Optimum Design of Steel Structures Using Evolutionary Algorithms

Zolisa Dolwana

Submitted in fulfillment of the academic
requirements for the degree of

MASTER OF ENGINEERING
in
Mechanical Engineering

in the
Department of Mechanical Engineering
Durban University of Technolgoy
Durban
January 2019

Declaration

The research work described in this thesis was carried out under the Department of Mechanical Engineering, Durban Univesity of Technology, under the supervision of Professor Pavel. Y Tabakov and Professor Sibusiso Moyo.

This desertation presents original work by the author and has not been submitted in any form for any degree or diploma to any University. Where use has been made of the work of others it is duly acknowledged in the text.

January, 2019.

Student: ______________________________ ______________________________

Mr Zolisa Dolwana Date

Supervisor: ______________________________ ______________________________

Prof.P.Y. Tabakov Date

Co-supervisor: ______________________________ ______________________________

Prof. S. Moyo Date

Dedication

I dedicate this thesis to my family and friends. A special thank you goes to Vuyo Dolwana for the support and the inspiration provided. Thank you for being there for me maNdungwana and all that you have done.

Acknowledgement

I would like to extend my sincere thank you to my God, through His unconditional love. He led me thus far in making this work a success. I would like to thank my two supervisors, Professor Pavel Y Tabakov, and Professor Sibusiso Moyo, for all their guidance and help through the course of this work. I also would like to thank my friend Andile Ntanjani for being there in times of need. How can I forget Mcebisi Mahlathi for his inspirations and pushing me to make sure that I complete this research.

My thank you also go to the Durban University of Technology and, particularly, the Directorate for Research and Postgraduate Support for their invaluable assistance and support that made this research work possible.

Abstract

The subject of this thesis is optimization of steel structures using evolutionary algorithms. Heuristic algorithms are used and compared for the best possible results both in two dimensional and three dimensional structures. The topology, shape and sizing of the optimization problem has been formulated based on practical real life problems. The design has to produce best results without violating the stress and displacement constraints. The design constraints satisfy the demands of steel material properties and the selected profiles.

Structural steel is discussed in detail on how they can be designed, and manufactured in both two dimensions (2-D) and three dimensions (3-D) to carry required loads and provide adequate rigidity. These types of structures are commonly found in the construction of buildings, bridges, transmission line towers, industrial sheds, automotive vehicles and ships etc. Steel exhibits desirable physical properties that make it one of the most versatile structural materials in use. Its great strength, uniformity, light weight, ease of use, and many other desirable properties makes it the material of choice for numerous structures such as steel bridges, high rise buildings, towers, and other structures. Steel structures are formed with a specific shape following certain standards of chemical composition and strength. During the course of construction steel can be joined by welding or bolting methods.

The structural steel problem is solved using population based methods, namely, the genetic algorithm (GA), particle swarm optimization (PSO) and big bang - big crunch (BB-BC). The quality of results produced using these heuristic methods has been studied in several problems.

The present study demonstrates how progress in modern evolutionary algorithms has revolutionized design optimization of engineering structures. The performance of an evolutionary algorithm called the big bang - big crunch algorithm is shown by example of the steel trusses where the minimum possible weight was determined subjected to stress and displacement constraints.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Structural steel is referred to as a structure made from organised combinations of structural steel members designed to carry loads in the most economical manner and provide adequate rigidity. The two key features of the structure is that it should first serve the purpose for which it is intended and this is achieved by proper functional planning and secondly it should have adequate strength to withstand direct and induced forces to which it may be subjected to during its life span. An inadequate assessment of forces and their effects on the structure may lead to excessive deformation and its failure. Therefore the design of steel structures includes functional planning, loads applicable, material selection and design method [13].

These structures are commonly found in the construction of buildings, bridges, transmission line towers, industrial sheds, manufacturing of automotive vehicles and ships. Steel exhibits desirable physical properties that make it one of the most versatile structural materials in use. Its great strength, uniformity, light weight, ease of use, and many other desirable properties makes it the material of choice for numerous structures. Steel structures are formed with a specific shape following certain standards of chemical composition and strength. They can also be defined as hot rolled products, with a cross section of special form like channels, equal leg angles and beams or joints.

As shown in Figure 1.1, structural steel can be rolled into various shapes and sizes in rolling mills. Usually sections having larger modulus of sections in proportion to their cross-sectional areas are preferred [13]. New designs are expected to improve performance, meet new stringent weight targets and at the same time they need to be more economical to manufacture.

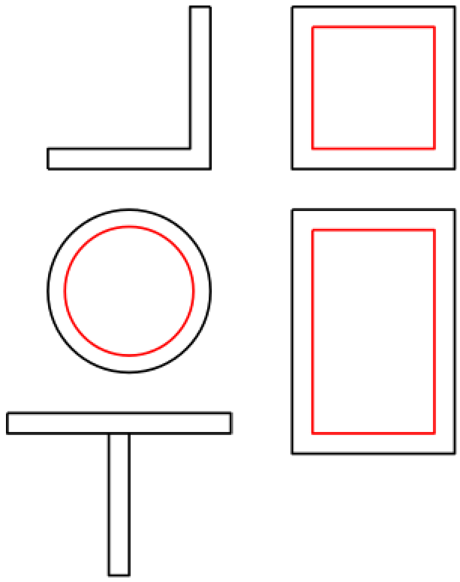


Figure 1.1: Standard profiles of steel structures

According to Duggal [13], angle sections were probably the first shapes rolled and produced in 1819 in America, followed by I-beam shape that was introduced by Zeros of France in 1849. In the early 1870 Channel sections and Tee sections were developed. All these standard profiles were made out of wrought iron. The design of steel sections is governed by cross-sectional area and section modulus. It has been seen from the literature as found in [13], and [42] that a variety of steel sections are rolled into a desired shape, but due to the limitations of rolling mills only a few are available. Also, if a section is in demand, it is rolled regularly but one which is in little demand is rolled on order and hence costs more. Therefore, the design is not only governed by sectional properties but also on availability of the steel sections in the market, which becomes a major consideration. Another factor governing the choice is the ease with which steel sections can be connected.

The three basic approaches of structural optimization are sizing, shape and topology optimization as shown in Figure 1.2 to Figure 1.4. Sizing optimization is about designing suitable dimensions of cross sections or selecting suitable thicknesses or cross sectional areas. Shape optimization is about finding the best suitable shape of the structure. In topology optimization the main aim of the designer is to find the optimum structure by changing the amount of material and material locations or components in the structure. These approaches are usually combined in order to obtain the best possible results.

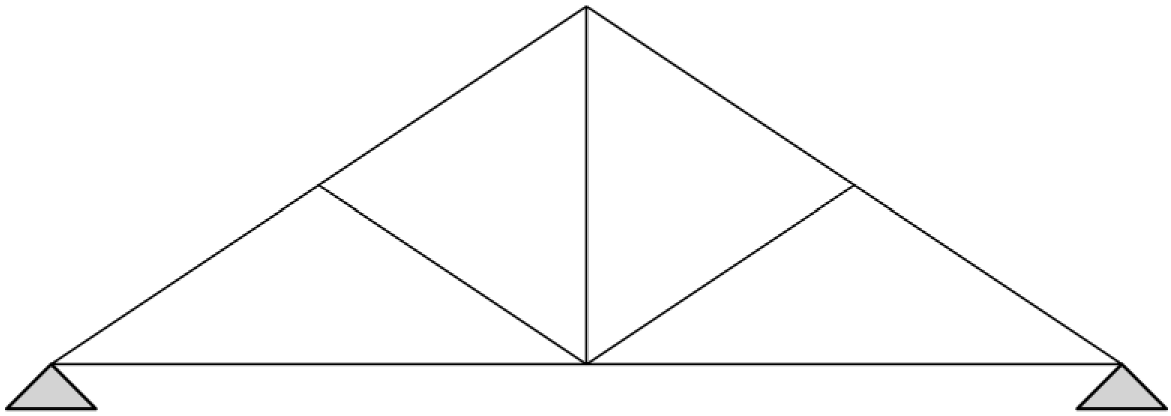


Figure 1.2: Sizing optimization of structural steel

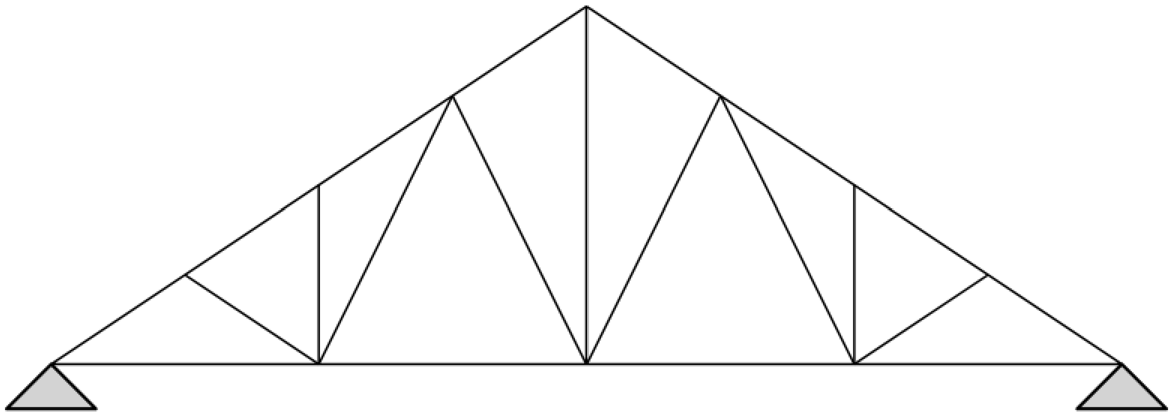


Figure 1.3: Shape optimization of structural steel

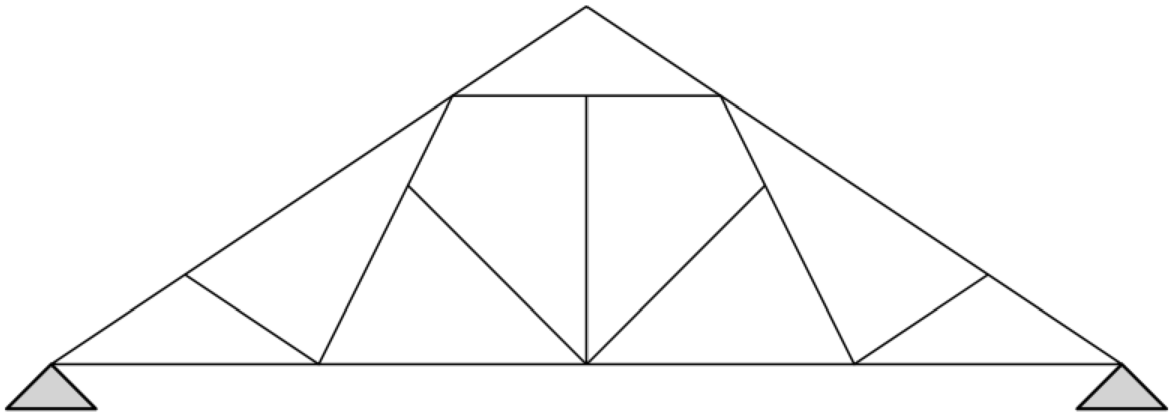


Figure 1.4: Topology optimization of structural steel

The task in sizing optimization is to select suitable profiles for each truss member that forms part of the structure being optimized. The literature as published by Floudas [20], and Nemhauser and Wolsely [50], deal with discrete optimization from a mathematical point of view. In optimization of structural steel the ultimate goal is weight and cost minimization of the structure and the role of design constraints is to take care that the structure is useful and possible to manufacture. The fabrication cost of the structure can be assumed to be directly proportional to the mass of structure. The cost of material depends on the type of welded steel structure and the minimum mass is not always the same as minimum costs of the structure. Cost optimization of welded structural steel is discussed further in detail in the literature as found in [18], [31], and [32].

Structural steel members can be joined by three well known methods, riveting, bolting and welding see Figure 1.5. In a paper by Leslie [44], rivets were allowed to fabricate large load bearing elements by connecting L-sections with flat plates. Often rivets are combined with bolts as they concern differentiated structural functions. Elements that are load bearing were fabricated with rivets either in the shop or in the field then assembled with bolts or rivets in the field. Unlike bolts, rivets can be considered as permanent fasteners like welding connections. The use of bolts is advantageous when dismantling is required and for applications where use of rivets was inappropriate, that is when the grip length is too long or connection between wrought and cast iron. The use of rivets had some advantages over bolts due to material cheapness and improved stiffness of connections and the ability to compensate holes misalignment etc.

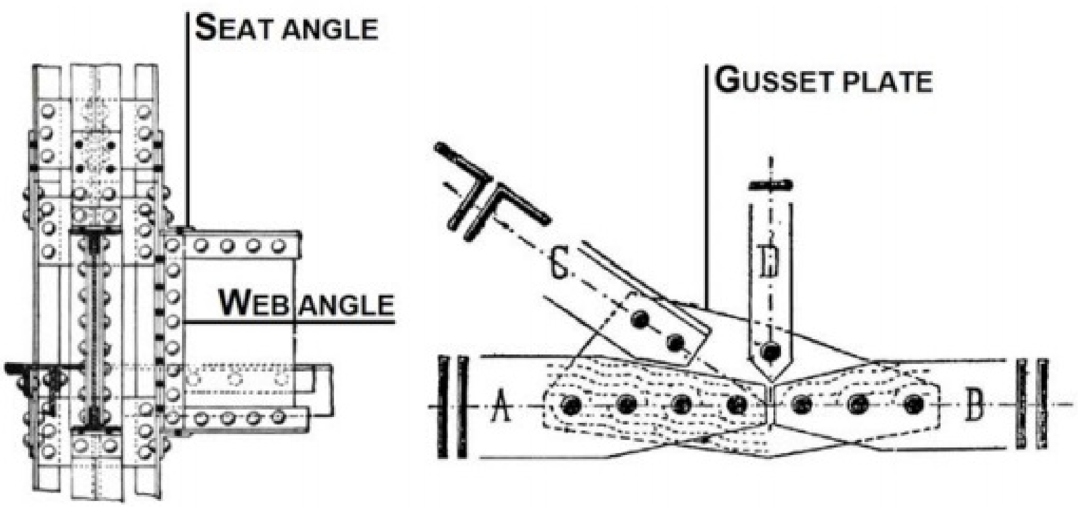


Figure 1.5: Truss work connection [22]

According to Leslie [44], the use of field riveting in the early 20^{th} century was considered as an efficient method of connection and cost saving. Later in the 20^{th} century things changed, the overall cost of riveted connections became higher than the bolting connection [23]. The use of bolting in the 20^{th} century progressively earned the status of leading fasteners [55]. After the second world war the use of field riveting progressively fell into disuse.

The use of riveting was negatively affected by a new welding technique. The welding technique mainly developed in the shipbuilding industry during the second world war as a way to increase vessels loading capacity and the self-weight decrease of 15% to 20% and easily make them watertight [61]. According to Epsion [16], the use of iron and steel in the construction of structural steel in the 1930s contributed to the cost effectiveness of building techniques. An example of hybrid fabrication was noticed in the year, 1931, during the construction of Lanaye Vierendeel steel bridge in Belgium in which both shop welding and field riveting were used [16]. The use of welding techniques became more popular than riveting. Numerous technical disadvantages were discovered about riveting, for example, increase of self-weight in riveted structures, high costs, cumbersome riveting equipment, noise pollution, increased numbers of man power required for riveting and long hours required for riveting. From a structural point of view the use of riveting and riveted connections proved its superiority compared to the welding technique, especially for structural work such as bridge construction subjected to fatigue loading [23]. The use of welding techniques led structural rivets on a slippery slope in the building sector from the 1930s onwards [16].

The research by Jalkanen [30], in tubular truss optimization, suggests that the joints of tubular truss should be designed so that there is no need to reinforce or to stiffen the joints afterwards. Strengthening of joints causes more extra work and it is more costly. All members in a tubular trusses are welded directly together and the dimensions of the members have to be chosen in such a way that the strength requirements for members and joints are considered at the same time.

The concept of optimization is a basic part of our daily lives. Optimization is used as a tool to achieve the best economic results in the engineering field. According to Christensen and Klarbring [12], optimization has been long studied through the globe in many disciplines of science and engineering. In the past structural optimization was overwhelmed with optimality criteria and mathematical programming based methods. Despite strong mathematical backgrounds and remarkable speed of convergence to the optimum, these methods have found limited applications in some optimization areas, such as discrete structural optimization. The need for selection of member sizes from a list of ready sections hampers a direct application of these methods to practical structural optimization problems. Numerous optimization techniques have been developed in the last two decades for optimum design of structural systems.

With the availability of computer codes, new and more sophisticated computational tools called meta-heuristics that have been developed, make it possible to find optimum solutions to problems from engineering practice. Structural optimization with meta-heuristic search methods have become more popular as a consequence of acquiring extensive accomplishment in dealing with a variety of practical and complex optimization tasks, where it is nearly impossible to come up with the optimum solution by traditional deterministic design procedures. Such tools are finding increasing industrial use due to their efficiency as well as ease in their implementations. These methods are recognized as one of the most practical approaches for solving many complex problems, and this is particularly true for many real-world problems that are combinatorial in nature.

The modern world is fast-paced and highly competitive which demands new and better products in a seemingly insatiable market. At the same time, there is pressure to ensure that the product costs less, is more efficient and capable and has lower environmental impact, among other requirements. In order to keep up with demands, designers are resorting to use computer-aided methods. Among these are the use of more advanced materials, and the use of optimization techniques [53]. According to the research by Adeli and Kumar [1], one of the key guiding principles of designing steel structures is that a design is considered complete, not when nothing more can be added, but when nothing more can be taken away. This is one of the underlying principles of optimisation . With the drive towards the use of minimum resources for modern designs, motivated by factors as diverse as cost and ecological impact, optimisation is increasingly being used to create practical designs. With the availability of powerful desktop computers, optimisation has become much easier to implement. Coupling the Finite Element Method (FEM) with optimisation routines has opened up vast new areas to optimisation methods.

1.2 Problem Statement

The structural steel is manufactured in two dimensions (2-D), and in three dimensions (3-D). The 2D and 3D structures are designed to carry loads and provide adequate rigidity. These types of structures are commonly found in the construction of buildings, bridges, transmission line towers, industrial sheds, automotive vehicles and ships etc. Steel exhibits desirable physical properties that make it one of the most versatile structural materials in use. Its great strength makes uniformity, light weight, ease of use, and many other desirable properties, make it the material of choice for numerous structures such as steel bridges, high rise buildings, towers, and other structures. Steel structures are formed with a specific shape following certain standards of chemical composition and strength. During the course of construction steel can be joined by the welding or bolting method.

Due to continuous inflation of the cost of materials, transportation and construction costs, etc. this necessitates the development of computer aided numerical algorithms that are capable of design optimisation of steel structures to minimise weight without reducing the structural strength or serviceability of the structure.

According to a study by Carbas and Hasancebi [10], design optimization of steel frames is a very popular topic in structural engineering due to savings in cost of the structures by using the optimization process. Although the final cost of a steel structure is affected by many factors, such as material, manufacturing, erection and transportation costs, the material cost of steel comprises a great deal of the overall cost of the structure. Hence, the design optimization of steel frames is focused on weight minimization in the literature based on the assumption that the use of least material leads to an economical design as well in terms of final cost of a structure. In this study, a discrete evolutionary algorithm (EA) is employed for the optimization of 2D and 3D steel frames and the achieved results are then compared with results achieved using a discrete particle swarm optimisation (PSO) and Big Bang-Big Crunch (BB-BC).

1.3 Aim

The research aim is to come up with a heuristic algorithm to be used by engineers and scientists in solving complex problems in structural steel by employing nature and bio-inspired algorithms in the industry. This is mainly due to complexity and non-linearity of the problems [15]. According to the literature [33], the computational drawbacks of existing numerical methods have forced researchers to rely on heuristic algorithms. Heuristic methods are powerful in obtaining the solution to optimization problems.

To achieve an optimum solution a developed computer aided numerical algorithm capable of optimizing the design of steel structures is applied to give an optimum solution. Due to continuous inflation of the cost of materials, transportation and construction costs, etc. an efficient algorithm is required to solve complex problems by mimicking the natural process of evolution and adaptation. It is common practice to observe structural safety always, while an economical design is pursued by the designer sometimes using intuition or experience, and occasionally using a trial and error process. However, despite the best effort of the designer, the optimum design cannot be reached in most cases, and even the design produced might sometimes be very far from the economical range [63].

The investigation and validation process of the research conducted are therefore to:

- Study the complexities of two dimensional (2D) and three dimensional (3D) steel structures in the real life environment.
- Ensure the design satisfies a practical design situation in which the most unfavourable loading cases are considered.
- Develop an efficient algorithm to optimise large complex structures with a high number of design parameters and provide an effective solution approach to tackle them.
- Understand the existing available optimization algorithms used.
- Run the simulations for a number of instances to verify the robustness of the proposed EA based approach in a 2D and 3D structural steel.

1.4 The outline of the thesis

The content of this thesis is divided into eight chapters. The first chapter provides a brief background on the history of steel structures and outlines the problem statement and research aim. It also reviews existing articles which are relevant in structural optimization and presents the investigation and validation process of the study.

The second chapter discusses briefly the algorithms used in steel structures, and the second part of it focuses on the design process of steel structures.

The third chapter focuses on finite element analysis with the first part of it introducing finite element analysis, what it is and how it works and why it is so important to use it when optimizing structural steel. The second part discusses the importance of things to take into consideration when performing a finite element analysis.

The fourth chapter discusses heuristic multi-purpose algorithms, Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Big Bang-Big Crunch (BB-BC). The common advantages and disadvantages are presented in detail for these population based methods. In chapter five, topology impact is looked at and how it influences the behaviour of the structural steel under optimization and the impact of the results when the topology of the structure is looked at in detail.

Chapter six considers some examples. Initially, the mutual efficiency of heuristic methods is compared in an academic test problem which is taken from the literature. The same problem is considered for optimization and results comparing mass, stress and deflection constraints using another population based method BB-BC. A number of problems ranging from simple to complex structural steel are optimized using BB-BC. Chapter seven discusses one of the most commonly used finite element analysis software (Prokon) in the structural design industry.

The results of thesis are summarised in chapter eight also giving some ideas for future research.

1.5 Chapter Summary

The first chapter provides a brief background on the history of steel structures and outlines the problem statement and research aim. It also reviews existing articles which are relevant in structural optimization and presents the investigation and validation process of the study.

Chapter 2

Design of structural steel

Presented here is a literature review of several research publications investigating the improved methods of optimization of steel structures using evolutionary algorithms to minimize weight. Structural optimization has been a topic of interest for over 100 years [62]. The 1960's showed simultaneously a flourishing due to the new finite element and mathematical programming algorithms, and a decay due to the lack of sufficient computer power to carry even the simplest optimization tasks.

This led in the 1970's to approximate structural synthesis techniques, which coupled with advances in computers allowed the solution of more significant problems, although mostly of academic interest. [62] in his survey states that for small and medium size problems the optimisation costs are small relative to the total computational effort, but as the number of design variables and constraints increase, conventional methods will not work, and new algorithms need to be developed. He goes on to suggest that the use of massive parallel or distributed computing systems as a possible practical solution. [1] researches about the development of efficient parallel optimization algorithms on shared memory multiprocessors such as encore multi-max and Cray supercomputer to optimise large structures on a cluster of workstations connected via local area network (LAN).

The selection of genetic algorithm is based on its adaptability to a high degree of parallelism. Two different approaches are used to transform the constrained structural optimization problem to an unconstrained optimization problem. A penalty function method and augmented Lagrangian approach. For the solution of the resulting simultaneous linear equations the iterative preconditioned conjugate gradient (PCG) method is used because of its low memory requirement. A dynamic load-balancing mechanism is developed to account for the unpredictable multi-user, multitasking environment of a networked cluster of workstations, heterogeneity of machines, and indeterminate nature of the interactive PCG equation solver.

Steel has been widely used for the construction of auto-mobile structures, bridges, ships, power lines and space stations etc. The use of steel frame structures for large, single and multi-storey buildings such as warehouses, distribution logistics centres, retail outlets, sports halls or the building frame of factories for commercial or industrial purpose permits the creation of buildings with large, uninterrupted floor areas [63]. The continuous inflation of the cost of materials, transportation and construction costs, etc. necessitates the development of computer aided numerical algorithms that are capable of optimizing the design of structures [10]. The main idea of the differential evolution algorithm is to use vector differences in the creation of new trial candidates to find better solutions [63]. For each population, the differential evolution algorithm iterates through the population and creates the trial candidate by vector mutation and a variant of uniform crossover.

According to Hultman [29], weight optimization of structures plays a major role in many engineering fields by minimising costs, leading to optimum material usage. In civil engineering, weight optimized structures are convenient since the transportation and construction work in connection with the build up is simplified. The advantage of having a weight optimized structure is that a minimum share of the load capacity is engaged by the structure itself. Structural optimization is also important in the aircraft and car industry where a lighter structure means a better fuel economy. Structural optimization is the subject of making an assemblage of materials sustain loads in the best way [12]. Optimum solutions are achieved by applying an efficient optimization technique called genetic algorithm (GA), as it is the most commonly referred to and probably the best known today. GA simulate the evolutionary principle of survival of the fittest by combining the best solutions to a problem in many generations to gradually improve the results. The initial population of solutions is created randomly, and as the evolution goes, the best individuals are combined in each generation until an optimal solution converges [29].

The study presented by Sirigiri [56], showing that tall buildings are spreading across the globe at an ever increasing rate. The global number of buildings, 200m or more in height, has risen from 286 to 602 in the last decade alone.This growth has been fuelled by a large variety of local and global motivations and therefore cannot be directly related to any single factor. Rapid economic growth in cities, scarcity of land and increasing population density demanded the need for tall buildings.

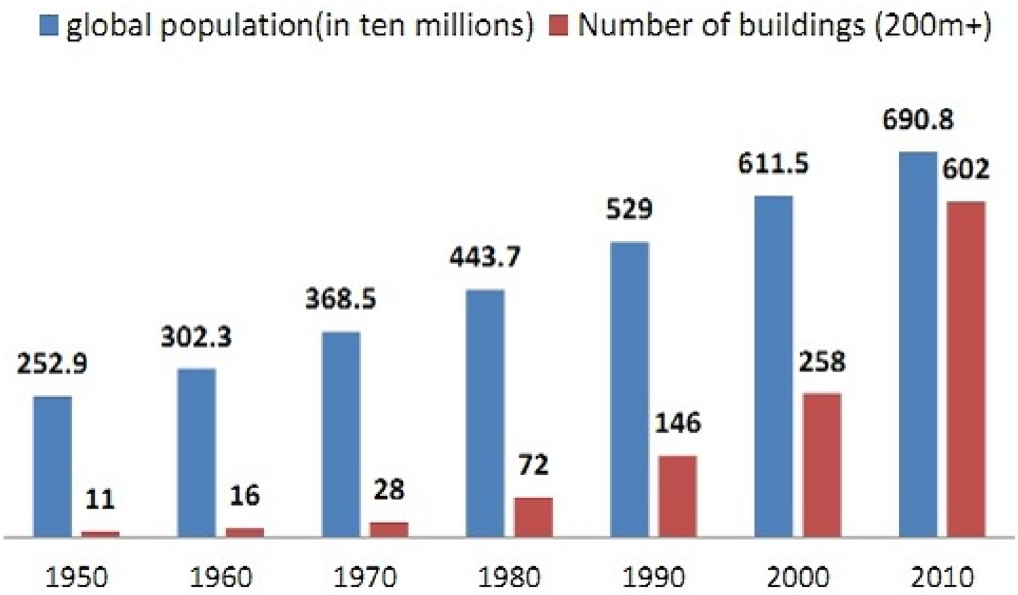


Figure 2.1: Global population and increase in number of buildings 200m+ (See Sirigi [56])

The increasing complexity of building architecture poses unique challenges in the structural design of modern tall buildings. Hence, innovative structural systems need to be evaluated to create an economical design that satisfies multiple design criteria. Design using traditional trial and error approaches can be extremely time consuming and the resultant design uneconomical.

2.1 Two dimensional (2D) steel structures

Two dimensional steel structures can be defined as structures, optimised within two plains either, the $x - y$ plains, $x - z$ or $z - y$ whereby the applied load is at a particular point on the above mentioned plains. The third plain is assumed to remain the same for all structures optimised. The minimum weight structures subjected to stress and displacement constraints are searched using computer aided design methods [53].The elements that form the structure are selected from the catalogues of available sections, chosen from commercially available standard profiles. Figure 2.2 below shows a typical example of an optimized steel structure.

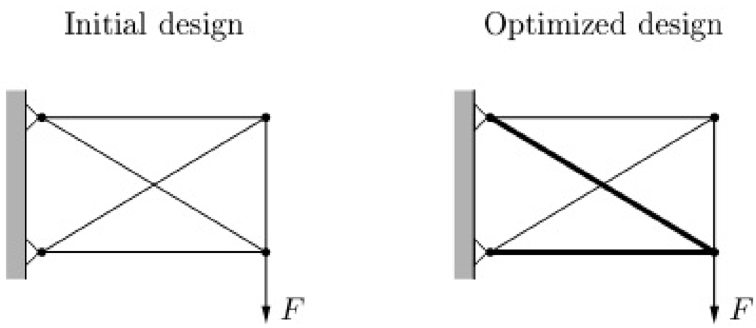


Figure 2.2: 2D Optimized structural problem [12]

Structural optimization problems are divided into three types, namely, sizing optimization, shape optimization and topology optimization as detailed by Christensen and Klarbring [12].

- Sizing optimization: In sizing optimization problems, the design variables are usually geometrical parameters such as thickness, length, width and cross sectional area of the optimized structure. A sizing optimization problem for a truss structure is shown in Figure 2.2.
- Shape optimization: Shape optimization deals with optimizing the overall shape of the structure such that the optimum design results in a structure with uniform stress distribution eliminating the stress concentration. In this case x represents the form or contour of some part of the boundary of the structural domain. The optimization consists in choosing the integration domain for the differential equations in an optimal way. The connectivity of the structure is not changed by shape optimization and new boundaries are not formed. A two-dimensional shape optimization problem is shown in Figure 2.3.

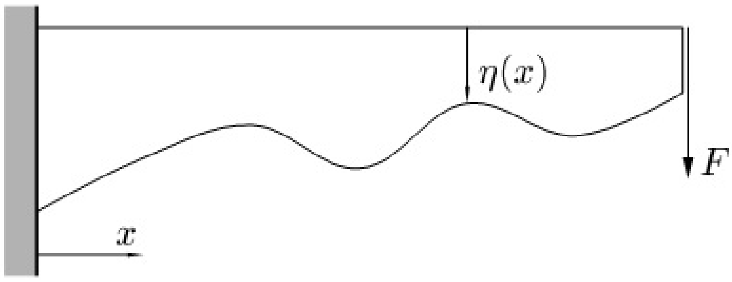


Figure 2.3: Shape optimization problem [12]

- Topology optimization: This is the most general form of structural optimization. In a discrete case, such as for a truss, it is achieved by taking cross-sectional areas of truss members as design variables and then allowing these variables to take the value zero, i.e., bars are removed from the truss. In this way the connectivity of nodes is variable therefore the topology of the truss changes as shown in Figure 2.4.

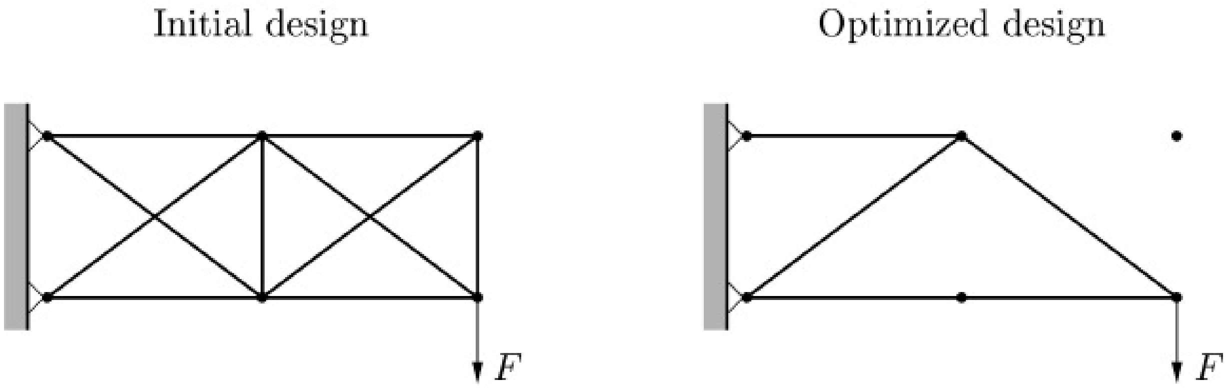


Figure 2.4: Topology optimization problem of a truss [12]

2.2 Three dimensional (3D) steel structures

The optimization algorithm for the minimum weight design of lateral load resisting steel frameworks subjected to multiple inter-story drift and member strength and sizing constraints in accordance with building code and fabrication requirements was developed by Chan and Wong [11]. The most economical standard steel sections to use for the structural members were automatically selected from commercially available standard section databases. Structural material scarcity and the need for efficiency in today's competitive world have forced engineers and reseachers to take greater interest in economical designs for structures [34].

In the 1960's Farkas and Jarmai [17], designed a series of roofs for vertical storage tanks for fluids covered by a soil layer. As shown in Fig. 2.3 below the roofs were constructed from welded stiffened plates.

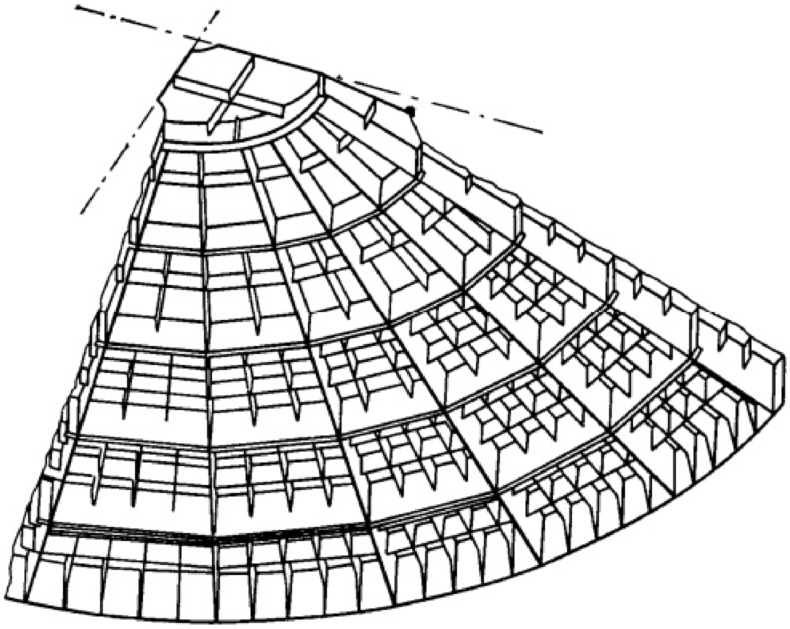


Figure 2.5: Strip section of a roof tank constructed from welded stiffened plates [17]

To achieve a minimum mass structure many thin ribs should be used, but this results in a very expensive structure, since the cost of welding is high. To minimize the cost, fewer and thicker stiffeners should be used. This optimization problem can be solved by the formulation of a mass or cost function and by the minimization of this objective function considering the design and fabrication constraints. For these constrained function minimization problems effective mathematical methods should be used.

2.3 The design process

According to the research conducted by Kirsch [41], the design process may be divided into four stages as shown below:

2.3.1 Functionality

The space required in an industrial building, the number of lanes required in a bridge, expected loads to be carried on a truss bridge etc. These are the examples of functional requirements, which are often established before entering the design process.

2.3.2 Conceptual design

This is the critical part of the design stage, because the designer is expected to do material selection, select the overall topology, and the type of structure using his ingenuity and engineering judgement to serve the structural systems functional purposes. For example, in a bridge design, the designer will decide whether it should be a truss bridge, an arch bridge or perhaps a cable stayed bridge with selected materials.

2.3.3 Optimization

Within the selected conceptual design considering desired constraints that satisfy the functional requirements achieving the optimal design. A typical example for a bridge design, would include the selection of the best geometry of a truss or the cross-sections of the members or minimizing the cost by using 7 least possible amount of material. Utilizing computers with optimization algorithms and software is most suitable to this step.

2.3.4 Detailing

After completion of the optimization stage, the results must be checked and modified if necessary. Engineering judgement, experience and the decision making process is necessary at this stage. This stage is usually controlled by market, social and aesthetic factors. Iterative procedures for the four stages are often required to find an acceptable final design. At the end, even the conceptual requirements are fulfilled, the final design may not be optimal. At that point, optimization techniques and computer aided design utilizing finite element method based software become the helpful and effective tools to make the best possible decision.

2.4 Chapter Summary

The second chapter discusses several research publications that are employed in investigating the improved methods of optimization of steel structures using evolutionary algorithms to minimize weight in 2D and 3D structures. Structural steel has been widely used for the construction of auto-mobile structures, bridges, ships, power lines and space stations etc. The continuous inflation of the cost of materials, transportation and construction costs, etc. necessitates the development of computer aided numerical algorithms that are capable of optimizing the design of structures using evolutionary algorithms for both 2D and 3D steel structures.

The second part focuses on the design process tools used in the design of steel structures. These methods are functionality, conceptual design, optimization and detailing. The design process methods become helpful and effective tools to make the best possible optimum solution.

Chapter 3

Employment of finite element method

3.1 Introduction

The finite element method (FEM) is an important numerical method used by modern engineers and mathematicians to analyse and solve simple to complex problems efficiently and effectively. The FEM for analysing structural parts has now been around for over 30 years, but although it is generally accepted as an extremely valuable tool, many engineers do not know how to go about using it and very few engineers understand it [6]. In the modern days it is commonly known as Finite Element Analysis (FEA). The typical problem areas of interest is used to employ FEM analysis of structural steel, fluid flow, heat transfer and electromagnetic potential. This method grew out of earlier techniques and has been quickly developed into a viable means of getting results quickly, with its integration with the modern computer. This means that an engineer can take a complex structure or component to model it and obtain the required results efficiently using computerised FEM. In this chapter, a simple introduction to the finite element method operation and its advantage of solving from simple to complex problems using matrix methods is provided.

3.2 Basic operation of finite element method

The first step of the FEM is to breakdown the structure into elements to make it suitable for numerical evaluation and implementation using the discrete method. This method is employed to simplify the problem solved to a finite number of unknowns. After this has been achieved, a final solution can be defined in terms of assumed approximating functions for each of the elements. These functions also known as interpolation functions are defined at the nodes of the elements which are usually placed on the element boundary, although it is possible to have interior nodes as well. The nodal values now become the new unknowns for the mathematical representation of the structure. The next step is the selection of the interpolation functions. These functions need to fill certain criteria and are typically chosen so that the field variables or its derivatives are across the element analysis. The degree of approximation depends on size and the number of elements, as well as the interpolation functions chosen.

The following steps are followed for Finite element analysis using a computer based technique:

1. The first step is to number the node elements of the structure.
2. Breakdown the structure into elements and number the elements that form the structure. The number of elements vary as per the problem being solved and therefore, the selection of elements are based on the problem solved.
3. Choose interpolation functions that are applicable to the structure using assigned nodes.
4. The properties for the individual elements are expressed using matrix equations calculated using the interpolation functions.
5. The matrix equations for the element properties are assembled to give the overall system equations. This enables the overall behaviour of the structure to be analysed.
6. The system equations are then solved to give the final results.

FEM has many advantages in solving structural problems. One of the principal advantages is that it can solve simple to complex structures in a few hours and it can also solve irregular shape problems without difficulty. Another added advantage is that it can allow the user to mesh (divide into elements) the model into different sized elements, which are useful when there are locations of high stress, so those areas would have a greater number of elements. Any type of load and boundary condition can be modelled by being replaced by equivalent nodal loads and boundary conditions.

3.2.1 Modelling consideration of finite element method

The first basic principle of finite element methods is discretisation of a structure into elements. Before the simulation process begins, the size of elements, shape, number and configuration of elements are selected carefully as they have an impact in increasing the computational efforts needed for obtaining the optimum solution. Some of the various considerations taken in the discretisation process are listed here below:

1. node location
2. element number
3. type of element
4. size of elements
5. simplifications afforded by the physical configurations of the body
6. finite representation of infinite bodies
7. node numbering scheme
8. automatic node generation
9. mesh size

The user inserts material properties for the problem being solved, and applies boundary conditions and applicable external loads.

3.3 Principle of finite element method

Alternative analysis methods are needed, and numerical methods are increasingly used to find close approximate solutions. The research by Forsythe and Wasow [21], introduced one of the more commonly used method called general finite difference scheme. However this method although capable of solving complex structures, was hampered by problems with irregular or unusual boundary conditions. The finite element analysis method grew out of the finite difference scheme and became widely adopted first in the aerospace industry to study stresses and then in other industries as its demand grew. It has been expanded to structural analysis, fluid flow, heat transfer and electromagnetic potential.

An example of the truss shown below in Figure 3.1 is used to indicate a simple model of a truss with nodes and applicable forces, where by the FEM is employed to explain the breaking up of the structure to be analysed into a finite number of parts or elements then analysing each part separately and then reassembling the structure to obtain the final results. In order to do this the nodes are used to link the elements together. This can be considered a piecewise polynomial interpolation. This means that a set of simultaneous algebraic equations is generated. With more complex structures, the equations become more numerous and complicated, hence the requirement and use of computers to solve them.

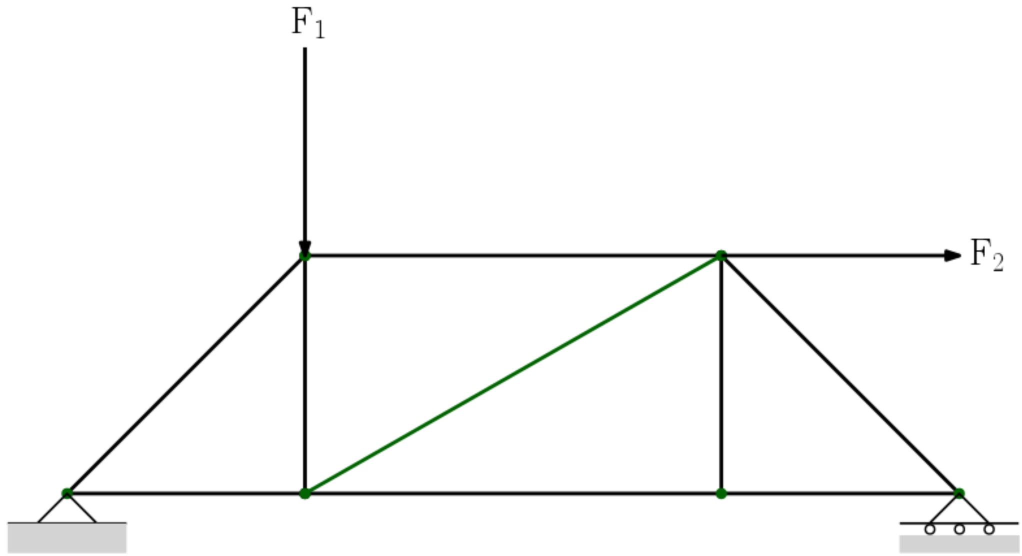


Figure 3.1: Truss with applied forces

Literature published by Benham and Crawford [6], is used to discuss the analysis of spring elements during FEM for structural steel. One of the examples as shown below in Figure 3.2 shows a spring with two ends able to move due to the deformation of the member and structural displacement.

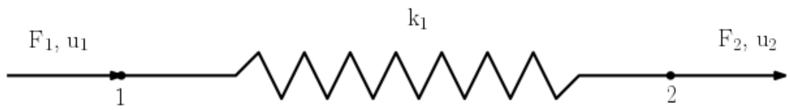


Figure 3.2: Single spring element [6]

The end points 1 and 2 are called *nodes*, where by F and u are the force and displacement values. The elements of the structure are considered as spring elements with the length and area A, k_1 is the stiffness of the spring given by the following equation:

$$k_1 = \frac{AE}{L}, \tag{3.1}$$

Where E is the Young's modulus for the material. The spring element, as shown in Figure 3.2 above, can be expanded into a stiffness matrix by first using the sign convention whereby the forces and displacements are considered positive in the x - direction. The following equations can be generated:

$$F_1 = k_1(u_1 - u_2) = k_1u_1 - k_1u_2 \tag{3.2}$$

$$F_2 = k_1(u_2 - u_1) = -k_1u_1 + k_1u_2. \tag{3.3}$$

Therefore the equations (3.2) and (3.3) can be written in a matrix form as:

$$\left\{ \begin{array}{c} \mathrm{F}_1 \\ \mathrm{F}_2 \end{array} \right\} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \left\{ \begin{array}{c} \mathrm{u}_1 \\ \mathrm{u}_2 \end{array} \right\}. \tag{3.4}$$

The size of the matrix depends on the the problem that is optimised or solved. In summary equation (3.4) can be re-written in the form:

$$\{F\} = |K^e| \{u\}. \tag{3.5}$$

The value of $|K^e|$ is the stiffness matrix for the spring element. Therefore, the spring element can be employed using matrix equations to solve simple to complex problems.

3.3.1 Definiton of element nodes and element geometry

Nodes are the end points of structural elements that interconnect the frame structure to be complete. As shown here below, in Figure 3.3, node numbering from 1 to 7 is made up of elements that are interconnected at nodal points to make up the frame structure. The nodes are also used as the boundary conditions to prevent the structure from rotating in X, Y and Z direction and displacement in x, y and z direction. Node point 1 and 7 are boundary conditions. Nodal forces applied at node points are uniquely defined by the displacement of nodes, for example node 4 is used to apply load P.

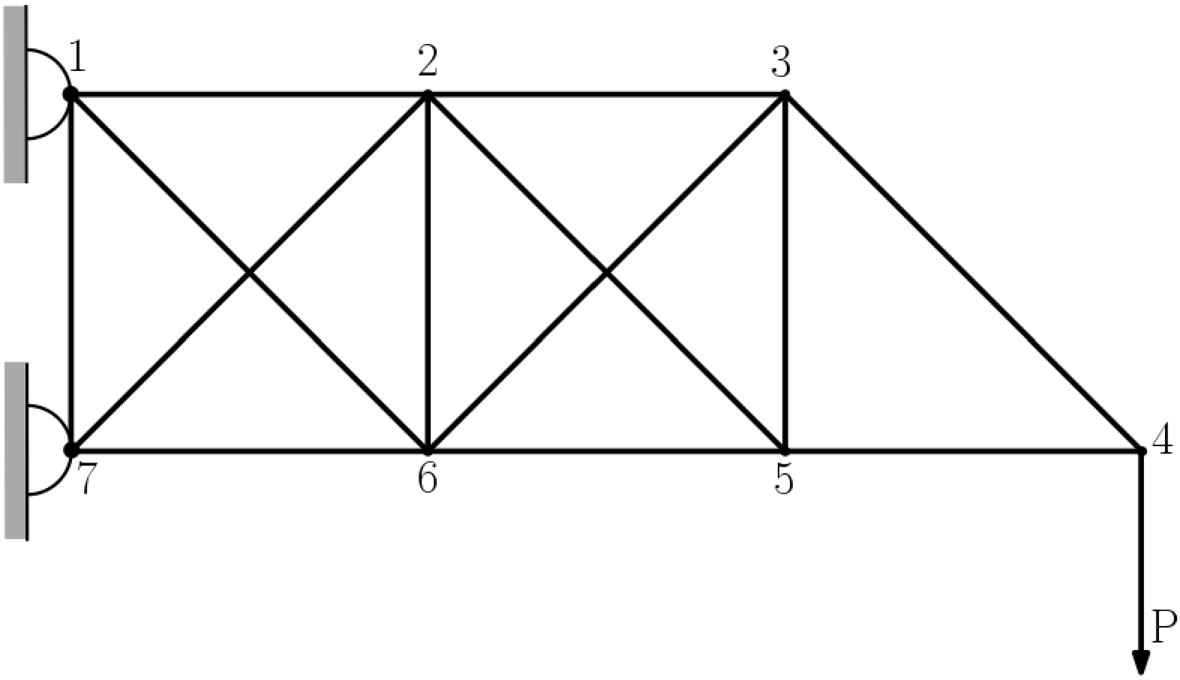


Figure 3.3: Node numbering

Each element possesses a set of distinguishing points called nodal points or nodes for short. Nodes serve a dual purpose: definition of element geometry, and home for degrees of freedom. When a distinction is necessary we call the former geometric nodes and the latter connection nodes.For most elements studied here, geometric and connector nodes coalesce.

Nodes are usually located at the corners or end points of elements, as illustrated in Figure 3.4. In the so-called refined or higher-order elements nodes are also placed on sides or faces, as well as possibly the interior of the element.

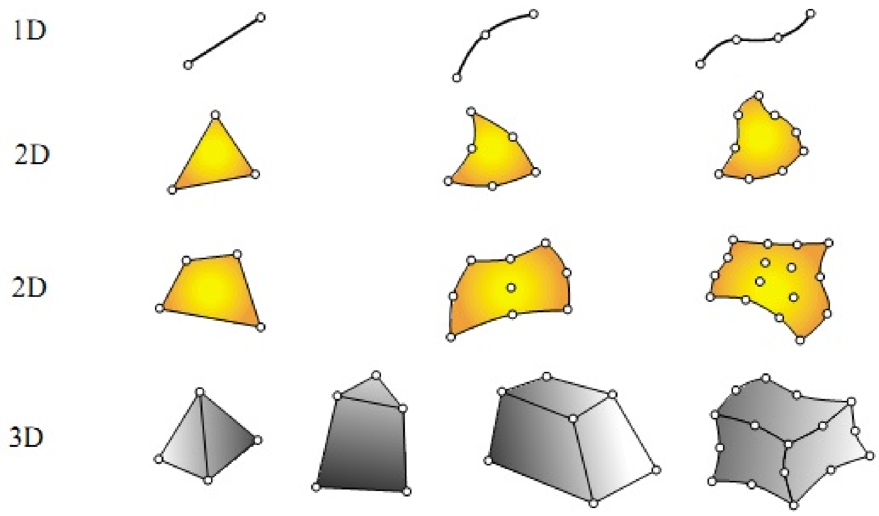


Figure 3.4: Finite element geometries in 1D to 3D [19]

The size of structural elements influences the convergence of the solution directly and that is why it is very important to choose the element size with care. If the selected size of the element is small, the final solution is expected to produce results that are more accurate. It is also important to remember that the use of the smaller size elements or mesh will result in more computational time required for the analysis.

The primary characteristics of a finite element are embodied in the element stiffness matrix. For a structural finite element, the stiffness matrix contains the geometric and material behaviour information that indicates the resistance of the element to deformation when subjected to loading. Such deformation may include axial, bending, shear, and torsional effects. For finite elements used in non structural analyses, such as fluid flow and heat transfer, the term stiffness matrix is also used, since the matrix represents the resistance of the element to change when subjected to external influences.

The truss shown in Figure 3.5 is made up of structural elements numbered from 1 to 12 connected at nodal points. These elements can be selected from different structural steel profiles like channels, I-beams, equal angle irons etc. The design engineer selects the required structural profile elements based on structural properties like cross sectional area, thickness, moment of inertia etc. required by the design.

When all the boundary conditions are inserted the equations of the system can be solved for the unknown stresses, displacements and the internal forces in each element obtained.

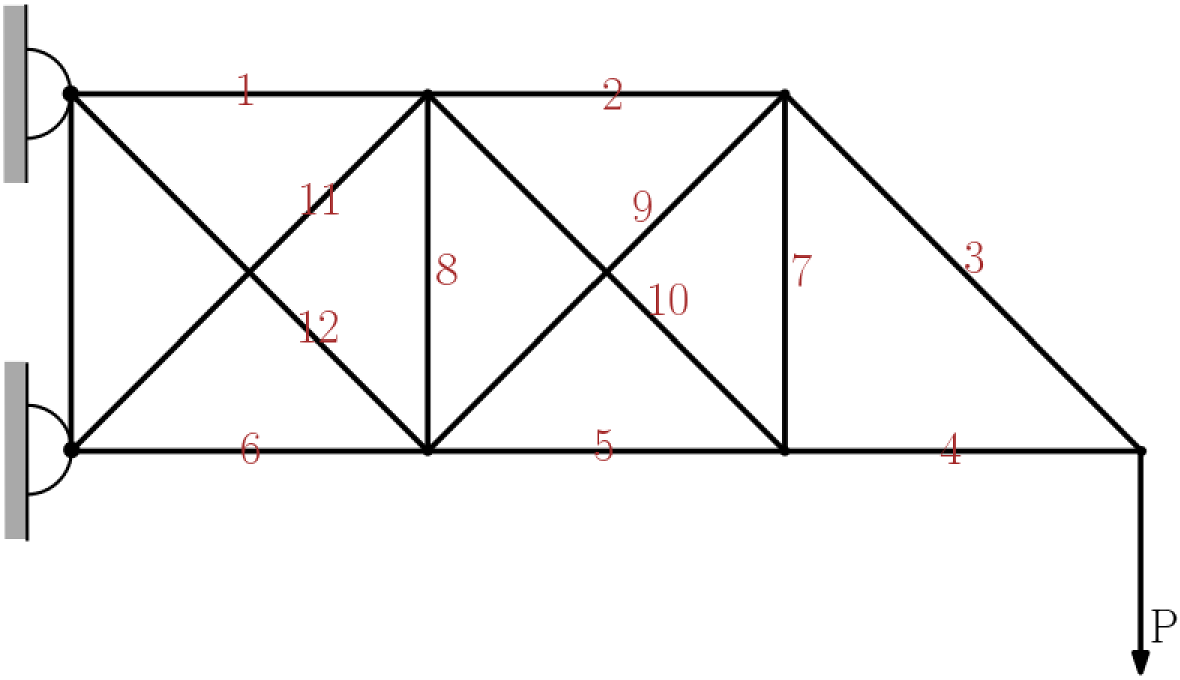


Figure 3.5: Structural elements

Selection of element geometry plays a big role in optimization of structural steel. The quality of results depends on the selected geometry.

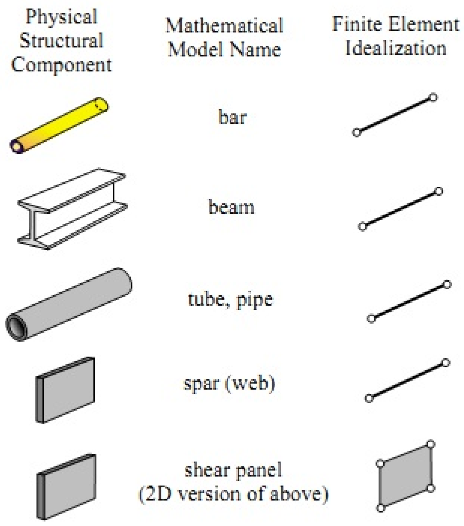


Figure 3.6: Primitive structural elements [19]

3.4 Continuum problems

The FEM arose out of the demand to analyse complex structures accurately. Before FEM was introduced, classical continuum methods were used to solve stress analysis problems. These methods had been developed over many decades and provided displacement and stress analysis results using various methods. However in order to use the classical methods many assumptions needed to be made. These included simplification of structures into two or three dimensions. As time goes on structures started to be more complex and the equations to model them became more and more complicated, so more sophisticated mathematical techniques were required to solve them. Often this resulted in inaccurate solutions. The typical design process then became a vastly simplified model which was used as a starting point to the design process. A prototype would then be built and tested and the final design would be developed from that.

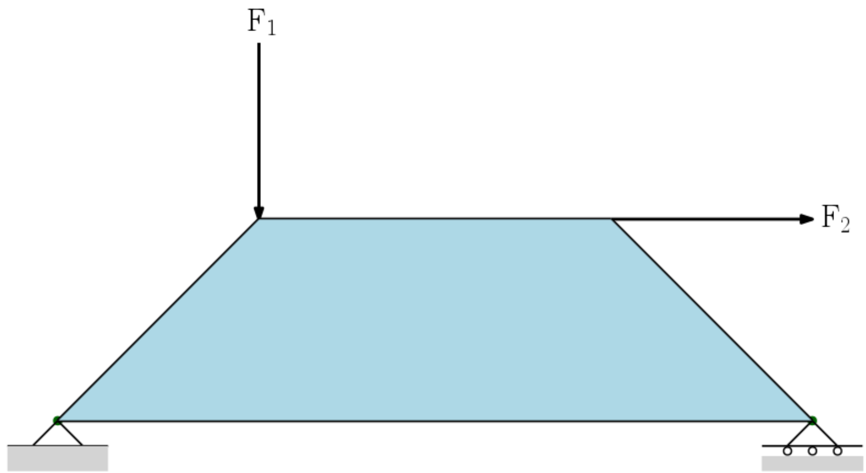


Figure 3.7: Plate with same shape as truss [6]

Continuum problems are based on the idea that all processes are given by field values that are defined at every point in space. The independent variables in continuum problems are the coordinates of time and space. Examples of continuum problems are those that involve temperature, electromagnetic fields, stress and displacement. All these problems arise from properties in nature that are given by partial differential equations and specified boundary conditions.

Continuum problems are sometimes called boundary value problems, since their solution is often wanted for a particular region specified by a boundary, on which boundary conditions are imposed. Boundaries are either open or closed. Open boundaries extend to infinity, and no boundary conditions are specified on the part at infinity [51]. Closed boundaries are those where conditions affecting the solution of the problem are specified everywhere.

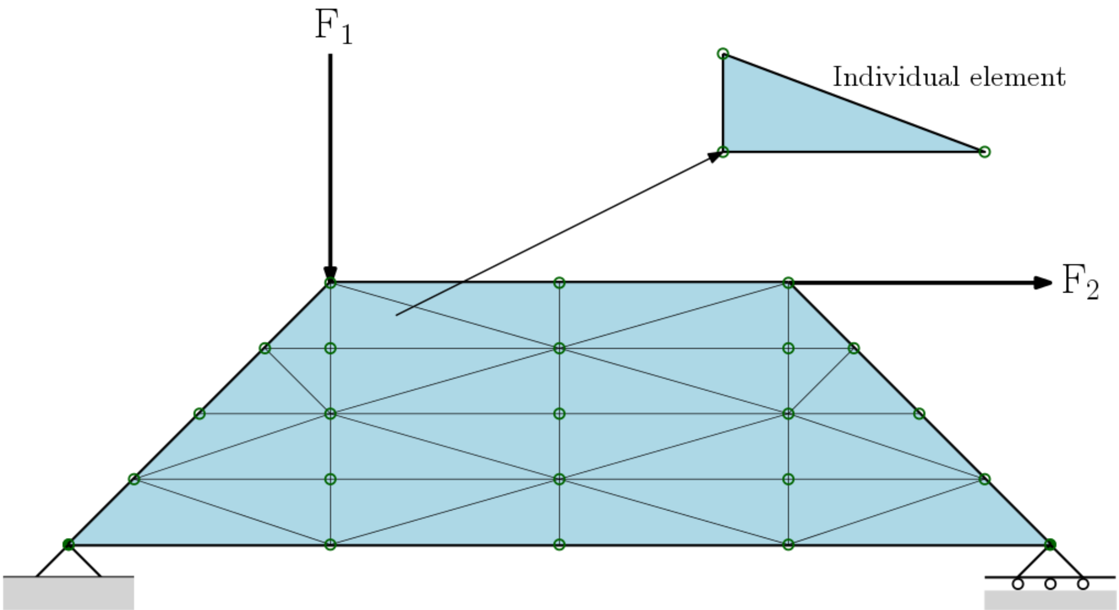


Figure 3.8: Meshed problem [6]

3.4.1 Problem statement

An example by Oden and Reddy [51], is used to define continuum problems using mathematical expressions. Where D is considered as a domain bounded by the surface $\sum$. The value of ϕ is a scalar function defined in the interior of D such that the behaviour of ϕ in D is given by;

$$L(\phi) - f = 0, \tag{3.6}$$

Where f is a known scalar function of the independent variables and L is a linear or non-linear differential operator. An assumption can be made that the physical parameters in the differential operator are known constants or functions. In n dimensions, second order of differential operators can be reduced, using suitable transformation to the form;

$$L() = \sum_{i=1}^{n} A_i \frac{\partial^2()}{\partial x_i^2} + \sum_{i=1}^{n} B_i \frac{\partial()}{\partial x_i} + ()C + D, \tag{3.7}$$

where A_i, B_i, C and D may be functions. The operator as given in equation 3.7 may be linear if A_i, B_i, C and D are functions only of the independent variables $(x_1, x_2, ..., x_n)$ and quasi-linear if $A_i, B_i, CandD$ are functions of x_i and the dependent parameter, as well as first derivatives of the dependent parameter. An operator is linear if;

$$L(f + g) = L(f) + L(g). \tag{3.8}$$

3.5 Chapter Summary

The third chapter focuses on finite element analysis with the first part of it introducing finite element analysis, what it is and how it works and why it is so important to use it when optimizing structural steel. Finite element method (FEM) is an important numerical method used by modern engineers and mathematicians to analyse and solve simple to complex problems efficiently and effectively. The FEM for analysing structural parts has now been around for over 30 years, but although it is generally accepted as an extremely valuable tool, many engineers do not know how to go about using it and very few engineers understand it [6]. In the modern days it is commonly known as Finite Element Analysis (FEA). The typical problem areas of interest is used to employ FEM analysis of structural steel, fluid flow, heat transfer and electromagnetic potential. This method grew out of earlier techniques and has been quickly developed into a viable means of getting results quickly, with its integration with the modern computer. This means that an engineer can take a complex structure or component to model it and obtain the required results efficiently using computerised FEM.

The chapter also discusses the steps to follow when conducting an FEA. A definition of FEA is explained in detail using mathematical equations and diagrams showing a simple model of a truss with nodes and applicable loads.

Chapter 4

Optimization of trusses using heuristic algorithms

A problem of saving material and cost cannot be overestimated in modern society. An optimal design optimization of complex structures often present significant difficulties and become even more challenging if the structure is made up of prefabricated elements. While methods of stress strain analysis developed faster, optimization techniques have been lagging behind. Requirements for modern structures and structural elements can be very high. Obviously, an accurate stress-strain analysis must accompany any optimization procedure. Various techniques were proposed initially for discrete optimization by researchers and engineers, see for example a review published by Arora, Hsieh and Huang [3]. However, a tremendous breakthrough came only with arrival of the evolutionary algorithms, particularly the genetic algorithms [28].

The genetic algorithm became the first evolutionary algorithm to win general acceptance and broad application see [24] and [27]. Although, the genetic algorithm greatly extended the scope of solved problems, the high dimensions of a functional (search) space remained a formidable obstacle to an effective optimization. The arrival of Particle Swarm Optimization (PSO) algorithm introduced by Kennedy and Eberhart [37] and more recently, the Big Bang - Big Crunch (BB-BC) optimization method introduced by Erol and Eksin [14], have greatly expanded dimensionality of optimizing problem. These are heuristic algorithms that incorporate random variation and selection. The selection done randomly and the information obtained in each cycle are used to choose the new points in the subsequent cycles. These algorithms do not require a given function to be derivable and an explicit relationship between the objective function and constraints is not needed.

The selection of heuristic optimization algorithms is large and there are also several different combinations. In many versions the idea is taken from nature like the evolution or the behaviour of the swarm. The wide applicability and the ability to solve, at least approximately, computationally hard discrete and mixed integer problems have made heuristic algorithms popular in structural optimization. This thesis considers the listed below population based heuristic algorithms and discuss in details:

1. Big Bang - Big Crunch Algorithm
2. Particle Swarm Optimization
3. Genetic Algorithm

Design optimization of frame structures are inclined to select suitable sections for elements that fulfil all design requirements while having the lowest possible cost. Optimization of structural steel is a critical and challenging activity that has received considerable attention in the last two decades. A large number of design variables, and increase of the search space and controlling a great number of design constraints are major preventive factors in performing optimum design in a reasonable computational time. Computational researchers have been greatly interested in the natural sciences to model and solve complex optimization problems by employing stochastic search techniques that mimic nature [15]. Such techniques are genetic algorithms that make use of the idea of survival of the fittest. Ant colony optimization imitates the way that ant colonies find the shortest route between the food and their nest [7]. The Big Bang-Big Crunch (BB-BC) optimization algorithm was introduced by Erol and Eksin [14], in 2006 as a new evolutionary algorithm. According to the authors, the BB-BC algorithm is based on the evolution of the universe. In the Big Bang phase, the energy dissipation produces disorder and randomness is the main feature of this phase, while in the Big Crunch phase, these points are drawn into a dense cluster with the center of gravity being the optimum solution of the optimization problem. By means of these techniques it became possible to determine the solution of discrete structural optimization problems more efficiently than with those based on mathematical programming methods. This is mainly due to complexity or non-linearity of the problems. These algorithms have been successfully applied to solve computational, complex and non-linear problems from different disciplines [15].

4.1 Big bang - big crunch algorithm

The complexity of structural steel and the need for efficiency in today's competitive world have forced engineers to have greater interest in the economical design of structures using evolutionary algorithms. The evolutionary algorithms provide efficient tools for performing structural optimum designs, in problems considered highly complicated and requires the use of a reliable multi-dimensional optimization method. With an increasing number of members, the terrain of the functional multi-dimensional space becomes very complex and the use of calculus-based methods proves to be ineffective. Such methods suffer from the lack of robustness, and therefore they are hampered by inauspicious features in the multi-dimensional space like "ridges", "canyons", "flat spots"and multiple extrema [34]. In addition to these limitations, they are local in scope; the optima they seek are the best in a neighbourhood of the current point.

The BB-BC algorithm is employed, since it seems to be the most effective optimizing technique for such types of problems [35]. According to the authors in [60], the BB-BC algorithm relies on one of the evolution theories of the universe, namely the Big Bang - Big Crunch theory. According to this theory, in the Big Bang phase the population of feature vectors randomly fills the space, while in the Big Crunch phase these points are drawn into a dense cluster with the centre of gravity being the optimum solution of the optimization problem. The BB-BC method has quickly demonstrated its superiority over other heuristic population-based search techniques when employed to perform structural optimization tasks. For example, for the optimal design of space trusses [9], skeletal structures [34], for parameter estimation in structural systems [60].

The BB-BC algorithm is a heuristic population based evolutionary optimization method. Among the merits of this method are computational simplicity, ability to handle multi-dimensional problems and very fast convergence. An amazing feature of the algorithm is that the convergence rate (space contraction) is independent of the dimensionality. However it seems that the implementation of it can be problematic when a noisy multimodal functional space is encountered, where there are few local minima or maxima of a similar magnitude. Fortunately, such problems are not common in structural design problems. The obtained results show that an optimum or near-optimum solution can be achieved efficiently and very fast even in highly complex multi-dimensional spaces. Obviously, the algorithm can be successfully used in many other engineering applications, see for example [59].

4.1.1 Big bang - big crunch problem formulation

The optimization problem can be stated as an extreme-value problem where the main objective is to find such a set of design parameters $(A_1, A_2, A_3, ..., A_N)$ which minimize the weight of the truss subjected to stress and strain constraints. Here A_i, $(i = 1, 2, 3..., N)$ are cross-sectional areas of the profiles selected from the given catalogue; N is the number of members in the truss. If the catalogue consists of C different elements, then the number of possible combinations is calculated as C^N, which becomes an astronomical number in the case moderately complex problems.

The truss weight, W, is calculated as

$$W = \sum_{i=1}^{N} \rho_i \ A_i \ l_i \to min, \tag{4.1}$$

where ρ is the material density and L_i are lengths of the truss members. Strictly speaking, we must also take into account additional metal parts used for connecting the members at joints. The optimizing problem is unconstrained and an appropriate penalty function should be used. Thus, for the truss made of N members and M joints, the fitness function f can be written as follows,

$$f = W + \alpha \ max \left(\frac{|\Delta_j|}{\Delta_{max}} - 1, 0 \right) + \beta \ max \left(\frac{|\sigma_i|}{\sigma_{max} - 1, 0} \right). \tag{4.2}$$

Here, $\Delta_j, j = 1, 2, 3, ..., M$ is the current displacement; $\sigma_i, i = 1, 2, 3, ..., N$ is the current stress. The maximum allowable displacement and stress are Δ_{max} and σ_{max}. The penalty coefficients, α and β, are problem dependent and must be large enough, in the range of $10^3 - 10^6$.

4.1.2 Big bang - big crunch algorithm

The optimization problem can be stated as extreme value problem where the main objective is to find such set of parameters $(x_1, x_2, ..., x_n)$ which maximise or minimise a quantity dependent upon them. The objective function is achieved when finding the minimum possible weight of the structure. The BB-BC optimization procedure can be briefly outlined as follows:

The initial population of feature vectors is randomly generated and spread over the entire search space, allowing also some individuals (within the range of 10%) to be generated outside the search space. Then all the points which fall outside the prescribed limits are placed at the boundaries. This will guarantee that the optimum solution point will not fall outside the domain filled in by the candidate points. The number of individuals in the population must be big enough in order not to miss the optimum point. However, the population size can be significantly reduced as the search domain shrinks. The fitness values are computed for every individual and then the centre of mass is calculated as follows,

$$x_c^{(i)} = \frac{\sum_{k=1}^{N_{pop}} f_k x_k^{(i)}}{\sum_{k=1}^{N_{pop}} f_k}, i = 1, 2, 3, ..., N, \tag{4.3}$$

where N is the number of parameters and N_{pop} is the population size. Based on the location of the centre of mass the boundaries of new contracted space are determined by

$$B_i = \frac{|x_{max}^{(i)} - x_{min}^{(i)}|}{\mu N_{gen} + 1}, \tag{4.4}$$

where N_{gen} is the current generation (iteration) number and μ is a contraction ratio, usually in range of $\mu = 1 \div 10$. The limits of the parameters in the above expression are calculated as follows:

$$x_{min}^{(i)} = \gamma x_c^{(i)} + (1 - \gamma) x_{best}^{(i)} - B_i \tag{4.5}$$

$$x_{max}^{(i)} = \gamma x_c^{(i)} + (1 - \gamma) x_{best}^{(i)} - B_i. \tag{4.6}$$

Here the empirical parameter γ $(0 \leq \gamma \leq 1)$ controls the influence of the global best solution $x_{best}^{(i)}$ on the boundaries of the new search space. In our case $\gamma = 0.7$. The new search space is now randomly filled with points and thus a new population is created. Hence the algorithm is repeated until the stop criteria are met. As the search space is contracted with each new iteration the algorithm arrives at the optimum point very fast.

4.2 Population based methods

The Genetic algorithm and particle swarm optimization are population based methods. These methods differ from the local search methods in such a way that there is a group of solutions instead of a single solution in GA and PSO. The population based methods are assumed to be a group of algorithms that can offer some extra benefits compared to a single individual. Different solutions can spread out into the search space which is explored more widely than by using only one solution (explicit parallelism). The number of rounds become smaller, while on the other hand parallel solutions are increased by the amount of calculations per iteration round. The efficiency of a group in a GA is based on competition and in PSO the efficiency is based on co-operation between individuals.

The analyses of population members are independent from each other. This has made the use of parallel processing to be user friendly because it reduces the calculation time efficiently and economically. The optimization algorithm runs in one computer and divides the analysis tasks to several other computers to be done simultaneously as it is illustrated in Kere and Jalkanen [38].

When using a population based method, there is no need to choose an initial guess because the first group of solutions are usually selected randomly. The efficiency of the selected method in a local search algorithm depends on the initial guess. If the initial guess is poor the results will also be poor and if the initial guess is close to the optimum, the convergence may be fast. This gives a possibility to an experienced user to use his or her professional skills and intuition in such a way that is not possible in population based methods. In some cases it can be really difficult to find a feasible initial solution and population based methods with randomly selected initial population make starting easier.

4.2.1 Particle swarm optimization

Optimization provides engineers with a variety of techniques to solve problems. These techniques can be categorised as two general groups, classical methods and meta-heuristic approaches. Classical methods are often based on mathematical programming, and many of meta-heuristic methods make use of the ideas from nature and do not suffer the discrepancies of mathematical programming. One of the meta-heuristic algorithms is Particle Swarm Optimization (PSO), this algorithm is based on the behaviour reflected in flocks of birds, bees and fish that adjust their physical movements to avoid predators, change of geographical area due to weather changes and seek for food [45]. When these animals move, there are implicit rules that each member of bird flock and swarm of bees has to abide by so that they can move in a synchronised manner without colliding.

During the movement each member in a flock keeps an optimum distance from the neighboring individuals so that the flock can move smoothly from a point of depature to the next point of arrival. Kennedy [36], defines PSO as a simulator of social behaviour, used to realize the movement of a birds flock. The algorithim is population based, having its population called swarm and each individual in the swarm called particle. Each particle flies through the problem space to search for optimum. Amongst other research communities the PSO method has been given considerable attention in recent years. According to Angeline [2], PSO is the most successful swarm intelligence inspired optimization algorithm. However the local search capability is poor since premature convergence occurs often. PSO uses two approaches to obtain integer numbers from continuous ones. The first one is discussed by Kennedy and Eberhart [37], where binary numbers are used in particle swarm optimization to achieve a discrete set. The second method in the algorithm is to round off the real optimum value to its nearest integer number in each iteration [45].

For particle, i, in the PSO approach the new position of $\mathbf{x}_{k+1}^i$ depends on the current position of $\mathbf{x}_k^i$ and so called $\mathbf{v}_{k+1}^i$, where;

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i. \tag{4.7}$$

In Eq. (4.8) below the velocity is calculated as follows:

$$\mathbf{v}_{k+1}^i = w\mathbf{v}_k^i + c_1 r_1(\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_2(\mathbf{p}_k^g - \mathbf{x}_k^i). \tag{4.8}$$

The best position for particle i is $\mathbf{p}_k^i$ and the best feasible position for the whole swarm is $\mathbf{p}_k^g$. Having w called inertia, r_1 and r_2 are called uniform random numbers r_1, r_2 ϵ [0,1] and c_1

and c_2 are the scaling parameters. The inertia value w controls how widely the search process is done in the search space. The value of c_1 indicates how much a particle trusts itself and c_2 indicates how much it trusts the swarm. The Equation 4.8 above indicates the idea of the last two terms connected to c_1 and c_2 and it directs the optimization process towards good potential areas in the search space as shown in Figure 4.1 and 4.2. Where w is $0.8 < w < 1.4$ and $c_1 = c_2 = 2$ are selected. The value of w can be changed dynamically so that it is bigger during early iteration rounds and becomes smaller later when there is time to focus on promising areas.

Basically the PSO algorithm imitates the social behaviour of a swarm e.g. birds or fish in nature. It is an algorithm used for continuous unconstrained optimization problems. As shown in equation 4.7, discrete design variables can be taken into account by simply rounding each design variable to the closest allowed value. Constraints can be handled again by penalizing unfeasible solutions according to the unfeasibility.

Figure 4.1 below shows how the swarm moves from point $\mathbf{x}_k^i$ to $\mathbf{x}_{k+1}^i$ in PSO algorithm. The inertia term $w\mathbf{v}_k^i$ widens the optimization process and the terms connected to $\mathbf{p}_k^g$ and $\mathbf{p}_k^i$ direct the search towards the optimum known solution.

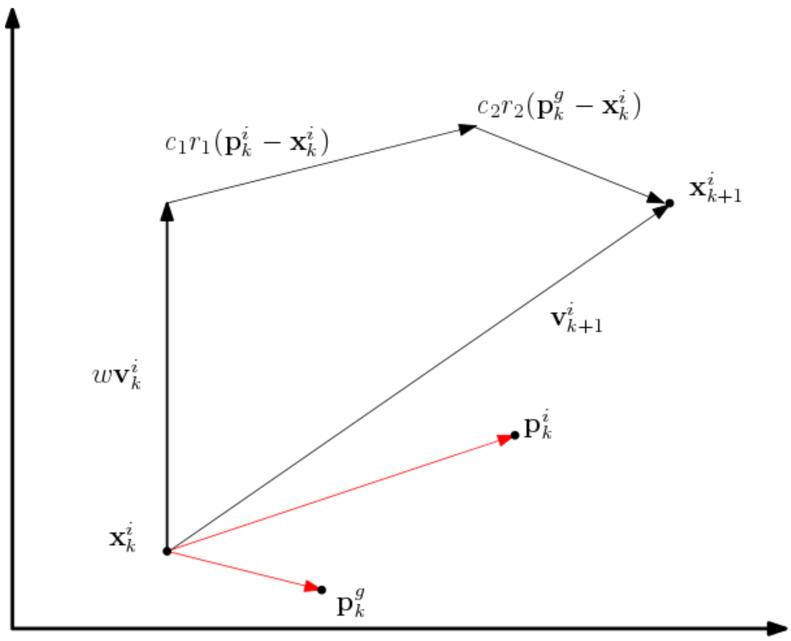


Figure 4.1: Swarm from point $\mathbf{x}_k^i$ *to* $\mathbf{x}_{k+1}^i$

The same process in Figure 4.1 is repeated in Figure 4.2 below, whereby the the swarm moves from point $\mathbf{x}_{k+1}^i$ to $\mathbf{x}_{k+2}^i$ in the PSO algorithm. The inertia term $w\mathbf{v}_k^i$ widens the optimization process and the terms connected to $\mathbf{p}_k^g$ and $\mathbf{p}_k^i$ direct the search towards optimum known solution.

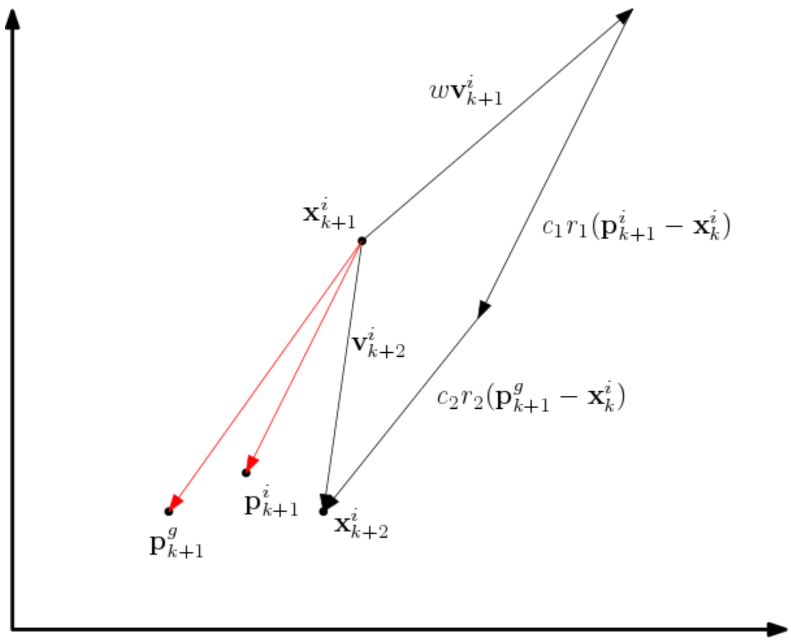


Figure 4.2: Swarm from point $\mathbf{x}_{k+1}^i$ *to* $\mathbf{x}_{k+2}^i$

In the PSO algorithm, the initial swarm and velocities can be selected randomly. As a terminating criterion the given number of iteration rounds can be used or the best known feasible objective function value can be observed and if there is no improvement during the few last rounds the optimization is terminated. The flowchart of the basic PSO algorithm is presented below Figure 4.3, and for further discussions see [52].

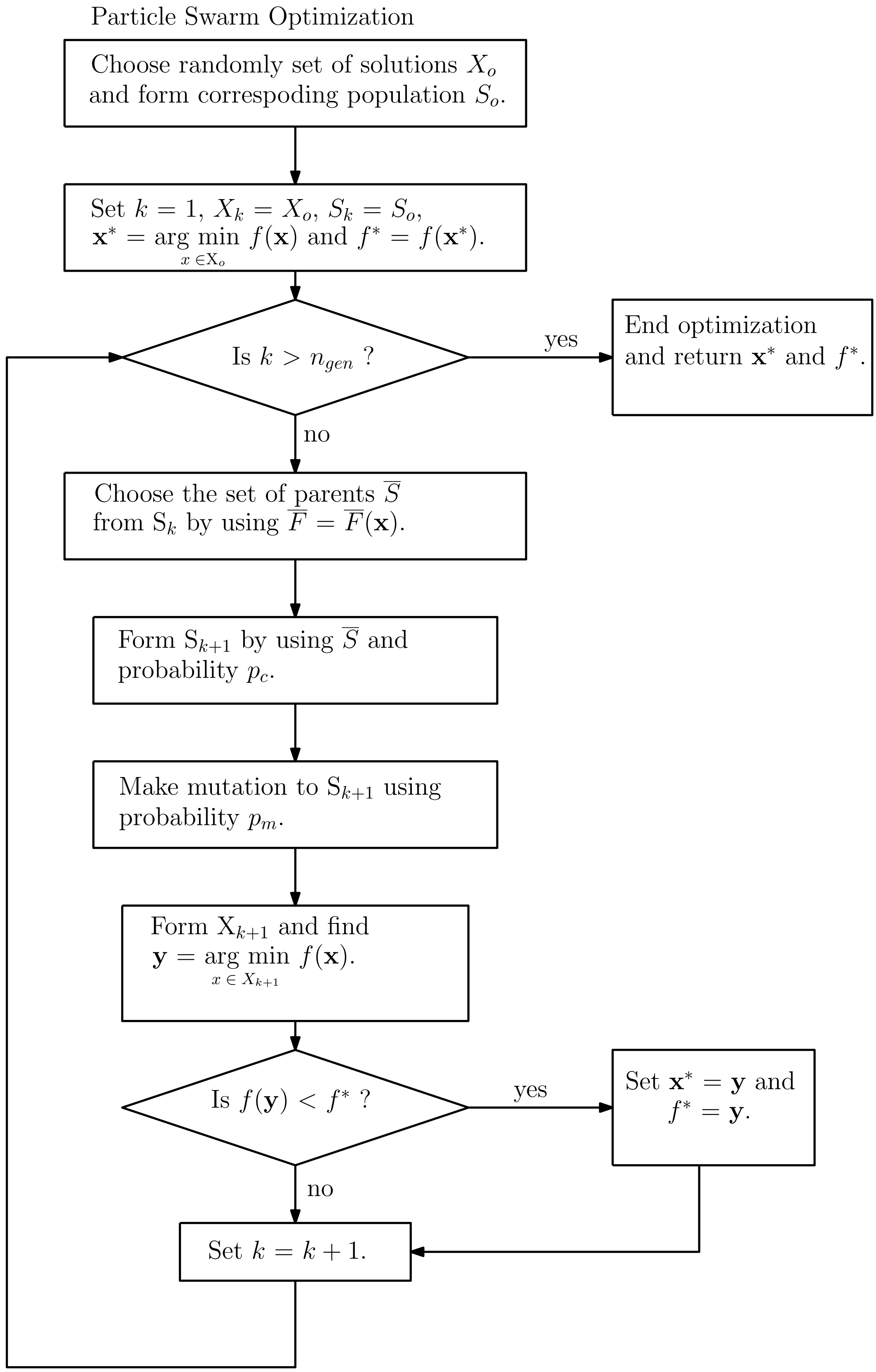


Figure 4.3: PSO algorithm flow chart

4.2.2 Genetic algorithm

Optimization of large steel structures using genetic algorithms (GA) has been widely used by engineers and researchers for optimum solutions. In Goldberg and Hollard [26], the GA is defined as a stochastic search technique, inspired by natural evolution, hereditary and survival of the fittest . Genetic algorithms are particularly effective for non-differentiable and discontinuous problems, and have been discovered more in recent years, and the research on genetic algorithms has spread from computer science to engineering and more recently, to fields such as molecular biology, immunology, economics, and physics [48]. The research by Michalewicz [47], on evolution programs uses the similar methodology though it promotes different data structures and corresponding adapted operators. All these search methods mimic nature by applying evolution strategies, motivated by natural hereditary and survival of the fittest. The advantages of applying a genetic algorithm to optimised design of structures includes discrete design variables, open format for constraint statements and multiple load cases [8].

In genetic algorithms the individuals in a population become better and better during the optimization process by simulating natural evolution processes using three major operations selection, crossover and mutation. The idea of genetic algorithm is to mimic Darwin's principles of natural selection and survival of the fittest individual. The origin of GA is in mid seventies by Holland [28], and since then, they have become the most popular heuristic optimization method.

The individuals of the populations are the encoded solutions of the optimization problem. The most popular method is to use binary coding in which a binary string corresponds to each solution. Each individual in the population is represented by a binary string which simulates the chromosome of natural species. Each bit in the binary string stands for the gene. The process continues for several iterations and the solution is given by the best found design. Another way to conduct encoding is to number the allowed discrete variable values from one forward and use these integers in encoding. The number of different codes is increased by this method. The length of individuals is decreased and prevents turning up such coded individuals that do not have counterparts among possible candidate solutions.

The quality of an individual is determined using a fitness function. This function is derived from the objective function and only positive values are obtained. In the minimization problem, the fitness function is formed using the simplest way as shown in Equation 4.9, where

$$\overline{F}(\mathbf{x}) = C - f(\mathbf{x}), \tag{4.9}$$

and C represents a big positive constant. In constraint optimization problems the feasibility of a solution has to be considered also in the fitness function. This can also be done using a local search algorithm called simulated annealing by penalizing the unfeasible solutions according to the unfeasibility given by,

$$\tilde{f}(x) = f(\mathbf{x}) \left[1 + \sum_{g_i(\mathbf{x})>0} R \, g_i(\mathbf{x}) \right], \tag{4.10}$$

where $\tilde{f}$ is the penalized objective function and R is the penalty. A suitable value for R is selected so that if the R value is too small, unfeasible solutions become too good and if R value is too big, only slightly unfeasible solutions become too poor. If all constraint functions have the same penalty, scaling of the penalty has to take place so that function values are in the same order of magnitude.

According to Pyrz [53], the evolutionary algorithm used to find the optimal truss can be stated as follows:

- Creation of an initial population of trusses
- Evaluation of the initial population by determining the fitness values
- If the termination condition is not satisfied do creation of new designs by recombining structures of the previous population i.e. selection of elements for mating, crossover and mutation.
- Selection of a new population of truss structures
- Evaluation of the new population of trusses

The integer encoding of design parameters using a mapping to discrete catalogue values of cross sectional areas has been applied in the presented approach. The chromosome representation of a truss structure composed of N bars is defined by a string of N corresponding integers I_i $(i = 1, ..., N)$. The value of I_i indicates the number of i-th bar cross section in the catalogue of K_i available elements $(1 \leq I_i \leq K_i)$. The formulation of the fitness f is based on the exterior penalty function and includes normalised constraint violation terms, weighted by penalty coefficients, as follows

$$f = C - \frac{W}{W_{max}} - \sum_{i=1}^{N} \alpha_i g_i^{\sigma} - \sum_{j=1}^{M} \beta_j g_j^{u}, \tag{4.11}$$

$$g_i^{\sigma} = max\left(\frac{|\sigma_i| - \sigma_{max}}{\sigma_i^*}, 0\right), \; g_j^{u} = max\left(\frac{|u_j| - u_{max}}{u_j^*}, 0\right), \tag{4.12}$$

where W_{max} is the maximal possible weight for available catalogues, σ_i^* and u_j^* are reference values, α_i, β_j are penalty coefficients, and C is a constant term to assume positive fitness. The values of W, σ_i and u_j result from the static analysis of the truss.

4.2.2.1 Selection

The selection process starts by sorting the population according to the values of the fitness function thus constructing a ranking. Individuals having a better solution obtain a higher rank. The individuals in the population are then selected in such a way that individuals having a higher ranking have a higher probability of being selected for reproduction in the next generation. For individual, i, to be selected in a population, the probability of selection p_i, depends on the values of the fitness function,

$$p_i = \frac{\overline{F_i}}{\sum_j \overline{F_j}}. \tag{4.13}$$

The basic idea behind selection is that it should be related to the fitness of each individual. The original scheme for its implementation is commonly known as the roulette wheel selection, because a common method of accomplishing this procedure can be thought of as roulette wheel being spin once for each available slot in the population. Where each solution has a slice of the roulette allocated in proportion to their fitness score as shown in Figure 4.3. In this scheme it is possible to choose the best individual more than once and chances are that the worst individual has a very slim chance of being selected.

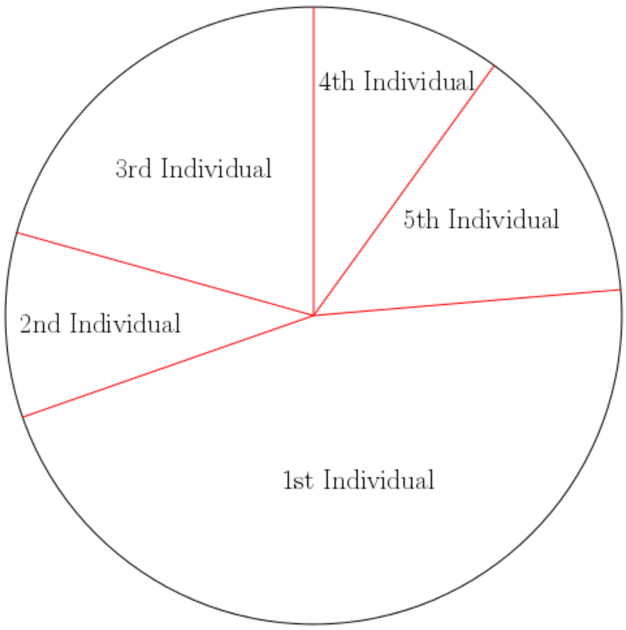


Figure 4.4: A roulette wheel with 5 slices

The above roulette wheel selection is one of the traditional GA selection techniques. Other selection methods are e.g. ranking selection and tournament selection. In roulette wheel selection the reproduction operator, is the proportionate reproductive operator where a string is selected from the mating pool with a probability proportional to the fitness. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individuals fitness values. A target value is set, which is a random proportion of the sum of the fitnesses in the population. The population is stepped through until the target value is reached. This is only a moderately strong selection technique, since fit individuals are not guaranteed to be selected for, but somewhat have a greater chance.

In ranking selection the proportions of the fitness function values are insignificant and the order of magnitude determines directly the probability of selection. Another possible selection method used is called tournament selection. In tournament selection individuals are randomly grouped in order to select the best individual as a new parent. There are only two candidates in the binary tournament selection. If one candidate is feasible and the other is unfeasible, the feasible one is selected. In the case that both individuals are unfeasible it is not necessary to use the penalty function approach because the less feasible can be selected otherwise. In other words it is not necessary to choose proper penalty parameters in the binary tournament selection process.

According to Back and Schwefel [4], this method called proportional selection, requires positive fitness values and a maximization task, so that the scaling functions can be utilized to transform the fitness values accordingly. Rather than using absolute fitness values, ranked-based selection methods utilize the indices of individuals when ordered according to fitness values to calculate the corresponding selection probabilities. Linear as well an non-linear mappings have been proposed for this type of selection operator. Another alternative called tournament selection works by taking a random uniform sample of a certain size $q > 1$ from the population, selecting the single best of these q individuals to survive for the next generation, and repeating the process until the new population is filled. This method gains increasing popularity because it is easy to implement, computationally efficient, and allows for fine-tuning the selective pressure by increasing or decreasing the tournament size q. While most of these selection operators have been introduced in the framework of a generational genetic algorithm, they can also be used in combination with the steady state and generation gap methods and of course in combination with the other branches of evolutionary computation.

4.2.2.2 Crossover

The crossover operation serves to create new individuals or produce offspring to the next generation. The crossover operation randomly cuts two parents into 2 pieces and joins two offspring from these parts. Two new off-springs are then generated. The number of pieces can vary from one point to two point crossovers. The crossover happens by some probability, p_c, and by probability, $1 - p_c$, where parents are directly copied to the next generation. The new off-springs have a better chance to be better than their parents. Usually, $p_c = 0.6. . . 0.8$, is known as the magnitude of crossover probability. In most cases the population size is kept constant and each pair of parents produce two off-springs.

Selection and crossover combine properties that already exist in an initial population. No additional information is brought into the system to cause a homogeneous population.

Crossover is further discussed in [4], where the variation operators of canonical genetic algorithms, mutations and recombinations are typically applied with a strong emphasis on recombination. The standard algorithm performs a so called one point crossover, where two individuals are chosen randomly from the population. A position in the bit-strings is randomly determined as the crossover point, and an offspring is generated by the two point recombination operator used in canonical algorithms as found in [4], is shown below:

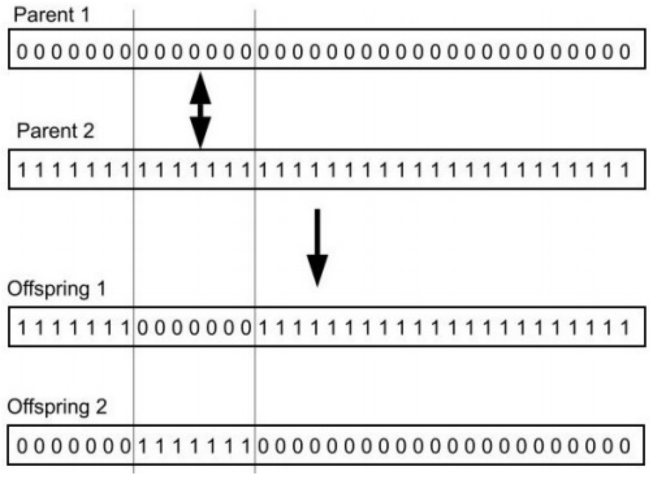


Figure 4.5: Two-point recombination operator used in canonical algorithm

The two recombination positions are chosen randomly. Numerous extensions of this operator, such as increasing the number of crossover points, uniform crossover (each bit is chosen randomly from the corresponding parental bits) and others have been proposed but as is the case in similar evaluation strategies no generally usefully recipe for the choice of a recombination operator can be given. The theoretical analysis of recombination, is still to a large extent, an open problem . Recent work on multi-parent recombination, where more than two individuals participate in generating a single offspring individual, clarifies that this generalization of recombination might yield a performance improvement in many application examples. Unlike evolution strategies, where it is either utilized for the creation of all members of the intermediate population (the default case) or not at all. The recombination operator in genetic algorithms is typically applied with a certain probability, p_c, and commonly proposed settings of the crossover probability are, $p_c = 0.6$ and $p_c \epsilon$ [0.75,0.95].

An example of two-point crossover is given in Figure 4.4. The recombination points marked by vertical lines are chosen randomly, and the offspring is created by exchanging the segment limited by two crossover points between the parents, thus creating two offspring. Typically, one of the offspring individuals is randomly selected to be chosen for the next generation while the other one is discarded.

4.2.2.3 Mutation

The mutation operator is introduced to simulate the errors that may arise during the copy process. This operator is designed to recover lost useful genes lost in the selection and crossover operation. Mutation serves to change each bit of a string slightly different from 0 to 1 or from 1 to 0 based on mutation probability, p_m. The mutation brings diversity to the population that hopefully leads to a better optimum solution. Usually the mutation probability, as found in [4] is given as, $p_m = 0{,}005 \ . \ . \ . \ 0{,}05$ where,

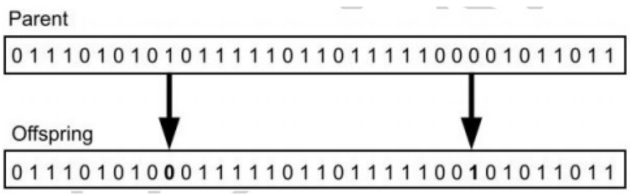


Figure 4.6: Mutation operator used in canonical genetic algorithms

The mutation operator is illustrated in Figure 4.5. In this example, two bits of the parent individual are inverted by mutation. In general, for $p_m = 1/l$, one bit on average is expected to mutate per individual, but in principle also multiple mutations are possible with exponentially decreasing probability. It should be noticed that this is an important property of the mutation operator, because changing multiple bits at the same time at least in principle facilitates the algorithms escape from local optima, even if the probability of this to happen might be vanishingly small.

Since the genetic algorithm is a stochastic search algorithm, it is possible that the best individual will not always survive. However this can be prevented by using elitism in which the best individual is automatically copied to the next generation. For example, the worst individual in the population can be replaced by the best one from the previous generation. In the genetic algorithm, the initial population is usually selected randomly. The usual optimization criterion is to wait for a certain number of generation to be full and then optimization can be terminated. Other possibilities are to observe the fitness function values or the diversity of population and if the fitness function value does not improve any-more or all individuals are too similar the optimization can be stopped. The flowchart of the basic GA is represented below Figure 4.7.

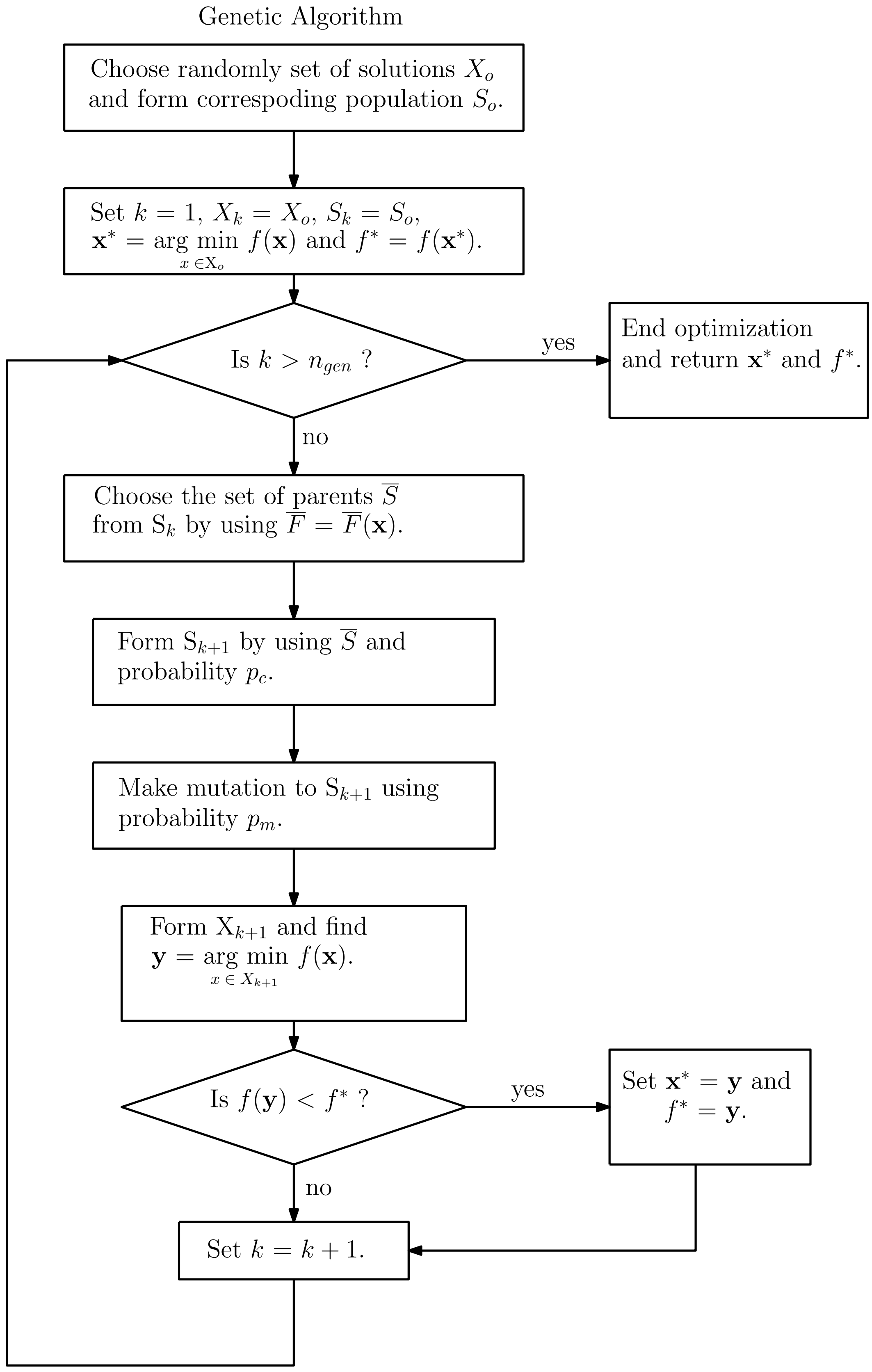


Figure 4.7: GA algorithm flow chart

4.3 Truss design problem formulation

Optimum design of structures includes finding optimum sections for members that minimizes the structural weight. The problem of minimum weight discrete design of truss structures under stress and displacement constraints can be expressed mathematically in the following form. Select cross sectional A_i of N truss elements $(i = 1, ..., N)$ where:

$$A_i \,\epsilon\, \{A_i^{min} = A_i^1; A^2; A_i^3; ...; A_i^{K_i} = A_i^{max}\}, \tag{4.14}$$

to minimize the weight of the structure,

$$W = \sum_{i=1}^{N} \rho_i \; A_i \; l_i \rightarrow min, \tag{4.15}$$

subject to constraints on element stresses,

$$|\sigma_i| \leq \sigma_{max}, \quad i = 1, ..., N, \tag{4.16}$$

and to constraints on nodes displacements

$$|u_j| \leq u_{max}, \quad j = 1, ..., M. \tag{4.17}$$

In this formulation, A_i is the cross sectional area, A_i^{min} and A_i^{max} are the lowest and the greatest available area respectively, l_i is the length and ρ_i is the material density of the i-th truss element. The superscript, K_i, stands for the number of available sections in the catalogue for the i-th bar. The stress in the i-th element σ_i is limited by the maximal stress value $\sigma_m ax$. The displacement of the j-th structural node u_j should not exceed the maximal displacement bound $u_m ax$. M stands for the number of structural nodes. The discrete design variables are cross sectional areas A_i of members, which have to be selected from finite sets of available sections. For technological reasons certain members of a structural zone can be joined into linking groups having identical bar characteristics. All possible variants of the resulting discrete structural optimization problem are hard to check explicitly in real applications. For N catalogues, each composed of K_i available elements, the number of all combinations equals to K_1 x K_2 x . . . x K_N.

4.4 General evolutionary algorithm approach

Evolutionary algorithms are stochastic search methods, inspired by natural evolution and genetics. According to Hollard [28], the evolutionary process occurring in nature is first mimicked as a GA and in recent years proved to be a practical search and optimization method showing certain significant advantages over conventional methods. A single individual of a population is affected by other individuals of the population for example, by food competition, predators, and mating, as well as by the environment for instance, by food supply and climate. The better an individual performs under these conditions the greater the chance to live for a longer while and generate off springs, who in turn inherit the disturbed parental genetic information. Over the course of evolution, this leads to a penetration of the population with the genetic information of individuals of above average fitness. The non deterministic nature of variation leads to a permanent production of novel genetic information and therefore to the creation of differing offspring. The following structure of a general evolutionary algorithm reflects on a high level of abstraction all essential components of standard implementations of evolutionary algorithms.

Algorithm 1

1. Begin
2. t = 0
3. Initialize P(t)
4. Evaluate P(t)
5. P'(t) = select 1 P'(t);
6. P' ' (t) : = variation (P' ' (t));
7. evaluate (P' ' (t));
8. P(t+1) = select 2 (P' ' (t) U Q);
9. t:=t+1;
10. od
11. end

The classical instances of evolutionary algorithms, namely genetic algorithms, evolutionary strategies, evolutionary programming and genetic programming, can all be described in the conceptual framework of the above pseudo code formulation. The following general features, however are common to all evolutionary algorithms and can therefore be seen as the defining properties of evolutionary computation.

4.4.1 Classical genetic algorithm

Referring to the general evolutionary algorithm outline as given in algorithm 1, the classical genetic algorithm is characterized by the following properties:

1. Individuals are represented as binary vectors of fixed length l, i.e., $\overrightarrow{x} \epsilon \{0,1\}^l$.
2. In case of the so called generational replacement, offspring and parent population sizes are identical $(\lambda = \mu), P(t+1) :=$ P”(t) (there is no environmental selection), and select 1 (mating selection) is the only selection operator.
3. A generation gap $\gamma < 1$ including the steady state case $\gamma = \ 1/\mu$ is sometimes used as an alternative to generational replacement.
4. Classical genetic algorithms do not use $\lambda > \mu$. The main emphasis is put on mating sections.
5. Crossover occurs in various instantiations and acts as main variation operator,while mutation is of secondary importance and acts as a background operator.

4.5 Chapter Summary

The complexity of structural steel and the need for efficiency in today's competitive world have forced engineers to have greater interest in the economical design of structures using heuristic evolutionary algorithms. The heuristic evolutionary algorithms provide efficient tools for performing structural optimum designs in problems considered highly complicated and requires the use of reliable multi-dimensional optimization method. With an increasing number of members, the terrain of the functional multi-dimensional space becomes very complex and the use of calculus-based methods proves to be ineffective.

The heuristic algorithms which are, GA, PSO, and BB-BC were then noted to be of good use in solving structural steel. The selection of heuristic optimization algorithms is based on the wide applicability and ability to optimize complex problems. BB-BC was considered as the most effective optimizing technique to solve complex structural steel problems because of its computational simplicity, ability to handle multi-dimensional problems and very fast convergence.

Chapter 5

Topology in truss optimization problem

Structural topology optimisation is the most general of the three categories yielding information on the shape, size and topology optmization. The topology of the truss is chosen from the group of selected topologies and the sizes of profiles from the given selection standard sections available in South Africa see Figure 5.1 below. Engineers in Aerospace, Civil/Structural and Automotive etc. use modernised topology optimization techniques to solve problems through the use of finite element method (FEM). The FEM method have been successfully utilised topology optimization in order to achieve weight savings in structures.The change in the dimensions of a truss affects its shape. The minimized objective function can be, for example, mass, cost or displacement or there can be conflicting criteria in a multi-criteria problem at the same time. Constraints take care of the demands of steel design rules (Euro-code 3). The truss can be plane or space truss and the problem can include several load cases. The following figure gives a profile of steel structures:

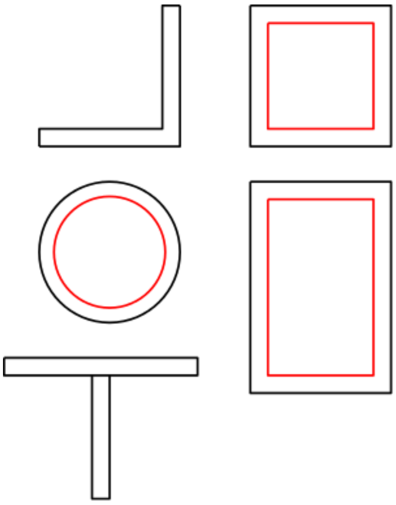


Figure 5.1: Standard profiles of steel structures

From the selected profile a truss can be built, with the best geometrical layout and optimization takes place using an algorithm aiming to get the best possible results. According to the study by Klarbring and Christensen [12], structural optimization are divided into three classes, size optimization, shape optimization and topology optimization. The Figures 5.2, 5.3 and 5.4 discusses further the three lasses of topology optimization. The size topology optimization problem is about finding the optimum cross sectional areas or thicknesses for structural members.

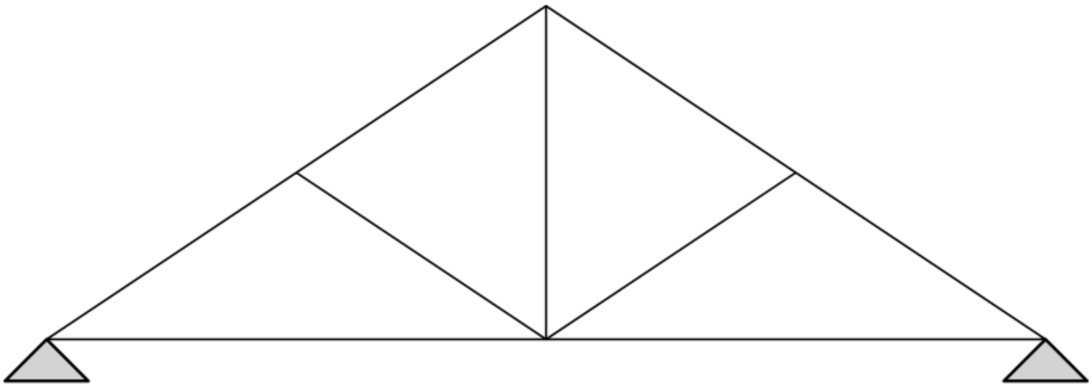


Figure 5.2: Sizing optimization of structural steel

The shape optimization is about the location of finite elements nodes that are set as design variables and the goal is to find the optimum location of the nodes. In other forms shape optimization is interpreted as finding the best function that describes the bounding shape of the structure.

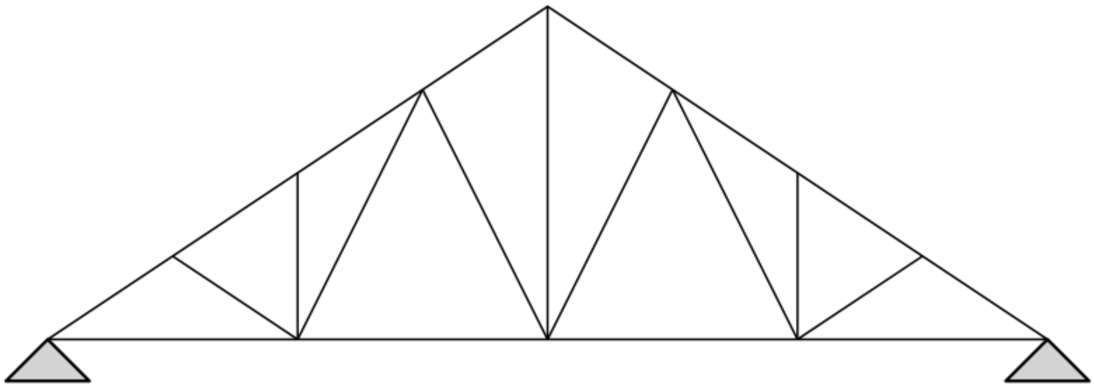


Figure 5.3: Shape optimization of structural steel

The topology optimization class is about finding the optimal connectivity between the structural members that comprise the structure. Ideally shape optimization is a subclass of topology optimization, and large number of nodes in the discretized structure could eliminate the need of shape optimization.

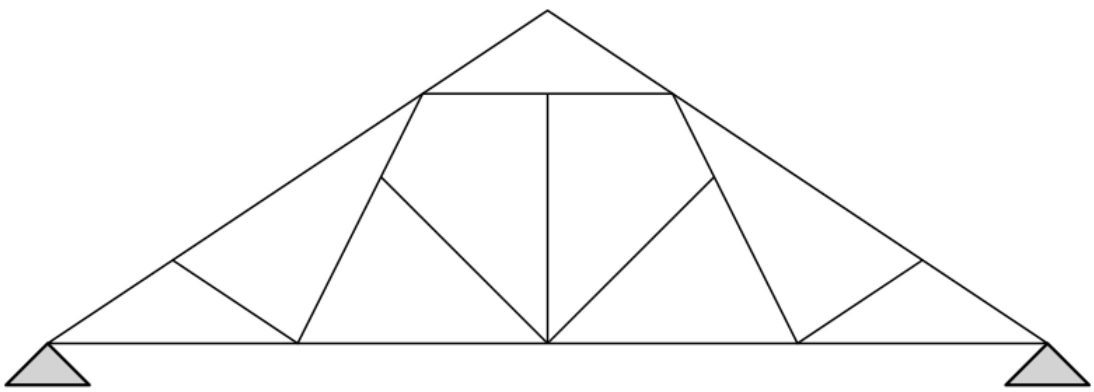


Figure 5.4: Topology optimization of structural steel

In the optimization problem the number of different members or groups of members is n_s in the truss Figure 5.2. A truss member means a web bar or a piece of chord between two joints. The number of allowed sections for each member or group of members is n_{sec_i}. It is assumed that the number and the size distribution of hollow sections can be arbitrary. The numbers of convertible truss dimensions is called n_{sh} and the available truss topologies is called n_t, which means that the shape and the topology of truss can change only limited depending on designers choices.

5.1 Topology optimization of two-dimensional (2D) trusses

The ground structure method is considered to be the simplest approach to topology optimization problems, where the necessary members are selected to form the truss of optimal topology from a highly connected ground structure with fixed nodal locations [40]. Sved and Ginos [58] consider solving the problem of ground structure as one of the challenges that may exist to find local optimal solutions. Since removing a member of the structure results in structures of lower weight and no stress constraints to be satisfied. On the other hand using too many members on the ground structure can lead to an increased computational time and increased optimization processes that can lead to non practical configurations. Here below are some of the listed studies in topology optimization of trusses using different approaches:

- Kirsch [40] introduced an algorithm to find an optimal topology. This algorithm can solve a sequence of linear programming problems by neglecting compatibility conditions between member strain and nodal displacement.
- Another algorithm having a global search that is based on the branch and bound algorithms is introduced by Ringertz [54]. This algorithm is useful to solve problems with multiple local optimal solutions.
- Nakamura and Ohsaki [49] came up with an optimum solution for topology optimization of trusses for specified fundamental frequency. In their optimization a design problem is formulated first with positive lower bounds for cross sectional areas. An optimal solution is considered as a function of a parameter defining the lower bounds. Then the parameter is decreased to zero in order to find the truss with optimal topology. It is quite challenging to apply Nakuma and Oshaki's method, however when applying their method to problems with stress constraints it utilises the conditions for global optimality.

Topology optimization is all about finding the optimal connectivity between the structural members that comprise the structure. Ideally shape optimization is a subclass of topology optimization, and large number of nodes in the discritized structure could eliminate the need of shape optimization.

The 6 nodes, 11 elements truss shown in Figure 5.5 below is an example of optimized 2D trusses. The goal of the example below is to show that the algorithm is capable of finding many different optimal solutions that enable the designer to compare them and select the best suitable one, since in some cases the constructability of optimal solution might be hard or expensive. Figure 5.5, 5.6, and 5.7 shows different optimal topologies.

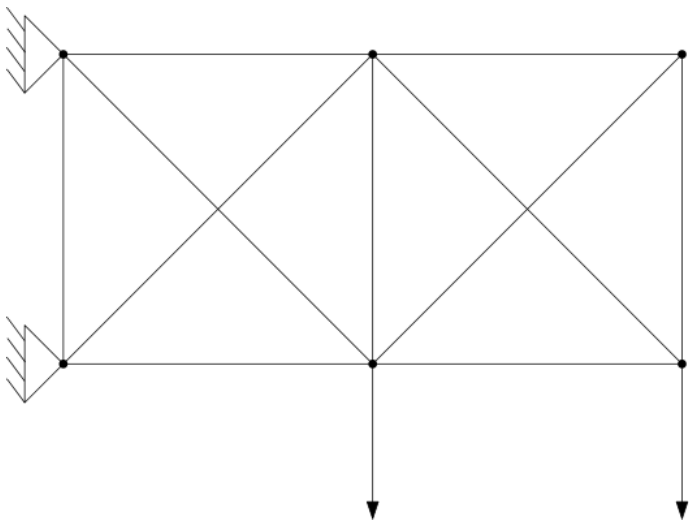


Figure 5.5: Initial truss design

The optimum topology is shown in Figure 5.7.

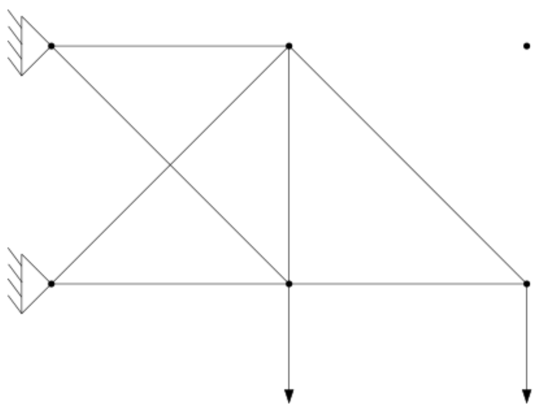


Figure 5.6: Topology optimized truss

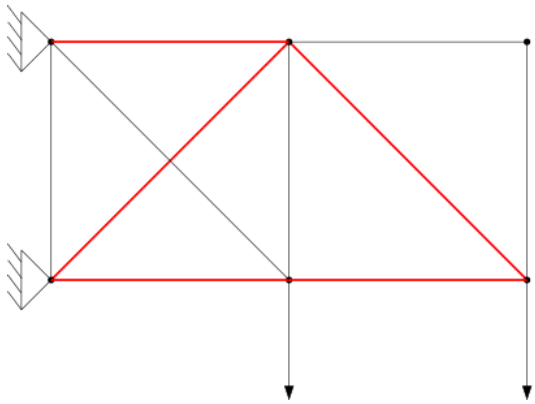


Figure 5.7: Truss cross section members optimized

5.2 Three dimensional optimization using topology

Structural optimization of discrete structures, namely trusses, grid shells and frames, is of great importance in structural engineering. A study by Gil and Andreu [25], shows that shape and sizing optimization have received much attention and are relatively mature areas of research. Researchers in the field of discrete topology optimization still face several challenges.The optimized new designs are expected to improve performance, meet new stringent weight targets and at the same time they need to be more economical to manufacture.

An engineer is required to first look at the best geometric layout for bars and nodes of the structure and secondly focus on the cross sectional area that needs to be evaluated. The topology of the structure is dependant on the structural shape designed by an engineer,and partly dependant on economical, construction techniques and environmental aspects. The Figure below shows a typical three dimensional bridge designed.

Figure 5.8: The first of forth bridge, built 1883-1890 [25]

Modern topology optimisation techniques can be applied to generalised problems through the use of the Finite Element (FE) methods, as a relatively recent innovation. Aerospace, automotive and mechanical engineers have successfully utilised topology optimisation in order to achieve weight savings in structures.

Optimal shape design of structural elements based on boundary variations results in final designs that are topologically equivalent to the initial choice of design and general, stable computational schemes for this approach often require some kind of re-meshing of the finite element approximation of the analysis problem.

5.2.1 Topology optimization

Topology optimization, also known as shape optimization refers to selecting the best elements in a given design space to maximize the use of material. Topology optimization is a branch of structural optimization. Bendsoe and Kikuchi [5], in their study discovered that research on numerical topology optimization started two decades ago and since then progress has been steadily converting it into a very mature discipline. The optimized structure evolves from one stage to another, meaning the final answer of a topology optimization run has a different shape than in the beginning. However, from a perspective of how to implement optimization, shape and topology are very different. Programs like Genesis are employed in the car manufacturing industry for shape optimization, also programs like Genesis corresponds to grid location optimization, while topology optimization corresponds to material property optimization. In topology optimization the idea is to design the material properties, so that the optimum solution values obtained are either 0,0 or nominal values. Elements with 0,0 material properties are discarded from the design, while elements with their nominal values are kept. Figure 5.9 below, shows an example of an optimized design of a car [43]. On the right, the red members are the key elements to be kept and the green elements are the elements discarded in the optimization.

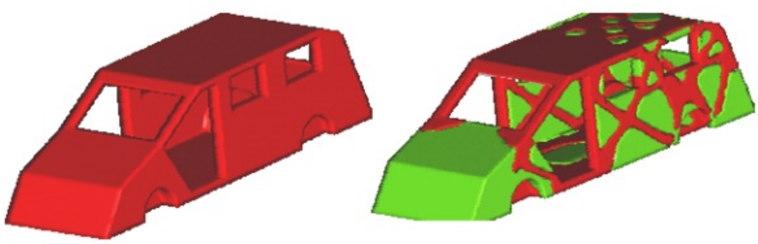


Figure 5.9: Topology optimization [43]

5.2.2 Topometry optimization

Topometry differs from topology by taking element by element sizing optimization methods into account. This allows the user to design the dimensions of each element individually, as opposed to traditional sizing, where elements are designed in groups. Genesis software also employs the topometry optimization for element by element optimization. Topometry optimization differs from sizing in that it has additional requirements. Some of these requirements are similar to topology optimization requirements and they are listed as follows:

- Topometry needs to include options for satisfying fabrication constraints, symmetry conditions and minimum member sizes.
- Another issue, which topometry has and sizing does not, is that topometry results might suffer from checker-boarding.

The design shown in Figure 5.10 below, on the left the initial thickness design of a car and on the right, the red members are the elements to be thickened to increase a desired natural frequency [43].

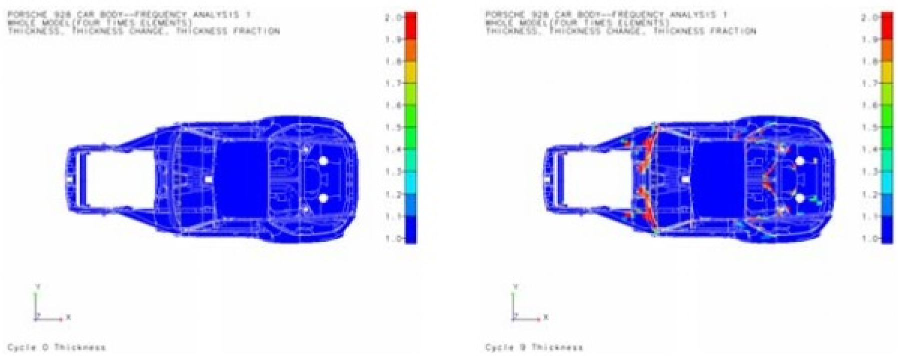


Figure 5.10: Topometry optimization [43]

5.3 Chapter Summary

The general form of the topology optimization problem is to determine the optimum distribution of material within a designable domain to fulfil a given objective. In this chapter, the topology impact was looked at and how it influences the behaviour of the structural steel under optimization and the impact of the results when the topology of the structure is looked at in detail.

Chapter 6

Numerical results and discussion using heuristic algorithms

6.1 The benchmark problem using BB-BC

In order to validate the proposed algorithm let us first consider a well known benchmark problem presented in [53]. For consistency with the author we stay here with the imperial units.

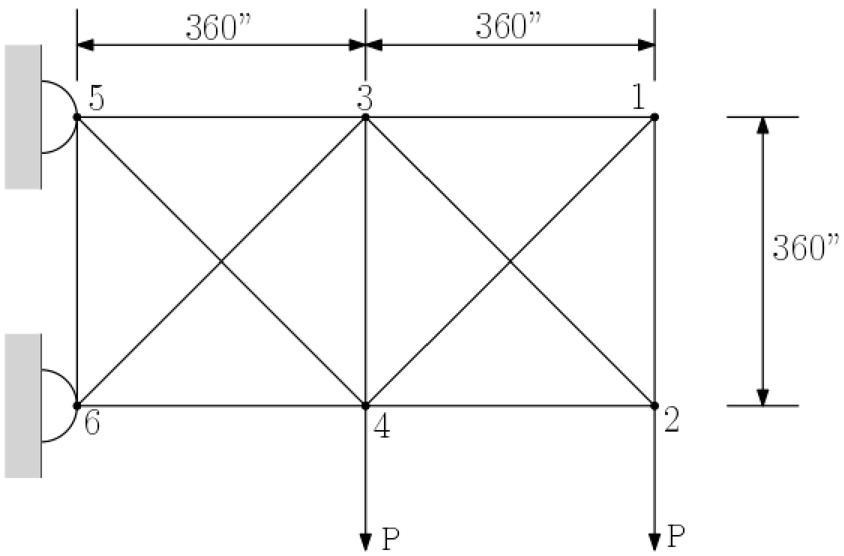


Figure 6.1: Ten bar cantilever truss [53]. P = 10^5 lb

The minimum weight design of this truss is subjected to a member stress limit $\sigma_{max} = \pm$ 25,000 lb/in^2 and the maximum allowable displacement $u_{max} = \pm 2$ in. The density of the steel in imperial units is $\rho = 0.1$ lb/in^3. The following catalogue of 24 different cross-sectional areas can be used for each of 10 truss members (all areas given in square inches):

Table 6.1 Selected cross-sectional areas for a 10 bar truss

0.1	1.0	2.0	3.0	4.0	5.0	6.0	7.0
8.0	9.0	10.0	12.0	14.0	16.0	18.0	20.0
22.0	24.0	26.0	28.0	30.0	32.0	34.0	36.0

The elaborated genetic algorithm was used by the author Pyrz [53], to find the best possible solution. The results obtained after 200 generations and the population number is 5111.29 pounds. The selected sections are as follows: 30.0, 0.1, 24.0, 14.0, 0.1, 0.1, 8.0, 22.0, 22.0 and 0.1. Our results following the same number of function evaluations is 5071.70 pounds. Two of the ten selected sections are different to those in the paper by Pyrz [53], namely, the third one is 22.0 and the sixth is 1.0 instead of 0.1. Also, it must be emphasized here that the GA algorithm used is much more complicated than the BB-BC.

The next graph demonstrates the convergence of the BB-BC algorithm using the contraction ratio $\mu = 1$;

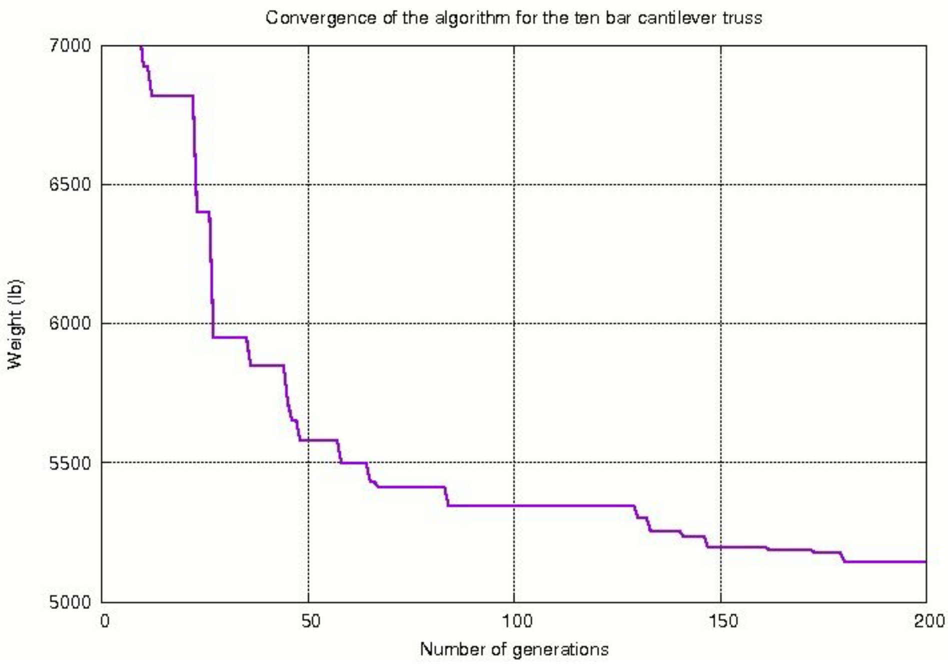


Figure 6.2: Convergence of the BB-BC algorithm for the ten bar cantilever truss ($24^{10} \approx 6.34$ x 10^{13} possible combinations.)

6.2 A comparison between GA and BB-BC

The present study is based on a comparison of two traditional heuristic approach methods BB-BC and GA. BB-BC is employed for the current optimization and compared with the work of Hojjat and Sanjay [1], using the GA. A detailed design is discussed by Hojjat and Sanjay [1], and Table 6.2 is used as a summary of data used for the BB-BC and GA. This benchmark problem consists of a small structure of nine (9) nodes and seventeen (17) element members. The design is limited by stress and displacement constraints. For BB-BC, all deflections in the structure are limited to 5.08 cm (2 in.) and the stresses are limited to 160 MPa, whereas in the GA the constraint used is the stress limited to 340 MPa.

The loading consists of a single downward load of 445 kN at the free end as shown in Figure 6.3. According to the research by Sivandandam [57], the rate of convergence in GA is determined by the magnitude of the selection pressure, high selection pressure results in higher convergence and execution time. Genetic Algorithms should be able to identify optimal or nearly optimal solutions under a wide range of selection scheme pressure. However, if the selection pressure is too low, the convergence rate will be slow, and the GA will take unnecessarily longer time to find the optimal solution. This research employs BB-BC to optimize a 17 bar truss, because of its faster convergence toward the optimal solution and ability to reproduce the same results consistently over many trials (see a comparison of Figure 6.4 and Figure 6.5). The results obtained are compared with the results as shown in Table 6.2. The comparison of GA and BB-BC convergence is summarised in Table 6.2.

Table 6.2 Summary of data comparison for GA and BB-BC optimization

Info	17 bar truss GA [1]	17 bar truss BB-BC
Number of elements	17	17
Number of nodes	9	9
Population size used	100	100
Range of cross sectional areas (cm^2)	0.645 - 129	12.24 - 97.59
Young's modulus of elasticity (GPa)	207	210
Material density (kg/m^3)	7215	7850
Number of design iterations	60	40
Allowable stress (MPa)	340	160
Allowable deflection (cm)	not listed	5.08
Minimum weight obtained (kN) [1]	11.54	14.39 new
Minimum weight obtained by [39] in (kN)	11.48	14.39 new

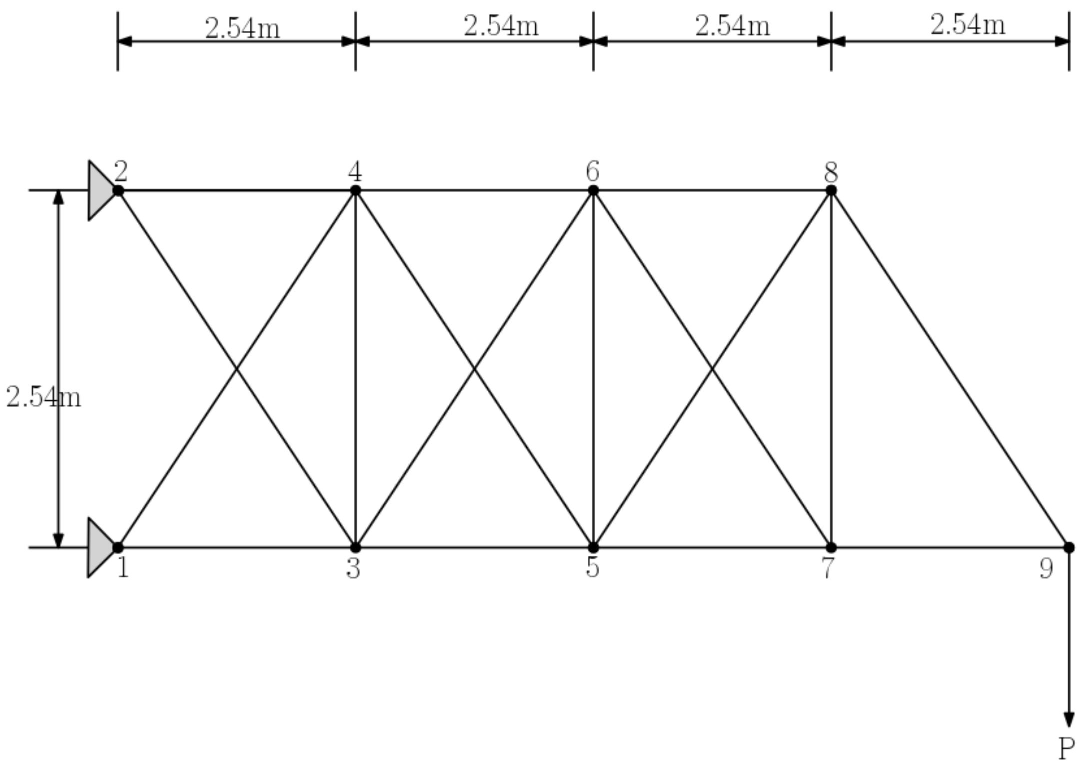


Figure 6.3: A 17 member truss. P = 445 kN [39]

The present study demonstrates how progress in modern evolutionary algorithms has revolutionized the design optimization of engineering structures. The use of an evolutionary algorithm called BB-BC algorithm is shown by the example of the steel trusses where the minimum possible weight is determined subjected to stress and displacement constraints.

6.2.1 GA Based structural optimization

The banchmark of GA presented in [1], is discussed in detail and later on compared with BB-BC that is also discussed in detail here below in section 6.2.2. In structural optimization, often the aim is to minimize the weight of the structure subjected to various loading under certain design constraints. The algorithm presented by Hojjat and Sanjay [1], in his paper is restricted to the axial load. For a structural consisting of N_E members classified into M groups, we need to find the cross-sectional areas $\mathbf{A} = \{A_1, A_2, A_3, ...A_M\}$, such that the total weight of the optimized structure transfer mechanism of distributed GA based on a penalty function method. Consequently, the total number of communications is the same in both cases. Considering that for small amounts of data transfer set-up time dominates the total communication overhead, the additional communication cost in the augmented Lagrangian

GA is insignificant. Further, the quality of searches with augmented Lagrangian approach is much improved, which more than compensates for slightly higher communication cost and resulting lower speed-up obtained.

A mutation probability of 0.005 is used to avoid a severe damage on the chromosomal structures of child individuals. In this benchmark problem the two-point crossover was applied and the number of bits used for encoding of cross-sectional areas is limited to 16. The summary include the following population parameters such as the size of population (q), the number of design iterations (s). The general data for this benchmark problem such as the number of nodes, number of elements stress constraints, deflection constraints and material properties are also listed in Table 6.2. For the solution of the resulting simultaneous linear equations the iterative preconditioned conjugate gradient (PCG) method was employed because of its low memory requirement, with a tolerance limit of 1.0 x 10^{-5}. This limit was found to be accurate for this optimization. However, for the purpose of this research only 40 iterations were taken into consideration from the study in [1], as shown in the graph in Figure 6.4. It indicates that when using augmented Lagrangian GA simulation it takes long and unpredicted amount of time to arrive in the optimum solution. It was noted that the stress constraint of 340 MPa used is very high and the cross sectional areas selected are ranging from 0.645 - 129 cm^2, this shows that the obtained results cannot be interpreted as a true solution in the real world environment.

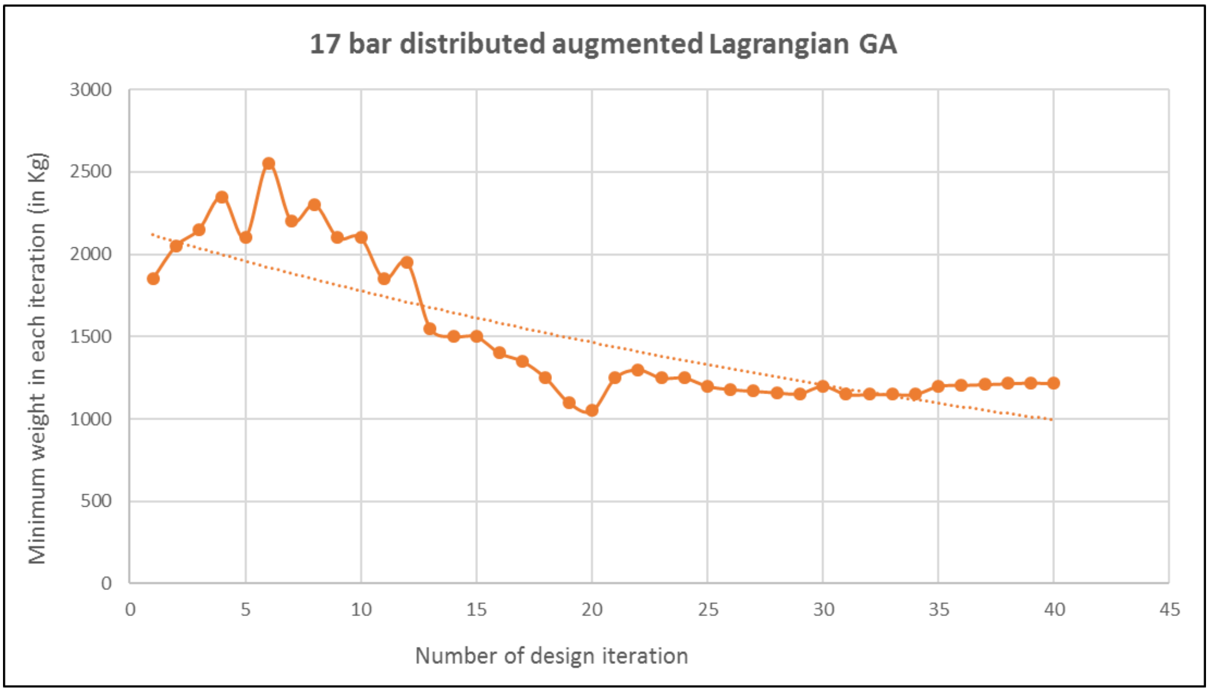


Figure 6.4: Variation of minimum weight for a 17 member truss using GA [1]

6.2.2 BB-BC Based structural optimization

The graph in Figure 6.5 demonstrates the convergence of the BB-BC algorithm. The contraction ratio used is $\mu = 1$. For this problem, European equal angles catalogue which consists of 25 sections is used and all areas are given in square centimetres. The selected material properties are based on the structural material used in the modern day industry with a material density of 7850 kg/m^3, Young's modulus of elasticity 210 GPa. Having a population of size of 100 the optimum solution is reached after 40 iteration as compared to GA in [1], the optimum solution is reached after 60 iterations. The difference is in the constraints used and the selected cross sectional areas. The design is limited both in tensile and compression stress of 160 MPa in BB-BC whereas in GA the limiting stress constraint is 340 MPa see the summary in Table 6.2 and the displacement constraint of 5.08 cm (2 in.), a safety factor of about 1.3 is reasonable used in the modern design of structural steel.

Figure 6.5 shows the variation of the weight associated with the highest fitness value in each iteration of the BB-BC. The use of BB-BC gave results that were proven to be much more progressive based on its fast convergence. It can be noted that with proper tuning the BB-BC optimization can reach the optimum, or near optimum, solution within a few seconds. It should be noted that the BB-BC optimizing algorithm seems to be much faster and considerably easier to use than the GA if the design parameters can be given in the form of coordinate numbers.

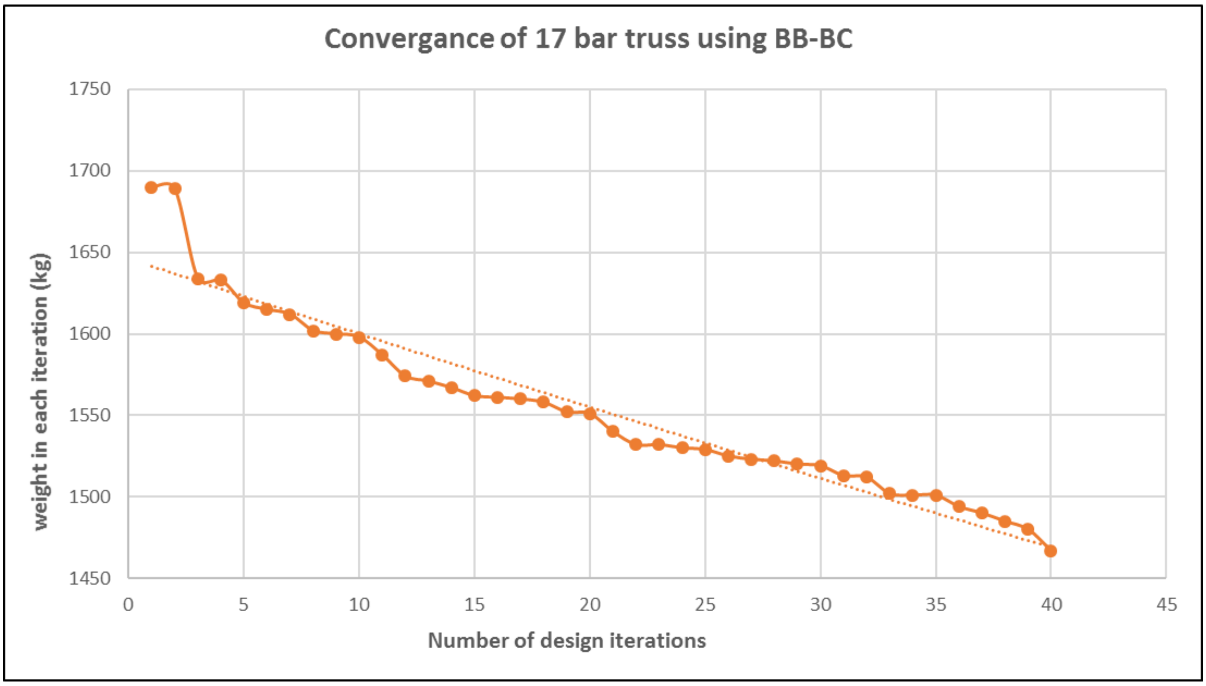


Figure 6.5: Variation of minimum weight for a 17 bar truss using BB-BC

Table 6.3 below, shows a comparison of selected cross sectional areas for structural members and the best results obtained. In 1984 an optimum weight of 11.48 kN was obtained by Khot and Berke [39], the same structure was used to compare the results obtained using augmented Lagrangian GA in 1995 by Hojjat and Sanjay [1], after 60 iterations an optimum weight of 11.54 was obtained as the best optimum solution. BB-BC was employed to optimize the same problem with the use of modern material, stress constraints and safety factors used in the structural industry an optimum weight of 14.39 kN was reached after 40 iterations.

Table 6.3 Cross-sectional areas of element members

Element number	Khot and Berke [39] (cm^2)	Lagrangian GA [1] (cm^2)	BB-BC New (cm^2)
1	102.77	103.41	97.59
2	0.65	0.69	21.18
3	77.80	78.60	97.59
4	0.65	0.71	12.24
5	51.99	54.30	68.35
6	35.87	36.87	22.71
7	76.97	73.10	72.74
8	0.65	0.68	12.24
9	51.25	47.1	45.46
10	0.65	0.74	25.37
11	26.16	26.1	40.31
12	0.65	0.65	12.24
13	36.45	36.2	40.31
14	25.80	26.10	29.27
15	35.85	33.24	21.18
16	0.65	0.69	19.15
17	35.99	34.10	25.37
	1148 kg	1154 kg	1467 kg

6.2.3 Comparison of results for GA and BB-BC

One of the most important things to look for when using a GA for structural optimization is violation of design constraints. In GA based structural optimization, there is no guarantee that the minimum weight found at each iteration is a possible solution. Hence it is very important to check the designs after every iteration made for constraint violation. For this benchmark problem the only active constraint is the displacement in node 9, Figure 6.3, which equals the upper imposed limit of 5.08 cm. Table 6.3 is used for the comparison of obtained mass in the paper by Khot and Berke [39], and current optimization using BB-BC.

The graph below in Figure 6.6, clearly shows a comparison between GA and BB-BC and how each algorithm arrives to to the optimum solution and the number of iterations taken.

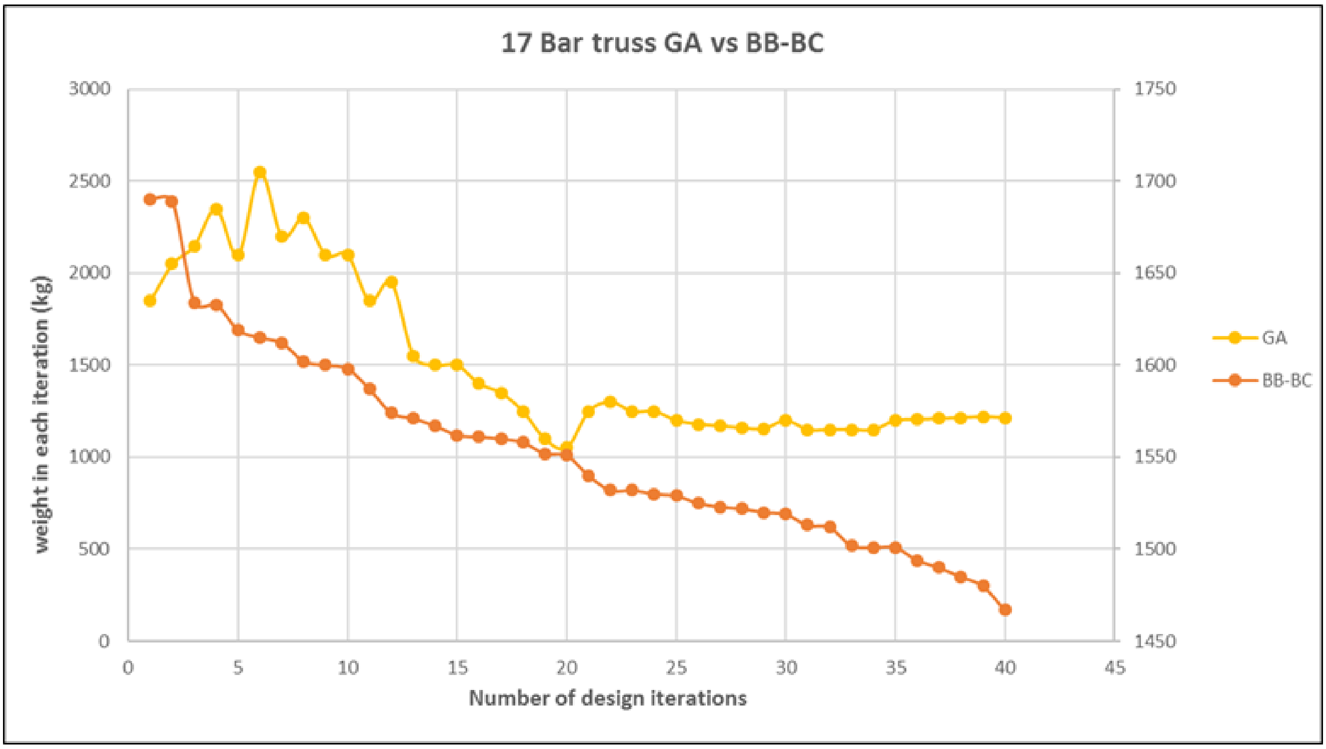


Figure 6.6: Results comparison GA vs BB-BC

The results in Figure 6.7 shows that GA suffers a lot from convergence speed and execution time. BB-BC owns a mechanism of group interactions that enhance the search for an optimal solution by a contraction factor between the best solution and the mass center. As shown in Figure 6.7, it can be concluded that as the number of iterations increase, the BB-BC moves towards the center mass, while on the other hand the GA takes time to move towards the center mass.

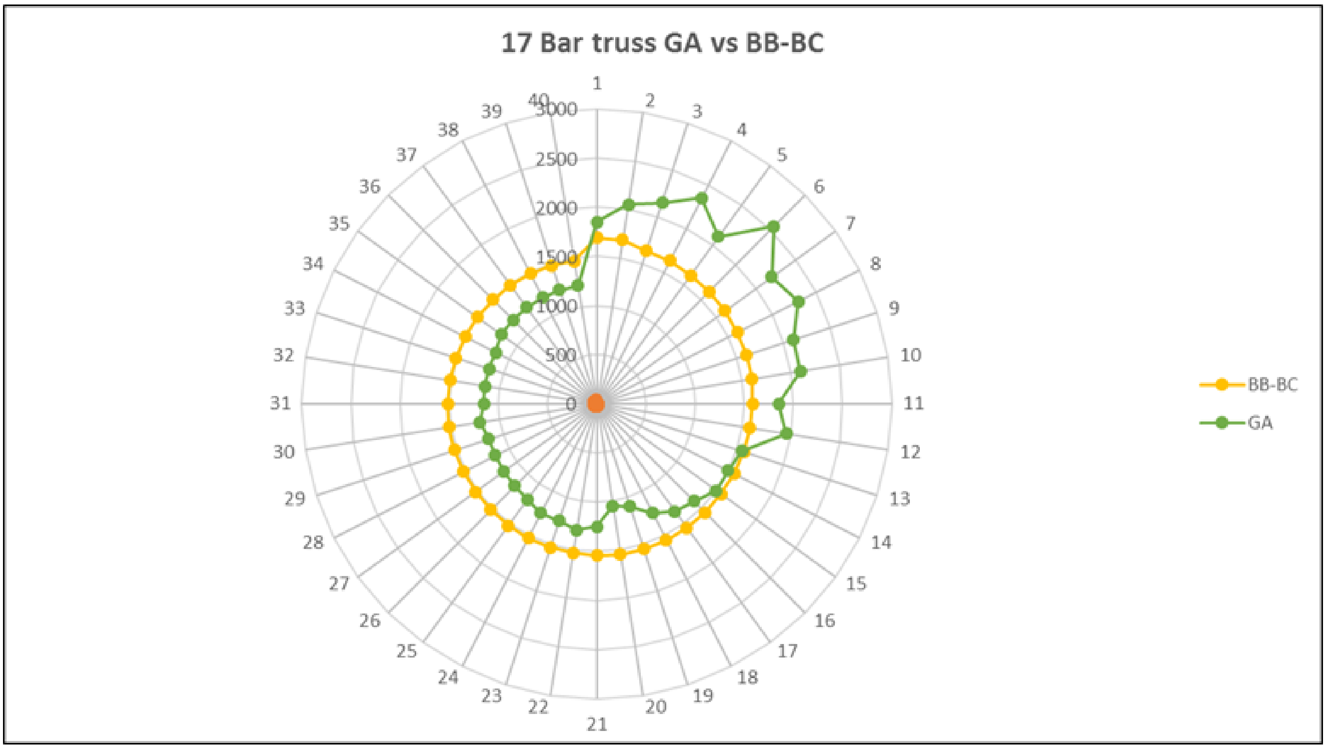


Figure 6.7: Results comparison GA vs BB-BC

6.3 Optimization of twenty nine bar trusses using BB-BC

The structural steel as shown in Figure 6.8, represents a possible solution for a large 29 bar steel structures. The catalogue and other parameters are the same as in problem in section 6.1. However, the contraction ratio μ was reduced to 0.1. As can be seen from Figure 6.8, the truss does not possess any symmetry and one of the members can be considered as a common one group. Thus, the total number of possible solutions in 29D feature space will be a tremendously huge number $25^{29} \approx 3.47 \text{ x } 10^{40}$.

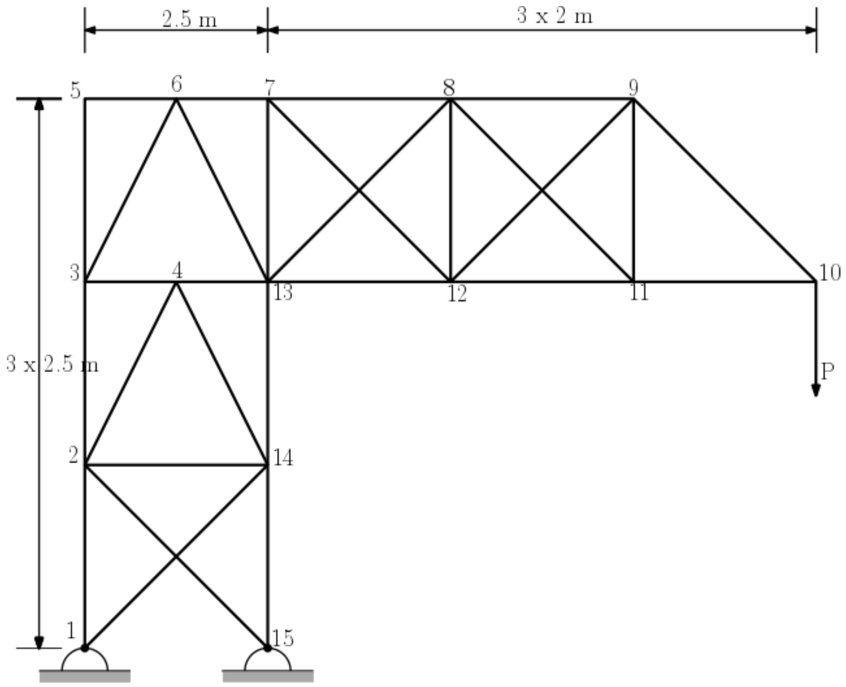


Figure 6.8: A large 29 bar truss. P = 300 kN

Despite the complexity of the problem the optimizing algorithm finds an optimum solution within a few seconds. Obviously, we do not know if the found solution is the best possible one. However, if not, it is definitely near the optimum solution. Two types of constraints are considered, first the fitness function is only subjected to the stress constraint $\sigma_{max} = 160$ MPa, and then the problem was solved within an additional displacement $\delta_{max} = 50$ mm constraint. Figure 6.9 clearly demonstrates the efficiency of the algorithm. The acceptable solution is obtained only after 100 iterations. Also, the performance is practically the same for both types of constraints contrary to findings in [53], where the genetic algorithms were used.

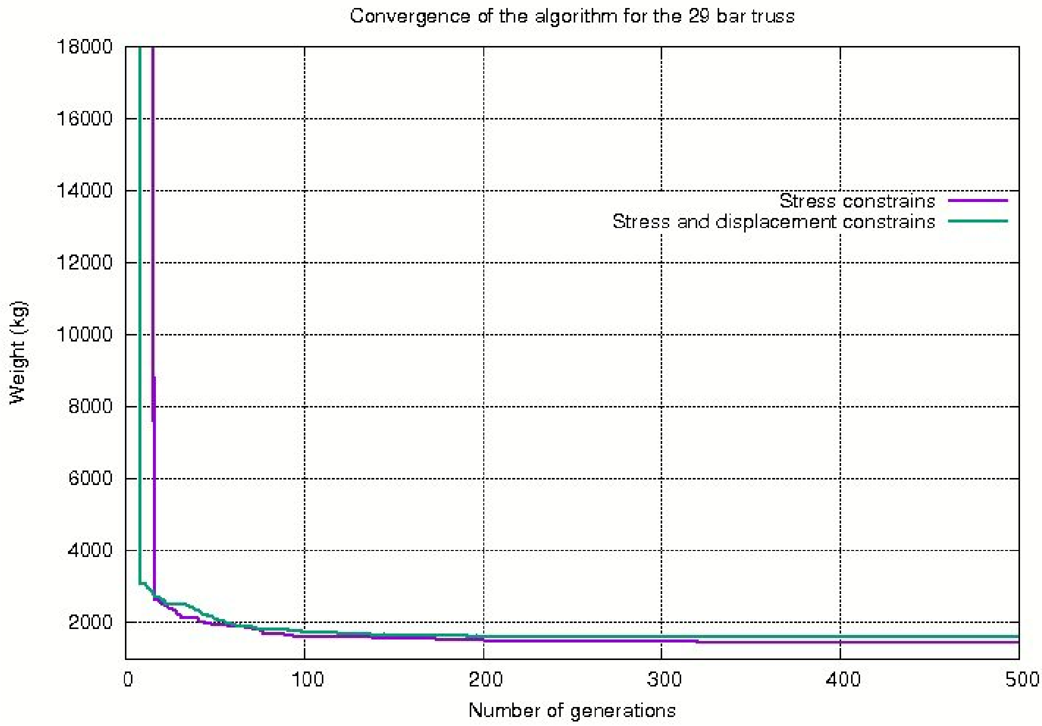


Figure 6.9: Convergence of the BB-BC algorithm for the 29 bar cantilever truss ($25^{29} \approx 3.47$ x 10^{40} possible combination).

The selected sections for a 29 bar trusses are presented in the table below:

Table 6.4 Selected cross-sectional areas

Truss Element	Stress Constrain Only		Stress and Displacement Constrain	
	Section No.	Section area (cm^2)	Section No.	Section area (cm^2)
1 - 2	11	45.46	7	58.66
1 - 14	25	12.24	25	12.24
2 - 15	25	12.24	25	12.24
2 - 14	25	12.24	25	12.24
2 - 3	11	45.46	8	52.10
2 - 4	25	12.24	25	12.24
3 -4	16	26.69	18	27.24
3 - 5	25	12.24	25	12.24
3 - 6	8	52.10	7	58.66
4 - 13	20	22.71	17	29.97
4 - 14	25	12.24	25	12.24
5 - 6	25	12.24	25	12.24
6 - 7	11	45.46	9	51.82
6 - 13	9	51.82	9	51.82
7 - 8	13	40.31	11	45.46
7 - 12	25	12.24	24	13.89
7 - 13	24	13.89	25	12.24
8 - 13	24	13.89	23	15.52
8 - 12	25	12.24	25	12.24
8 - 11	25	12.24	25	12.24
8 - 9	19	25.37	17	29.97
9 - 12	23	15.52	22	19.15
9 - 11	25	12.24	25	12.24
9 - 10	19	25.37	17	29.97
10 - 11	23	19.15	22	19.15
11 - 12	20	22.71	19	25.37
12 - 13	13	40.31	12	43.02
13 -14	5	65.14	2	83.51
14 - 15	5	65.14	2	83.51
	Weight: 1440.78 kg		Weight: 1607.61 kg	

The maximum stress obtained in the truss in the first problem is -158.8MPa (element 8 - 13). Since the displacements were not constraints, the maximum value determined is 57 mm. The obvious location of the maximum displacement is at the joint where the force is applied. In the case when both constraints were enforced, we get the maximum stress equal to - 155.34 MPa (element 6 - 13), and the maximum displacement is 50 mm. Such good results can be attributed to proper constraint handling by the proposed algorithm. The practical examples considered in this study are purely arbitrary and serve only here to demonstrate the efficiency of the optimizing algorithm. Any other trusses and other structures with discrete set structural components can be analysed in a similar way.

6.4 Optimization of fifty two bar complex truss using BB-BC

A large complex truss of 52 elements and 26 nodes is optimized using BB-BC algorithm. The same catalogue and other parameters used in the 29 bar truss is used on this problem. The contraction ratio used is 0.1 and the population size is 100. As can be seen from Figure 6.10 the truss does not posses any symmetry. The design is limited by both stress and displacement constraints. The following section areas were selected which consist of 52 sections (all areas given are in centimetres (cm)):

Table 6.5 Selected cross-sectional areas for a 52 bar truss

97.59	12.24	27.24	27.24	97.59
43.02	13.89	97.59	29.27	34.83
97.59	34.83	29.27	97.59	34.83
65.14	45.46	97.59	72.74	52.10
58.66	61.79	33.93	22.71	46.06
19.15	29.27	51.82	12.24	58.66
58.66	19.15	29.27	21.18	45.46
29.69	13.89	19.15	97.59	72.74
52.10	97.59	34.83	13.89	97.59
58.66	15.52	97.59	29.69	52.10
97.59	97.59			

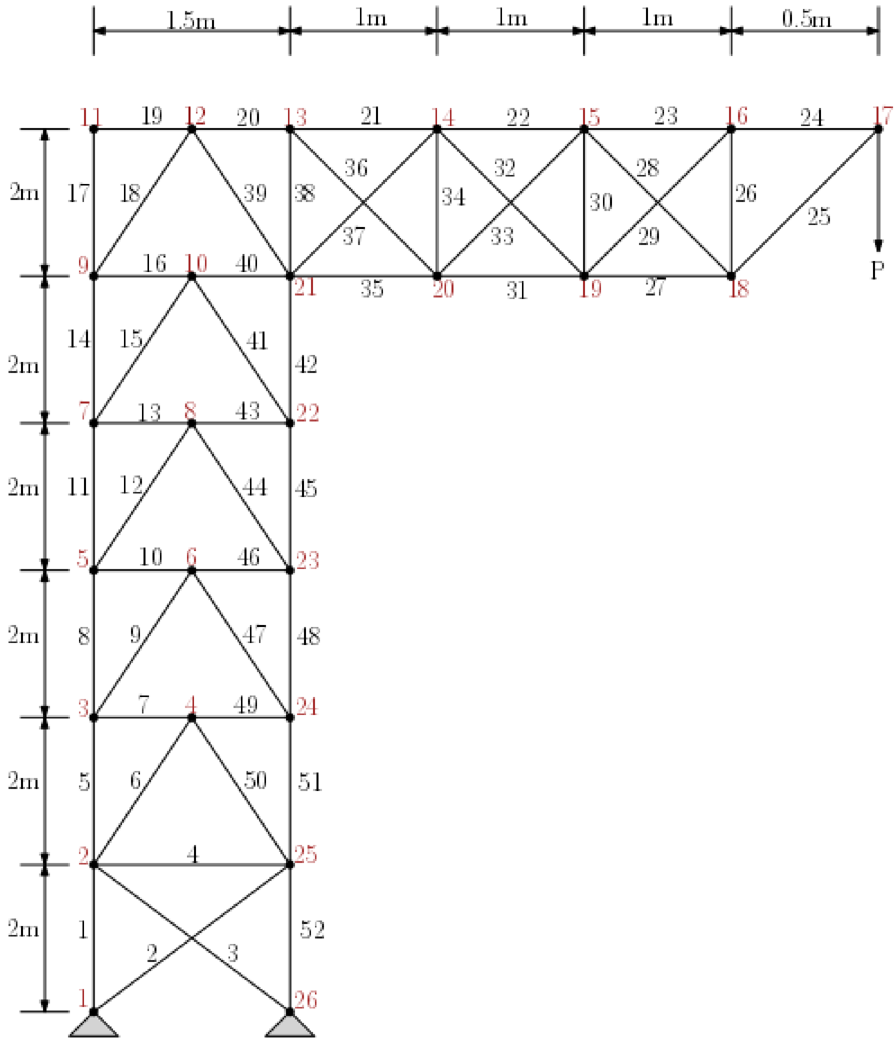


Figure 6.10: A large complex 52 bar truss. P = 400 kN

Optimizing large complex structures is challenging especially when using the GA. Despite the complexity of the problem the BB-BC is employed to find the optimum solution within a few seconds. The results obtained are considered as the best possible solution. However if not, they are definitely near the optimum solution. There are two types of constraints that are taken into consideration, first the fitness function is only subjected to the stress constraint of σ_{max} = 160 MPa, and the displacement constraint of δ_{max} = 5.08 cm. An important consideration in optimum structural design is violation of design constraints. A number of thirty (30) iterations were done and the best obtained optimal weight was found in iteration fifteen (15) with a mass of 3544.29 kg. The maximum stress obtained in the truss is -156.44 MPa (element 14 - 21) and the maximum displacement is 5.02 cm (element 19 -20). Based on the results obtained it can be concluded that the actual stresses and the displacement are within the allowable limits.

The graph below shows the variation of the weight associated with the stringent highest fitness value in each iteration or generation of the BB-BC. The optimal weight obtained after 30 iterations is 3544 kg.

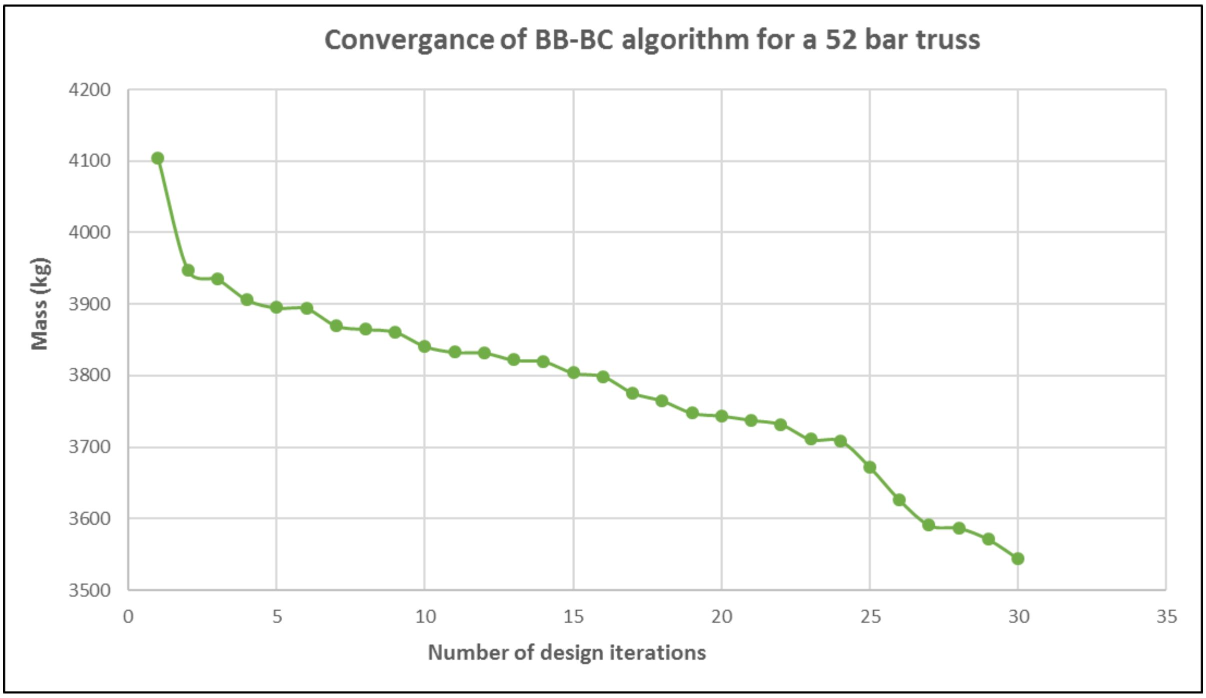


Figure 6.11: BB-BC algorithm for a 52 bar truss

Table 6.6 Elements

Truss Element	Section No.	Section Area (cm^2)
1 - 2	1	97.59
1 - 25	25	12.24
2 -26	18	27.24
2 - 25	18	27.24
2 - 3	1	97.59
2 - 4	12	43.02
3 - 4	24	13.89
3 - 5	1	97.59
3 - 6	17	29.27
5 - 6	14	34.83
5 -7	1	97.59
5 - 8	14	34.83
7 - 8	17	29.27
7 - 9	1	97.59
7 - 10	14	34.83
9 - 10	5	65.14
9 - 11	11	45.46
9 - 12	1	97.59
11 - 12	3	72.74
12 - 13	8	52.10
13 - 14	7	58.66
14 -15	6	61.79
15 -16	15	33.93
16 - 17	20	22.71
17 -18	10	46.06
16 - 18	22	19.15
18 -19	17	21.27
15 - 18	9	51.82
16 - 19	25	12.24

15 - 19	7	58.66
19 - 20	7	58.66
14 - 19	22	19.15
15 - 20	17	29.27
14 - 20	21	21.18
20 - 21	11	45.46
13 - 20	16	29.69
14 - 21	24	13.89
13 - 21	22	19.15
12 - 21	1	97.59
10 - 21	3	72.74
10 - 22	8	29.69
21 - 22	1	13.89
8 - 22	14	19.15
8 -23	24	97.59
22 - 23	1	72.74
6 - 23	7	58.66
6 - 24	23	15.52
23 -24	1	97.59
4 - 24	16	29.69
4 - 25	8	52.10
24 - 25	1	97.59
25 - 26	1	97.59
	Optimized weight	Weight: 3544.29 kg

6.5 Three bar simple truss using BB-BC

In order to demonstrate better how the BB-BC algorithm works we first consider a simple three bar truss, Figure 6.12. For this and the following problem we use a European equal angles catalogue which consist of 25 sections (all areas are given in square centimetres):

Table 6.7 Selected cross-sectional areas for a 3 bar truss

97.59	83.51	72.74	68.35	65.14	61.79
58.66	52.10	51.82	46.06	45.46	43.02
40.31	34.83	33.93	29.69	29.97	27.24
25.37	22.71	21.18	19.15	15.52	13.89
12.24					

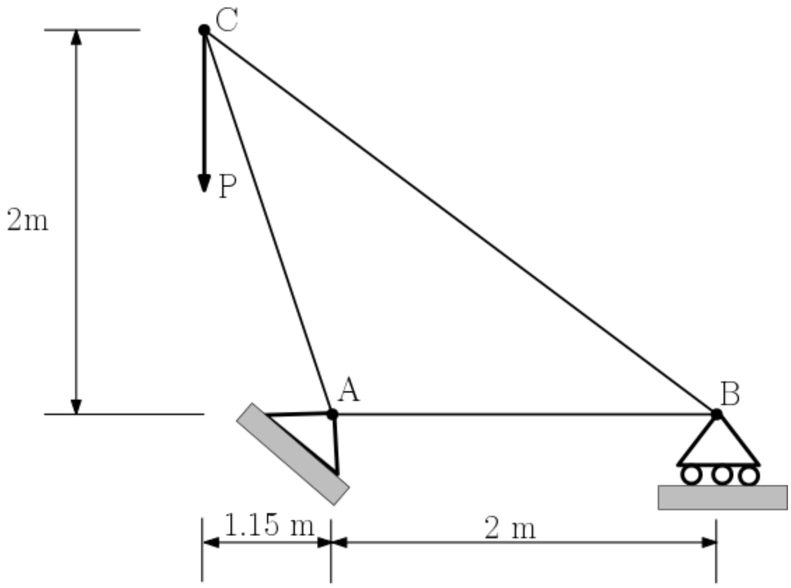


Figure 6.12: Three bar truss. P = 500 kN

The following design parameters are used in this problem and they are listed as follows: density $\rho = 7850$ kg/m^3, maximum stress $\sigma_{max} = 160$ MPa, maximum deflection $\delta_{max} = 50$ mm.

Having the population size of 100 the optimum solution is already reached in the seventh generation, and minimum weight obtained is 251.57 kg. The following sections have been selected:AB - 29.27, AC - 58.66 and CB - 33.93. Next three figures show how the population is contracted and the optimum solution obtained. The range, from 1 to 25, on the three axes corresponds to the ordinal numbers of the profiles in the catalogue and the search is conducted in continuous space.

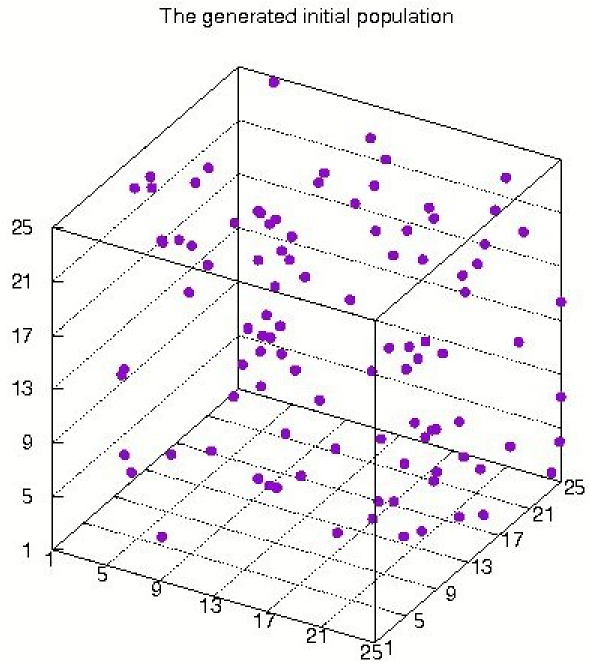


Figure 6.13: The generated initial population of 100 feature vectors.

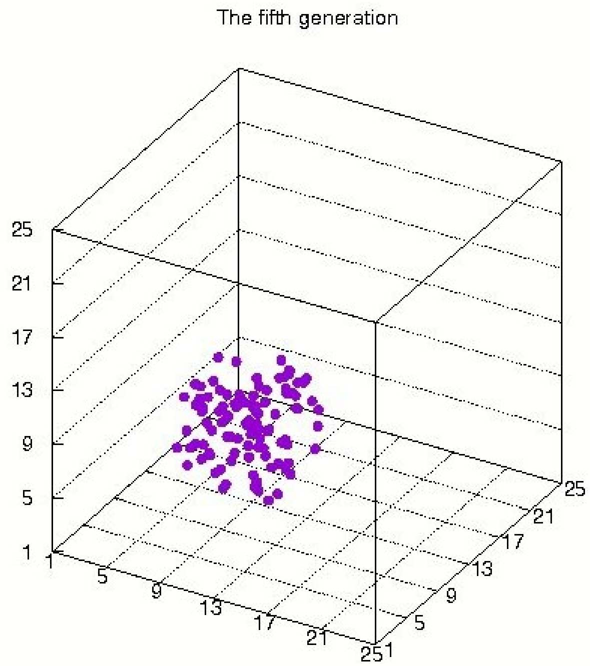


Figure 6.14: The feature space in the fifth generation

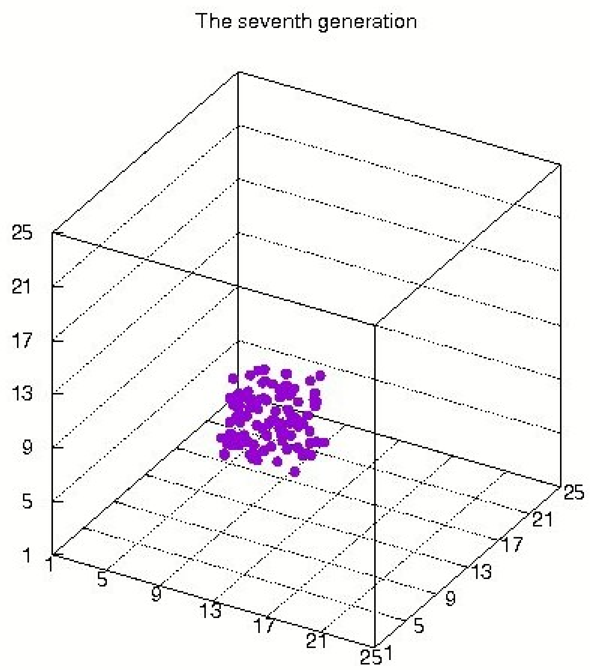


Figure 6.15: The final seventh generation. The optimal solution found at (17, 7, 15).

From the above listed figures it can be seen how fast the feature vectors move towards the optimum point.

6.6 Chapter Summary

A well known GA 10 bar cantilever truss problem was used as an academic banch mark problem which is taken from the literature. The BB-BC algorithm was employed to verify the GA obtained results by using tables and graphs to compare the performance of the two heuristic algorithms. A number of problems ranging from simple to complex structural steel are optimized using BB-BC. Upon comparison of the two heuristic algorithms, BB-BC was found as an algorithm of choice, due to its faster convergence and better obtained results.

Chapter 7

Structural optimization using Prokon FEA

7.1 Sizing, shape and topology optimization using Prokon finite element analysis

A model of the truss is built using Prokon analysis for steel structures in order to optimise the topology and the weight of the structure. The material selected to conduct the analysis is a common known structural material SANS 50025 GrS355 JR, with the following material properties listed as follows: ρ = 7850 kg/m^3, modulus of elasticity E = 210 GPa, material yield strength f_y = 355 MPa. The three basic approaches of structural optimization which are sizing, shape and topology optimization as discussed in chapter one will be looked at on the two problems and the results will be compared in order to get the best possible results using Prokon FEA. The material properties will remain the same.

7.1.1 Sizing, shape, and topology optimization problem 1

The truss is modelled with the length L = 20 m and the topology of the truss is kept fixed at a height of H = 1.5 m as shown in Figure 7.1 below. The truss is fixed at both ends and with the applied nodal load of P = 21 kN, the total load applied is 84 kN. The members selected are made of rectangular hollow cross sections with the horizontal selected to be 60 x 40 x 4.5 mm and diagonals 50 x 25 x 3.5 mm.

According to Rautaruuki [46], the gas shielded arc welding is the most common used welding technology in the manufacturing of tubular trusses in machine workshops. The design is to select the values of the size of the design, so that the cost of the truss is minimized and the strength requirements for all members are met and the strength for requirements for the welded joints are met.

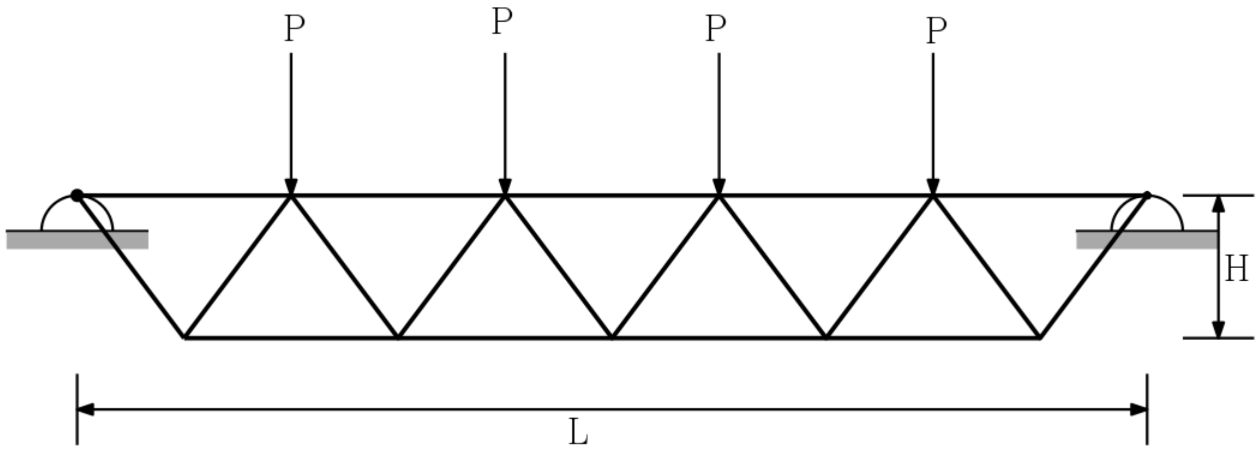


Figure 7.1: Structural analysis problem 1

The design model as shown in Figure 7.1 is completed using Prokon FEA, this software is capable of finding the optimum solutions within a few seconds. The method of trial and error is employed during optimization, with the two types of constraints considered are the maximum displacement $\delta_{max} = 50$ mm and the maximum stress constraint $\sigma_{max} = 319.5$ MPa. The topology and length of the trusses are altered during topology and weight optimization for better results.

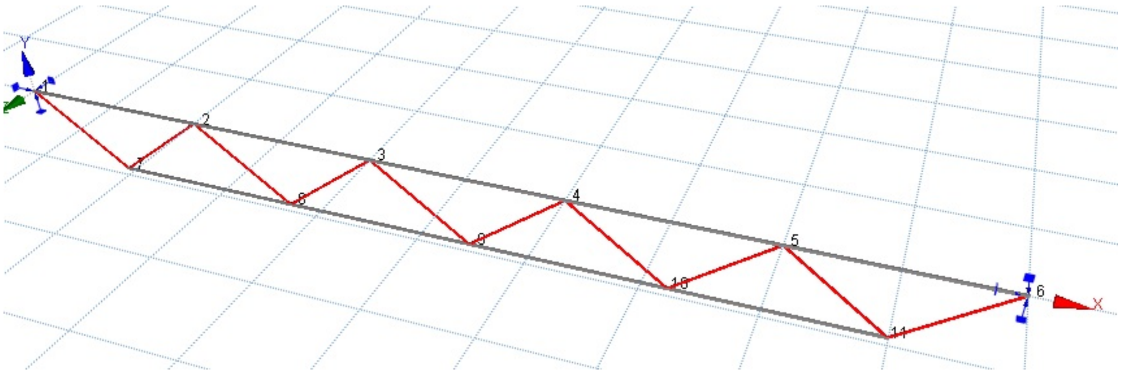


Figure 7.2: Prokon structural analysis problem 1

Nodal loads and constraints during a trial and error in order to get the optimum solution.

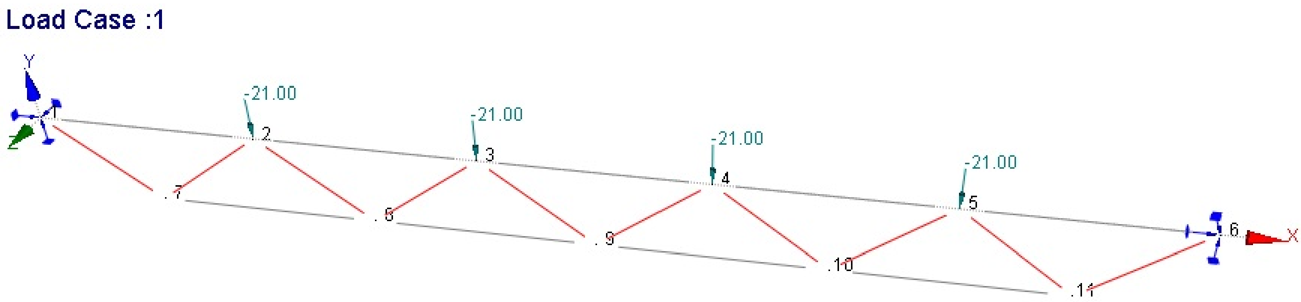


Figure 7.3: Prokon structural analysis nodal loading problem 1

7.1.2 Results

Despite the complexity of the problem Prokon FEA finds an optimum solution within a few seconds. Figure 7.4 shows the deflection of the structural members caused by the applied load, P. As shown below the actual deflection of $\delta_{actual} = 48.10$ mm is less than the maximum allowable deflection $\delta_{max} = 50$ mm. The stress results, in Appendix C problem 1, show that the actual stress is $\sigma_{actual} = 268$ MPa found in members 6-11 and 1-7, and these stress values are less than the maximum allowable stress $\sigma_{max} = 319.5$ MPa.

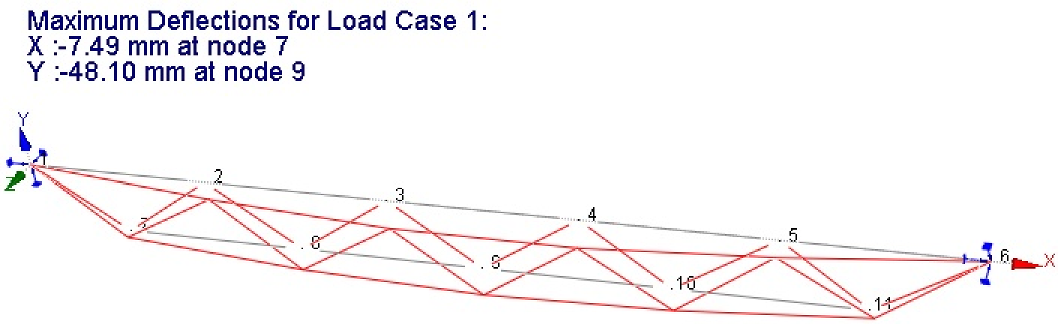
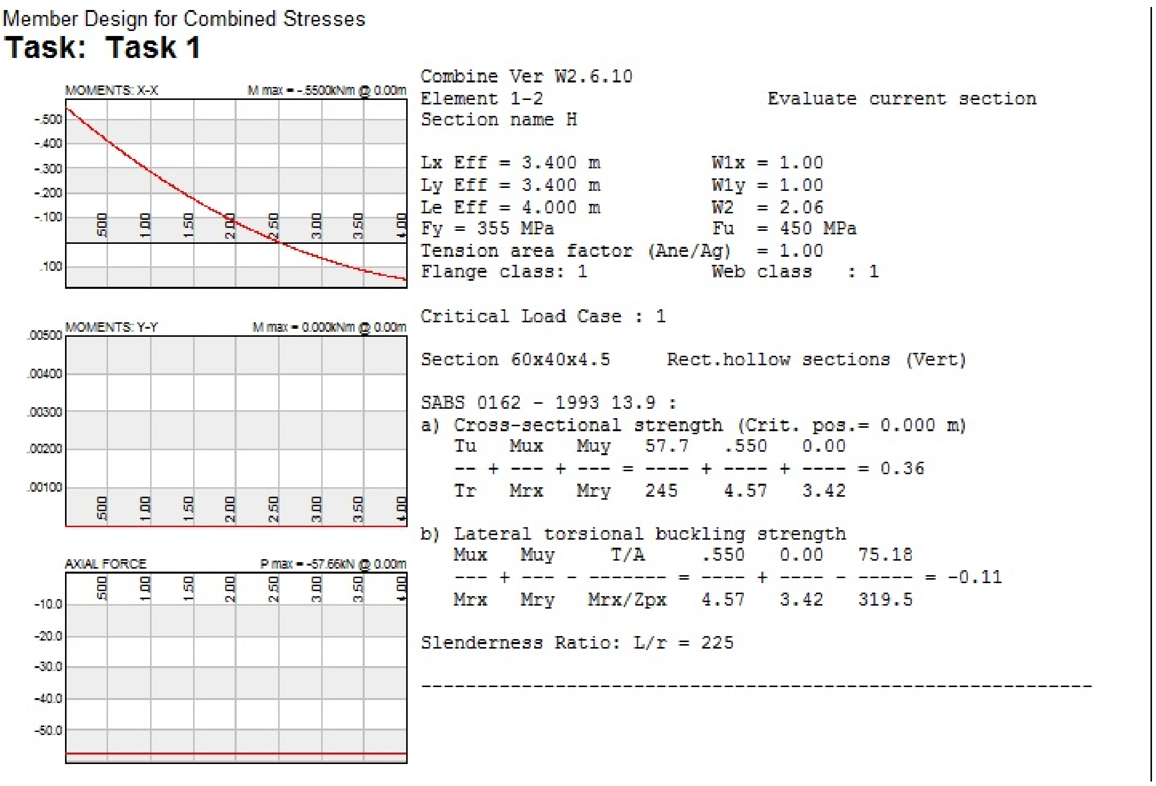


Figure 7.4: Prokon structural analysis deflection problem1

Figure 7.5 shows a summary of a detailed combined stress for each element. For full detailed calculations for each element in combined stresses see Appendix A. The comparison is done using stress ratios and bending moment ratios, when these ratios are combined they must give a value that is less than 1 as shown in the summarised calculation in Figure 7.5. The maximum calculated bending moment in element 1-2 as shown in the graph of moment x-x is -5500 kNm.




```
Combine Ver W2.6.10
Element 1-2                     Evaluate current section
Section name H

Lx Eff = 3.400 m            W1x = 1.00
Ly Eff = 3.400 m            W1y = 1.00
Le Eff = 4.000 m            W2  = 2.06
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu   Mux   Muy   57.7   .550   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.36                  OK
   Tr   Mrx   Mry   245    4.57   3.42

b) Lateral torsional buckling strength
   Mux   Muy      T/A      .550   0.00   75.18
   --- + --- - ------- = ---- + ---- - ----- = -0.11           OK
   Mrx   Mry   Mrx/Zpx   4.57   3.42   319.5

Slenderness Ratio: L/r = 225                                   OK

------------------------------------------------------------
```

Figure 7.5: Prokon member design for combined stresses problem 1 (Appendix A)

The total optimized weight of the structure is 310.3 kg. This mass will be compared with the one that will be obtained in problem 2, to check when the topology of the structure if altered, and how does the truss behave with the same applied load.

7.1.3 Sizing, shape and topology optimization problem 2

The truss is modelled with length, L = 20 m, and the topology of the truss is kept fixed at a height, H = 1.5 m as shown in Figure 7.6 below. The truss is fixed at both ends and with the applied increased nodal load of P = 45 kN. The total load applied is 180 kN. The members selected are made of rectangular hollow cross sections with the horizontal element selected to be 60 x 40 x 4.5 mm,vertical element 50 x 25 x 3.5 mm and diagonal element 50 x 25 x 3.5 mm.

According to Rautaruuki [46], the gas shielded arc welding is the most common used welding technology in the manufacturing of tubular trusses in machine workshops. The design is to select the values of the size of the design, so that the cost of the truss is minimized and the strength requirements for all members and the strength for requirements for the welded joints are met.

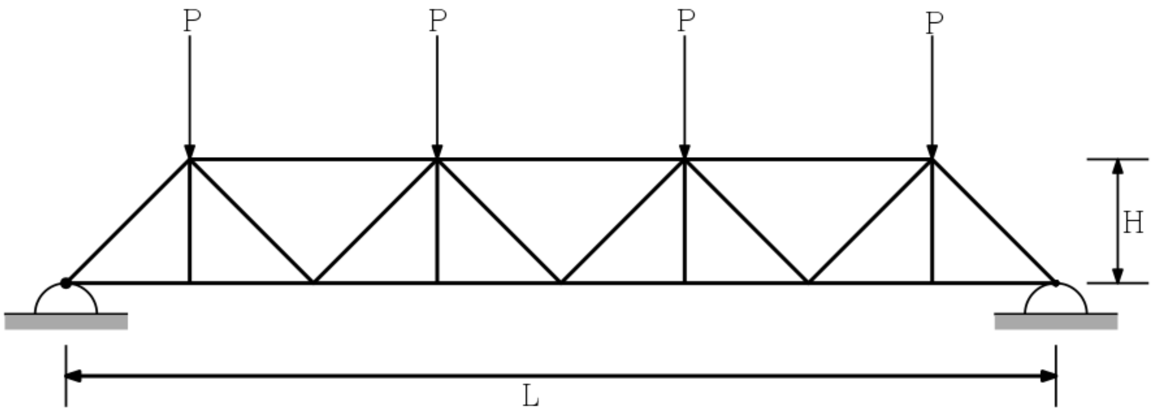


Figure 7.6: Structural analysis problem 2

The design model as shown in Figure 7.7 is completed using Prokon FEA, this software is capable of finding the optimum solution within few seconds. The method of trial and error is employed during optimization, with the two types of constraints considered which is maximum displacement δ_{max} = 50 mm and the maximum stress constraint σ_{max} = 319.5 MPa. The maximum load applied during topology and weight optimization is guided by the maximum deflection and maximum stress constraints in order to get best optimum results.

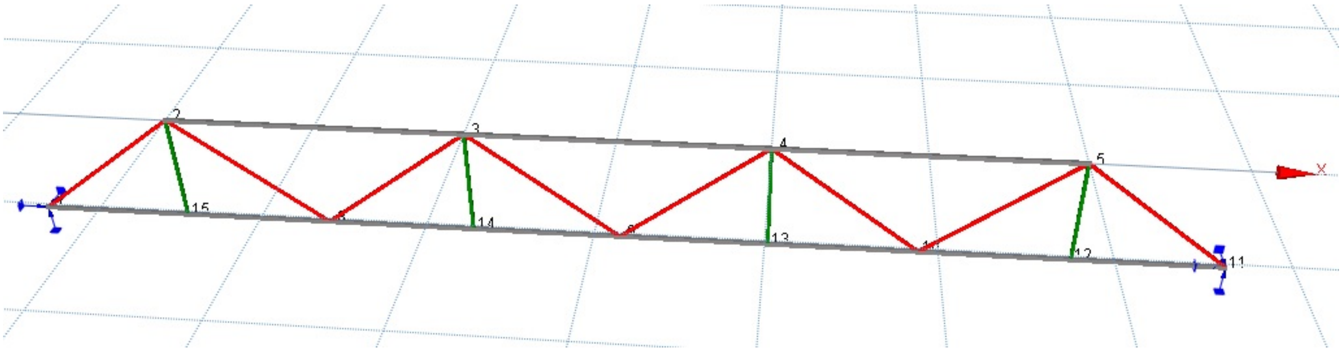


Figure 7.7: Prokon structural analysis problem 2

Nodal loads and constraints during a trial and error in order to get the optimum solution.

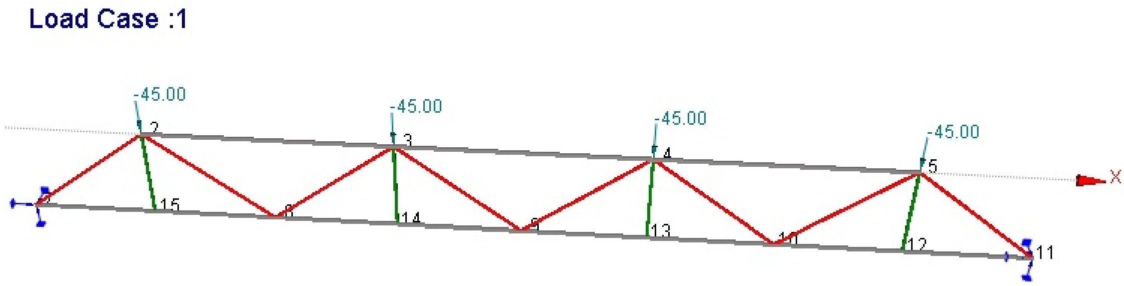


Figure 7.8: Prokon structural analysis nodal loading problem 2

7.1.4 Results

Despite the complexity of the problem, Prokon FEA, finds an optimum solution within a few seconds. Figure 7.9 shows the deflection of the structural members caused by the applied load P. As shown below the actual deflection of $\delta_{actual} = 47.92$ mm is less than the maximum allowable deflection $\delta_{max} = 50$ mm. The stress results as shown in Appendix C problem 2, shows that the actual stress is $\sigma_{actual} = 259.7$ MPa found in member 9-14 and 9-13, and these stress values are less than the maximum allowable stress $\sigma_{max} = 319.5$ MPa.

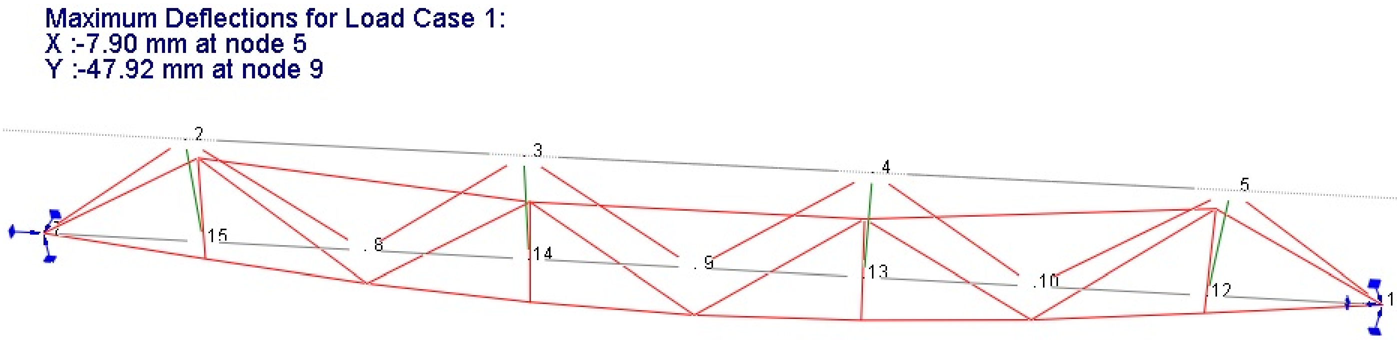


Figure 7.9: Prokon structural analysis deflection problem 2

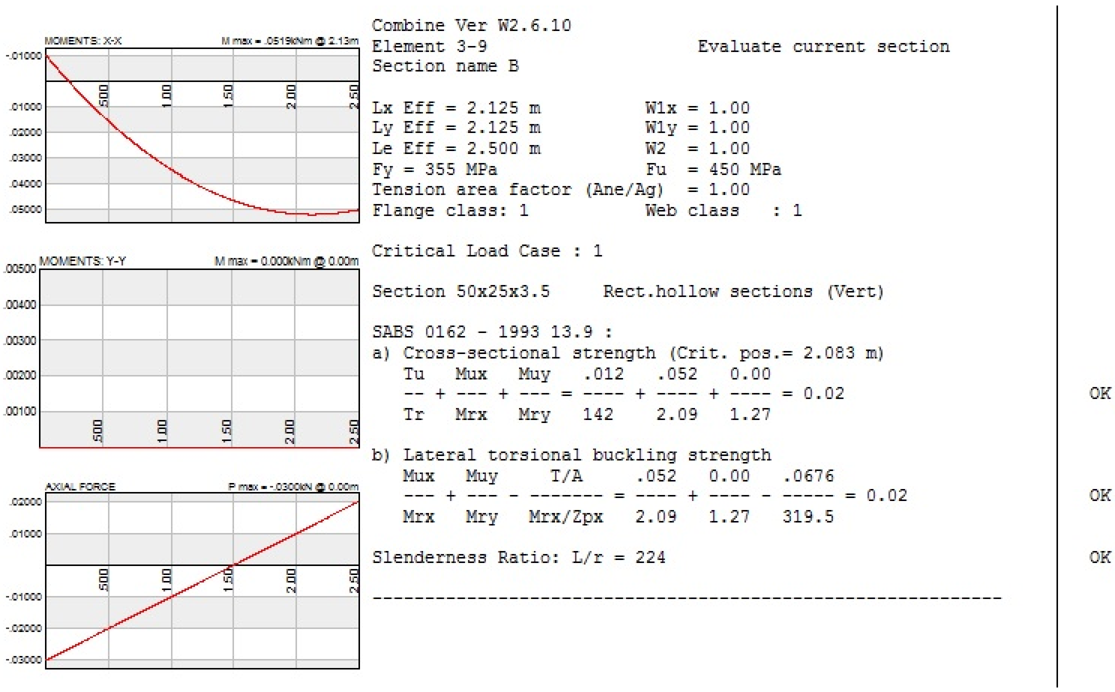


Figure 7.10: Prokon member design for combined stresses problem 2 (Appendix A)

The total optimized weight of the structure obtained in problem 2 is 457.9 kg. This mass will be compared with the one obtained in problem number 1 section 7.1.2, to check when the topology of the structure is altered how does the truss behave with the same applied load.

The stress comparison of problem 1 in section 7.1.2 and problem 2 in section 7.1.4 shows that the change in topology of the truss helps to reduce the stresses induced in the members. As it can be seen that the actual stress is $\sigma_{actual} = 268$ MPa found in members 6-11 and 1-7 in problem 1 is greater than the actual stress found in problem 2 $\sigma_{actual} = 259.7$ MPa found in member 9-14 and 9-13.

7.2 Chapter Summary

One of the most commonly used finite element analysis software packages for optimizing or designing steel structures in the structural design industry is Prokon. This computer software was used to verify the obtained results using the BB-BC. Few models of trusses were built using Prokon in order to optimise the topology and the weight of the structures. The common structural materials were selected to conduct the analysis. The three basic approaches of structural optimization which are sizing, shape and topology optimization as discussed in chapter one were employed. Prokon is a reliable FEA tool that is used by structural engineers to design different sizes of steel structures.

Chapter 8

Conclusion and recommendations

The first goal of the study was to formulate an optimization problem for trusses in such a way that this is not just theoretical but also useful in real life applications. This means that the topology, shape and sizing optimization of the plane or space trusses is considered, the manufacturing cost is taken into consideration, design constraints are based on the structural steel design rules and the selection of material and profiles is not limited to a few sizes.

The second goal was to solve the formulated problem efficiently with the use of three heuristic optimization algorithms that are population based particle swarm optimization, genetic algorithm and big bang-big crunch. The efficiency of these algorithms has been studied and compared in several examples. The mass has been optimized under the stress and deflection constraint.

Numerical examples show that heuristic algorithms are usable tools in optimizing structural steel problems. Discrete design variables and the demand of steel design rules which lead to rather awkward constraints are not obstacles for these algorithms. On the other hand heuristic algorithms suffer from high computational cost and uncertainty because they demand thousands of finite element analysis (FEA) per run and there has to be several of these runs in the case of stochastic algorithms. In addition, heuristic algorithms do not always work. They need proper tuning and offer no guarantee concerning the quality of the final results.

The BB-BC was noted to produce the best possible results for a ten bar truss, Figure 6.2. The BB-BC improves rapidly the value of the objective function during early iteration rounds when compared with the GA in section 6.2. The implementation of BB-BC can be considered the simplest when compared with the GA and PSO algorithms and it can be used to solve simple complex problems which produce good results in a short space of time.

The present study demonstrates how progress in modern evolutionary algorithms has revolutionized the design optimization of engineering structures. The performance of an evolutionary algorithm called the BB-BC algorithm is shown by example of the steel trusses where the minimum possible weight was determined subjected to stress and displacement constraints. It can be seen that with proper tuning the BB-BC optimization can reach the optimum, or near optimum, solution within a few seconds even in a highly complex twenty nine (29) and fifty two (52) dimensional search space. It is obvious that similar results can be achieved for various other types of problems like an optimum fibre orientation in laminated composite structures, shape optimization, analysis of manufacturing tolerances and many others. It should be also noted that the BB-BC optimizing algorithm seem to be much faster and considerably easier to use than the GA if the design parameters can be given in the form of the coordinate numbers.

In conclusion it can be said that it is worthwhile to include the topology and the size design variables when optimizing problems in steel structures. The optimization problem can be solved using GA, PSO or BB-BC. BB-BC is a better choice because it is easy to use and has a faster convergence in producing the best possible results. The designer has to make a number of test runs and a lot of time is consumed when using GA. In all problems, convergence to the right optimum solution, was obtained without problems for BB-BC. BB-BC proved its ability to reproduce the same results consistently over many trials as compared to PSO and GA algorithms.

For future studies complex three dimensional problems in steel structures will be optimized using heuristic methods that are well tuned to produce best possible results that can be interpreted in line with the real world. Also future studies include truss optimization problems which need to be improved, for example, taking the eccentricities of joints and the steel grades as new design variables. The selection of criteria could be increased and new constraints added due to fatigue loading and fire safety. Comparison of the mutual efficiency should be continued and also other heuristic algorithms like tabu search (TS), simulated annealing (SA) etc. should be considered. The comparison can be developed by giving an exact definition for the good performance of a heuristic algorithm which should also pay attention to parallel computing. In multi-criteria problems the use of the special versions of heuristic algorithms should be studied instead of the constraint method. Constraint methods are after all better suitable for cases where the designer's preference of criteria is already somehow fixed.

Bibliography

[1] Hojjat Adeli and Sanjay Kumar. Distributed genetic algorithm for structural optimization. In *Journal of Aerospace Engineering*, volume 8. American Society of Civil Engineers, July pages 124–140, 1995.

[2] Peter J Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *International Conference on Evolutionary Programming*, pages 601–610. Springer, 1998.

[3] JS Arora, MW Huang, and CC Hsieh. Methods for optimization of nonlinear problems with discrete variables: a review. *Structural and Multidisciplinary Optimization*, 8(2):69–85, 1994.

[4] Thomas Back and H-P Schwefel. Evolutionary computation: An overview. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 20–29. IEEE, 1996.

[5] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, 71(2):197–224, 1988.

[6] P. P Benham and R. J Crawford. *Mechanics of Engineering Materials*. John Wiley & Sons, Inc., John Wiley & Sons, pages 466–487, 1987.

[7] E Bonabeau, M Dorigo, and G Theraulaz. Swarm intelligence (p. 307), Oxford University Press US, 1999.

[8] Charles Camp, Shahram Pezeshk, and Guozhong Cao. Optimized design of two-dimensional structures using a genetic algorithm. *Journal of structural engineering*, 124(5):551–559, 1998.

[9] Charles V Camp. Design of space trusses using big bang–big crunch optimization. *Journal of Structural Engineering*, 133(7):999–1008, 2007.

[10] Serdar Carbas and Oguzhan Hasancebi. Optimum design of steel space frames via bat inspired algorithm. In *10th World Congress on Structural and Multidisciplinary Optimization*, pages 1–10, 2013.

[11] Chun-Man Chan and Kin-Ming Wong. Structural topology and element sizing design optimisation of tall steel frameworks using a hybrid oc–ga method. *Structural and Multidisciplinary Optimization*, 35(5):473–488, 2008.

[12] Peter W Christensen and Anders Klarbring. *An introduction to structural optimization*, volume 153. Springer Science & Business Media, 2008.

[13] SK Duggal. *Design of steel structures.* Tata McGraw-Hill Education, pages 1–5, 2000.

[14] Osman K Erol and Ibrahim Eksin. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 37(2):106–111, 2006.

[15] Cengİz Eser. *Optimum Design of Steel Structures Via Artificial Bee Colony (ABC) Algorithm and SAP2000.* PhD thesis, Middle East Technical University. Retrieved from http://etd.lib.metu.edu.tr/upload/12617083/index.pdf, 2014.

[16] Bernard Espion. The vierendeel bridges over the albert canal, belgium–their significance in the story of brittle failures. *Steel Construction*, 5(4):238–243, 2012.

[17] József Farkas and Károly Jármai. *Optimum design of steel structures.* pages 1-10,Springer, 2013.

[18] József Farkas, Luis MC Simões, and Károly Jármai. Minimum cost design of a welded stiffened square plate loaded by biaxial compression. *Structural and Multidisciplinary Optimization*, 29(4):298–303, 2005.

[19] Carlos A Felippa. Introduction to finite element methods. *University of Colorado, USA*, pp 5–29, 2001.

[20] Christodoulos A Floudas. *Nonlinear and mixed-integer optimization: fundamentals and applications.* pages 3–10, Oxford University Press, 1995.

[21] George-E Forsythe and Wolfgang Richard Wasow. Finite-difference methods for partial differential equations. pp 1960.

[22] Joseph Kendall Freitag. *Architectural Engineering: With Special Reference to High Building Construction, Including Many Examples of Chicago Office Buildings.* J. Wiley & sons, 1895.

[23] Dario Gasparini and David Simmons. American truss bridge connections in the 19th century. ii: 1850–1900. *Journal of performance of constructed facilities*, 11(3):130–140, 1997.

[24] Mitsuo Gen and Runwei Cheng. *Genetic algorithms and engineering optimization*, volume 7. John Wiley & Sons, pages 3–20, 2000.

[25] Lluıs Gil and Antoni Andreu. Shape and cross-section optimisation of a truss structure. *Computers & Structures*, 79(7):681–689, 2001.

[26] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[27] David E Goldberg and Manohar P Samtani. Engineering optimization via genetic algorithm. In *Electronic computation*, pages 471–482. ASCE, 1986.

[28] JH Holland. Adaptation in natural and artificial systems, univ. of mich. press. *Ann Arbor*, 1975.

[29] Max Hultman. Weight optimization of steel trusses by a genetic algorithm-size, shape and topology optimization according to eurocode. *TVBK-5176*, 2010.

[30] Jussi Jalkanen. *Tubular truss optimization using heuristic algorithms*. pages 1–20, 2007.

[31] K Jármai and J Farkas. Optimum cost design of welded box beams with longitudinal stiffeners using advanced backtrack method. *Structural and multidisciplinary optimization*, 21(1):52–59, 2001.

[32] Károly Jármai and Jozsef Farkas. Cost calculation and optimisation of welded steel structures. *Journal of Constructional Steel Research*, 50(2):115–135, 1999.

[33] O Kamal, O EL-Mahdy, M Nour, and G EL-Komy. Optimization of steel structures using particle swarm technique. pages 2–5, 2008.

[34] A Kaveh and S Talatahari. A discrete big bang-big crunch algorithm for optimal design of skeletal structures. *Asian Journal of Civil Engineering*, 11(1):103–122, 2010.

[35] Ali Kaveh and S Talatahari. Size optimization of space trusses using big bang–big crunch algorithm. *Computers & structures*, 87(17):1129–1140, 2009.

[36] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[37] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4104–4108. IEEE, 1997.

[38] Petri Kere and Jussi Jalkanen. Parallel particle swarm-based structural optimization in a distributed grid computing environment. In *3 rd AIAA Multidisciplinary Design Optimization Specialist Conference*, pages 2-10, 2007.

[39] NS Khot and L Berke. Structural optimization using optimality criteria methods. *Air Force Flight Dynamics Lab*, pages 47–74, 1984.

[40] Uri Kirsch. Optimal topologies of truss structures. *Computer Methods in Applied Mechanics and Engineering*, 72(1):15–28, 1989.

[41] Uri Kirsch. *Optimum structural design: concepts, methods, and applications.* McGraw-Hill Companies, pages 2–28, 1981.

[42] Dennis Lam, Thien-Cheong Ang, and Sing-Ping Chiew. *Structural steelwork: design to limit state theory.* Crc Press, pages 1-7, 2007.

[43] Juan Pablo Leiva, Brian C Watson, and Iku Kosaka. A comparative study of topology and topometry structural optimization methods within the genesis software. In *7th World Congress of Structural and Multidisciplinary Optimization*, volume 2, pages 1–5, 2007.

[44] Thomas Leslie. Built like bridges: Iron, steel, and rivets in the nineteenth-century skyscraper. *Journal of the Society of Architectural Historians*, 69(2):234–261, 2010.

[45] Hongbo Liu, Shichang Sun, and Ajith Abraham. Particle swarm approach to scheduling work-flow applications in distributed data-intensive computing environments. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 2, pages 661–666. IEEE, 2006.

[46] Rautaruukki Metform. Rautaruukki's structural hollow section manual. *Finnish) Hameenlinna*, pages 9–20,1997.

[47] Zbigniew Michalewicz. Gas: What are they? In *Genetic algorithms+ data structures= evolution programs*, pages 13–30. Springer, 1994.

[48] Melanie Mitchell. Ld davis, handbook of genetic algorithms, pages 325–330, 1998.

[49] Tsuneyoshi Nakamura and Makoto Ohsaki. A natural generator of optimum topology of plane trusses for specified fundamental frequency. *Computer methods in applied mechanics and engineering*, 94(1):113–129, 1992.

[50] George L Nemhauser and Laurence A Wolsey. Integer and combinatorial optimization. interscience series in discrete mathematics and optimization. *ed: John Wiley & Sons*, 1988.

[51] John Tinsley Oden and Junuthula Narasimha Reddy. *An introduction to the mathematical theory of finite elements*. Courier Corporation, pages 197–210, 2012.

[52] Vagelis Plevris, Apostolis Batavanis, and Manolis Papadrakakis. Optimum design of steel structures with the particle swarm optimization method based on ec3. *Computational Methods in Structural Dynamics and Earthquake Engineering*, 2011.

[53] Mariusz Pyrz. Evolutionary algorithm integrating stress heuristics for truss optimization. *Optimization and Engineering*, 5(1):45–57, 2004.

[54] ULF TORBJÖRN RINGERTZ. A branch and bound algorithm for topology optimization of truss structures. *Engineering Optimization*, 10(2):111–124, 1986.

[55] JL Rumpf. Riveted and bolted connections. *Structural Steel Design*, 2, 1964.

[56] Mamatha Sirigiri. *Preliminary Structural Design Optimization of Tall Buildings Using GS-USA© Frame3D*. PhD thesis, Arizona State University, 2014.

[57] SN Sivanandam and SN Deepa. Genetic algorithm optimization problems. In *Introduction to Genetic Algorithms*, pages 165–209. Springer, 2008.

[58] G Sved and Z Ginos. Structural optimization under multiple loading. *International Journal of Mechanical Sciences*, 10(10):803–805, 1968.

[59] Pavel Y Tabakov and Sibusiso Moyob. An efficient design optimization of anisotropic cylinders using evolutionary algorithms. pages 2–5, 2015.

[60] Hesheng Tang, Jin Zhou, Songtao Xue, and Liyu Xie. Big bang-big crunch optimization for parameter estimation in structural systems. *Mechanical Systems and Signal Processing*, 24(8):2888–2897, 2010.

[61] Pieter Truijens. Sound: a historical connection method for steel constructions. 2001.

[62] Garret N Vanderplaats. Structural optimization: where we've been and where we're going. *WIT Transactions on The Built Environment*, 31:10–11, 1970.

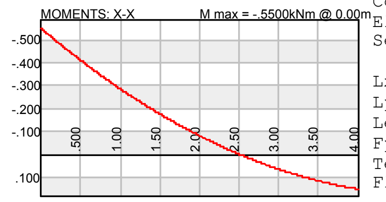
[63] AT Vu and F Werner. Optimization of steel frame structures based on differential evolution algorithm. In *18th International Conference on the application of computer science and mathematics in architecture and civil engineering, Weimar*, pages 2–6, 2009.

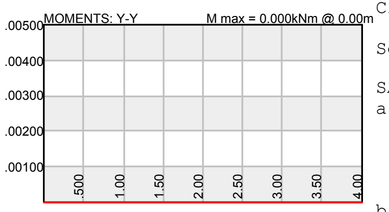
Appendix A - Finite element analysis results

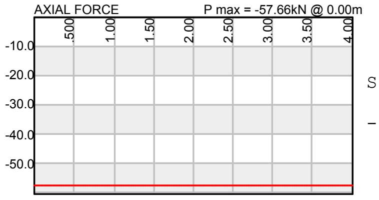
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

Member Design for Combined Stresses

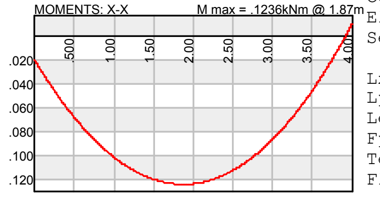
Task: Task 1

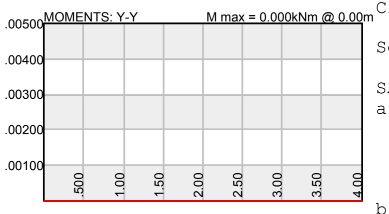


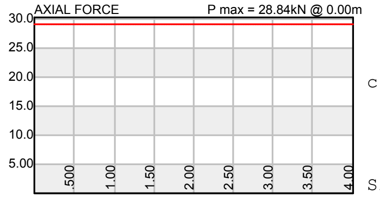




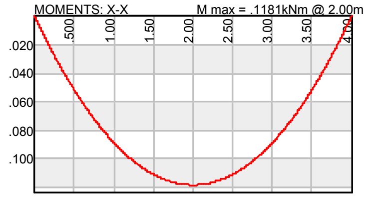

```
Combine Ver W2.6.10
Element 1-2                    Evaluate current section
Section name H

Lx Eff = 3.400 m           W1x = 1.00
Ly Eff = 3.400 m           W1y = 1.00
Le Eff = 4.000 m           W2  = 2.06
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu   Mux   Muy   57.7   .550   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.36                 OK
   Tr   Mrx   Mry   245    4.57   3.42

b) Lateral torsional buckling strength
   Mux    Muy     T/A      .550   0.00   75.18
   --- + --- - ------- = ---- + ---- - ----- = -0.11          OK
   Mrx    Mry    Mrx/Zpx    4.57   3.42   319.5

Slenderness Ratio: L/r = 225                                  OK

------------------------------------------------------------
```

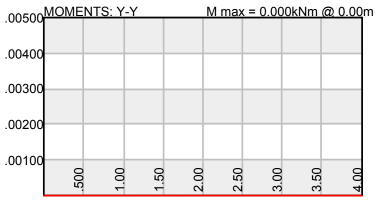


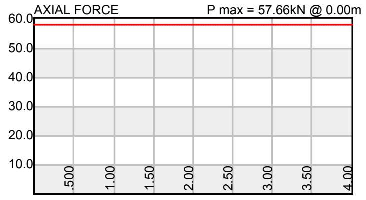




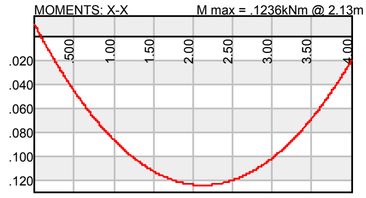

```
Combine Ver W2.6.10
Element 2-3                    Evaluate current section
Section name H

Lx Eff = 3.400 m           W1x = 1.00
Ly Eff = 3.400 m           W1y = 1.00
Le Eff = 4.000 m           W2  = 1.00
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 2.000 m)
   Cu   Mux   Muy   28.8   .123   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.14                 OK
   Cr   Mrx   Mry   245    4.57   3.42

b) Overall member strength
   Cu   U1xMux   U1yMuy   28.8   .250   0.00
   -- + ------ + ------ = ---- + ---- + ---- = 1.17           FAIL
   Cr    Mrx      Mry     25.8   4.57   3.42

c) Lateral torsional buckling strength
   Cu   U1xMux   U1yMuy   28.8   .250   0.00
   -- + ------ + ------ = ---- + ---- + ---- = 1.17           FAIL
   Cr    Mrx      Mry     25.8   4.57   3.42

Slenderness Ratio: L/r = 225                                  FAIL

------------------------------------------------------------
```

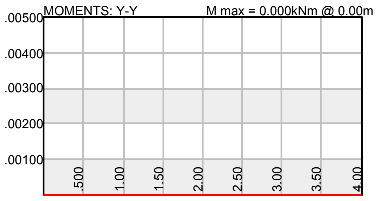


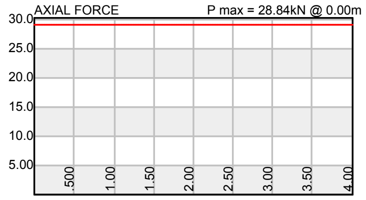





```
Combine Ver W2.6.10
Element 3-4                    Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 1.00
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 2.000 m)
   Cu    Mux    Muy    57.7    .118    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.26              OK
   Cr    Mrx    Mry    245     4.57    3.42

b) Overall member strength
   Cu    U1xMux    U1yMuy    57.7    12E18    0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****        FAIL
   Cr    Mrx       Mry       25.8    4.57     3.42

c) Lateral torsional buckling strength
   Cu    U1xMux    U1yMuy    57.7    12E18    0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****        FAIL
   Cr    Mrx       Mry       25.8    4.57     3.42

Slenderness Ratio: L/r = 225                               FAIL

-----------------------------------------------------------
```

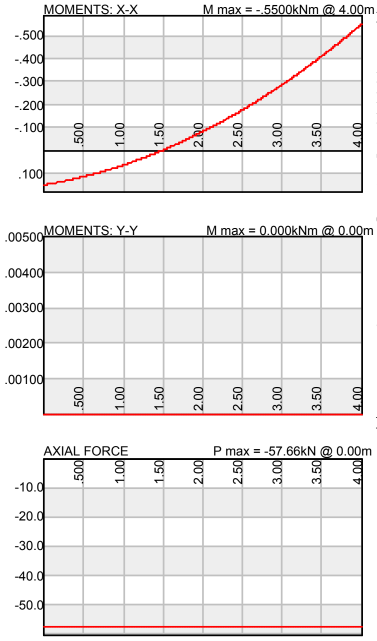




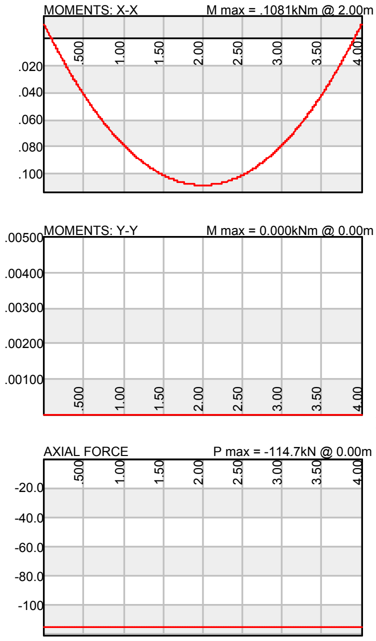



```
Combine Ver W2.6.10
Element 4-5                    Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 1.00
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 2.000 m)
   Cu    Mux    Muy    28.8    .123    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.14              OK
   Cr    Mrx    Mry    245     4.57    3.42

b) Overall member strength
   Cu    U1xMux    U1yMuy    28.8    .250    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 1.17        FAIL
   Cr    Mrx       Mry       25.8    4.57    3.42

c) Lateral torsional buckling strength
   Cu    U1xMux    U1yMuy    28.8    .250    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 1.17        FAIL
   Cr    Mrx       Mry       25.8    4.57    3.42

Slenderness Ratio: L/r = 225                               FAIL

-----------------------------------------------------------
```

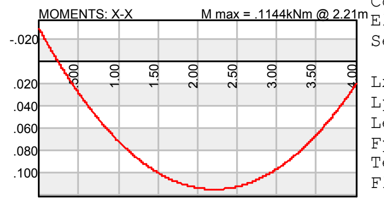
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

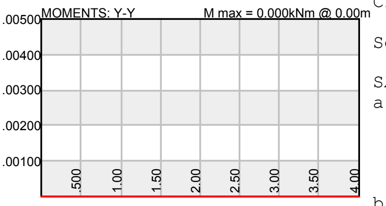


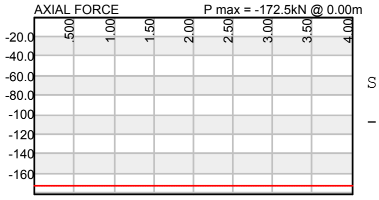

```
Combine Ver W2.6.10
Element 5-6                    Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 2.06
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 4.000 m)
   Tu    Mux    Muy    57.7    .550    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.36          OK
   Tr    Mrx    Mry    245     4.57    3.42

b) Lateral torsional buckling strength
   Mux    Muy      T/A      .550    0.00    75.18
   --- + --- - ------- = ---- + ---- - ----- = -0.11   OK
   Mrx    Mry    Mrx/Zpx    4.57    3.42    319.5

Slenderness Ratio: L/r = 225                           OK

------------------------------------------------------------
```



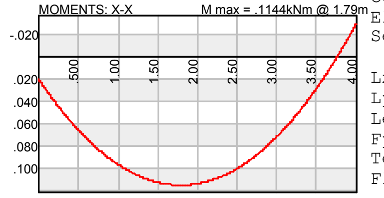

```
Combine Ver W2.6.10
Element 7-8                    Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 1.00
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 2.000 m)
   Tu    Mux    Muy    115     .108    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.49          OK
   Tr    Mrx    Mry    245     4.57    3.42

b) Lateral torsional buckling strength
   Mux    Muy      T/A      .108    0.00    149.5
   --- + --- - ------- = ---- + ---- - ----- = -0.44   OK
   Mrx    Mry    Mrx/Zpx    4.57    3.42    319.5

Slenderness Ratio: L/r = 225                           OK

------------------------------------------------------------
```

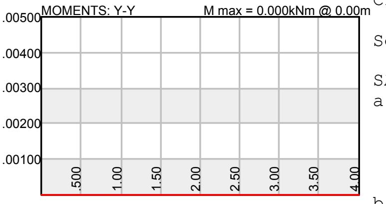


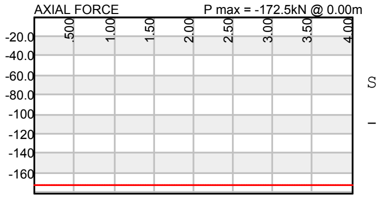





```
Combine Ver W2.6.10
Element 8-9                      Evaluate current section
Section name H

Lx Eff = 3.400 m            W1x = 1.00
Ly Eff = 3.400 m            W1y = 1.00
Le Eff = 4.000 m            W2  = 1.00
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 2.333 m)
   Tu    Mux    Muy    172     .114    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.73            OK
   Tr    Mrx    Mry    245     4.57    3.42

b) Lateral torsional buckling strength
   Mux     Muy      T/A       .114    0.00    224.9
   --- + --- - ------- = ---- + ---- - ----- = -0.68     OK
   Mrx     Mry     Mrx/Zpx    4.57    3.42    319.5

Slenderness Ratio: L/r = 225                             OK

------------------------------------------------------------
```

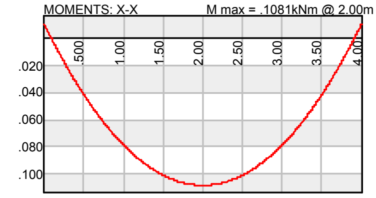


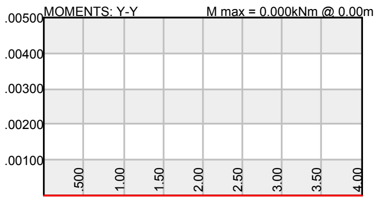


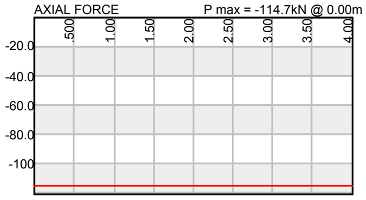



```
Combine Ver W2.6.10
Element 9-10                      Evaluate current section
Section name H

Lx Eff = 3.400 m            W1x = 1.00
Ly Eff = 3.400 m            W1y = 1.00
Le Eff = 4.000 m            W2  = 1.00
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 1.667 m)
   Tu    Mux    Muy    172     .114    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.73            OK
   Tr    Mrx    Mry    245     4.57    3.42

b) Lateral torsional buckling strength
   Mux     Muy      T/A       .114    0.00    224.9
   --- + --- - ------- = ---- + ---- - ----- = -0.68     OK
   Mrx     Mry     Mrx/Zpx    4.57    3.42    319.5

Slenderness Ratio: L/r = 225                             OK

------------------------------------------------------------
```

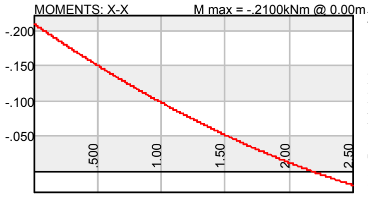
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

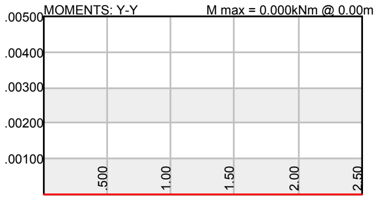


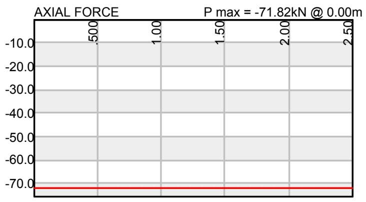




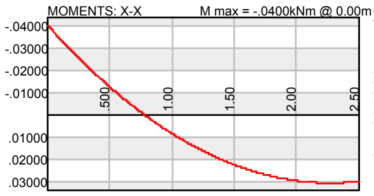

```
Combine Ver W2.6.10
Element 10-11                       Evaluate current section
Section name H

Lx Eff = 3.400 m            W1x = 1.00
Ly Eff = 3.400 m            W1y = 1.00
Le Eff = 4.000 m            W2  = 1.00
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 2.000 m)
   Tu    Mux    Muy    115     .108    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.49          OK
   Tr    Mrx    Mry    245     4.57    3.42

b) Lateral torsional buckling strength
   Mux    Muy      T/A        .108    0.00    149.5
   --- + --- - ------- = ---- + ---- - ----- = -0.44   OK
   Mrx    Mry    Mrx/Zpx    4.57    3.42    319.5

Slenderness Ratio: L/r = 225                           OK

------------------------------------------------------------
```

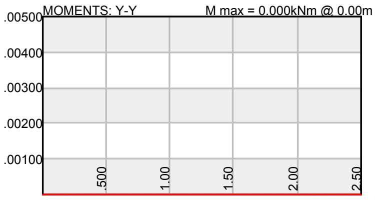


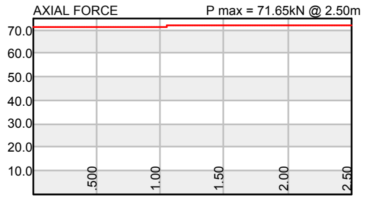




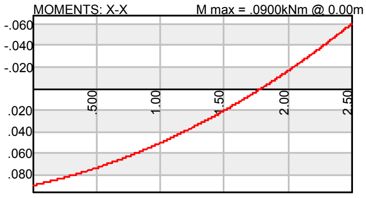

```
Combine Ver W2.6.10
Element 1-7                       Evaluate current section
Section name B

Lx Eff = 2.125 m            W1x = 1.00
Ly Eff = 2.125 m            W1y = 1.00
Le Eff = 2.500 m            W2  = 1.85
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu    Mux    Muy    71.8    .210    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.61          OK
   Tr    Mrx    Mry    142     2.09    1.27

b) Lateral torsional buckling strength
   Mux    Muy      T/A        .210    0.00    161.8
   --- + --- - ------- = ---- + ---- - ----- = -0.41   OK
   Mrx    Mry    Mrx/Zpx    2.09    1.27    319.5

Slenderness Ratio: L/r = 224                           OK

------------------------------------------------------------
```

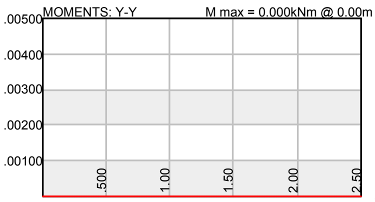


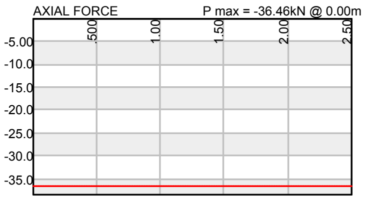





```
Combine Ver W2.6.10
Element 2-7                        Evaluate current section
Section name B

Lx Eff = 2.125 m           W1x = 1.00
Ly Eff = 2.125 m           W1y = 1.00
Le Eff = 2.500 m           W2  = 2.50
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)   = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 50x25x3.5      Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Cu    Mux    Muy    71.6    .040    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.52              OK
   Cr    Mrx    Mry    142     2.09    1.27

b) Overall member strength
   Cu    U1xMux    U1yMuy    71.7    40E17    0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****        FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

c) Lateral torsional buckling strength
   Cu    U1xMux    U1yMuy    71.7    40E17    0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****        FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

Slenderness Ratio: L/r = 224                               FAIL

------------------------------------------------------------
```

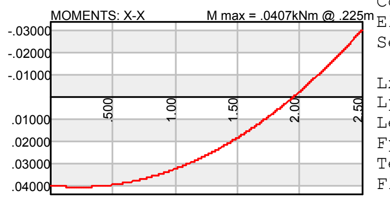


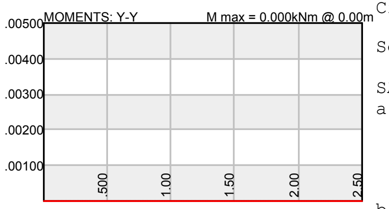


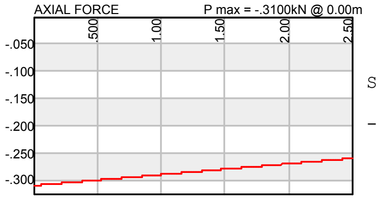



```
Combine Ver W2.6.10
Element 2-8                        Evaluate current section
Section name B

Lx Eff = 2.125 m           W1x = 1.00
Ly Eff = 2.125 m           W1y = 1.00
Le Eff = 2.500 m           W2  = 2.50
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)   = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 50x25x3.5      Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu    Mux    Muy    36.5    .090    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.30              OK
   Tr    Mrx    Mry    142     2.09    1.27

b) Lateral torsional buckling strength
   Mux    Muy      T/A      .090    0.00    82.12
   --- + --- - ------- = ---- + ---- - ----- = -0.21       OK
   Mrx    Mry    Mrx/Zpx    2.09    1.27    319.5

Slenderness Ratio: L/r = 224                               OK

------------------------------------------------------------
```

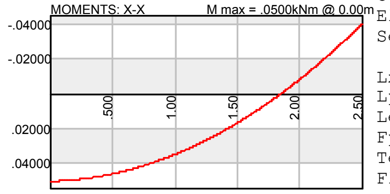
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

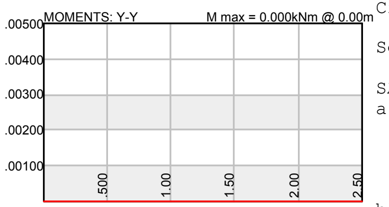


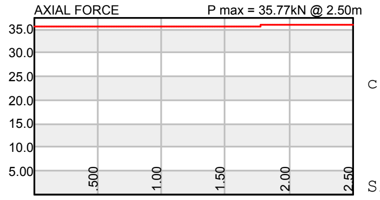





```
Combine Ver W2.6.10
Element 3-9                        Evaluate current section
Section name B

Lx Eff = 2.125 m            W1x = 1.00
Ly Eff = 2.125 m            W1y = 1.00
Le Eff = 2.500 m            W2  = 1.00
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 50x25x3.5      Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.208 m)
   Tu    Mux    Muy    .306    .041    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.02            OK
   Tr    Mrx    Mry    142     2.09    1.27

b) Lateral torsional buckling strength
   Mux    Muy      T/A      .041    0.00    .6982
   --- + --- - ------- = ---- + ---- - ----- = 0.02      OK
   Mrx    Mry    Mrx/Zpx    2.09    1.27    319.5

Slenderness Ratio: L/r = 224                             OK

------------------------------------------------------------
```

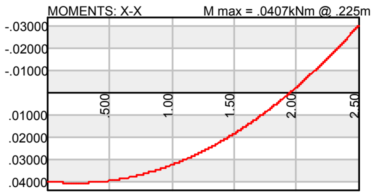


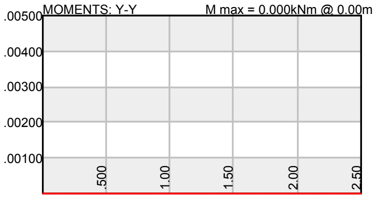


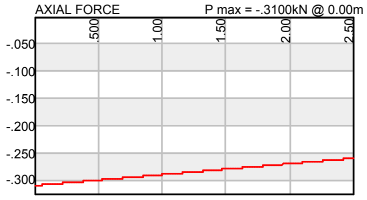



```
Combine Ver W2.6.10
Element 3-8                        Evaluate current section
Section name B

Lx Eff = 2.125 m            W1x = 1.00
Ly Eff = 2.125 m            W1y = 1.00
Le Eff = 2.500 m            W2  = 2.50
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 50x25x3.5      Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Cu    Mux    Muy    35.7    .050    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.28            OK
   Cr    Mrx    Mry    142     2.09    1.27

b) Overall member strength
   Cu    U1xMux    U1yMuy    35.8    .152    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 2.45      FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

c) Lateral torsional buckling strength
   Cu    U1xMux    U1yMuy    35.8    .152    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 2.45      FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

Slenderness Ratio: L/r = 224                             FAIL

------------------------------------------------------------
```

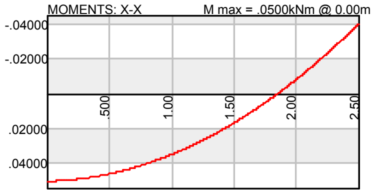
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

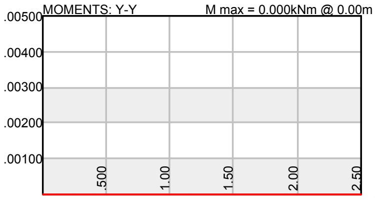


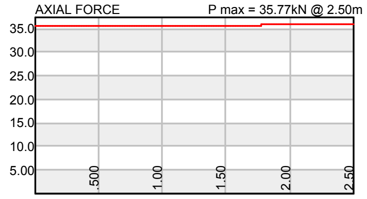





```
Combine Ver W2.6.10
Element 4-9                     Evaluate current section
Section name B

Lx Eff = 2.125 m            W1x = 1.00
Ly Eff = 2.125 m            W1y = 1.00
Le Eff = 2.500 m            W2  = 1.00
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.208 m)
   Tu    Mux    Muy    .306    .041    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.02                 OK
   Tr    Mrx    Mry    142     2.09    1.27

b) Lateral torsional buckling strength
   Mux    Muy      T/A      .041    0.00    .6982
   --- + --- - ------- = ---- + ---- - ----- = 0.02           OK
   Mrx    Mry    Mrx/Zpx    2.09    1.27    319.5

Slenderness Ratio: L/r = 224                                  OK

-------------------------------------------------------------
```

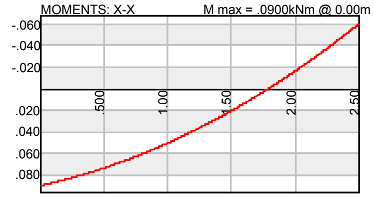


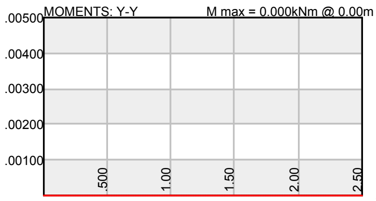


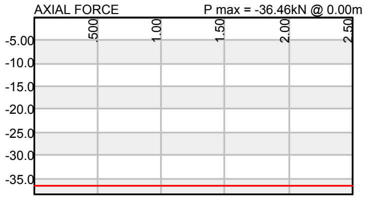



```
Combine Ver W2.6.10
Element 4-10                     Evaluate current section
Section name B

Lx Eff = 2.125 m            W1x = 1.00
Ly Eff = 2.125 m            W1y = 1.00
Le Eff = 2.500 m            W2  = 2.50
Fy = 355 MPa                Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1             Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Cu    Mux    Muy    35.7    .050    0.00
   -- + --- + --- = ---- + ---- + ---- = 0.28                 OK
   Cr    Mrx    Mry    142     2.09    1.27

b) Overall member strength
   Cu    U1xMux    U1yMuy    35.8    .152    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 2.45           FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

c) Lateral torsional buckling strength
   Cu    U1xMux    U1yMuy    35.8    .152    0.00
   -- + ------ + ------ = ---- + ---- + ---- = 2.45           FAIL
   Cr     Mrx       Mry      15.0    2.09    1.27

Slenderness Ratio: L/r = 224                                  FAIL

-------------------------------------------------------------
```

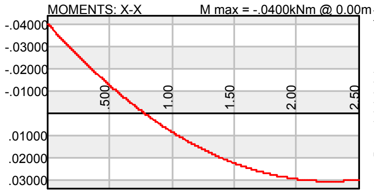
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number			Sheet
	Job Title			
	Client			
	Calcs by	Checked by	Date	

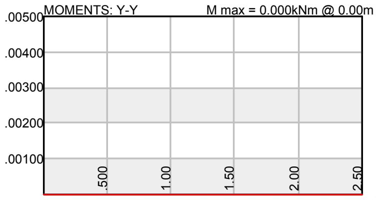


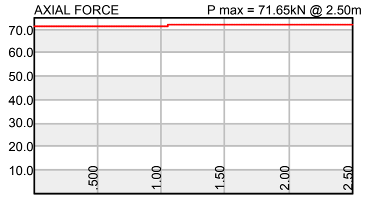





```
Combine Ver W2.6.10
Element 5-10                       Evaluate current section
Section name B

Lx Eff = 2.125 m           W1x = 1.00
Ly Eff = 2.125 m           W1y = 1.00
Le Eff = 2.500 m           W2  = 2.50
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)   = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu   Mux   Muy   36.5   .090   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.30          OK
   Tr   Mrx   Mry   142    2.09   1.27

b) Lateral torsional buckling strength
   Mux   Muy     T/A      .090   0.00   82.12
   --- + --- - ------- = ---- + ---- - ----- = -0.21   OK
   Mrx   Mry   Mrx/Zpx   2.09   1.27   319.5

Slenderness Ratio: L/r = 224                           OK

------------------------------------------------------------
```

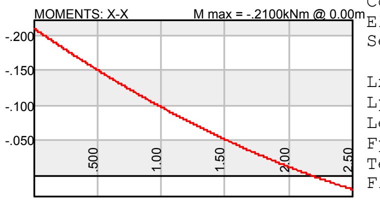


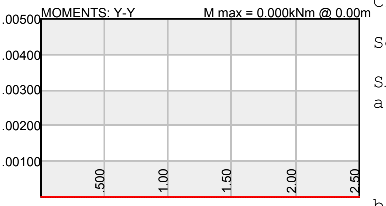


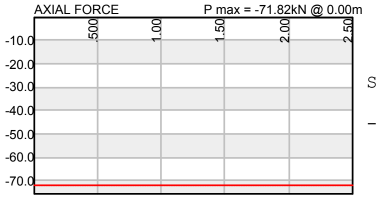



```
Combine Ver W2.6.10
Element 5-11                       Evaluate current section
Section name B

Lx Eff = 2.125 m           W1x = 1.00
Ly Eff = 2.125 m           W1y = 1.00
Le Eff = 2.500 m           W2  = 2.50
Fy = 355 MPa               Fu  = 450 MPa
Tension area factor (Ane/Ag)   = 1.00
Flange class: 1            Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.8.1 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Cu   Mux   Muy   71.6   .040   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.52          OK
   Cr   Mrx   Mry   142    2.09   1.27

b) Overall member strength
   Cu   U1xMux   U1yMuy   71.7   40E17   0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****    FAIL
   Cr    Mrx      Mry     15.0   2.09   1.27

c) Lateral torsional buckling strength
   Cu   U1xMux   U1yMuy   71.7   40E17   0.00
   -- + ------ + ------ = ---- + ---- + ---- = ****    FAIL
   Cr    Mrx      Mry     15.0   2.09   1.27

Slenderness Ratio: L/r = 224                           FAIL

------------------------------------------------------------
```

PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

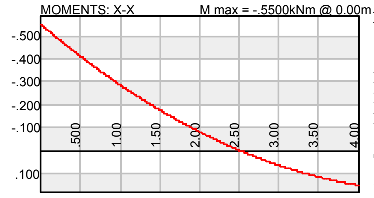


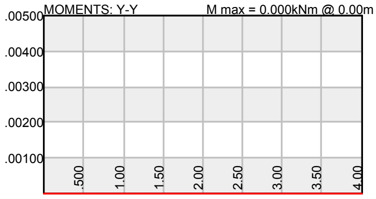


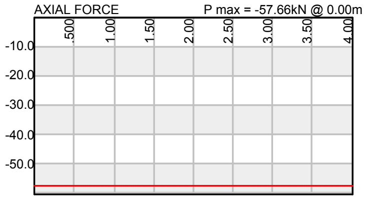



```
Combine Ver W2.6.10
Element 6-11                    Evaluate current section
Section name B

Lx Eff = 2.125 m          W1x = 1.00
Ly Eff = 2.125 m          W1y = 1.00
Le Eff = 2.500 m          W2  = 1.85
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 50x25x3.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu    Mux   Muy   71.8   .210   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.61                 OK
   Tr    Mrx   Mry   142    2.09   1.27

b) Lateral torsional buckling strength
   Mux   Muy     T/A      .210   0.00   161.8
   --- + --- - ------- = ---- + ---- - ----- = -0.41          OK
   Mrx   Mry   Mrx/Zpx   2.09   1.27   319.5

Slenderness Ratio: L/r = 224                                  OK

------------------------------------------------------------
```

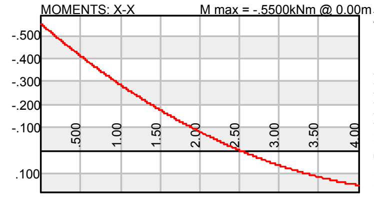
PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title		
	Client		
	Calcs by	Checked by	Date

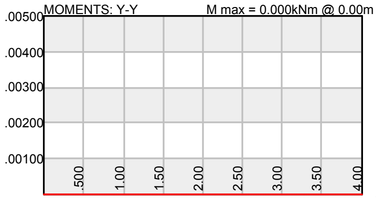


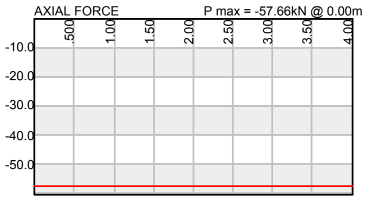





```
Combine Ver W2.6.10
Element 1-2                     Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 2.06
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu   Mux   Muy   57.7   .550   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.36             OK
   Tr   Mrx   Mry   245    4.57   3.42

b) Lateral torsional buckling strength
   Mux   Muy     T/A     .550   0.00   75.18
   --- + --- - ------- = ---- + ---- - ----- = -0.11      OK
   Mrx   Mry   Mrx/Zpx   4.57   3.42   319.5

Slenderness Ratio: L/r = 225                              OK

------------------------------------------------------------
```








```
Combine Ver W2.6.10
Element 1-2                     Evaluate current section
Section name H

Lx Eff = 3.400 m          W1x = 1.00
Ly Eff = 3.400 m          W1y = 1.00
Le Eff = 4.000 m          W2  = 2.06
Fy = 355 MPa              Fu  = 450 MPa
Tension area factor (Ane/Ag)  = 1.00
Flange class: 1           Web class   : 1

Critical Load Case : 1

Section 60x40x4.5     Rect.hollow sections (Vert)

SABS 0162 - 1993 13.9 :
a) Cross-sectional strength (Crit. pos.= 0.000 m)
   Tu   Mux   Muy   57.7   .550   0.00
   -- + --- + --- = ---- + ---- + ---- = 0.36             OK
   Tr   Mrx   Mry   245    4.57   3.42

b) Lateral torsional buckling strength
   Mux   Muy     T/A     .550   0.00   75.18
   --- + --- - ------- = ---- + ---- - ----- = -0.11      OK
   Mrx   Mry   Mrx/Zpx   4.57   3.42   319.5

Slenderness Ratio: L/r = 225                              OK

------------------------------------------------------------
```

Appendix B - Computer programs

Information data

```
INTEGER PopSize ! Population size
INTEGER NParam ! Number of parameters
INTEGER NGen ! Maximum number of generations
REAL*8 Beta ! Parameter controlling the influence of the global best solution
INTEGER*1 BetaType ! Fixed beta or random
INTEGER*1 ProblemType ! min or max

PARAMETER (PopSize = 100, NParam = 17, NGen = 1000, &
Beta = 0.7, BetaType = 0, ProblemType = 0)

REAL*8 ParMin(NParam) ! Minimum values of parameters
REAL*8 ParMax(NParam) ! Maximum values of parameters

DATA ParMin/NParam*1.0/, ParMax/NParam*25.0/

TYPE INDIVIDUAL
SEQUENCE
REAL*8 X(NParam)
REAL*8 Fitness
END TYPE INDIVIDUAL

TYPE(INDIVIDUAL) POPULATION(POPSIZE)
```

!* Beta = [0, 1], for example, if
!* Beta = 0 - center of mass where the best individual is
!* Beta = 1 - actual center of mass is calculated

!* BetaType = 0 - Fixed values
!* BetaType = 1 - Random number between 0 and 1

!* ProblemType = 0 - Minimization
!* ProblemType = 1 - Maximization

Main file

```
INTEGER PopSize ! Population size
INTEGER NParam ! Number of parameters
INTEGER NGen ! Maximum number of generations
REAL*8 Beta ! Parameter controlling the influence of the global best solution
INTEGER*1 BetaType ! Fixed beta or random
INTEGER*1 ProblemType ! min or max

PARAMETER (PopSize = 100, NParam = 17, NGen = 1000, &
Beta = 0.7, BetaType = 0, ProblemType = 0)

REAL*8 ParMin(NParam) ! Minimum values of parameters
REAL*8 ParMax(NParam) ! Maximum values of parameters

DATA ParMin/NParam*1.0/, ParMax/NParam*25.0/

TYPE INDIVIDUAL
SEQUENCE
REAL*8 X(NParam)
REAL*8 Fitness
END TYPE INDIVIDUAL
```

TYPE(INDIVIDUAL) POPULATION(POPSIZE)

!* Beta = [0, 1], for example, if
!* Beta = 0 - center of mass where the best individual is
!* Beta = 1 - actual center of mass is calculated

!* BetaType = 0 - Fixed values
!* BetaType = 1 - Random number between 0 and 1

!* ProblemType = 0 - Minimization
!* ProblemType = 1 - Maximization

Objective function

REAL*8 FUNCTION OBJFUNCTION(ar,nelements)
implicit none
integer*1 ipenalty
integer nnodes,nsupports
integer*2 nelements
integer i,j
REAL*8 ar(nelements),area(55)
real*8 nodes(55,2),forces(55,2),dx,dy,length(55)
real*8 sinus(55),cosinus(55),pi,pxy(110),p(55)
integer displ(55,2),nelnode(55),elnode(55,6)
real*8 e(55),fe(55),eldelta(55,4)
integer elements(55,2),i1,i2,order,n,ip,ied
integer ind,din1,din2,din3,din4,nen,iel,ner
real*8 det,delta(55),s,kel(4,4,55),h(4,4,55)
real*8 sigma
real*8 catalogue(25)
real*8 alpha,beta,gs,gu,rho,W,sigma$_{m}ax, delta_{m}ax$
$real * 8intforces(55, 4), Kgl$
$ALLOCATABLE :: Kgl(:, :)$

```
common/elchar/fe,sinus,cosinus
common/elmatrix/kel
common/tmatrix/h
COMMON/TRES/DELTA,INTFORCES,ORDER
COMMON/TDATA/FORCES,NODES,E,ELEMENTS,DISPL,NNODES
COMMON/geom/length
COMMON/WEIGHT/W
COMMON/CAT/catalogue
COMMON/PENALTY_FUNC/ipenalty
COMMON/DENSITY/rho

DO I = 1,NELEMENTS
AREA(I) = CATALOGUE(INT(AR(I)+0.5))
END DO

!*** CALCULATIONS OF geometric and physical characteristics

pi = 4.0*atan(1.0)
do 150 i = 1,nelements
i1 = elements(i,1)
i2 = elements(i,2)
dx = nodes(i2,1)- nodes(i1,1)
dy = nodes(i2,2)- nodes(i1,2)
length(i) = sqrt(dx*dx + dy*dy)
sinus(i) = dy/length(i)
cosinus(i) = dx/length(i)
fe(i) = e(i)*area(i)/length(i)
150 continue

!Com: Order of the system
order = 0
do i = 1, nnodes
do j = 1,2
if (displ(i,j) .ne. 0) then
order = order + 1
```

```
displ(i,j) = order
end if
end do
end do
!Com: Number of elements joining each node and their numbers
nelnode = 0

do i = 1, nnodes
n = 1
do j = 1, nelements
if ((elements(j,1).eq.i) .or. (elements(j,2).eq.i)) then
nelnode(i) = nelnode(i) + 1
elnode(i,n) = j
n = n + 1
end if
end do
end do
!Com: Make a table later (for printing)!2
!c do i = 1, nnodes
!c print*, i, nelnode(i),(elnode(i,j),j=1,nelnode(i))
!c end do
!******************

ALLOCATE(Kgl(order,order))

Kgl = 0.0

CALL elstiffness(nelements)

!COM: BUILD THE STIFFNESS MATRIX FOR THE WHOLE SYSTEM

DO 1000 IND = 1, nnodes
din1 = 0
din2 = 0
```

```
din3 = 0
din4 = 0
!*** MAIN NODE AND DIAGONAL end
IF(DISPL(IND,1) .NE. 0) THEN
din1 = displ(ind,1)
DO 200 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
Kgl(din1,din1)=Kgl(din1,din1)+kel(1,1,elnode(ind,iel))
if (displ(ind,2) .ne. 0) Kgl(din1+1,din1)=Kgl(din1+1,din1)+kel(2,1,elnode(ind,iel))
else
Kgl(din1,din1)=Kgl(din1,din1)+kel(3,3,elnode(ind,iel))
if (displ(ind,2) .ne. 0)Kgl(din1+1,din1)=Kgl(din1+1,din1)+kel(4,3,elnode(ind,iel))
end if
200 CONTINUE
END IF

IF(DISPL(IND,2) .NE. 0) THEN
din2 = displ(ind,2)

DO 300 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
Kgl(din2,din2)=Kgl(din2,din2)+kel(2,2,elnode(ind,iel))
if (displ(ind,1) .ne. 0)Kgl(din2-1,din2)=Kgl(din2-1,din2)+kel(1,2,elnode(ind,iel))
else
Kgl(din2,din2)=Kgl(din2,din2)+kel(4,4,elnode(ind,iel))
if (displ(ind,1) .ne. 0)Kgl(din2-1,din2)=Kgl(din2-1,din2)+kel(3,4,elnode(ind,iel))
end if
300 CONTINUE
END IF
!*** OPPOSITE ENDS OF ELEMENTS
DO 400 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
nen = elements(elnode(ind,iel),2)
else
nen = elements(elnode(ind,iel),1)
```

```
end if

if(displ(nen,1) .ne. 0) then
din3 = displ(nen,1)
if(din1 .ne. 0) then
Kgl(din3,din1) = Kgl(din3,din1)+kel(3,1,elnode(ind,iel))
end if
if(din2 .ne. 0) then
Kgl(din3,din2) = Kgl(din3,din2)+kel(3,2,elnode(ind,iel))
end if
end if

if(displ(nen,2) .ne. 0) then
din4 = displ(nen,2)
if(din1 .ne. 0) then
Kgl(din4,din1) = Kgl(din4,din1)+kel(4,1,elnode(ind,iel))
end if
if(din2 .ne. 0) then
Kgl(din4,din2) = Kgl(din4,din2)+kel(4,2,elnode(ind,iel))
end if
end if
400 continue
1000 CONTINUE

!Com: Vector of the external forces
ip = 1
do i = 1,nnodes
if (displ(i,1) .ne.0) then
p(ip) = forces(i,1)
ip = ip+1
end if
if (displ(i,2) .ne.0) then
p(ip) = forces(i,2)
ip = ip+1
end if
```

```
end do
!Com: Print the vector of the external forces
!Com: Inverse the stiffness matrix
call rsudm(kgl,order,0,NER,DET,IED)

!Com: Global displacements

delta = 0.0

do i = 1, order
s = 0
do j = 1,order
s = s + Kgl(i,j) * p(j)
end do
delta(i) = s
end do

!Com: Local displacements and internal forces
eldelta = 0.0

do i = 1, nelements
if(displ(elements(i,1),1) .ne. 0)eldelta(i,1) = delta(displ(elements(i,1),1))

if(displ(elements(i,1),2) .ne. 0) eldelta(i,2) = delta(displ(elements(i,1),2))

if(displ(elements(i,2),1) .ne. 0) eldelta(i,3) = delta(displ(elements(i,2),1))

if(displ(elements(i,2),2) .ne. 0) eldelta(i,4) = delta(displ(elements(i,2),2))
!print*, ' el displ:'
!write(*,'(3g12.4)') eldelta(i,1),eldelta(i,2),eldelta(i,3)
!write(*,'(g12.4)') eldelta(i,4)
! pause
end do
```

do i = 1, nelements
end do
!Com: Internal forces
CALL transform(nelements)

intforces = 0.0
do i = 1, nelements

intforces(i,1) = h(1,1,i)*eldelta(i,1)+h(1,2,i)*eldelta(i,2)+&
h(1,3,i)*eldelta(i,3)+h(1,4,i)*eldelta(i,4)

intforces(i,3) = h(3,1,i)*eldelta(i,1)+h(3,2,i)*eldelta(i,2)+&
h(3,3,i)*eldelta(i,3)+h(3,4,i)*eldelta(i,4)
!print*, intforces(i,1), intforces(i,3)
end do
!pause
!***

sigma$_m ax = 160.0$
$delta_m ax = 0.05$
!$sigma_m ax = 340.0$

gs = 0.0

do i = 1, nelements
sigma = abs(intforces(i,3))/area(i)
gs = gs + max(abs(sigma)/sigma$_m ax - 1.0, 0.0) * *2$
$enddo$

gu = 0.0
do i = 1,ORDER
gu = gu + max(abs(delta(i))/delta$_m ax - 1.0, 0.0) * *2$

enddo

alpha = 1.0E06; beta = 1.0E06
rho = 7850.0
! rho = 7275.0

W = 0.0

do 111 i =1, nelements
111 W = W + rho*length(i)*area(i)

OBJFUNCTION = (W + alpha*gs + beta*gu)
!OBJFUNCTION = (W + alpha*gs) ! ONLY STRESSES

!print*, 'OBJ=',objfunction
! pause
RETURN
END

Pelican

SUBROUTINE PELICAN
IMPLICIT NONE
INCLUDE 'bdata.f'
REAL*8 OBJFUNCTION
INTEGER RDIM,K
INTEGER worst$_{i}nd$
$REAL * 8, ALLOCATABLE :: RNV(:)$
$REAL * 8NPARMIN(NPARAM), NPARMAX(NPARAM), OUTSIDE$
$REAL * 8SF, SFC(nparam), eps, sigma$
$REAL * 8CONTRACTION_{R}ATIO$
$REAL * 8PERCENT$
$REAL * 8length(55), catalogue(25), area(55), w, rho$
$INTEGERI, J, GEN$

$PARAMETER(CONTRACTION_RATIO = 0.1)$
$TYPE(INDIVIDUAL)Centroid, BestIndividual, WorstIndividual, CurrentBestIndividual$
$COMMON/BI/BestIndividual$
$COMMON/WI/WorstIndividual, worst_ind$
$COMMON/CE/Centroid$
$COMMON/geom/length$
!$COMMON/WEIGHT/W$
$COMMON/CAT/catalogue$
$COMMON/DENSITY/rho$

!COM: GENERATE AN INITIAL POPULATION AND CALCULATE FITNESS

OUTSIDE = 0.1 !(10

DO I = 1, NPARAM
NPARMIN(I) = PARMIN(I) + PARMIN(I)*OUTSIDE
NPARMAX(I) = PARMAX(I) + PARMAX(I)*OUTSIDE
END DO

RDIM = POPSIZE * NPARAM

ALLOCATE(RNV(RDIM))
CALL RANDOM(RNV,RDIM)
K = 1

DO 200 I = 1, POPSIZE
DO 100 J = 1, NPARAM
POPULATION(I)
K = K + 1
IF (ABS(POPULATION(I)
POPULATION(I)
END IF
IF (ABS(POPULATION(I)
POPULATION(I)

```
END IF

100 CONTINUE

POPULATION(I)
write(*,'(11g12.3)')(population(i)
population(i)

200 CONTINUE
DEALLOCATE(RNV)

!open(13, file='initpop.txt')
!do i= 1,popsize !write(13,'(11g16.3)')&
! (population(i)
!enddo
! close(13)

!COM: FIND THE BEST INDIVIDUAL
!c CALL BESTINDIV(POPULATION)

OPEN(21, FILE='solution41.txt')

eps = 1.0e-05
!*** MAIN LOOP *** MAIN LOOP *** MAIN LOOP ***

CALL BESTINDIV(population)
CurrentBestIndividual = BestIndividual

DO 999 GEN = 1, NGEN !MAIN LOOP !!!!!!!!!!!!!
!COM: Calculate centroid if neccessary

CALL BESTINDIV(POPULATION)
```

```
if (BestIndividual
CurrentBestIndividual = BestIndividual
end if

IF (beta .EQ. 0.0) THEN
centroid = BestIndividual
ELSE
do i = 1, popsize
if((abs(population(i)
(population(i)
population(i)
end if

if((abs(population(i)
(population(i)
population(i)
end if
end do

sf = 0.0

do i = 1, popsize
sf = sf + (1./POPULATION(I)
end do

sfc = 0.0

do j = 1, nparam
do i = 1, popsize
sfc(j) = sfc(j) + population(i)
end do
end do
```

do i = 1, nparam
centroid
!c print*, centroid
end do
!c pause

centroid

END IF

!COM: DETERMINE THE BOUNDARIES OF NEW CONTRACTED SPACE

IF (BetaType .eq. 1) THEN
ALLOCATE(RNV(1))
CALL RANDOM(RNV,1)

DO I = 1, nparam
sigma = abs((ParMax(I)-ParMin(I))/(CONTRACTION$_{R}ATIO * GEN + 1$))

nparmin(i) = rnv(1)*centroid
(1.0-rnv(1))*CurrentBestIndividual
IF (nparmin(i) .lt. PARMIN(i)) nparmin(i)=PARMIN(i)

nparmax(i) = rnv(1)*centroid
(1.0-rnv(1))*CurrentBestIndividual
IF (nparmax(i) .gt. PARMAX(i)) nparmax(i)=PARMAX(i)

END DO
DEALLOCATE(RNV)
ELSE
DO I = 1, nparam
sigma = abs((ParMax(I)-ParMin(I))/(CONTRACTION$_{R}ATIO * GEN + 1$))

```
nparmin(i) = beta*centroid
(1.0-beta)*CurrentBestIndividual
IF (nparmin(i) .lt. PARMIN(i)) nparmin(i)=PARMIN(i)

nparmax(i) = beta*centroid
(1.0-beta)*CurrentBestIndividual
IF (nparmax(i) .gt. PARMAX(i)) nparmax(i)=PARMAX(i)

!print*,beta,centroid
!print*,nparmin(i),nparmax(i)
END DO
END IF
!pause
!COM: POPULATE THE NEW SOLUTION DOMAIN

ALLOCATE(RNV(RDIM))
CALL RANDOM(RNV,RDIM)
K = 1

DO 150 I = 1, POPSIZE
DO 150 J = 1, NPARAM
POPULATION(I)
K = K + 1
150 CONTINUE

DEALLOCATE(RNV)

do i = 1, popsize
POPULATION(I)
enddo

CALL WorstIndiv(POPULATION)
```

POPULATION(worst$_i nd$) = $CurrentBestIndividual$

```
!IF (GEN .EQ. 5) THEN
! open(15, file='pop5.txt')
! do i= 1,popsize
! write(15,'(11g16.3)')&
! (population(i)
! enddo
! close(15)
! END IF
!!
! IF (GEN .EQ. 7) THEN
! open(17, file='pop7.txt')
! do i= 1,popsize
!write(17,'(11g16.3)')&
! (population(i)
! enddo
! close(17)
! END IF

DO I = 1,NPARAM
AREA(I) = CATALOGUE(INT(CurrentBestIndividual
END DO

W = 0.0

do i =1, NPARAM
W = W + rho*length(i)*area(i)
end do

WRITE(21,'(I6,2G16.6)')GEN,CurrentBestIndividual
WRITE(*,'(I6,2G16.6)')GEN,CurrentBestIndividual

999 CONTINUE
```

```
BestIndividual = CurrentBestIndividual
!*** END OF MAIN LOOP ***
END ! OF THE PROGRAM

SUBROUTINE BESTINDIV(POP)
IMPLICIT NONE
INCLUDE 'bdata.f'
INTEGER J,IND
REAL*8 best
TYPE(INDIVIDUAL) pop(popsize),BestIndividual
COMMON/BI/BestIndividual

best = pop(1)
ind = 1
!COM: Looking for minimum (smallest individual)
do j = 1,popsize
if (best .gt. pop(j)
best = pop(j)
ind = j
end if
end do
BestIndividual = pop(ind)
return
end
!**********************
SUBROUTINE WorstIndiv(POP)
IMPLICIT NONE
INCLUDE 'bdata.f'
INTEGER J,IND
REAL*8 worst
TYPE(INDIVIDUAL) pop(popsize),WorstIndividual
COMMON/WI/WorstIndividual,ind

worst = pop(1)
ind = 1
```

!COM: Looking for maximum (largest individual)
do j = 1,popsize
if (worst .lt. pop(j)
worst = pop(j)
ind = j
end if
end do
WorstIndividual = pop(ind)
return
end

!**********************
SUBROUTINE RANDOM(RND,RN)
implicit none
INTEGER RN
REAL*8, dimension(RN) :: rnd
integer :: isize,idate(8)
integer, allocatable :: iseed(:)

call DATE$_{A}ND_{T}IME(VALUES = idate)$
$callRANDOM_{S}EED(SIZE = isize)$
$allocate(iseed(isize))$
$callRANDOM_{S}EED(GET = iseed)$
$iseed = iseed * (idate(8) - 500)$
$callRANDOM_{S}EED(PUT = iseed)$
$callrandom_{n}umber(rnd)$
$RETURN$
END
$! *$

Truss optimization

```
SUBROUTINE TRUSS2D(nelements,area)
implicit none
integer nnodes,nsupports
integer*2 nelements
integer i,j
real*8 nodes(55,2),forces(55,2),dx,dy,length(55)
real*8 sinus(55),cosinus(55),pi,pxy(110),p(55)
integer displ(55,2),nelnode(55),elnode(55,6)
real*8 area(nelements),e(55),fe(55),eldelta(55,4)
integer elements(55,2),i1,i2,order,n,ip,ied
integer ind,din1,din2,din3,din4,nen,iel,ner
real*8 det,delta(55),s,kel(4,4,55),h(4,4,55)
real*8 intforces(55,4),Kgl
ALLOCATABLE :: Kgl(:,:)

common/elchar/fe,sinus,cosinus
common/elmatrix/kel
common/tmatrix/h
COMMON/TDATA/FORCES,NODES,E,ELEMENTS,DISPL,NNODES
COMMON/TRES/DELTA,INTFORCES,ORDER
COMMON/geom/length

!*** CALCULATIONS OF geometric and physical characteristics

pi = 4.0*atan(1.0)
do 150 i = 1,nelements
i1 = elements(i,1)
i2 = elements(i,2)
dx = nodes(i2,1)- nodes(i1,1)
dy = nodes(i2,2)- nodes(i1,2)
length(i) = sqrt(dx*dx + dy*dy)
sinus(i) = dy/length(i)
cosinus(i) = dx/length(i)
```

```
fe(i) = e(i)*area(i)/length(i)
150 continue

!Com: Order of the system
order = 0
do i = 1, nnodes
do j = 1,2
if (displ(i,j) .ne. 0) then
order = order + 1
displ(i,j) = order
end if
end do
end do
!print*, 'Order =', order
!pause
!Com: Number of elements joining each node and their numbers
nelnode = 0

do i = 1, nnodes
n = 1
do j = 1, nelements
if ((elements(j,1).eq.i) .or. (elements(j,2).eq.i)) then
nelnode(i) = nelnode(i) + 1
elnode(i,n) = j
n = n + 1
end if
end do
end do
!Com: Make a table later (for printing)!2
!c do i = 1, nnodes
!c print*, i, nelnode(i),(elnode(i,j),j=1,nelnode(i))
!c end do
!******************

ALLOCATE(Kgl(order,order))
```

```
Kgl = 0.0

CALL elstiffness(nelements)

!COM: BUILD THE STIFFNESS MATRIX FOR THE WHOLE SYSTEM

DO 1000 IND = 1, nnodes
din1 = 0
din2 = 0
din3 = 0
din4 = 0
!*** MAIN NODE AND DIAGONAL end
IF(DISPL(IND,1) .NE. 0) THEN
din1 = displ(ind,1)
DO 200 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
Kgl(din1,din1)=Kgl(din1,din1)+kel(1,1,elnode(ind,iel))
if (displ(ind,2) .ne. 0) Kgl(din1+1,din1)=Kgl(din1+1,din1)+kel(2,1,elnode(ind,iel))
else
Kgl(din1,din1)=Kgl(din1,din1)+kel(3,3,elnode(ind,iel))
if (displ(ind,2) .ne. 0)Kgl(din1+1,din1)=Kgl(din1+1,din1)+kel(4,3,elnode(ind,iel))
end if
200 CONTINUE
END IF

IF(DISPL(IND,2) .NE. 0) THEN
din2 = displ(ind,2)

DO 300 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
Kgl(din2,din2)=Kgl(din2,din2)+kel(2,2,elnode(ind,iel))
if (displ(ind,1) .ne. 0)Kgl(din2-1,din2)=Kgl(din2-1,din2)+kel(1,2,elnode(ind,iel))
else
```

```
Kgl(din2,din2)=Kgl(din2,din2)+kel(4,4,elnode(ind,iel))
if (displ(ind,1) .ne. 0)Kgl(din2-1,din2)=Kgl(din2-1,din2)+kel(3,4,elnode(ind,iel))
end if
300 CONTINUE
END IF
!*** OPPOSITE ENDS OF ELEMENTS
DO 400 iel = 1, nelnode(ind)
if(elements(elnode(ind,iel),1) .eq. ind) then
nen = elements(elnode(ind,iel),2)
else
nen = elements(elnode(ind,iel),1)
end if

if(displ(nen,1) .ne. 0) then
din3 = displ(nen,1)
if(din1 .ne. 0) then
Kgl(din3,din1) = Kgl(din3,din1)+kel(3,1,elnode(ind,iel))
end if
if(din2 .ne. 0) then
Kgl(din3,din2) = Kgl(din3,din2)+kel(3,2,elnode(ind,iel))
end if
end if

if(displ(nen,2) .ne. 0) then
din4 = displ(nen,2)
if(din1 .ne. 0) then
Kgl(din4,din1) = Kgl(din4,din1)+kel(4,1,elnode(ind,iel))
end if
if(din2 .ne. 0) then
Kgl(din4,din2) = Kgl(din4,din2)+kel(4,2,elnode(ind,iel))
end if
end if
400 continue
1000 CONTINUE
!Com: Vector of the external forces
```

```
ip = 1
do i = 1,nnodes
if (displ(i,1) .ne.0) then
p(ip) = forces(i,1)
ip = ip+1
end if
if (displ(i,2) .ne.0) then
p(ip) = forces(i,2)
ip = ip+1
end if
end do
!Com: Print the vector of the external forces
!Com: Inverse the stiffness matrix
call rsudm(kgl,order,0,NER,DET,IED)

!Com: Global displacements
delta = 0.0
do i = 1, order
s = 0
do j = 1,order
s = s + Kgl(i,j) * p(j)
end do
delta(i) = s
end do

!Com: Local displacements and internal forces
eldelta = 0.0
do i = 1, nelements
if(displ(elements(i,1),1) .ne. 0)eldelta(i,1) = delta(displ(elements(i,1),1))

if(displ(elements(i,1),2) .ne. 0)eldelta(i,2) = delta(displ(elements(i,1),2))

if(displ(elements(i,2),1) .ne. 0)eldelta(i,3) = delta(displ(elements(i,2),1))
```

```
if(displ(elements(i,2),2) .ne. 0)eldelta(i,4) = delta(displ(elements(i,2),2))
end do

do i = 1, nelements
end do
!Com: Internal forces
CALL transform(nelements)
intforces = 0.0
do i = 1, nelements

intforces(i,1) = h(1,1,i)*eldelta(i,1)+h(1,2,i)*eldelta(i,2)+&
h(1,3,i)*eldelta(i,3)+h(1,4,i)*eldelta(i,4)

intforces(i,3) = h(3,1,i)*eldelta(i,1)+h(3,2,i)*eldelta(i,2)+&
h(3,3,i)*eldelta(i,3)+h(3,4,i)*eldelta(i,4)
end do

RETURN
END
!*———————————————————————————————————————-*
subroutine elstiffness(nel)
implicit none
integer i
integer*2 nel
real*8 f,s,c
real*8 fe(55),sinus(55),cosinus(55)
real*8 k(4,4,55)
common/elchar/fe,sinus,cosinus
common/elmatrix/k

k = 0.0
do i = 1, nel
f = fe(i)
c = cosinus(i)
```

```
s = sinus(i)
k(1,1,i) = f*c*c
k(1,2,i) = f*s*c
k(1,3,i) = -f*c*c
k(1,4,i) = -f*s*c

k(2,1,i) = f*s*c
k(2,2,i) = f*s*s
k(2,3,i) = -f*s*c
k(2,4,i) = -f*s*s

k(3,1,i) = -f*c*c
k(3,2,i) = -f*s*c
k(3,3,i) = f*c*c
k(3,4,i) = f*s*c

k(4,1,i) = -f*s*c
k(4,2,i) = -f*s*s
k(4,3,i) = f*s*c
k(4,4,i) = f*s*s
end do
return
end
!*****************************************************************
!*** TRANSFORMATION MATRIX FOR CALCULATION OF THE INTERNAL FORCES
***
!*****************************************************************
subroutine transform(nel)
implicit none
integer i
integer*2 nel
real*8 f,s,c
real*8 fe(55),sinus(55),cosinus(55)
real*8 h(4,4,55)
common/elchar/fe,sinus,cosinus
```

```
common/tmatrix/h
h = 0.0
do i = 1, nel
f = fe(i)
c = cosinus(i)
s = sinus(i)
h(1,1,i) = f*c
h(1,2,i) = f*s
h(1,3,i) = -f*c
h(1,4,i) = -f*s

h(2,1,i) = 0.0
h(2,2,i) = 0.0
h(2,3,i) = 0.0
h(2,4,i) = 0.0

h(3,1,i) = -f*c
h(3,2,i) = -f*s
h(3,3,i) = f*c
h(3,4,i) = f*s

h(4,1,i) = 0.0
h(4,2,i) = 0.0
h(4,3,i) = 0.0
h(4,4,i) = 0.0
end do
return
end
!*----------------------------------------------------------*
SUBROUTINE RSUDM(A,N1,N2,NER,DET,IED)
DIMENSION IND(75)
REAL*8 A(1),PT,SW,DET
DETER=1.0
IED=0
N=N1
```

```
MAT=N+N2
IM=N1
NM=N-1
IVC=1-IM
DO 11 MA=1,N
PT=0.0D+0
IVC=IVC+IM
IV1=IVC+MA-1
IV2=IVC+NM
DO 2 I=IV1,IV2
IF(ABS(A(I))-ABS(PT))2,2,1
1 PT=A(I)
LP=I
2 CONTINUE
IF(PT)3,15,3
3 IC=LP-IVC+1
IND(MA)=IC
IF(IC-MA)6,6,4
4 DETER=-DETER
IC=IC-IM
I3=MA-IM
DO 5 I=1,MAT
IC=IC+IM
I3=I3+IM
SW=A(I3)
A(I3)=A(IC)
5 A(IC)=SW
6 DETER=DETER*PT
IE=INT(ALOG10(ABS(DETER)+1.E-20)+20)-20
DETER=DETER/10.**IE
IED=IED+IE
PT=1./PT
I3=IVC+NM
DO 7 I=IVC,I3
7 A(I)=-A(I)*PT
A(IV1)=PT
```

```
I1=MA-IM
IC=1-IM
DO 10 I=1,MAT
IC=IC+IM
I1=I1+IM
IF(I-MA)8,10,8
8 JC=IC+NM
SW=A(I1)
I3=IVC-1
DO 9 I2=IC,JC
I3=I3+1
9 A(I2)=A(I2)+SW*A(I3)
A(I1)=SW*PT
10 CONTINUE
11 CONTINUE
DO 14 I1=1,N
MA=N+1-I1
LP=IND(MA)
IF(LP-MA)12,14,12
12 IC=(LP-1)*IM+1
JC=IC+NM
IVC=(MA-1)*IM+1-IC
DO 13 I2=IC,JC
I3=I2+IVC
SW=A(I2)
A(I2)=A(I3)
13 A(I3)=SW
14 CONTINUE
DET=DETER
NER=0
RETURN
15 NER=-1
DET=DETER
WRITE(*,17)MA
17 FORMAT(' ',I4)
RETURN
```

END

Appendix C - Finite element analysis stress results problem 1 and 2 results

PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number			Sheet
	Job Title: Sizing, shape and topology optimization problem 1			
	Client			
	Calcs by	Checked by	Date: 20 September 2018	

Member Design for Axial Force - Strut Ver W2.6.01

```
==============================================================================
Task Title:        Task 1
Frame input file: C:\Users\zolie\Dropbox\Masters\Work\1 Truss Opt 2D.A03
Data read from:    C:\Prokon\data\Project1\Sf.out
Code of practice: SABS 0162 - 1993
Design approach:  Select lightest sections
Profile:           Angles (Equal leg)
```

Design parameters	
Kv factor	0.85
Kx factor	0.85
Ky factor	1.00
Ane/Ag	1.00
Fy MPa	300.00
Fu MPa	450.00

Maximum L/r ratios		
L Case	Compr.	Tension
1	200	300

```
Element       Length    L.C. Force  L/R  Crit     Section    Pc      σc Result
               (m)           (kN)        Axis              (MPa)  (MPa)
------------------------------------------------------------------------------
GROUP H          Angles (Equal leg)

1-2           4.000   1      -57.66  288   V        60x60x4 270.0  122.4   OK
2-3           4.000   1       28.84  191   V        90x90x6  46.4   27.2   OK
3-4           4.000   1       57.66  193   V        90x90x8  45.4   41.5   OK
4-5           4.000   1       28.84  191   V        90x90x6  46.4   27.2   OK
5-6           4.000   1      -57.66  288   V        60x60x4 270.0  122.4   OK
7-8           4.000   1     -114.70  288   V        60x60x4 270.0  243.5   OK
8-9           4.000   1     -172.46  291   V        60x60x6 270.0  249.6   OK
9-10          4.000   1     -172.46  291   V        60x60x6 270.0  249.6   OK
10-11         4.000   1     -114.70  288   V        60x60x4 270.0  243.5   OK

Group mass = 212.8 kg

GROUP B          Angles (Equal leg)

6-11          2.500   1      -71.82  239   V        45x45x3 270.0  268.0   OK
1-7           2.500   1      -71.82  239   V        45x45x3 270.0  268.0   OK
2-7           2.500   1       71.65  135   V        80x80x6  83.2   76.6   OK
2-8           2.500   1      -36.46  271   V        40x40x3 270.0  155.1   OK
3-8           2.500   1       35.77  155   V        70x70x6  67.8   44.0   OK
3-9           2.500   1       -0.31  271   V        40x40x3 270.0    1.3   OK
4-9           2.500   1       -0.31  271   V        40x40x3 270.0    1.3   OK
4-10          2.500   1       35.77  155   V        70x70x6  67.8   44.0   OK
5-10          2.500   1      -36.46  271   V        40x40x3 270.0  155.1   OK
5-11          2.500   1       71.65  135   V        80x80x6  83.2   76.6   OK

Group mass = 97.6 kg

Total mass for task = 310.3 kg
```

PROKON Software Consultants (Pty) Ltd Internet: http://www.prokon.com E-Mail : mail@prokon.com	Job Number		Sheet
	Job Title Sizing, shape and topology optimization problem 2		
	Client		
	Calcs by	Checked by	Date 20 September 2018

Member Design for Axial Force - Strut Ver W2.6.01

```
===========================================================================
Task Title:        Task 1
Frame input file: C:\Users\zolie\Dropbox\Masters\Work\2 Truss Opt 2D - Copy.A03
Data read from:    C:\Prokon\data\Project1\Sf.out
Code of practice: SABS 0162 - 1993
Design approach:   Select lightest sections
Profile:           Angles (Equal leg)
```

Design parameters	
Kv factor	0.85
Kx factor	0.85
Ky factor	1.00
Ane/Ag	1.00
Fy MPa	300.00
Fu MPa	450.00

Maximum L/r ratios		
L Case	Compr.	Tension
1	200	300

```
Element       Length    L.C. Force  L/R  Crit     Section    Pc       σc Result
               (m)           (kN)        Axis               (MPa)  (MPa)
-----------------------------------------------------------------------------
GROUP H          Angles (Equal leg)

2-3           4.000   1      181.60  145   V    120x120x12  75.3   66.0   OK
3-4           4.000   1      242.43  114   V    150x150x10 105.7   82.7   OK
4-5           4.000   1      181.60  145   V    120x120x12  75.3   66.0   OK
10-12         2.000   1       60.78  124   V       70x70x6  94.3   74.8   OK
11-12         2.000   1       61.17  124   V       70x70x6  94.3   75.2   OK
9-13          2.000   1      -61.02  217   V       40x40x3 270.0  259.7   OK
10-13         2.000   1      -60.93  217   V       40x40x3 270.0  259.3   OK
8-14          2.000   1      -60.93  217   V       40x40x3 270.0  259.3   OK
9-14          2.000   1      -61.02  217   V       40x40x3 270.0  259.7   OK
7-15          2.000   1       61.17  124   V       70x70x6  94.3   75.2   OK
8-15          2.000   1       60.78  124   V       70x70x6  94.3   74.8   OK

Group mass = 330.5 kg

GROUP B          Angles (Equal leg)

2-7           2.500   1      150.46  108   V     100x100x8 114.0   97.1   OK
2-8           2.500   1      -76.24  214   V       50x50x3 270.0  255.8   OK
3-8           2.500   1       75.95  135   V       80x80x6  83.2   81.2   OK
3-9           2.500   1       -0.03  271   V       40x40x3 270.0    0.1   OK
4-9           2.500   1       -0.03  271   V       40x40x3 270.0    0.1   OK
4-10          2.500   1       75.95  135   V       80x80x6  83.2   81.2   OK
5-10          2.500   1      -76.24  214   V       50x50x3 270.0  255.8   OK
5-11          2.500   1      150.46  108   V     100x100x8 114.0   97.1   OK

Group mass = 118.5 kg

GROUP V          Angles (Equal leg)

2-15          1.500   1        0.65  163   V       40x40x3  63.0    2.8   OK
3-14          1.500   1       -0.24  264   V       25x25x3 270.0    1.7   OK
4-13          1.500   1       -0.24  264   V       25x25x3 270.0    1.7   OK
5-12          1.500   1        0.65  163   V       40x40x3  63.0    2.8   OK

Group mass = 8.9 kg

Total mass for task = 457.9 kg
```