2014

Harmony Search Optimization and Damage Tolerance of Structural Systems

R. Bryan Peiffer
Lehigh University

Harmony Search Optimization and Damage Tolerance of Structural Systems

by

R. Bryan Peiffer

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Structural Engineering

Department of Civil and Environmental Engineering

Lehigh University

Bethlehem, Pennsylvania

May, 2014

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

____

Date

____

Thesis Advisor

____

Chairperson of Department

Acknowledgements

I would like to express my sincere appreciation to my thesis advisor, Dan M. Frangopol, for his guidance and for the opportunity to work under him on this project. I would also like to express my gratitude for his generosity and support from start to finish. Also, thank you to my fellow student researchers especially Dr. Frangopol's research group for their help throughout.

Finally, I would like to thank my family, Scott, Tracy and Bradley Peiffer for their support and guidance in all my endeavors. I would also like to extend my heartiest gratitude to my fiancée, Carrie Landis, for her encouragement and advice. This thesis would not have been possible without her love and support.

The opinions, findings, and conclusions expressed in this thesis are those of the author's and do not necessarily reflect the views of the individuals acknowledged above.

Table of Contents

Tables

Figures

Abstract

The aim of this thesis is to develop a MATLAB computer model for optimum design of steel structures via harmony search (HS) algorithm. The objective of the optimization routine is to provide a minimum weight structure while satisfying prescribed constraints such as strength and displacement limitations. The HS algorithm is a relatively new optimization method that has shown promise when adapted to structural optimization problems. Unlike other optimization routines, limited research has been presented incorporating this mathematical model. The author of this thesis decided to test the applications of the HS algorithm in structural engineering problems.

The HS algorithm is a meta-heuristic search method recently developed and adapted to optimization problems. It uses a stochastic derivative, which utilizes the experiences of musicians in Jazz improvisation to find optimal solutions. It differs from classical calculus bases optimization techniques that require gradient information by giving each decision variable a probability of selection.

Three examples have been provided showing the capabilities of the HS for least weight optimization of truss and frame structures. Both continuous and discrete optimization routines are present in this thesis. In the discrete optimization routine, standard steel shapes were used in accordance to the American Institute of Steel Construction (AISC) shape database. Strength and displacement constraints from the 2005 AISC load and resistance factor design specification were used to design.

In addition to least weight optimization, damage tolerance optimization of structural systems was also considered. Least weight optimization was performed accounting for probable future damage to the structure. A general mathematical model for damage tolerant optimization is presented. This method is based on serviceability, ultimate and residual capacity requirements.

It is shown that the HS algorithm can be a powerful tool for optimization problems, particularly structural engineering optimization problems. It has the ability to handle complex problems that would be very challenging to solve by traditional methods. It also, has been shown to be competitive with several other well know meta-heuristic optimization methods.

Chapter 1: Introduction

1.1 General

One of the most catastrophic structural failures in modern history has been the collapse of the World Trade Center towers in 2001. Most individuals do not think twice about the structural systems they encounter on a daily basis. They typically view structural assemblages such as the twin towers as invulnerable structures incapable of collapse. However, the collapse of the twin towers, among other failures, shed light to the public that structures are vulnerable.

The collapse of the twin towers was found to be from a pancaking action that resulted in the towers crushing themselves completely after the planes struck. The term progressive collapse was a widely used term in the structural engineering community after the events. While progressive collapse is not a new phenomenon, it has been a source of increased interest due to these large-scale failures. Because of these failures, increased attention has been focused upon the concepts of structural robustness, reliability, damage and redundancy. The importance of design procedures that provide redundant and robust structures is widely recognized to reduce further failures.

In its most simplistic form, a structure is "any assemblage of materials which is intended to sustain loads" [1]. Typically, if an engineering structure fails it will result in loss of life or at the very least significant injury. For this reason, a great deal of effort and work goes into the design of a structure so it can properly sustain prescribed loadings. However, failures still occur. Structural failure can be induced by a wide variety of events such as, deterioration of the structure over time (corrosion), sudden impact damage (blast), natural events (earthquake, tornado, and typhoon) and improper initial design. [1]

The concept of robustness, redundancy and static indeterminacy are key in many design philosophies and widely recognized as an important aspect in structural engineering. However, finding a consistent definition of the redundancy and robustness can be challenging. For example, the definition of redundancy may be provided in terms of collapse load, number of plastic hinges, the probability of system failure, etc. Others tend to use the term redundancy and static indeterminacy interchangeably. It has been that the degree of static indeterminacy does not correlate to structural redundancy. Structures with lower degree of static indeterminacy can often times have greater redundancy than their higher degree counter parts. This is due to the fact the redundancy relies on a wide variety of factors like, member size, material properties, structural topology, loading sequence and applied loading. Generally, redundancy is the ability of a structural system to redistribute loads among members that cannot be sustained by another member due to damage. Whereas, robustness is the ability of a structural system to sustain a specific amount of damage not disproportionate to the cause of the damage itself. [2]

In this thesis, the effects of prescribed damage scenarios on several structural systems are investigated. The structural systems investigated are then damaged from progressive deterioration of member material properties. The amount of damage is prescribed by a damage index associated with specific patterns of cross-sectional deterioration. Once the damage is defined, structural performance will be evaluated and compared to the original intact structure.

1.2 Problem statement

The methods of finding optimum design solutions for structural systems can be very cumbersome to solve by hand, due to the large number of design variables present in the problem. The designer must decide which parameters are important for their current problem. Typically, in structural optimization problems, minimum weight is the desired search criterion. Optimal design of structural systems are normally limited by several constraints such as choice of material, feasible strength, displacements, deflection, size constraints, load cases, support conditions, and beam-column behavior. This research utilizes Harmony Search optimization algorithm for optimizing structural systems member sizes with both discrete and continuous design variables.

1.3 Objectives

The depth of this thesis is to develop a computer model that utilizes the harmony search algorithm for optimization of steel structures. Strength constraints from the AISC Load and Resistant Factor Design specification will be used along with, displacement, deflection and member size constraints. This model will then be adapted for damage tolerant optimization. Lastly, a brief section will discuss the correlation between damage and reliability.

1.4 Methodology

In order to achieve the objectives presented in section 1.3, the following approach was taken:

1. Perform a literature review of previous research related to optimization and damage tolerance
2. Develop a suitable harmony search algorithm code
3. Test the developed code to benchmark examples
4. Implement damage tolerance constraints into the harmony search routine
5. Compare optimization results
6. Draw conclusions from the results

1.5 Organization of thesis

Chapter 2 presents the literature review of the AISC code, beam-column connections, behavior of semi-rigid connections, optimization and uncertainty in engineering. Chapter 3 presents an explanation of optimization and various techniques followed by a detailed overview of harmony search optimization. Chapter 4 covers the topic of risk and reliability in engineering. Chapter 5 presents the modeling of structural systems. Chapter 6 presents design examples of the topics covered in previous chapters. Chapter 7 presents a conclusion and recommendations for future work.

Chapter 2: Literature Review

2.1 Introduction

This chapter discusses background information that is relevant to the current study. As discussed in Chapter 1, the goal of this research is to provide a computer model capable of optimizing steel structures. The information presented in this chapter provides a foundation for achieving this goal. The following sections discuss the AISC code specifications, connections, optimization, harmony search algorithm, reliability and structural damage.

2.2 AISC-LRFD specification of connections

Connections are the components that hold a steel structure together. Typically, structural connections are bolted or welded together in different configurations depending on the system. According to AISC, there are several types of steel connections: simple framing (unrestrained), rigid-frame (fully restrained), semi-rigid framing (partially restrained) and truss connections. [3]

2.2.1 Truss connections

In truss connections, only axial forces are transferred through the connection. They allow a full range of rotation and are considered one of the most simplistic connections. [3]

2.2.2 Simple connections

A simple connection, also known as a shear connection, can transmit shear and a negligible moment force through the connection. The connection allows unrestrained relative rotation between the framing elements and shall have sufficient rotation capacity to accommodate the required rotation determined by analysis. Inelastic rotation of the connection is acceptable. [3]

2.2.3 Moment connections

Moment connections, unlike simple connections are capable of transmitting moment forces across the connection. The LRFD specification for structural steel buildings classifies two types of moment connections: fully restrained (FR) and partially restrained (PR). When connection restraint is considered strength, stiffness and ductility characteristics must be included in the analysis and design of the structural system. [3]

1. Fully-Restrained (FR) Moment Connections transfer the moment force while allowing a negligible amount of rotation between members. In analysis, this may be assumed as zero rotation between members or fully rigid connections. The connection shall have sufficient strength and stiffness to maintain the original angle between members at the strength limit states. They are particularly useful when a framing system needs to provide more flexural resistance and reduce lateral deflections. [3]
2. Partially-Restrained (PR) Moment Connections transfer the moment force while allowing rotation between members. They have insufficient rigidity to maintain the original angle between the column and beam. In analysis, force-deformation response characteristics of the connection shall be included. [3]

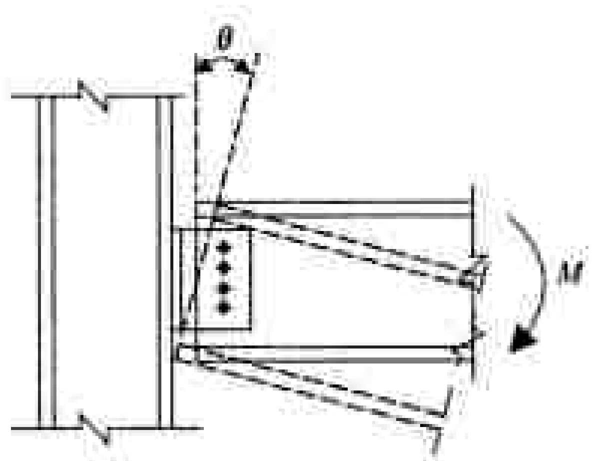


Figure 1 - Connection Rotation [6]

2.3 Types of connections

2.3.1 Single web angle

The single web angle connection is shown in figure 2 and consists of an angle connecting the web of the beam to the column flange. Number of bolts, angle thickness, web thickness and column thickness, influence the connections behavior. [4]

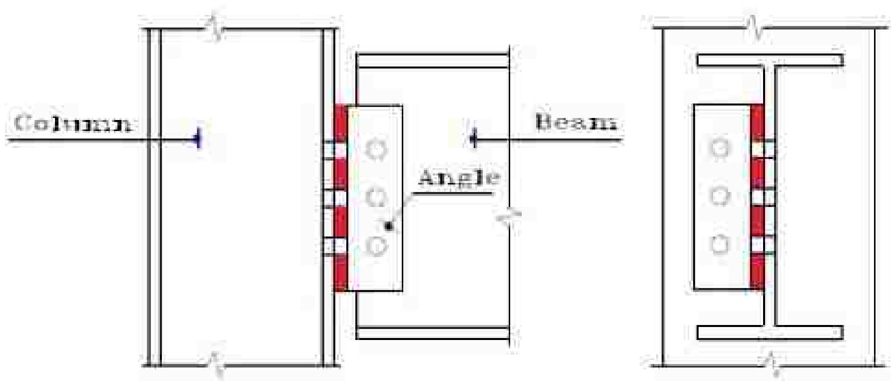


Figure 2 - Single web angle [6]

2.3.2 Double web angle

The double web angle connection shown in figure 3 and consists of two angles connecting the web of the beam to the column flange. Number of bolts, angle thickness, web thickness and column thickness, influence the connections behavior. [4]

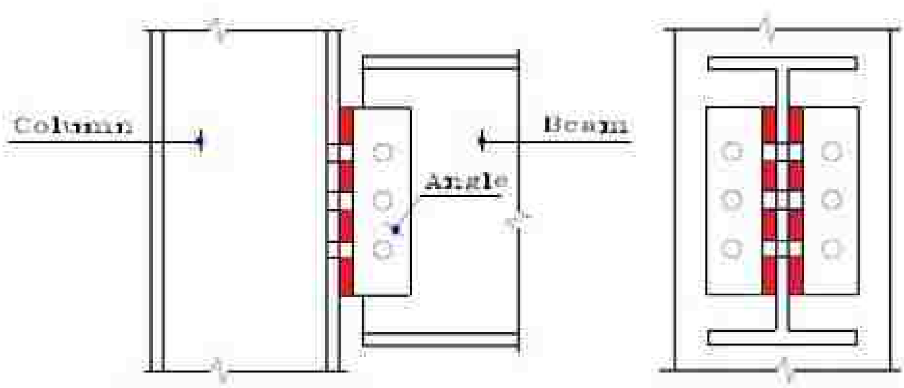


Figure 3 - Double web angle [6]

2.3.3 Header plate

The header plate connection is shown in figure 4 and consists of an end plate that's length is less than the depth of the beam, welded to the beam and bolted to the column. The behavior of this connection is influenced by plate thickness, plate depth and beam-web thickness. This connection performs similar to the double web angle connection. [4]

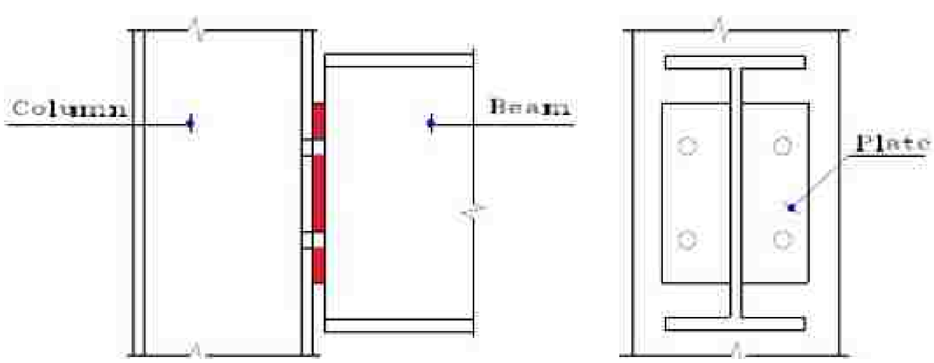


Figure 4 - Header plate [6]

2.3.4 Top and seat angles

The top and seat angle shown in figure 5 consists of two angles that are connected to the top and bottom flanges of the beam then connected to the flange of the column. The top angle is used for lateral stability and is not considered to carry gravity loading. The bottom or seat angle only transfers vertical loading and provides an insignificant amount of moment restraint. The number of bolts and angle thickness influence behavior. [4]

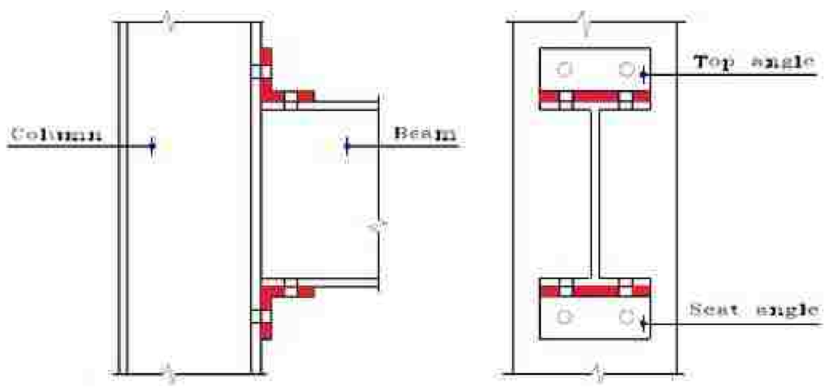


Figure 5 - Top and set angle [6]

2.3.5 Top and seat angles with double web angles

The top and seat angle with double web angles is a combination of the top and seat angle connection and double web angle connection as seen in figure 6. Depth and thickness of the angles, column flange or web thickness and gauge distance of bolts in the vertical angles govern this connections behavior. Plate thickness, column flange thickness and moment arm for column flange bolts influence the behavior of this connection. [4]

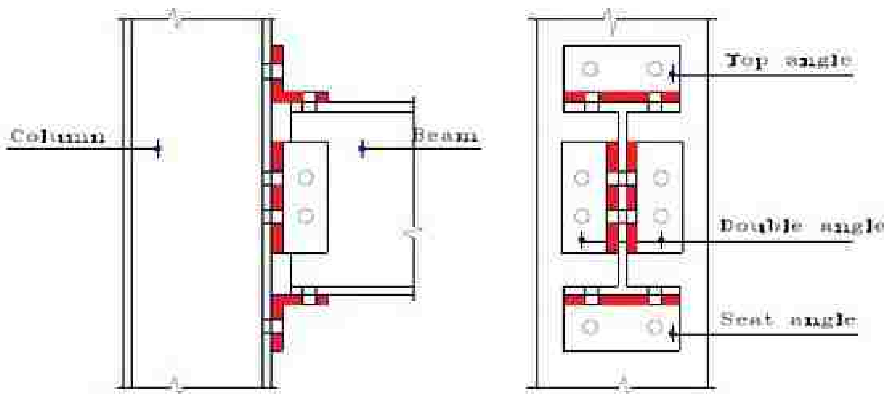


Figure 6 - Top and seat with double web angle [6]

2.3.6 Extended end plate without column stiffeners

The extended end plate connection shown in figure 7 is comprised of a plate welded to the end of the beam then fastened to the column. The plate extends past both the tension and compression flanges of the beam. Plate thickness, column flange thickness and moment arm for column flange bolts influence connection behavior. [4]

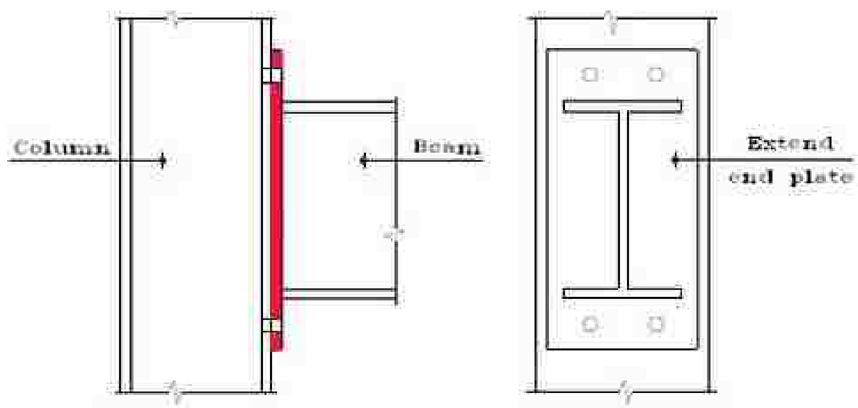


Figure 7 - Extended end plate [6]

2.3.7 Extended end plate with column stiffeners

The extended end plate with column stiffeners seen in figure 8 is the same configuration as the previous connection only with the addition of column stiffeners. [4]

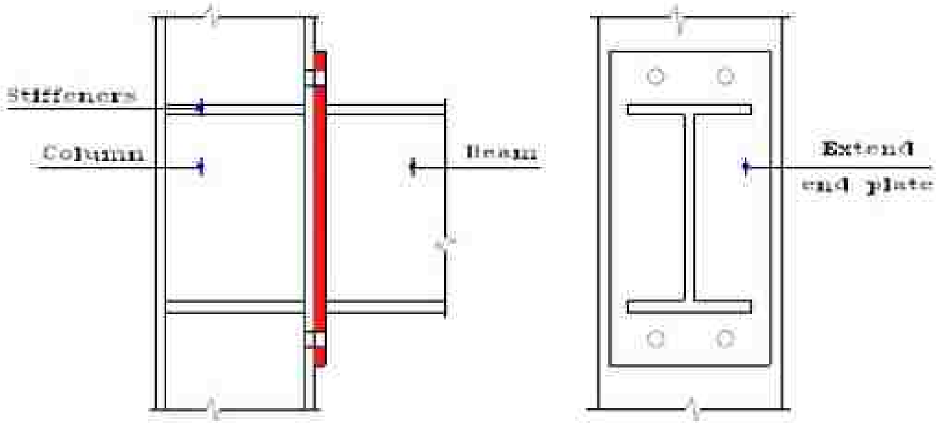


Figure 8 - Extended end plate with stiffeners [6]

2.3.8 T-Stub

The T-stub connection shown in figure 9 is similar to the top and seat connection except tee sections replaces the angles. This configuration provides a very rigid joint. T-stub thickness and width influence the behavior of this connection. [4]

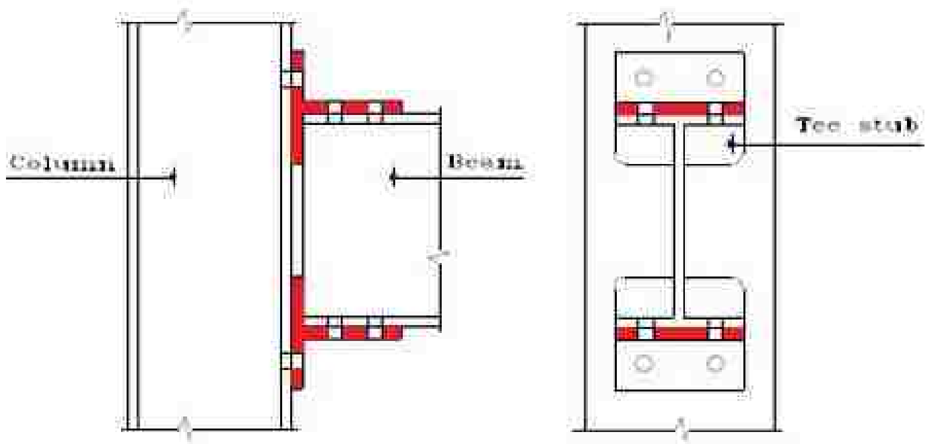


Figure 9 - T-stub [6]

2.4 Behavior and modeling of semi-rigid connections

A beam-column connection is subjected to axial and shear forces along with bending and torsion moments. When working with planar frames, the torsion moments are often neglected. Axial and shear deformations are typically neglected because they are small relative to bending deformation of most connections. This leaves only the rotational deformation of the connection to be considered in semi-rigid connection framing. A semi-rigid connection is able to rotate through an angle θ_r due to an applied moment M previously shown in figure 1. The angle θ_r is the relative rotation of the beam and the column taken at the connection.[4]

Several connections were experimentally tested by Frye and Morris to show the rotation-moment relationship. The connections moment-rotation behavior is non-linear in nature and falls between fully fixed and ideally pinned. The relationship of several types of connections is shown in figure 10. [5]

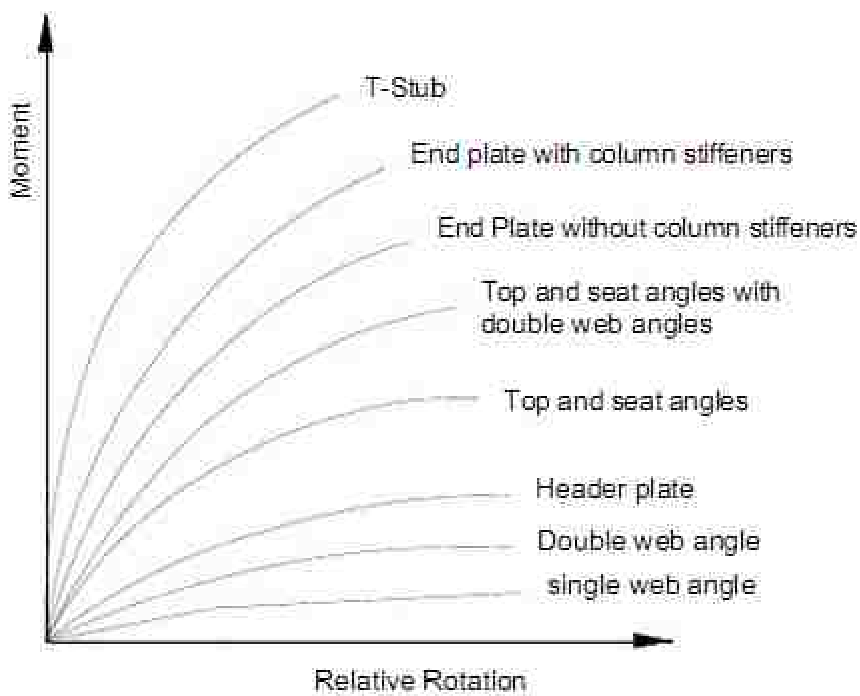


Figure 10 - Moment rotation curves [5]

2.5 Mathematical modeling of semi-rigid connections

The most precise and dependable knowledge of beam-column connection behavior is found through experimental testing, but these test can be very complex and expensive and are not practical for design practice. To take in account connection behavior in structural analysis, connections are typically represented by mathematical models representing rotation-moment relationships. The non-linear behavior of a connection is difficult to represent exactly by mathematical representation and the models used are approximates due to simplifications.

2.5.1 Linear model

The most simplistic connection model is single-stiffness linear model proposed by Batho, Rathbun and Baker with the following expression: [6]

$$M = R * \emptyset_r \tag{1}$$

where M represents the connection moment and R and θ represent the stiffness and rotation respectively.

2.5.2 Polynomial model

The linear model is an over simplification of connection behavior and does not represent the true behavior of a connection. Polynomial models were proposed to provide a more accurate representation of connection behavior. Frye and Morris used an odd power polynomial model to represent the moment-rotation curve as follows: [6]

$$\emptyset = C_1(KM) + C_2(KM)^3 + C_3(KM)^5 \tag{2}$$

where θ is the connection rotation and M is the moment acting on the connection. The variable K is the standardization factor determined by the connection type and geometry and C1, C2 and C3 are curve-fitting constants obtained by using the least squares method. These constants for various connection types can be seen in table 1. [5]

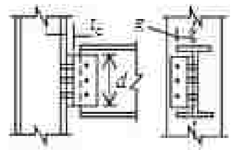
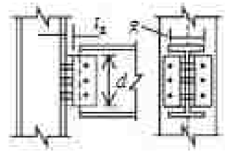
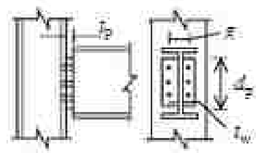
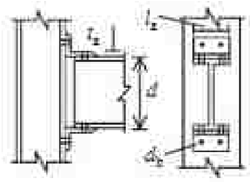
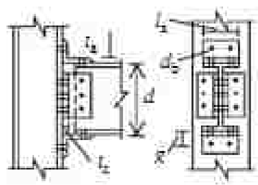
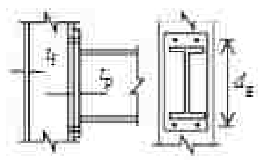
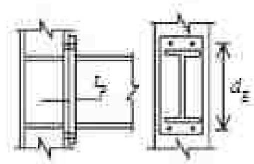
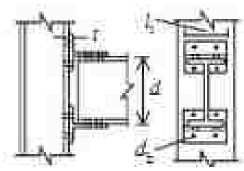
Single web-angle connection		$C_1 = 1.67 \times 10^{5}$ $C_2 = 8.56 \times 10^{-2}$ $C_3 = 1.35 \times 10^{-3}$ $K = d_a^{-2.4} t_c^{-1.81} g^{0.15}$
Double web-angle connection		$C_1 = 1.45 \times 10^{-1}$ $C_2 = 6.79 \times 10^{1}$ $C_3 = 4.09 \times 10^{5}$ $K = d_a^{-2.4} t_c^{-1.81} g^{0.15}$
Header plate connection		$C_1 = 6.14 \times 10^{-2}$ $C_2 = 1.08 \times 10^{-3}$ $C_3 = 6.05 \times 10^{-3}$ $K = t_p^{-1.6} g^{1.6} d_p^{-2.3} t_w^{-0.5}$
Top- and seat-angle connections		$C_1 = 2.59 \times 10^{-1}$ $C_2 = 2.88 \times 10^{2}$ $C_3 = 3.31 \times 10^{4}$ $K = d^{-1.5} t_a^{-0.5} l_a^{-0.7} d_b^{-1.1}$
Top- and seat-angle with double web-angle connection		$C_1 = 1.50 \times 10^{-3}$ $C_2 = 5.60 \times 10^{-3}$ $C_3 = 4.35 \times 10^{-3}$ $K = d^{-1.287} t_a^{-1.128} t_c^{-0.415} l_a^{-0.694} (g - 0.5 d_b)^{1.35}$
Extend end-plate connection without column stiffeners		$C_1 = 8.91 \times 10^{-1}$ $C_2 = -1.20 \times 10^{4}$ $C_3 = 1.75 \times 10^{5}$ $K = d_g^{-2.4} t_p^{-0.4} t_f^{-1.5}$
Extended end-plate connection with column stiffeners		$C_1 = 2.60 \times 10^{-1}$ $C_2 = 5.36 \times 10^{2}$ $C_3 = 1.31 \times 10^{7}$ $K = d_g^{-2.4} t_p^{-0.6}$
T-stub connection		$C_1 = 6.42 \times 10^{-2}$ $C_2 = 1.77 \times 10^{2}$ $C_3 = -2.03 \times 10^{4}$ $K = d^{-1.5} t^{-0.5} l_t^{-0.7} d_b^{-1.1}$

Table 1 - Moment rotation curve fitting equations [6]

2.5.4 Three-parameter power model

Chen and Kishi adopted the power model to represent the rotation-moment relationship for beam-column connections as shown:

$$M = \frac{R_{ki} * \emptyset}{\{1+[\emptyset/\emptyset_0]^n\}^{1/n}} \quad 3$$

$$R_{ki} = \frac{dM}{d\emptyset} = \frac{R_{ki}}{\{1+[\emptyset/\emptyset_0]^n\}^{(n+1)/n}} \quad 4$$

$$\emptyset = \frac{M}{R_{ki}(1+[\mathrm{M}/\mathrm{M_u}]^n\}^{1/n}} \quad 5$$

where R_{ki} representes the initial stiffness, M_u is the ultimate moment capacity and n is shape parameter.[6]

2.6 Optimization of steel structure

As today's world continues to increase in population with world resources declining the need for economical designs is at the forefront for structural engineers. More structures are needed for living and production than ever before which is why these structures need to be designed using the minimum amount of material available. Due to this need, optimization algorithms prove to be a useful tool when designing steel structures. These algorithms can be implemented while staying within design constraints specified from steel design code and search for a minimum weight or cost structure. Formulation of these optimization algorithms is through mathematical models with discrete design variables. The reason for discrete design variables is so the design model can adopt standard steel sections commonly used in practice.

Due to the complexity of structural optimization problems, heuristic search optimization methods have been the preferred choice for designers. Genetic algorithms, simulated annealing and ant colony optimization are some of the more popular heuristic search methods used in present optimization problems. These methods are easily adaptable to structural engineering optimization problems.

Several papers have focused on the optimization of steel structures.

Pezeshk et al.., (2000) [7], researched the design of nonlinear framed structures using genetic optimization. The paper presented a genetic algorithm approach for optimum design of 2D frames using discrete structural elements. The designs were in compliance with the AISC-LRFD (1994) code. Both linear and geometrically nonlinear analysis were performed to learn how P-Δ effects influenced optimal designs. It was concluded that the P-Δ effects did not significantly influence the optimal designs, but in some cases, it could yield a better design. In addition, it was found that the proposed optimization approach was effective optimization technique.

Hayalioglu et al.., (2001) [8], researched optimum load and resistance factor design of steel space frames using genetic algorithm. The paper presented a genetic algorithm to design moment-resisting space frames subjected to AISC-LRFD specifications for minimum weight. They utilized standard steel sections from AISC wide-flange (W) shapes. The proposed frame was subjected to wind loading in accordance to the Uniform Building Code (UBC). Comparisons between AISC-ASD and AISC-LRFD designs were made and showed that the former code resulted in lighter structures for the presented examples.

Hayalioglu and Degertekin (2005) [9], researched minimum cost design of steel frames with semi-ridged connections and column bases via genetic optimization. The optimization algorithm obtained the minimum total cost which comprised of total member and connection costs by selecting suitable sections from the AISC wide-flange (W) shapes. Displacement, stress and size constraints in accordance to the AISC-LRFD code were imposed on the frame. Comparisons were made between AISC-ASD and AISC-LRFD and the former code provided lighter structures. They also compared semi-rigid connections to rigid connections and found that reducing connection stiffness caused an increase in both frame cost and sway. The reason for these increases is the more flexible frame the larger the displacements which was compensated by increasing the member size to stay within code constraints.

Lee and Geem (2005) [10], presented a structural optimization method based on harmony search meta-heuristic algorithm. The algorithm was conceptualized using the musical process of Jazz musicians searching for a perfect stat of harmony. The advantage of HS is that unlike other optimization methods it does not require initial values and uses a random search routine instead of a gradient search. Several structural truss examples were present in the study to show the effectiveness and robustness of the new approach. The findings showed that the HS can be a powerful search and optimization method for solving structural engineering problems.

Saka, (2008) [11], researched optimum design of steel sway frames using harmony search algorithm to British Standard BS5950. The optimum design algorithm developed imposed behavior and performance constraints in accordance to BS5950. The optimization routine used standard sections from the Universal beam and column sections of the Britich Code. Optimization results obtained from the harmony search algorithm were compared to genetic algorithms and produced lighter results.

2.7 Harmony search algorithm in structural engineering

Harmony search (HS) is a relatively new meta-heuristic search algorithm developed by Geem et al. [10]. The original purpose for the algorithm was for solving combinatorial optimization problems in applied mathematics, but it can be adapted to a wide variety of optimization problems. HS has been applied to a range of civil and structural engineering problems such as the optimization of trusses, frames, dams, etc. [12]

2.8 Uncertainty and damage in structural engineering

In its most simplistic form, a structure has been defined as "any assemblage of materials which is intended to sustain loads". Typically, if an engineering structure fails it will result in loss of life or at the very least significant injury. For this reason, a great deal of effort and work goes into the design of a structure so it can properly sustain prescribed loadings. However, failures still occur. Structural failure can be induced by a wide variety of events such as, deterioration of the structure over time (corrosion), sudden impact damage (blast), natural events (earthquake, tornado, and typhoon) and improper initial design. [1]

The concept of robustness, redundancy and static indeterminacy are key in many design philosophies and widely recognized as an important aspect in structural engineering. However, finding a consistent definition of the redundancy and robustness can be challenging. For example, the definition of redundancy may be provided in terms of collapse load, number of plastic hinges, the probability of system failure, etc. Others tend to use the term redundancy and static indeterminacy interchangeably. It has been presented that the degree of static indeterminacy does not correlate to structural redundancy. Structures with lower degree of static indeterminacy can often times have greater redundancy than their higher degree counter parts. This is due to the fact the redundancy relies on a wide variety of factors like, member size, material properties, structural topology, loading sequence and applied loading. Generally, redundancy is the ability of a structural system to redistribute loads among members that cannot be sustained by another member due to damage. Whereas, robustness is the ability of a structural system to sustain a specific amount of damage not disproportionate to the cause of the damage itself. [13]

All aspects of life come with uncertainty and the same holds true for all sectors of engineering. Structural engineers make many decisions during the design and construction of structures. Many of these decisions are made with uncertainty, but not often considered due to the uncertainty being accounted for in design codes. Uncertainties arise in many aspects of structural engineering in the form of nominal capacities, resistance factors, design loads and load factors. Allowable stress design utilizes a safety factor to handle these uncertainties whereas load and resistance factor design has multiple factors. However, not all uncertainty can be accounted for in the design code. Structural engineers have to be aware of the uncertainty present in their calculations and be able to account for it accordingly.

2.10 Concluding remarks

Based on the study of several papers, it was concluded that the new meta-heuristics algorithm harmony search proved to be a powerful tool for optimization of structural systems. Many of the publications reviewed showed the Frye-Morris polynomial model provides an accurate representation for moment-rotation connection behavior. Lastly, it was found that the extended end plate connection is a popular connection used in steel structures and optimization routines.

Chapter 3: Optimization

3.1 Introduction

As previously discussed in chapters 1 and 2, mathematical optimization, simply is the process of making something the best it can possibly be. Traditionally, optimization is performed using calculus-based methods such as numerical linear and nonlinear programming methods. These methods require substantial gradient information and can be sensitive to starting points. They are ideal for obtaining global optimum points in relatively simple models. However, real-world engineering problems tend to be very complex in nature and prove hard to solve using traditional methods. More than one optimal point may be present in these complex problems and the results would be very sensitive to the selected starting point. The optimal solution may not necessarily be the global optimum for the problem. In addition to these issues, objective functions and constraints can have multiple or sharp peaks resulting in difficult or unstable gradient computations. The drawbacks of traditional techniques led to the need of other optimization methods. Researchers utilized meta-heuristic algorithms based on simulations to solve these complex problems. These algorithms are typically based around natural phenomena and each have a unique set of rules and randomness intrinsically built in. The following sections provide a brief overview of some of the more popular meta-heuristic algorithms followed by an in depth explanation of the harmony search method.

3.2 Heuristic optimization techniques

Heuristic comes from the Greek work heuriskein, which means to discover. In optimization, it refers to solution strategy by trial and error to produce a reasonable solution to complex optimization problems. Due to the complexity of some problems it would be infeasible to search for all possible solutions, the aim of this strategy is to find an acceptable solution in a reasonable amount of time. There is no way to know if the best solution can be found, nor will the algorithm work or why. A heuristic algorithm is an efficient and practical approach that has been shown to provide good results, but no guarantee of optimality.

Heuristic algorithms typically fall into three broad categories; simulated annealing, traditional genetic algorithm and evolutionary algorithms. The last two categories are very similar but have slight differences in the specifics of the algorithms.

3.2.1 Genetic algorithm (GA)

Genetic algorithms mimic the process of biological evolution in order to solve problems and to model evolutionary systems. The foundation for GAs revolves around the premise that over many generations, natural populations evolve according to the principles of natural selection, survival of the fittest. By replicating this process GAs are able to "evolve" solutions to real world problems. The main goal of GAs is the survival of robust solutions and elimination of the weak solutions in a population. GAs were first proposed by John Holland in the 1960s and further developed by Holland and his students [14]. The procedure for the genetic algorithm can be time consuming and the optimum solution may not be global ones, but they are feasible both mathematically and practically.

3.2.2 Simulated annealing (SA)

Simulated annealing (SA) is an accepted local-search technique, which utilizes the analogy between the way metals cool and freeze into a minimum energy crystalline structure (the annealing process). SA approaches the optimization problem by navigating the search space iteratively stepping from one solution to another solution. It begins at a "high" temperature, which enables it to have wide range of solutions so it can move freely around the solution space. As the temperature declines, it will settle into a relatively small range, ultimately giving an optimal solution. It was developed in 1983 to deal with highly nonlinear problems by Metropolis et al.. [15], and proposed by Kirkpatrick et al.. for optimization.

3.2.3 Ant colony optimization algorithm (ACO)

Ant colony optimization (ACO) is an algorithm based off ant methodology for finding food, and it is used to solve discrete optimization problems. The optimization problem can be transformed into a problem of finding the best path on a graph. The "ants" incrementally build solutions by moving among the graph. It utilizes several artificial characteristics such as memory, visibility and discrete time to come to an optimum solution. Dorigo et al.. was the first to utilize this method for optimization problems [16].

3.2.4 Harmony search optimization algorithm (HS)

Harmony search (HS) is a meta-heuristic algorithm that Geem et al.. developed in 2001 and makes use of the analogy between the performance of musicians and searching for optimal solutions. When musicians, play a song, he/she selects musical notes to give the best overall harmony [10,12]. The optimization solution vector is analogous to the harmony created by the musicians, whereas the musician's improvisations are analogous to the optimization search schemes. HS algorithm, unlike previous mentioned methods, does not require initial values for the decision variables. It utilizes a stochastic random search and has light mathematical requirements so it can easily be adapted to a wide range of optimization problems [10,12].

3.3 Basic of harmony search algorithm

The definition of harmony is the combination of simultaneously sounded musical notes to produce chords and chord progressions having a pleasing effect. Do, Re, Mi, Fa, Sol, La, and Si are notes, which represent a specific singular sound. HS algorithm imitates musical improvisation process where the musicians try to find a better harmony. Musicians are always striving to attain the best harmony, which can be accomplished through numerous practices of changing the notes that are played. Figure 11 gives a visual representation of the analogy between music and mathematical representation.

3.4 Harmony search optimization algorithm in structural engineering

Figure 12 illustrates the analogy once again in terms of a steel frame design. As explained by Lee and Geem, harmony memory (HM) is the most crucial part of the HS methodology. Geem's inspiration for HS was musically driven, in particular, the improvisation of jazz music. The method jazz musicians use to select their notes can be broken down into three different categories; play from memory, play from memory with a slightly different pitch, or randomly play another note. Utilization of these three processes makes up the core of the harmony search algorithm [10,12].

Many jazz performances comprise of several musicians each playing a different instrument, such as a guitarist, saxophonist and a pianist. Each member has a range of pitches they are capable of playing; guitarist [Do, Re, Mi]; saxophonist [Mi, Fa, Sol]; pianist [Sol, La, Si]. Each one is capable of playing any of their available pitches. Consider the following notes are played: guitarist Do; saxophonist Mi; pianist Sol. This would result in a harmony of [Do, Mi, Sol] [10,12].

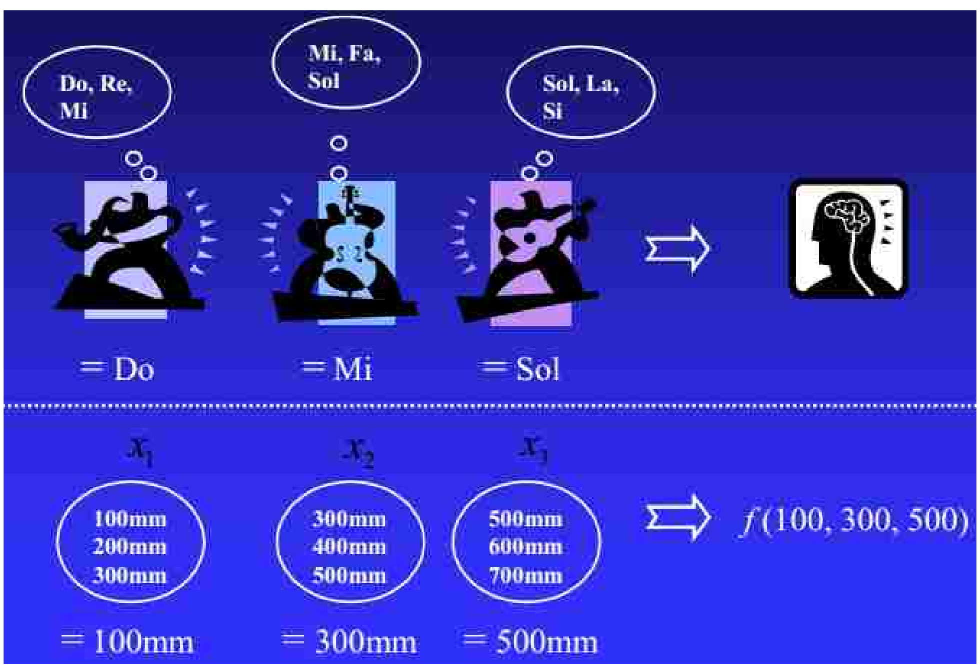


Figure 11 - Harmony search analogy [12]

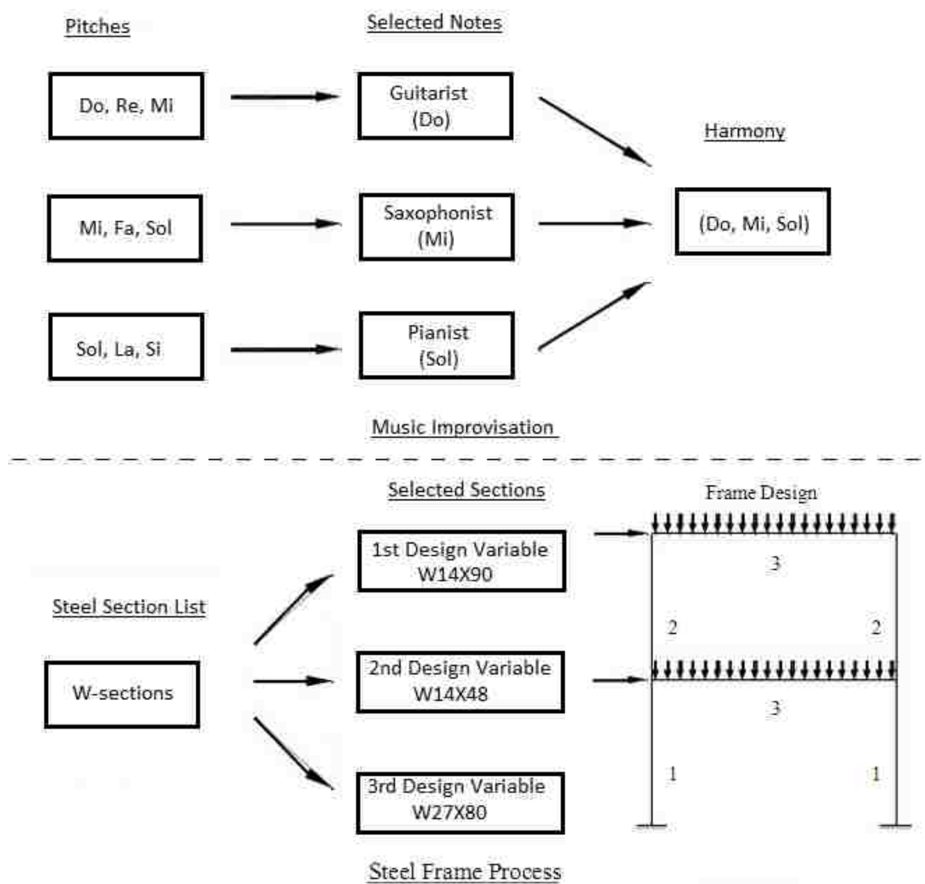


Figure 12 - Harmony search and steel frames [12]

3.4.1 Initialize the harmony search parameters

The HS algorithm parameters are selected in the first step. They are problem dependent and can be adjusted accordingly. These parameters are as followed:

- Harmony Memory Consideration Rate (HMCR).
- Harmony Memory Size (HMS).
- Pitch Adjustment Rate (PAR).
- Number of Improvisations (NI).

3.4.2 Initialize harmony memory

In the second step, the harmony memory (HM) matrix is randomly generated with design variables. Each row of the harmony memory matrix contains the values of design variables which were randomly selected from feasible solutions. The matrix has N columns where N represents the total number of design variables and it has HMS rows, which was previously selected. The harmony memory matrix can be seen in equation 6 [10,12].

$$\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \begin{matrix} \rightarrow & f(x^1) \\ \rightarrow & f(x^2) \\ \rightarrow & \vdots \\ \rightarrow & f(x^{HMS-1}) \\ \rightarrow & f(x^{HMS}) \end{matrix} \quad 6$$

3.4.3 Improvise a new harmony

In the third step, a new harmony vector, $x' = (x_1', x_2', \cdots, x_N')$ is improvised. There are three rules to choose a value for each decision variable: memory consideration (HMCR), pitch adjustment (PAR) and random selection (RN). In harmony memory considering rate, the value of the first decision variable can be chosen from any discrete or continuous value in the specified HM range with the probability of HMCR which varies between 0 and 1. Values of the other decision variables can be chosen in the same manner. However, there is still a chance where the new value can be randomly chosen from the entire set possible values with the probability of (1-HMCR) [10,12].

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \cdots, x_i^{HMS}\} \; w.p. \; HMCR \\ x_i' \in X_i \quad w.p. \; (1 - HMCR) \end{cases} \quad 7$$

Any component of the new harmony vector, whose value was chosen from the HM, is then examined to determine whether it should be pitch-adjusted. This operation uses pitch adjusting parameter (PAR) that sets the rate of pitch-adjustment decision as follows [10,12]:

$$x_i' \leftarrow \begin{cases} YES \;\; w.p. \;\; PAR \\ NO \;\; w.p. \;\; (1-PAR) \end{cases} \quad 8$$

If the pitch adjustment decision for $x_i^{'}$ is yes, $x_i^{'}$ is replaced with $x_i(k)$ (the k^{th} element in X_i), and the pitch-adjusted value of $x_i(k)$ becomes

$$x_i' \leftarrow x_i(k+c) \text{ for discrete design variables} \quad 9$$
$$x_i' \leftarrow x_i' + \alpha \text{ for continuous design variables}$$

The algorithm chooses a value form a neighboring index m with the same probability [10,12].

3.4.4 Update the harmony memory

If the new harmony $x^{'} = (x_1^{'}, x_2^{'}, \cdots, x_N^{'})$ is better than the worst harmony in the HM in terms of objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM [10,12].

3.4.5 Termination criteria

In the final step, the computation is terminated when the termination criterion is satisfied, typically a prescribed maximum number of iterations. Otherwise, Steps 3 and 4 are repeated until the termination criteria have been met [10,12]

3.4.6 Harmony search flow chart

Flow Chart Legend	
A_i: Descrete size variables (i=1,2, ... , *n*)	**PVS(*)**: Possible value set for A_i
HMCR: Harmony memory considering rate	**nPVS**: Number of possible value sets for A_i
PAR: Pitch adjustment rate	**NDHV(*)**: New discrete harmony vector
HMS: Harmony memory size	**E1**: Memory considerations process
HM(*,*): Harmony memory	**E2**: Pitch adjustment process
ran **:** Random number in the range 0.0 to 1.0	**E3**: Randomization process

Table 2 - Harmony search flow chart legend

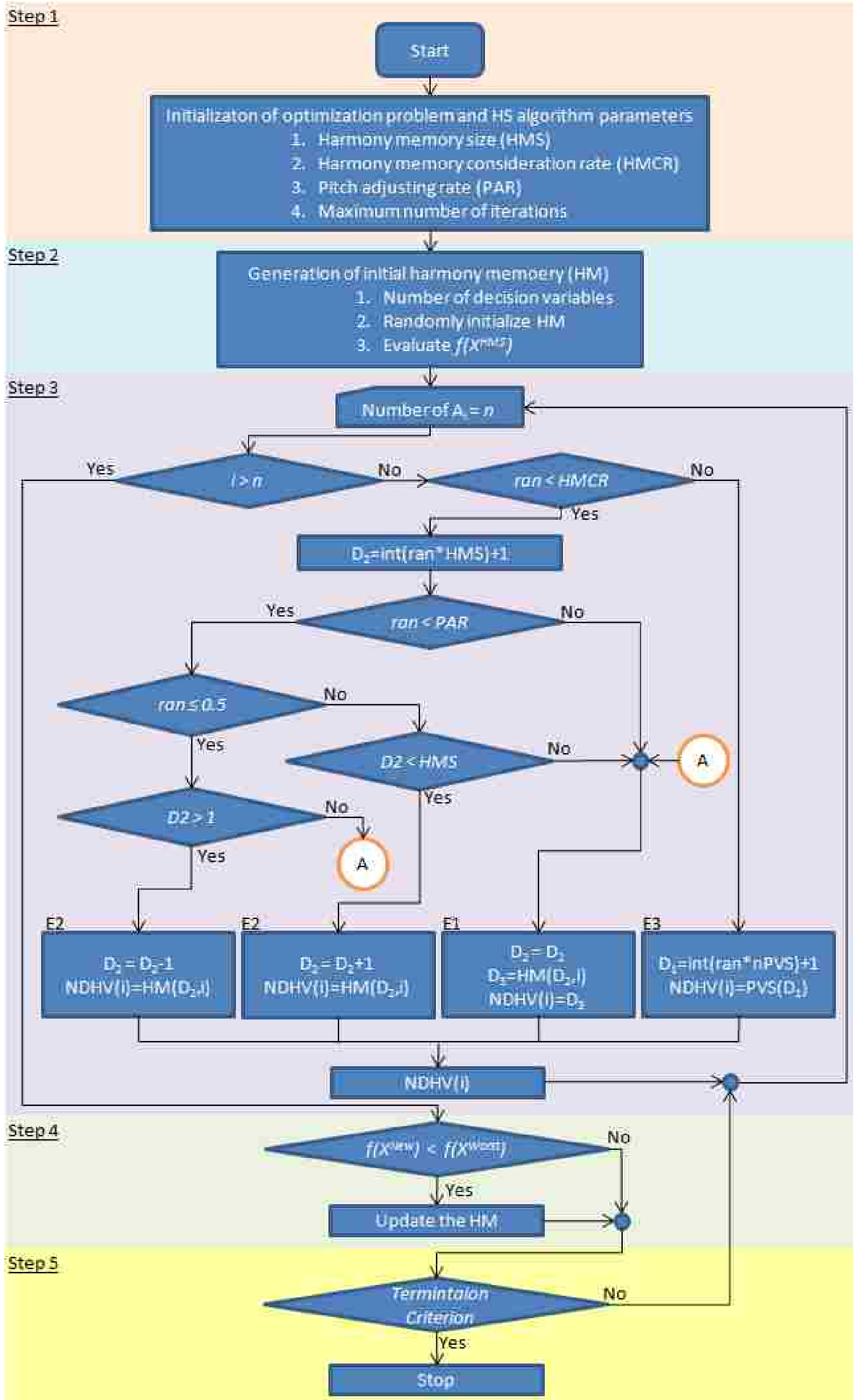


Figure 13 - Harmony search flow chart [12]

3.5 Comparison between harmony search and other optimization techniques

3.5.1 Harmony search example: 10-bar planar truss

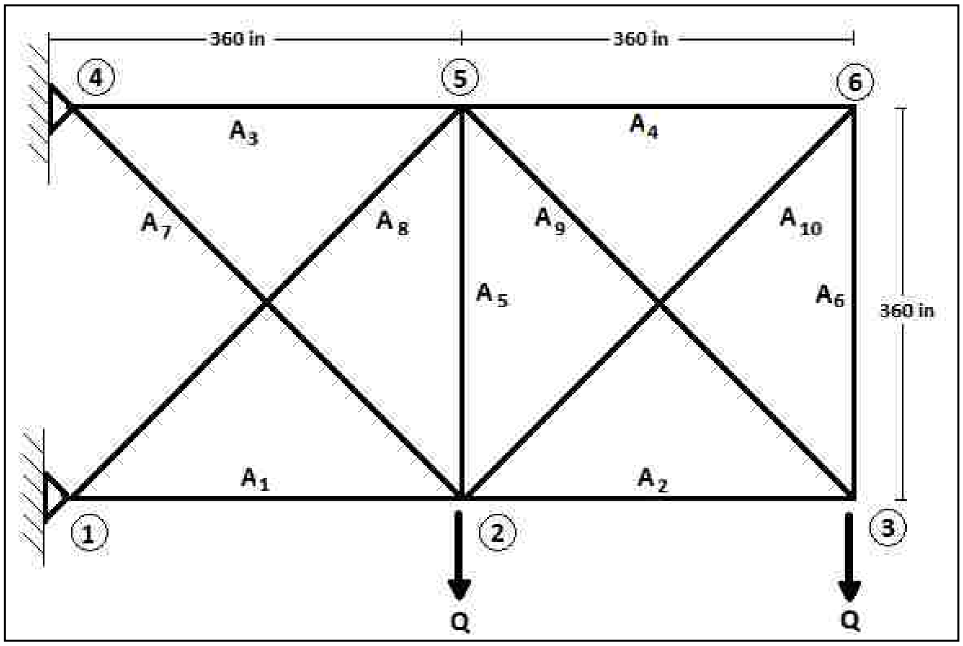


Figure 14 - 10 bar truss configuration

The cantilever truss, shown in figure 14, has been previously analyzed using various mathematical optimization methods. The material density for the truss was 0.1 lb/in^3 and the modulus of elasticity was 10,000 ksi. Stress limitations were +/- 25 ksi, and displacements at each node were limited to +/- 2.0 inches in both x and y directions. The loading case used was with two single loads of Q=100 kips. The minimum cross-sectional area of the members was 0.1in^2 and there were no maximum area limitations. Table 1 provides the data from this thesis along with several other publications for the same configuration. The one of most interest to this paper would be the findings of Kang Seok Lee [10]. Lee utilized a basic harmony search to find a minimum weight of

5057.88 lbs, which is the lowest of all the studies found in this thesis. The same harmony search methodology was ran using MATLAB for this thesis, it performed better than most of the other reports, but was unable to replicate the same results Lee found. The results were very close with only a 0.13% difference. Different formulation of the constraints and the constraint handling methods could be the reason for the discrepancy. The MATLAB code ran for this thesis utilized a static penalty function in the handling of constraints. The harmony search optimization code used for this problem provided an adequate answer and will be used for the damage tolerant optimization routines.

Table 1 : 10 Bar Truss Optimization Results [12]															
Member/Area Variable	This Thesis	Kang Seok Lee	Lamberti and Pappalettere		Xu and Grandhi	Stander et al..	Sunar and Belegundu	Khan and Willmert	Rizzi	Dobbs and Nelson	Gellatly and Berke	Venkayya	Schmit and Miura		Schmit and Farshi
A_1	24.60	22.71						24.17	23.93	23.29	20.03	23.41	23.97	23.76	24.26
A_2	15.04	15.27						14.81	17.73	15.43	15.60	14.91	14.73	14.59	14.26
A_3	30.08	30.15						30.98	30.73	30.50	31.35	30.42	30.57	30.67	33.43
A_4	0.10	0.10	*	*	*	*	*	0.10	0.10	0.10	0.10	0.13	0.37	0.10	0.10
A_5	0.10	0.10						0.10	0.10	0.10	0.14	0.10	0.10	0.10	0.10
A_6	0.54	0.54						0.41	0.10	0.21	0.24	0.10	0.36	0.10	0.10
A_7	7.34	7.54						7.55	8.54	7.65	8.35	8.70	8.55	8.58	8.39
A_8	20.99	21.56						21.05	20.95	20.98	22.21	21.08	21.11	21.07	20.74
A_9	21.23	21.45						20.94	21.84	21.82	22.06	21.08	20.77	20.96	19.69
A_{10}	0.10	0.10						0.10	0.10	0.10	0.10	0.19	0.32	0.10	0.10
Weight (lbs)	**5064.34**	**5057.88**	**5060.88**	**5060.96**	**5065.25**	**5060.85**	**5060.90**	**5066.98**	**5076.66**	**5080.00**	**5112.00**	**5084.90**	**5107.30**	**5076.85**	**5089.00**

Table 3 - 10 bar truss optimization comparison [12]

3.6 Damage tolerant optimization

3.6.1 General mathematical formulation

Find:

$$\text{Design Variables } X = [X_{1,}X_2, \dots \ , X_n]^T \ \in \ S \qquad (10)$$

That minimizes:

$$\text{Objective Function: } f(X) = [f_1(X), f_2(X), \dots \ , f_i(X), \dots \ , f_m(X)]^T \qquad (11)$$

The feasible set S belongs to an n-space determined by a set of equality and inequality conditions:

$$\text{Equality } g(X) = 0 \qquad (12)$$

$$\text{Inequality } h(X) < \ 0 \ or > 0 \qquad (13)$$

For damage tolerant optimization the objective function is composed of:

Weight	(W)	14
Intact Capacity	(C_U)	15
Residual Capacity	(C_R)	16
Displacements	(Δ)	17

Damage tolerant objective function:

$$f(X) = [W, C_U, C_R, \Delta]^T \qquad (18)$$

The design variable vector for a damage tolerant truss system is made up of member areas

$$X = [A_{1,}A_2, \dots \ , A_n]^T \qquad (19)$$

Bounds can be imposed on the cross-sectional areas resulting in the following feasible set for damage tolerant optimization

$$S = \{X \ \in \ R^n : \ A_{i,min} \le A_i \le A_{i,max} \ i = 1,2, \dots, n\} \qquad (20)$$

The above multi-objective problem can be transformed into a series of single objective minimization problems using the ∈-constraint method.

$$\min V \qquad 21$$

Satisfying

$$C_U \geq C_U^0 \qquad 22$$

$$C_R \geq C_R^0 \qquad 23$$

$$\Delta_{j,max} \leq \Delta_j^0 , \quad j = 1,2, \dots , m \qquad 24$$

$$A_{i,min} \leq A_i \leq A_{i,max}, \quad i = 1,2, \dots , n \qquad 25$$

The required residual capacity C_R^0 can be varied to cover the entire solution set. For this study we were focused on the influence of residual capacity requirements on the optimization results.

Frangopol and Klisinksi proposed a three load level checking design; nominal load (Q_N), ultimate load (Q_U), and the residual load (Q_R). Nominal and ultimate loads are used to check the serviceability and ultimate capacity requirements, respectively. The residual load is used to check the residual capacity requirement under potential future damage conditions to the structural system. Damage conditions can be represented by reductions in stiffness of members, complete removal of members, or combination of these.

In this thesis, damage conditions are assessed by complete removal of a structural member. The removal of a structural member creates a damaged structural system that will have a different performance from the original intact structure. This process can be repeated for every member since it is assumed all damage conditions are equally probable. Once the member is removed, the damaged structural system is analyzed to find out its load capacity. The damage system that provides the lowest capacity will define the residual capacity of the intact structural system, as follows:

$$C_R = \min(C_1, C_2, \dots, C_i, \dots, C_n) \qquad 17$$

where C_i is the capacity of the structure having member i removed from the system. This approach is only valid for statically indeterminate structures. Statically determinate systems require the contribution of all members to function. If a member was removed from a determinate system it would collapse. Statically indeterminate structures are inherently redundant. This allows the possible removal of one or more members from the system without the potential of collapse. Some systems may still have critical members present that cannot be removed without resulting in a system collapse. The correct indeterminate configuration must be chosen for the system to allow the consideration of residual capacity in structural optimization.

Chapter 4: Structural Reliability

4.1 Introduction

Structural reliability revolves around the uncertainties associated with the design of structures and assessing the safety of the structure. Reliability is a relatively new technique in the structural engineering field that became prominent in the 1980's. It was first implemented in the AISC code in the form of the load resistance factor design (LRFD) in 1986 as an alternative to the existing allowable stress design (ASD).

Most engineering problems are solved under the assumption of deterministic values (i.e. no randomness is involved in the value being used). In dealing with real world problems, uncertainties are unavoidable. Engineers must recognize the presence of uncertainty and account for it appropriately. Uncertainty can be classified into two expansive categories: First, those associated with natural randomness (aleatory) and second those associated with inaccuracies in human prediction and estimations (epistemic). When engineers are dealing with uncertainty, their goal is to reduce the total amount present in the current problem. The total uncertainty in a problem is the combination of both aleatory and epistemic uncertainties. Aleatoric cannot be reduced due to its intrinsic randomness, i.e., one would be hard pressed to limit the amount of earthquakes, storms, and other natural events. However, epistemic can be reduced by increasing our knowledge and providing better estimations and predictions. [17]

Epistemic uncertainty in structural engineering would be material properties such as yield strength, modulus of elasticity, thickness, etc. The random behavior of the basic strength can cause the strength of the structure to vary beyond acceptable limits. To account for this random behavior one must quantify the uncertainty, or randomness to account for this fluctuation of material properties. Uncertainty may be calculated using simulation techniques, such as Monte-Carlo simulation, which allows the values to be generated based on their statistical distribution (probability density function). Alternatively, the uncertainty can be estimating via first-order reliability method (FORM) or second-order reliability method (SORM). [17, 18]

The truss structure is defined as being made up of elements that can be in one of two states, an initial linear elastic state (safe) or a final zero-stiffness state (failure). This type of behavior is consistent with brittle material properties. For this type of structure, one can identify sequences of element failures that would lead to a structural collapse (failure event). Structural failure will occur if any failure event were to arise, i.e., the structural failure is a union of all failure sequences. This basic format allows conventional probability formulations in terms of unions and intersections to be used to represent the structure failure event. [18]

An individual structural member is considered safe or reliable when the capacity of the member exceeds the demand being placed on the member. A degree of uncertainty will be associated with both the load (L) and the resistance (R). To understand the random nature of L and R the uncertainty must be quantified and evaluated. This is typically done through a series of test such as the ones performed by Galambos and Ravindra in 1978 on the properties of steel [19]. From these results, probability density functions can be formulated to give a representation of the random properties of the material. The probability of safe performance (P_s) can be expressed as:

$$P_s = P(R > L) = P(R - L > 0) = \int \int_{R>L} f_{R,L}(r, l) dr dl \quad (26)$$

where $f_R(r)$ and $f_L(l)$ are the probability density functions of R and L and $f_{R,L}(r, l)$ is their joint probability density function. [18]

This concept can be delineated by figure 1 where the independent failure probability of both the L and the R is shown. If an incremental load value dl is considered, the probability of the load value falling into the interval dl and the strength value simultaneously exceeding the load value at that point gives the reliability of that segment dP_s which can be expressed as:

$$dP_s = f_L(l) dl \int_l^{\infty} f_R(r)\, dr = f_L(l) dl[1 - F_R(l)] \quad (27)$$

where: F_R represents the cumulative distribution function of R and $F_R(l)$ is indicated as area A_r in figure 15. The term $f_L(l)dl$ is represented by area, A_l. [18]

Since the reliability of the members involves the probability of the strength exceeding the load, the total reliability (P_s) of the member is expressed as:

$$P_s = \int dP_s = \int_{-\infty}^{\infty} f_L(l) [\int_l^{\infty} f_R(r) dr] dl = \int_{-\infty}^{\infty} f_L(l)[1 - F_R(l)]\, dl \quad (28)$$

Failure is defined as the probability that the member will not survive. This means that the probability of failure (P_f) can be expressed as:

$$P_f = 1 - P_s = 1 - P(R \geq L) = \int_{-\infty}^{\infty} f_L(l) F_R(l)\, dl \quad (29)$$

The failure probability is often computed from the reliability index β.

$$P_f = \Phi(-\beta) \quad 30$$

where Φ is the distribution function of the standard normal variety. The reliability index graphically depicts the shortest distance from the origin to a failure surface in standard normal space. [18]

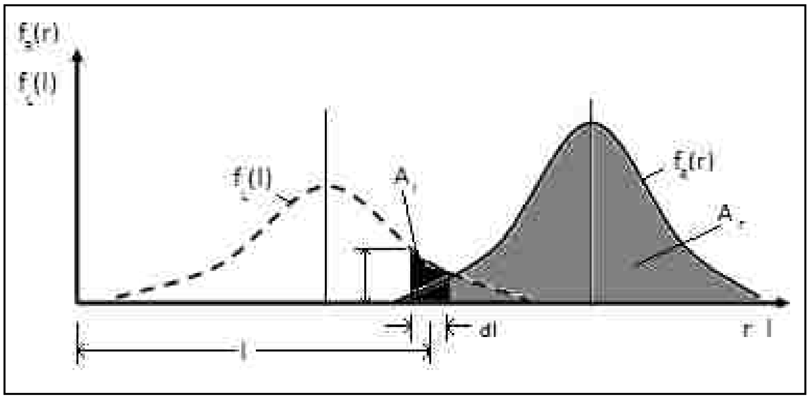


Figure 15 - Reliability diagram [18]

Chapter 5: Modeling of steel structures

5.1 Introduction

One of the most critical steps in structural analysis is the modeling process of how members will relate to each other. A model, via finite element analysis software or user defined code needs to provide an accurate representation of its members and components to function properly. One of the most difficult parts of structural analysis is developing a sound and accurate representation of these members. Rarely, if ever, it is possible to model a structural system exactly as it occurs in nature, the user must make some general assumptions about how the structure will behave. This assumptions assume structural material deform according to basic mechanics of materials. The degree of accuracy typically depends several factors such as the complexity of the model, time and cost.

5.2 MATLAB

Structural analysis for this thesis was performed in the MATLAB computer program using the direct stiffness method. Using the stiffness method requires an understanding of the concept of kinematic degrees of freedom (DOF). The kinematic degrees of freedom of a body are those motions that describe its position relative to some arbitrary base position. For example, if we consider a point in Cartesian space we can measure its movement by three translations u, v, and w in the x, y, and z directions. A rigid body in space can have rotational movement as well. To consider these movements we need to measure θx, θy, and θz, the rotations around the x, y, and z axes respectively, to completely describe the motion of points on a rigid body. This assumption assumes rigid body motion. If one were to consider a deformable body, there would be an infinite number of degrees of freedom. Each point on the body could move relative to its surrounding points.

When analyzing structures, part of the challenge is to identify which degrees of freedom are to be used. If the structure is a deformable body with an infinite number of degrees of freedom, we must choose between "exact methods" and "approximate or numerical methods." The exact methods require the solution to differential equations with the appropriate boundary conditions applied. However, due to complex and irregular shapes solving these differential equations can be very difficult, if not impossible. This is why approximate methods are typically used to solve engineering problems.

Classical approximate solutions are usually based on approximating the displacement or stress fields in the body with series approximations or finite differences. This reduces the degrees of freedom from infinity to the number of coefficients in the approximating function. The accuracy of these methods depends heavily on how well the approximating function simulates the actual solution.

In finite element methods, the structure is approximated as a series of discrete elements that use various techniques to represent internal behavior associated with the element. Typically, when representing buildings, bridges, and other structures line elements are used. These are finite elements with nodes located at each end of the element. The structural degrees of freedom are all of the element degrees of freedom. This can result in a very large number of degrees of freedom. However, with the advancements of computers large problems can be solved with minimal effort.

5.2.1 Modeling of truss

The forces in a truss element are completely determined if the displacements of the joints and the axial loads applied directly to the element are known. If we ignore element loads, the behavior of an entire truss can be determined if displacements of all the nodes are known; the nodal displacements are the degrees of freedom for formulating the problem. Truss members are represented by line elements that only support axial forces.

5.2.2 Modeling of Frame

Planar beam elements have 4 degrees of freedom. When combined with the truss elements degree of freedom it can be used to represent a frame element with 6 degrees of freedom (3 per node) capable of recognizing both axial and bending deformations. However, at the element level, these two basic types of response do not intact with each other as long as small deflections are considered. Bending of the element does not change the length of the member.

5.2.3 Stiffness method flow chart

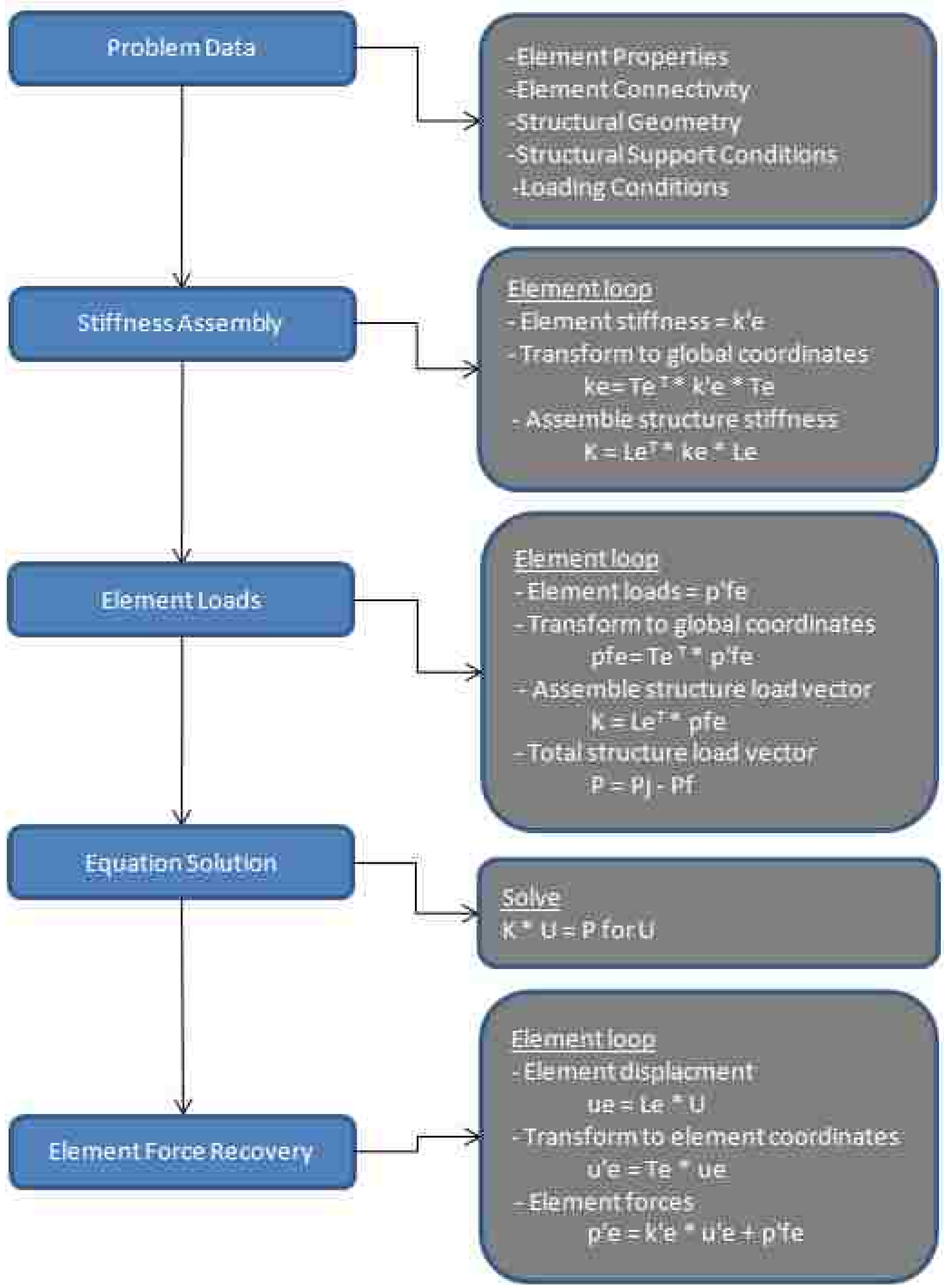


Figure 16 - Stiffness method flow chart

5.2.4 Sources of Nonlinearity

In linear elastic analysis, the materials investigated are assumed linear elastic, meaning the material is unyielding and its properties unchanging. The equations of equilibrium are formulated on the geometry of the unloaded initial structural configuration. It assumes very small subsequent deformations. This approach has the ability to treat axial force, bending moments and torques as uncoupled actions in stiffness equations. Several options are available to address the issues from the linear elastic assumptions. These can be generalized into two categories, geometric nonlinearities and material nonlinearities. The first category, geometric nonlinearities continues to treat the structure as an elastic material but includes the effects of deformation and finite displacements in formulating the equations of equilibrium. The latter category, material nonlinearity considers the effect of changes in member material properties under load.

5.2.5 Semi-rigid connection nonlinearity

Semi-rigid connections are a source of material nonlinearities. The modeling of semi-rigid connections is typically handled by modifying the member stiffness matrix to account for the connection flexibility via end-fixity factors. The implementation of the concept of end-fixity factor into frame analysis can be done by multiplying the rigid member stiffness matrix S_i, by a correction matrix, C_{e-i} seen in equations 31. [6]

$$K_i^{SR} = S_i * C_{e-i} \qquad 31$$

$$S_i = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & \frac{-EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{-12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4Ei}{L} & 0 & \frac{-6EI}{L^2} & \frac{2EI}{L} \\ \frac{-EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & \frac{-12EI}{L^3} & \frac{-6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{-6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{-6EI}{L^2} & \frac{4Ei}{L} \end{bmatrix} \quad 32$$

$$C_{e-i} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4r_2 - 2r_1 + r_1 r_2}{4 - r_1 r_2} & \frac{-2Lr_1(1 - r_2)}{4 - r_1 r_2} & 0 & 0 & 0 \\ 0 & \frac{6(r_1 - r_2)}{L(4 - r_1 r_2)} & \frac{-3r_1(2 - r_2)}{4 - r_1 r_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{4r_1 - 2r_2 + r_1 r_2}{4 - r_1 r_2} & \frac{2Lr_2(1 - r_1)}{4 - r_1 r_2} \\ 0 & 0 & 0 & 0 & \frac{6(r_1 - r_2)}{L(4 - r_1 r_2)} & \frac{-3r_2(2 - r_1)}{4 - r_1 r_2} \end{bmatrix} \quad 33$$

where end-fixity factors $\mathbf{r_1}$ and $\mathbf{r_2}$ are defined by:

$$r_j = \frac{1}{1 + 3EI/R_j L} \quad (j = 1,2) \quad 34$$

where end connection spring stiffness, R_j, is defined by the Frye and Morris curve fitting model in chapter 2. To take into account the nonlinear behavior of semi-rigid connections, an iterative process is used to obtain the solution. In each iteration, the member stiffness is modified by the correction matrix with updated end-fixity factors $\mathbf{r_1}$ and $\mathbf{r_2}$. [6]

5.2.6 Geometric nonlinearity

To account for geometric nonlinearities in structural systems a second-order elastic analysis needs to be performed. For rigid frames, the computer based second-order elastic analysis is often done as an iterative procedure and the stiffness matrix of each member is composed of the elastic stiffness matrix and the geometrical stiffness matrix as show in equation 35: [6]

$$K_i = S_i + G_i \tag{35}$$

$$G_i = \frac{N}{L}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6}{5} & \frac{L}{10} & 0 & \frac{-6}{5} & \frac{L}{10} \\ 0 & \frac{L}{10} & \frac{2L^2}{15} & 0 & \frac{-L}{10} & \frac{-L^2}{30} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-6}{5} & \frac{-L}{10} & 0 & \frac{6}{5} & \frac{-L}{10} \\ 0 & \frac{L}{10} & \frac{-L^2}{30} & 0 & \frac{-L}{10} & \frac{2L^2}{15} \end{bmatrix} \tag{36}$$

The geometrical stiffness matrix can also take into account semi-rigid connections with the addition of a correction matrix, C_{g-i} as seen in the following equations. [6]

$$G_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & C1 & C2 & 0 & C3 & C4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & C5 & C6 & 0 & C7 & C8 \end{bmatrix} \tag{37}$$

where

$$C1 = -C3 = \frac{-4}{5L(4 - r_1 r_2)^2}(8r_1^2 r_2 - 13r_2^2 r_1 - 32r_1^2 - 8r_2^2 + 25r_1 r_2 + 20) \tag{38}$$

$$C2 = \frac{r_1}{5(4 - r_1 r_2)^2}(16r_2^2 + 25r_2^2 r_1 - 96r_1 r_2 + 128r_1 - 28r_2) \tag{39}$$

$$C4 = \frac{4r_2}{5(4 - r_1 r_2)^2}(16r_1^2 - 5r_1^2 r_2 + 9r_1 r_2 - 28r_1 + 8r_2) \tag{40}$$

$$C5 = -C7 = \frac{-4}{5L(4 - r_1 r_2)^2}(8r_2^2 r_1 - 13r_1^2 r_2 - 32r_2^2 - 8r_1^2 + 25r_1 r_2 + 20) \tag{41}$$

$$C6 = \frac{4r_1}{5(4 - r_1 r_2)^2}(16r_2^2 - 5r_2^2 r_1 + 9r_1 r_2 - 28r_2 + 8r_1) \tag{42}$$

$$C8 = \frac{r_2}{5(4 - r_1 r_2)^2}(16r_1^2 + 25r_1^2 r_2 - 96r_1 r_2 + 128r_2 - 28r_1)$$

5.4 SAP2000

SAP2000 is a general-purpose engineering software package ideal for analysis and design of structural systems. It can represent a wide range of systems from basic 2-D systems to complex 3-D structures. Modeling of the structural elements is typically handled through a graphical user interface (GUI) that allows the user to define members, loading, material properties, etc. The user also has the option to edit an input file to define the structural system.

SAP2000 was utilized to validate the MATLAB model. Both analysis methods were performed on the three story, two bay frame with semi-rigid connections shown in

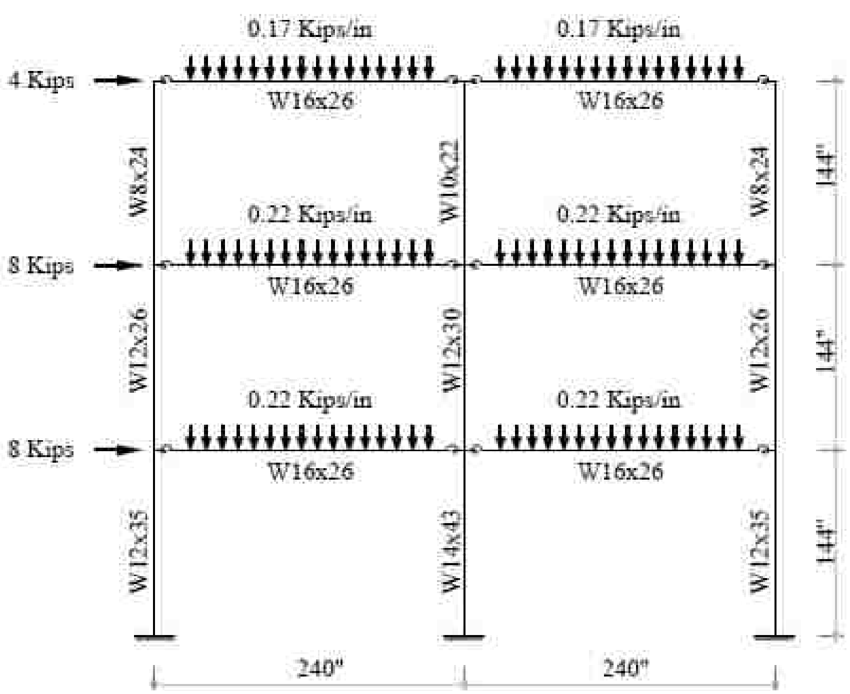


Figure 17 - Frame diagram

figure 17. The material properties of the structural members were in accordance to the AISC steel design manual and the modulus of elasticity was set at 30,000ksi for the analysis.

To account for semi-rigid connections the moment rotation curve shown in figure 18 was used. The moment rotation curve is representative of an extended end plate connection without column stiffeners and the curve fitting constancies for this configuration can be seen in table 1 in chapter 2. The end plate was assumed to have a thickness of 0.685" using 1" diameter bolts with a spacing between bolt groups equal to the member depths plus 6".

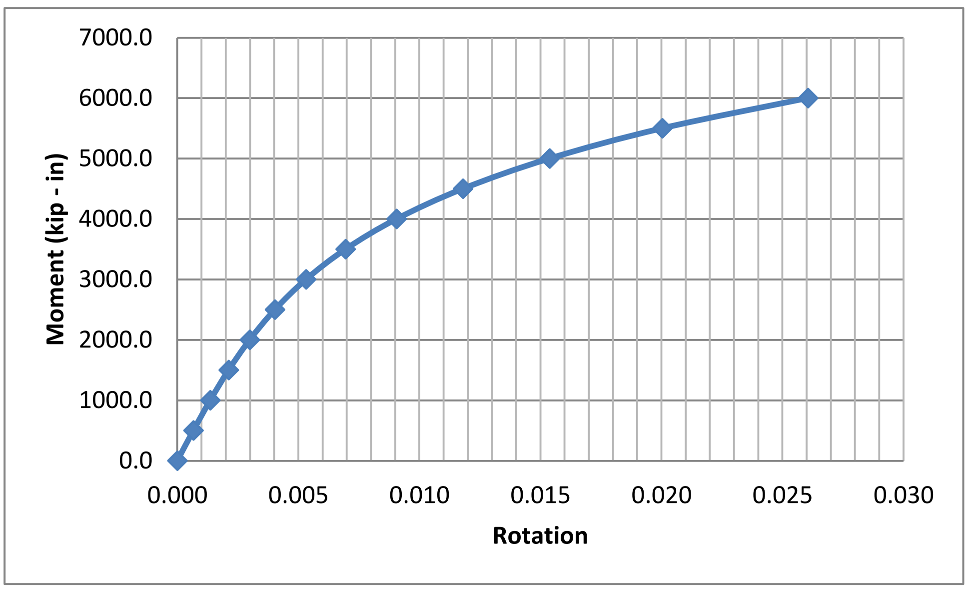


Figure 13 - Extended end plate moment rotation curve

SAP2000 does not have the capabilities to specify moment rotation interaction of connections. Instead, a rotational spring must be used at the connection points to represent the connection flexibility. For the testing analysis a secant stiffness of $6.35x10^5$ (K.in/rad) was used as the member partial fixity values in both computer models.

	Nonlinear analysis of Semi-rigid Frame		
	MATLAB	SAP2000	Percent Diffrence
Upper right corner displacmenet (in.)	1.12	1.15	2.64%
Maximum column base moment (K-in)	919.4	905.61	1.51%

Table 4 - Analysis comparison

The results obtained from both models were relatively similar when compared to one another. The nodal displacements of the MATLAB model were all within 1-3% of the SAP model. Also, element bending, shear and axial loading vary from about 1-2% of each other. The results show that the MATLAB model is an acceptable representation of the structural system and can be used for the optimization process.

Chapter 6: Design Examples

6.1 Damage tolerant truss

The 10-bar truss configuration shown in figure 2 with three proportional loads Q will be analyzed for damage tolerant optimization as outlined in Chapter 3. Geometrical, mechanical and loading characteristic were assumed to be deterministic. The modulus of elasticity was assumed to be E = 29,000ksi. Each member was assigned its own individual cross sectional area A_i as shown in figure 2. The material was assumed to be brittle with yielding stresses of +/- 25ksi. Buckling constraints were also applied to each individual member. The minimum and maximum cross-sectional areas were A_{imin}=0.1 and A_{imax}=infinity. The initial cross sectional areas of the truss were set at A_i =1inch.

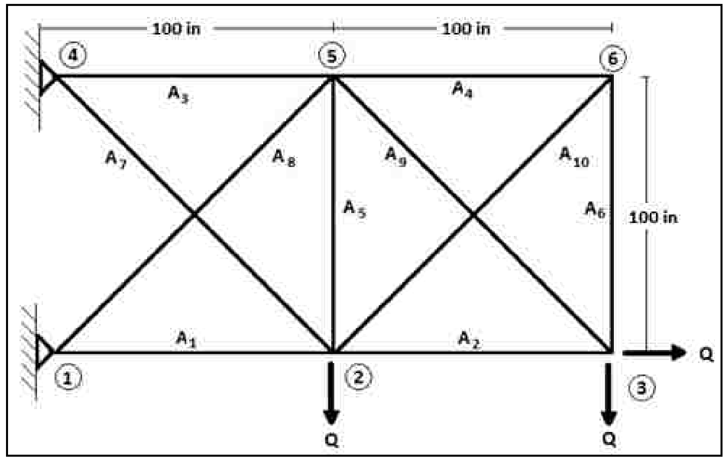


Figure 14 - Ten bar truss configuration optimization

6.1.2 Solving damage tolerant optimization problem

The design variables for this problem are restricted to the cross sectional areas of the structural members. The geometry of the structure and material properties are considered fixed. The cross sectional areas of the members are the design variables A_i, subjected to size constraints

$$A_{i,min} \ < \ A_i \ < A_{i,max} \, ; i = 1, \ldots, n \qquad 44$$

where $A_{i,min}$ and $A_{i,max}$ are the minimum and maximum member areas, respectively. In this problem the minimum member area is limited to 0.1 in^2 and the maximum area is not limited.

The objective function for the optimization problem is to minimize the volume of the structure:

$$V = \sum_{i=1}^{n} l_i a_i \quad (45)$$

where l_i is the total length of the members anda_i is the corresponding area of the member. The volume V can be multiplied by the unit weight of the structure to provide an adequate assessment of the structures cost. The following constraints must be satisfied.

Ultimate load carrying capacity requirement:

$$C_U \geq C_U^0 \quad (46)$$

where C_U and C_U^0 are the actual and the required ultimate load carrying capacity of the system respectively.

Serviceability requirements:

$$\Delta_i \leq \Delta_i^0, for\ all\ i \quad (47)$$

where Δ_i and Δ_i^0 are the maximum and the allowable elastic displacement at section i, respectively. It is logical to compute the displacements under the nominal load Q_N.

Residual capacity requirement:

$$C_R \geq C_R^0 \quad (48)$$

where C_R and C_R^0 are the actual and the required residual capacity of the system, respectively.

Reserve strength factor is defined as:

$$R_1 = C_U/Q_N \tag{49}$$

The reserve factor is a measurement of strength that compares the ultimate load to the nominal loading. The reserve strength factor, R_1, can range from value of 0 when the intact structure has no loading effect, to a value of 1.0 when the nominal load on the intact structure equals its capacity, C_U.

Residual strength factor is defined as:

$$R_2 = C_R/C_U \tag{50}$$

The residual strength factor, R_2, is used to show the strength of a structure once it is in a damage state. This value can range from 0 when the damage structure is collapsed to a theoretical value of 1.0 when the damage structure can carry the same load capacity as the intact structure. Frangopol and Klisinski show that for a given structural configuration, loading and material behavior there is always a maximum value of residual strength. This is due to the fact that the residual capacity of the structural system, C_R, cannot increase over a certain threshold without raising the ultimate capacity of the intact structure, C_U.

6.1.3 Optimization results

First, the behavior of the initial intact structure was investigated. The initial structure had a volume of 1165.69in^3. Table 5 shows the stress distribution for the intact structure along with all (ten) possible damage scenarios. The ultimate capacity of the intact structure was 11.92 kips and the residual capacity was 5.06 kips. This resulted in a residual strength factor of 0.425. The governing constraint for the intact ultimate strength was buckling for member 8. The governing residual capacity was the removal of member 7; when removed the buckling constraint in member 8 was once again reached.

10-Bar Truss Intact and Damaged Trusses: Ultimate Capacities and Associated Stresses $A_1 = A_2 = A_3 = A_4 = A_5 = A_6 = A_7 = A_8 = A_9 = A_{10} = 1\ in^3$; $V_{INTACT} = 1165.69\ in^4$											
Element Removed	Load Level (kips)	Stress In Elements (ksi)									
		1	2	3	4	5	6	7	8	9	10
NONE	11.92	-13.72	3.68	22.04	3.68	1.88	3.68	19.41	**-14.31**	11.66	-5.20
1	5.06	---	0.96	15.18	0.96	6.02	0.96	0.00	**-14.31**	5.80	-1.35
2	11.44	-12.76	---	21.56	0.00	-1.32	0.00	18.05	**-14.31**	16.18	0.00
3	5.89	-17.68	2.95	---	2.95	-8.84	2.95	**25.00**	8.33	4.17	-4.17
4	11.44	-12.76	0.00	21.56	---	-1.32	0.00	18.05	**-14.31**	16.18	0.00
5	13.22	-16.32	3.10	23.34	3.10	---	3.10	23.08	**-14.31**	14.31	-4.39
6	11.44	-12.76	0.00	21.56	0.00	-1.32	---	18.05	**-14.31**	16.18	0.00
7	5.06	0.00	0.96	15.18	0.96	6.02	0.96	---	**-14.31**	5.80	-1.35
8	8.84	-17.68	3.50	8.84	3.50	-5.33	3.50	**25.00**	---	7.54	-4.96
9	10.12	-12.46	10.12	17.90	10.12	7.78	10.12	17.62	-11.00	---	**-14.31**
10	11.44	-12.76	0.00	21.56	0.00	-1.32	0.00	18.05	**-14.31**	16.18	---

Note: Bolded Stress represent failure of corresponding member via stress limits or buckling limits.

Table 5 - Truss damage conditions

An interesting observation happened from the removal of member 5. The ultimate loading for the truss actually increased due to the distribution of forces shedding load from the 8th member. This allowed a higher load to be placed on the structure before the buckling constraint in member 8 was reached.

Next, we will look at the same truss configuration optimized in two different ways. First it was optimized for minimum volume under ultimate capacity requirements and second for minimum volume under both ultimate capacity and residual capacity requirements. The results of these optimizations along with the initial truss results can be found in table 7.

The initial truss has a volume of 1165.69 and an ultimate loading of 11.92 kips due to buckling constraints. The optimized truss for only ultimate loading had a significant decrease in total volume by over half coming in at 560.40 in^3. This truss configuration sacrifices a significant amount residual capacity from the original intact truss, reducing the residual capacity of the truss by 70.36%

This brings us to the next optimization routine, to consider both ultimate capacity and residual capacity of the system. Optimizing the truss with the same ultimate and residual capacities as the initial structure resulted in a decrease in volume from 1165.69 to 659.73 or a reduction of 43.4%. The residual capacity was then varied while keeping the ultimate capacity requirement fix to show different scenarios of optimization. The truss was able to stay under the initial volume while having a significant increase in residual strength of the system. These gains become capped at a residual factor of around 0.71 due to additional increase of the residual capacity results in an increase of the ultimate capacity. Additional optimizations were run for an increased ultimate capacity and residual capacity.

Due to the low complexity of this problem, both Harmony Search and Gradient methods were able to be used for minimization. As expected the gradient methods provided the global minimum values. The harmony search method is a metaherustic algorithm and does not guarantee a global optimal solution. However, the search routine was

Optimized for CU		
	Gradient Methods	Harmony Search
A1	0.4109	0.4067
A2	0.1000	0.1000
A3	1.0195	1.0241
A4	0.1000	0.1002
A5	0.1000	0.1058
A6	0.1000	0.1001
A7	0.5811	0.5748
A8	1.3407	1.3522
A9	0.6465	0.6464
A10	0.1000	0.1000
Volume	560.3994	561.7659

Table 6 - Optimization comparison

not far off of the calculus based optimization as seen in table 6. It also implements stochastic optimization methods making the final solution vary slightly from run to run. Due to these factors, a majority of the figures and tables were produced with the calculus methods for more consistent results. The final results for the truss system can be seen in table 7.

10-Bar Truss : Optimization													
Areas (in^2)										Volume (in^3)	Ultimate Capacity (kips)	Residual Capacity (kips)	Residual Factor
A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	V	C_U	C_R	R_2
Initial													
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1165.69	11.92	5.06	0.42
Optimized for CU													
$V \rightarrow Min$ $C_U \geq 11.92$ *kips* $0.1 \leq A_i \leq \infty$,i=1,2, ... , 10													
0.41	0.10	1.02	0.10	0.10	0.10	0.58	1.34	0.65	0.10	560.40	**11.92**	1.50	0.13
Optimized for CU and CR													
$V \rightarrow Min$ $C_U \geq 11.92$ *kips* $C_R \geq Q_F$ $0.1 \leq A_i \leq \infty$,i=1,2, ... , 10													
0.61	0.20	0.95	0.20	0.34	0.20	0.86	1.18	0.57	0.29	659.73	**11.92**	**5.06**	0.42
0.72	0.24	0.90	0.24	0.39	0.24	1.02	1.03	0.54	0.34	688.05	**11.92**	**6.00**	0.50
0.84	0.28	0.84	0.28	0.45	0.28	1.19	0.89	0.51	0.40	719.12	**11.92**	**7.00**	0.59
0.96	0.32	0.96	0.32	0.51	0.32	1.36	0.91	0.49	0.45	792.33	**11.92**	**8.00**	0.67
1.02	0.34	1.02	0.34	0.53	0.34	1.44	0.96	0.48	0.48	835.60	**11.92**	**8.50**	0.71
1.03	0.34	1.03	0.34	0.54	0.34	1.46	0.97	0.49	0.49	845.26	11.92	**8.60**	0.72
1.20	0.40	1.20	0.40	0.63	0.40	1.70	1.13	0.62	0.57	991.25	**15.00**	**10.00**	0.67
1.20	0.40	1.51	0.40	0.66	0.40	1.70	1.74	0.92	0.57	1152.42	**20.00**	**10.00**	0.50

Note : Active constraints are indicated in bold

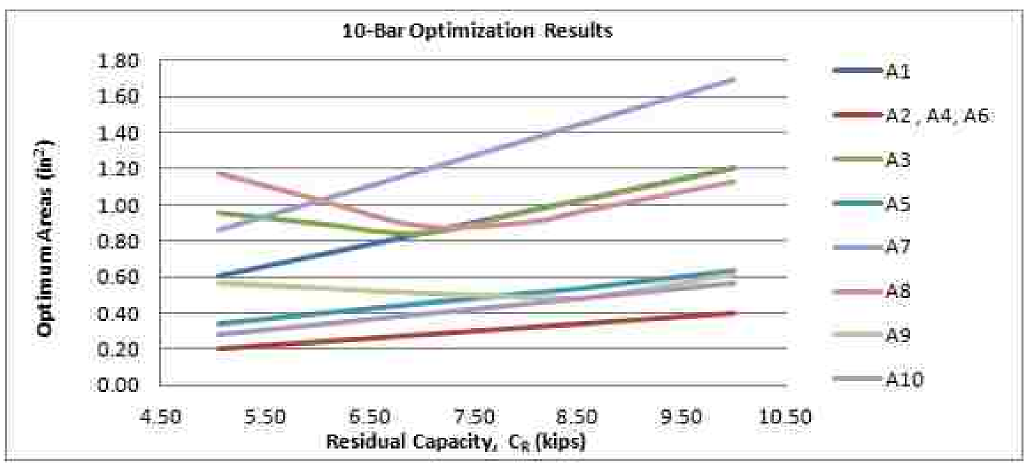


Table 7 - Truss results

6.2 Frame Optimization

The established harmony search optimization algorithm is used to formulate a minimum weight steel frame design. The constraints imposed on the problem will be in accordance to the 2005 AISC-LRFD strength requirements, displacement limitations, and size constraints for beam-column elements. Figure 20 shows the frame configuration, dimensions, loading and grouping of members. The optimum results collected from the harmony search optimization consider the design of rigid and semi-rigid steel connections and account for linear and nonlinear effects. The obtained results will be compared to an identical structure being optimized with the Genetic Algorithm Technique. The optimization program has discrete variables, which match the AISC shape database. All members with a weight less than 200 lbs were considered in the

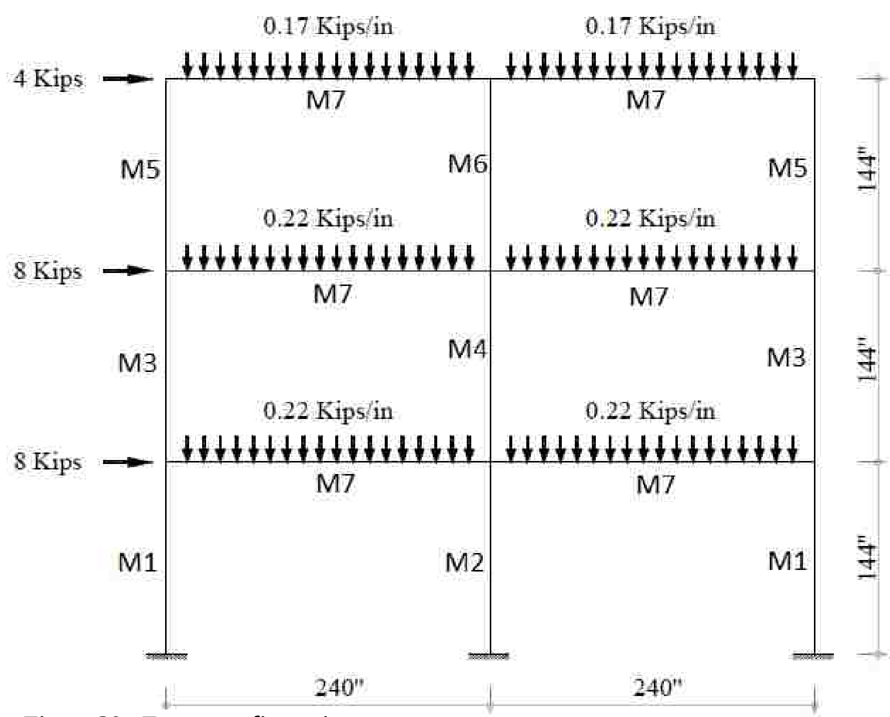


Figure 20 - Frame configuration

optimization routine.

6.2.2 Frame design parameters

- A36 grade steel
- Young's modulus E=30,000ksi
- Allowable total drift (H/300) = 1.44"
- Allowable interstory drift (h/300) = 0.48"
- Allowable beam deflection (L/240) = 1"
- Out of plane effective length for columns (Ky) = 1.0
- Length of the unbraced compression flange for each column was calculated during the optimization process.
- Floor stringers were assumed to be at L/6 points of the beam span resulting in the out of plane unbraced length (Ky) = 40"

6.2.3 Harmony search design parameters

The following harmony search parameters were selected based on literature recommendations and several trials of the problem to achieve the most refined optimal solution.

- **Harmony memory size (HMS) -** The value for harmony memory was equal to 25. This value limits the number of solutions stored in algorithm memory. It was found that a value of 25 had a good tradeoff between run time and accuracy.

- **Harmony memory consideration rate (HMCR) -** The value for the HMCR was equal to 0.9, which reflects the probability for selecting a value from memory. Once again, it was found that a 90% consideration rate provided a good balance between time and accuracy.
- **Pitch adjusting rate (PAR) -** The pitch-adjusting rate was equal to a value of 0.45, like the HMCR this value reflects the probability of pitch adjustment. Increased values of PAR caused the solution to converge on non optimal designs where as lower values would not converge on the global maximum.
- **Termination criterion -** The termination criteria was set as a maximum number of iterations of 8000. After several trials, it was found that the solution typically converged around 6000. The run time for 8000 iterations with the other parameters listed above takes roughly 75 minutes for the nonlinear analysis. Whereas the linear rigid analysis takes approximately 5 minutes over the same iterations.
- **Number of runs -** The optimal solution from the harmony search is not a global optimal. Because of this, the optimal solutions will vary from run to run. Ten independent runs were performed to get an average minimum weight structure.

6.2.2 Objective function

The objective for the optimization process is to achieve a minimum weight steel frame design. The design variables for this problem are the AISC steel members. The weight of the frame can be expressed in the following equation:

$$W(x) = \sum_{K=1}^{ng} A_k \sum_{\mathrm{i=1}}^{\mathrm{mk}} \mathrm{\rho_i \, L_i} \qquad 51$$

where ng is the total number of member groups, mkrepresents the total number of members in that group. The terms $\mathrm{\rho}$, $\mathrm{A_K}$, and $\mathrm{L_i}$ represent member density, area, and length respectively.

6.2.3 Unconstrained objective penalty function

The unconstrained penalty formula calculates the weight of the new design with an included penalty if any constraints have violation and can be expressed as:

$$\varphi(x) = W(x)[1 + KC]^{\epsilon} \qquad 52$$

where K = Penalty constant, C = Constraint violation function and ∈ = Penalty function exponent. For this design example the values of K=1.0 and ∈ = 2.0 were used

6.2.4 Constraint violation function formula

$$C = \sum_{i=1}^{n_t} C_i^{md} \sum_{i=1}^{n_s} C_i^{id} \sum_{i=1}^{n_c} C_i^{sc} \sum_{i=1}^{n_f} C_i^{sb} \sum_{i=1}^{n_f} C_i^{d} \sum_{i=1}^{n_c} C_i^{I} \qquad 53$$

where C_i^{md} is the constraint violations for max drift, C_i^{id} is the constraint violations for interstory drift, $C_i^{sc}$is the constraint violations for column sizes, C_i^{sb} is the constraint violations for beam sizes, C_i^{d} is the constraint violations for deflections and C_i^{I} is the constraint violations for the LRFD interaction equations. The constraint violations are determined based on the following equation:

$$\mathrm{C_i} = \begin{cases} 0 & \text{if } \beta_i \leq 0 \\ \beta_i & \text{if } \beta_i > 0 \end{cases} \quad 54$$

6.2.5 Drift constraints

$$\beta_i^t = \frac{|\Delta_t|}{|\Delta_t^u|} - 1.0 \leq 0 \quad 55$$

$$\beta_i^d = \frac{|\Delta_i|}{|\Delta_i^u|} - 1.0 \leq 0 \quad i = 1,2, \dots, n_s \quad 56$$

where Δ_t is the maximum top story displacement, Δ_t^u is the allowable top story displacement, Δ_i is interstory displacement, Δ_i^u is allowable interstory displacement and ns is the number of stories.

6.2.6 Size constraints

$$\beta_i^{sc} = \frac{d_t}{d_b} - 1.0 \leq 0 \quad i = 1,2, \dots, n_c \quad 57$$

$$\beta_i^{sb} = \frac{b_{bf}}{b_{cf}} - 1.0 \leq 0 \quad i = 1,2, \dots, n_f \quad 58$$

where d_t is the depth of the top compression member, d_b is the depth of the bottom compression member, b_{bf} beam flange width, $b_{cf}$is the column flange width, n_c is the number of compression members and n_f is the number of floors.

6.2.7 Deflection constraints

$$\beta_i^{db} = \frac{d_{db}}{d_{db}^u} - 1.0 \leq 0 \quad i = 1,2, \dots, n_b \quad 59$$

where d_{db} is the deflection of the beam, d_{db}^{u} is the allowable beam deflection and n_b is the number of beams.

6.2.8 Strength constraints

Interaction equations from AISC-LRFD were used as the strength constraints for this problem. Doubly and singly symmetric members in flexure and compression should comply with AISC equations H1-1a or H1-1b.

(a) For $\frac{\mathrm{P_r}}{\mathrm{P_c}} \geq 0.2$

$$\beta_i^I = \frac{\mathrm{P_r}}{\mathrm{P_c}} + \frac{8}{9}\left(\frac{\mathrm{M_{rx}}}{\mathrm{M_{cx}}} + \frac{\mathrm{M_{ry}}}{\mathrm{M_{cy}}}\right) - 1.0 \leq 0 \quad 60$$

(b) For $\frac{\mathrm{P_r}}{\mathrm{P_c}} < 0.2$

$$\beta_i^I = \frac{\mathrm{P_r}}{2\mathrm{P_c}} + \left(\frac{\mathrm{M_{rx}}}{\mathrm{M_{cx}}} + \frac{\mathrm{M_{ry}}}{\mathrm{M_{cy}}}\right) - 1.0 \leq 0 \quad 61$$

where $\mathrm{P_r}$ is the required axial compressive strength, $\mathrm{P_c}$ is the available axial compressive strength, $\mathrm{M_r}$ is the required flexural strength, $\mathrm{M_c}$ is the available flexural strength, x and y refer to strong and weak axis bending respectively.

Lateral torsional buckling (LTB) should be checked depending on the unbraced length $\mathrm{L_b}$ as follows:

(1) $\mathrm{L_b} \leq \mathrm{L_p}$

$$\mathrm{M_n} = \mathrm{M_p} = \mathrm{F_y Z_x} \quad 62$$

(2) $\mathrm{L_p} < L_\mathrm{b} \leq \mathrm{L_r}$,

$$M_n = C_b\left[M_p - \left(M_p - 0.7F_yS_x\right)\left(\frac{L_b - L_p}{L_r - L_p}\right)\right] \le M_p \tag{63}$$

(3) $L_b > L_r$

$$M_n = S_xF_{cr} \tag{65}$$

where

$$M_r = 0.7F_yS_x \tag{65}$$

$$F_{cr} = \frac{C_b\pi^2E}{\left(\frac{L_b}{r_{ts}}\right)^2}\sqrt{1 + 0.078\frac{Jc}{S_xh_o}\left(\frac{L_b}{r_{ts}}\right)^2} \tag{66}$$

$$L_p = 1.76r_y\sqrt{\frac{E}{F_y}} \tag{67}$$

$$L_r = 1.95r_{ts}\frac{E}{0.7F_y}\sqrt{\frac{Jc}{S_xh_o}}\sqrt{1 + \sqrt{1 + 6.76(\frac{0.7F_y}{E}\frac{S_xh_o}{Jc}}} \tag{68}$$

$$r_{ts}^2 = \frac{\sqrt{I_yC_w}}{S_x} \tag{69}$$

$$C_b = \frac{12.5M_{max}}{2.5M_{max} + 3M_a + 4M_b + 3M_c} \le 3.0 \tag{70}$$

where c equals 1 for doubly symmetric shapes and h_o is the distance between flange centroids.

6.2.8.1 Column strength

AISC Column strength is computed from the following equations:

$$P_n = A_g F_{cr} \quad 71$$

(a) $\frac{KL}{r} \leq 4.71\sqrt{\frac{E}{F_y}}$

$$F_{cr} = \left[0.658^{\frac{F_y}{F_e}}\right] F_y \quad 72$$

(b) $\frac{KL}{r} > 4.71\sqrt{\frac{E}{F_y}}$

$$F_{cr} = 0.877 F_e \quad 73$$

$$F_e = \frac{\pi^2 E}{\left(\frac{KL}{r}\right)^2} \quad 74$$

where K is the effective length factor. The effective length factor for unbraced compression flange for each column is calculated throughout the design process from the following equation:

$$K = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \quad 75$$

where A and B represent the top and bottom of the column and the restraint factor G is calculated as

$$F_e = \frac{\Sigma(I_C/L_C)}{\Sigma(I_B/L_B)} \quad 76$$

where I_C and $I_B$are moment of inertias and L_C and L_B are unbraced length of the column and beam respectively.

6.2.9 Rigid frame results

Ten separate optimization routines were performed for the structural frame with rigid and semi-rigid connections. From the constraints, it was apparent that the limiting factor for the frame design was strength constraints for rigid connections and a combination of strength and displacement for semi-rigid connections. The lateral drift of the structures were well below acceptable values for all trials with rigid connections. The maximum drift for the rigid design was 0.76" while the semi-rigid frame was at 1.43" due to a reduction in the frames stiffness.

The program was set with a max

Frame Optimization Rigid Connections		
Frame Analysis	Optimum Weight	Max Improvisation
1	6908	8000
2	6461	8000
3	6973	8000
4	6791	8000
5	6754	8000
6	6729	8000
7	6804	8000
8	7065	8000
9	6539	8000
10	6430	8000
Average (lbs)	6745	
Standard Devation (lbs)	213	
Minimum (lbs)	6430	

Table 8 - Rigid frame results

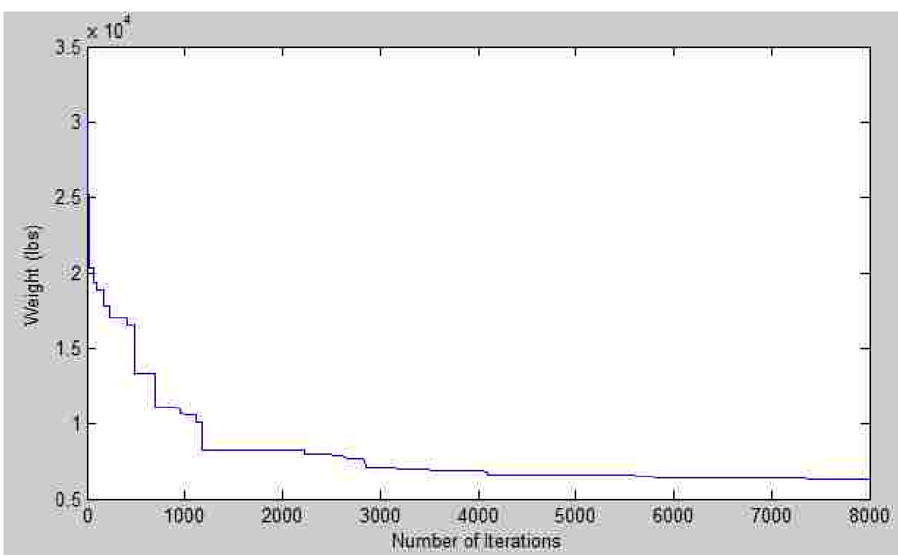


Figure 21 - Harmony search iterations

iteration of 8000. This value has a lot of control over the optimization results as the more iterations the better chance of a lower weight design. However, it does have a point of diminishing returns. Several extra hours of computer time could be needed for a few extra thousand iterations that may only improve the final solution by a fraction of a percent. The design results for the rigid frame and semi-rigid frame analysis are shown in tables 8 and 9 respectively. The results from table 10 show that harmony search optimization provided a lighter frame compared to the genetic algorithm used by Saka. The results show a reduction in

Frame Optimization Semi-Rigid Connections		
Frame Analysis	Optimum Weight	Max Improvisation
1	6314	8000
2	6498	8000
3	6895	8000
4	6805	8000
5	6531	8000
6	6425	8000
7	6431	8000
8	6779	8000
9	6031	8000
10	6305	8000
Average (lbs)	6501	
Standard Devation (lbs)	265	
Minimum (lbs)	6031	

Table 9 - Semi-rigid frame results

Non-linear frame analysis					
Group	Member	GA (Saka, 2003)		HS	
		Rigid	Extended end plate	Rigid	Extended end plate
1	Column	W24X55	W18X36	W16X26	W18X40
2	Column	W18X35	W24X68	W21X68	W18X50
3	Column	W16X31	W14X26	W12X30	W16X31
4	Column	W18X35	W24X68	W8X28	W10X33
5	Column	W12X40	W8X18	W10X17	W14X22
6	Column	W12X35	W18X35	W8X31	W8X21
7	Beam	W16X26	W16X26	W16X31	W12X26
Weight (lbs)		7404	7092	6461	6031

Table 10 - Frame optimization comparison

weight of 14.9% for the semi-rigid frame and 12.7% for the rigid frame structure. The results could be improved upon with more efficient harmony search code to allow for a great amount of iterations.

The code was once again implemented for damage tolerance optimization. Due to the extreme number of constraints present in a damage tolerant optimization of the frame structure only one analysis was performed. The same formulation used in the damage tolerant truss example was used for the frame optimization. The initial conditions from the previous frame optimization problem were replicated for the damage tolerant truss. A reduction in member stiffness was used to simulate damage to individual members. In this example members were reduced to 50% of initial stiffness. The loading for the damage structure was reduced by 50% to give a residual factor, R_2 of 0.50. The damage tolerant simulation run time took approximately 25 hours. Once again, a more efficient use of code could greatly reduce computational time needed. The damage tolerant frame was found to have an optimal weight of 7059.9 lbs. This represents a weight increase of about 17% over the non-damage tolerant design.

6.3 Damage Tolerance and Reliability

Using the previously stated reliability formulation, the five bar truss shown in figure 22 will be analyzed for reliability. As mentioned earlier, this truss configuration has been optimized for several loading conditions in another report. The member areas used for this problem are highlighted in table 11. The material properties of the structure resemble brittle behavior with a compressive stress limit of -10 ksi and a tensile stress limit of 20 ksi. The modulus of elasticity was deemed deterministic and fixed at 29,000 ksi. The truss has an ultimate capacity of 8.256 kips and a residual capacity of 4.128 kips. This allows the structure to support a load of 4.128 kips after the complete loss of any member. The random variables used for

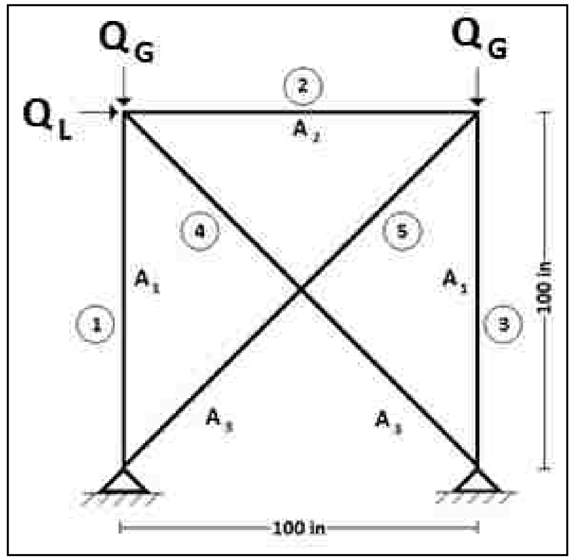


Figure 22 - 5 bar truss

Optimization Based On Ultimate Capacity and Residual Capacity						
Ultimate Capactiy Cu (ksi)	Residual Capactiy Cr (ksi)	Area (in2)			Volume (in3)	Residual Factor R2
		A1	A2	A3		
8.256	2.477	0.951	0.248	0.99	494.925	0.300
8.256	3.302	0.969	0.33	0.965	499.737	0.400
8.256	3.535	0.969	0.335	0.964	500.069	0.428
8.256	4.128	0.946	0.413	1.168	560.609	0.500
8.807	4.954	0.991	0.495	1.401	643.951	0.563
10.274	5.779	1.156	0.578	1.635	751.199	0.562
11.743	6.605	1.321	0.661	1.868	858.583	0.562

Table 12 - 5 bar optimization results

Random Variables	Distribution Type	Mean Value (ksi)	Coefficient of Variation
σ_1	Lognormal	20	0.11
σ_2	Lognormal	20	0.11
σ_3	Lognormal	20	0.11
σ_4	Lognormal	20	0.11
σ_5	Lognormal	20	0.11
Q_g	Normal	1 to 8.256	0.2
Q_l	Normal	1 to 8.256	0.2

Table 11 - Random variables

this example can be seen in table 12.

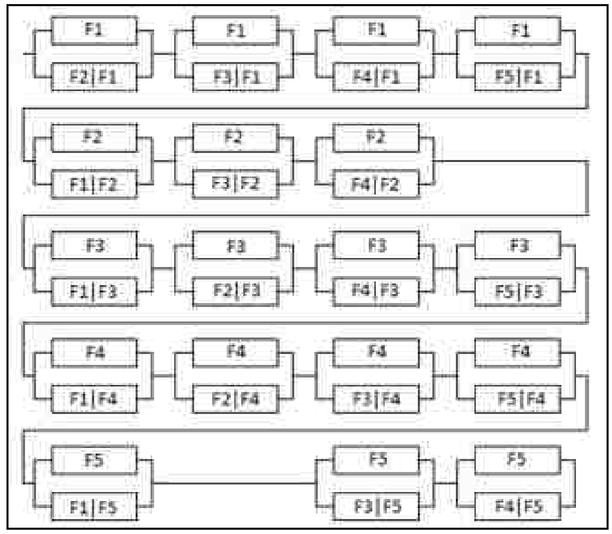


Figure 23 - Series parallel model

To determine the system reliability, there needs to be a mathematical model representing the behavior of the system and the relationship of its components with respect to the overall system. This is accomplished by considering all possible failure modes present to the structure. The five bar truss is statically indeterminate to the first degree. This results in the configuration needing two members to fail to have a complete structural failure. From this, we can create a series-parallel model for the truss. In this figure, the failure of each individual member is represented by the term $F(i)$. With the intersecting probabilities being represented by $F(i)|F(j)$, which means the failure of member "i" given member "j" has already failed. In total, we should have twenty-five failure paths for this structure. However, when a member is removed, forces throughout the system are redistributed. This can be seen in the removal of member 2, which results in no force present in the fifth member. A zero force member is present once again after the removal of the fifth member. This reduces the total number of failure paths by two with giving a total of twenty-three as shown in figure 23. [2]

The truss reliability was computed using RELSYS (RELiability of SYStems), a FORTRAN 77 computer program developed by Estes and Frangopol in (1998). The program works by first computing the reliability of all the system components in a given series-parallel system. The system is then continuously reduced to equivalent components until it is left with one component for the entire system. Series and parallel events are solved separately and equivalent alpha vectors are used to account for the correlation between failure events. [18]

For the truss series parallel system shown, 19 reductions were needed to find the overall system reliability. First, the 18 parallel failure events shown were reduced to a corresponding equivalent event, then the 18 equivalent failure events are represented in a series configuration, which was reduced once again to find the overall failure event. The truss system reliability index and failure probability for several loading magnitudes can be seen in figure 24 and 25 respectively.

As expected probability of the system failing under the 4.128 kip loading was extremely small, so small it can be considered as zero. This trend continued up until a loading of 5

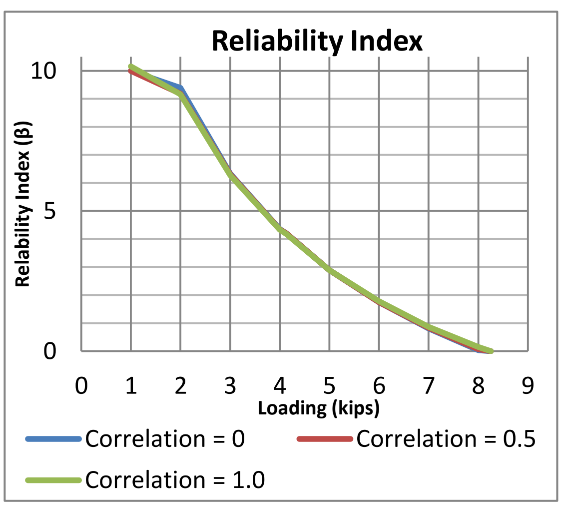


Figure 24 - Reliability index

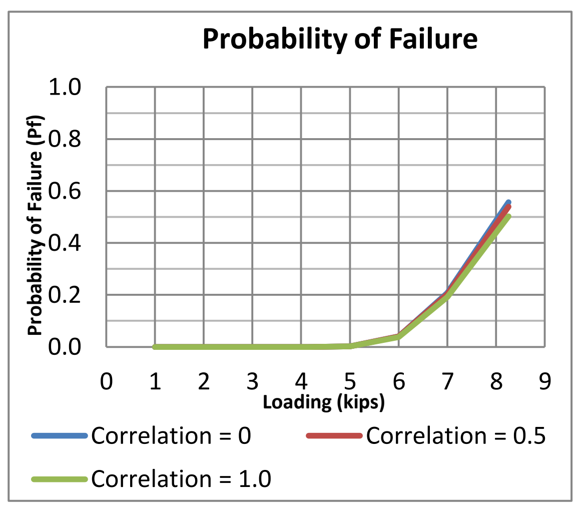


Figure 25 - Probability of failure

kips. Once the loading surpassed the 5-kip threshold, the probability of failure starts to rise. The ultimate load of the structure at 8.256 kips had a probability of failure right around 50% and a corresponding reliability index of 0.0.

Correlation between random variables will affect the overall truss reliability. Correlation between the resistances was varied from uncorrelated ($\rho_{\sigma_i,\sigma_j} = 0.0$), 50% correlated ($\rho_{\sigma_i,\sigma_j} = 0.5$) and fully correlated ($\rho_{\sigma_i,\sigma_j} = 1.0$). Results for each case were plotted and are the results were relatively similar for each correlation case. Full correlation between stresses resulted in the highest reliability index whereas the uncorrelated results gave the lowest reliability index. The effects of correlation between other random variables for the system could also be investigated. This shows the importance of accurately representing the problem data to achieve proper reliability results.

6.3.1 Effects of Damage

There are several definitions of structural damage. The term can be defined as any strength deficiency introduced during the design or construction phase of the structure as well as any deterioration of strength caused by external loading and/or environmental conditions during

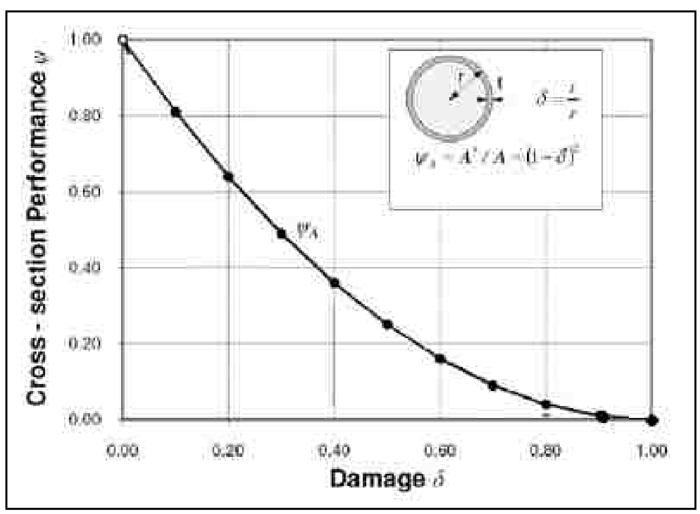


Figure 26 - Damage effects

the lifetime of the structure. For this example, we will investigate the effects of local damage to specific truss members. The damage has been classified using a damage index associated with the progressive deterioration of the member properties (area). This damage index can range from $0 \leq \delta \leq 1$, with zero representing no damage and one representing complete loss of member. The relationship between cross sectional performance and the damage index relationship can be seen in figure26. For a circular cross-section undergoing uniform damage on the external boundary, the initial area will be reduced by the following equation:

$$A_D = (1 - \delta)^2 * A_I \quad 77$$

$$\delta = t/r \quad 78$$

where, A_D is the damaged cross-sectional area and A_I is the initial area.

Using this representation of damage, each member of the truss was subjected to the full range of damage and the corresponding reliability index was calculated, figure 27. The loading on the truss was considered fixed at 3 kips for the damage conditions. This will ensure that the reliability index of the system will be greater than zero. It can be seen that damage to members 1 and 2 result in

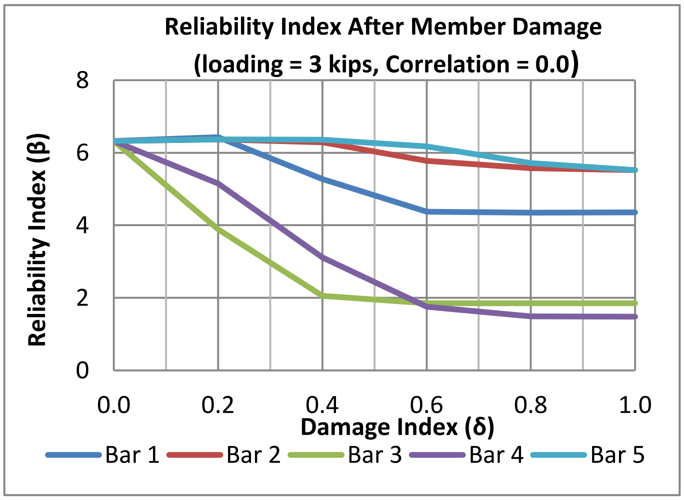


Figure 27 - Damage reliability index

small reduction to the reliability index. Actually, when these members are only slightly damaged the reliability index increases slightly due to the loadings being redistributed to other members. The members that are of interest would be members 3 and 4 since damage to these members cause a significant reduction in the reliability index. It is interesting to note that member 3 should be the first member to fail in the system; however, it does not have the lowest reliability index due to damage. Damage to member 4 actually results in the greatest reduction to the reliability index. This is due to the redistribution of loads when member 4 is removed. The removal of the fourth member puts member 2 and 3 into a significant amount of axial compression resulting in higher failure probabilities for these members. Whereas, the removal of member 3 only puts member 4 into a large amount of axial compress.

6.3.2 Measure of Redundancy

Several methods for the quantification of structural redundancy are presented in Frangopol and Curley (1987) and Fu and Frangopol (1990) [2]. The method adapted for this thesis was the probabilistic redundant index approach. This can be expressed by the following equations:

$$\beta_R = \frac{\beta_{intact}}{(\beta_{intact} - \beta_{damaged})} \qquad 79$$

where, β_{intact} represents the reliability index of the intact system; and β_{damage} represents the reliability index for the damaged system. The probabilistic redundant index β_R varies from zero to infinity. With zero indicating a structural collapse and infinity an intact structure. The probabilistic redundancy index for this problem can be seen in figure 28.

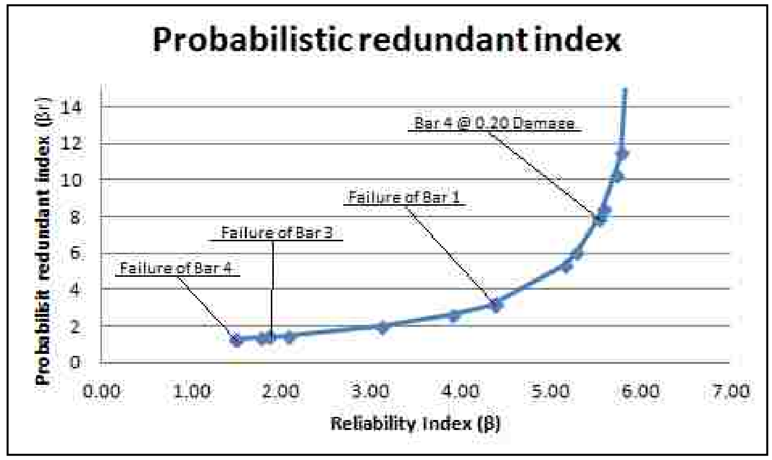


Figure 28 - Redundancy index

An approach to calculating component and system reliability of trusses has been presented. The techniques to quantify and account random variables, redundancy and damage are covered. The optimization methods used in previous work provided good results and correlated with the findings of this thesis. Probabilistic concepts should be utilized when dealing with unknown variables and behavior of the structure needs to be looked at beyond single-element failures by looking at complete structural failure.

Chapter 7: Conclusion

The goal of this research to develop a computer model capable of optimizing the weight of various steel structures has been presented. The recent developments in meta-heuristic optimization algorithms have provided researchers with a wide variety of acceptable methods for optimization. The harmony search algorithm proved to be a well suitable approach for structural optimization. Due to its stochastic random searches, derivative information is unnecessary which allows for the algorithm to easily be implemented. Further research is being done to improve upon this relatively new technique. New approaches have the algorithm constantly changing the search parameters in real time during the optimization process allowing for a more successful code.

The optimum design of steel structures using harmony search algorithm has provided three minimum weight structures. This technique can be very beneficial to both clients and designers from a cost standpoint. The ability to provide a minimum weight deign can be correlated to a reduced cost of the structural system. The designer can also consider damage tolerance in his/her design with minimal change to the original coding. As seen a minimal amount of weight increase could lead to improved structural performance.

Further work could be done to improve the harmony search coding algorithm to increase speed and performance. Also, damage conditions considered in these examples were simplistic. Therefore, further research could be performed to provide a more accurate representation of damage.

Chapter 8: Researcher's Biography

R. Bryan Peiffer was born on April 9, 1988 in Harrisburg, Pennsylvania to Scott and Tracy Peiffer. After graduating from Red Land High School, Bryan moved on to attend The Pennsylvania State University to study Architectural Engineering. During his time at Penn State, he became more interested in the studies of building systems, specifically structural systems. He chose to pursue the Structural Option within the Architectural Engineering Program. After graduation from Penn State, Bryan decided to continue his education in the structural field and pursue a Master's of Science in Structural Engineering from Lehigh University.

Bryan currently resides in Hackensack, NJ with his fiancée, Carrie Landis. In early 2014, Bryan ventured out of academia and into the professional world of engineering. He is currently a full time Jr. Engineer at McLaren Engineering. He hopes to continue with his professional development and work towards receiving his Professional Engineering license in the future.

References

1. Gordon (1978), *Structures: Or, Why Things Don't Fall down*. New York: Plenum, Print.
2. Frangopol and Curley (1987): "Effects of Damage and Redundancy on Structural Reliability." *Journal of Structural Engineering* 113.7 : 1533. Print.
3. American Institute of Steel Construction, Manual of Steel Construction, 13th Edition. Chicago: AISC, 2005.
4. Geschwindner (2008) *Unified Design of Steel Structures.* Hoboken, NJ: Wiley, Print.
5. Frye and Morris. "Analysis of Flexibly Connected Steel Frames." *Canadian Journal of Civil Engineering* 2.3 (1975): 280-91. Print.
6. Chen and Lui (2005) *Handbook of Structural Engineering.* 2nd ed. Boca Raton, FL: CRC, Print.
7. Pezeshk et al. (2000) "Design of Nonlinear Framed Structures using Genetic Optimization." *Journal of Structural Engineering* 126.3382. Print.
8. Hayalioglu (2001) "Optimum Load and Resistance Factor Design of Steel Space Frames Using Genetic Algorithm.: *Structural and Multidisciplinary Optimization* 21.4 292-99. Print
9. Hayalioglu and Degertekin (2005) "Minimum Cost Design of Steel Frames with Semi-rigid Connections and Column bases via Genetic Optimization." *Computers & Structures* 83.21-22 1849-863. Print

10. Lee and Geem (2005) "A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice." *Computer Methods in Applied Mechanics and Engineering* 194.36-38, 3902-933. Print.
11. Saka (2009) "Optimum Design of Steel Sway Frames to BS5950 Using Harmony Search Algorithm." *Journal of Constructional Steel Research* 65.1, 36-43. Print.
12. Geem (2009). *Harmony Search Algorithms for Structural Design Optimization*. Berlin: Springer, Print.
13. Gongkang and Frangopol (1990) "Reliability-Based Vector Optimization of Structural Systems." *Journal of Structural Engineering* 116.8, 2143. Print.
14. Holland (1975) *Adaptation in Natural and Artificial Systems: An Introd. Analysis with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor: Univ. of Michigan Pr., Print.
15. Metropolis et al. (1953) "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics* 21.6, 1087. Print.
16. Dorigo et al. (1992) "An investigation of some properties of an ant algorithm." *In: Proceedings of the Parallel Problem Solving from Nature Conference* (PPSN 1992), Brussels, Belgium.
17. Ang et al. (2007) *Probability Concepts in Engineering: Emphasis on Applications in Civil & Environmental Engineering*. New York: Wiley, Print.
18. Estes (1997) *A system reliability approach to the lifetime optimization of inspection and repair of highway bridges.* Ph. D. University of Colorado. United States

19. Hess et al. (2002). "Uncertainties in Material and Geometric Strength and Load Variables." *Naval Engineers Journal* 114.2 139-66. Print.

20. Frangopol and Klisinksi (1991) Damage-tolerant design of deterministic and nondeterministic structural systems. *Proceedings of the ASCE Tenth Conference on Electronic Computation.* Indianapolis, Indian, April 29-May 1

21. Mistree (1983) Design of damage tolerant structural systems. Engineering Optimization Vol. 6, 141-144.

22. Sun et al. (1976) "Fail-Safe Optimal Design Of Structures." *Engineering Optimization* 2.1 43-53. Print.

23. Arora, J. et al.. (1980). Optimal design of large structures for damage tolerance. *The American Institute of Aeronautics and Astronautics, Vol. 118*, 563-570.

24. Shupe and Mistree. (1987). Compromise: an effective approach for the design of damage tolerant structural systems. *Computers & Structures, Vol. 27,* No. 3, pp. 406-415.

25. Frangopol et al. (1991). Optimization of damage-tolerant structural systems. *Computers and Structures, Vol. 40*, No. 5, 1084-1095.

26. Frangopol and Klisinksi (1989). Weight-strength-redundancy interaction in optimum design of three-dimensional brittle-ductile trusses. *Computers & Structures, Vol. 31,* No. 5, pp 775-787.

27. Frangopol and Klisinksi (1989). Material Behavior and optimum design of structural systems. *Journal of structural engineering, ASCE, Vol. 115,* No.5, pp 1054-1075.

28. Christensen and Klarbring, (2009) *An Introduction to Structural Optimization*. [Dordrecht]: Springer

29. Yang (2010) *Engineering Optimization: An Introduction with Metaheuristic Applications*. Hoboken, NJ: John Wiley

30. Shupe (1987) "Compromise: An Effective Approach for the Design of Damage Tolerant Structural Systems." *Computers & Structures* 27.3 407-15.

31. Frangopol (1989) "Weight-strength-redundancy Interaction in Optimum Design of Three-dimensional Brittle-ductile Trusses." *Computers & Structures* 31.5 77587.

32. Frangopol and Klisinski (1989) "Material Behavior and Optimum Design of Structural Systems." *Journal of Structural Engineering* 115.5, 1054.

33. Dobbs and Nelson (1976) "Application of Optimality Criteria to Automated Structural Design." *AIAA Journal* 14.10, 1436-443.

34. Khalifa (2011) *Design Optimization of semi rigid steel framed structures to AISC-LRFD using Harmony search algorithm.* M.S. University of Gaza. Palestine.

35. John (1987) "Minimum Weight Design of Trusses Using Improved Move Limit Method of Sequential Linear Programming." *Computers & Structures* 27.5, 583-91.

36. Khan et al. (1979) "An Optimality Criterion Method for Large-Scale Structures." *AIAA Journal* 17.7, 753-61.

37. Lamberti (2003) "Move Limits Definition in Structural Optimization with Sequential Linearv Programming. Part II: Numerical Examples." *Computers & Structures* 81.4 ,215-38.

38. Schmit and Farshi (1974): "Some Approximation Concepts for Structural Synthesis." *AIAA Journal* 12.5, 692-99.

39. Sunar and Belegundu (1991) "Trust Region Methods for Structural Optimization Using Exact Second Order Sensitivity." *International Journal for Numerical Methods in Engineering* 32.2, 275-93.

40. Venkayya (1971) "Design of Optimum Structures." *Computers & Structures* 1.1-2 265-309.

41. Xu and Grandhi (1998) "Effective Two-Point Function Approximation for Design Optimization." *AIAA Journal* 36.122269-275.

Appendix A - W-Shape Selection List

		W	A	d	b_f	t_w	t_f	$b_f/2t_f$	h/t_w	I_x	Z_x	S_x	r_x	I_y	Z_y	S_y	r_y	J	C_w
1	W40X199	199	58.8	38.7	15.8	0.650	1.07	7.39	52.6	14900	869	770	16.0	695	137	88.2	3.45	18.3	246000
2	W27X194	194	57.1	28.1	14.0	0.750	1.34	5.24	31.8	7860	631	559	11.7	619	136	88.1	3.29	27.1	111000
3	W36X194	194	57.0	36.5	12.1	0.765	1.26	4.81	42.4	12100	767	664	14.6	375	97.7	61.9	2.56	22.2	116000
4	W14X193	193	56.8	15.5	15.7	0.890	1.44	5.45	12.8	2400	355	310	6.50	931	180	119	4.05	34.8	45900
5	W24X192	192	56.5	25.5	13.0	0.810	1.46	4.43	26.6	6260	559	491	10.5	530	126	81.8	3.07	30.8	76300
6	W18X192	192	56.2	20.4	11.5	0.960	1.75	3.27	16.7	3870	442	380	8.28	440	119	76.8	2.79	44.7	38000
7	W30X191	191	56.1	30.7	15.0	0.710	1.19	6.35	37.7	9200	675	600	12.8	673	138	89.5	3.46	21.0	146000
8	W12X190	190	56.0	14.4	12.7	1.06	1.74	3.65	9.16	1890	311	263	5.82	589	143	93.0	3.25	48.8	23600
9	W36X182	182	53.6	36.3	12.1	0.725	1.18	5.12	44.8	11300	718	623	14.5	347	90.7	57.6	2.55	18.5	107000
10	W21X182	182	53.6	22.7	12.5	0.830	1.48	4.22	22.6	4730	476	417	9.40	483	119	77.2	3.00	30.7	54400
11	W40X183	183	53.3	39.0	11.8	0.650	1.20	4.92	52.6	13200	774	675	15.7	331	88.3	56.0	2.49	19.3	118000
12	W27X178	178	52.5	27.8	14.1	0.725	1.19	5.92	32.9	7020	570	505	11.6	555	122	78.8	3.25	20.1	98400
13	W14X176	176	51.8	15.2	15.7	0.830	1.31	5.97	13.7	2140	320	281	6.43	838	163	107	4.02	26.5	40500
14	W24X176	176	51.7	25.2	12.9	0.750	1.34	4.81	28.7	5680	511	450	10.5	479	115	74.3	3.04	23.9	68400
15	W18X175	175	51.4	20.0	11.4	0.890	1.59	3.58	18.0	3450	398	344	8.20	391	106	68.8	2.76	33.8	33300
16	W30X173	173	50.9	30.4	15.0	0.655	1.07	7.04	40.8	8230	607	541	12.7	598	123	79.8	3.42	15.6	129000
17	W36X170	170	50.0	36.2	12.0	0.680	1.10	5.47	47.7	10500	668	581	14.5	320	83.8	53.2	2.53	15.1	98500
18	W12X170	170	50.0	14.0	12.6	0.960	1.56	4.03	10.1	1650	275	235	5.74	517	126	82.3	3.22	35.6	20100
19	W33X169	169	49.5	33.8	11.5	0.670	1.22	4.71	44.7	9290	629	549	13.7	310	84.4	53.9	2.50	17.7	82400
20	W40X167	167	49.3	38.6	11.8	0.650	1.03	5.76	52.6	11600	693	600	15.3	283	76.0	47.9	2.40	14.0	99700
21	W21X166	166	48.8	22.5	12.4	0.750	1.36	4.57	25.0	4280	432	380	9.36	435	108	70.0	2.99	23.6	48500
22	W24X162	162	47.8	25.0	13.0	0.705	1.22	5.31	30.6	5170	468	414	10.4	443	105	68.4	3.05	18.5	62600
23	W27X161	161	47.6	27.6	14.0	0.660	1.08	6.49	36.1	6310	515	458	11.5	497	109	70.9	3.23	15.1	87300
24	W36X160	160	47.0	36.0	12.0	0.650	1.02	5.88	49.9	9760	624	542	14.4	295	77.3	49.1	2.50	12.4	90200
25	W14X159	159	46.7	15.0	15.6	0.745	1.19	6.54	15.3	1900	287	254	6.38	748	146	96.2	4.00	19.7	35600
26	W18X158	158	46.3	19.7	11.3	0.810	1.44	3.92	19.8	3060	356	310	8.12	347	94.8	61.4	2.74	25.2	29000
27	W33X152	152	44.9	33.5	11.6	0.635	1.06	5.48	47.2	8160	559	487	13.5	273	73.9	47.2	2.47	12.4	71700
28	W12X152	152	44.7	13.7	12.5	0.870	1.40	4.46	11.2	1430	243	209	5.66	454	111	72.8	3.19	25.8	17200
29	W36X150	150	44.3	35.9	12.0	0.625	0.940	6.37	51.9	9040	581	504	14.3	270	70.9	45.1	2.47	10.1	82200
30	W40X149	149	43.8	38.2	11.8	0.630	0.830	7.11	54.3	9800	598	513	15.0	229	62.2	38.8	2.29	9.36	80000
31	W30X148	148	43.6	30.7	10.5	0.650	1.18	4.44	41.6	6680	500	436	12.4	227	68.0	43.3	2.28	14.5	49400
32	W27X146	146	43.2	27.4	14.0	0.605	0.975	7.16	39.4	5660	464	414	11.5	443	97.7	63.5	3.20	11.3	77200
33	W21X147	147	43.2	22.1	12.5	0.720	1.15	5.44	26.1	3630	373	329	9.17	376	92.6	60.1	2.95	15.4	41100
34	W24X146	146	43.0	24.7	12.9	0.650	1.09	5.92	33.2	4580	418	371	10.3	391	93.2	60.5	3.01	13.4	54600
35	W14X145	145	42.7	14.8	15.5	0.680	1.09	7.11	16.8	1710	260	232	6.33	677	133	87.3	3.98	15.2	31700
36	W18X143	143	42.0	19.5	11.2	0.730	1.32	4.25	22.0	2750	322	282	8.09	311	85.4	55.5	2.72	19.2	25700

37	W33X141	141	41.5	33.3	11.5	0.605	0.960	6.01	49.6	7450	514	448	13.4	246	66.9	42.7	2.43	9.70	64400
38	W36X135	135	39.9	35.6	12.0	0.600	0.790	7.56	54.1	7800	509	439	14.0	225	59.7	37.7	2.38	7.00	68100
39	W12X136	136	39.9	13.4	12.4	0.790	1.25	4.96	12.3	1240	214	186	5.58	398	98.0	64.2	3.16	18.5	14700
40	W30X132	132	38.8	30.3	10.5	0.615	1.00	5.27	43.9	5770	437	380	12.2	196	58.4	37.2	2.25	9.72	42100
41	W21X132	132	38.8	21.8	12.4	0.650	1.04	6.01	28.9	3220	333	295	9.12	333	82.3	53.5	2.93	11.3	36000
42	W14X132	132	38.8	14.7	14.7	0.645	1.03	7.15	17.7	1530	234	209	6.28	548	113	74.5	3.76	12.3	25500
43	W24X131	131	38.6	24.5	12.9	0.605	0.960	6.70	35.6	4020	370	329	10.2	340	81.5	53.0	2.97	9.50	47100
44	W33X130	130	38.3	33.1	11.5	0.580	0.855	6.73	51.7	6710	467	406	13.2	218	59.5	37.9	2.39	7.37	56600
45	W18X130	130	38.3	19.3	11.2	0.670	1.20	4.65	23.9	2460	290	256	8.03	278	76.7	49.9	2.70	14.5	22700
46	W27X129	129	37.8	27.6	10.0	0.610	1.10	4.55	39.7	4760	395	345	11.2	184	57.6	36.8	2.21	11.1	32500
47	W30X124	124	36.5	30.2	10.5	0.585	0.930	5.65	46.2	5360	408	355	12.1	181	54.0	34.4	2.23	7.99	38600
48	W21X122	122	35.9	21.7	12.4	0.600	0.960	6.45	31.3	2960	307	273	9.09	305	75.6	49.2	2.92	8.98	32700
49	W14X120	120	35.3	14.5	14.7	0.590	0.940	7.80	19.3	1380	212	190	6.24	495	102	67.5	3.74	9.37	22700
50	W12X120	120	35.2	13.1	12.3	0.710	1.11	5.57	13.7	1070	186	163	5.51	345	85.4	56.0	3.13	12.9	12400
51	W18X119	119	35.1	19.0	11.3	0.655	1.06	5.31	24.5	2190	262	231	7.90	253	69.1	44.9	2.69	10.6	20300
52	W33X118	118	34.7	32.9	11.5	0.550	0.740	7.76	54.5	5900	415	359	13.0	187	51.3	32.6	2.32	5.30	48300
53	W24X117	117	34.4	24.3	12.8	0.550	0.850	7.53	39.2	3540	327	291	10.1	297	71.4	46.5	2.94	6.72	40800
54	W30X116	116	34.2	30.0	10.5	0.565	0.850	6.17	47.8	4930	378	329	12.0	164	49.2	31.3	2.19	6.43	34900
55	W27X114	114	33.6	27.3	10.1	0.570	0.930	5.41	42.5	4080	343	299	11.0	159	49.3	31.5	2.18	7.33	27600
56	W10X112	112	32.9	11.4	10.4	0.755	1.25	4.17	10.4	716	147	126	4.66	236	69.2	45.3	2.68	15.1	6020
57	W21X111	111	32.6	21.5	12.3	0.550	0.875	7.05	34.1	2670	279	249	9.05	274	68.2	44.5	2.90	6.83	29200
58	W14X109	109	32.0	14.3	14.6	0.525	0.860	8.49	21.7	1240	192	173	6.22	447	92.7	61.2	3.73	7.12	20200
59	W30X108	108	31.7	29.8	10.5	0.545	0.760	6.89	49.6	4470	346	299	11.9	146	43.9	27.9	2.15	4.99	30900
60	W12X106	106	31.2	12.9	12.2	0.610	0.990	6.17	15.9	933	164	145	5.47	301	75.1	49.3	3.11	9.13	10700
61	W18X106	106	31.1	18.7	11.2	0.590	0.940	5.96	27.2	1910	230	204	7.84	220	60.5	39.4	2.66	7.48	17400
62	W24X104	104	30.7	24.1	12.8	0.500	0.750	8.50	43.1	3100	289	258	10.1	259	62.4	40.7	2.91	4.72	35200
63	W24X103	103	30.3	24.5	9.00	0.550	0.980	4.59	39.2	3000	280	245	10.0	119	41.5	26.5	1.99	7.07	16600
64	W27X102	102	30.0	27.1	10.0	0.515	0.830	6.03	47.1	3620	305	267	11.0	139	43.4	27.8	2.15	5.28	24000
65	W21X101	101	29.8	21.4	12.3	0.500	0.800	7.68	37.5	2420	253	227	9.02	248	61.7	40.3	2.89	5.21	26200
66	W16X100	100	29.4	17.0	10.4	0.585	0.985	5.29	24.3	1490	198	175	7.10	186	54.9	35.7	2.51	7.73	11900
67	W10X100	100	29.3	11.1	10.3	0.680	1.12	4.62	11.6	623	130	112	4.60	207	61.0	40.0	2.65	10.9	5150
68	W14X99	99.0	29.1	14.2	14.6	0.485	0.780	9.34	23.5	1110	173	157	6.17	402	83.6	55.2	3.71	5.37	18000
69	W30X99	99.0	29.0	29.7	10.5	0.520	0.670	7.80	51.9	3990	312	269	11.7	128	38.6	24.5	2.10	3.77	26800
70	W18X97	97.0	28.5	18.6	11.1	0.535	0.870	6.41	30.0	1750	211	188	7.82	201	55.3	36.1	2.65	5.86	15800
71	W12X96	96.0	28.2	12.7	12.2	0.550	0.900	6.76	17.7	833	147	131	5.44	270	67.5	44.4	3.09	6.85	9410
72	W24X94	94.0	27.7	24.3	9.07	0.515	0.875	5.18	41.9	2700	254	222	9.87	109	37.5	24.0	1.98	5.26	15000
73	W27X94	94.0	27.6	26.9	10.0	0.490	0.745	6.70	49.5	3270	278	243	10.9	124	38.8	24.8	2.12	4.03	21300
74	W21X93	93.0	27.3	21.6	8.42	0.580	0.930	4.53	32.3	2070	221	192	8.70	92.9	34.7	22.1	1.84	6.03	9940
75	W14X90	90.0	26.5	14.0	14.5	0.440	0.710	10.2	25.9	999	157	143	6.14	362	75.6	49.9	3.70	4.06	16000

76	W30X90	90.0	26.3	29.5	10.4	0.470	0.610	8.52	57.5	3610	283	245	11.7	115	34.7	22.1	2.09	2.84	24000
77	W16X89	89.0	26.2	16.8	10.4	0.525	0.875	5.92	27.0	1300	175	155	7.05	163	48.1	31.4	2.49	5.45	10200
78	W10X88	88.0	26.0	10.8	10.3	0.605	0.990	5.18	13.0	534	113	98.5	4.54	179	53.1	34.8	2.63	7.53	4330
79	W12X87	87.0	25.6	12.5	12.1	0.515	0.810	7.48	18.9	740	132	118	5.38	241	60.4	39.7	3.07	5.10	8270
80	W18X86	86.0	25.3	18.4	11.1	0.480	0.770	7.20	33.4	1530	186	166	7.77	175	48.4	31.6	2.63	4.10	13600
81	W27X84	84.0	24.7	26.7	10.0	0.460	0.640	7.78	52.7	2850	244	213	10.7	106	33.2	21.2	2.07	2.81	17900
82	W24X84	84.0	24.7	24.1	9.02	0.470	0.770	5.86	45.9	2370	224	196	9.79	94.4	32.6	20.9	1.95	3.70	12800
83	W21X83	83.0	24.4	21.4	8.36	0.515	0.835	5.00	36.4	1830	196	171	8.67	81.4	30.5	19.5	1.83	4.34	8630
84	W14X82	82.0	24.0	14.3	10.1	0.510	0.855	5.92	22.4	881	139	123	6.05	148	44.8	29.3	2.48	5.07	6710
85	W12X79	79.0	23.2	12.4	12.1	0.470	0.735	8.22	20.7	662	119	107	5.34	216	54.3	35.8	3.05	3.84	7330
86	W10X77	77.0	22.7	10.6	10.2	0.530	0.870	5.86	14.8	455	97.6	85.9	4.49	154	45.9	30.1	2.60	5.11	3630
87	W16X77	77.0	22.6	16.5	10.3	0.455	0.760	6.77	31.2	1110	150	134	7.00	138	41.1	26.9	2.47	3.57	8590
88	W24X76	76.0	22.4	23.9	8.99	0.440	0.680	6.61	49.0	2100	200	176	9.69	82.5	28.6	18.4	1.92	2.68	11100
89	W18X76	76.0	22.3	18.2	11.0	0.425	0.680	8.11	37.8	1330	163	146	7.73	152	42.2	27.6	2.61	2.83	11700
90	W14X74	74.0	21.8	14.2	10.1	0.450	0.785	6.41	25.4	795	126	112	6.04	134	40.5	26.6	2.48	3.87	5990
91	W21X73	73.0	21.5	21.2	8.30	0.455	0.740	5.60	41.2	1600	172	151	8.64	70.6	26.6	17.0	1.81	3.02	7410
92	W12X72	72.0	21.1	12.3	12.0	0.430	0.670	8.99	22.6	597	108	97.4	5.31	195	49.2	32.4	3.04	2.93	6540
93	W18X71	71.0	20.9	18.5	7.64	0.495	0.810	4.71	32.4	1170	146	127	7.50	60.3	24.7	15.8	1.70	3.49	4700
94	W24X68	68.0	20.1	23.7	8.97	0.415	0.585	7.66	52.0	1830	177	154	9.55	70.4	24.5	15.7	1.87	1.87	9430
95	W21X68	68.0	20.0	21.1	8.27	0.430	0.685	6.04	43.6	1480	160	140	8.60	64.7	24.4	15.7	1.80	2.45	6760
96	W14X68	68.0	20.0	14.0	10.0	0.415	0.720	6.97	27.5	722	115	103	6.01	121	36.9	24.2	2.46	3.01	5380
97	W10X68	68.0	19.9	10.4	10.1	0.470	0.770	6.58	16.7	394	85.3	75.7	4.44	134	40.1	26.4	2.59	3.56	3100
98	W8X67	67.0	19.7	9.00	8.28	0.570	0.935	4.43	11.1	272	70.1	60.4	3.72	88.6	32.7	21.4	2.12	5.05	1440
99	W16X67	67.0	19.6	16.3	10.2	0.395	0.665	7.70	35.9	954	130	117	6.96	119	35.5	23.2	2.46	2.39	7300
100	W18X65	65.0	19.1	18.4	7.59	0.450	0.750	5.06	35.7	1070	133	117	7.49	54.8	22.5	14.4	1.69	2.73	4240
101	W12X65	65.0	19.1	12.1	12.0	0.390	0.605	9.92	24.9	533	96.8	87.9	5.28	174	44.1	29.1	3.02	2.18	5780
102	W21X62	62.0	18.3	21.0	8.24	0.400	0.615	6.70	46.9	1330	144	127	8.54	57.5	21.7	14.0	1.77	1.83	5960
103	W24X62	62.0	18.2	23.7	7.04	0.430	0.590	5.97	50.1	1550	153	131	9.23	34.5	15.7	9.80	1.38	1.71	4620
104	W14X61	61.0	17.9	13.9	10.0	0.375	0.645	7.75	30.4	640	102	92.1	5.98	107	32.8	21.5	2.45	2.19	4710
105	W10X60	60.0	17.7	10.2	10.1	0.420	0.680	7.41	18.7	341	74.6	66.7	4.39	116	35.0	23.0	2.57	2.48	2640
106	W18X60	60.0	17.6	18.2	7.56	0.415	0.695	5.44	38.7	984	123	108	7.47	50.1	20.6	13.3	1.68	2.17	3850
107	W8X58	58.0	17.1	8.75	8.22	0.510	0.810	5.07	12.4	228	59.8	52.0	3.65	75.1	27.9	18.3	2.10	3.33	1180
108	W12X58	58.0	17.0	12.2	10.0	0.360	0.640	7.82	27.0	475	86.4	78.0	5.28	107	32.5	21.4	2.51	2.10	3570
109	W16X57	57.0	16.8	16.4	7.12	0.430	0.715	4.98	33.0	758	105	92.2	6.72	43.1	18.9	12.1	1.60	2.22	2660
110	W21X57	57.0	16.7	21.1	6.56	0.405	0.650	5.04	46.3	1170	129	111	8.36	30.6	14.8	9.35	1.35	1.77	3190
111	W24X55	55.0	16.2	23.6	7.01	0.395	0.505	6.94	54.6	1350	134	114	9.11	29.1	13.3	8.30	1.34	1.18	3870
112	W21X55	55.0	16.2	20.8	8.22	0.375	0.522	7.87	50.0	1140	126	110	8.40	48.4	18.4	11.8	1.73	1.24	4980
113	W18X55	55.0	16.2	18.1	7.53	0.390	0.630	5.98	41.1	890	112	98.3	7.41	44.9	18.5	11.9	1.67	1.66	3430
114	W10X54	54.0	15.8	10.1	10.0	0.370	0.615	8.15	21.2	303	66.6	60.0	4.37	103	31.3	20.6	2.56	1.82	2320

115	W14X53	53.0	15.6	13.9	8.06	0.370	0.660	6.11	30.9	541	87.1	77.8	5.89	57.7	22.0	14.3	1.92	1.94	2540
116	W12X53	53.0	15.6	12.1	10.0	0.345	0.575	8.69	28.1	425	77.9	70.6	5.23	95.8	29.1	19.2	2.48	1.58	3160
117	W21X50	50.0	14.7	20.8	6.53	0.380	0.535	6.10	49.4	984	110	94.5	8.18	24.9	12.2	7.64	1.30	1.14	2570
118	W18X50	50.0	14.7	18.0	7.50	0.355	0.570	6.57	45.2	800	101	88.9	7.38	40.1	16.6	10.7	1.65	1.24	3040
119	W16X50	50.0	14.7	16.3	7.07	0.380	0.630	5.61	37.4	659	92.0	81.0	6.68	37.2	16.3	10.5	1.59	1.52	2270
120	W12X50	50.0	14.6	12.2	8.08	0.370	0.640	6.31	26.8	391	71.9	64.2	5.18	56.3	21.3	13.9	1.96	1.71	1880
121	W10X49	49.0	14.4	10.0	10.0	0.340	0.560	8.93	23.1	272	60.4	54.6	4.35	93.4	28.3	18.7	2.54	1.39	2070
122	W21X48	48.0	14.1	20.6	8.14	0.350	0.430	9.47	53.6	959	107	93.0	8.24	38.7	14.9	9.52	1.66	0.803	3950
123	W14X48	48.0	14.1	13.8	8.03	0.340	0.595	6.75	33.6	484	78.4	70.2	5.85	51.4	19.6	12.8	1.91	1.45	2240
124	W8X48	48.0	14.1	8.50	8.11	0.400	0.685	5.92	15.9	184	49.0	43.2	3.61	60.9	22.9	15.0	2.08	1.96	931
125	W18X46	46.0	13.5	18.1	6.06	0.360	0.605	5.01	44.6	712	90.7	78.8	7.25	22.5	11.7	7.43	1.29	1.22	1720
126	W16X45	45.0	13.3	16.1	7.04	0.345	0.565	6.23	41.1	586	82.3	72.7	6.65	32.8	14.5	9.34	1.57	1.11	1990
127	W10X45	45.0	13.3	10.1	8.02	0.350	0.620	6.47	22.5	248	54.9	49.1	4.32	53.4	20.3	13.3	2.01	1.51	1200
128	W12X45	45.0	13.1	12.1	8.05	0.335	0.575	7.00	29.6	348	64.2	57.7	5.15	50.0	19.0	12.4	1.95	1.26	1650
129	W21X44	44.0	13.0	20.7	6.50	0.350	0.450	7.22	53.6	843	95.4	81.6	8.06	20.7	10.2	6.37	1.26	0.770	2110
130	W14X43	43.0	12.6	13.7	8.00	0.305	0.530	7.54	37.4	428	69.6	62.6	5.82	45.2	17.3	11.3	1.89	1.05	1950
131	W18X40	40.0	11.8	17.9	6.02	0.315	0.525	5.73	50.9	612	78.4	68.4	7.21	19.1	10.0	6.35	1.27	0.810	1440
132	W16X40	40.0	11.8	16.0	7.00	0.305	0.505	6.93	46.5	518	73.0	64.7	6.63	28.9	12.7	8.25	1.57	0.794	1730
133	W12X40	40.0	11.7	11.9	8.01	0.295	0.515	7.77	33.6	307	57.0	51.5	5.13	44.1	16.8	11.0	1.94	0.906	1440
134	W8X40	40.0	11.7	8.25	8.07	0.360	0.560	7.21	17.6	146	39.8	35.5	3.53	49.1	18.5	12.2	2.04	1.12	726
135	W10X39	39.0	11.5	9.92	7.99	0.315	0.530	7.53	25.0	209	46.8	42.1	4.27	45.0	17.2	11.3	1.98	0.976	992
136	W14X38	38.0	11.2	14.1	6.77	0.310	0.515	6.57	39.6	385	61.5	54.6	5.87	26.7	12.1	7.88	1.55	0.798	1230
137	W16X36	36.0	10.6	15.9	6.99	0.295	0.430	8.12	48.1	448	64.0	56.5	6.51	24.5	10.8	7.00	1.52	0.545	1460
138	W18X35	35.0	10.3	17.7	6.00	0.300	0.425	7.06	53.5	510	66.5	57.6	7.04	15.3	8.06	5.12	1.22	0.506	1140
139	W12X35	35.0	10.3	12.5	6.56	0.300	0.520	6.31	36.2	285	51.2	45.6	5.25	24.5	11.5	7.47	1.54	0.741	879
140	W8X35	35.0	10.3	8.12	8.02	0.310	0.495	8.10	20.5	127	34.7	31.2	3.51	42.6	16.1	10.6	2.03	0.769	619
141	W14X34	34.0	10.0	14.0	6.75	0.285	0.455	7.41	43.1	340	54.6	48.6	5.83	23.3	10.6	6.91	1.53	0.569	1070
142	W10X33	33.0	9.71	9.73	7.96	0.290	0.435	9.15	27.1	171	38.8	35.0	4.19	36.6	14.0	9.20	1.94	0.583	791
143	W16X31	31.0	9.13	15.9	5.53	0.275	0.440	6.28	51.6	375	54.0	47.2	6.41	12.4	7.03	4.49	1.17	0.461	739
144	W8X31	31.0	9.13	8.00	8.00	0.285	0.435	9.19	22.3	110	30.4	27.5	3.47	37.1	14.1	9.27	2.02	0.536	530
145	W14X30	30.0	8.85	13.8	6.73	0.270	0.385	8.74	45.4	291	47.3	42.0	5.73	19.6	8.99	5.82	1.49	0.380	887
146	W10X30	30.0	8.84	10.5	5.81	0.300	0.510	5.70	29.5	170	36.6	32.4	4.38	16.7	8.84	5.75	1.37	0.622	414
147	W12X30	30.0	8.79	12.3	6.52	0.260	0.440	7.41	41.8	238	43.1	38.6	5.21	20.3	9.56	6.24	1.52	0.457	720
148	W8X28	28.0	8.25	8.06	6.54	0.285	0.465	7.03	22.3	98.0	27.2	24.3	3.45	21.7	10.1	6.63	1.62	0.537	312
149	W14X26	26.0	7.69	13.9	5.03	0.255	0.420	5.98	48.1	245	40.2	35.3	5.65	8.91	5.54	3.55	1.08	0.358	405
150	W16X26	26.0	7.68	15.7	5.50	0.250	0.345	7.97	56.8	301	44.2	38.4	6.26	9.59	5.48	3.49	1.12	0.262	565
151	W12X26	26.0	7.65	12.2	6.49	0.230	0.380	8.54	47.2	204	37.2	33.4	5.17	17.3	8.17	5.34	1.51	0.300	607
152	W10X26	26.0	7.61	10.3	5.77	0.260	0.440	6.56	34.0	144	31.3	27.9	4.35	14.1	7.50	4.89	1.36	0.402	345
153	W8X24	24.0	7.08	7.93	6.50	0.245	0.400	8.12	25.9	82.7	23.1	20.9	3.42	18.3	8.57	5.63	1.61	0.346	259

154	W14X22	22.0	6.49	13.7	5.00	0.230	0.335	7.46	53.3	199	33.2	29.0	5.54	7.00	4.39	2.80	1.04	0.208	314
155	W10X22	22.0	6.49	10.2	5.75	0.240	0.360	7.99	36.9	118	26.0	23.2	4.27	11.4	6.10	3.97	1.33	0.239	275
156	W12X22	22.0	6.48	12.3	4.03	0.260	0.425	4.74	41.8	156	29.3	25.4	4.91	4.66	3.66	2.31	0.848	0.293	164
157	W8X21	21.0	6.16	8.28	5.27	0.250	0.400	6.59	27.5	75.3	20.4	18.2	3.49	9.77	5.69	3.71	1.26	0.282	152
158	W10X19	19.0	5.62	10.2	4.02	0.250	0.395	5.09	35.4	96.3	21.6	18.8	4.14	4.29	3.35	2.14	0.874	0.233	104
159	W12X19	19.0	5.57	12.2	4.01	0.235	0.350	5.72	46.2	130	24.7	21.3	4.82	3.76	2.98	1.88	0.822	0.180	131
160	W8X18	18.0	5.26	8.14	5.25	0.230	0.330	7.95	29.9	61.9	17.0	15.2	3.43	7.97	4.66	3.04	1.23	0.172	122
161	W10X17	17.0	4.99	10.1	4.01	0.240	0.330	6.08	36.9	81.9	18.7	16.2	4.05	3.56	2.80	1.78	0.845	0.156	85.1
162	W12X16	16.0	4.71	12.0	3.99	0.220	0.265	7.53	49.4	103	20.1	17.1	4.67	2.82	2.26	1.41	0.773	0.103	96.9
163	W8X15	15.0	4.44	8.11	4.02	0.245	0.315	6.37	28.1	48.0	13.6	11.8	3.29	3.41	2.67	1.70	0.876	0.137	51.8
164	W10X15	15.0	4.41	9.99	4.00	0.230	0.270	7.41	38.5	68.9	16.0	13.8	3.95	2.89	2.30	1.45	0.810	0.104	68.3
165	W12X14	14.0	4.16	11.9	3.97	0.200	0.225	8.82	54.3	88.6	17.4	14.9	4.62	2.36	1.90	1.19	0.753	0.0704	80.4
166	W8X13	13.0	3.84	7.99	4.00	0.230	0.255	7.84	29.9	39.6	11.4	9.91	3.21	2.73	2.15	1.37	0.843	0.0871	40.8
167	W10X12	12.0	3.54	9.87	3.96	0.190	0.210	9.43	46.6	53.8	12.6	10.9	3.90	2.18	1.74	1.10	0.785	0.0547	50.9
168	W8X10	10.0	2.96	7.89	3.94	0.170	0.205	9.61	40.5	30.8	8.87	7.81	3.22	2.09	1.66	1.06	0.841	0.0426	30.9

Appendix B - Harmony Search Sample Frame MATLAB Optimization Code

```
clear all; clc; close all;
tic;
[numerics, strings]=xlsread('Full Catalog Section');
FCS=numerics(:,3:20);
IbeamStr=strings(2:169,1);

NVAR=7;                         % number of variables
NG=36;                          % number of inegualty constraints
NH=0;                           % number of egualty constraints
MaxItr=8000;                    % maximum number of iterations
HMS=25;                         % harmony memory size
HMCR=0.9;                       % harmony consideration rate  0< HMCR <1
PAR=0.45;                       % minumum pitch adjusting rate


ro=0.28359924220274; %density of steel lb/in3
for kk=1:10
% initialize random HM
for i=1:HMS
    for j=1:NVAR;
        k=randi(length(IbeamStr));
        HM(i,j)=IbeamStr(k);
    end
    for j=1:NVAR
        Z=strcmp(IbeamStr,HM(i,j));
        [r,c]=find(Z);
    end
    for j=1:NVAR;
    Z=strcmp(IbeamStr,HM(i,j));
    [r,c]=find(Z);
    HMnew(i,j)=r;
    end

area=[FCS(HMnew(i,1),2);FCS(HMnew(i,2),2);FCS(HMnew(i,3),2);FCS(HMnew(
i,4),2);FCS(HMnew(i,5),2);FCS(HMnew(i,6),2);FCS(HMnew(i,7),2)];

depth=[FCS(HMnew(i,1),3);FCS(HMnew(i,2),3);FCS(HMnew(i,3),3);FCS(HMnew
(i,4),3);FCS(HMnew(i,5),3);FCS(HMnew(i,6),3);FCS(HMnew(i,7),3)];

flange_width=[FCS(HMnew(i,1),4);FCS(HMnew(i,2),4);FCS(HMnew(i,3),4);FC
S(HMnew(i,4),4);FCS(HMnew(i,5),4);FCS(HMnew(i,6),4);FCS(HMnew(i,7),4)]
;

web_thickness=[FCS(HMnew(i,1),5);FCS(HMnew(i,2),5);FCS(HMnew(i,3),5);F
CS(HMnew(i,4),5);FCS(HMnew(i,5),5);FCS(HMnew(i,6),5);FCS(HMnew(i,7),5)
];

flange_thickness=[FCS(HMnew(i,1),6);FCS(HMnew(i,2),6);FCS(HMnew(i,3),6
```

```
);FCS(HMnew(i,4),6);FCS(HMnew(i,5),6);FCS(HMnew(i,6),6);FCS(HMnew(i,7)
,6)];

inertia=[FCS(HMnew(i,1),9);FCS(HMnew(i,2),9);FCS(HMnew(i,3),9);FCS(HMn
ew(i,4),9);FCS(HMnew(i,5),9);FCS(HMnew(i,6),9);FCS(HMnew(i,7),9)];
    [D,A,Max,Ma,Mb,Mc]=NLstiffness(area,inertia,depth,web_thickness);

[C]=constraint(depth,web_thickness,flange_width,flange_thickness,Max,M
a,Mb,Mc,D,A,inertia);

score(i)=fitness(FCS(HMnew(i,1),2),FCS(HMnew(i,2),2),FCS(HMnew(i,3),2)
,FCS(HMnew(i,4),2),FCS(HMnew(i,5),2),FCS(HMnew(i,6),2),FCS(HMnew(i,7),
2),ro,C);
end

[worstcost worst]=max(score);
HM
score'
% MainHarmony

for t=1:MaxItr;
    %countt=t
        for i=1:NVAR;
            ran1=rand(1);
            if (ran1 < HMCR);
                index = randi(HMS,1);
                NCHV(i) = HM(index,i);
                ran2=rand(1);
                if (ran2 < PAR);
                    NCHV=NCHV;
                    if(ran2 < 0.5);
                        Z=strcmp(IbeamStr,NCHV(i));
                        [r,c]=find(Z);
                            if (r < 168)   ;
                                NCHV(i)=IbeamStr(r+1);
                            elseif (r < 167)
                                NCHV(i)=IbeamStr(r+2);
                            else
                                NCHV(i)=IbeamStr(r);
                            end
                    else
                        Z=strcmp(IbeamStr,NCHV(i));
                        [r,c]=find(Z);
                            if (r > 1)
                                NCHV(i)=IbeamStr(r-1);
                            elseif (r > 2)
                                NCHV(i)=IbeamStr(r-2);
                            else
                                NCHV(i)=IbeamStr(r);
                            end
                    end
```

```
                end
            else
                k=randi(length(IbeamStr));
                NCHV(i)=IbeamStr(k);
            end
        end


    for g=1:NVAR
        Z=strcmp(IbeamStr,NCHV(g));
            [r,c]=find(Z);
            NCHVnew(g)=r;
    end

area1=[FCS(NCHVnew(1),2);FCS(NCHVnew(2),2);FCS(NCHVnew(3),2);FCS(NCHVn
ew(4),2);FCS(NCHVnew(5),2);FCS(NCHVnew(6),2);FCS(NCHVnew(7),2)];

depth1=[FCS(NCHVnew(1),3);FCS(NCHVnew(2),3);FCS(NCHVnew(3),3);FCS(NCHV
new(4),3);FCS(NCHVnew(5),3);FCS(NCHVnew(6),3);FCS(NCHVnew(7),3)];

flange_width1=[FCS(NCHVnew(1),4);FCS(NCHVnew(2),4);FCS(NCHVnew(3),4);F
CS(NCHVnew(4),4);FCS(NCHVnew(5),4);FCS(NCHVnew(6),4);FCS(NCHVnew(7),4)
];

web_thickness1=[FCS(NCHVnew(1),5);FCS(NCHVnew(2),5);FCS(NCHVnew(3),5);
FCS(NCHVnew(4),5);FCS(NCHVnew(5),5);FCS(NCHVnew(6),5);FCS(NCHVnew(7),5
)];

flange_thickness1=[FCS(NCHVnew(1),6);FCS(NCHVnew(2),6);FCS(NCHVnew(3),
6);FCS(NCHVnew(4),6);FCS(NCHVnew(5),6);FCS(NCHVnew(6),6);FCS(NCHVnew(7
),6)];

inertia1=[FCS(NCHVnew(1),9);FCS(NCHVnew(2),9);FCS(NCHVnew(3),9);FCS(NC
HVnew(4),9);FCS(NCHVnew(5),9);FCS(NCHVnew(6),9);FCS(NCHVnew(7),9)];

[D1,A1,Max1,Ma1,Mb1,Mc1]=NLstiffness(area1,inertia1,depth1,web_thickne
ss1);

[C1]=constraint(depth1,web_thickness1,flange_width1,flange_thickness1,
Max1,Ma1,Mb1,Mc1,D1,A1,inertia1);

NEWfit=fitness(FCS(NCHVnew(1),2),FCS(NCHVnew(2),2),FCS(NCHVnew(3),2),F
CS(NCHVnew(4),2),FCS(NCHVnew(5),2),FCS(NCHVnew(6),2),FCS(NCHVnew(7),2)
,ro,C1);
    if NEWfit < worstcost
       HM(worst,:)=NCHV;
       score(worst)=NEWfit;
    end
    [worstcost worst]=max(score);
    [a b]=min(score);
    xmin=HM(b,:);
```

```
    fmin=score(b);
    cc(t)=fmin;
end
[x y]=min(score);
HMmin=HM(y,:)
for g=1:NVAR
    Z=strcmp(IbeamStr,HMmin(g));
    [r,c]=find(Z);
    HMminNUM(g)=r;
end

area2=[FCS(HMminNUM(1),2);FCS(HMminNUM(2),2);FCS(HMminNUM(3),2);FCS(HM
minNUM(4),2);FCS(HMminNUM(5),2);FCS(HMminNUM(6),2);FCS(HMminNUM(7),2)]
;

depth2=[FCS(HMminNUM(1),3);FCS(HMminNUM(2),3);FCS(HMminNUM(3),3);FCS(H
MminNUM(4),3);FCS(HMminNUM(5),3);FCS(HMminNUM(6),3);FCS(HMminNUM(7),3)
];

flange_width2=[FCS(HMminNUM(1),4);FCS(HMminNUM(2),4);FCS(HMminNUM(3),4
);FCS(HMminNUM(4),4);FCS(HMminNUM(5),4);FCS(HMminNUM(6),4);FCS(HMminNU
M(7),4)];

web_thickness2=[FCS(HMminNUM(1),5);FCS(HMminNUM(2),5);FCS(HMminNUM(3),
5);FCS(HMminNUM(4),5);FCS(HMminNUM(5),5);FCS(HMminNUM(6),5);FCS(HMminN
UM(7),5)];

flange_thickness2=[FCS(HMminNUM(1),6);FCS(HMminNUM(2),6);FCS(HMminNUM(
3),6);FCS(HMminNUM(4),6);FCS(HMminNUM(5),6);FCS(HMminNUM(6),6);FCS(HMm
inNUM(7),6)];

inertia2=[FCS(HMminNUM(1),9);FCS(HMminNUM(2),9);FCS(HMminNUM(3),9);FCS
(HMminNUM(4),9);FCS(HMminNUM(5),9);FCS(HMminNUM(6),9);FCS(HMminNUM(7),
9)];

[D2,A2,Max2,Ma2,Mb2,Mc2]=NLstiffness(area2,inertia2,depth2,web_thickne
ss2);

[C2]=constraint(depth2,web_thickness2,flange_width2,flange_thickness2,
Max2,Ma2,Mb2,Mc2,D2,A2,inertia2);
    SCOREmin=score(y)
    D2(160)
    plot(cc);
    toc
end
```

Appendix C - Nonlinear Stiffness MATLAB Code

```
function [Displacementi,Forces,TotalForces,StiffnessMatrix] =
StiffnessNONlin(Displacement,pe,area,inertia,depth,web)

nel=60;                 %number of elmenets
nnel=2;                 %number of nodes per element
ndof=3;                 %number of DOFs per node
edof=nnel*ndof;         %number of DOFs per element
nnode=57;               %total number of nodes in system
sdof=nnode*ndof;


% Member Coords
MC=[ 0      0      0     36;    240      0    240     36;    480      0    480
36;      0     36      0     72;    240     36    240     72;    480     36
480     72;      0     72      0    108;    240     72    240    108;    480
72    480    108;     0    108      0    144;   240    108    240    144;  480
108    480    144;0    144     60    144;  60    144    120    144;120    144
180    144;180    144    240    144;240    144    300    144;300    144    360
144;360    144    420    144;420    144    480    144;0    144      0
180;240    144    240    180;480    144    480    180;0    180      0
216;240    180    240    216;480    180    480    216;0    216      0
252;240    216    240    252;480    216    480    252;0    252      0
288;240    252    240    288;480    252    480    288;0    288     60    288;60
288    120    288;120    288    180    288;180    288    240    288;240    288
300    288;300    288    360    288;360    288    420    288;420    288    480
288;0    288      0    324;240    288    240    324;480    288    480    324;0
324      0    360;240    324    240    360;480    324    480    360;0    360
0    396;240    360    240    396;480    360    480    396;0    396      0
432;240    396    240    432;480    396    480    432;0    432     60    432;60
432    120    432;120    432    180    432;180    432    240    432;240    432
300    432;300    432    360    432;360    432    420    432;    420    432
480    432];
% Member Nodal Connectivity
nodes=[55      1;56      2;57      3;1      4;2      5;3      6;4      7;5
8;6      9;7     10;8     14;9     18;10     11;11     12;12     13;13
14;14     15;15     16;16     17;17     18;10     19;14     20;18     21;19
22;20     23;21     24;22     25;23     26;24     27;25     28;26     32;27
36;28     29;29     30;30     31;31     32;32     33;33     34;34     35;35
36;28     37;32     38;36     39;37     40;38     41;39     42;40     43;41
44;42     45;43     46;44     50;45     54;46     47;47     48;48     49;49
50;50     51;51     52;52     53;53     54];
E=30000;
A=[area(1) area(2) area(1) area(1) area(2) area(1) area(1) area(2)
area(1) area(1) area(2) area(1) area(7) area(7) area(7) area(7)
area(7) area(7) area(7) area(7) area(3) area(4) area(3) area(3)
area(4) area(3) area(3) area(4) area(3) area(3) area(4) area(3)
area(7) area(7) area(7) area(7) area(7) area(7) area(7) area(7)
area(5) area(6) area(5) area(5) area(6) area(5) area(5) area(6)
```

```
area(5) area(5) area(6) area(5) area(7) area(7) area(7) area(7)
area(7) area(7) area(7) area(7)];
I=[inertia(1) inertia(2) inertia(1) inertia(1) inertia(2) inertia(1)
inertia(1) inertia(2) inertia(1) inertia(1) inertia(2) inertia(1)
inertia(7) inertia(7) inertia(7) inertia(7) inertia(7) inertia(7)
inertia(7) inertia(7) inertia(3) inertia(4) inertia(3) inertia(3)
inertia(4) inertia(3) inertia(3) inertia(4) inertia(3) inertia(3)
inertia(4) inertia(3) inertia(7) inertia(7) inertia(7) inertia(7)
inertia(7) inertia(7) inertia(7) inertia(7) inertia(5) inertia(6)
inertia(5) inertia(5) inertia(6) inertia(5) inertia(5) inertia(6)
inertia(5) inertia(5) inertia(6) inertia(5) inertia(7) inertia(7)
inertia(7) inertia(7) inertia(7) inertia(7) inertia(7) inertia(7)];
d=[depth(1) depth(2) depth(1) depth(1) depth(2) depth(1) depth(1)
depth(2) depth(1) depth(1) depth(2) depth(1) depth(7) depth(7)
depth(7) depth(7) depth(7) depth(7) depth(7) depth(7) depth(3)
depth(4) depth(3) depth(3) depth(4) depth(3) depth(3) depth(4)
depth(3) depth(3) depth(4) depth(3) depth(7) depth(7) depth(7)
depth(7) depth(7) depth(7) depth(7) depth(7) depth(5) depth(6)
depth(5) depth(5) depth(6) depth(5) depth(5) depth(6) depth(5)
depth(5) depth(6) depth(5) depth(7) depth(7) depth(7) depth(7)
depth(7) depth(7) depth(7) depth(7)];
M=[pe(3,13) pe(6,16) pe(3,17) pe(6,20) pe(3,33) pe(6,36) pe(3,37)
pe(6,40) pe(3,53) pe(6,56) pe(3,57) pe(6,60)];
N=[pe(4,:)];


C1=1.83*10^-3;
C2=1.04*10^-4;
C3=6.38*10^-6;
tp=0.685;
dg=[depth(7)+6];
tf=1;
Kcon1=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Kcon2=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Kcon3=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Kcon4=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Kcon5=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Kcon6=(dg^-2.4)*(tp^-0.4)*(tf^-1.5);
Theta1=C1*(Kcon1*M(1))^1+C2*(Kcon1*M(1))^3+C3*(Kcon1*M(1))^5;
Theta2=C1*(Kcon2*M(2))^1+C2*(Kcon2*M(2))^3+C3*(Kcon2*M(2))^5;
Theta3=C1*(Kcon2*M(3))^1+C2*(Kcon2*M(3))^3+C3*(Kcon2*M(3))^5;
Theta4=C1*(Kcon1*M(4))^1+C2*(Kcon1*M(4))^3+C3*(Kcon1*M(4))^5;
Theta5=C1*(Kcon3*M(5))^1+C2*(Kcon3*M(5))^3+C3*(Kcon3*M(5))^5;
Theta6=C1*(Kcon4*M(6))^1+C2*(Kcon4*M(6))^3+C3*(Kcon4*M(6))^5;
Theta7=C1*(Kcon4*M(7))^1+C2*(Kcon4*M(7))^3+C3*(Kcon4*M(7))^5;
Theta8=C1*(Kcon3*M(8))^1+C2*(Kcon3*M(8))^3+C3*(Kcon3*M(8))^5;
Theta9=C1*(Kcon5*M(9))^1+C2*(Kcon5*M(9))^3+C3*(Kcon5*M(9))^5;
Theta10=C1*(Kcon6*M(10))^1+C2*(Kcon6*M(10))^3+C3*(Kcon6*M(10))^5;
Theta11=C1*(Kcon6*M(11))^1+C2*(Kcon6*M(11))^3+C3*(Kcon6*M(11))^5;
Theta12=C1*(Kcon5*M(12))^1+C2*(Kcon5*M(12))^3+C3*(Kcon5*M(12))^5;
R1=M(1)/Theta1;
```

```
R2=M(2)/Theta2;
R3=M(3)/Theta3;
R4=M(4)/Theta4;
R5=M(5)/Theta5;
R6=M(6)/Theta6;
R7=M(7)/Theta7;
R8=M(8)/Theta8;
R9=M(9)/Theta9;
R10=M(10)/Theta10;
R11=M(11)/Theta11;
R12=M(12)/Theta12;
r=ones(60,2);
r(13,1)=1/(1+((3*E*I(7))/(R1*60)));
r(16,2)=1/(1+((3*E*I(7))/(R2*60)));
r(17,1)=1/(1+((3*E*I(7))/(R3*60)));
r(20,2)=1/(1+((3*E*I(7))/(R4*60)));
r(33,1)=1/(1+((3*E*I(7))/(R5*60)));
r(36,2)=1/(1+((3*E*I(7))/(R6*60)));
r(37,1)=1/(1+((3*E*I(7))/(R7*60)));
r(40,2)=1/(1+((3*E*I(7))/(R8*60)));
r(53,1)=1/(1+((3*E*I(7))/(R9*60)));
r(56,2)=1/(1+((3*E*I(7))/(R10*60)));
r(57,1)=1/(1+((3*E*I(7))/(R11*60)));
r(60,2)=1/(1+((3*E*I(7))/(R12*60)));


L=zeros(60,1);
% Member 1
i=1;
r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
```

```
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 2
    i=2;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 3
    i=3;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 4
```

```
    i=4;r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 5
    i=5;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 6
    i=6;
    r1(i)=r(i,1);
r2(i)=r(i,2);
```

```
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 7
    i=7;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 8
    i=8;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
```

```
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 9
    i=9;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 10
    i=10;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
```

```
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 11
    i=11;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 12
    i=12;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
```

```
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 13
    i=13;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 14
    i=14;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
```

```
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 15
    i=15;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 16
    i=16;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
```

```
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 17
    i=17;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 18
    i=18;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
```

```
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 19
    i=19;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 20
    i=20;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
```

```
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 21
    i=21;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 22
    i=22;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
```

```
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 23
    i=23;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 24
    i=24;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
```

```
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 25
    i=25;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 26
    i=26;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
```

```
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 27
    i=27;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 28
    i=28;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 29
```

```
    i=29;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 30
    i=30;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 31
    i=31;
    r1(i)=r(i,1);
```

```
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 32
    i=32;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 33
    i=33;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
```

```
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 34
    i=34;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 35
    i=35;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
```

```
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 36
    i=36;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 37
    i=37;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
```

```
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 38
    i=38;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 39
    i=39;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
```

```
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 40
    i=40;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 41
    i=41;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
```

```
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 42
    i=42;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 43
    i=43;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
```

```
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 44
    i=44;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 45
    i=45;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
```

```
        for j = 1:3*nnode;
            for g = 1:6;
                if index(g) == j;
                    LocM(g,j,i) = 1;
                end
            end
        end
    K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
    % Member 46
        i=46;
        r1(i)=r(i,1);
    r2(i)=r(i,2);
    N(i)=pe(4,i);
    L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
    te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
    si(:,:,i) = Si(E,A(i),L(i),I(i));
    cei(:,:,i)= Cei(r1(i),r2(i),L(i));
    gi(:,:,i) = Gi(N(i),L(i));
    cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
    ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
    k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
    nd(1)=nodes(i,1);
    nd(2)=nodes(i,2);
    index=feeldof(nd,nnel,ndof).';
    LocM(:,:,i) = zeros(6,(3*nnode));
        for j = 1:3*nnode;
            for g = 1:6;
                if index(g) == j;
                    LocM(g,j,i) = 1;
                end
            end
        end
    K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
    % Member 47
        i=47;
        r1(i)=r(i,1);
    r2(i)=r(i,2);
    N(i)=pe(4,i);
    L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
    te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
    si(:,:,i) = Si(E,A(i),L(i),I(i));
    cei(:,:,i)= Cei(r1(i),r2(i),L(i));
    gi(:,:,i) = Gi(N(i),L(i));
    cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
    ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
    k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
    nd(1)=nodes(i,1);
    nd(2)=nodes(i,2);
    index=feeldof(nd,nnel,ndof).';
    LocM(:,:,i) = zeros(6,(3*nnode));
        for j = 1:3*nnode;
            for g = 1:6;
```

```
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 48
    i=48;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 49
    i=49;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
```

```
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 50
    i=50;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 51
    i=51;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
```

```
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 52
    i=52;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 53
    i=53;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
```

```
% Member 54
    i=54;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 55
    i=55;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 56
    i=56;
```

```
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 57
    i=57;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 58
    i=58;
    r1(i)=r(i,1);
r2(i)=r(i,2);
```

```
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 59
    i=59;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);
% Member 60
    i=60;
    r1(i)=r(i,1);
r2(i)=r(i,2);
N(i)=pe(4,i);
L(i) = sqrt((MC(i,3)-MC(i,1))^2 +(MC(i,4)-MC(i,2))^2);
```

```
te(:,:,i) = Te(MC(i,1),MC(i,2),MC(i,3),MC(i,4));
si(:,:,i) = Si(E,A(i),L(i),I(i));
cei(:,:,i)= Cei(r1(i),r2(i),L(i));
gi(:,:,i) = Gi(N(i),L(i));
cgi(:,:,i)= Cgi(r1(i),r2(i),L(i));
ki(:,:,i)=si(:,:,i)*cei(:,:,i)+gi(:,:,i)*cgi(:,:,i);
k(:,:,i)=te(:,:,i)'*ki(:,:,i)*te(:,:,i);
nd(1)=nodes(i,1);
nd(2)=nodes(i,2);
index=feeldof(nd,nnel,ndof).';
LocM(:,:,i) = zeros(6,(3*nnode));
    for j = 1:3*nnode;
        for g = 1:6;
            if index(g) == j;
                LocM(g,j,i) = 1;
            end
        end
    end
K(:,:,i)=LocM(:,:,i)'*k(:,:,i)*LocM(:,:,i);


StiffnessMatrix=K(:,:,1)+K(:,:,2)+K(:,:,3)+K(:,:,4)+K(:,:,5)+K(:,:,6)+
K(:,:,7)+K(:,:,8)+K(:,:,9)+K(:,:,10)+K(:,:,11)+K(:,:,12)+K(:,:,13)+K(:
,:,14)+K(:,:,15)+K(:,:,16)+K(:,:,17)+K(:,:,18)+K(:,:,19)+K(:,:,20)+K(:
,:,21)+K(:,:,22)+K(:,:,23)+K(:,:,24)+K(:,:,25)+K(:,:,26)+K(:,:,27)+K(:
,:,28)+K(:,:,29)+K(:,:,30)+K(:,:,31)+K(:,:,32)+K(:,:,33)+K(:,:,34)+K(:
,:,35)+K(:,:,36)+K(:,:,37)+K(:,:,38)+K(:,:,39)+K(:,:,40)+K(:,:,41)+K(:
,:,42)+K(:,:,43)+K(:,:,44)+K(:,:,45)+K(:,:,46)+K(:,:,47)+K(:,:,48)+K(:
,:,49)+K(:,:,50)+K(:,:,51)+K(:,:,52)+K(:,:,53)+K(:,:,54)+K(:,:,55)+K(:
,:,56)+K(:,:,57)+K(:,:,58)+K(:,:,59)+K(:,:,60);
Kcc=StiffnessMatrix(163:171,163:171);
Kcu=StiffnessMatrix(163:171,1:162);
Kuc=StiffnessMatrix(1:162,163:171);
Kuu=StiffnessMatrix(1:162,1:162);

ff=(zeros(sdof,1));
ff(28)=(1/10)*8;
ff(82)=(1/10)*8;
ff(136)=(1/10)*4;

Pfef=zeros(6,60);
M11=(((1/10)*0.22*60^2)/12)*(3*r(13,1)*(2-r(13,2))/(4-
r(13,1)*r(13,2)));
M12=(((1/10)*0.22*60^2)/12)*(3*r(13,2)*(2-r(13,1))/(4-
r(13,1)*r(13,2)));
M21=(((1/10)*0.22*60^2)/12)*(3*r(16,1)*(2-r(16,2))/(4-
r(16,1)*r(16,2)));
M22=(((1/10)*0.22*60^2)/12)*(3*r(16,2)*(2-r(16,1))/(4-
r(16,1)*r(16,2)));
M31=(((1/10)*0.22*60^2)/12)*(3*r(17,1)*(2-r(17,2))/(4-
r(17,1)*r(17,2)));
```

```
M32=(((1/10)*0.22*60^2)/12)*(3*r(17,2)*(2-r(17,1))/(4-
r(17,1)*r(17,2)));
M41=(((1/10)*0.22*60^2)/12)*(3*r(20,1)*(2-r(20,2))/(4-
r(20,1)*r(20,2)));
M42=(((1/10)*0.22*60^2)/12)*(3*r(20,2)*(2-r(20,1))/(4-
r(20,1)*r(20,2)));
M51=(((1/10)*0.22*60^2)/12)*(3*r(33,1)*(2-r(33,2))/(4-
r(33,1)*r(33,2)));
M52=(((1/10)*0.22*60^2)/12)*(3*r(33,2)*(2-r(33,1))/(4-
r(33,1)*r(33,2)));
M61=(((1/10)*0.22*60^2)/12)*(3*r(36,1)*(2-r(36,2))/(4-
r(36,1)*r(36,2)));
M62=(((1/10)*0.22*60^2)/12)*(3*r(36,2)*(2-r(36,1))/(4-
r(36,1)*r(36,2)));
M71=(((1/10)*0.22*60^2)/12)*(3*r(37,1)*(2-r(37,2))/(4-
r(37,1)*r(37,2)));
M72=(((1/10)*0.22*60^2)/12)*(3*r(37,2)*(2-r(37,1))/(4-
r(37,1)*r(37,2)));
M81=(((1/10)*0.22*60^2)/12)*(3*r(40,1)*(2-r(40,2))/(4-
r(40,1)*r(40,2)));
M82=(((1/10)*0.22*60^2)/12)*(3*r(40,2)*(2-r(40,1))/(4-
r(40,1)*r(40,2)));
M91=(((1/10)*0.17*60^2)/12)*(3*r(53,1)*(2-r(53,2))/(4-
r(53,1)*r(53,2)));
M92=(((1/10)*0.17*60^2)/12)*(3*r(53,2)*(2-r(53,1))/(4-
r(53,1)*r(53,2)));
M101=(((1/10)*0.17*60^2)/12)*(3*r(56,1)*(2-r(56,2))/(4-
r(56,1)*r(56,2)));
M102=(((1/10)*0.17*60^2)/12)*(3*r(56,2)*(2-r(56,1))/(4-
r(56,1)*r(56,2)));
M111=(((1/10)*0.17*60^2)/12)*(3*r(57,1)*(2-r(57,2))/(4-
r(57,1)*r(57,2)));
M112=(((1/10)*0.17*60^2)/12)*(3*r(57,2)*(2-r(57,1))/(4-
r(57,1)*r(57,2)));
M121=(((1/10)*0.17*60^2)/12)*(3*r(60,1)*(2-r(60,2))/(4-
r(60,1)*r(60,2)));
M122=(((1/10)*0.17*60^2)/12)*(3*r(60,2)*(2-r(60,1))/(4-
r(60,1)*r(60,2)));

V11=(((1/10)*0.22*60)/2)+(M11+M12)/60;
V12=(((1/10)*0.22*60)/2)+-(M11+M12)/60;
V21=(((1/10)*0.22*60)/2)+(M21+M22)/60;
V22=(((1/10)*0.22*60)/2)+-(M21+M22)/60;
V31=(((1/10)*0.22*60)/2)+(M31+M32)/60;
V32=(((1/10)*0.22*60)/2)+-(M31+M32)/60;
V41=(((1/10)*0.22*60)/2)+(M41+M42)/60;
V42=(((1/10)*0.22*60)/2)+-(M41+M42)/60;
V51=(((1/10)*0.22*60)/2)+(M51+M52)/60;
V52=(((1/10)*0.22*60)/2)+-(M51+M52)/60;
V61=(((1/10)*0.22*60)/2)+(M61+M62)/60;
V62=(((1/10)*0.22*60)/2)+-(M61+M62)/60;
V71=(((1/10)*0.22*60)/2)+(M71+M72)/60;
```

```
V72=(((1/10)*0.22*60)/2)+-(M71+M72)/60;
V81=(((1/10)*0.22*60)/2)+(M81+M82)/60;
V82=(((1/10)*0.22*60)/2)+-(M81+M82)/60;
V91=(((1/10)*0.17*60)/2)+(M91+M92)/60;
V92=(((1/10)*0.17*60)/2)+-(M91+M92)/60;
V101=(((1/10)*0.17*60)/2)+(M101+M102)/60;
V102=(((1/10)*0.17*60)/2)+-(M101+M102)/60;
V111=(((1/10)*0.17*60)/2)+(M111+M112)/60;
V112=(((1/10)*0.17*60)/2)+-(M111+M112)/60;
V121=(((1/10)*0.17*60)/2)+(M121+M122)/60;
V122=(((1/10)*0.17*60)/2)+-(M121+M122)/60;

Mw22=(((1/10)*0.22*60^2)/12);
Mw17=(((1/10)*0.17*60^2)/12);
Vw22=(((1/10)*0.22*60)/2);
Vw17=(((1/10)*0.17*60)/2);

w_22=[0;Vw22;Mw22;0;Vw22;-Mw22];
w_17=[0;Vw17;Mw17;0;Vw17;-Mw17];

NF1=[0;V11;M11;0;V12;-M12];
NF2=[0;V21;M21;0;V22;-M22];
NF3=[0;V31;M31;0;V32;-M32];
NF4=[0;V41;M41;0;V42;-M42];
NF5=[0;V51;M51;0;V52;-M52];
NF6=[0;V61;M61;0;V62;-M62];
NF7=[0;V71;M71;0;V72;-M72];
NF8=[0;V81;M81;0;V82;-M82];
NF9=[0;V91;M91;0;V92;-M92];
NF10=[0;V101;M101;0;V102;-M102];
NF11=[0;V111;M111;0;V112;-M112];
NF12=[0;V121;M121;0;V122;-M122];

Pfef(:,13)=NF1;
Pfef(:,14)=w_22;
Pfef(:,15)=w_22;
Pfef(:,16)=NF2;
Pfef(:,17)=NF3;
Pfef(:,18)=w_22;
Pfef(:,19)=w_22;
Pfef(:,20)=NF4;
Pfef(:,33)=NF5;
Pfef(:,34)=w_22;
Pfef(:,35)=w_22;
Pfef(:,36)=NF6;
Pfef(:,37)=NF7;
Pfef(:,38)=w_22;
Pfef(:,39)=w_22;
Pfef(:,40)=NF8;
Pfef(:,53)=NF9;
Pfef(:,54)=w_17;
```

```
Pfef(:,55)=w_17;
Pfef(:,56)=NF10;
Pfef(:,57)=NF11;
Pfef(:,58)=w_17;
Pfef(:,59)=w_17;
Pfef(:,60)=NF12;

for i = 1:60
    pf=transpose(te(:,:,i))*Pfef(:,i);
    QQ=transpose(LocM(:,:,i))*pf;
    pfi(:,i)=QQ;
end


Pf=sum(pfi,2);
Pu=ff-Pf;
Uc=ff(163:171);
Uu=Kuu^-1*(Pu(1:162)-Kuc*Uc);
D1=(zeros(171,1));
D1(163:171,1)=Uc;
D1(1:162,1)=Uu;
Displace=D1;
Reactions=Kcu*Uu+Kcc*Uc;
R=Reactions;

for j = 1:60
    u=LocM(:,:,j)*D1;
    ue=te(:,:,j)*u;
    p=(ki(:,:,j)*ue)+Pfef(:,j);
    p(6)=p(6)*-1;
    pei(:,j)=p;
end

Displacementi=Displace+Displacement;
Forces=pei;
TotalForces=pei+pe;

end
```