      2     9.6 0.666666667 %Node 13
    3.3333333    11.2 0.666666667 %Node 14
      2    11.2 0.666666667 %Node 15
    3.3333333    12.8 0.666666667 %Node 16
      2    12.8 0.666666667 %Node 17
240 3.3333333    14.4 0.666666667 %Node 18
      2    14.4 0.666666667 %Node 19
    3.3333333     16 0.666666667 %Node 20
      2     16 0.666666667 %Node 21
    3.3333333    17.6 0.666666667 %Node 22
245      2    17.6 0.666666667 %Node 23
    3.3333333    19.2 0.666666667 %Node 24
      2    19.2 0.666666667 %Node 25
    3.3333333    20.8 0.666666667 %Node 26
      2    20.8 0.666666667 %Node 27
250 3.3333333    22.4 0.666666667 %Node 28
      2    22.4 0.666666667 %Node 29
    3.3333333     24 0.666666667 %Node 30
      2     24 0.666666667 %Node 31
    5.3333334      0 -2.6666668 %Node 32
255      0      0 -2.6666668 %Node 33
    0.6370081 2.6666666 -2.029658767 %Node 34
    1.2740162 5.3333333 -1.392650667 %Node 35
    2.6666667 2.6666666 -2.029658767 %Node 36
    2.6666667 5.3333333 -1.392650667 %Node 37
260 4.0593172 5.3333333 -1.392650667 %Node 38
    4.6963253 2.6666666 -2.029658767 %Node 39
    2.6666667      8 -0.755642567 %Node 40
    3.4223091      8 -0.755642567 %Node 41
    1.9110243      8 -0.755642567 %Node 42
265 3.3333333     9.6 -0.666666667 %Node 43
      2     9.6 -0.666666667 %Node 44
    3.3333333    11.2 -0.666666667 %Node 45
      2    11.2 -0.666666667 %Node 46
    3.3333333    12.8 -0.666666667 %Node 47
270      2    12.8 -0.666666667 %Node 48
    3.3333333    14.4 -0.666666667 %Node 49
      2    14.4 -0.666666667 %Node 50
    3.3333333     16 -0.666666667 %Node 51
      2     16 -0.666666667 %Node 52
275 3.3333333    17.6 -0.666666667 %Node 53
      2    17.6 -0.666666667 %Node 54
    3.3333333    19.2 -0.666666667 %Node 55
      2    19.2 -0.666666667 %Node 56

```

```

3.3333333      20.8 -0.666666667      %Node 57
280           2      20.8 -0.666666667      %Node 58
3.3333333      22.4 -0.666666667      %Node 59
           2      22.4 -0.666666667      %Node 60
3.3333333      24 -0.666666667      %Node 61
           2      24 -0.666666667      %Node 62
285 3.4223091      8      0      %Node 63
1.9110243      8      0      %Node 64
4.0593172 5.3333333      0      %Node 65
1.2740162 5.3333333      0      %Node 66
4.6963253 2.6666666      0      %Node 67
290 0.6370081 2.6666666      0]; %Node 68

Dof=[1 2 3;      %Node 1
4 5 6;      %Node 2
7 8 9;      %Node 3
295 10 11 12;      %Node 4
13 14 15;      %Node 5
16 17 18;      %Node 6
19 20 21;      %Node 7
22 23 24;      %Node 8
300 25 26 27;      %Node 9
28 29 30;      %Node 10
31 32 33;      %Node 11
34 35 36;      %Node 12
37 38 39;      %Node 13
305 40 41 42;      %Node 14
43 44 45;      %Node 15
46 47 48;      %Node 16
49 50 51;      %Node 17
52 53 54;      %Node 18
310 55 56 57;      %Node 19
58 59 60;      %Node 20
61 62 63;      %Node 21
64 65 66;      %Node 22
67 68 69;      %Node 23
315 70 71 72;      %Node 24
73 74 75;      %Node 25
76 77 78;      %Node 26
79 80 81;      %Node 27
82 83 84;      %Node 28
320 85 86 87;      %Node 29
88 89 90;      %Node 30
91 92 93;      %Node 31
94 95 96;      %Node 32
97 98 99;      %Node 33
325 100 101 102;      %Node 34
103 104 105;      %Node 35
106 107 108;      %Node 36

```

```

109 110 111;           %Node 37
112 113 114;         %Node 38
330 115 116 117;      %Node 39
118 119 120;         %Node 40
121 122 123;         %Node 41
124 125 126;         %Node 42
127 128 129;         %Node 43
335 130 131 132;      %Node 44
133 134 135;         %Node 45
136 137 138;         %Node 46
139 140 141;         %Node 47
142 143 144;         %Node 48
340 145 146 147;      %Node 49
148 149 150;         %Node 50
151 152 153;         %Node 51
154 155 156;         %Node 52
157 158 159;         %Node 53
345 160 161 162;      %Node 54
163 164 165;         %Node 55
166 167 168;         %Node 56
169 170 171;         %Node 57
172 173 174;         %Node 58
350 175 176 177;      %Node 59
178 179 180;         %Node 60
181 182 183;         %Node 61
184 185 186;         %Node 62
187 188 189;         %Node 63
355 190 191 192;      %Node 64
193 194 195;         %Node 65
196 197 198;         %Node 66
199 200 201;         %Node 67
202 203 204;];      %Node 68

360 indexVec = [1  1  1  2  2  2  2  2  2  2  2  2  ...
                2  2  1  1  1  2  2  2  2  2  2  2  2
                ...
                2  2  1  1  1  2  2  2  2  2  2  2  2
                ...
                2  2  1  1  1  2  2  2  2  2  2  2  2
                ...
365 2  2  3  3  4  4  5  5  6  6  7  7  8
                ...
                8  3  3  4  4  5  5  6  7  7  8  8  3
                ...
                3  4  4  6  5  5  7  7  8  8  3  3  4
                ...
                4  6  6  5  5  7  7  8  8  9  9  9  9
                ...

```

```

10      10      11      11      11      11      11      11      11      11      11      11
370      11      ...
11      11      11      11      11      11      11      11      12      11      12      12
      12      ...
12      12      12      12      12      12      12      12      12      12      12      12
      12      ...
12      12      12      12      12      12      12      12      12      12      12      12
      12      ...
12      12      12      12      12      12      12      12      12      12      13      13
      13      ...
13      14      14      14      14      14      14      14      14      14      14      14
      14      ...
375      14      14      14      14      14      14      14      14      14      14      14      14
      14      ...
14      14      14      14      14      14      14      14      14      14      14      15
      15      ...
15      15      10      10      6      6];

[Ex, Ey, Ez]=cooridxtr(Edof, Coord, Dof, 2);

```

getFEMtruss_sectionProperties.m

```

%% getFEMtrusssectionProperties: Provides section properties
   specified by the bitstring of the individual provided
function [elementArray] = getFEMtruss_sectionProperties(individual
, options)
% Preallocating the elementArray, else it will grow incrementally
...
elementArray=cell(15, 8);
5 for elementNum=1:15
b = elementNum*5;
a = b - (5-1);
crossSectionNum = bin2dec(sprintf('%-1d',individual(a:b))) + 1;
%% populate the array of cross-section areas via the structural
   lookup
10 %All sections below are S355JR CHS, conforming to
   %(SANS 10162:2011, Part 1).
   % A: Area (x 10^3 mm^3)
   % J: St Venant torsional constant (x 10^3 mm^4)
   % ru: Radius of gyration about uu (mm)
15 % rv: Radius of gyration about vv (mm)
   % yo: Shear centre y-coordinate of section (mm)
   % b: leg of the section (mm)
switch crossSectionNum
case 1
20 A = 0.142;
ru = 9.43;
rv = 4.83;

```

```
J = 0.476;
desc = '25 x 25 x 3';
25 crossSectionNum = 1;
yo = 8.075;
b = 25;
case 2
A = 0.226;
30 ru = 9.14;
rv = 4.8;
J = 1.98;
desc = '25 x 25 x 5';
crossSectionNum = 2;
35 yo = 7.750;
b = 25;
case 3
A = 0.174;
ru = 11.3;
40 rv = 5.81;
J = 0.635;
desc = '30 x 30 x 3';
crossSectionNum = 3;
yo = 9.687;
45 b = 30;
case 4
A = 0.278;
ru = 11.1;
rv = 5.75;
50 J = 2.58;
desc = '30 x 30 x 5';
crossSectionNum = 4;
yo = 9.447;
b = 30;
55 case 5
A = 0.308;
ru = 15.2;
rv = 7.77;
J = 1.92;
60 desc = '40 x 40 x 4';
crossSectionNum = 5;
yo = 13.011;
b = 40;
case 6
65 A = 0.379;
ru = 15.1;
rv = 7.73;
J = 3.56;
desc = '40 x 40 x 5';
70 crossSectionNum = 6;
yo = 12.869;
```

```
b = 40;
case 7
A = 0.448;
75 ru = 14.9;
rv = 7.7;
J = 5.92;
desc = '40 x 40 x 6';
crossSectionNum = 7;
80 yo = 12.728;
b = 40;
case 8
A = 0.43;
ru = 17;
85 rv = 8.71;
J = 4.17;
desc = '45 x 45 x 5';
crossSectionNum = 8;
yo = 14.566;
90 b = 45;
case 9
A = 0.48;
ru = 19;
rv = 9.73;
95 J = 4.58;
desc = '50 x 50 x 5';
crossSectionNum = 9;
yo = 16.263;
b = 50;
100 case 10
A = 0.569;
ru = 18.9;
rv = 9.68;
J = 7.62;
105 desc = '50 x 50 x 6';
crossSectionNum = 10;
yo = 16.263;
b = 50;
case 11
110 A = 0.741;
ru = 18.6;
rv = 9.63;
J = 17;
desc = '50 x 50 x 8';
115 crossSectionNum = 11;
yo = 15.839;
b = 50;
case 12
A = 0.691;
120 ru = 22.9;
```

```
rv =      11.7;
J =   9.36;
desc =      '60 x 60 x 6';
crossSectionNum =   12;
125 yo =      19.658;
b = 60;
case 13
A =   0.903;
ru =      22.6;
130 rv =      11.6;
J =   21;
desc =      '60 x 60 x 8';
crossSectionNum =   13;
yo =      19.375;
135 b = 60;
case 14
A =   1.11;
ru =      22.3;
rv =      11.6;
140 J =   39.2;
desc =      '60 x 60 x 10';
crossSectionNum =   14;
yo =      19.092;
b = 60;
145 case 15
A =   1.06;
ru =      26.6;
rv =      13.6;
J =   25;
150 desc =      '70 x 70 x 8';
crossSectionNum =   15;
yo =      22.769;
b = 70;
case 16
155 A =   1.31;
ru =      26.3;
rv =      13.5;
J =   46.8;
desc =      '70 x 70 x 10';
160 crossSectionNum =   16;
yo =      22.486;
b = 70;
case 17
165 A =   1.23;
ru =      30.6;
rv =      15.6;
J =   29.1;
desc =      '80 x 80 x 8';
crossSectionNum =   17;
```

```
170 yo =      26.304;
    b = 80;
    case 18
    A =  1.51;
    ru =      30.3;
175 rv =      15.5;
    J =  54.5;
    desc =      '80 x 80 x 10';
    crossSectionNum =  18;
    yo =      26.022;
180 b = 80;
    case 19
    A =  1.79;
    ru =      30;
    rv =      15.5;
185 J =  91.2;
    desc =      '80 x 80 x 12';
    crossSectionNum =  19;
    yo =      25.597;
    b = 80;
190 case 20
    A =  1.71;
    ru =      34.3;
    rv =      17.5;
    J =  62.4;
195 desc =      '90 x 90 x 10';
    crossSectionNum =  20;
    yo =      29.416;
    b = 90;
    case 21
200 A =  2.03;
    ru =      34;
    rv =      17.4;
    J =  104;
    desc =      '90 x 90 x 12';
205 crossSectionNum =  21;
    yo =      29.133;
    b = 90;
    case 22
    A =  1.92;
210 ru =      38.3;
    rv =      19.5;
    J =  70.3;
    desc =      '100 x 100 x 10';
    crossSectionNum =  22;
215 yo =      32.810;
    b = 100;
    case 23
    A =  2.27;
```

```
ru =      38;
220 rv =     19.4;
    J =   118;
    desc =   '100 x 100 x 12';
    crossSectionNum =   23;
    yo =    32.527;
225 b = 100;
    case 24
    A =   2.79;
    ru =    37.5;
    rv =    19.3;
230 J =   221;
    desc =   '100 x 100 x 15';
    crossSectionNum =   24;
    yo =    32.103;
    b = 100;
235 case 25
    A =   2.75;
    ru =    46;
    rv =    23.5;
    J =   143;
240 desc =   '120 x 120 x 12';
    crossSectionNum =   25;
    yo =    39.598;
    b = 120;
    case 26
245 A =   3.39;
    ru =    45.6;
    rv =    23.3;
    J =   269;
    desc =   '120 x 120 x 15';
250 crossSectionNum =   26;
    yo =    39.032;
    b = 120;
    case 27
255 ru =    57.6;
    rv =    29.3;
    J =   347;
    desc =   '150 x 150 x 15';
    crossSectionNum =   27;
260 yo =    49.497;
    b = 150;
    case 28
    A =   5.1;
    ru =    57.1;
265 rv =    29.2;
    J =   584;
    desc =   '150 x 150 x 18';
```

```

crossSectionNum = 28;
yo = 49.073;
270 b = 150;
case 29
A = 7.63;
ru = 77;
rv = 39.2;
275 J = 1070;
desc = '200 x 200 x 20';
crossSectionNum = 29;
yo = 66.185;
b = 200;
280 case 30
A = 9.06;
ru = 76.4;
rv = 39;
J = 1800;
285 desc = '200 x 200 x 24';
crossSectionNum = 30;
yo = 65.620;
b = 200;
case 31
290 A = 9.06;
ru = 76.4;
rv = 39;
J = 1800;
desc = '200 x 200 x 24';
295 crossSectionNum = 31;
yo = 65.620;
b = 200;
case 32
300 A = 9.06;
ru = 76.4;
rv = 39;
J = 1800;
desc = '200 x 200 x 24';
305 crossSectionNum = 32;
yo = 65.620;
b = 200;
end;
%Unit conversions from cm based units to m
A = A * 1e-3; % (10^-3 mm^2 to m^2)
310 ru = ru * 1e-3; % (mm to m)
rv = rv * 1e-3; % (mm to m)
J = J * 1e-9; % (10^-3 mm^4 to m^4)
yo = yo * 1e-3; % (mm to m)
b = b * 1e-3; % (mm to m)
315 elementArray(elementNum,1) = {A};
elementArray(elementNum,2) = {ru};

```

```

elementArray(elementNum,3) = {rv};
elementArray(elementNum,4) = {J};
elementArray(elementNum,5) = {desc};
320 elementArray(elementNum,6) = {crossSectionNum};
elementArray(elementNum,7) = {yo};
elementArray(elementNum,8) = {b};
end;

```

gtrans.m

```

function [G,L]=gtrans(ex,ey,ez,eo)
b=[ ex(2)-ex(1); ey(2)-ey(1); ez(2)-ez(1) ];
L=sqrt(b'*b);
n1=b/L;
5
lc=sqrt(eo*eo');
n3=eo/lc;

n2(1)=n3(2)*n1(3)-n3(3)*n1(2);
10 n2(2)=-n1(3)*n3(1)+n1(1)*n3(3);
n2(3)=n3(1)*n1(2)-n1(1)*n3(2);

A=[n1';n2;n3];O=zeros(3,3);
15 G=[ A O O O;
      O A O O;
      O O A O;
      O O O A];

```

assem.m

```

%% Stiffness Matrix and force vector assembly function
function [K,f]=assem(edof,K,Ke,f,fe)

[nie,n]=size(edof);
5 t=edof(:,2:n);
for i = 1:nie
    K(t(i,:),t(i,:)) = K(t(i,:),t(i,:))+Ke;
    if nargin==5
        f(t(i,:))=f(t(i,:))+fe;
10 end
end
end

```

bar3e.m

```

%% 3 dimensional element stiffness matrix function

```

```

function [Ke]=bar3e(ex,ey,ez,ep)

    E=ep(1);  A=ep(2);
5
    b=[ ex(2)-ex(1); ey(2)-ey(1); ez(2)-ez(1) ];
    L=sqrt(b'*b);

    Kle=E*A/L*[1 -1;
10              -1 1];

    n=b'/L;  G=[  n  zeros(size(n));
                zeros(size(n))  n  ];

15
    Ke=G'*Kle*G;

```

bar3s.m

```

%% 3 dimensional system stiffness matrix function
function [es]=bar3s(ex,ey,ez,ep,ed)

    b=[ ex(2)-ex(1); ey(2)-ey(1); ez(2)-ez(1) ];
5
    L=sqrt(b'*b);

    n=b'/L;  G=[  n  zeros(size(n));
                zeros(size(n))  n  ];

10
    E=ep(1); A=ep(2); Kle=E*A/L*[ 1 -1;
                                   -1 1];

    N=E*A/L*[-1 1]*G*ed';
    es=N;

```

coordxtr.m

```

%% Coordinate transformation function
function [Ex,Ey,Ez]=coordxtr(Edof,Coord,Dof,nen)

    [nel,dum]=size(Edof);
5
    ned=dum-1;
    [n,nsd]=size(Coord);
    [n,nd]=size(Dof);
    nend=ned/nen;

10
    for i = 1:nel
        nodnum=zeros(1,nen);
        for j = 1:nen

```

```

        check=Dof(:,1:nend)-ones(n,1)*Edof(i,(j-1)*nend+2:j*nend
        +1);
        [indx,dum]=find(check==0);
15      nodnum(j)=indx(1);
        end

        Ex(i,:)=Coord(nodnum,1)';
20      if nsd>1
            Ey(i,:)=Coord(nodnum,2)';
        end
        if nsd>2
            Ez(i,:)=Coord(nodnum,3)';
        end
25      end
end

```

extract.m

```

function [ed]=extract(edof,a)

[nie,n]=size(edof);

5      t=edof(:,2:n);

        for i = 1:nie
            ed(i,1:(n-1))=a(t(i,:))';
        end
end

```

solveq.m

```

function [d,Q]=solveq(K,f,bc)

        if nargin==2 ;
            d=K\f ;
5      elseif nargin==3;
            [nd,nd]=size(K);
            fdof=[1:nd]';

            %
10          d=zeros(size(fdof));
            Q=zeros(size(fdof));

            %
            pdof=bc(:,1);
            dp=bc(:,2);
            fdof(pdof)=[];
15          %
            s=K(fdof,fdof)\(f(fdof)-K(fdof,pdof)*dp);
            %
            d(pdof)=dp;

```

```

    d(fdof)=s;
20 end
    Q=K*d-f;

```

eldisp3.m

```

function eldisp3(hfig,ex,ey,ez,eo,ed,nD,plotpar,magnfac,interpmethod
)

global Stopanim
Stopanim=0;
5
if isempty(hfig),hfig=gcf;end
%view(2)
if nargin<7,nD=[];end
%if ~strcmp(lower(get(gcf,'Name')),'linvib mesh plot'),
10 % error('A call to function eldraw3 has to precede the call to
    eldisp3')
%end
if nargin<6,error('Too few input arguments'),end
[nx,mx,lx]=size(ex);[ny,my,ly]=size(ey);[nz,mz,lz]=size(ez);
[no,mo,lo]=size(eo);[nd,md,ld]=size(ed);
15 nel=nx;nen=mx;
if (nx~=ny || nx~=nz),error('Row dimensions of ex, ey and ez do
    not match'),end
if (mx~=my || mx~=mz),error('Column dimensions of ex, ey and ez do
    not match'),end
if (lx~=ly || lx~=lz),error('Layer dimensions of ex, ey and ez do
    not match'),end
if ~isempty(eo),
20 % if (no~=nx || lo~=lx),error('Some or all dimensions of eo are
    in error'),end
end
if lx>1,
    if lx~=ld,error('Layer dimensions of ex, ey and ez do not match
        dimension of ed'),end
end
25
if nen==2,% Beam and bar elements
    if isempty(nD),nD=1;end
elseif nen==4,% Plate and tetra elements
    if isempty(nD),nD=2;end
30 else
    error('Sorry. Only works for beam, bar and plates elements');
end

```

```

35 if nargin<8,plotpar=[3 4 4];end;if isempty(plotpar),plotpar=[3 4
    4];end

    if plotpar(1)==1,linestyle=': ';elseif plotpar(1)==2,linestyle='--'
        ;
    elseif plotpar(1)==3,linestyle='-';else
        error('Sorry. This linestyle does not exist');
40 end

    if plotpar(2)==1,color='k';elseif plotpar(2)==2,color='g';
    elseif plotpar(2)==3,color='y';elseif plotpar(2)==4,color='r';else
        error('Sorry. This color does not exist');
45 end

    if plotpar(3)==1,symbol='.';elseif plotpar(3)==2,symbol='*';
    elseif plotpar(3)==3,symbol='o';elseif plotpar(3)==4,symbol='None'
        ;else
        error('Sorry. This symbol does not exist');
50 end

    if nargin<9,magnfac=[];end
    if isempty(magnfac),
        charL=sqrt(((max(max(max(ex)))-min(min(min(ex))))^2+ ...
55             (max(max(max(ey)))-min(min(min(ey))))^2+ ...
             (max(max(max(ez)))-min(min(min(ez))))^2);
        magnfac=charL/max(max(abs(ed(:,[1:3 7:9]))))/20;
    end

60 ax=axis;
    if length(ax)>4
        minx=min([min(min(min(ex))) ax(1)]);
        miny=min([min(min(min(ey))) ax(3)]);
        minz=min([min(min(min(ez))) ax(5)]);
65 maxx=max([max(max(max(ex))) ax(2)]);
        maxy=max([max(max(max(ey))) ax(4)]);
        maxz=max([max(max(max(ez))) ax(6)]);
        dx=0.05*(maxx-minx);dy=0.05*(maxy-miny);dz=0.05*(maxz-minz);
        axis([minx-dx maxx+dx miny-dy maxy+dy minz-dz maxx+dz]);
70 else
        minx=min([min(min(min(ex))) ax(1)]);
        miny=min([min(min(min(ey))) ax(3)]);
        maxx=max([max(max(max(ex))) ax(2)]);
        maxy=max([max(max(max(ey))) ax(4)]);
75 dx=0.05*(maxx-minx);dy=0.05*(maxy-miny);
        axis([minx-dx maxx+dx miny-dy maxy+dy]);
    end

80 nframes=size(ed,3);

```

```

if nframes>1,Movie=moviein(nframes);end
saveas(gcf,'tempfig','fig');
for J=1:nframes
    close(gcf);
85    open('tempfig.fig');
    if lx==1,JJ=1;else JJ=J;end
    Ed=real(ed(:,:,J));
    for I=1:nel,
        if nen==2,%    Bar or Beam elements
90        [G,L]=gtrans(ex(I,:,JJ),ey(I,:,JJ),ez(I,:,JJ),eo(I,:,JJ));
        edl=G*Ed(I,:);%Local displacements
        %    [Excd,Eycd]=beam2crd([0 L],[0 0], ...
        %                            [edl(1) edl(2) edl(6) edl(7) edl(8)
        edl(12)],magnfac);
        %    [Excd,Ezcd]=beam2crd([0 L],[0 0], ...
95        %                            [edl(1) edl(3) -edl(5) edl(7) edl(9) -
        edl(11)],magnfac);
        %    Excd=Excd-L*(0:length(Excd)-1)/(length(Excd)-1);

        dL=0.001*L;
        x=[0 L];
100        y=[edl(1) edl(7)];
        Excd=magnfac*interp1(x,y,L*(0:1/20:1));
        x=[0 dL L-dL L];
        y=[edl(2) edl(2)+dL*edl(6) edl(8)-dL*edl(12) edl(8)];
        Eycd=magnfac*interp1(x,y,L*(0:1/20:1),interpmethod(I));
105        y=[edl(3) edl(3)-dL*edl(5) edl(9)+dL*edl(11) edl(9)];
        Ezcd=magnfac*interp1(x,y,L*(0:1/20:1),interpmethod(I));

        gd=G(1:3,1:3)*[Excd(:)';Eycd(:)';Ezcd(:)'];
        excd=gd(1,:)+ex(I,1,JJ)+(ex(I,2,JJ)-ex(I,1,JJ))*(0:length(
            Excd)-1)/(length(Excd)-1);
110        eycd=gd(2,:)+ey(I,1,JJ)+(ey(I,2,JJ)-ey(I,1,JJ))*(0:length(
            Excd)-1)/(length(Excd)-1);
        ezcd=gd(3,:)+ez(I,1,JJ)+(ez(I,2,JJ)-ez(I,1,JJ))*(0:length(
            Excd)-1)/(length(Excd)-1);
        hline(I)=line(excd,eycd,ezcd);
        elseif nen==4 && nD==2;%Plate elements
        hline=line([ex(I,1,JJ);ex(I,2,JJ);ex(I,3,JJ);ex(I,4,JJ);ex(
            I,1,JJ)], ...
115        [ey(I,1,JJ);ey(I,2,JJ);ey(I,3,JJ);ey(I,4,JJ);ey(I,
            1,JJ)], ...
        [ez(I,1,JJ);ez(I,2,JJ);ez(I,3,JJ);ez(I,4,JJ);ez(I,
            1,JJ)]+ ...
        magnfac*[Ed(I,1);Ed(I,2);Ed(I,3);Ed(I,4);Ed(I,1)])
        ;
    else
        error('Sorry. Only works for beam, bar and plate elements')
    ;

```

```

120     end
        end

        set(hline,'color',color,'linestyle',linestyle,'Marker',symbol);
        if nframes>1,Movie(:,J)=getframe;end
125     set(hline,'Visible','off');
end
%ha=uimenu(gcf,'Label','Stop Animation','Callback','global
        Stopanim;Stopanim=1;');
if nframes>1,
    disp('Pausing. Stop animation by clicking on the "Stop
        Animation" menu item')
130     while ~Stopanim
            movie(Movie,5);
        end
    else
        set(hline,'Visible','on');
135 end

```

eldraw3.m

```

function eldraw3(ex,ey,ez,plotpar,elnum)

    if ~((nargin==3)|| (nargin==4)|| (nargin==5))
        disp('??? Wrong number of input arguments!')
5        return
    end

    a=size(ex); b=size(ey); c=size(ez);

10    if ((a-b)== [0 0]) &((b-c)== [0 0] )
        nel=a(1);nen=a(2);
    else
        disp('??? Check size of coordinate input arguments!')
        return
15    end
    if nargin==3;
        plotpar=[1 1 1];
    end

20    [s1,s2]=pltstyle(plotpar);

    x0=sum(ex')/nen; y0=sum(ey')/nen; z0=sum(ez')/nen;

25    if nen==2
        x=ex'; y=ey'; z=ez';
    end

```

```

        xc=x; yc=y; zc=z;
30  else
        disp('!!! Sorry, this element is currently not supported!')
        return
    end
35  axis('equal')
    hold on
    view(3)
    plot3(xc,yc,zc,s1)
    plot3(x,y,z,s2)
40  if nargin==5
        for i=1:nel
            text(x0(i),y0(i),z0(i),int2str(elnm(i)))
        end
    end
45  xlabel('x'); ylabel('y'); zlabel('z');
    hold off

```

pltstyle.m

```

function [s1,s2]=pltstyle(plotpar)

    if plotpar(1)==1 ; s1='-';
    elseif plotpar(1)==2 ; s1='--';
5   elseif plotpar(1)==3 ; s1=':.';
    else disp('??? Error in variable plotpar(1)!');
        return;
    end

10  if plotpar(2)==1 ; s1=[s1,'k'];
    elseif plotpar(2)==2 ; s1=[s1,'b'];
    elseif plotpar(2)==3 ; s1=[s1,'m'];
    elseif plotpar(2)==4 ; s1=[s1,'r'];
    else disp('??? Error in variable plotpar(2)!');
15  return;
    end

    if plotpar(3)==1 ; s2='ko';
    elseif plotpar(3)==2 ; s2='k*';
20  elseif plotpar(3)==0 ; s2='k.';
    else disp('??? Error in variable plotpar(3)!');
        return;
    end

```

MATLAB code for Chapter 4

The graphical LCC optimization approach used for an arbitrary tower segment:

monopole.m

```

%monopole.m

clear all                % Clear all stored data
5 clc                    % Clear command window
%Global statement for sharing variable info between various m-
  files
global E sig_cr rho_st phi g c r a w_turb
global rho_air cd rotor_w speed LP L del fy

10 %Initializing Material variables for S355JR Structural Steel
E = 200e+09;            % Modulus of Elasticity of S355JR Steel
fy = 355e+06;          % Yield stress for S355JR steel
fu = 470e+06;          % Plastic/ultimate stress for S355JR steel
rho_st = 7850;         % Density of steel 7850 kg/m3
15 g = 9.81;           % gravitational acceleration
c = 16;                % Cost of Structural steelin R/kg (Feb
  2013)
phi = 0.9;            % Structural factor for Steel

%Initialising variables for Wind Load Calculations as per SANS
  10160-3
20 r = 8.0;            % Wind Turbine rotor radius
rho_air = 1.225;       % density of air
cd = 1.0;              % drag coefficient
v_b0 = 28.0;          % fundamental basic wind speed in m/s
c_prob = 0.94622;     % Return Period: 20 Years
25 v_b = c_prob*v_b0;
v_bpeak = 1.4*v_b;    % Peak wind speed (gust factor)
cr = 1.070;           % Terrain roughness factor at 24 m
speed = cr*v_bpeak;   % Peak wind velocity
a = 0.333333333;     % Induction factor a, ideally designed
  rotor a =1/3
30 rotor_w = 0.5*rho_air*(speed*speed)*pi*(r^2)*4*a*(1-a); %
  concentrtd wind load on rotor at hub level (Peterson, 2010)

%Initialising geometric variables of Tower and Turbine
LP = 24.530;          % Location of rotor load on monopole tower
  (0.530 m higher)
L = 24.0;             % Length of Monopole Tower
35 del = (24000/180)/1000; % Allowable Tower deflection - (24000/400 =
  0.060 m) (SANS10160 and 10162)

```

```

dt_ratio = 36.62;           % d/t ratio limit for local buckling of
    CLASS 3 SECTIONS (SANS 10162-1)
w_turb = 7030;             % Weight of Turbine Assembly in Newtons

40 x1=0.02:0.01:3;
x2=0.025:0.01:3;
[X1 X2] = meshgrid(x1,x2);

%The objective function is evaluated over the entire grid
45 f1 = obj_ex3(X1,X2);

%Constraints are evaluated by calling the functions
ineq1 = ineq1_ex3(X1, X2);
ineq2 = ineq2_ex3(X1, X2);
50 ineq3 = ineq3_ex3(X1, X2);
ineq4 = ineq4_ex3(X1, X2);

%Right hand side values of the constraints
g1val = phi*sig_cr;
55 g2val = phi*sig_cr;
g3val = del;
g4val = dt_ratio;

figure(1)
60 [C1 han1] = contour(x1,x2,f1,[0,100000,300000, ...
    500000, 700000, 900000, 1100000, 1300000, 1800000, 2400000,
    3000000, 5000000], 'g-');
clabel(C1,han1);
set(gca,'xtick', [0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
    2.75 3.0])
set(gca,'ytick', [0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
    2.75 3.0])
65 xlabel('outside diameter','Fontname','calibri', ...
    'FontSize',12);
ylabel('inside diameter','Fontname','calibri', ...
    'FontSize',12);
grid
70 hold on

%figure(2) %a new figure window to be added
contour(x1,x2,ineq1,[g1val,g1val], 'r-');
hold
75 %draw another contour at 10% the constraint boundary
%contour(x1,x2,ineq1,[0.1*g1val,0.1*g1val], 'b-');
set(gca,'xtick', [0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
    2.75 3.0])
set(gca,'ytick', [0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
    2.75 3.0])

```

```

xlabel('outside diameter','Fontname','calibri', ...
80   'FontSize',12);
ylabel('inside diameter','Fontname','calibri', ...
   'FontSize',12);
hold on
grid
85
%The following code is useful for the consolidated figure ->
   Uncomment
% [C2 han2] = contour(x1,x2,ineq1,[g1val,g1val],'r-');
% clabel(C2,han2);
% set(h2,'LineWidth', 1)
90 % k2 = gtext('g1');
% set(k2,'FontName','Times','Fontweight','bold',...
%   'FontSize', 14, 'Color','red')

%figure(3)
95 contour(x1,x2,ineq2,[g2val,g2val],'r--');
hold on
contour(x1,x2,ineq2,[0.1*g2val,0.1*g2val],'b--');
set(gca,'xtick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
2.75 3.0])
set(gca,'ytick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
2.75 3.0])
100 xlabel('outside diameter','Fontname','calibri', ...
   'FontSize',12);
ylabel('inside diameter','Fontname','calibri', ...
   'FontSize',12);
hold on
105 grid

%The following code is useful for the consolidated figure ->
   Uncomment
% [C3 han3] = contour(x1,x2,ineq2,[g2val,g2val],'r--');
% clabel(C3,han3);
110 % set(h3,'LineWidth', 1)
% k3 = gtext('g2');
% set(k3,'FontName','Times','Fontweight','bold',...
%   'FontSize', 14, 'Color','red')

115 %figure(4)
contour(x1,x2,ineq3,[g3val,g3val],'ro-');
hold on
contour(x1,x2,ineq3,[0.1*g3val,0.1*g3val],'bo-');
set(gca,'xtick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
2.75 3.0])
120 set(gca,'ytick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
2.75 3.0])
xlabel('outside diameter','Fontname','calibri', ...

```

```

        'FontSize',12);
ylabel('inside diameter','Fontname','calibri', ...
        'FontSize',12);
125 hold on
    grid

    %The following code is useful for the consolidated figure ->
        Uncomment
    % [C4 han4] = contour(x1,x2,ineq3,[g3val,g3val],'b-');
130 % clabel(C4,han4);
    % set(h4,'LineWidth', 1)
    % k4 = gtext('g3');
    % set(k4,'FontName','Times','Fontweight','bold',...
    %     'FontSize', 14, 'Color','blue')
135
    %figure(5)
    contour(x1,x2,ineq4,[g4val,g4val],'r-');
    hold on
    contour(x1,x2,ineq4,[0.001*g4val,0.001*g4val],'b-');
140 set(gca,'xtick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
        2.75 3.0])
    set(gca,'ytick',[0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0 2.25 2.5
        2.75 3.0])
    xlabel('outside diameter','Fontname','calibri', ...
        'FontSize',12);
    ylabel('inside diameter','Fontname','calibri', ...
145     'FontSize',12);
    hold on
    grid

    %The following code is useful for the consolidated figure ->
        Uncomment
150 % [C5 han5] = contour(x1,x2,ineq4,[g4val,g4val],'b--');
    % clabel(C5,han5);
    % set(h5,'LineWidth', 1)
    % k5 = gtext('g4');
    % set(k5,'FontName','Times','Fontweight','bold',...
155 %     'FontSize', 14, 'Color','blue');

    %The equality and inequality constraints are not written with 0 on
        the
    %right hand side. If you do write them that way, you would have to
        include
    %[0, 0] in the contour commands
160
    %Finding the optimum solution values using fmincon
    x0 = [1.5 1.25];
    options = optimset(@obj);

```

```
[x,fval,exitflag,output] = fmincon(@obj,x0,[],[],[],[],[],[],[],
    @confun,options);
```

obj.m

```
%obj.m

function retval = obj_ex3(X1, X2)

5 %Global statement for sharing variable info between various m-
  files
global E sig_cr rho_st phi g c r a w_turb
global rho_air cd rotor_w speed LP L del fy

area = 0.25*pi*(X1.^2 - X2.^2);      % Area matrix for X1 and X2
  values
10 mass = area.*L*rho_st;            % Mass matrix for X1 and X2
  values
retval = mass*c;                    % Cost matrix for X1 and X2
  values
```

ineq1.m

```
%ineq1.m

function retval = ineq1_ex3(X1, X2)

5 %Global statement for sharing variable info between various m-
  files
global E sig_cr rho_st phi g c r a w_turb
global rho_air cd rotor_w speed LP L del fy

%Maximum allowable compression force and bending force
  interaction constr.
10 area = 0.25*pi*(X1.^2 - X2.^2);    %Matrix

  % Load Factor for slender, non-redundant structures
gamma = 1.5;

15 % Design Moment at base of tower structure
Md = 1698.2*10^3;

  % R = radius to centre of shell
R = abs((X1 + X2)/4);
20 % t = thickness of shell
t = abs((X1 - X2)/2);
```

```

% Nd = axial compressive force due to weight of tower
Nd = 64.6*10^3;
25
% Axial Stress and Bending Stress
sig_ad = Nd./(R.*t*2*pi);
sig_bd = Md./(pi*t.*R.^2);

30 % Reduction factor eps_a and eps_b
eps_a = 0.83./((1+(R./t)*0.01).^0.5);
eps_b = 0.1887 + 0.8113.*eps_a;
eps = (eps_a.*sig_ad + eps_b.*sig_bd)./(sig_ad + sig_bd);

35 % Elastic critical compressive stress
sig_el = E./((R./t)*(3*0.91).^0.5);

% The relative slenderness ratio for local buckling
lamda_a = (fy./(eps.*sig_el)).^0.5;
40
for i = 1:298
    for j = 1:299
if lamda_a(i,j) <= 0.3,
    sig_cr = fy;
45 else
    if ((lamda_a(i,j) > 0.3) && (lamda_a(i,j) <= 1.0)),
        sig_cr = (1.5 - 0.913*(lamda_a).^0.5)*fy;

    else
50         %if (L <= 1.42*R*(R./t).^0.5),
            sig_cr = fy;
        %end
    end
end
55 end
end

% For global buckling Nel is the Euler force for a cantilever
column
Nel = 0.25*(pi^3)*E*(R.*R.*R).*t./(L^2);
60
% The relative slenderness ratio for global buckling is
lamda_r = (sig_cr./(Nel./(2*pi*R.*t))).^0.5;

% The core radius k of a tubular structure and e
65 k = R.*0.5;
e = 0.34*(lamda_r-0.2).*k;

if lamda_r <= 0.2,
    e = 0;
70 end

```

```

if e > (0.002*L),
    del_e = (e - 0.002*L);
    e = e + del_e;
75 end

%LHS of the inequality equation becomes and is returned as the
    actual
%stress value from all the loads. sig_cr is also calculated
LHS = sig_ad + (Nel./(Nel - Nd)).*((Md + Nd.*e)./(pi*R.*R.*t));
80
retval = LHS;

```

ineq2.m

```

%ineq2.m

function retval = ineq2_ex3(X1, X2)

5 % Global statement for sharing variable info between various m-
    files
global E sig_cr rho_st phi g c r a w_turb
global rho_air cd rotor_w speed LP L del fy

% Maximum allowable Shear Force at the tower base constraint
10 area = 0.25*pi*(X1.^2 - X2.^2); %Matrix
inertia = pi*(X1.^4 - X2.^4)/64; %Matrix
q = (X1.*X1 + X1.*X2 + X2.*X2)/6.0;
t = (X1 - X2)/2;
Sd = 79.6*10^3;
15
retval = Sd.*q./inertia;

```

ineq3.m

```

%ineq3.m

function retval = ineq3_ex3(X1, X2)

5 %Global statement for sharing variable info between various m-
    files
global E sig_cr rho_st phi g c r a w_turb
global rho_air cd rotor_w speed LP L del fy

% Maximum allowable deflection constraint
10 %area = 0.25*pi*(X1.^2 - X2.^2); %Matrix
inertia = pi*(X1.^4 - X2.^4)/64; %Matrix

```

```
fd = 0.5*rho_air*speed*speed*cd*X1;  
defl_twr = fd*L^4./(8*E*inertia);  
defl_rot = (2.0*rotor_w*L^3 - rotor_w*L*L*LP)./(E*inertia);  
15 retval = defl_twr + defl_rot;
```

ineq4.m

```
%ineq4.m  
  
%Local buckling constraint (d/t ratio)  
function retval = ineq4_ex3(X1, X2)  
5 retval = X1./(0.5*(X1 - X2));
```

Appendix B: PWT and Cutting Times Tables

Post weld treatments time

Table B.1: Time needed for different PWT techniques (Farkas and Jarmai, 2008).

Method	T_0 (<i>min/m</i>)
Grinding	60
TIG* dressing	18
Hammer peening	4
UIT**	15

* Tungsten Inert Gas treatment method, often used to improve fatigue performance

** Ultrasonic Impact Treatment

Plate cutting time for welding

Table B.2: Cutting time of plates for 1 *mm* length, T_{CP} (*min/mm*) in the function of weld size a_w (*mm*) for longitudinal fillet welds, T-, V- and $\frac{1}{2}$ V-butt welds (Farkas and Jarmai, 2008).

Cutting technology	thickness (<i>mm</i>)	$10^3 T_{CP} = 10^3 C_{CP} t^n$
Acetylene (normal speed)	2-15	1.1388 $t^{0.25}$
Acetylene (high speed)	2-15	0.9561 $t^{0.25}$
Stabilized gas-mix (normal speed)	2-15	1.1906 $t^{0.25}$
Stabilized gas-mix (normal speed)	2-15	1.0858 $t^{0.23}$
Propane (normal speed)	2-15	1.2941 $t^{0.24}$
Propane (high speed)	2-15	1.1051 $t^{0.25}$

Table B.3: Cutting time of plates for 1 mm length, T_{CP} (min/mm) in the function of weld size a_w (mm) for fillet, longitudinal X- and K-butt welds (Farkas and Jarmai, 2008).

Cutting technology	thickness (mm)	$10^3 T_{CP} = 10^3 C_{CP} t^n$
Acetylene (normal speed)	10-40	$0.8529 t^{0.36}$
Acetylene (high speed)	10-40	$0.6911 t^{0.38}$
Stabilized gas-mix (normal speed)	10-40	$0.8991 t^{0.36}$
Stabilized gas-mix (normal speed)	10-40	$0.6415 t^{0.44}$
Propane (normal speed)	10-40	$0.9565 t^{0.36}$
Propane (high speed)	10-40	$0.7870 t^{0.38}$

Appendix C: Wind Load Calculations on Wind Turbine Towers

Wind load calculations

Wind loads on any structure depend on several factors. These include wind velocity, terrain conditions and characteristics, size, shape and structural response. Conventional design assumes wind to act as a horizontal pressure normal to the surface or face of a structure (Karpát, 2013). Eurocode 1 Part 2-4 can be used to calculate wind loads on a wind turbine tower. A comparison with SANS 10160:2011 - Part 3 should be investigated and compared.

The average wind force, F_w acting on a structural component can be calculated using equation C.1.

$$F_w = c_e(z)q_{ref}c_s c_d c_f A_{ref} \quad (\text{C.1})$$

where z is the reference height for external wind load, A_{ref} is the reference area of the structural element. c_s and c_d are the size and dynamic factors, respectively. The combined structural factor, $c_s c_d$ takes into account the effect of wind actions from the non-simultaneous occurrence of peak wind pressure on the surface with the effect of vibrations due to turbulence. c_f is the force factor and q_{ref} is the velocity pressure from the basic value of the fundamental wind speed v_{ref} with an air density of $\rho = 1.225 \text{ kg/m}^3$. The velocity pressure is calculated using equation C.2.

$$q_{ref} = \frac{1}{2}\rho v_{ref}^2 \quad (\text{C.2})$$

c_e is the exposure factor at a certain height z and can be found as follows:

$$c_e(z) = c_r^2 c_t^2 (1 + 7I_v) \quad (\text{C.3})$$

where I_v is the turbulence intensity and is defined by the expression in equation C.5

$$I_v = \frac{k_r}{c_r c_t} \quad (\text{C.4})$$

where

$$c_r = k_T \ln\left(\frac{z}{z_0}\right) \quad (\text{C.5})$$

k_r is the turbulence factor. k_T is the terrain factor and c_t the orography factor, z_0 is the roughness length (ranges between 0.003 and 1.0) and depends on the ground roughness and the distance with uniform terrain roughness in an angular sector around the wind direction. The values can be obtained from the respective tables in Eurocode 1 Part 2-4. The force factor, c_f is calculated as shown in expression C.6.

$$c_f = \psi_\lambda c_{f0} \quad (\text{C.6})$$

where ψ_λ is the end-effect factor and c_{f0} is the force coefficient of structures or structural elements without free-end flow. c_{f0} is a function of Reynold's number at the specific flow conditions and can be obtained from the appropriate tables in Eurocode 1 Part 2-4.

The uniformly distributed wind load over a shell segment (note that this should be considered for each segment separately) is calculated using expression C.7.

$$P_w = q_{ref} c_s c_d c_f D \quad (\text{C.7})$$

where D is the average diameter of either the wind turbine (assuming a tapered tower shape) or when segments are considered, D is the average diameter of the shell segment.

It should be noted, that the South African loading code, SANS 10160:2011 Part 1-3 includes an approach similar to the one followed from the Eurocode above.

Appendix D: FEM Solver verification check

PROKON 25-Bar Model Results

PROKON 2.5 structural analysis and design software was used to verify and check the accuracy of the native MATLAB FEM solver that is implemented in the multi-objective optimization algorithm for the fitness evaluation of the individuals. The PROKON analysis was performed only for the 25-Bar truss tower model. After the algorithm converged and the graphical figure of the fittest design was produced, the fittest design was modeled in PROKON using the same boundary conditions, loads and elements for comparison.

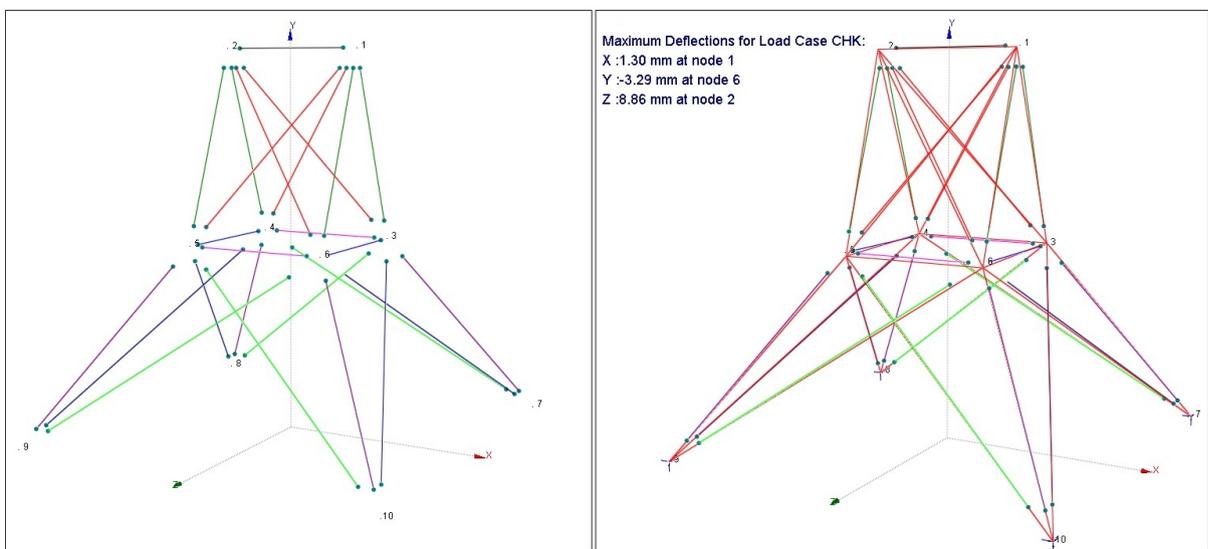


Figure D.1: Undeformed (left) and deformed (right) 25-Bar tower PROKON model used to verify displacement results from natively developed MATLAB FEM-Solver that is implemented in the multi-objective optimization

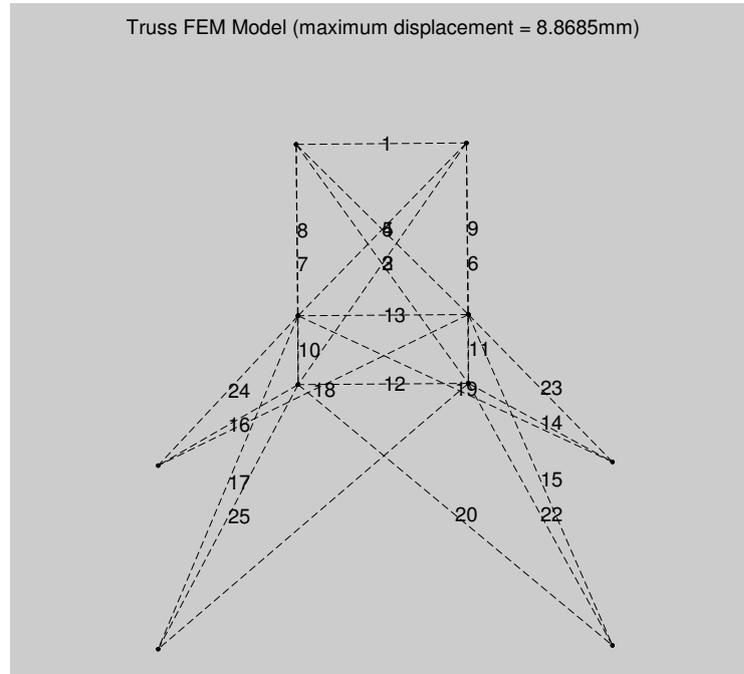


Figure D.2: 25-Bar FEM model from the multi-objective optimization yields same results as PROKON analysis

The results obtained are exactly the same and are presented for reference purposes. Figure D.1 and Table D.4 show the results obtained from a linear analysis in PROKON. Figure D.2 depicts the results from the MATLAB FEM solver for the same configuration as in Figure D.1 and Table D.4. The maximum nodal deflection value obtained is $\delta_{max} = 8.86 \text{ mm}$. The results are identical and therefore satisfactory. The element forces have been compared as well and exactly the same results are obtained, thus the native solver implemented in this thesis performs structural analysis for space truss tower structures accurately.

Table D.5 is a summary of the element mass values obtained from the PROKON output file used in a further crosscheck. The value of 221 kg is the same as obtained for the fittest design.

Table D.4: PROKON Nodal Point Displacements at SLS

Node	Lcase	X-disp (m/m)	Y-disp (m/m)	Z-disp (m/m)
1	CHK	1.30	-1.34	8.81
1	LC1	1.30	-1.34	8.81
2	CHK	0.08	-1.22	8.86
2	LC1	0.08	-1.22	8.86
3	CHK	0.31	1.47	-0.32
3	LC1	0.31	1.47	-0.32
4	CHK	-0.07	1.56	-0.34
4	LC1	-0.07	1.56	-0.34
5	CHK	0.59	-3.24	-0.54
5	LC1	0.59	-3.24	-0.54
6	CHK	-0.28	-3.29	-0.56
6	LC1	-0.28	-3.29	-0.56
7	CHK	0.00	0.00	0.00
7	LC1	0.00	0.00	0.00
8	CHK	0.00	0.00	0.00
8	LC1	0.00	0.00	0.00
9	CHK	0.00	0.00	0.00
9	LC1	0.00	0.00	0.00
10	CHK	0.00	0.00	0.00
10	LC1	0.00	0.00	0.00

Table D.5: PROKON Statistical Data of 25-Bar Tower

Section	Designation	Total Mass (kg)
SEC1	A=64.516	0.3
SEC2	A=387.10	14.2
SEC3	A=2193.54	65.9
SEC4	A=64.516	0.7
SEC5	A=1096.77	11.6
SEC6	A=580.64	29.6
SEC7	A=322.58	16.4
SEC8	A=2193.54	82.3
TOTAL		221.0