Global Optimization of Design Parameters of Network Arch Bridges

by

Nazrul Islam

MASTER OF SCIENCE IN CIVIL ENGINEERING (STRUCTURAL)

Department of Civil Engineering

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

September, 2010

Global Optimization of Design Parameters of Network Arch Bridges

by

Nazrul Islam

A thesis submitted to the Department of Civil Engineering of Bangladesh University of Engineering and Technology, Dhaka, in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE IN CIVIL ENGINEERING (STRUCTURAL)

Department of Civil Engineering

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

September, 2010

The thesis titled **"Global Optimization of Design Parameters of Network Arch Bridges"** submitted by **Nazrul Islam, Roll No.: 100704335, Session: October 2007** has been accepted as satisfactory in partial fulfilment of the requirement for the degree of M.Sc. Engineering (Civil and Structural) on 5th September, 2010.

BOARD OF EXAMINERS

Dr. Raquib Ahsan Associate Professor Department of Civil Engineering BUET, Dhaka.	**Chairman** (Supervisor)
Dr. Md. Zoynul Abedin Professor and Head Department of Civil Engineering BUET, Dhaka.	**Member** (Ex-officio)
Dr. Abdur Rouf Professor Department of Civil Engineering BUET, Dhaka.	**Member**
Mr. Sabbir Siddiquee Director, Building and Structural Engineering DPM Consultant Ltd Dhaka	**Member** (External)

DEDICATION

To Those

Who Are Praised

By

Allah

DECLARATION

It is hereby declared that except for the contents where specific reference have been made to the work of others, the studies contained in this thesis is the result of investigation carried out by the author. No part of this thesis has been submitted to any other university or other educational establishment for a Degree, Diploma or other qualification (except for publication).

____

Nazrul Islam

ACKNOWLEDGEMENT

IN THE NAME OF ALLAH, MOST GRACIOUS, MOST MERCIFUL

All praises to the Sustainer of the worlds, and grace, honour and salutations on the Chief of Apostles and Seal of Prophets, Muhammad Sallallahu Alaihissalam, his family, companions and those who followed him in an excellent manner and invited mankind towards Allah, till the Day of Resurrection.

I would like to thank Allah Subha'nahu Wa Ta'ala the Exalted for His blessing that allowed me to complete this thesis and I pray that He accepts it as a work of sincerity and benefit.

I would like to express my deep and sincere gratitude to my thesis supervisor, Dr. Raquib Ahsan, Associate Professor of Civil Engineering Department of BUET for his dynamic supervision, continuous guidance, invaluable suggestion, and enthusiastic encouragement throughout various stages of this research. His active interest in this topic and valuable advice was the source of author's inspiration. I am also grateful to Professor Dr. Abdur Rouf of Civil Engineering Department of BUET for his help and valuable suggestions regarding this work.

During this work I have collaborated with many colleagues for whom I have great regard, and I wish to extend my warmest thanks to all those who have helped me with my work in the Department of Civil Engineering of BUET.

Finally, I extend my acknowledgement and heartfelt love to my family members who are always a constant source of inspiration throughout my life.

ABSTRACT

Structural optimization of network arch bridges is performed in this study. Optimization has been achieved through execution of a simulator, evaluation of a constrained objective function and adjustment of system parameters in an iterative and directed way. Objective is to minimize the total material cost required for hanger, arch concrete and arch reinforcement of network arch bridges. The optimization problem is characterized by having mixed design variables. Structural analysis of network arch bridges is performed by a finite element simulator, ANSYS. Optimization of structural design of the bridge is performed through a global optimization algorithm EVOP. An interfacing environment has been developed by integrating ANSYS with EVOP. The interfacing has been verified through some benchmark problems of optimization. Finally parameters of a tied arch bridge are invoked in the optimization process as the initial design and after completion of optimization process, optimal design variables i.e. hanger arrangement, no of hangers, cross section required for hanger, arch section and rise to span ratio of arch, are obtained within a range of design constant parameters. Response parameters of the arch bridge with optimum design variables are analyzed regarding hanger stress, bending moment, axial force and influence line for bending moment in arch and amount of reinforcement required in the arch. Results are compared with the initially designed arch. Results show that arch bridges with optimal design variables using global optimization technique shows significant improvement over the arch bridges designed initially. Optimal design variables confirm significant reduction of bending moment in arch than the arch bridges with vertical hangers. Based on optimum design criteria it is found that circular arch geometry requires shallower arch than that required for parabolic arch. In addition, within the range of design constant parameters it is observed that 36% to 40% of total cost can be saved for circular and parabolic arches if design is optimized. Results also show that parabolic arch with optimum design variables is more economic than the optimal arch bridges with circular arch geometry. The interfacing environment developed in the study opens the door for simulation driven most economical design of any type of structure.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AASHTO	American Association of State Highway and Transportation Officials
AISC	American Institute of Steel Construction
APDL	ANSYS Parametric Design Language
ASTM	American Society for Testing and Materials
EVOP	Evolutionary Operation
FEA	Finite Element Analysis
GA	Genetic Algorithm
LRFD	Load and Resistance Factor Design
RHD	Roads and Highway Department
SQP	Sequential Quadratic Programming
UPF	User Programmable Features

LIST OF NOTATION

A_e	Effective Net Area
A_g	Gross Area of Member
A_h	Cross Sectional Area of Cable of Hanger
a, b, c, d	Complex Vertices
B_h	Arch Width
B_w	Width of Bridge
b	Width of Cantilever Beam Section
C	Location of Centroid of 'Complex'
C_{AC}	Cost of Concrete
C_{AS}	Cost of Reinforcement
C_{HC}	Cost of Cable of Hangers
C_T	Total Cost
CRT_i	Strength Criteria of Arch
CRT^L	Lower Limit of Strength Criteria of Arch
CRT^U	Lower Limit of Strength Criteria of Arch
d_1	Distance of the "Centre" of the Interaction Diagram to the Point which Represents the Acting Forces and Moments
d_2	Distance of the "Centre" of the Interaction Diagram to the Point which Represents the Homothetic Ultimate Forces and Moments
d_c	Reinforcement Cover for Arch Section
DL	Dead Load
E_C	Modulus of Elasticity for Concrete
E_b	Modulus of Elasticity of Cantilever Beam
$f_{ca}^{'}$	Concrete Compressive Strength at 28 days for Arch and Bracing
$f_{cd}^{'}$	Concrete Compressive Strength at 28 days for Deck
f_u	Ultimate Strength of Cable of Hanger
f_y	Arch Reinforcement Yield Stress
f_i	Design Tensile Strength
f^L	Lower Limit of Design Tensile Strength
f^U	Upper Limit of Design Tensile Strength
F_y	Specified Minimum Yield Stress of the Type of Steel Being Used

F_u	Specified Minimum Tensile Strength of the Type of Steel Being Used
$f(x)$	Objective Function
f(a), f(b), f(c), f(d)	Function Values of Complex Vertices
F. S	Factor of Safety
$G(x)_i$	Implicit Values
$G(x)_i^L$	Lower Limit of Implicit Constraints
$G(x)_i^U$	Upper Limit of Implicit Constraints
h	Rise of Arch
HS	Standard Single Vehicle Load
HL	Standard Lane Load
H_h	Arch Depth
H_2	Horizontal Thrust of Arch
i	Hanger node number along the Arch
I	Impact Factor
l	Length of Cantilever Beam
k	Number of Vertices in Complex of EVOP
L	Span of Arch
L_a	Half Length of Arch Axis
LL	Live Load
LRFD	Load and Resistance Factor Design
M	Bending Moment
ng	Vertex of a Complex having Highest Function Value
nh	Vertex of a Complex having Second Highest Function Value
Ns	Vertex of a Complex having Lowest Function Value
N_h	Total Number of Hangers
N_{h1}	Number of Hangers for Set1 Cable in Each Arch
$NDIV$	Number of Design Variables
NIC	Number of Implicit Constraints
p	Uniform Pressure
POS_i	End Hanger Inclination with Vertical
POS^L	Lower Limit of End Hanger Inclination with Vertical

POS^U	Upper Limit of End Hanger Inclination with Vertical
P_n	Nominal Axial Strength
R1, R2, R3,R4	Radius of Various Circles
r	Half the Distance Between Two Outer Ribs
R_h	Rise to Span Ratio
RNR_i	Reinforcement Factor of the Arch
RNR^L	Lower Limit of Reinforcement Factor of the Arch
RNR^U	Upper Limit of Reinforcement Factor of the Arch
t	Thickness of Beam at the Point of Load Application
TK	Design Variables for Thickness of cantilever Beam
T_1	Trial Point of a Complex
T_2	New Feasible Point of Complex
UP_{HC}	Unit Price of Cable
UP_{AC}	Unit Price of Concrete of Arch
UP_{AS}	Unit Price of Steel of Arch
V_{AC}	Volume of Concrete
W	Uniformly Distributed Load on Arch
W_{AS}	Weight of Reinforcement
W_{HC}	Weight of Cables
X	The Centroid of Complex which Lies in the Infeasible Area
x_k	Design Variables
x_k^L	Lower Limit of Design Variables
x_k^U	Upper Limit of Design Variables
x_g	Location of Vertex n_g
x_r	Location of Trial Point
α	Reflection Coefficient
β	Contraction Coefficient
γ	Expansion Coefficient
σ_{max}	Extreme Hanger Stress
$\sigma_{v\ max}$	Von Misses Stress
δy_{max}	Maximum Deflection of Cantilever
φ_t	Resistance Factor for Tension
$\varphi 1$	Start Angle for Set 1 Cable

$\varphi 2$	Start Angle for Set 2 Cable
$\Delta\varphi_1$	Angle Change for Set 1 Cable
$\Delta\varphi_2$	Angle Change for Set 2 Cable
$\varphi 1_n$	End Angle for Hanger Inclination Set1
$\varphi 2_n$	End Angle for Hanger Inclination Set2
Φ_{cpx}	Convergence Parameter of EVOP
υ	Poisson's Ratio of Concrete
υ_s	Poisson's Ratio of Cantilever Beam
θ	Cross Angle

CHAPTER 1

INTRODUCTION

1.1 Background

In general, optimization is concerned with achieving the best outcome of a given objective while satisfying certain restrictions. Naturally human beings, guided and influenced by their natural surroundings almost instinctively perform optimization of all functions. Human beings attempt to economize energy or minimize discomfort and pain. The impetus is to exploit the available limited resources in a manner that maximizes output or profit.

One of the most fundamental principles of all living being in our world is the search for an optimal state. It begins in the microcosm where atoms in matter try to form bonds in order to minimize the energy of their electrons [1]. As long as humankind exists, they strive for perfection in many areas. Human beings want to reach a maximum degree of happiness with the least amount of effort. In economy, profit must be maximized and costs should be as low as possible. Therefore, optimization is one of the oldest of sciences which even extends into daily life [2].

Formulation of optimization problems is quite mathematical and it can be classified into two groups – Analytical Method and Numerical Method. Out of numerical optimization methods global optimization algorithms are a branch of applied mathematics and numerical analysis that employ measures that prevent convergence to local optima and increase the probability of finding a global optimum i.e. the best feasible design according to preselected quantitative measure of effectiveness [3]. Application of this algorithm in the field of structural optimization problems is studied in this thesis.

Structural optimization as a discipline has been developing rapidly over the years. The aim of structural optimization is to determine the values of structural design variables in order to minimize or maximize an objective function of a structure while satisfying given constraints. Objective function and constraints involve with important structural performance metrics such as cost, stress, mass, deformation or natural frequencies [4].

Evaluation of structural performance requires structural analysis after establishing an analytical model or finite element model which represents the structural behavior under application of external loadings. Application of structural optimization in minimum weight design of structures was first recognized by the aerospace industry where aircraft structure design is controlled more by weight than by cost considerations [5]. In other industries dealing with civil, mechanical, and automotive engineering systems, cost is the primary consideration although the weight of the system does affect its cost.

In this study, optimization of network arch bridges has been paid particular attention as a structural optimization problem. Various design variables are involved in the optimization process of structural elements of these bridges to minimize cost of superstructure. A few works on this subject have been performed in recent years. In 1980, Tveit [6] developed some concepts regarding hanger arrangement to reduce bending moments in arch and also in tie. In 2003, Brunn et al. [7] adopted a hypothesis for optimization of hanger arrangement that arch should be a part of a circle and hanger should be arranged in such a way that hanger intersections lie on the radii of the arch circle. Brunn et al. adopted trial and error based traditional method for optimization of various design parameters. In 2008, Tveit [8] suggested that optimal network arches can be achieved if some hangers cross other hangers at least twice.

Previous works on optimization of tied arch bridges with inclined hangers has been adopted based on some hypothesis and followed traditional method of design. However the optimization problem is characterized by global optimization problem. In general, the traditional optimization approaches have difficulties in dealing with global optimization problems. One of the main limitations of traditional optimization approaches is that they can easily be entrapped in local minima i.e. best possible solution may not be achieved. On the other hand global optimization methods seek the best possible solution to a set of criteria. A probabilistic global optimization method EVOP developed by Ghani [9] has high probability for seeking best possible solution. It does not require training and initial known population size as required by genetic algorithm. Moreover it is well suited for mixed design variables. Therefore a global optimization approach is required to identify the optimum design parameters of network arch bridges utilizing EVOP. Again simulation driven optimization using global optimization method is needed to analyze structural optimization of such bridges for obtaining most economical design.

1.2 Objectives of the Present Study

The objective of this study is to find a most economical structural design of superstructure particularly arch and hangers of network arch bridges using a global optimization algorithm EVOP. Specific aim is to minimize cost by

- Optimizing arch geometric shape, its rise to span ratio and best arch section.
- Optimizing the number, arrangement and cross sectional dimensions of hangers.

1.3 Scope and Methodology of the Study

Optimization of arch and hangers of network arch bridges is performed through a global optimization algorithm EVOP. EVOP is interfaced with a finite element software ANSYS for evaluation of structural response of the bridge. The optimization problem is formulated as a mixed integer-discrete nonlinear programming problem where objective is to minimize the total cost of superstructure of bridge by considering the cost of hangers, arch and reinforcement of arch. Cost of deck is not considered in the present study. Global optimization algorithm EVOP is interfaced with finite element software so that any structure which requires finite element analysis can be optimized programmatically for most economical design. Optimal design variables of network arch bridges are determined using EVOP and the results are then compared with initial design. Effects of arch geometry on optimal design of arch bridges are also determined.

1.4 Organization of the Thesis

The thesis is organized into eight chapters. **Chapter 1** is the current chapter, which introduces the work presented in this thesis. **Chapter 2** reviews the available literature discussing various studies conducted on network arch bridges and simulation driven optimization. This chapter also includes a literature survey on global optimization methods. **Chapter 3** describes the optimal design problem statement of the bridge. **Chapter 4** describes interfacing of a global optimization algorithm, EVOP with finite element software, ANSYS. **Chapter 5** presents the development of finite element model, analysis and design of network arch bridges. **Chapter 6** presents results obtained from optimization of FE model of network arch bridges and shows comparison of optimal design with traditional method of design for different geometry of arch. Finally, **chapter 7** draws conclusion of the current work and discusses recommendations for future work in the area of optimal design of arch bridges.

CHAPTER 2

LITERATURE REVIEW

2.1 General

This chapter reviews the available literature discussing various studies conducted on network arch bridges. This chapter also includes a literature survey on global optimization methods. Past works on simulation driven optimization has been summarized at the end of this chapter.

2.2 Arch Bridges

An arch is a girder, usually curved in form, which develops reactions with inwardly directed horizontal components under the action of vertical loads only [10]. The arch bridge comprises arch rib of curved structural member spanning an opening and serving as a support for the loads above the opening. Nomenclature used to describe arch bridges is outlined on Figure 2.1. Arch bridges of true or perfect arch, theoretically, is one in which only a compressive force acts at the centroid of each element of the arch which is not practically possible for more than one loading condition. In real case, arch bridges are usually subject to multiple loadings (dead load, live load, temperature, etc.) which produce bending stresses in the arch rib that are generally small compared with the axial compressive stress.

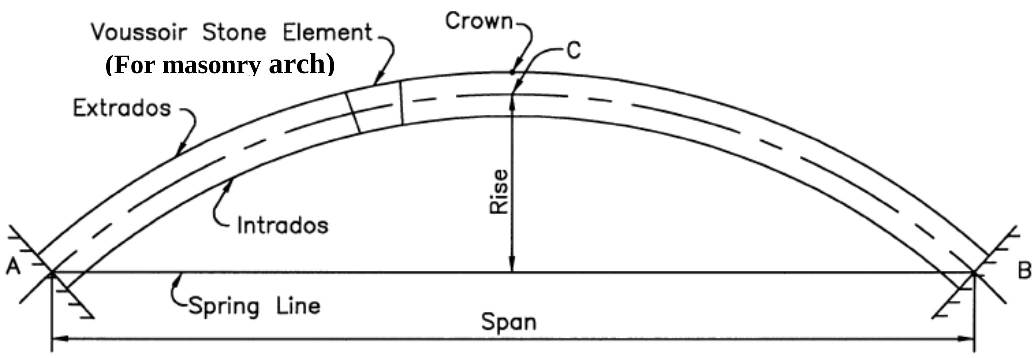


Figure 2.1: Arch Nomenclature [11]

There are many different types and arrangements of arch bridges. A deck arch is one where the bridge deck which includes the structure that directly supports the traffic loads is located above the crown of the arch. The deck arch is also known as a true or perfect arch. Some deck arches are shown in Figures 2.2 (a), (c) - (h), (p) and (q). A through-arch is one where the bridge deck is located at the springline of the arch shown in Figure 2.2 (i). A half-through arch is that where the bridge deck is located at an elevation between a deck arch and a through arch. Figures 2.2 (b) and (k) are the type of half through arch. A further classification refers to the articulation of the arch. A fixed arch is depicted in Figure 2.1 and implies no rotation possible at the supports, A and B. A fixed arch is indeterminate to the third degree. A three-hinged arch that allows rotation at A, B, and C is statically determinate. A two-hinged arch allows rotation at A and B and is indeterminate to one degree.

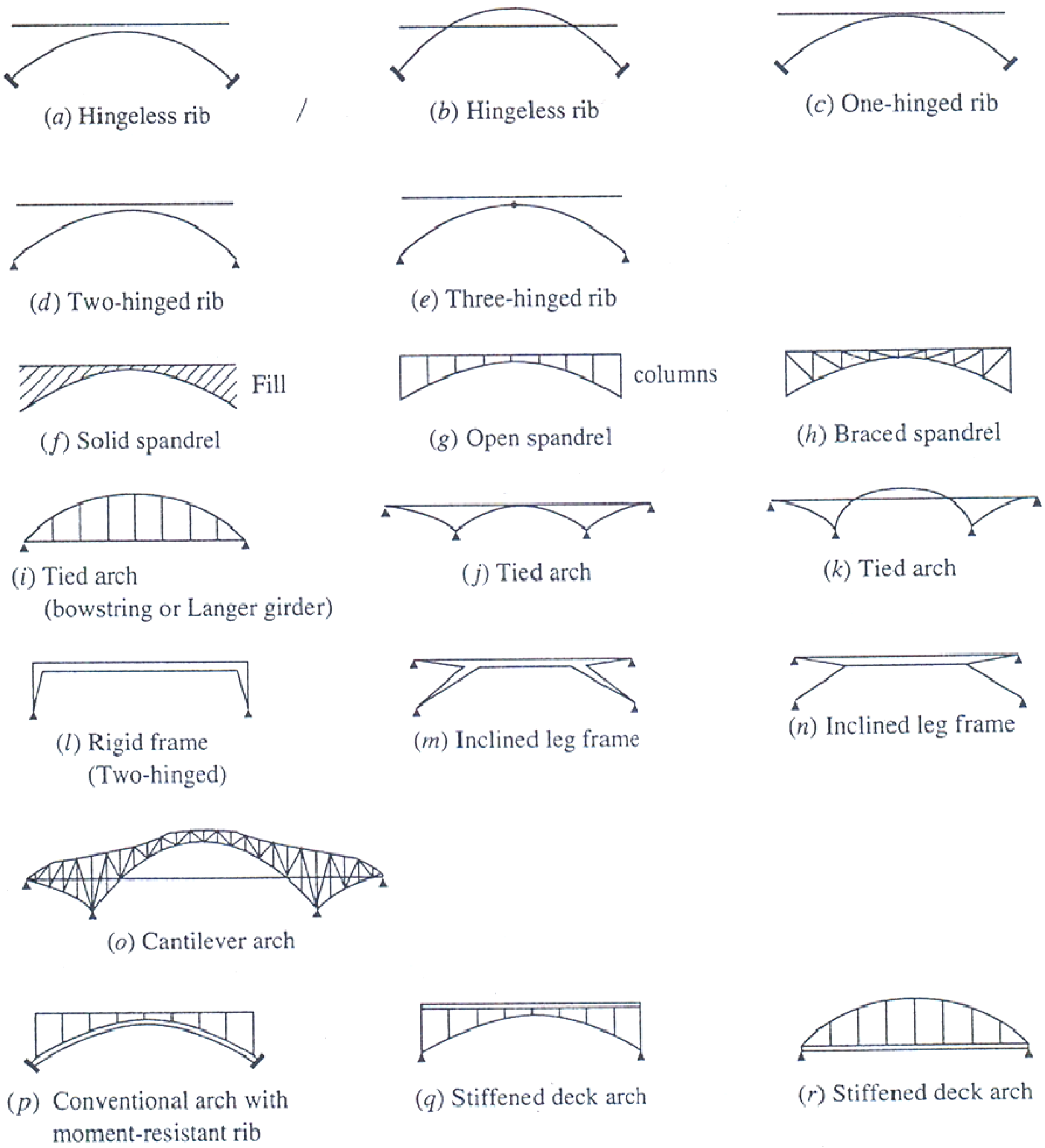


Figure 2.2: Various Types of Arch Bridges [12]

A tied arch is one where the reactive horizontal forces acting on the arch ribs are supplied by a tension tie at deck level of a through or half-through arch. The tension tie is usually a steel plate girder or a steel box girder and, depending on its stiffness, is capable of carrying a portion of the live loads. A weak tie girder, however, requires a deep arch rib and a thin arch rib requires a stiff deep tie girder. Since they are dependent on each other, it is possible to optimize the size of each according to the goal established for aesthetics and/ or cost.

While most through or half-through arch bridges are constructed with two planes of vertical arch ribs there have been a few constructed with only one rib with the roadways cantilevered on each side of the rib.

Hangers usually consist of wire ropes or rolled sections. The hangers are usually vertical but truss like diagonal hangers have also been used as shown in Figure 2.3. These diagonal hangers result in smaller deflections and a reduction in the bending moments in the arch rib and deck [11].

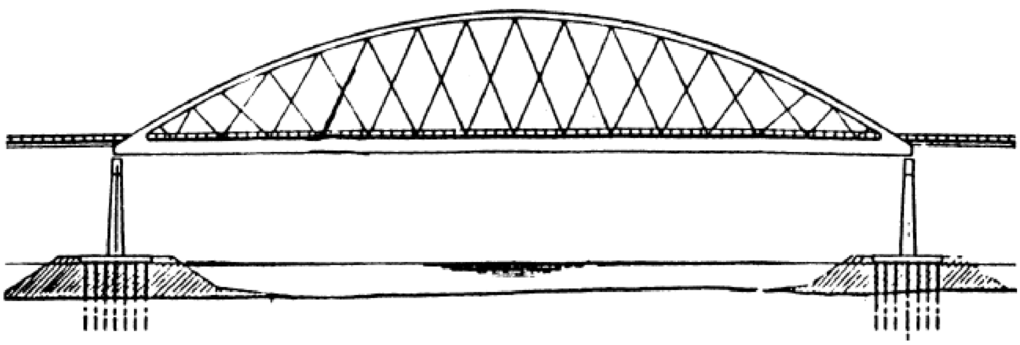


Figure 2.3: Arch with Diagonal Hangers [11]

2.3 Network Arch Bridges

Network arch bridges can be defined as the arch bridges with inclined hangers and multiple intersections. Beneficial structural behavior of this kind of bridge leads to economical bridge members mainly subjected to axial forces. Furthermore, the high stiffness and therefore small deflections favor the application of network arches for high speed railway as well as roadway transportations. Network arches seem to be very competitive for road bridges of spans of 135 to 160 m [13]. Construction of network arches can bring economic advantages due to significant savings of steel compared to other arch bridges. Compared with conventional bridges, the network arch, where the tie is a concrete slab, usually saves over two thirds of the steel weight. Figure 2.4 shows significant reduction of steel weight in network arch bridges compared with amount of steel in different types of highway bridges which was modified by Tveit [6] over compilation of steel weight required in various highway bridges by Herzog [14]. According to Tveit [6], if the Avik Sound bridge is designed in a network arch arrangement then total steel weight required for the bridge is the dot in Figure 2.4.

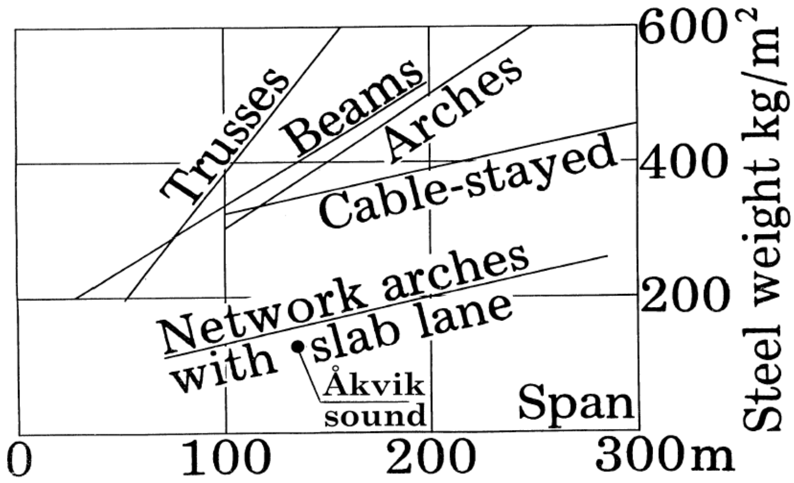


Figure 2.4: Amount of Steel in Various Types of Highway Bridges [13]

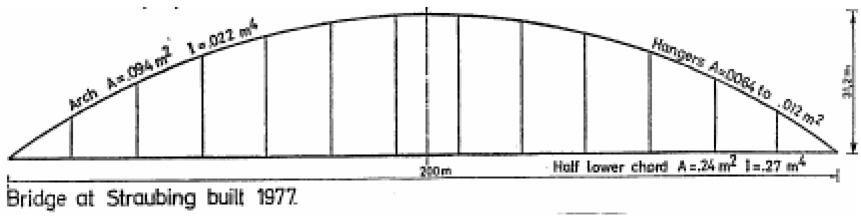
In 2003, Tveit [13] showed that a network arch could be seen as a simply supported beam. The arch is the compression zone, and the tie is the tension zone. The hangers are the web. Most of the shear force is taken by the vertical component of the compressive force in the arch. Some of the variation in the shear force is taken by the hangers. The arrangement of the hangers has considerable influence on the structural behavior. The arrangement governs on the forces and force variations within the network arch depending on many parameters, as for example span, rise, number of hangers, loading or arch curvature. Tveit [13] introduced an optimized hanger arrangement of the simplified network arch with regard to the mentioned parameters. This improved hanger arrangement provides a simple method of designing network arches with small hanger forces and small bending moments in the arch.

In 2003, Brunn et al. [7] adopted a trial and error based optimization for designing hanger arrangement. Brunn proposed that the best hanger arrangements would be achieved if hanger intersections lie on the radii of the arch circle. In addition Tveit applied a radial oriented hanger arrangement to the Avik Sound bridge and showed that radial hanger arrangement comprises influence line ordinates of 98% axial force in arch, 79% hanger force and 81% bending moments in arch of the values of Avik Sound network arch [9]. Tveit also showed in [8] that radial hanger arrangement provides significant reduction of influence line ordinates of bending moment in arch and deck over vertical hanger arrangement. He applied network arch bridge concept on the bridge of Straubing built in 1977 (Figure 2.5). Figures 2.5 (c) and (d) show significant reduction of bending moment in arch and deck for the network arch bridge over the existing bridge of Sraubing.

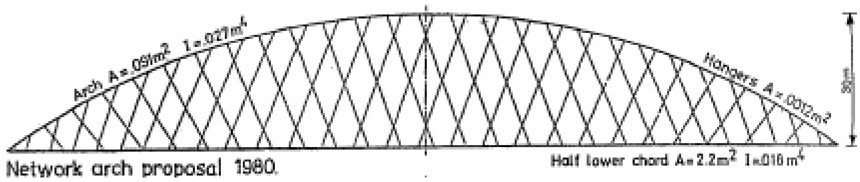
One of the assumptions to achieve optimal hanger arrangement adopted by Brunn et al. [7] and also by Tveit [13] is that the arch is part of a circle because it makes fabrication easy. In 2008, Tveit [8] defined optimal network arches as arch bridges where some hangers cross other hangers at least twice. The fact believed by Tveit for using crossing hangers is that for an evenly distributed load a parabolic arch with vertical hangers is a very good solution. However for uneven loads it is often better to use crossing hangers like network.

Optimization of hanger arrangement by Brunn et al. [7] stands on a hypothesis which assumes that radial arrangement of hangers minimizes bending in the arch about the horizontal axis. Brunn showed that for circular arches a uniformly distributed load that is directed along the radii of the arch circle causes no bending moments in the arch. If the hangers transfer forces to the arch at their nodal points shown in Figure 2.6 in such a way that direction of resultant of hanger forces stay radial then bending in arch will be the lowest.

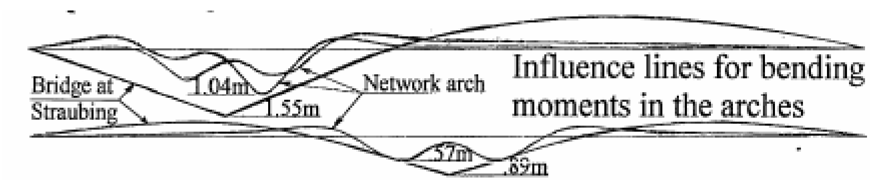
Brunn et al. [7] concludes in his research that the ratio between the number of hangers and the span [in meters] should not exceed 0.48 for a span lower than 100 m. For longer spans a lower ratio and for shorter spans a higher one will give reasonable results. For railway bridges live to dead load ratio hardly affects hanger slopes. Brunn found that the smallest variation in arch bending moments occurs if all hangers cross the arch at an angle, θ of about 45º shown in Figure 2.7. For the smallest variation in hanger forces, hangers need to be steeper according to Brunn. Brunn also states that a higher rise is advantageous than lower rise that minimizes bending moments in arch and in deck.



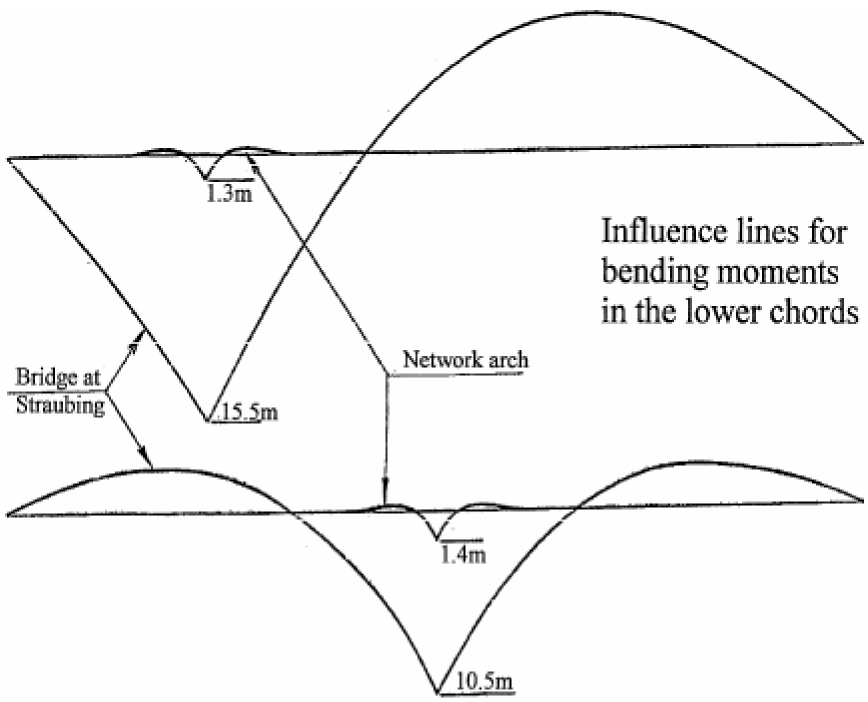
(a) Bridge at Straubing Built in 1977



(b)Network Arch Bridge Proposal in 1980

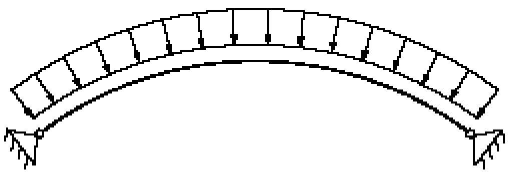


(c) Comparison of Influence Line of Bending Moment in Arch

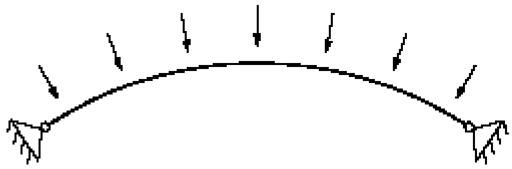


(d) Comparison of Influence Line of Bending Moment in Deck

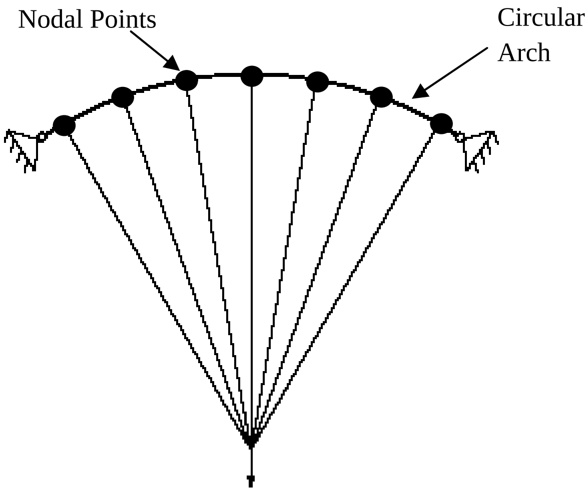
Figure 2.5: Influence Line for the Lower and Upper Chord of Two Tied Arch Bridges [13]



(a) Uniformly Distributed on Arch



(b) Discretisation of the Uniformly Distributed Load



(c) Discretized Load of (b) is Tranferred from Hangers to Circular Arch

Figure 2.6: Transformation of Vertical Loads to the Arch Nodal Points in Radial Direction [7]

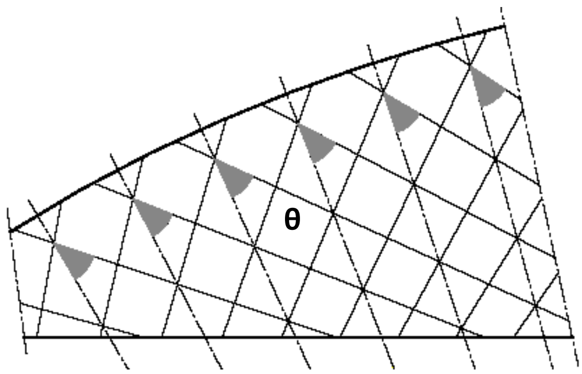


Figure 2.7: Cross Angle, θ Adopted by Bunn et al. [7]

The study to obtain optimal hanger arrangement by Brunn et al [7] was followed by a hypothesis regarding arch geometry and hanger arrangement, and traditional method of design. In general, the traditional optimization approaches have difficulties in dealing with global optimization problems. One of the main limitations of traditional optimization techniques is that they can easily be entrapped in local minima. Moreover, these techniques cannot generate or even use the global information needed to find the global minimum for a function with multiple local minima. Global optimization of network arch bridge system considering all possible design variables has not yet been performed. Therefore, engineering simulation using global optimization technique for optimization of network arch bridge is required. A short description of global optimization methods and its various types are outlined in the next article.

2.4 Global Optimization Method

An optimization technique transforms the conventional design process of trial and error into a formal and systematic procedure that yields a design that is the best possible solution of an optimization problem; mathematically, minima or maxima of an objective function. In real-life design problems, functions of many variables have a large number of local minima and maxima that is not the best possible solution. To find out an arbitrary local optimum is relatively straightforward by using local optimization methods or gradient based methods. Finding the global maximum or minimum i.e. optimum values of a function is much more challenging in real-life design problem. The challenge can be overcome by proper use of global optimization algorithms for obtaining global optimum values. The global optimization algorithms are a branch of applied mathematics and numerical analysis that employ measures that prevent convergence to local optima and increase the probability of finding a global optimum.

2.4.1 Classification of Global Optimization Methods

Generally, global optimization algorithms can be divided in two basic classes [15]:

- **i)** Deterministic and
- **ii)** Probabilistic algorithms.

In each execution step of a deterministic algorithm, there exists at most one way to proceed. This algorithm does not contain instructions that use random numbers in order to decide what to do or how to modify data. Deterministic algorithms are most often used if a clear relation between the characteristics of the possible solutions and their effectiveness for a given problem exists. Then, the search space can efficiently be explored using for example a divide and conquer scheme. Deterministic algorithms will always produce the same results when given the same inputs. But if the relation between a solution candidate and its fitness are not so obvious or too complicated, or the dimensionality of the search space is very high, it becomes harder to solve a problem deterministically. Adopting deterministic methods for solving such problems may result in exhaustive enumeration of the search space, which is not feasible even for relatively small problems.

A probabilistic algorithm includes at least one instruction that acts on the basis of random numbers. In other words, a probabilistic algorithm violates the constraint of determinism. There are two general classes of randomized algorithms [15]:

i) Las Vegas and
ii) Monte Carlo algorithms.

A Las Vegas algorithm is a randomized algorithm that never returns a wrong result. Either it returns the correct result, reports a failure, or does not return at all. If a Las Vegas algorithm returns, its outcome is deterministic (but not the algorithm itself). The termination however cannot be guaranteed. There usually exists an expected runtime limit for such algorithms – their actual execution however may take arbitrarily long. In summary, it can be said that a Las Vegas algorithm terminates with a positive probability and is (partially) correct.

A Monte Carlo algorithm always terminates. Its result however can be correct or incorrect. In contrast to Las Vegas algorithms, Monte Carlo algorithms always terminate but are (partially) correctly only with a positive probability.

Heuristics used in global optimization are functions that help decide which one of a set of possible solutions is to be examined next. On one hand, deterministic algorithms usually employ heuristics in order to define the processing order of the solution candidates. An example for such a strategy is informed search. Probabilistic methods, on the other hand, may only consider those elements of the search space in further computations that have been selected by the heuristic. A rough taxonomy of global optimization methods is depicted in Figure 2.8.

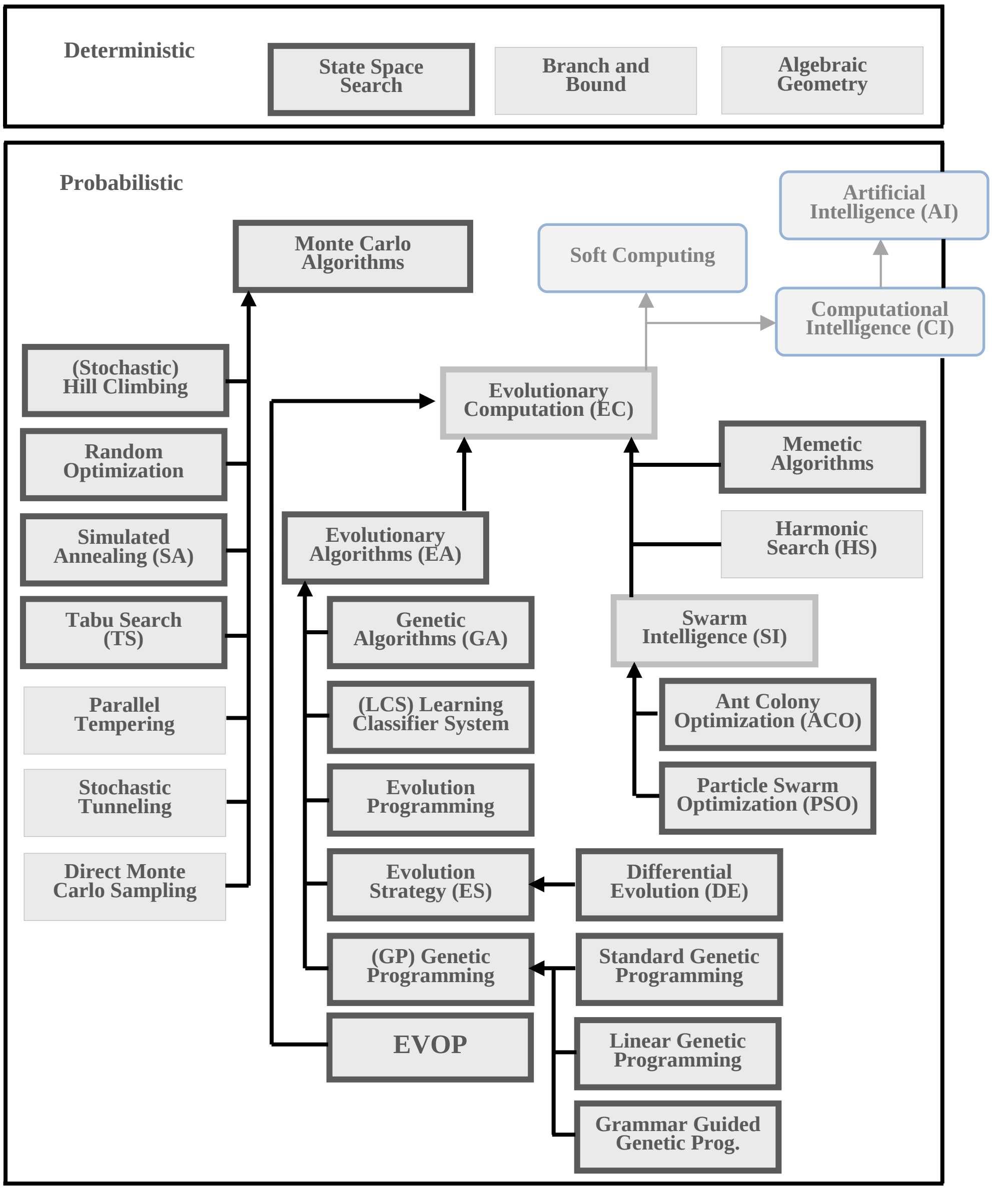


Figure 2.8: Taxonomy of Global Optimization Algorithms [15]

Out of global optimization algorithms, evolutionary operation EVOP is a probabilistic type optimization algorithm for constrained parameter optimization. The idea of EVOP for optimization is to apply a continuous and systematic scheme of small changes in the control variables of a process. The effects of these modifications are evaluated and the process is slowly shifted into the direction of improvement. EVOP is originally introduced by Box [16] in late 1950s. In 1989, Ghani [9] developed a new approach of EVOP which has high probability to seek the best possible solution of an optimization problem. The optimization process of EVOP developed by Ghani

[9] is described in appendix A. Capabilities of EVOP by Ghani in solving optimization problems is outlined here.

i) The method has high probability for locating the global minimum within few restarts.
ii) The method is good to minimize functions of mixed continuous, discrete and integer variables. It is ideally suited for system optimization of real life design.
iii) It does not require training and initial known population size as required by genetic algorithm.
iv) Ability to deal with nonlinear objective and constraining functions directly without the requirement of gradients or sub-gradients.
v) Objective and constraining functions can possess finite number of discontinuities.
vi) Scaling of objective and constraining functions are not necessary.

EVOP developed by Ghani [9] has been used in the optimization problem of network arch bridges. Optimization is driven by simulation and in the following article past works of simulation driven optimization is described.

2.5 Simulation Driven Optimization

Simulation driven optimization in design is defined as optimization process intended for design where the major functions and related processes of design are verified and optimized with the support of computer based product model simulations. Simulation is used here in the meaning of imitating the behavior of a real system by constructing and experimenting with a model of the system, where a model is defined as a simplified representation or description, describing only those aspects considered to be relevant in the context of one point of view. For non-trivial problems, the simulations normally have to be performed with numerical models and computer based tools.

To be able to analyze and to simulate the behavior of real engineering problems with reasonable accuracy, one has to rely on numerical techniques. These techniques are based on discretization of the whole geometrical domain, or the part of it that is relevant. Common numerical techniques of continuum mechanics are the finite difference method (FD), the boundary element method (BE), and the finite element (FE) method. These discretization based simulation techniques are very general. With the tools available today, it is possible to model and to solve most types of engineering problems.

The finite element method (FEM) is the best known, and by far the most widely used discretization method. Many thousands of engineers throughout the world make use of this technique on a daily basis. For that reason, design automation with finite element analysis as a simulation and evaluation tool is becoming more and more desirable. The ability to do automatic design iteration has constantly been a popular research and engineering topic. Various works have already been accomplished for design optimization of finite element models.

Honeywell Engines & Systems, Phoenix, has automated the design process by integrating numerical optimization programs with ANSYS [17]. For design optimization of an automotive universal joint considering manufacturing cost Cristello and Kim [18] used MATLAB to conduct the Sequential Quadratic Programming (SQP) optimization and executed finite element analysis by ANSYS batch solver for structural response.

In 2000, Roman et al. [19] applied Genetic Algorithms (GA's) to solve typical problems in structural optimization, integrating ANSYS on a User Programmable Features (UPF) level to evaluate the objective function. Hepworth et al. [20] developed an automated composite finite element tool in a commercial optimization and process flow software called iSIght by SIMULA to optimize the modal frequencies of a composite wing model of aircraft using Genetic Algorithm. The iSight-FD process flow is as shown in Figure 2.9.

The ANSYS program uses two optimization methods to accommodate a wide range of optimization problems [21]:

- The Subproblem approximation method is an advanced zero-order method that can be efficiently applied to most engineering problems.

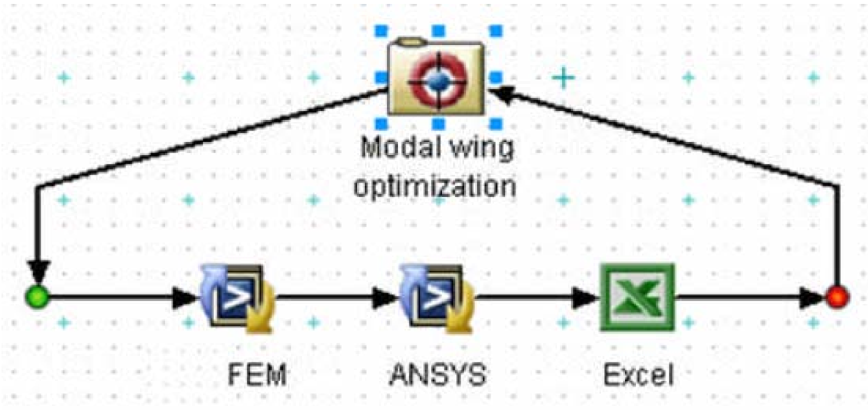


Figure 2.9: iSight-FD Process Flow [20]

- The first order method is based on design sensitivities and is more suitable for problems that require high accuracy.

For both the Subproblem approximation and first order methods, the program performs a series of analysis-evaluation-modification cycles. That is, an analysis of the initial design is performed, the results are evaluated against specified design criteria, and the design is modified as necessary. The process is repeated until all specified criteria are met.

In addition to the two optimization techniques, the ANSYS program offers a set of strategic tools that can be used to enhance the efficiency of the design process. For example, a number of random design iterations can be performed. The initial data points from the random design calculations can serve as starting points to feed the optimization methods described.

The basic architecture of optimization data flow [21] followed by the above methods is shown in Figure 2.10.

Usually structural deigns follows iterative design optimization which follows consecutive iteration which is shown in Figure 2.11. ANSYS has the capability for simulation driven optimization which follows the optimization process shown in Figure 2.12.

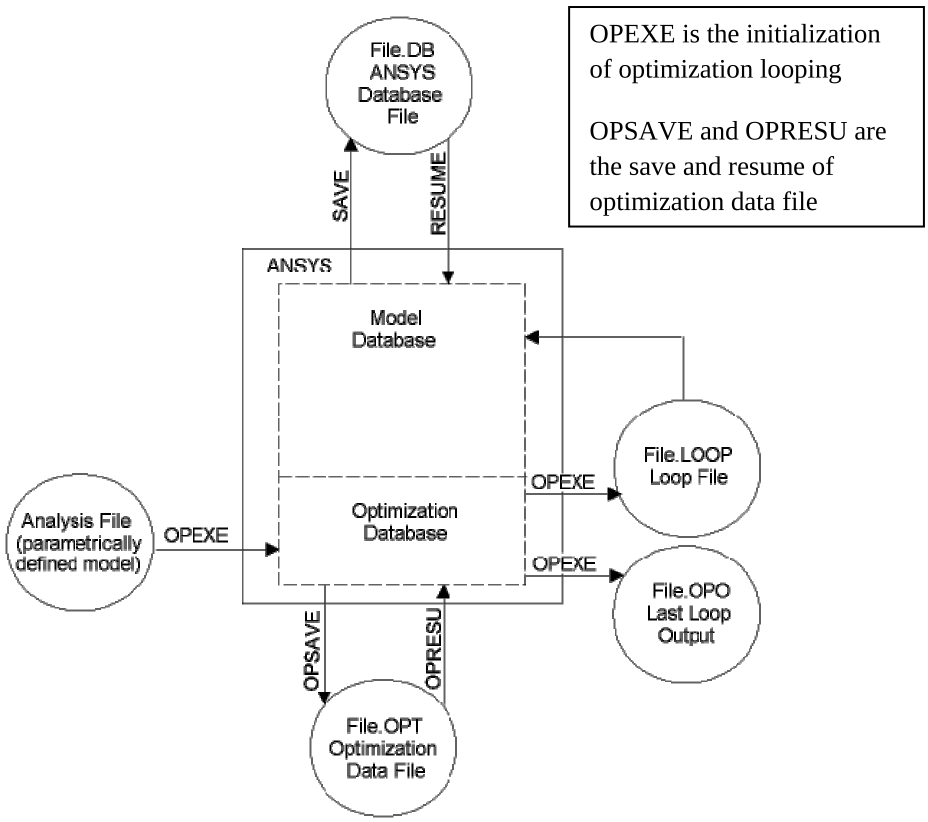


Figure 2.10: Optimization Data Flow in ANSYS [21]

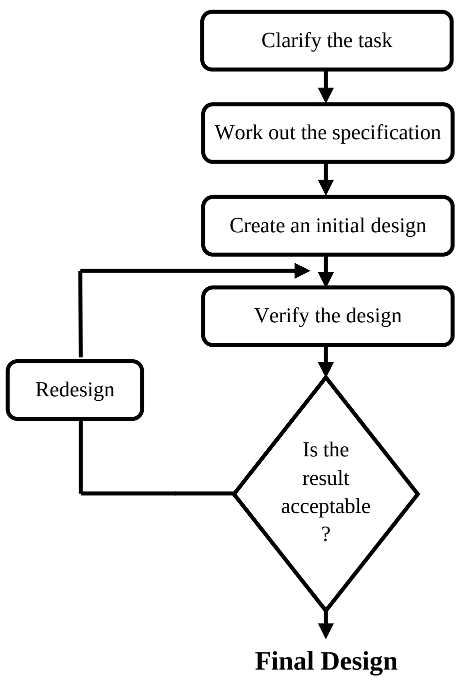


Figure 2.11: Iterative Design Optimization [22], [23]

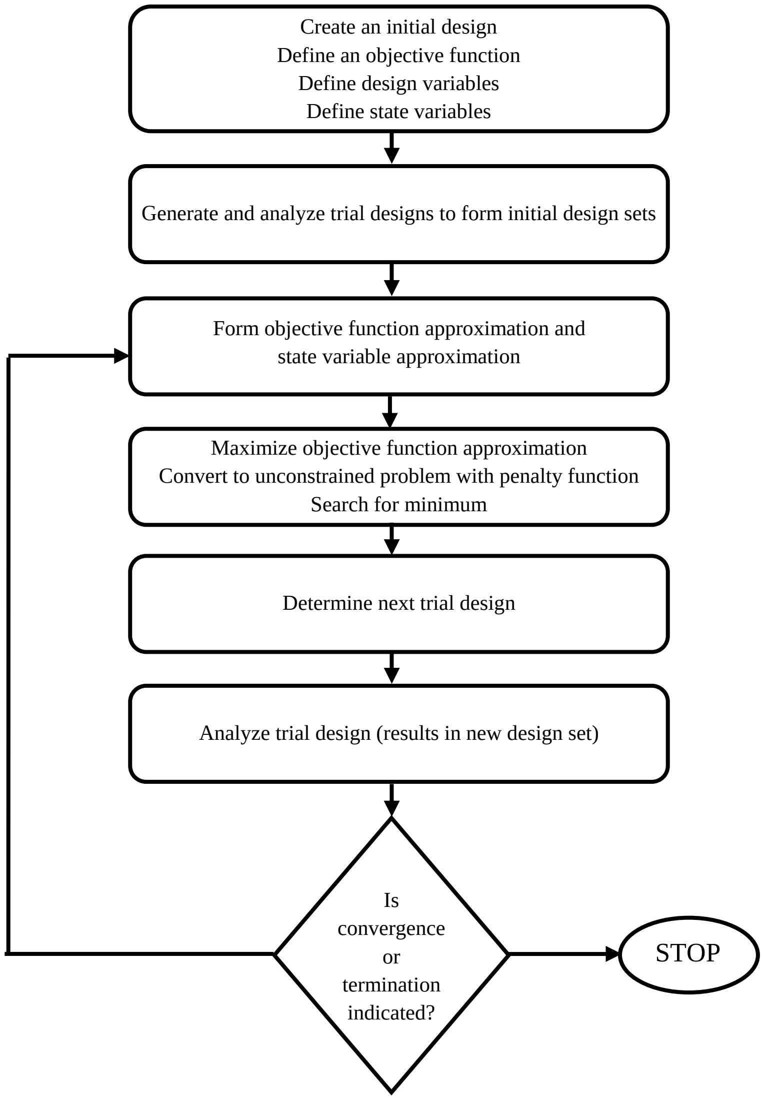


Figure 2.12: Design Optimization Process in ANSYS [24]

Optimization process in ANSYS is gradient based and optimum results may be entrapped in local minima instead of global minima [21]. Several global optimization methods which are non gradient based such as genetic algorithm, evolutionary algorithm, pattern search method and surrogate based method is excellent to seek global optimum. The basic optimization process of these methods referred by Kiemele et al. [25] is shown in Figure 2.13.

2.6 Conclusion

Textbooks suggest that arch bridges with inclined hangers result in smaller deflections and a reduction in the bending moments in the arch rib and deck. Some works already

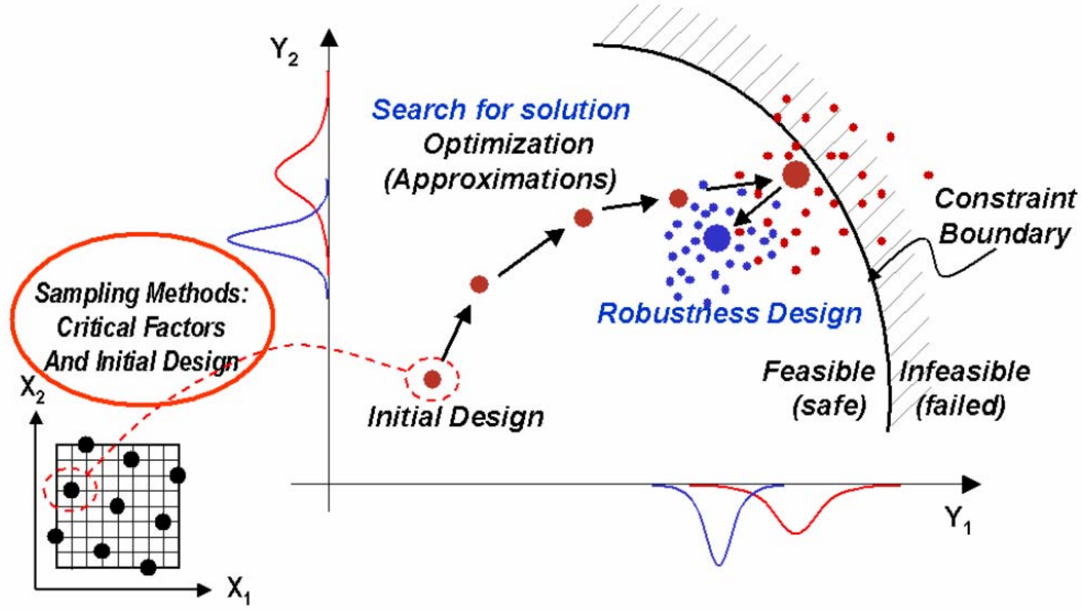


Figure 2.13: Design Optimization Flow in Non Gradient Based Methods [25]

have been performed regarding optimization of hanger arrangement to minimize this bending moment in arch. The works are based on some hypothesis and optimization is based on trial and error. To identify the optimum design parameters which are the best possible solution for minimization of cost of these bridges, an optimization approach is required. Simulation driven optimization using global optimization method may be a suitable method for obtaining most economical design of such kind of bridges.

CHAPTER 3

FORMULATION OF OPTIMIZATION PROBLEM

3.1 General

Most economical design of network arch bridges satisfying all design criteria makes up of an optimization problem. The present chapter formulates this problem statement systematically for proper utilization of the problem in an optimization algorithm. Objective function, explicit and implicit constraints required for optimization algorithms are described in the following sub-sections of this chapter.

3.2 Optimal Design Problem Statement

The optimization problem has been formulated as a mixed integer-discrete nonlinear programming problem and solved using a global optimization algorithm. The problem formulation is as follows.

find design variables $$x_k = \{x_1, x_2, x_3 .... x_{NDIV}\} \quad (3.1)$$

to minimize objective function $f(x)$

subjected to implicit constraints $$G(x)_i^L \le G(x)_i \le G(x)_i^U \quad (3.2)$$

with lower and upper bounds of design variables $$x_k^L \le x_k \le x_k^U \quad (3.3)$$

where *i = 1, 2… … NIC*, *k=1, 2 … … NDIV* and *NIC* is total no of implicit constraints, *NDIV* is total no of design variables. $G(x)_i$ and *f(x)* are implicit constraints and objective function respectively for response of design variables x_k. The functions $G(x)_i$ and *f(x)* are to be evaluated from finite element simulation.

The optimization problem will use constraints derived from the AASHTO [26] and AISC [27] design equations. Implicit constraints will limit maximal hanger forces, maximal stresses and displacements in both arch and deck and explicit constraints will limit the design variables.

3.3 Objective Function

The objective of the present study is cost minimization of superstructure of network arch bridges by considering the total cost of hangers and arch including cost of materials, fabrication and installation. Unit costs used in this study are based on RHD [28]. Total cost, C_T is defined according to the following relationship:

$$C_T = C_{HC} + C_{AC} + C_{AS} \tag{3.4}$$

Where C_{HC}, C_{AC} and C_{AS} are the costs of cable of hangers, concrete section of arch and amount of reinforcement required all over the arch respectively. Costs of individual components are calculated as:

$$C_{HC} = UP_{HC}W_{HC} \tag{3.5}$$

$$C_{AC} = UP_{AC}V_{AC} \tag{3.6}$$

$$C_{AS} = UP_{AS}W_{AS} \tag{3.7}$$

Where, UP_{HC} , UP_{AC} and UP_{AS} are the unit prices of materials, fabrication and installation of hangers, concrete of arch and the reinforcement required in the arch respectively; W_{HC}, V_{AC} and W_{AS} are the weight of the cables, volume of the concrete required in arches and weight of reinforcement required in arches respectively.

3.4 Design Variables and Design Constant Parameters

For a particular span and width of arch bridge, a large number of parameters control the design such as location of hanger nodes along the arch, location of hanger nodes on the deck, number of hangers, cross sectional area of cable of hanger, cross sectional dimensions of arch and rise of the arch. These control parameters are explained as design variables in Table 3.1.

Design constant parameters under consideration such as various material properties, superimposed dead loads, AASHTO live load, rectangular shape of arch cross section, arch span and unit costs of materials including fabrication and installation etc. are enlisted in Table 3.2. Figure 3.1 shows all the design variables and constant parameters considered in the study.

Table 3.1: Design Variables with Explicit Constraints

Design Variables	Variable Type	Explicit Constraint	
		Lower bound	Upper Bound
Number of Hangers (Each Arch), N_h	Integer	4	60
Start Angle for Hanger Inclination Set1, $\varphi 1$	Continuous	0°	80°
Start Angle for Hanger Inclination Set2, $\varphi 2$	Continuous	0°	80°
Angle Change for $\varphi 1$, $\Delta\varphi 1$	Continuous	-2°	2°
Angle Change for $\varphi 2$, $\Delta\varphi 2$	Continuous	-2°	2°
Cross Sectional Area of Cable of Hanger, A_h	Discrete	96.8 mm^2	2929 mm^2
Arch Width, B_h	Discrete	250 mm	3000 mm
Arch Depth, H_h	Discrete	250 mm	4000 mm
Rise to Span Ratio, R_h	Continuous	0.14	0.25

Table 3.2: Design Constant Parameters

	Design Constant Parameters	Value
Material Properties	Modulus of Elasticity for Concrete, E_C	24800 MPa
	Poisson's Ratio of Concrete, υ	0.2
	Concrete Compressive Strength at 28 days (Arch and Bracing), f_{ca}'	25 MPa
	Concrete Compressive Strength at 28 days (Deck), f_{cd}'	50 MPa
	Ultimate Strength of Cable of Hanger, f_u	1520 MPa
	Arch Reinforcement Yield Stress, f_y	413 MPa
Geometric Properties	Span, L	100 m
	Width of Bridge, B_w	10 m
	Arch Section	Rectangular
	Arch Shape	Circular, Parabolic
Loading	Standard Vehicle Load	AASHTO HS 20-44 Single and Lane
	No of Lane	Two
	Wearing Surface	1436 N/m^2
Cost Parameters	Unit Price for Cable of Hanger, UP_{HC}	223 Tk/kg [28]
	Unit Price for Concrete of Arch, UP_{AC}	10527 Tk/m^3 [28]
	Unit Price for Reinforcement of Arch, UP_{AS}	80 Tk/kg [28]
General	Design Code	AASHTO 2002, AISC 2005

Hanger arrangement adopted in the study is based on equidistant nodes along the arch from which the hangers slope down with a certain inclination until they reach the tie. The design variables describing the hanger arrangement are the start angle and the change of the inclination from one hanger to the next. Two sets of start angle φ_1 and φ_2 and the corresponding angle change within the specified range in Table 3.1 enables

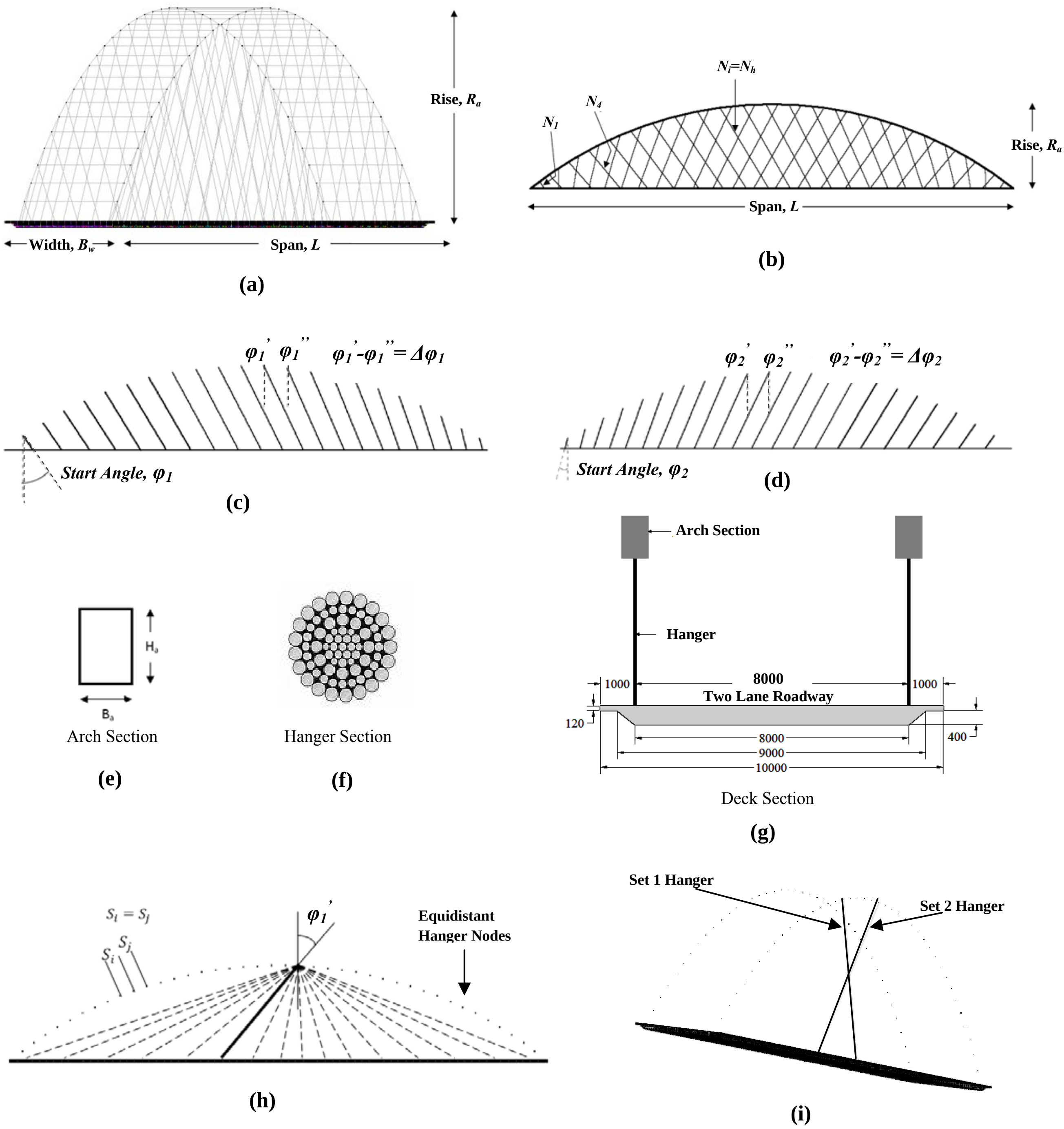


Figure 3.1: Network Arch Bridge and Design Variables (a) 3D View of Network Arch Bridge (b) Typical Hanger Arrangement Showing Numbering of Hangers(c) Hanger Set 1 and Its Inclination with Vertical (d) Hanger Set 2 and Its Inclination with Vertical (e) Arch Section (f) Hanger Section (g) Deck Section (h) Alternate Hanger Position from One of Equidistant Hanger Nodes along the Arch (i) Two Sets of Hanger

any type of hanger arrangement. Figure 3.1 (h) shows equidistant hanger nodes along the arch from which any hanger nodes on arch project hanger depending upon inclination of that hanger, φ'_{ij} and

$$\varphi'_{ij} = \varphi_j + i\Delta\varphi_j \tag{3.8}$$

Where, *i* is the hanger number along the arch,

j=1 for set 1 hanger and 2 for set 2 hanger

φ_j= Start angle for set 1 and set 2 hangers

$\Delta\varphi_j$= Angle change for set 1 and set 2 hangers

3.5 Explicit Constraints

These are specified upper and lower bounds of design variables shown in Table 3.1 which are derived from geometric requirements, availability of materials and practical considerations. The constraint is defined as

$$x_k^L \leq x_k \leq x_k^U; \qquad k = 1, \dots \dots \dots, NDV \tag{3.9}$$

Where x_k^U and x_k^L are upper and lower bounds of design variable, x_k.

3.5.1 Explicit Constraints for Hanger Arrangement

Control parameters for hanger arrangement are the location of hanger nodes along the arch and the location of hanger nodes on the deck. The design variables for controlling these two parameters are two set of start angle φ_1 and φ_2 for two set of hanger respectively which is shown in Figures 3.1 (c) and 3.1 (d) and the corresponding change of the inclination $\Delta\varphi_1$ and $\Delta\varphi_2$ from one hanger to the next for two sets of hanger respectively. Set 1 hangers will have to produce start angle φ_1 anticlockwise with vertical and set 2 hangers will produce start angle φ_2 clockwise with vertical. The lower limit of start angle for both cases is taken as 0° which represents the start vertical hanger. The upper limit of start angle for both sets is taken as 80° which represents maximum possible hanger location on deck for the hanger projecting from start node of the arch. Inclination of hangers varied from -2° to +2° for both sets of hanger which enables various hanger arrangements.

Vehicle load is symmetrically applied for reduction of solution time of finite element analysis. Therefore symmetric hanger arrangement is used by taking design variables $\varphi_2 = \varphi_1$ and $\Delta\varphi_2 = -\Delta\varphi_1$.

3.5.2 Explicit Constraints for Number of Hangers

Effect of number of hangers on optimization of arch and hangers is studied in the present study. For this study it is considered that number of hangers in each arch may vary from 4 to 60.

3.5.3 Explicit Constraints for Cable Section of Hanger

Design variable for cable section of hanger is considered as cable cross sectional area according to ASTM A586 standard [29]. The lower limit of cross section of cable according to the standard is taken as 96.8 mm^2 and upper limit of that of hanger is taken as 2929 mm^2.

3.5.4 Explicit Constraints for Arch Section

Design variables for rectangular arch section are the arch width and arch depth. Minimum 1% reinforcement is defined in the arch section and design reinforcement after analysis is constrained in the implicit function. The lower limit for both arch width and arch depth is taken as 250 mm and upper limit of arch width and arch depth are taken as 3000 mm and 4000 mm respectively. Buckling in the plane of arch and moment magnification plays important roles to determine the arch depth [30]. Ratio of span to rib depth for most of the existing arch bridges is 70 to 80. Nettleton [30] suggested a ratio 75 which is a good average figure for concrete arch. For some architectural reasons the ratio may be lower such as in the Arroye Seco Bridge, this ratio is 53. Again the ratio of springing depth to crown depth for most two hinged arch is from 1.55 to 1.72 whereas Nettleton suggested a value of that ratio 1.7. The present study considered same springing and crown depth all over the section. According to AASHTO [26] arch ribs is to be reinforced as compression members and confinement reinforcement is to be provided as required for columns. So the lower limit for both is considered as 250 mm from reinforced concrete design consideration.

3.5.5 Explicit Constraints for Arch Rise

Design variable for arch rise is considered in terms of arch rise to span ratio. The rise to span ratio for arches varies widely. The ratio for almost all arch bridges is within the range of 0.16 to 0.22 [11]. Though a tied arch with suspended roadway over a navigational channel has full freedom to vary the rise to span ratio to suit economy and appearance but most of the tied arches have a rise to span ratio of about 0.2 [30]. The reason for limiting this ratio is the economy. Though an increase of rise decreases arch thrust inversely with the rise to span ratio, reducing the axial stress from dead and live load, the bending stress from temperature change and the axial tension in the tie but offsetting these effects from the standpoint of economy is the increased length of the arch rib. This greater length increases the quantity of material and dead load. It also increases the buckling length in the plane of the arch and moment magnification factor. The lengths of suspenders are increased. The total length of lateral bracing between the ribs is increased, and the wind overturning and stresses are increased. As the optimization module considers economy for rise to span ratio as a design variable, cost minimization of arch and hanger is the objective of optimization algorithm. Then explicit constraints regarding this parameter include most available ranges of arch bridges. The lower and upper limit of rise to span ratio considered in this study is taken as 0.14 to 0.25 respectively.

3.6 Implicit Constraints

These constraints limit response of the bridge. A total seven implicit constraints are considered according to the AASHTO [26] Standard Specifications, AISC [27] design equations and physical constraints which are enlisted in Table 3.3. These constraints are categorized into four groups:

- **i)** Constraints regarding design of hangers
- **ii)** Constraints regarding design of arch
- **iii)** Constraints regarding hanger position

iv) Constraints regarding in plane stability of arch

Table 3.3: Implicit Constraints

Response	Implicit Constraint	
	Lower bound	Upper Bound
Extreme Hanger Stress, σ_{max}	0	0.75*F_u or 0.9* Fy
Strength Criteria of Arch, CRT	0	1
Design Reinforcement Factor in Arch, RNR	1%	8%
End Angle for Hanger Inclination Set1, $\varphi 1_n$	-80°	80°
End Angle for Hanger Inclination Set2, $\varphi 2_n$	-80°	80°
In Plane Slenderness ratio	0.1	22
Arch Depth/ Width Ratio	0.25	6

3.6.1 Implicit Constraints Regarding Design of Hangers

Constraint for maximal hanger stress is given by

$$f^L \leq f_i \leq f^U \tag{3.10}$$

Where, f^U is the upper limit and f^L is the lower limit of hanger stress respectively. f_i is the actual extreme hanger stress measured from finite element analysis of all available hangers for all possible load combinations and considering standard HS20-44 all vehicle positions acting on the deck. The upper limit of hanger stress is taken as design tensile strength of cable according to AISC [27] and lower limit is taken as zero considering cables are in tension only.

The design tensile strength, $\varphi_t\ P_n$ of hanger is the lower value obtained according to the limit states of tensile yielding in the gross section and tensile rupture in the net section according to AISC [27].

i) For tensile yielding in the gross section:

$$P_n = F_y A_g \tag{3.11}$$

$$\varphi_t = 0.9 \text{ (LRFD)}$$

ii) For tensile rupture in the net section:

$$P_n = F_u A_e \tag{3.12}$$

$$\varphi_t = 0.75 \text{ (LRFD)}$$

Where

A_e = Effective net area

A_g = Gross area of member

F_y = Specified minimum yield stress of the type of steel being used

F_u = Specified minimum tensile strength of the type of steel being used

LRFD = Load and Resistance Factor Design

3.6.2 Implicit Constraints Regarding Design of Arch

Design of arch is based on strength criterion. Strength criterion is the capacity ratio of the section calculated from column capacity interaction volume which measures the stress condition of the section [31]. The section capacity interaction volume is generated on the three dimensional interaction failure surfaces as shown in Figure 3.2. In addition to axial compression and biaxial bending, the formulation allows for axial tension and biaxial bending considerations.

The capacity ratio is basically a factor that gives an indication of the stress condition of the section with respect to the capacity of the column. In other words, if the axial force and biaxial moment set for which the section is being checked is amplified by dividing it by the reported capacity ratio, the point defined by the resulting axial force and biaxial moment set will lie on the failure (or interaction volume) surface. For determination of capacity ratio or strength criterion, ultimate strain state homothetic to the acting point, P (Figure 3.3) with respect to interaction volume centre is determined.

This strength criterion is defined as the ratio between the distance of the "centre" of the diagram (point A of the Figure 3.3) to the point which represents the acting forces and moments (point P of the Figure 3.3) and to the point which represents the homothetic ultimate forces and moments (point B of Figure 3.3). Then the criterion, CRT is defined as

$$CRT = \frac{d_1}{d_2} \tag{3.13}$$

If CRT=1, then the point lies on the interaction surface and the section is stressed to capacity.
If CRT<1, then the point lies within the interaction volume and the section capacity is adequate.
If CRT>1, then the point lies outside the interaction volume and the section is over stressed.

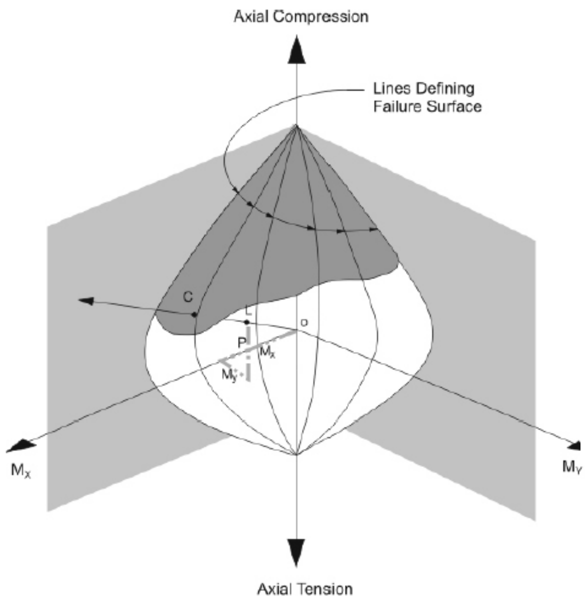


Figure 3.2: 3D Interaction Surface [31]

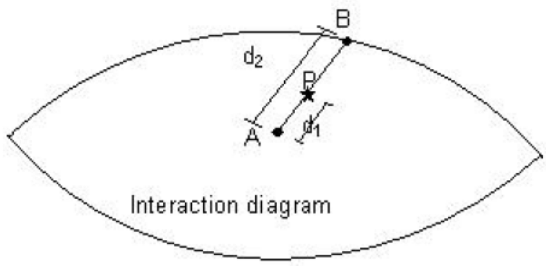


Figure 3.3: Interaction Diagram at Ultimate Strain State Homothetic to the Acting Point, P [32]

Constraint for design of arch is given by

$$CRT^L \leq CRT_i \leq CRT^U \tag{3.14}$$

Where, CRT^U is the upper limit of strength criterion of the arch section and CRT^L is lower limit of strength criterion of the section. CRT_i is the actual extreme strength criterion of the section measured from finite element analysis of all available elements of arch for all possible load combinations and considering standard HS20-44 of all vehicle positions acting on the deck. The upper limit of strength criterion of arch section, CRT^U is equals to 1.0 and lower limit of strength criterion of arch section, CRT^L is equals to 0.0.

Another constraint for design of arch is the upper and lower bounds of reinforcement percentage in arch. The constraint is defined by

$$RNR^L \le RNR_i \le RNR^U \tag{3.15}$$

RNR^U is the upper limit of reinforcement factor of the arch and RNR^L is lower limit of reinforcement factor of the section. RNR_i is the actual extreme reinforcement factor of the section designed after finite element analysis of all available elements of arch for all possible load combinations and considering standard HS20-44 loading of all vehicle positions acting on the deck.

The upper limit of reinforcement factor of arch section, RNR^U is equal to 8.0% and lower limit of reinforcement factor of arch section, RNR^L is equal to 1.0% [33].

For the design of sections under axial load and biaxial bending, an optimisation process is carried out for determination of reinforcement factor for the acting axial load and biaxial bending in each element of arch section through successive iterations, in which it is intended that the safety factor of the section will be 1.00.

3.6.3 Implicit Constraints Regarding Hanger Position

Constraints for hanger position are given by

$$POS^L \le POS_i \le POS^U \tag{3.16}$$

Where POS^U and $POS^L$are the upper and lower limit of end hanger inclination with vertical and POS_i is the end hanger inclination for each analysis and design case in the optimization process. End hanger inclination with vertical, POS_i is calculated as

$$POS_i = \varphi_1 + N_{h1} \times \Delta\varphi_1 \tag{3.17}$$

Where, N_{h1} , φ_1 and $\Delta\varphi_1$ are the number of hangers, start angle and is angle change for set 1 hangers respectively.

End hanger inclination represents the hanger location on deck for the hanger projecting from end node of the arch. Maximum absolute possible end inclination is taken within +80° to $-80°$. If the end hanger inclination is not within specified limit then some end hangers might not project from arch node to proper node on deck as shown in Figures 3.4 and 3.5.

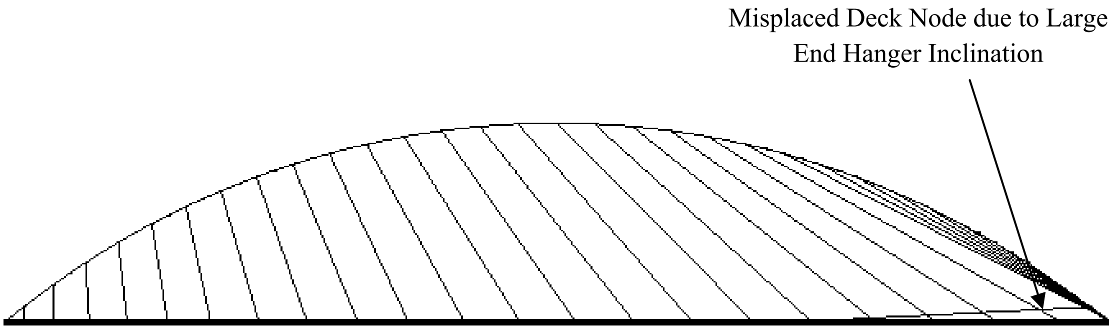


Figure 3.4: Effect of Large $\Delta\varphi_1$ on Hanger Arrangement

3.6.4 Constraints Regarding In Plane Stability of Arch

In plane stability of arch is measured for tied arch by the slenderness effects between points of lateral support and between suspenders in the vertical plane of arch with suspended roadway. Slenderness effect has been evaluated by approximate procedure in lieu of rational method as per AASHTO 2002 [26]. So implicit constraints regarding in plane stability of arch is considered for a maximum slenderness ratio of 22 considering that tied arch is not braced against side sway in the vertical plane of arch.

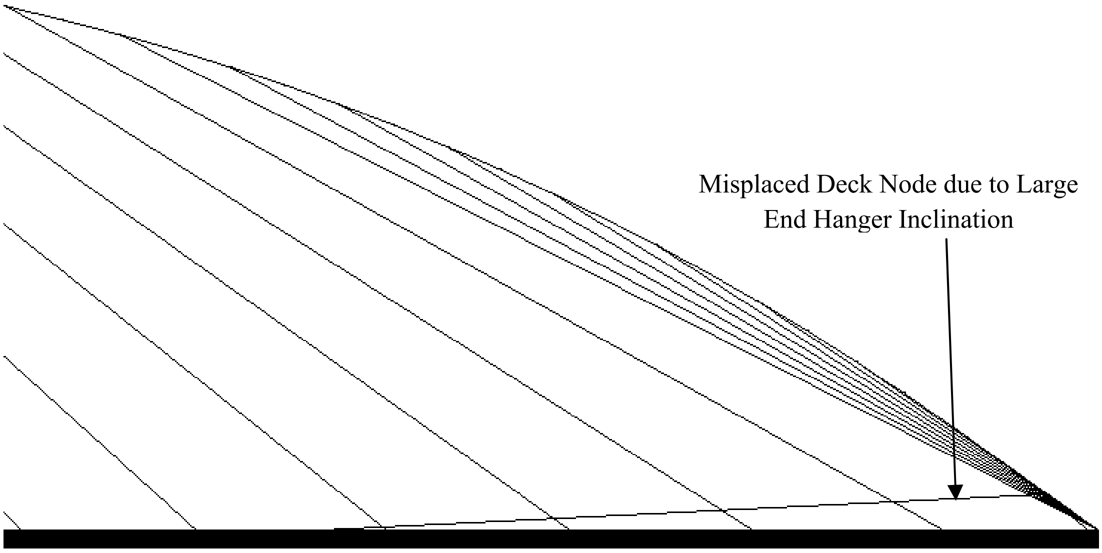


Figure 3.5: Misplaced End Hanger

3.7 Conclusion

Most economical design of network arch bridges may be achieved if constraints mentioned above are satisfied within a range of design constant parameters. The optimization problem formulated in this chapter may be used for various optimization methods however a global optimization technique will be incorporated to seek best possible solution.

CHAPTER 4

INTERFACING GLOBAL OPTIMIZATION ALGORITHM, EVOP WITH FINITE ELEMENT SOFTWARE, ANSYS

4.1 General

Design optimization using finite element analysis requires a loosely coupled or black-box interface between user-supplied simulation codes and optimization algorithms. The loosely coupled interface between them enables a dynamic data exchange by reading and writing short data files. The approach does not require access to the source code of user's simulation software which is advantageous for most structural optimization problem [34]. Same approach has been incorporated for optimization of network arch bridges by interfacing an optimization method, EVOP with a finite element software ANSYS. Data flow and methodology of interfacing developed in the study is described in this chapter. Some benchmark problems of optimization are also solved at the end of this chapter for verification of this interfacing.

4.2 Advantage of FEA Software, ANSYS for Interfacing with EVOP

The commercial ANSYS package was used for the FE meshing, modeling analysis module. The advantage of using ANSYS as a basic FEA package for structural optimization includes the following [17]:

i) It integrates pre-processing, post-processing and analysis solution in one package seamlessly.

ii) It provides named parameters for conveniently connecting variables and response between the optimizer and itself.

iii) It provides very powerful macro language. It is enough for accessing almost any database and response in the package.

iv) It provides interface to call the external programs in the macro language. This makes the interaction between the optimizer and the analysis easier.

The conclusion is that the flexibility of this package makes it reasonable for structural optimization purpose.

4.3 Interfacing ANSYS with EVOP in the Optimization Process

To interface the simulation with evolutionary operation, EVOP which is originally written in FORTRAN, a platform is established by visual C++. The platform is used to transfer the parameters from EVOP to the simulator input file and to extract the response values of interest from the simulator's output file for return to EVOP. The overall interfacing process is executed in the following architecture.

i) The user creates an input file for batch mode execution in ANSYS. The design variables, objective functions and constraints are then linked with named parameters in ANSYS.

ii) The optimization program written in FORTRAN reads in the input file through the platform written in visual C++ and detects the relationship between the design variables, objective function, constraint values and the named parameters inside ANSYS. The communication protocol is established in this way through the platform written in visual C++ which is linked with FORTRAN and as well as FE software ANSYS.

iii) When the optimization program proposes a new design, the value of the associated named parameters in the input is changed.

iv) The optimizer then feeds the input file into ANSYS, and the desired response is saved in parameters and written to a file. The optimizer gets the response by reading this file.

v) The optimizer selects a new design, and repeats step 2 to 4, until a convergence is reached.

In the whole optimization process, the platform in visual C++, EVOP in FORTRAN and ANSYS are interlinked in the process as shown in Figure 4.1.

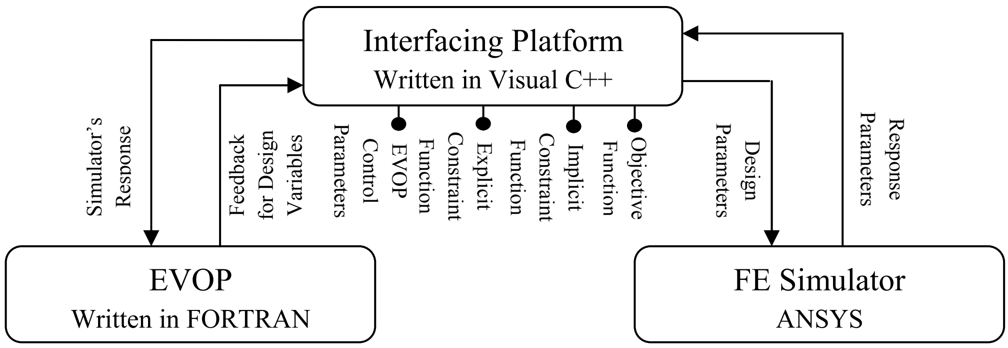


Figure 4.1: Black Box Interfacing

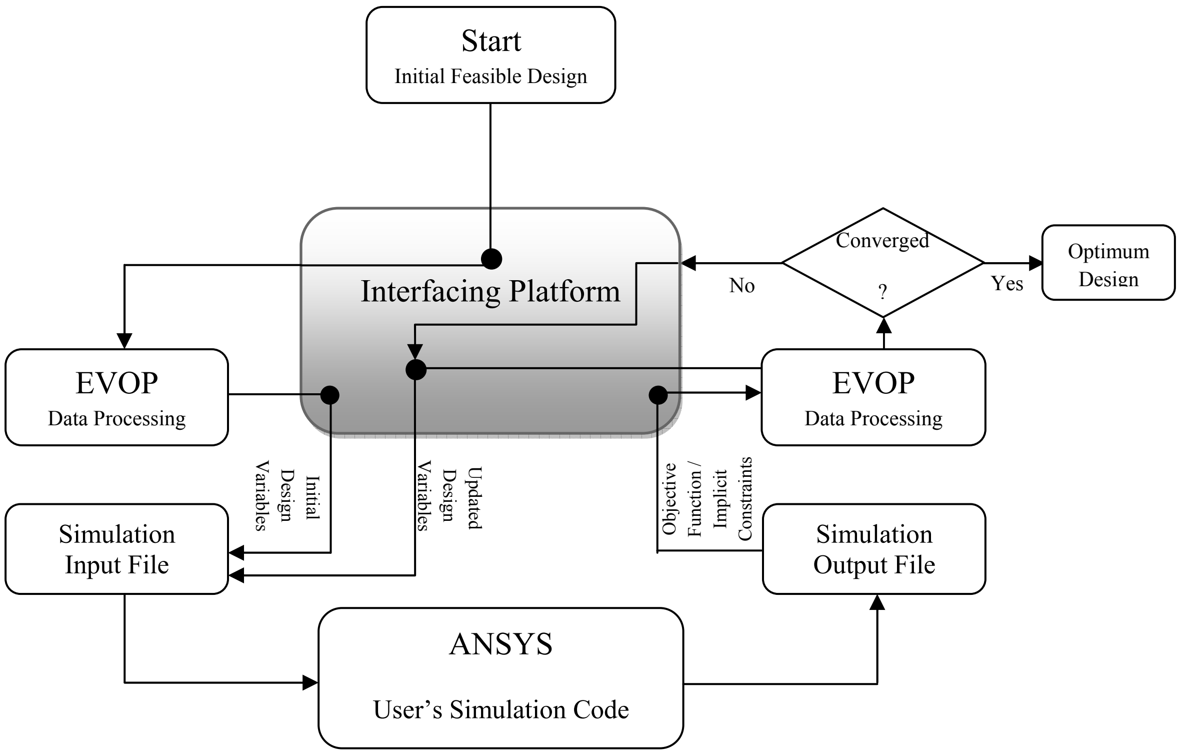


Figure 4.2: Optimization Flow Chart

Figure 4.2 shows optimization flow chart denoting file input/ output operations. An initial feasible design is invoked through the interfacing platform which runs EVOP and acts as platform for data structure definition. Design variables are intelligently allocated by EVOP. EVOP transfers the parameters to user's simulation code through the interfacing platform. Simulation output file provides necessary information for EVOP through platform and updated design variables are dynamically decided by the optimization platform for new simulation. The process is repeated until all of the simulation code runs required by the iterative study have been completed. The solution is then said to be converged and optimum results are recorded.

4.4 Verification of Interfacing

This article describes verification of interfacing ANSYS with optimization algorithm EVOP. An example of verification manual of ANSYS [35] "Shape Optimization of a Cantilever Beam" is solved to check the EVOP optimization process. Another example of shape optimization of a bracket by EVOP optimization process is described which also verifies the interfacing.

4.4.1 Test Case 1 for Structural Shape Optimization

The benchmark problem referred by Prasad [36] to minimize the weight (volume) of a cantilever beam shown in Figure 4.3 subject to an end moment, M is taken as first test case for structural shape optimization. Conditions for optimization are that the stress $\sigma_{v\ max}$ may not exceed 206.9 MPa anywhere nor may the deflection δy_{max} be greater than 12.7 mm. The thickness of the beam may vary along the length, with the thickness at the point of load application held constant at a value t.

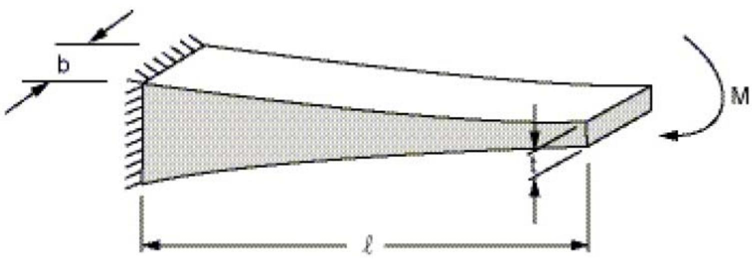


Figure 4.3: Cantilever Beam Problem Sketch

i) Problem Formulation

The problem is formulated similarly as in article 3.2.

ii) Objective Function

The objective of problem formulation is to minimize the volume of a cantilever beam subject to an end moment, M.

iii) Design Variables and Constant Parameters

Analysis of the cantilever beam shown in Figure 4.3 is sub structured by using half-symmetry solid model. Half model of the beam shown in Figure 4.4 is divided into four parts. Half symmetry is applied in X direction. Thickness of the beam at the point of loading is constant and thickness at four sections is considered as variable. Design variables and constant parameters for the problem are shown in Tables 4.1 and 4.2 respectively.

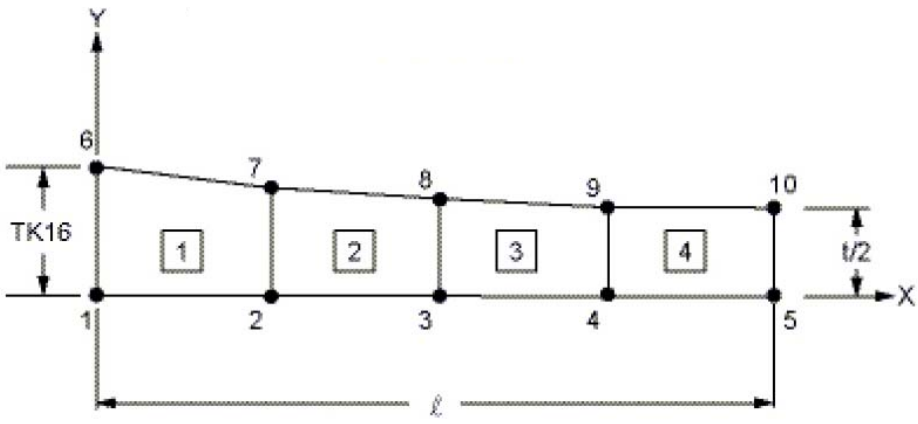


Figure 4.4: Half-Symmetry Solid Model, Showing Keypoints, Areas and Design Variable TK16 [35]

Table 4.1: Design Variables with Explicit Constraints for Test Case 1

Design Variables	Explicit Constraint	
	Lower bound	Upper Bound
TK_{16}	3.81 mm	6.86 mm
TK_{27}	3.81 mm	6.86 mm
TK_{38}	3.81 mm	6.86 mm
TK_{49}	3.81 mm	6.86 mm

Table 4.2: Design Constant Parameters for Test Case 1

	Design Constant Parameters	Value
Material Properties	Modulus of Elasticity, E_b	68.97 X 10^3 MPa
	Poisson's Ratio, υ_s	0.3
Geometric Properties	l	3.05 mm
	b	25.4 mm
	$t/2$	3.81 mm
Loading	M	50.84 N-m

iv) Implicit Constraints

According to the problem statement upper bound of stresses anywhere in the beam is taken as 206.9 MPa and maximum deflection of the beam must be less or equal to 12.7 mm. Implicit constraints are tabulated in Table 4.3.

Table 4.3: Implicit Constraints for Test Case 1

Response	Implicit Constraint	
	Lower bound	Upper Bound
Stress, σ_{max}	0	206.9 MPa
Deflection, δ_{max}	1.25 mm	12.7 mm

v) Finite Element Modeling

Half symmetry solid model of the beam is modeled with the help of ten keypoints shown in Figure 4.5. The keypoints defining the outer fiber are connected with a cubic spline. Translation in X and rotation about Z for the nodes associated along X direction are restrained for simulating half symmetry. Again translation in X for the nodes associated along Y direction is restrained for simulating support condition. Four noded 2-D Structural Solid Plane42 with two degrees of freedom at each node has been used in modeling the beam.

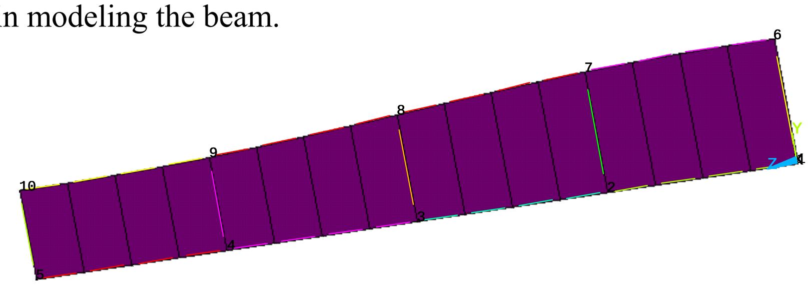


Figure 4.5: Half-Symmetry Finite Element Model, Showing Keypoints

vi) Optimization Process

Two separate ANSYS Parametric Design Language (APDL) scripts for implicit and objective functions are created and linked with the optimization engine EVOP through the interfacing platform. The problem is also solved independently using the subproblem approximation method and the first order gradient method of ANSYS, both using the same starting design. The optimal solution is recorded, and deflection and stress contour for optimal solution is depicted in Figures 4.6 and 4.7 respectively.

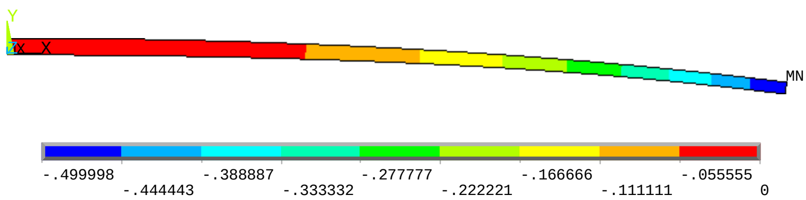


Figure 4.6: Contour of Deflection for Optimal Design Variables

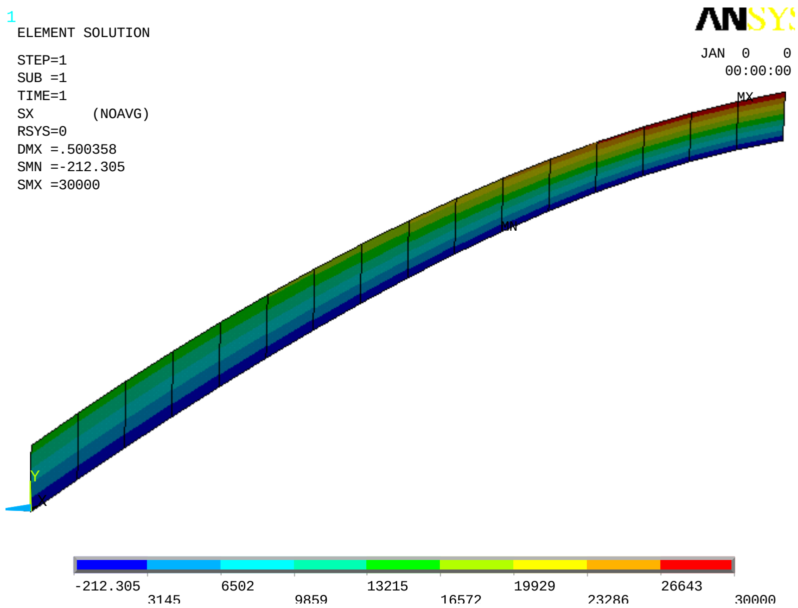


Figure 4.7: Contour for Stress for Optimal Design Variables

vii) Result Comparison

The two optimization approaches of ANSYS have possibilities to be entrapped in local minima [19]. However results show that those two optimization approaches are good for simple shape optimization of the specified problem. Present section does not compare global optimization method, EVOP with other traditional optimization methods, rather shows verification of interfacing and optimization process of EVOP. Results enlisted in Tables 4.4, 4.5 and 4.6 show that EVOP reaches the target closer than other optimization methods. Again Subproblem approximation method requires some geometric state variables to ensure left-to-right taper of the beam. These additional state variables are not used in EVOP which demonstrates robustness of the algorithm.

Table 4.4: Result Comparison by EVOP

Response	Target	EVOP	Ratio
Volume, mm^3	58993.4	59085.2	1.001
Deflection, δ_{max}, mm	12.7	12.6975	0.999
Stress, σ_{max}, MPa	206.9	206.9	1.000

Table 4.5: Result Comparison by Subproblem Approximation Method [35]

Response	Target	Subproblem Approximation Method	Ratio
Volume, mm^3	58993.4	59321.2	1.04
Deflection, δ_{max}, mm	12.7	12.675	0.998
Stress, σ_{max}, MPa	206.9	205.1	0.991

Table 4.6: Result Comparison by First Order Method [35]

Response	Target	First order Method	Ratio
Volume, mm^3	58993.4	59157.3	1.003
Deflection, δ_{max}, mm	12.7	12.7	1.00
Stress, σ_{max}, MPa	206.9	205.29	0.992

4.4.2 Test Case 2 for Structural Shape Optimization

A bracket as shown in Figure 4.8 is made of 7075-T651 aluminum (E = 71018 MPa, υ= 0.33). An initial geometry of the bracket is given in the figure (dimensions in mm). The bracket is clamped at the left hub and carries a downward load at the right hub. The load is modeled as a uniform pressure p = 50 N/mm^2 as shown. The bracket has to be designed so that it attains the minimum weight while the allowable stress is assumed to be, $F_y/1.5$ where the yield strength of the material is assumed to be F_y = 524 MPa. Here, R1, R2, R3, R4 and W are used as the design variables. Based on some physical constraints, R1 is limited to be no greater than 45 mm.

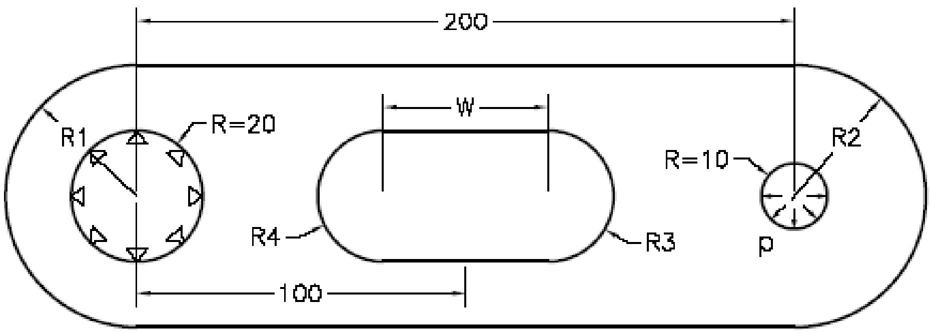


Figure 4.8: Bracket Problem Sketch

i) Problem Formulation

The problem is formulated similarly as in article 3.2.

ii) Objective Function

The objective of the model is to minimize the volume of the bracket subject to a downward load at the right hub.

iii) Design Variables and Constant Parameters

Design variables and constant parameters for the problem are shown in Tables 4.7 and 4.8 respectively. Solid Model showing keypoints and design variables is depicted in Figure 4.9. In the figure line connecting keypoints K_1 and K_3 and the line connecting K_2 and K_4 are tangential to the circle of left and right hub.

Table 4.7: Design Variables with Explicit Constraints for Test Case 2

Design Variables	**Explicit Constraint**	
	Lower bound	**Upper Bound**
R_1	25 mm	45 mm
R_2	15 mm	45 mm
R_3	5 mm	45 mm
R_4	5 mm	45 mm
W	5 mm	70 mm

Table 4.8: Design Constant Parameters for Test Case 2

	Design Constant Parameters	**Value**
Material Properties	Modulus of Elasticity, E	71018 MPa
	Poisson's Ratio, υ	0.33
Geometric Properties	L	200 mm
	Radius of left hub	20 mm
	Radius of rightt hub	10 mm
Loading	p	50 MPa
	F.S.	1.5

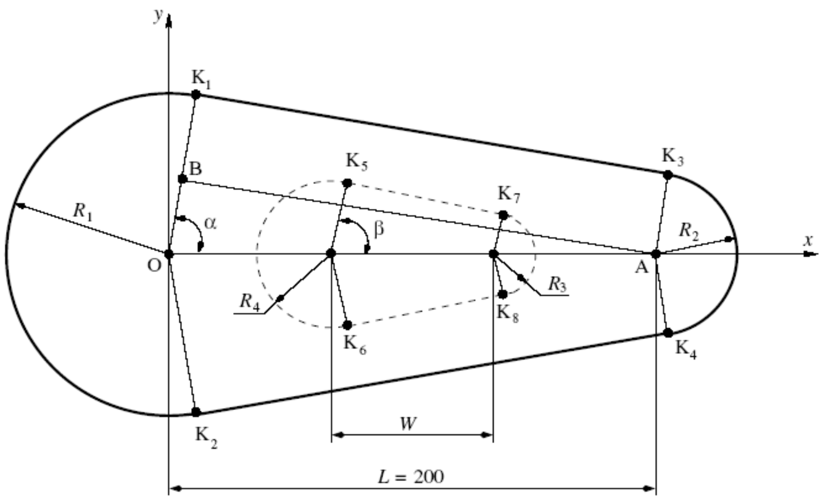


Figure 4.9: Solid Model, Showing Keypoints and Design Variable

iv) Implicit Constraints

According to the problem statement upper bound of the stress anywhere of the beam is taken as $F_y/F.S$ where F_y is the yield strength of the material taken as 524 MPa and F. S is the factor of safety taken as 1.5. Again seven geometric state variables are defined based on physical constraint. State variables are:

$$S_1 = R_1 - R_4 \tag{4.1}$$

$$S_2 = R_2 - R_3 \tag{4.2}$$

$$S_3 = W + R_3 + R_4 \tag{4.3}$$

$$S_4 = W^2 - (R_4 - R_3)^2 \tag{4.4}$$

$$S_5 = \frac{R_1 - R_2}{L} \tag{4.5}$$

$$S_6 = \frac{R_4 - R_3}{W} \tag{4.6}$$

$$S_7 = W \tag{4.7}$$

Lower bound S_7 is taken as RR where

$$RR = Larger\ of\ R_3\ and\ R_4$$

Implicit constraints are shown in Table 4.9.

Table 4.9: Implicit Constraints for Test Case 2

Response	**Implicit Constraint**	
	Lower bound	**Upper Bound**
Von Misses Stress, σ_{max}	0 MPa	349.33 MPa
S_1	5 mm	45 mm
S_2	5 mm	45 mm
S_3	5 mm	160 mm
S_4	0 mm	W^2
S_5	-1	+1
S_6	-1	+1
S_7	RR	70

v) Finite Element Modeling and Analysis

Eight noded 2-D Structural Solid Plane82 with two degree of freedom at each node has been used in modeling the bracket. Two dimensional complete finite element model showing boundary conditions and loadings is shown in Figure 4.10 Again enlarged picture of boundary conditions and loadings are shown in Figures 4.11 and 4.12 respectively.

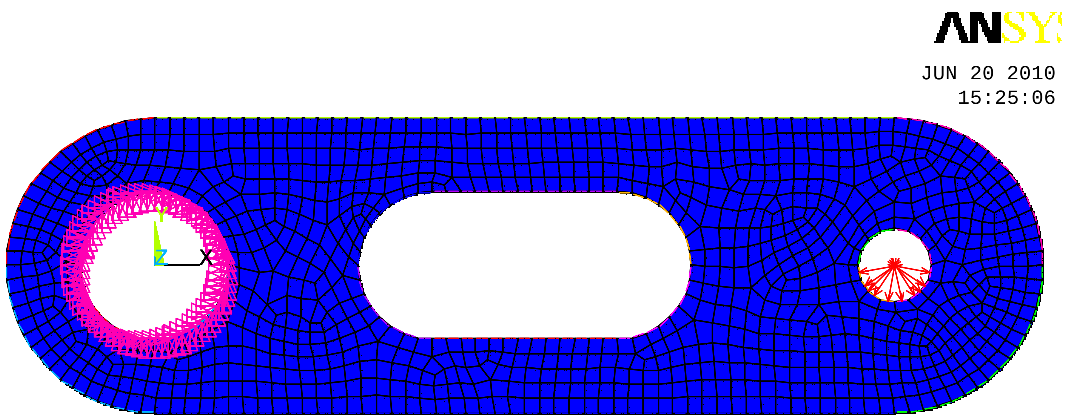


Figure 4.10: Complete Finite Element Model of Bracket

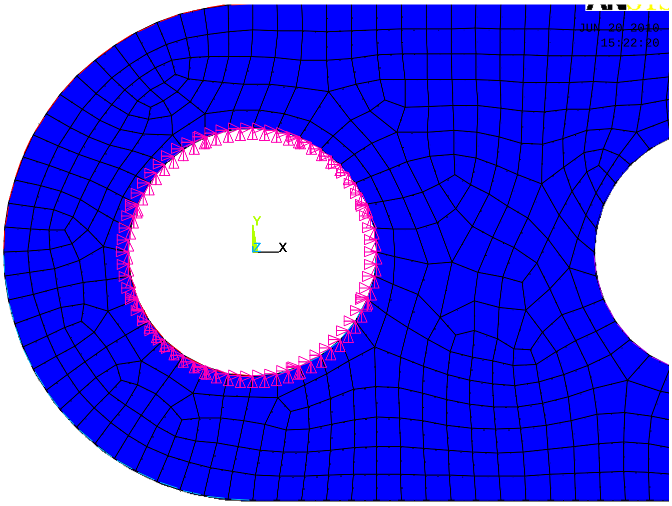


Figure 4.11: Boundary Conditions in Left Hub of Bracket (Enlarged)

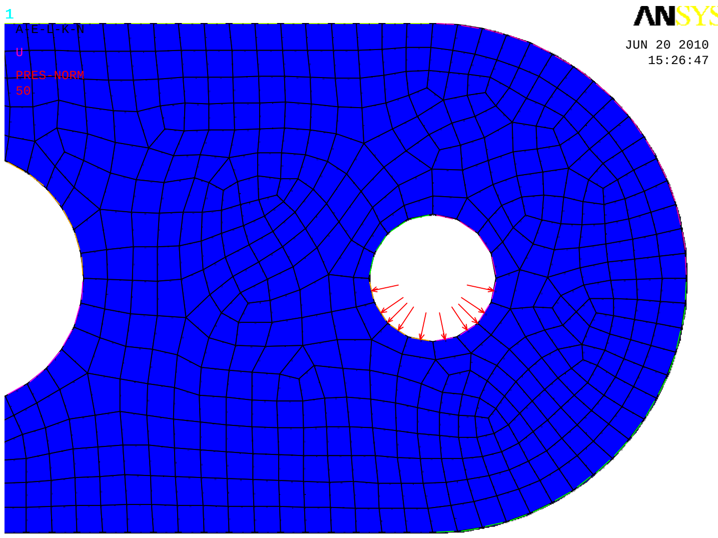


Figure 4.12: Uniform Pressure in Right Hub of Bracket (Enlarged)

vi) Optimization Process

Two separate ANSYS Parametric Design Language (APDL) scripts for implicit and objective functions are created and linked with the optimization engine EVOP. The problem is also solved independently using the traditional optimization method, the first order method of ANSYS, starting with the same initial design as in EVOP.

vii) Optimum Design of Bracket

Optimum design of the bracket is determined by an optimization engine discussed in this chapter. After completion of optimization process optimal design variables of the bracket are attained. Von Misses stress distribution for the optimal design variable is depicted in Figure 4.13 against that of the model before optimization.

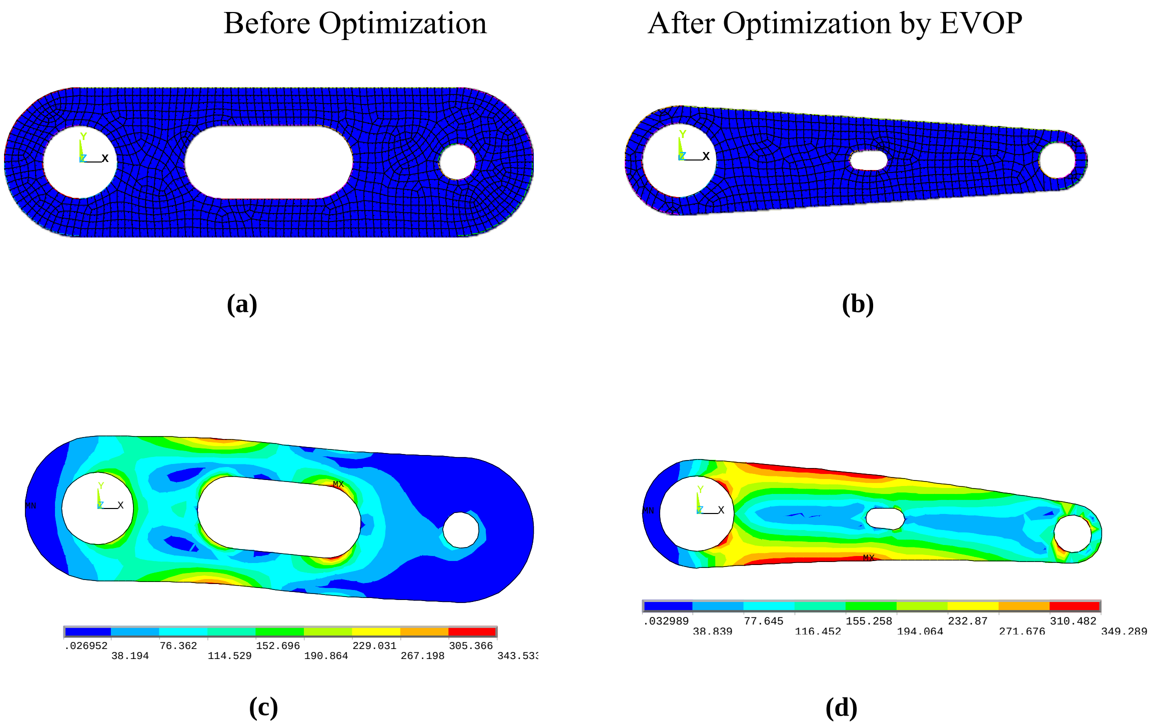


Figure 4.13: Shape Optimization by EVOP (a) Model Before Optimization (b) Model After Optimization (c) Von Misses Stress Before Optimization (d) Von Misses Stress After Optimization

viii) Result Comparison

The design optimization by EVOP shows 45.52% reduction of structure weight whereas first order method reduces 45.41%. Results extracted from two optimization approaches are compared in Tables 4.10 and 4.11. It is observed that EVOP shows better result than the first order gradient method and the specified optimization problem solved using global optimization algorithm EVOP is successful in the field of verification of interfacing platform developed in the study.

Table 4.10: Result Comparison of Objective Function for Test Case 2

	Objective Function	Initial	Final	% Reduction in Volume
Optimization by EVOP	Volume, mm^3	16199	8825.01	45.52%
Optimization by First Order Method	Volume, mm^3	16199	8843.8	45.41%

Table 4.11: Result Comparison for Test Case 2

Parameter	Before Optimization	Optimization by EVOP	Optimization by First Order Method
	Initial	**Final**	**Final**
Volume, mm^3	16199	8825.01	8843.8
Stress, σ_{max}, MPa	344.58	349.28	349.65
R_1, mm	40	28.853	28.568
R_2, mm	40	15.585	15.996
R_3, mm	20	5.628	5.5874
R_4, mm	20	5.247	5.9427
W, mm	50	9.918	8.2467

4.5 Conclusion

Application of optimization engine, developed in this study, draws successful verification of interfacing of a global optimization algorithm, EVOP with finite element software, ANSYS on benchmark optimization problems. The interfacing opens the door for simulation driven most economical design of any type of structure which requires finite element analysis.

CHAPTER 5

FINITE ELEMENT MODELING AND ANALYSIS OF NETWORK ARCH BRIDGES

5.1 General

This chapter describes finite element modeling and analysis of network arch bridges by using commercial finite element software, ANSYS. The chapter attempts to describe bridge modeling, vehicle modeling and validation of the finite element model with theoretical results. The outline of this chapter follows the basic modeling process by a description of structural geometry, definition of material and section properties of the components making up the structure, and description of boundary conditions and loads acting on the structure. Post processing regarding design of hanger and design of arch after finite element analysis is also outlined at the end of this chapter.

5.2 Network Arch Bridge

A tied arch is one where the reactive horizontal forces acting on the arch ribs are supplied by a tension tie at deck level by the bottom chord (either tie-rods or the deck itself) of a through or half-through arch. The tension tie is usually a steel plate girder or a steel box girder or the deck itself. The hangers used in tied arch bridges are usually vertical but truss like diagonal hangers can also be used. Diagonal hangers result in smaller deflections and a reduction in the bending moments in the arch rib and deck. The inclined hangers with multiple intersections named the tied arch as network arch which results lesser bending moments and shear forces in network arches and deck. Network arch bridge in Czech Republic of a span 41 meters is shown in Figure 5.1 which was opened to road traffic in 2004. Figure 5.2 shows alternate network arch bridge design concept of Lake Champlain bridge project of USA [37].

Major structural elements of the superstructure of network arch bridges consist of deck, hangers, arches and arch bracings shown in Figure 5.3.

Figure 5.1: Network Arch Bridge in Czech Republic Opened to Road Traffic in 2004 [8]

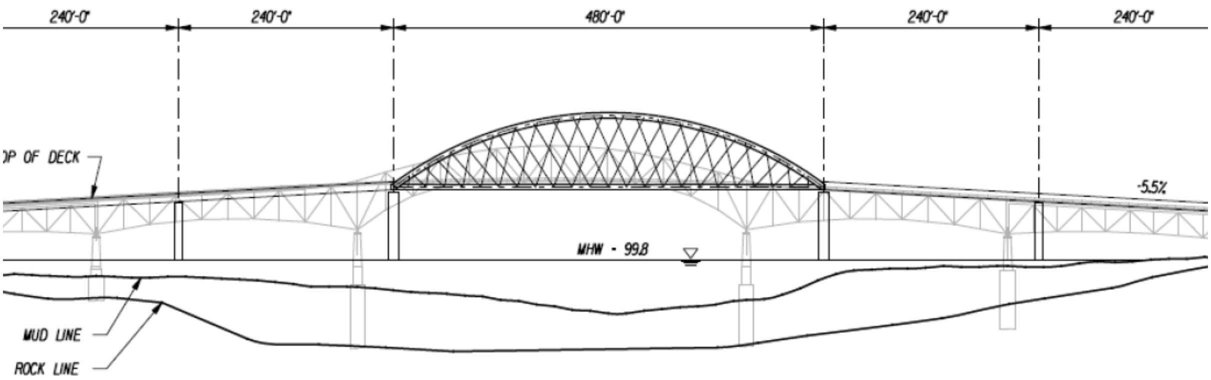


Figure 5.2: Alternate Network Arch Bridge Design Concept of Lake Champlain Bridge Project of USA [37]

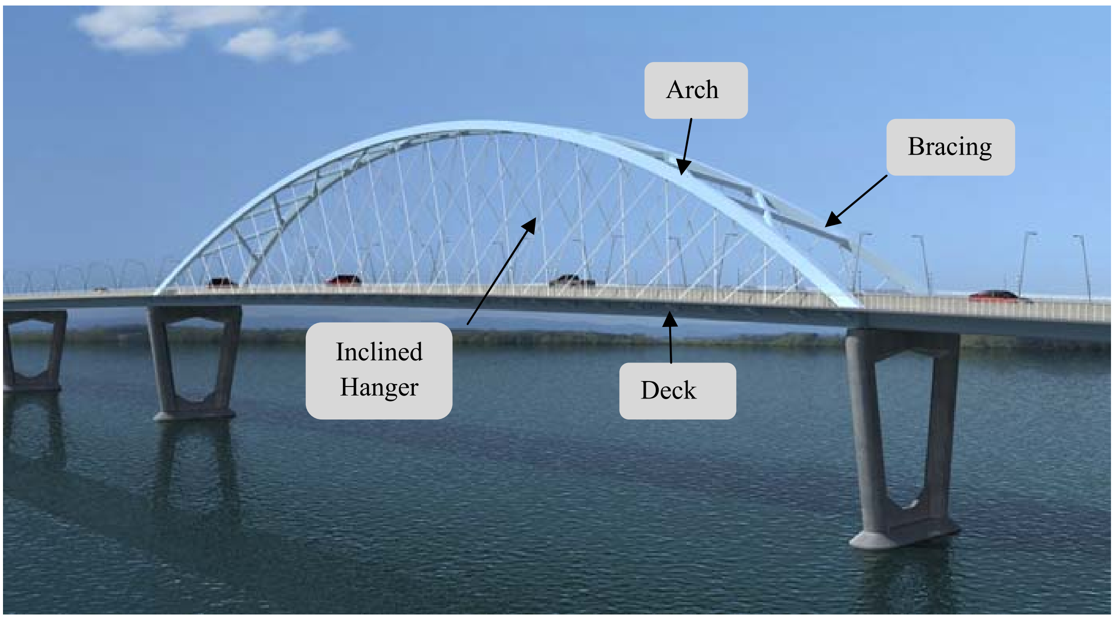


Figure 5.3: Structural Element of Network Arch Bridge

5.3 Finite Element Model of Network Arch Bridges

The complete 3D finite element model of the bridge has been introduced by assigning appropriate element types and geometries, meshing, assigning proper boundary and load conditions. ANSYS is chosen for finite element simulation of the arch bridge. Beam, link and solid elements have been used for the simulation of arch, hanger and deck of the bridge respectively. Surface element is incorporated on deck nodes for 3D structural surface effect. A load family is assigned for the highway load by the vehicle type AASHTO standard HS 20-44 truck and lane load. The basic finite element modeling, analysis and design steps are as shown in Figure 5.4.

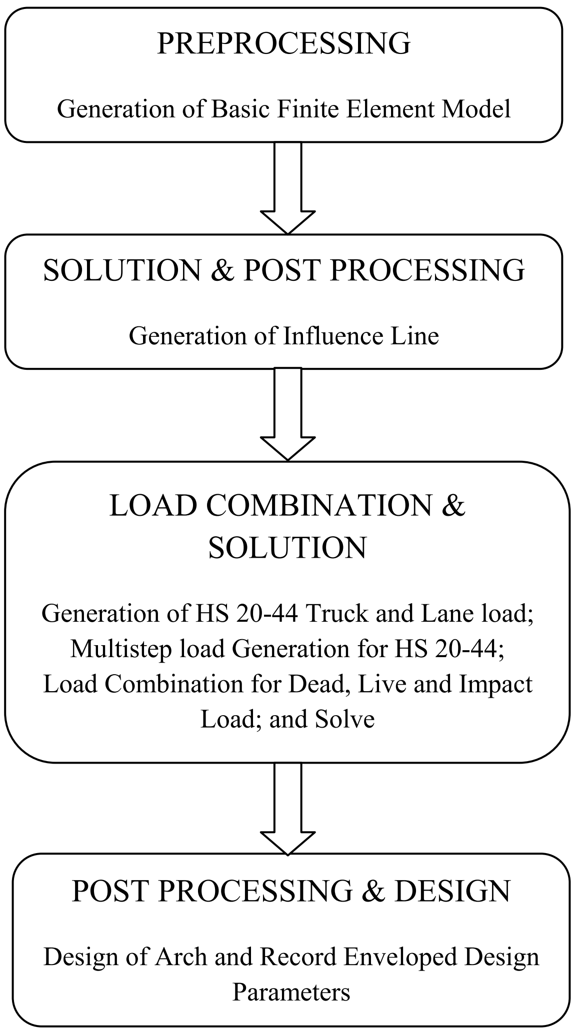


Figure 5.4: Finite Element Modeling, Analysis and Design Steps of Network Arch Bridge

The complete finite element model of the bridge is shown in Figure 5.5 and basis of modeling are described in following articles.

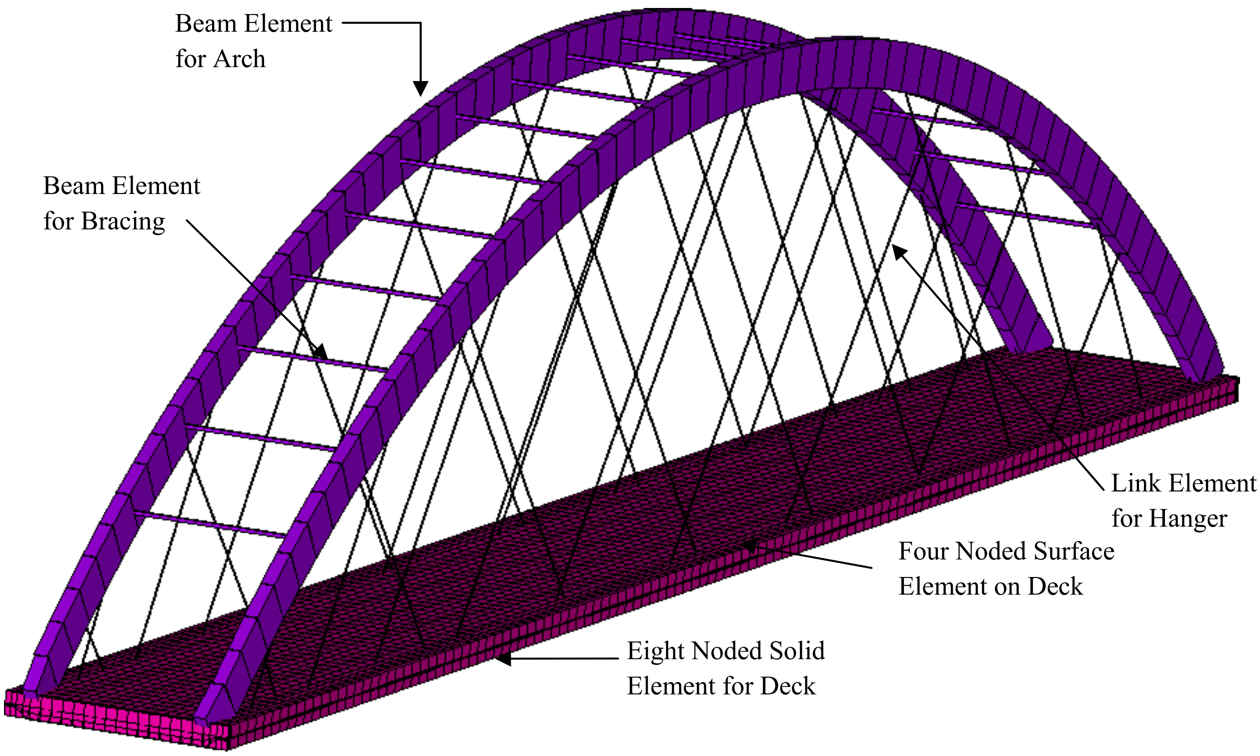


Figure 5.5: Typical Finite Element Model of Network arch Bridges

5.3.1 ELEMENT TYPES

Element types of ANSYS [21] used for modeling the structural elements of the bridge are Link10 for modeling tension only cable, Beam4 for modeling arch and bracing, Solid45 for modeling deck and Surf154 for generation of surface effects on deck. Properties of element types and their capabilities are described below.

i) LINK10 - Tension-only or Compression-only Spar

LINK10 is a 3-D spar element having the unique feature of a bilinear stiffness matrix resulting in a uniaxial tension-only (or compression-only) element. With the tension-only option, the stiffness is removed if the element goes into compression (simulating a slack cable or slack chain condition). This feature is useful for static guy-wire applications where the entire guy wire is modeled with one element. It may also be used in dynamic analyses (with inertia or damping effects) where slack element capability is desired but the motion of the slack elements is not of primary interest.

LINK10 has three degrees of freedom at each node: translations in the nodal x, y, and z directions. No bending stiffness is included in either the tension-only (cable) option or the compression-only (gap) option but may be added by superimposing a beam element with very small area on each LINK10 element. Stress stiffening and large deflection capabilities are available.

ii) **BEAM4** 3-D Elastic Beam

BEAM4 is a uniaxial element with tension, compression, torsion, and bending capabilities. The element has six degrees of freedom at each node: translations in the nodal x, y, and z directions and rotations about the nodal x, y, and z axes. Stress stiffening and large deflection capabilities are included.

iii) **SURF154** 3-D Structural Surface Effect

SURF154 is used for various load and surface effect applications. It may be overlaid onto an area face of any 3-D element. The element is applicable to 3-D structural analyses. Various loads and surface effects may exist simultaneously. The element is defined by four to eight nodes and the material properties.

iv) **SOLID45** 3-D Structural Solid

SOLID45 is used for the 3-D modeling of solid structures. The element is defined by eight nodes having three degrees of freedom at each node: translations in the nodal x, y, and z directions. The element has plasticity, creep, swelling, stress stiffening, large deflection, and large strain capabilities.

5.3.2 Material Properties

Arch and arch bracing are considered to be made of reinforced concrete, hanger is made of zinc-coated steel structural strand of ASTM A586 [29] standard and deck is of reinforced concrete. In the finite element modeling deck is introduced for vehicle movement and to transfer load to arch through hanger. Material properties of reinforced concrete and cable are shown in Table 5.1.

Table 5.1: Material Properties

Material	Parameters	Value
Concrete	Modulus of Elasticity for Concrete, E_c	24800 MPa
	Poisson's Ratio, υ	0.2
	Concrete Compressive Strength at 28 days (Arch and Bracing)	25 MPa
	Concrete Compressive Strength at 28 days (Deck)	50 MPa
	Unit Weight, γ	24.5 N/m^3
Reinforcement	Arch Reinforcement Yield Stress, f_y	413 MPa
	Modulus of Elasticity, E_s	200 X 10^3 MPa
	Poisson's Ratio, υ	0.3
	Density, ρ	7833 kg/m^3
Cable of Hanger	Ultimate Strength of Cable of Hanger	1520 MPa
	Modulus of Elasticity for Concrete, E_{sc}	195 X 10^3 MPa
	Poisson's Ratio, υ	0.3
	Density, ρ	8000 kg/m^3
Surface Elements	Density, ρ	0.00 kg/m^3

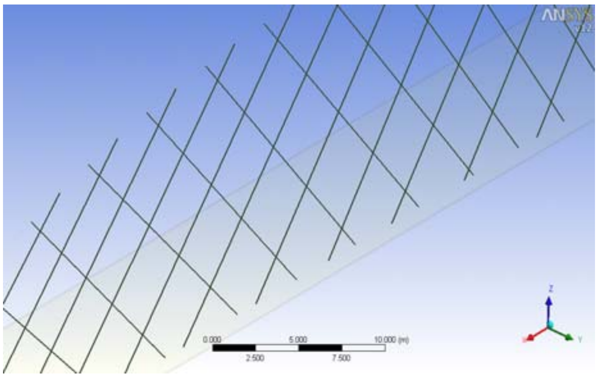
Linear material model is taken for analysis of concrete, reinforcement and cable in tension. Material model for cable in compression is removed by using nonlinear convergence check. For design of reinforcement and cable, bilinear material is used and material model for design of concrete is considered nonlinear.

5.3.3 Geometrical Parameters

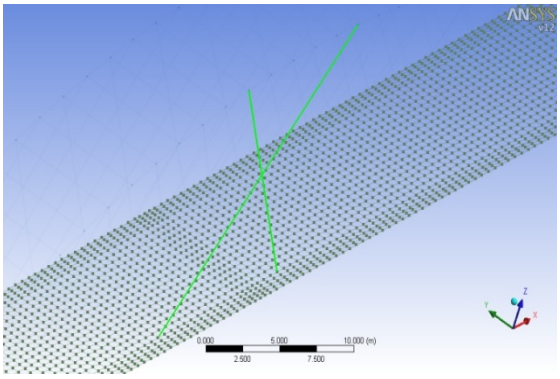
A two lane 100 m roadway bridge comprised of arches, deck, hangers and bracings is modeled for FE analysis. Circular and parabolic arch geometry of rectangular section of arch within a range of sectional dimensions specified in Table 3.1 is considered in the analysis. Rise to span ratio of arch, number of hangers, hanger arrangement and hanger sectional dimensions of ASTM A586 standard specified in Table 3.1 is used in the analysis. Optimization process dynamically allocates any of them intelligently to seek optimum result.

5.3.4 Finite Element Meshing and Node Connectivity

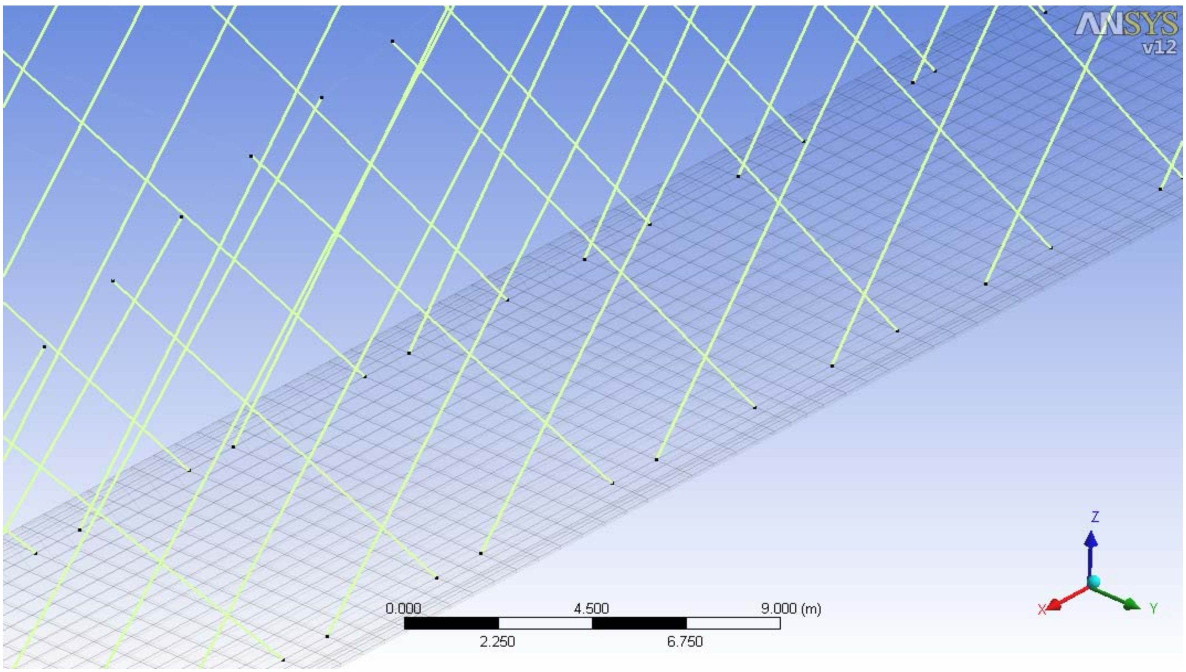
Eight nodded brick element Solid45 is used to model the deck. Deck is divided into elements in such a way that each mesh size of the deck element is 0.5 meter. Surf154 element of road surface is used on deck top surface and mesh size of surface element is same as that of deck. Beam4 is is used for modeling arch. A series of equidistant nodes are created in the arch geometry to model the arch. Arch is modeled by generating beam elements in between nodes. Again equidistant nodes along the arch and nodes of deck in the same vertical plane of arch are used to generate hangers by connecting those two sets of node through link10 elements. Figure 5.6 shows hanger node connectivity of the bridge along with deck mesh sizes and Figure 5.7 shows vierendeel type bracing that is used in FE model.



(a) In Plan View of Hangers



(b) Nodes on the Deck and Two Set of Hangers



(c) 3-D View of Generation of Hangers by Connecting Nodes on the Arch and the Deck

Figure 5.6: Hanger Node Connectivity along with Meshing in Deck

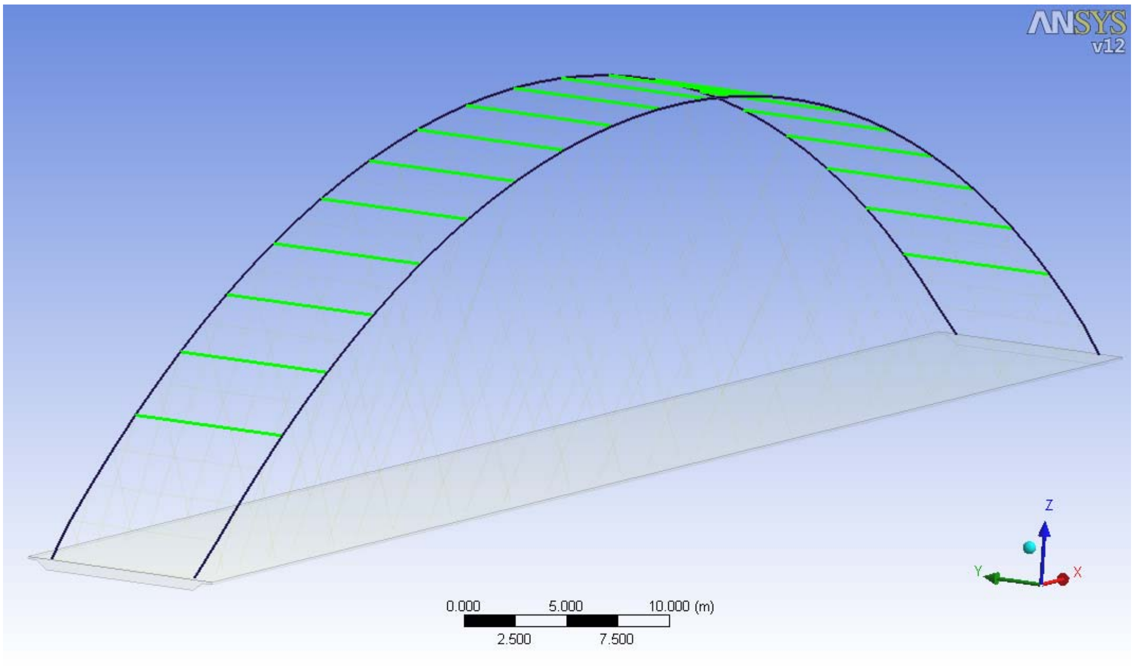


Figure 5.7: Bracing between Two Ribs

5.3.5 Boundary Conditions

Tied arch bridges is generally simply supported. Support in the deck is implied to control the degree of freedom on lower face end nodes of solid elements of concrete deck. Three rotational degrees of freedom are released in one lower face end nodes of solid elements. In another lower face end nodes of solid elements, three rotational and two horizontal degrees of freedom are released. Boundary conditions in the finite element model are shown in Figures 5.8 and 5.9.

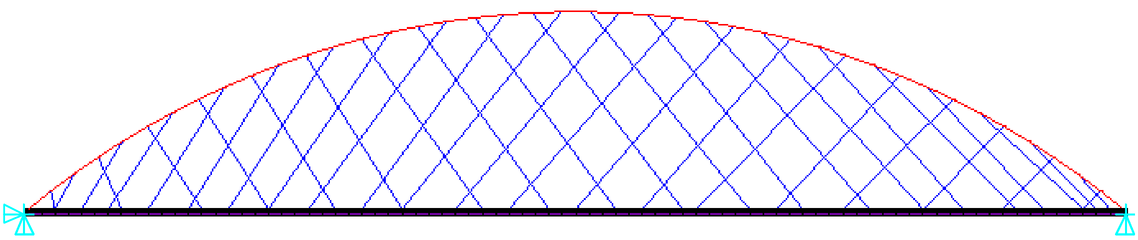


Figure 5.8: Boundary Condition in FE Model

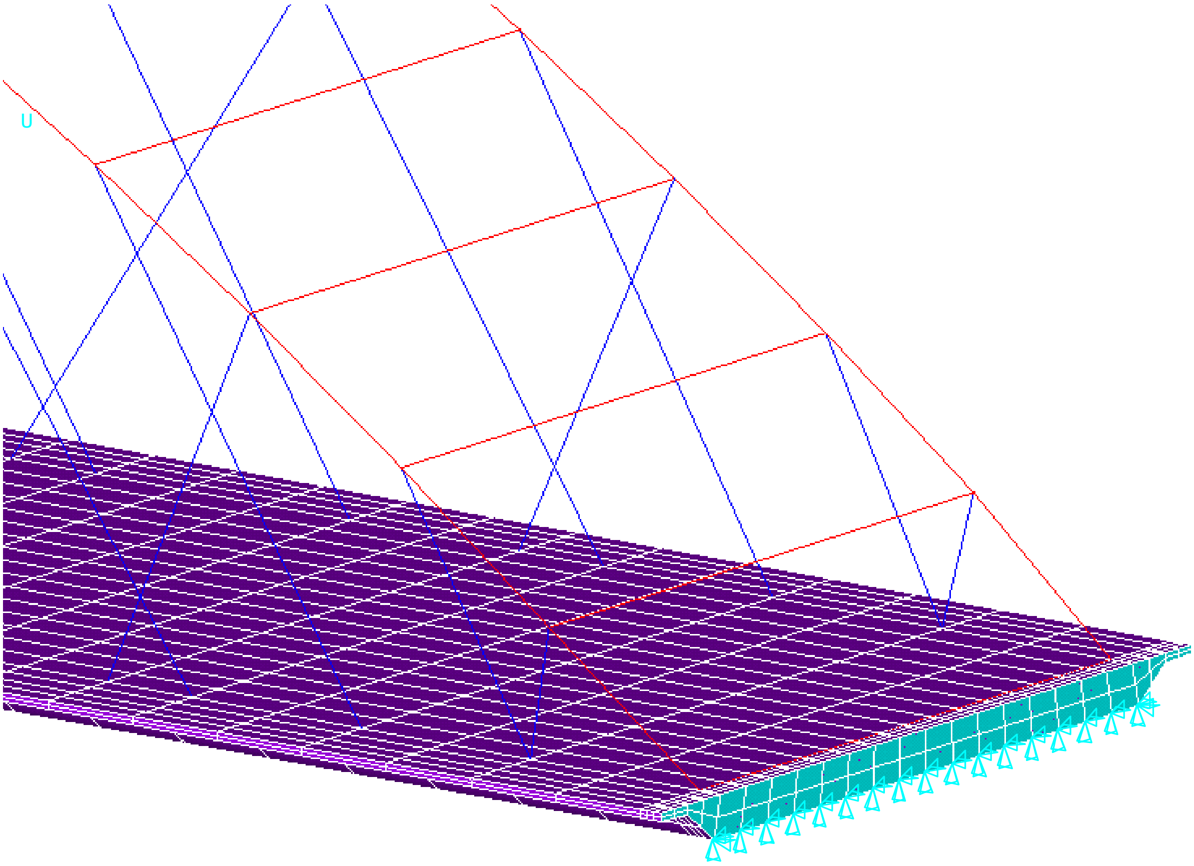


Figure 5.9: Support in Deck of FE Model

5.4 Vehicle Modeling

Vehicle is modeled as external forces on deck elements. External forces are simulated from AASHTO Standard Vehicle. Constant velocity is considered for vehicle movement and external forces are applied in deck elements forward in a step by step.

5.4.1 Standard Vehicle

The highway truck loading considered in this study is HS20-44 highway standard design truck and lane loading of AASHTO [26] shown in Figures 5.10, 5.11 and 5.13. Simplified AASHTO truck loading configuration is described in Figure 6.12.

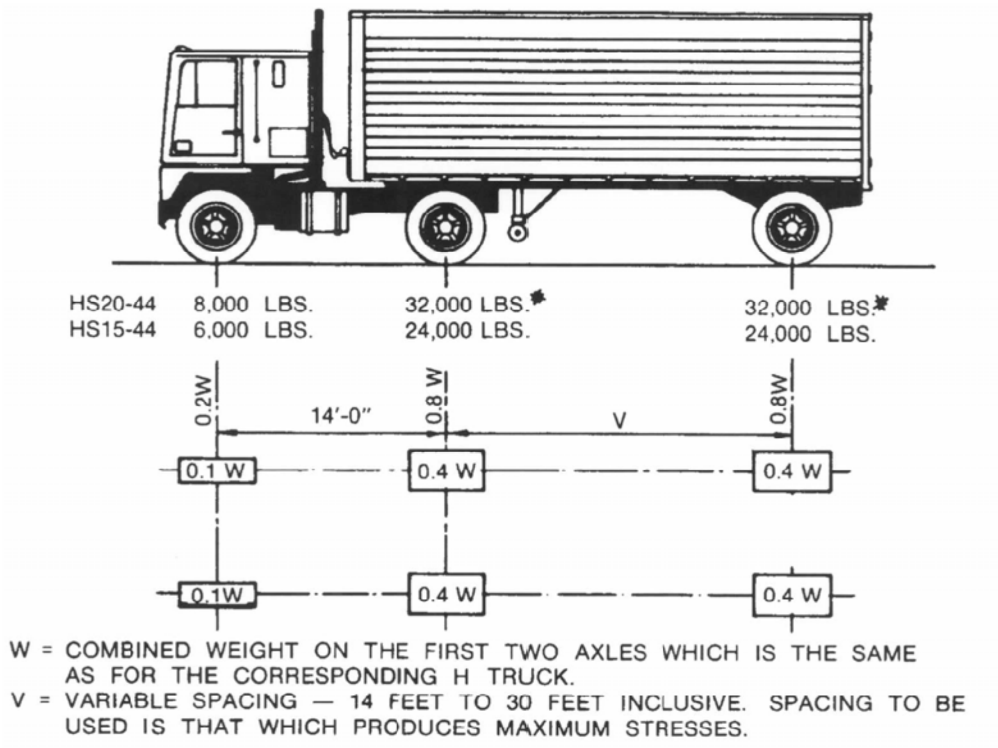


Figure 5.10: Axle Load and Wheel Load of AASHTO HS Standard Truck [26]

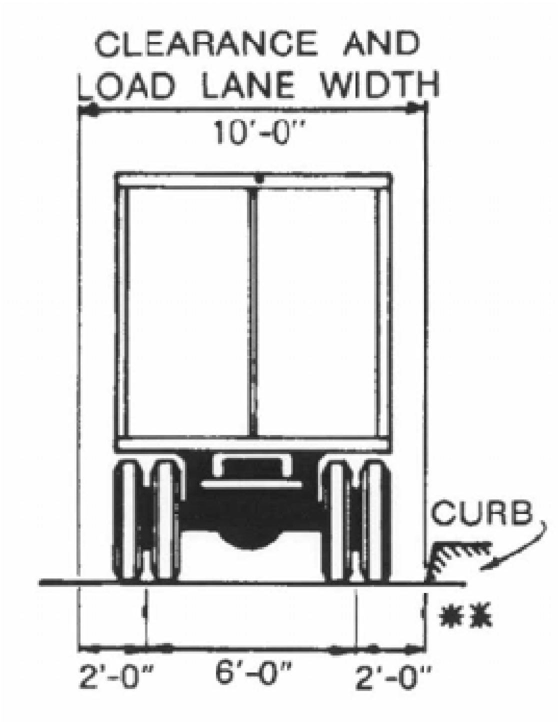


Figure 5.11: Clearances and Load Lane Width of AASHTO HS 20-44 Standard Truck [26]

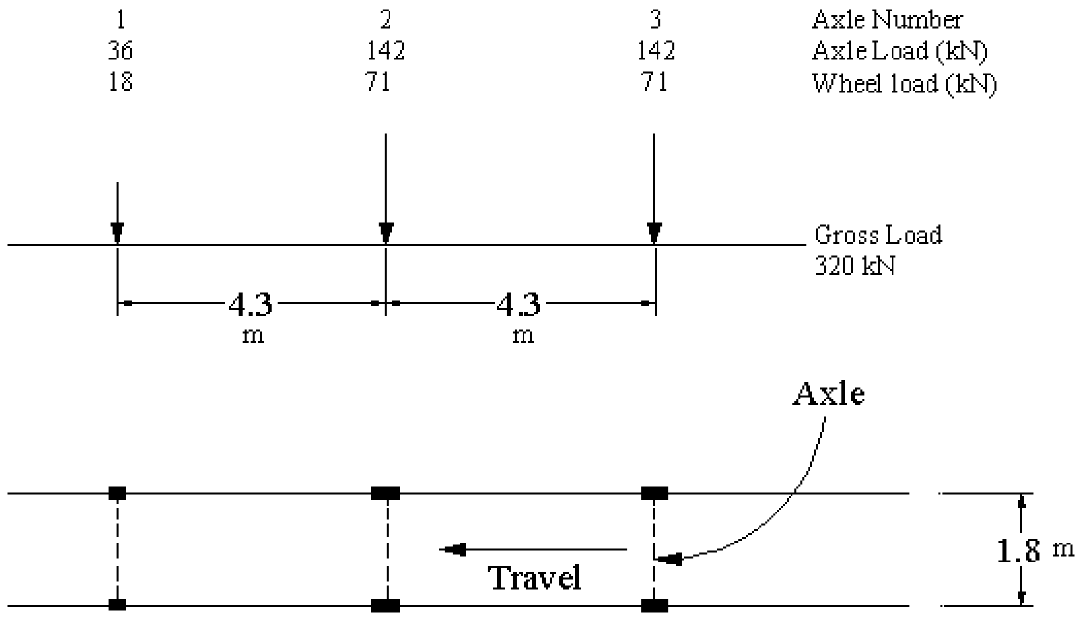


Figure 5.12: Simplified Truck Loading Configuration of AASHTO HS 20-44

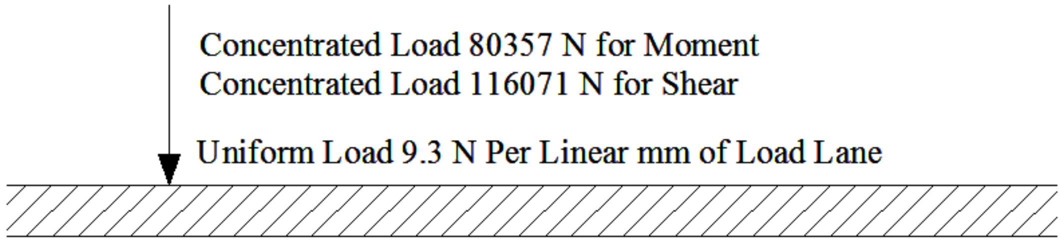


Figure 5.13: AASHTO HS 20-44 Lane Loading [26]

5.5 Loading

The vehicle truck load is idealized as pairs of concentrated forces moving along the deck in two paths parallel to the centerline of the bridge lanes at different wheel positions. Design lane load is applied on deck as uniform pressure in each bridge lane in specific region. Concentrated load of HS 20-44 lane loading is also applied at the region for maximum effect of moment. The vehicle is assumed to move with constant velocity. Finite element modeling of the bridge including vehicle truck and lane loads is shown in Figure 5.14.

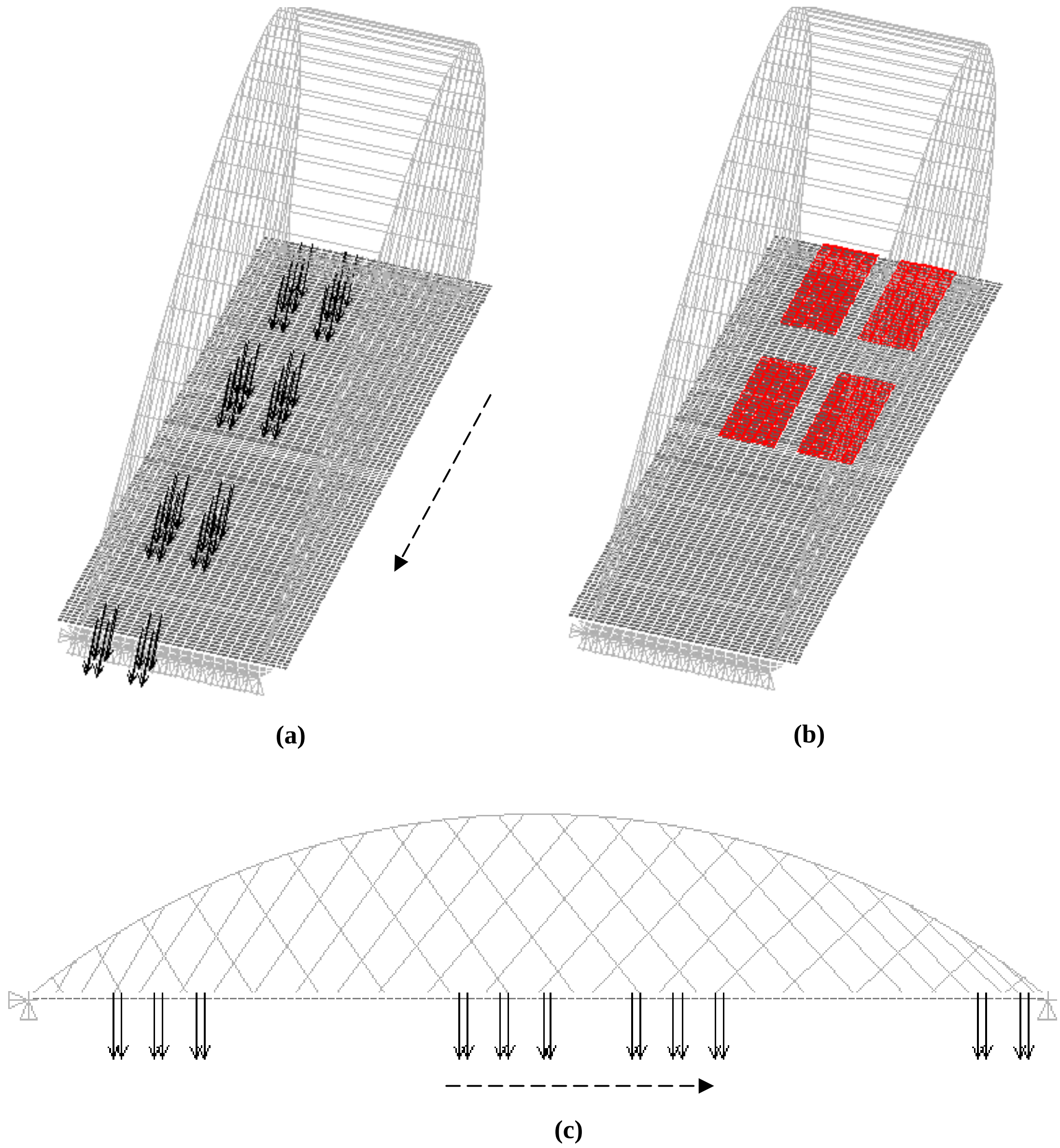


Figure 5.14: Various Live Load Positions (a) Single Truck Load in a Lane (b) Lane Load Positions (c) Elevation of Network Arch Bridge Showing Various Truck Positions

5.5.1 Influence Line Generation

Influence lines of moments have been generated to develop the positive and negative moment zones in the arch. It is helpful to determine maximum positive and negative moments in arch for design lane loading.

5.5.2 Multistep Load Generation

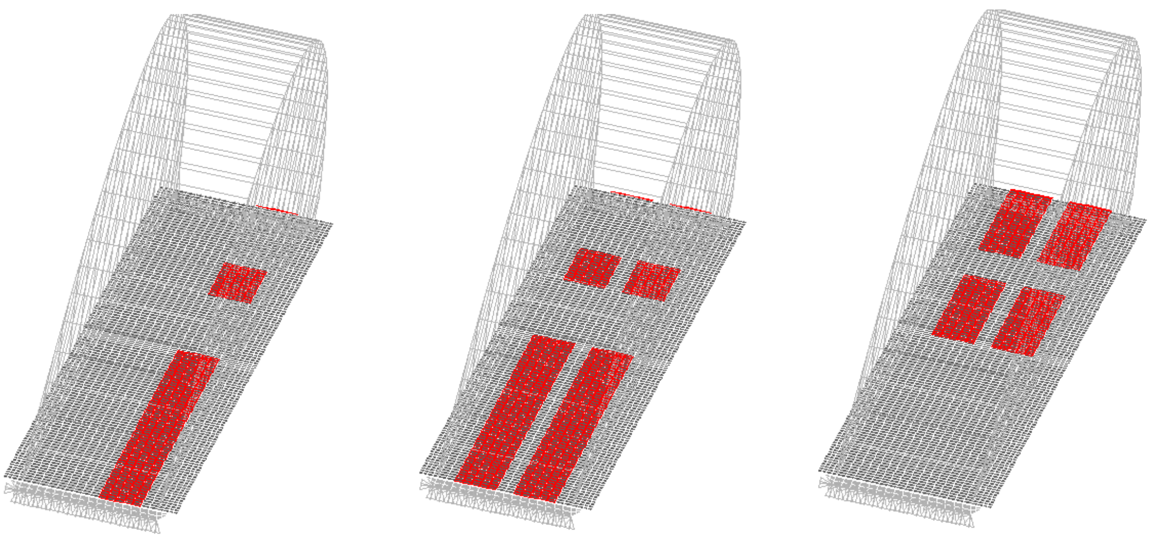
A single moving load is generated in two lanes of bridge to produce maximum stresses in bridge elements. Therefore depending upon the load positions multistep load for single truck and lane load has been generated. Tables 5.2 and 5.3 show multistep truck and lane load load generation on deck respectively. Ten numbers of single vehicle positions are considered in each lane. When truck loads are in two lanes, permutation of single vehicle positions of both lanes is required for determining maximum effect on structural elements. That permutation is number 3 shown in Table 5.2. Maximum effect on structural elements due to lane load, eighteen vehicle positions are considered in each lane. When loads are in two lanes, eighteen vehicle positions for both lane load positions are considered for maximum response in the bridge. In addition multiple presence factor is taken as 1 for single and two lane vehicle positions according to AASHTO [26].

Table 5.2: Multistep Design Truck Generation

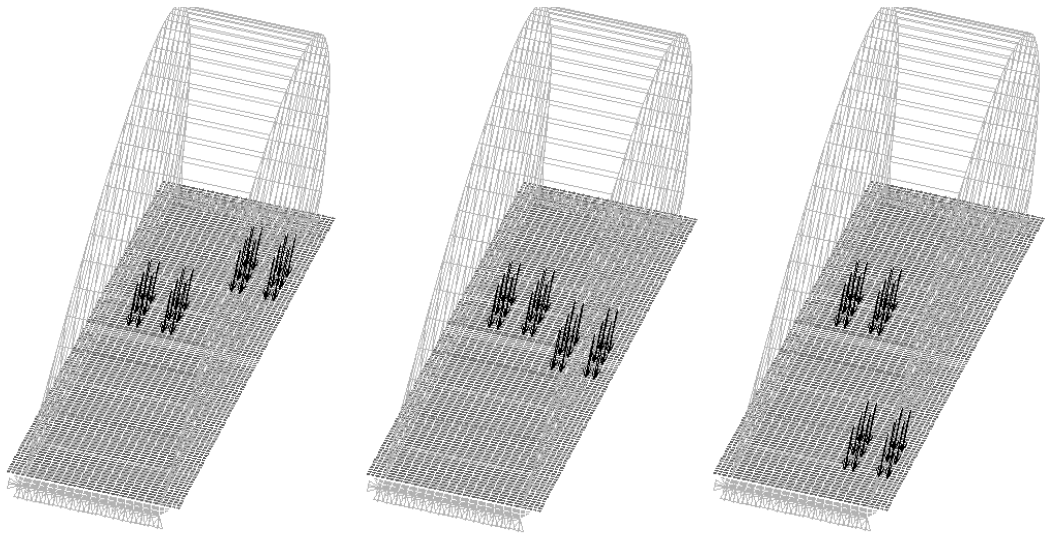
Permutation	Lane 1	Lane 2	Scale Factor for Multiple Presence	No of Vehicle Position
1	HS		1.00	10
2		HS	1.00	10
3	HS	HS	1.00	100

Table 5.3: Multistep Design Lane Generation

Permutation	Lane 1	Lane 2	Scale Factor for Multiple Presence	No of Vehicle Position
1	HL		1.00	18
2		HL	1.00	18
3	HL	HL	1.00	18



(a)



(b)

Figure 5.15: Typical Multi Load Step (a) For Lane Load (b) For Truck Load

Some multistep truck and lane loads are depicted in Figure 5.15. Plans of multi load steps for truck load are shown in Figure 5.16.

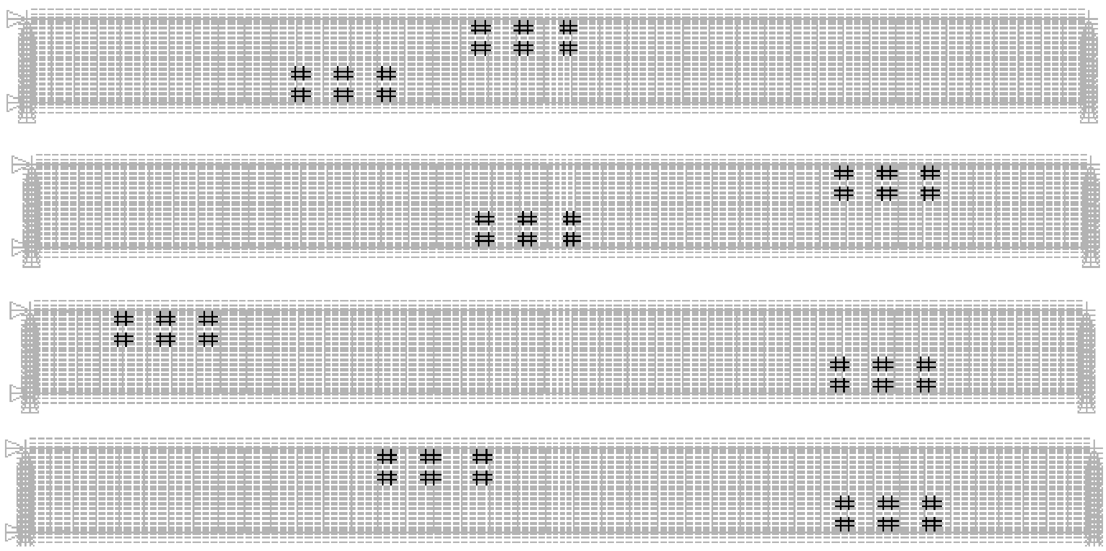


Figure 5.16: Typical Multi Load Step for Truck Load in Plan of Deck

5.6 Load Cases and Load Combinations

The bridge is designed to carry the following loads.

a) Dead load which includes self weight of the structure and wearing course
b) Live load (or the weight of applied moving load of vehicles) with impact.

Load combination for which maximum effect in the bridge determined is

$$1.3\ DL + 2.17\ (LL + I)$$

Where DL, LL and I are dead, live and impact load respectively.

5.7 Validation of Finite Element Model of Arch Bridge

Finite element model of network arch bridge with its loading conditions has been validated with theoretical results of arch analysis. Arch analysis of circular and parabolic geometry of arch is already developed in textbooks. Arch thrust and influence line at various locations are already available where analysis was carried out for arch only without hangers and decks. Following subsection will describe the validation of results from finite element analysis of modified model of network arch bridge developed in the study.

5.7.1 Validation of Finite Element Model for Parabolic Geometry of Arch

Influence line for horizontal thrust at support and bending moment in different position of two hinged parabolic arch are already established theoretically in text books. Out of them the theoretical results obtained from Coates [38] have been compared with the results obtained from simplified finite element model of the bridge developed in the study. A typical parabolic arch analyzed by Coates [38] is shown in Figure 5.17. In the figure X_p is the distance of unit load location and X_i is the distance of any position of arch where structural forces needed to determine. k and a are the factors for position of the arch. H_2 and R are the reaction at supports and h and L are rise and span respectively.

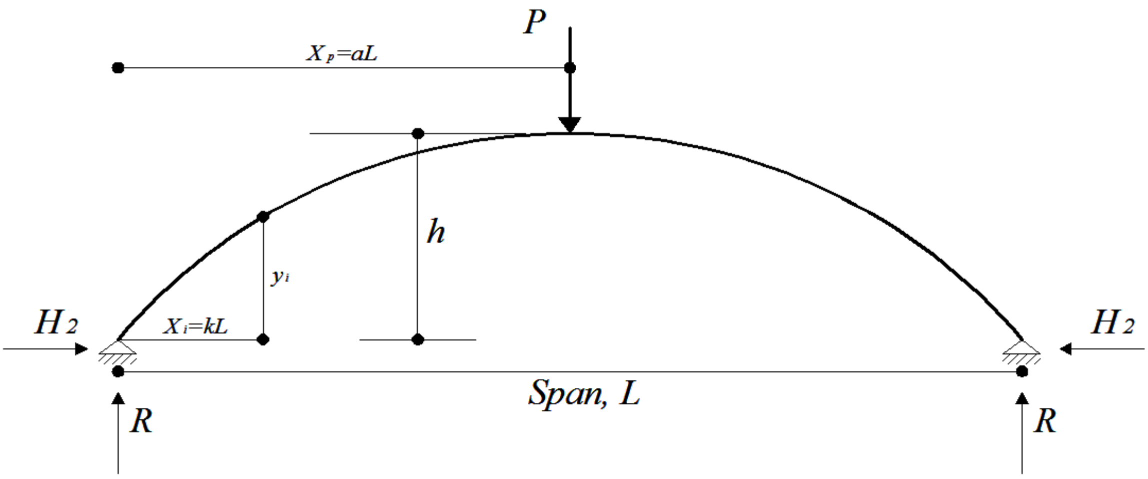


Figure 5.17: Geometry, Support and Load Conditions of Parabolic Geometry of Arch

The 3D finite element model which is developed in this study has been used for validation by modifying element properties. Vertical hanger arrangement is provided and boundary conditions are applied at two ends of arch. Deck stiffness is modified to approximately zero. Unit weight of deck concrete and cable of hanger is modified to zero. Unit load is applied at root of vertical hangers and thus the modified FE model transfer that load to arch in the same line of vertical hanger which is shown in Figure 5.18. Horizontal thrust at support for various rise span ratio for a unit load acting at crest of parabolic arch is recorded and results have been verified which is shown in Figures 5.19 and 5.20.

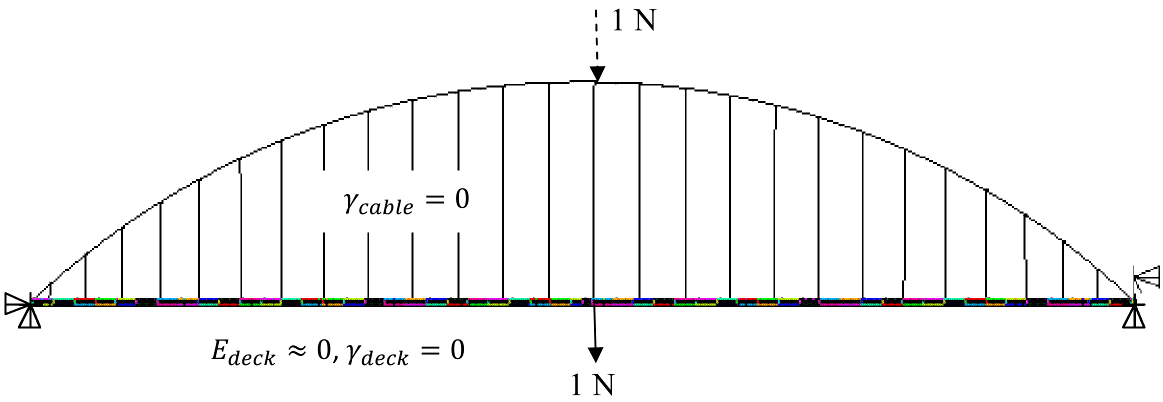


Figure 5.18: Simplification of Finite Element Model for Generating Influence Line

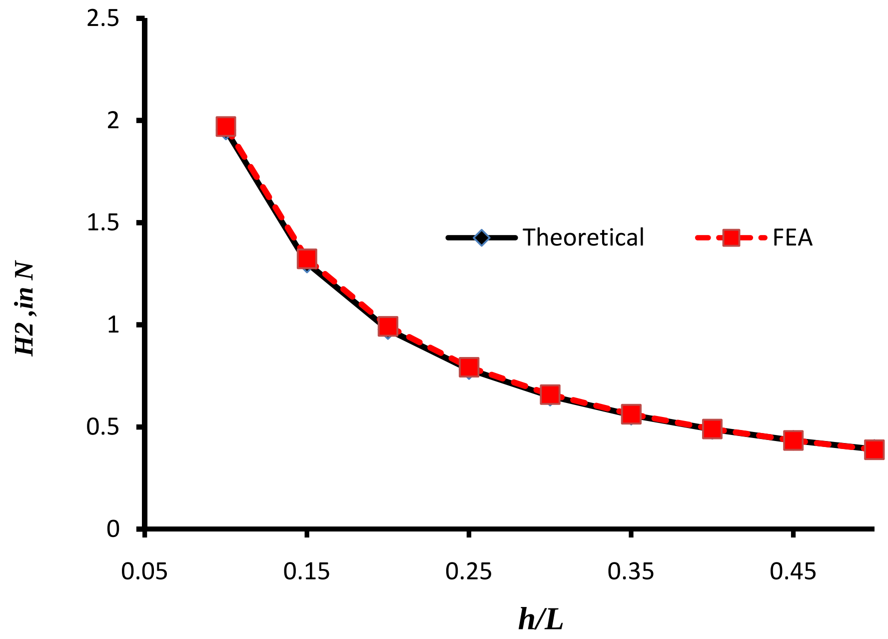


Figure 5.19: Horizontal Thrust (for Unit Load ar Crest) at Different Rise Span Ratio

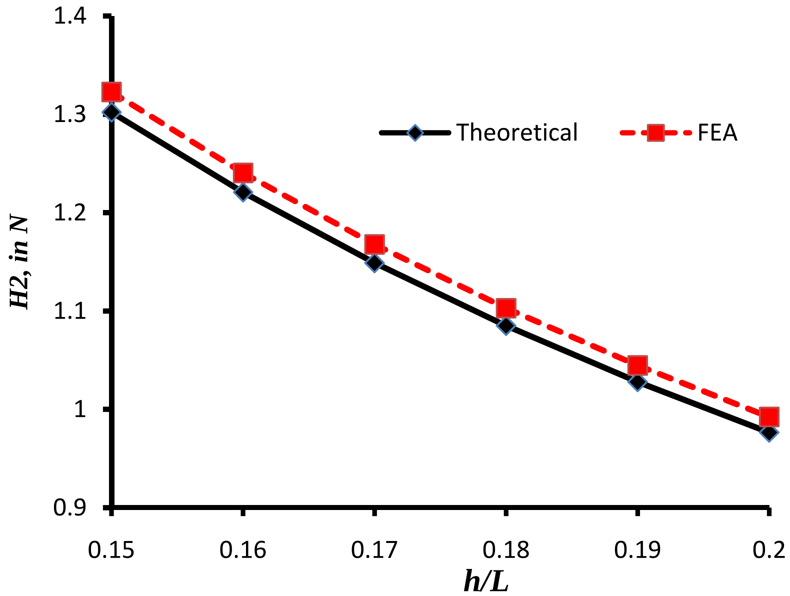


Figure 5.20: Horizontal Thrust (for Unit Load ar Crest) at Different Rise Span Ratio (from 0.15 to 0.20)

Theoretical influence line for horizontal thrust has been generated from the Equation 5.1 developed by Aswani [39] and has been plotted in Figure 5.21. By considering rise span ratio as multiplier of Equation 5.1, theoretical and FE based influence line of horizontal thrust is compared which is shown in Figure 5.22.

$$I = 5L/8h * k * (1 - 2k^2 + k^3) \tag{5.1}$$

Where, I is the influence line ordinate for thrust in arch and k, h and L are the notations described in Figure 5.17.

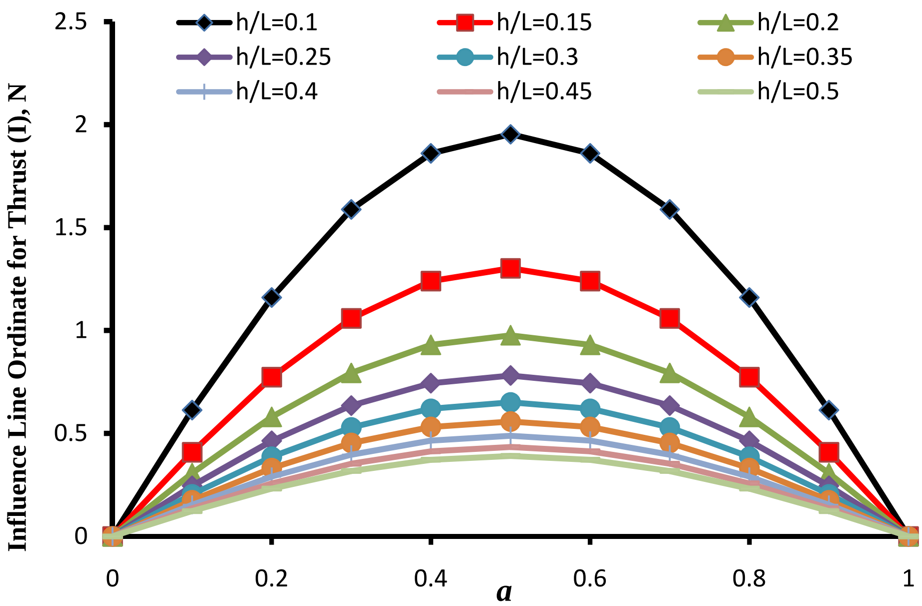


Figure 5.21: Theoretical Influence Line for Thrust for Different Rise Span Ratio

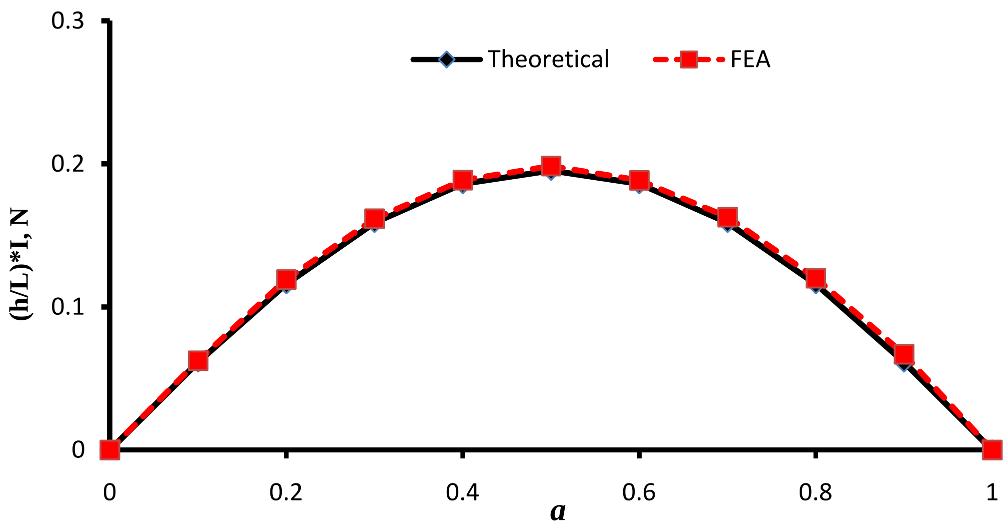


Figure 5.22: Influence Line for Thrust with Multiplier of Rise Span Ratio

Bending moments are recorded at different locations of arch after FE analysis by moving unit load along the deck and the influence line (with multiplier of rise span ratio) for bending moment is plotted and compared with theoretical results which are shown in Figures 5.23 to 5.27.

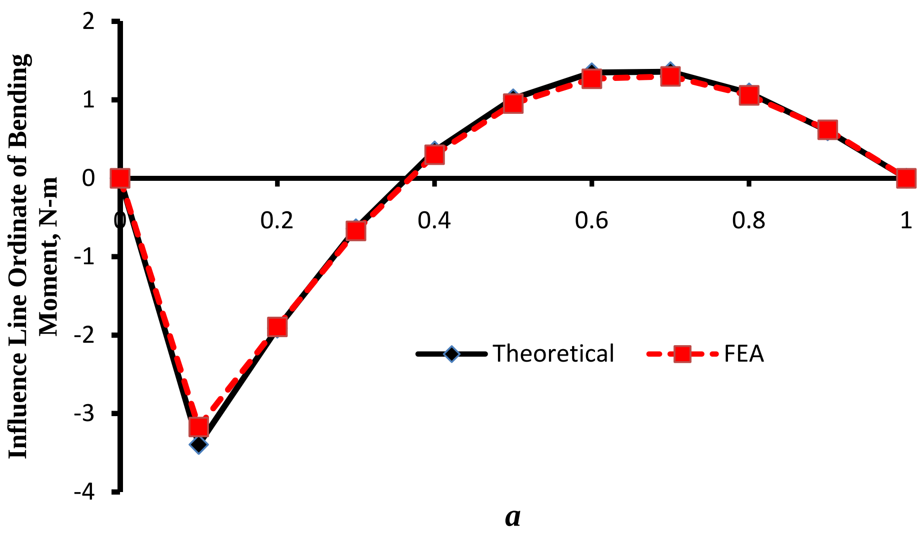


Figure 5.23: Influence Line for Bending Moment at *X = 0.1L*

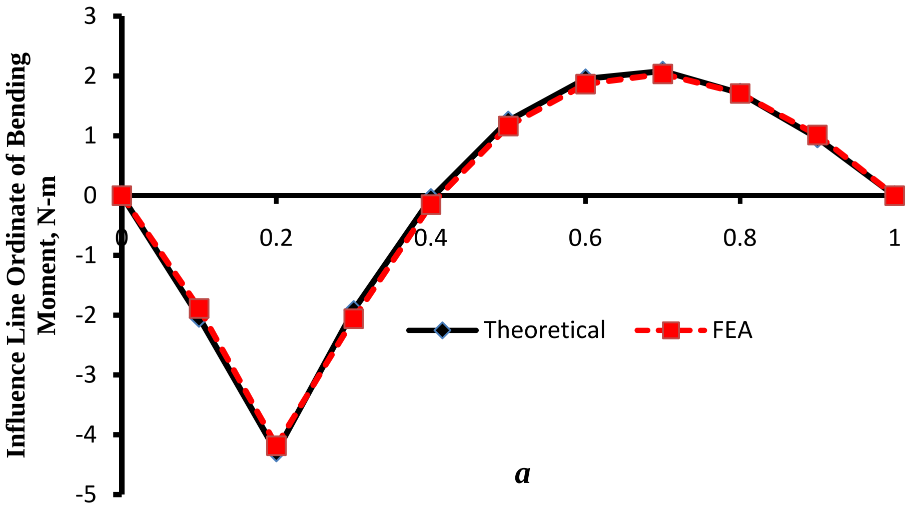


Figure 5.24: Influence Line for Bending Moment at *X = 0.2L*

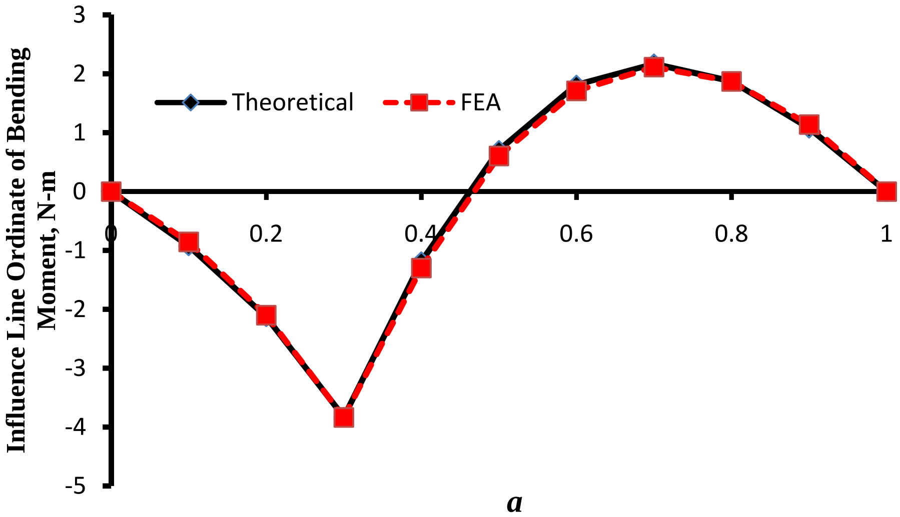


Figure 5.25: Influence Line for Bending Moment at *X = 0.3L*

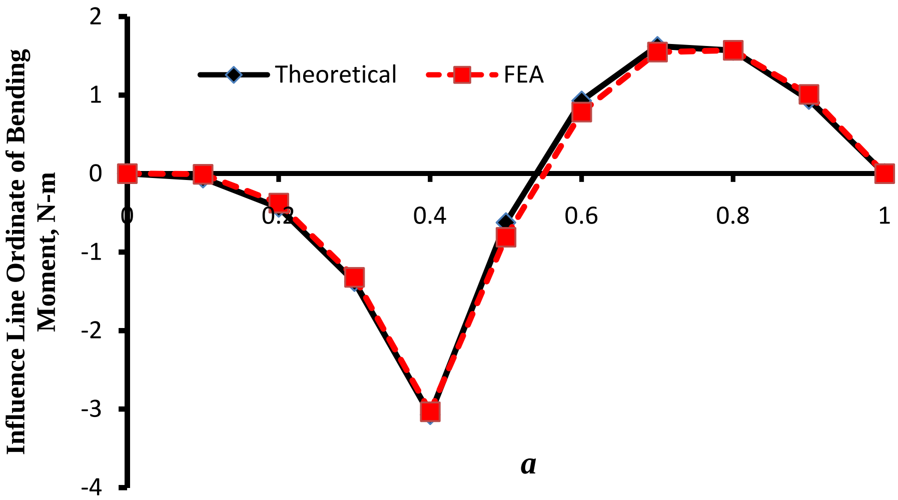


Figure 5.26: Influence Line for Bending Moment at *X = 0.4L*

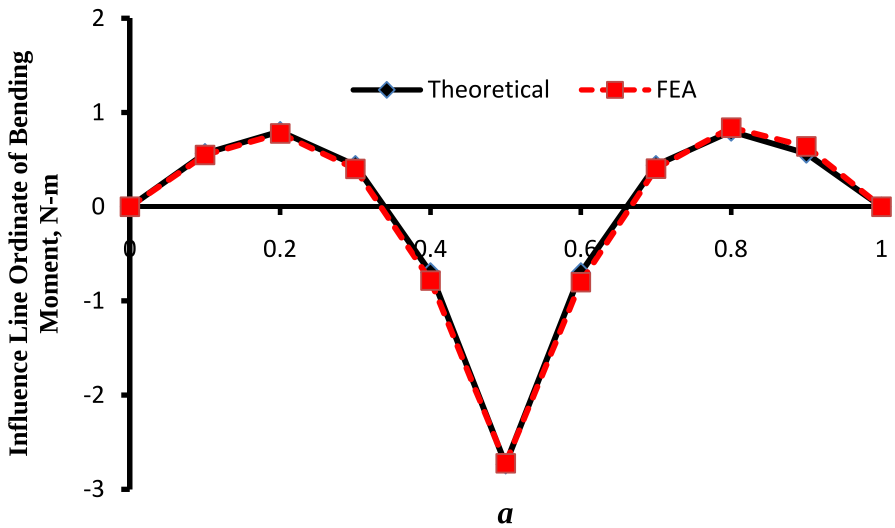


Figure 5.27: Influence Line for Bending Moment at *X = 0.5L*

Bending moment diagram and deflection for unit load at 0.2L and load at 0.5L of deck is plotted after finite element analysis which is shown in Figures 5.28 to 5.31.

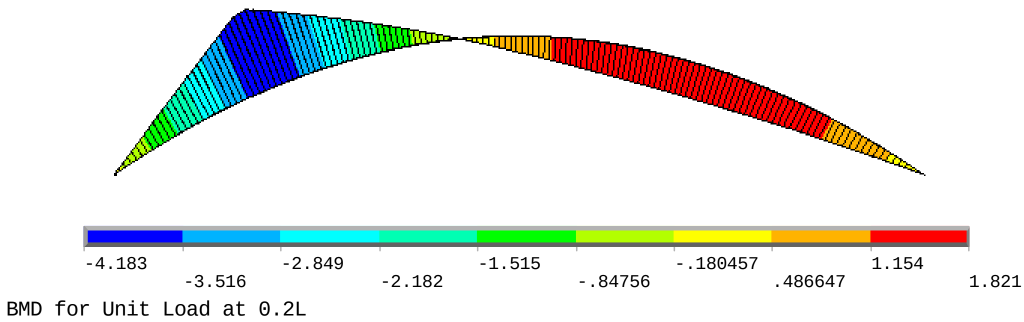


Figure 5.28: Bending Moment (N-m) Diagram for Unit Load at 0.2L of Deck

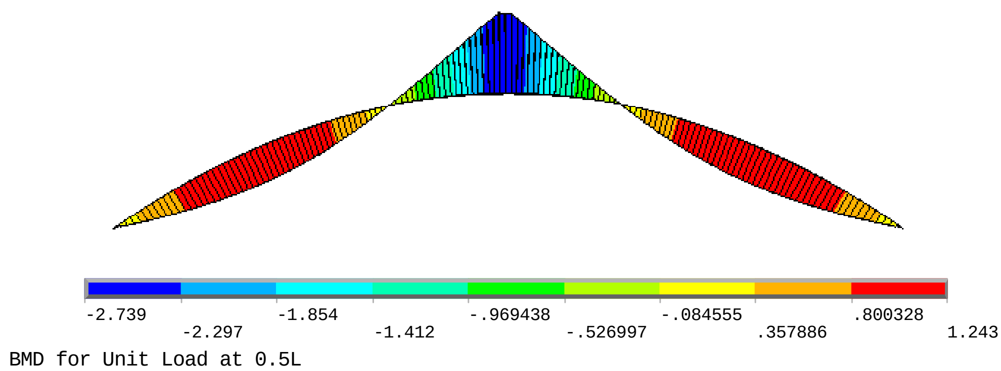


Figure 5.29: Bending Moment (N-m) Diagram for Unit Load at 0.5L of Deck

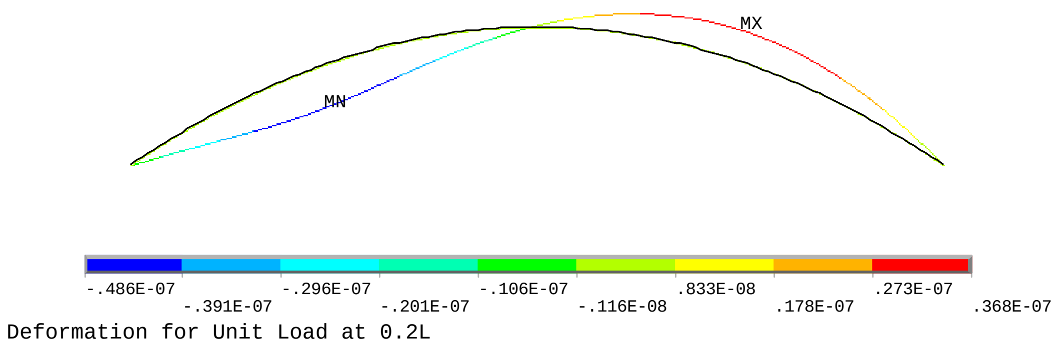


Figure 5.30: Deflection (m) of Arch for Unit Load at 0.2L of Deck

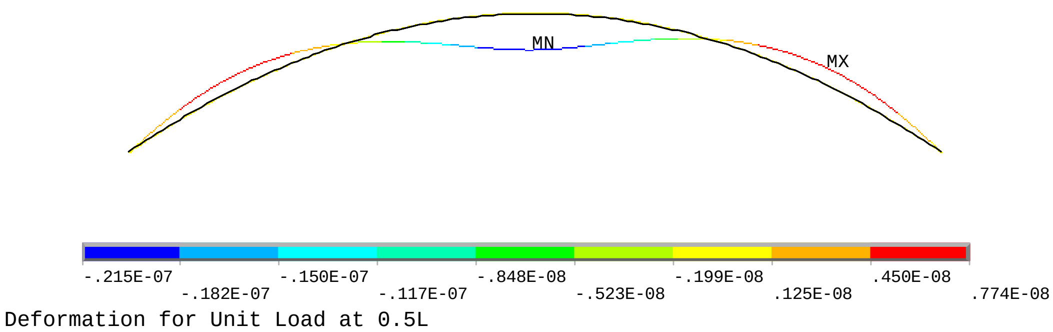


Figure 5.31: Deflection (m) of Arch for Unit Load at 0.5L of Deck

5.7.2 Validation of Finite Element Model for Circular Geometry of Arch

Horizontal thrust for circular geometry of two hinged arch for different rise span ratio has been developed theoretically by Salvadori [40] for load conditions shown in Figure 5.32. The results have been compared with simplified finite element model. Simplification of the finite element model has been done by removing link 10 (cables) and Solid 45 (deck) elements and boundary conditions are applied at two ends of arch. Analysis was done for self weight of arch. When rise span ratio equals to zero then total vertical reaction, R_1 is recorded and for different rise span ratio unit weight of arch concrete section is scaled in such a way that total reaction, R for each rise span ratio becomes equal to R_1. Then horizontal thrust recorded at support is compared with theoretical results which are shown in Figure 5.33.

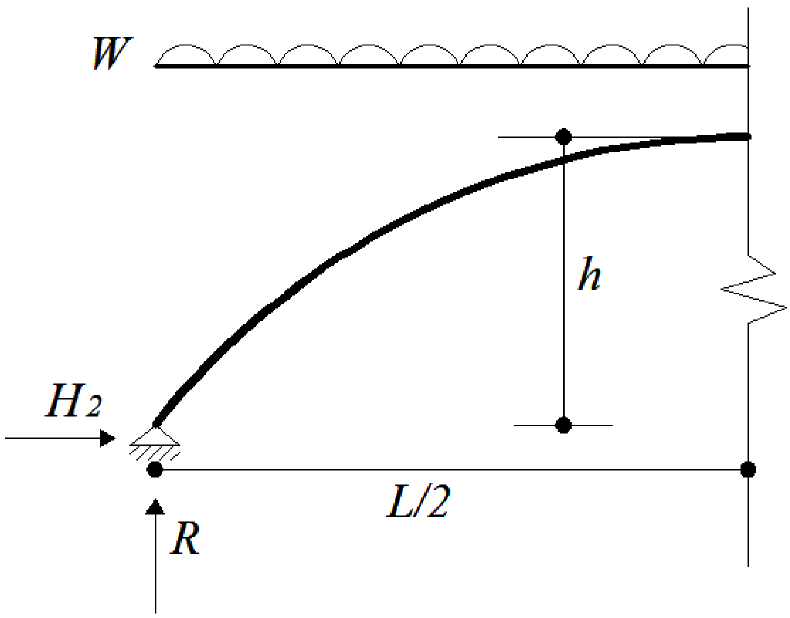


Figure 5.32: Geometry, Support and Load Conditions of Circular Geometry of Arch

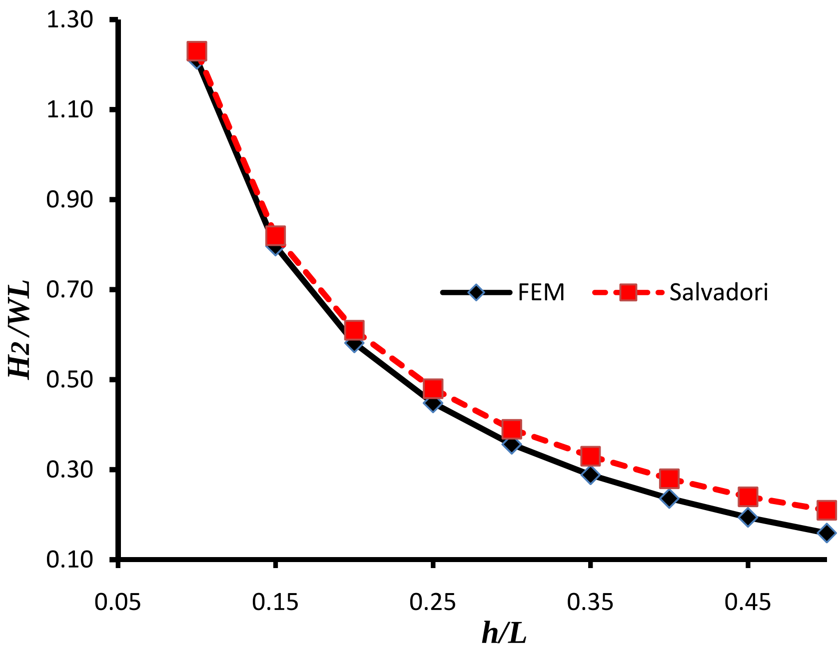


Figure 5.33: Horizontal Thrust at Different Rise Span Ratio

5.8 Finite Element Analysis

Generally a structural analysis of tied arch bridge requires a static analysis by taking consideration for effect of extension of cable of hanger, effect of rib shortening temperature change and effect of in-plane buckling. Large deflection analysis is not required for in-plane buckling for tied arch bridges because these are insensitive to deformations [26]. In the present study of finite element analysis includes:

i) Static analysis for gravity loads. Material model for analysis is linear for all types of elements but the finite element analysis is nonlinear due to changing-status elements of cable.

Rib shortening effect is neglected by assuming proper detailing in arch rib [26]. Again effect of temperature change is not considered in the analysis by assuming that there will be no differential temperature between the rib and tie. Approximate evaluation of slenderness effect for in-plane buckling analysis is followed by AASHTO [26].

Typical bending moment diagram of arch, axial force diagram of arch and hanger, deformation contour, stress diagram for arch and typical tessela stress in arch are shown in Figures 5.34 to 5.41 for some of different load cases.

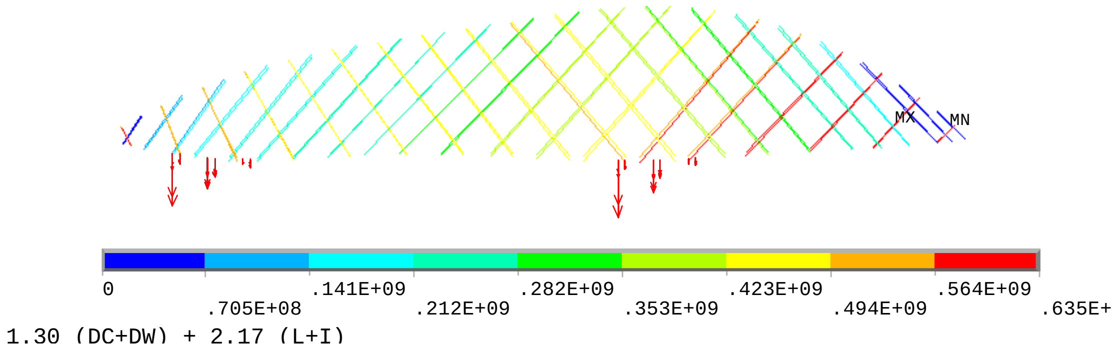


Figure 5.34: Stress Contour in Hangers

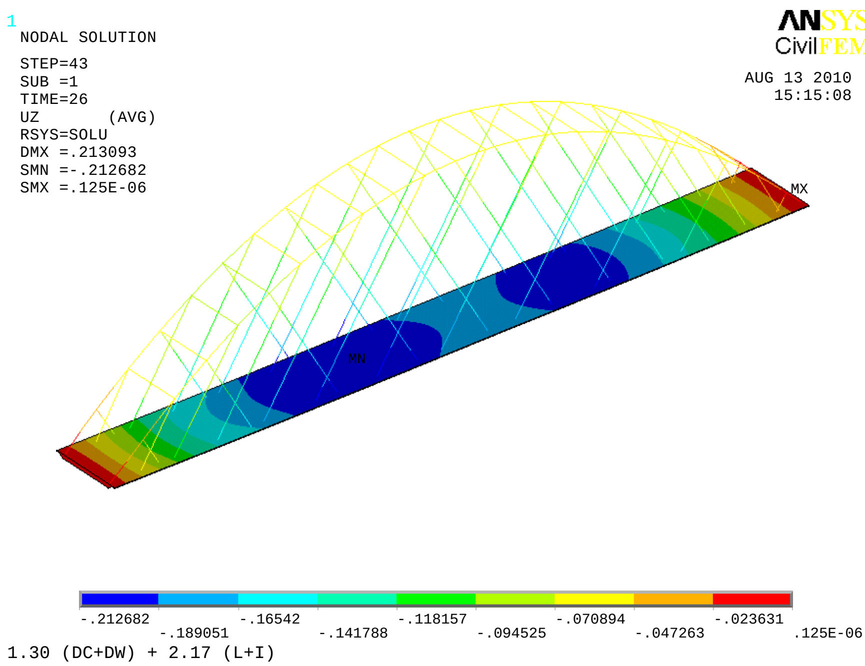


Figure 5.35: Deflection (m) Contour for Load Step 43

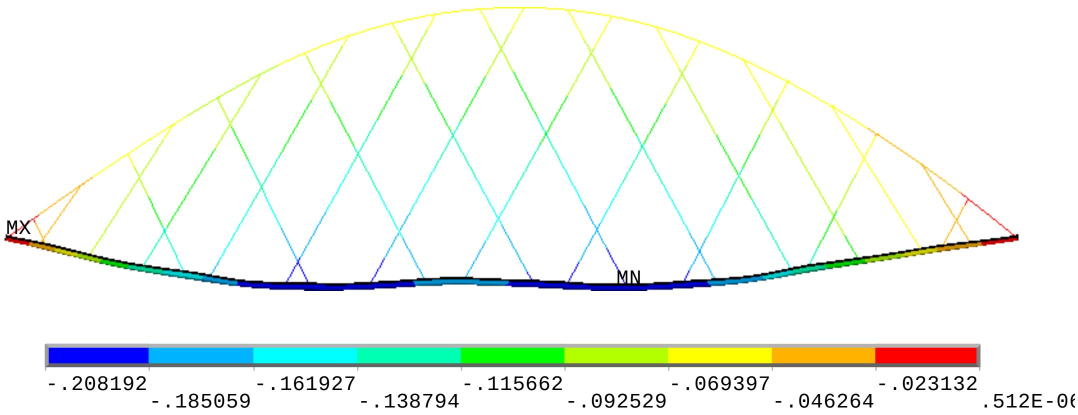


Figure 5.36: Deflection (m) Contour for Load Step 43 (Front View)

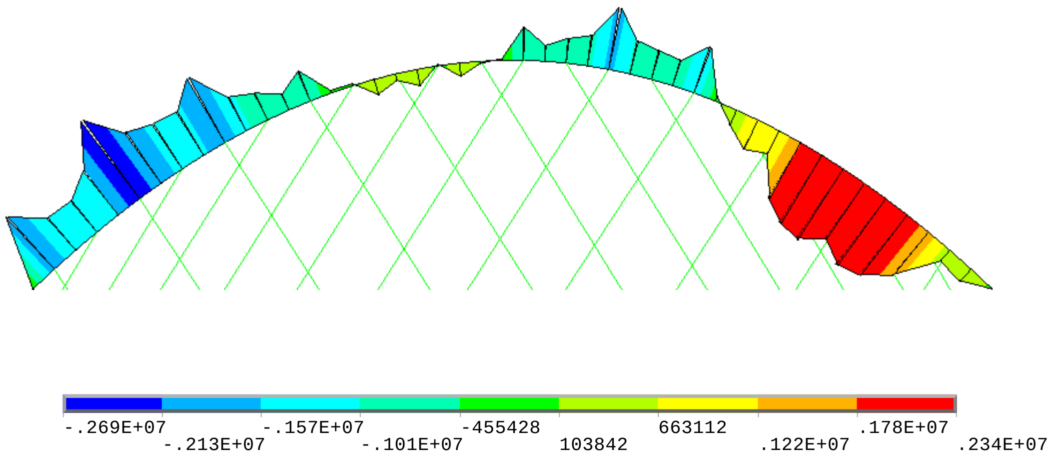


Figure 5.37: Bending Moment (N-m) Diagram of Arch for Load Step 43

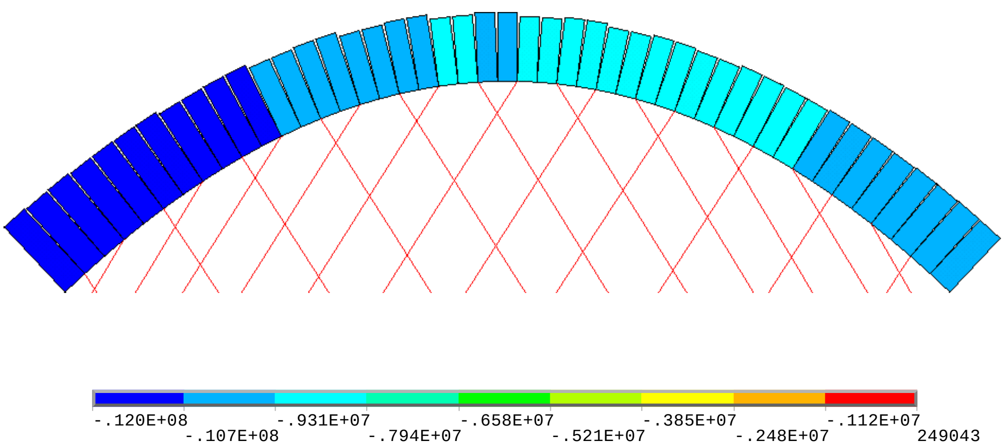


Figure 5.38: Axial Force (N) Diagram of Arch for Load Step 43

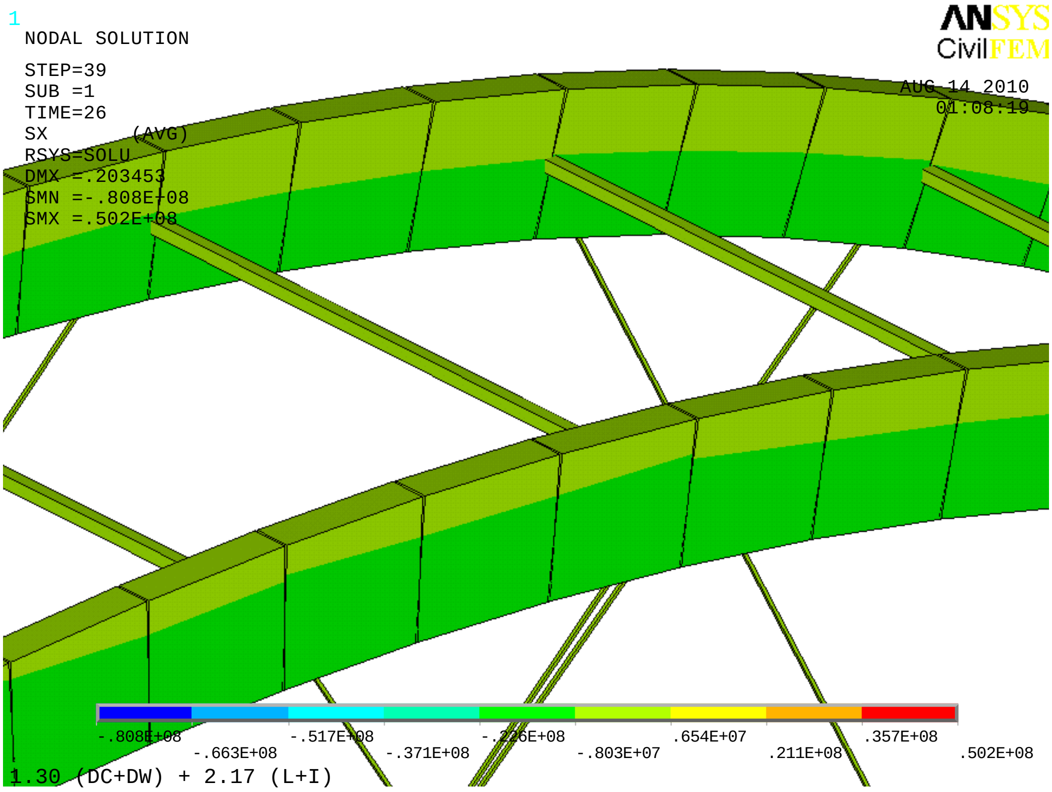


Figure 5.39: Stress (N/m^2) Contour in Arch for Load Step 39

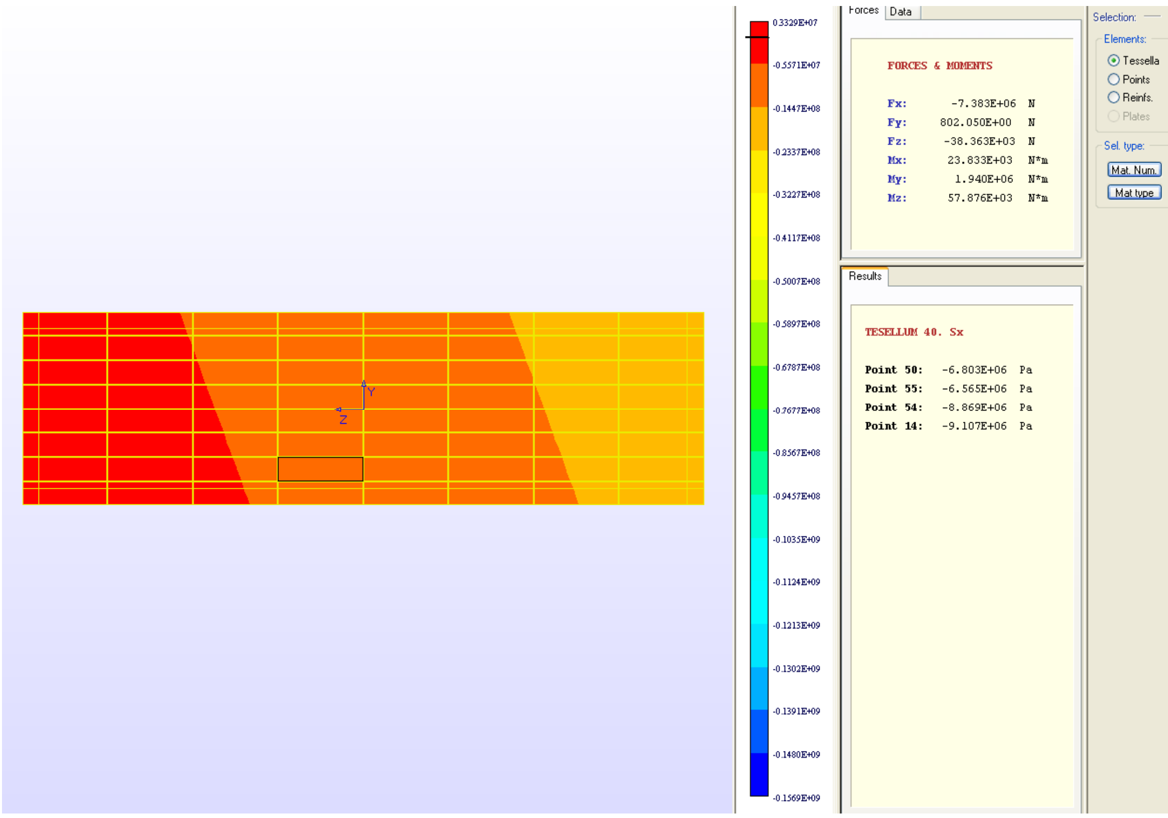


Figure 5.40: Typical Stress Contour at Tessela of Arch Section

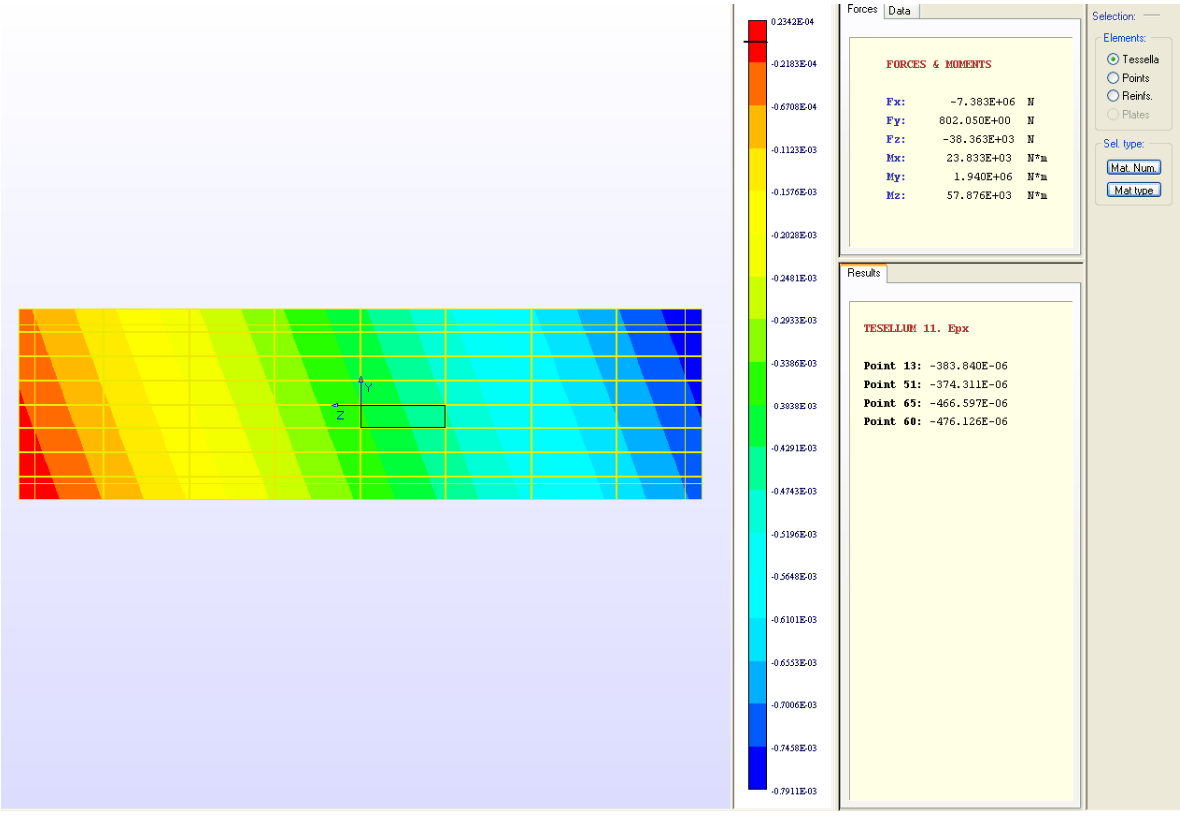


Figure 5.41: Typical Strain Contour at Tessela of Arch Section

5.9 Design and Post Processing for Optimization

After performing finite element analysis of network arch bridges, ANSYS CivilFEM [32] has been used for structural evaluation and then the response is utilized by the optimization algorithm for further data processing. Scripts have been written to record maximum hanger stress of all available hangers for all possible load combinations and considering all vehicle positions acting on the deck. Similarly maximum percentage of steel required in each arch element and total enveloped reinforcement, for all possible load combinations and considering all vehicle positions acting on the deck, is recorded following design of arch using CivilFEM. From post processing consideration, two different scripts of FE analysis and design are written and results from analysis are post processed subsequently which is available for object function and implicit function of EVOP.

5.9.1 Design of Arch

Design of arch using ANSYS CivilFEM [32] followed subsequent steps.

1. Cross section of the arch is determined by optimization algorithm and that arch section is modeled by initial 1% reinforcement in pre processing stage of finite element analysis.
2. After finite element analysis, CivilFEM uses required responses in terms of forces and moments and start design according to AASHTO code.
3. A strength criterion is determined from interaction diagrams of that arch section of initial reinforcement. If the strength criteria is found more than one than the interaction diagram is modified by increasing the reinforcement percentage in such a way that strength criteria reaches to one.

5.9.2 Postprocessing of Arch Design

Design of arch performed by CivilFEM is further post processed for implicit and objective function of optimization algorithm. Maximum percentage of reinforcement is recorded for all load combinations and the result is written in a separate file to make available for implicit function of EVOP. Similarly enveloped maximum percentage of reinforcement is recorded for each element of arch section and total amount of reinforcement and its cost, total volume of concrete of arch and its cost is written in a separate file to make available for objective function of EVOP.

5.9.3 Postprocessing of Hanger Design

Constraints for design of hanger are already assigned in optimization algorithm. For each evaluation of structure maximum hanger stress is recorded after finite element analysis and the result is written in a separate file to make available for implicit function of EVOP. Again total amount of steel and its cost is written in a separate file to make it available for objective function of EVOP.

5.10 Conclusion

Finite element modeling, analysis and design steps of network arch bridge are described in this chapter. Results are post processed for further data processing of optimization algorithm. A series of function evaluation is required by the optimization algorithm, EVOP and EVOP guides finite element software for analysis of the structure by batch execution and terminate after each data processing. Total iteration of FE analysis is completed when the result of optimization is converged to its optimum design.

CHAPTER 6

RESULTS OF OPTIMIZATION

6.1 General

This chapter describes results of simulation driven optimization of network arch bridge design which is accomplished, in the study, by considering various design variables satisfying their explicit and implicit constraints and to minimize cost of some elements of superstructure. Cost parameters are taken from RHD [28]. Circular and parabolic arch geometries have been considered as design constant parameters to identify parametric effect on the design. Finally traditional method of design of arch, hanger and hanger arrangements of a typical two lane 100 meter roadway bridge is compared with the optimum design extracted from global optimization. Again optimal design of arch with optimal hanger arrangement is compared with the arch of optimum design variables but with vertical hangers.

6.2 Optimum Design

Optimum design of network arch bridge is conducted by an optimization engine discussed in earlier Chapter 4 through an optimization process. A feasible design of the bridge in the form of ANSYS APDL is invoked to the optimization engine and after completion of optimization process optimal design variables of the bridge are obtained. Design variables are hanger arrangement, number of hangers, cross sectional dimensions of hangers and rectangular arch, and rise to span ratio of arch. Explicit constraints of design variables are taken from architectural and practical consideration and from design code. Then a tied arch bridge consisting of vertical hangers for both circular and parabolic arches is taken as the initial feasible design for the problem and finally the optimization process results in a network arch bridge. The process is simply outlined as shown in Figure 6.1.

6.3 Optimum Design for Circular Geometry of Arch

Optimum design for the circular geometry of arch is compared with initial feasible design in Figures 6.2 and 6.3. Optimization process for circular arch geometry for a typical two lane 100 meter roadway bridge results in 20 no of hangers per arch. Optimal hanger arrangement exists if hanger inclination with vertical varies from

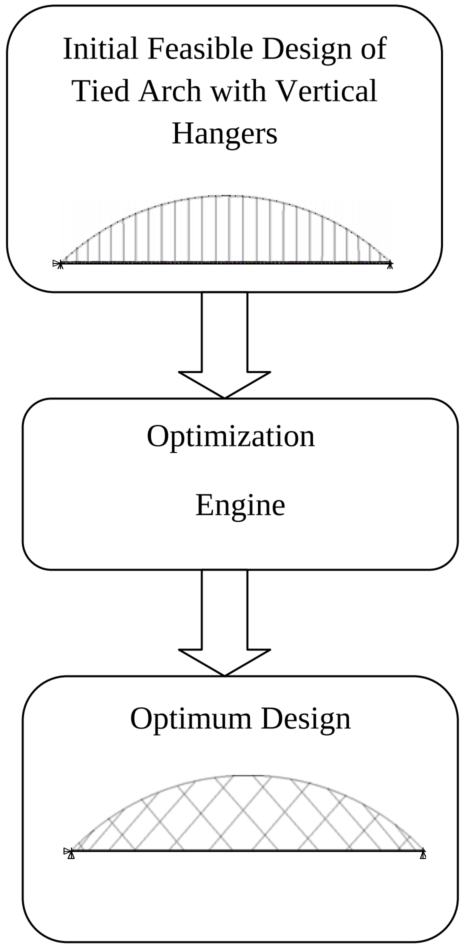
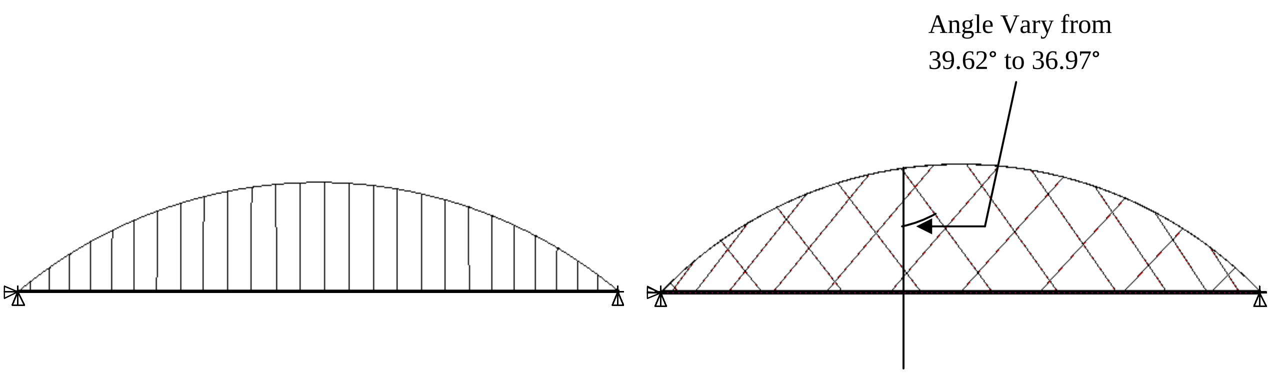
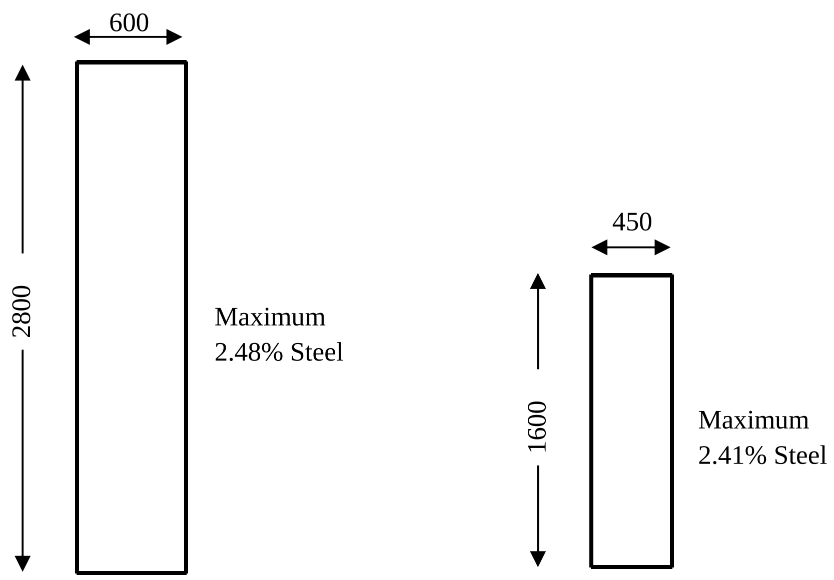


Figure 6.1: Optimization Process Results Network Arch



(a) **(b)**

Figure 6.2: Circular Arch Geometry with (a) Initial Design Variables (b) Optimum Design Variables



(a) **(b)**

Figure 6.3: Arch Section of Circular Arch Geometry (a) Before Optimization (b) After Optimization

Table 6.1: Result Comparison of Design Variables and Response Parameters of Circular Arch Geometry

	Parameter	Unit	Before optimization	After optimization
Explicit Constraints (Design Variables)	Number of Hangers, N_h	-	26	20
	Start Angle for Hanger Inclination Set1, $\varphi 1$	Degree	0	39.62
	Start Angle for Hanger Inclination Set2, $\varphi 1$	Degree	0	36.97
	Angle Change for $\varphi 1$, $\Delta\varphi 1$	Degree	0	-0.265
	Angle Change for $\varphi 2$, $\Delta\varphi 2$	Degree	0	0.265
	Cross Sectional Area of Cable of Hanger, A_h	mm^2	800	1548.4
	Arch Width, B_a	mm	600	450
	Arch Depth, H_a	mm	2800	1600
	Rise to Span Ratio, R_a	-	0.18	0.2105
Implicit Constraints	Extreme Hanger Stress, σ_{max}	MPa	901.6	1101.62
	Strength Criteria of Arch, CRT	-	1	1
	Maximum Design Reinforcement Factor in Arch, RNR	%	2.482	2.41
	End Angle for Hanger Inclination Set1, $\varphi 1_n$	Degree	0	36.97
	End Angle for Hanger Inclination Set2, $\varphi 2_n$	Degree	0	39.62
	Slenderness Ratio	-	9.45	15.1
	Arch Depth to Width ratio	-	4.67	3.55

39.62 to 36.97 degree. Zinc coated bridge wire of ASTM A586 of cross sectional area 1548.4 mm^2 is found optimum for the present design constants. Again arch rise to span ratio of 0.2105 is found optimum and rectangular arch section of 450 mm by 1600 mm is the most economical design. After design, maximum arch reinforcement percentage is found 2.41%.

Result comparison of traditional and optimal design for circular arch geometry is enlisted in Table 6.1.

6.3.1 Comparison of Influence Lines for Circular Arch

Arch with optimum design variables is compared with the arch of optimal design variables but with vertical hanger arrangement. Finite element analysis has been conducted for generation of ordinates of influence lines at different locations of the arch. Finally influence lines for optimal and vertical hanger arrangements with the same arch section are compared. A typical arch bridge configuration is shown in Figure 6.4 where X_i is the distance of arch position where influence line ordinates is needed to determine and K is a factor for position of arch along the span, L, and X_p is the distance of unit load position and a is the factor for position of unit load along the span. For different values of K influence lines for bending moment at the location of KL are shown in Figures 6.5 to 6.13.

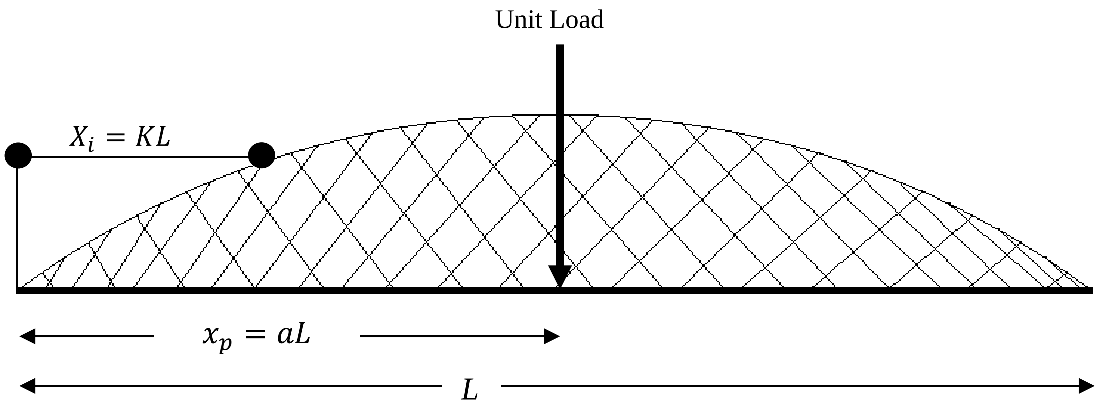


Figure 6.4: Location of Influence Line along the Arch

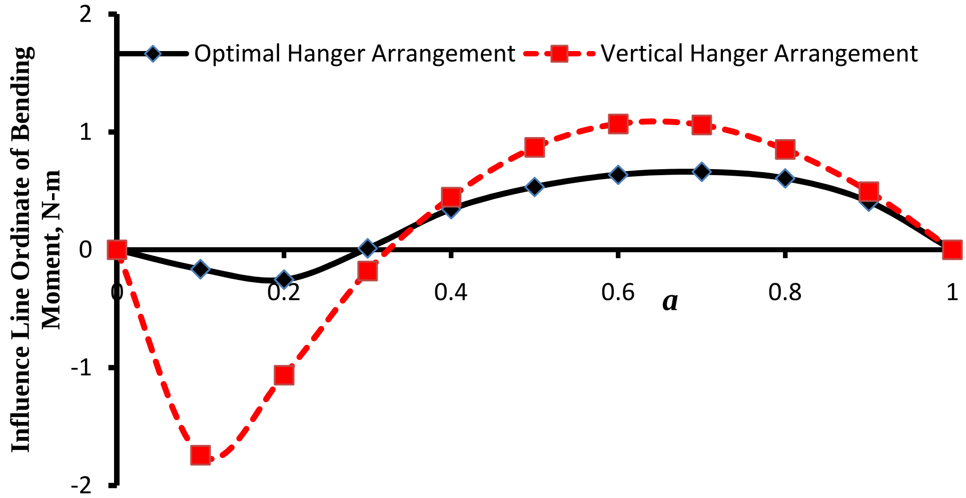


Figure 6.5: Influence Line for Bending Moment at *X = 0.1L*

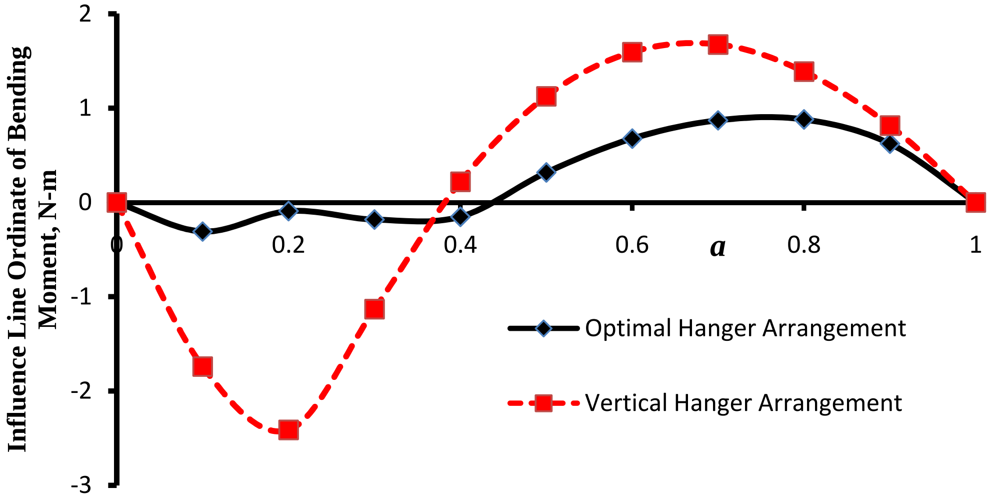


Figure 6.6:Influence Line for Bending Moment at *X = 0.2L*

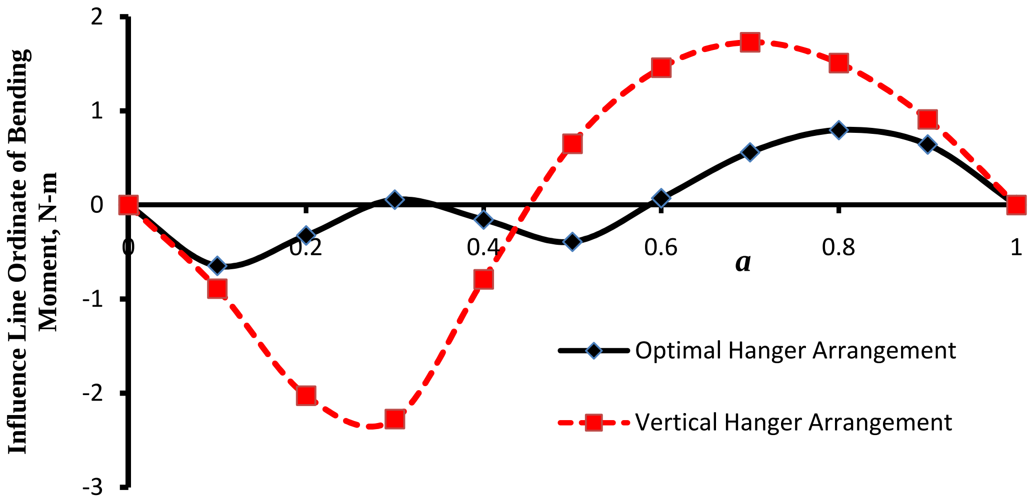


Figure 6.7: Influence Line for Bending Moment at *X = 0.3L*

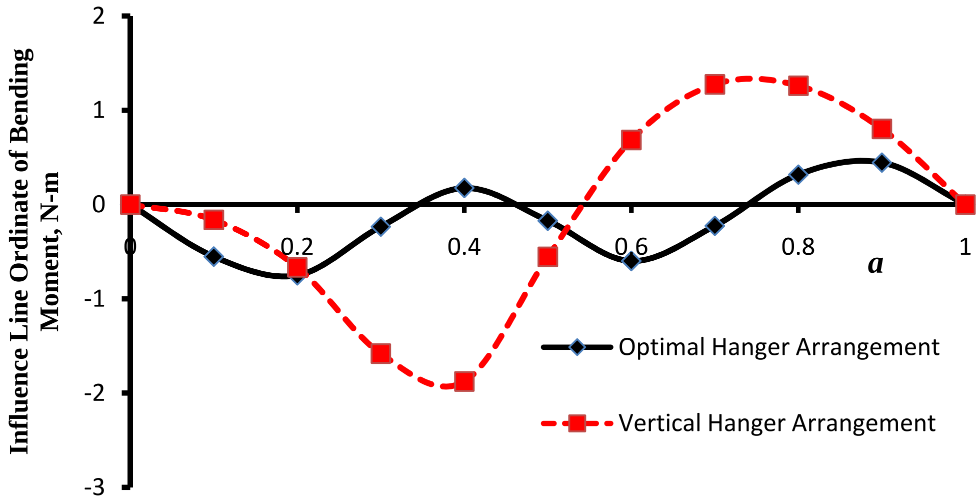


Figure 6.8: Influence Line for Bending Moment at *X = 0.4L*

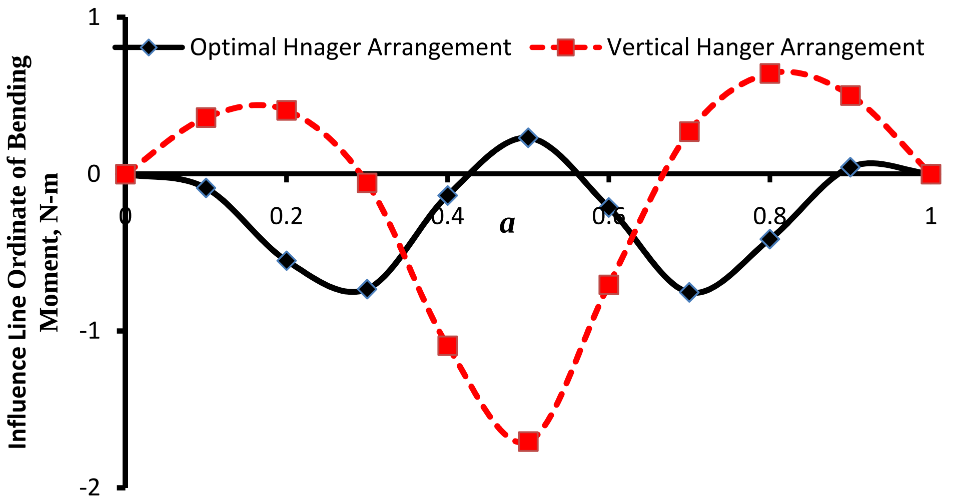


Figure 6.9: Influence Line for Bending Moment at *X = 0.5L*

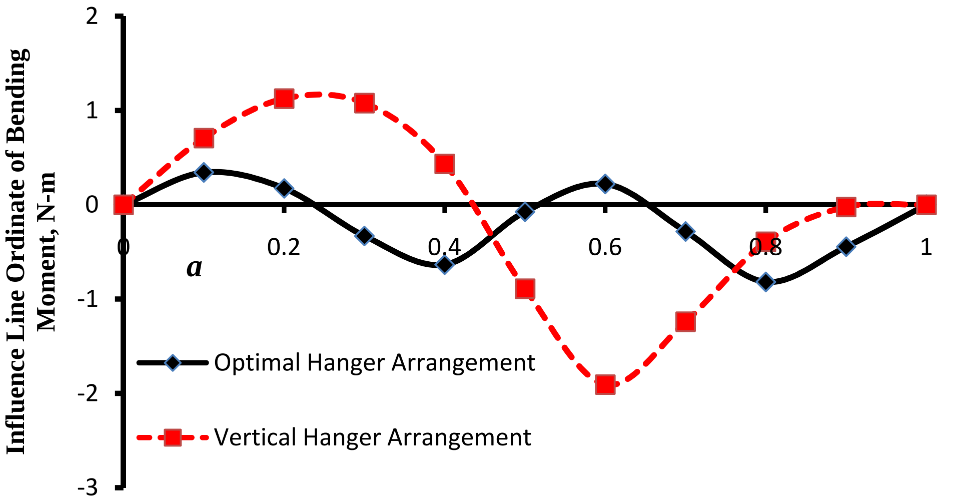


Figure 6.10: Influence Line for Bending Moment at *X = 0.6L*

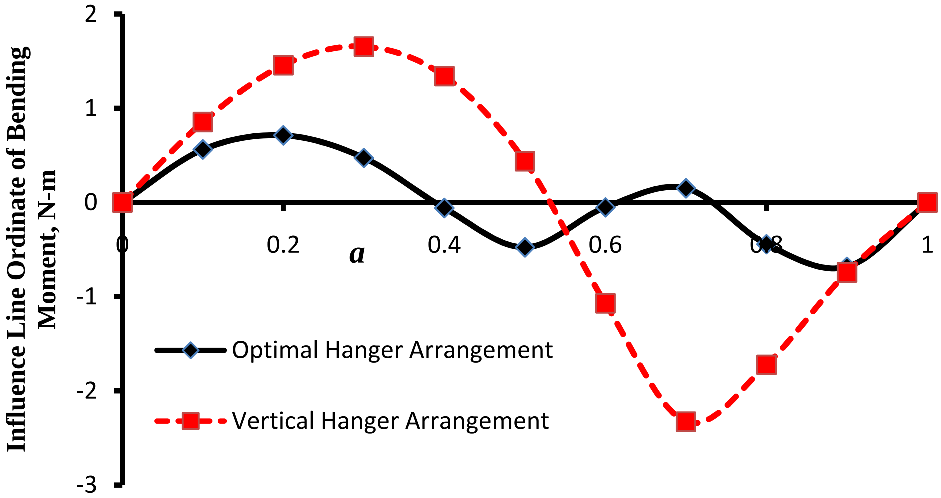


Figure 6.11: Influence Line for Bending Moment at *X = 0.7L*

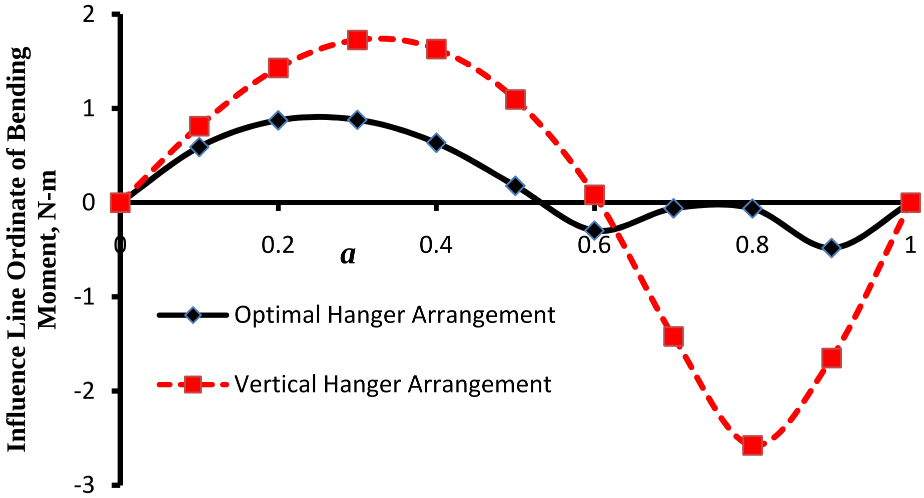


Figure 6.12: Influence Line for Bending Moment at *X = 0.8L*

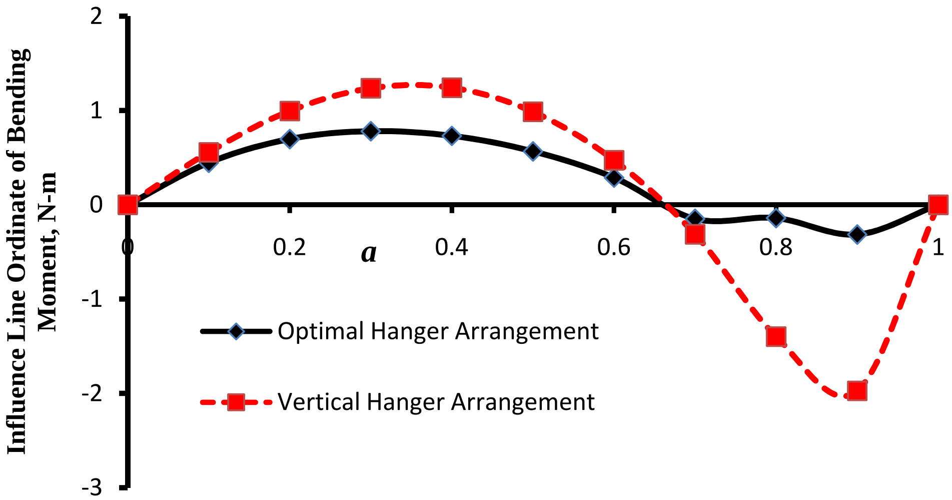


Figure 6.13: Influence Line for Bending Moment at *X = 0.9L*

Ratio of influence line ordinate for bending moment is defined for understanding the affect of optimization as

$$Ratio\ of\ Influence\ Line\ Ordinate\ for\ Bending = \frac{Influence\ Line\ Ordinate\ for\ Vertical\ Hanger\ Arrangement}{Influence\ Line\ Ordinate\ for\ Optimal\ Hanger\ Arrangement}$$

Ratio of influence line ordinate for bending at different location is depicted in Figure 6.14. The figure shows that influence line ordinate for vertical hanger arrangement is approximately 2 to 40 times larger than that of optimal hanger arrangement.

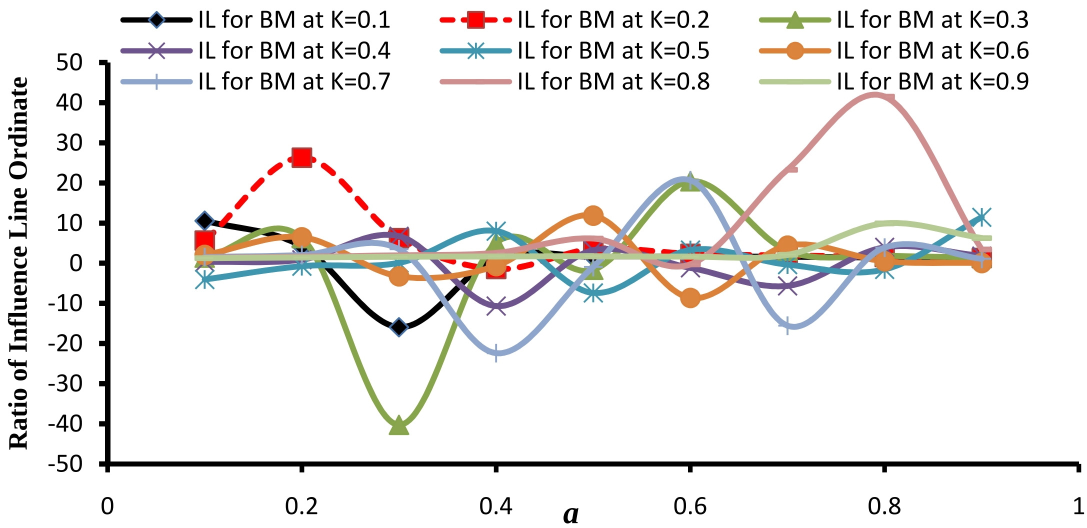


Figure 6.14: Ratio of Influence Line Ordinate for Bending Moment at *X* = *KL*

6.3.2 Comparison of Results for Bending Moment, Axial Force, Strength Criteria and Reinforcement Percentage for Circular Arch

An intelligent computer program is developed which can detect proper positions of wheel and lane loads for maximizing positive or negative moment in the arch. Out of 36 lane load steps and 100 single vehicle load steps, location of load for load step 47 is shown in Figure 6.15. Load step 47 is considered for comparison because in this step location of load for vertical and optimal hanger arrangement is same Application of the load step in finite element analysis is shown in Figure 6.16. Comparison of bending moment, axial force, strength criteria and reinforcement percentage in the arch for vertical and optimal hanger arrangements is shown in Figures 6.17 to 6.25.

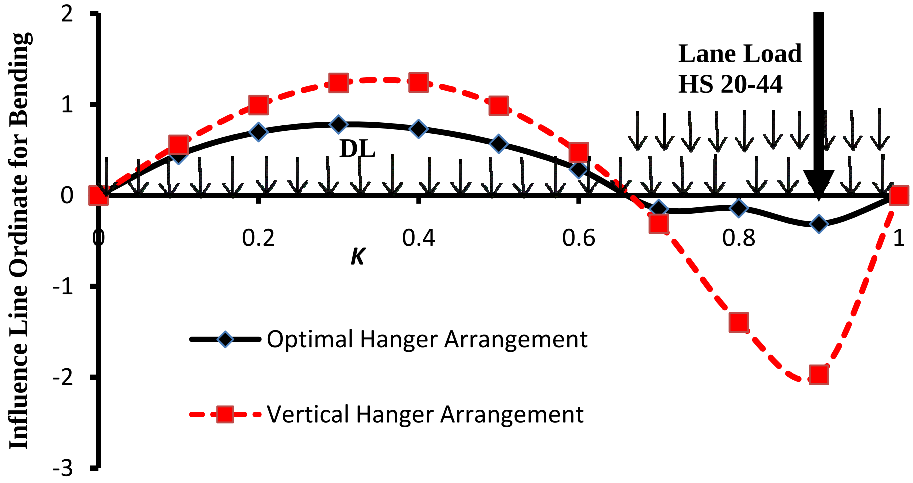
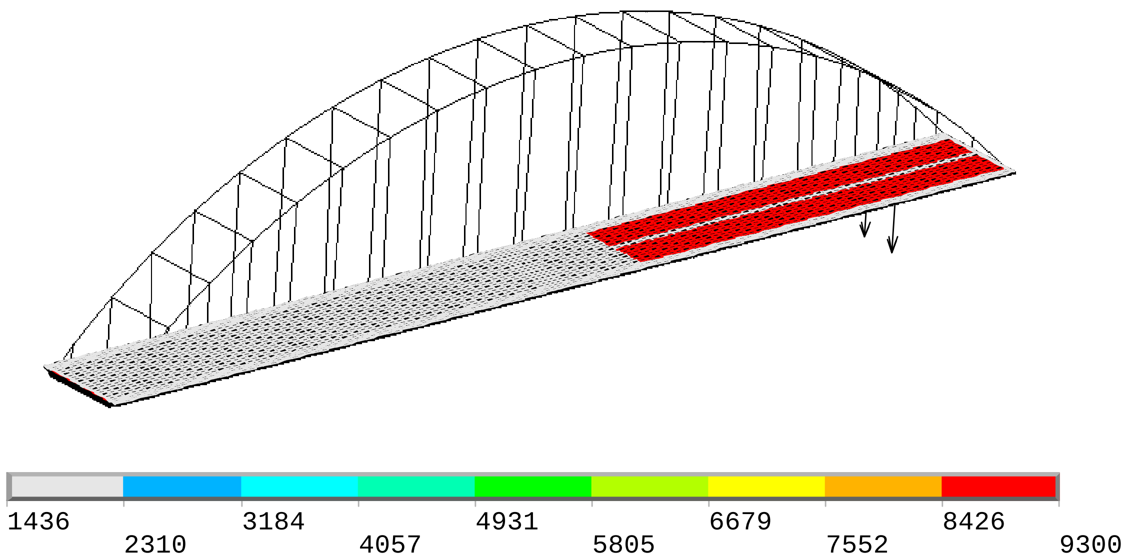
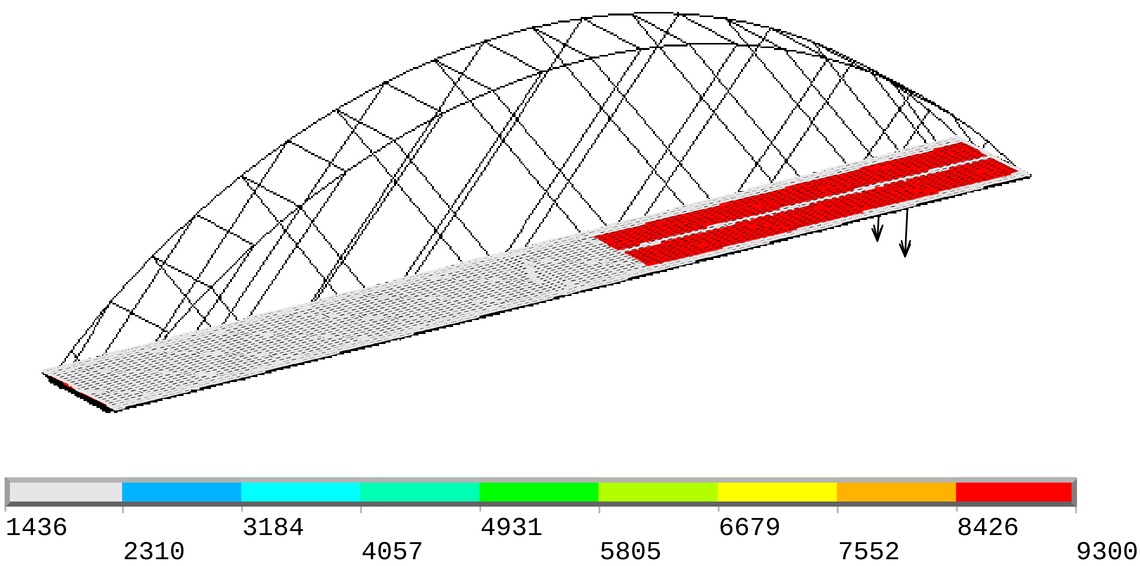


Figure 6.15: Influence Line Diagrams for Bending Moment at *X = 0.9L* for Load Position at Step 47



(a)



(b)

Figure 6.16: Load Step 47 Showing Live Load on the Deck of Arch (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

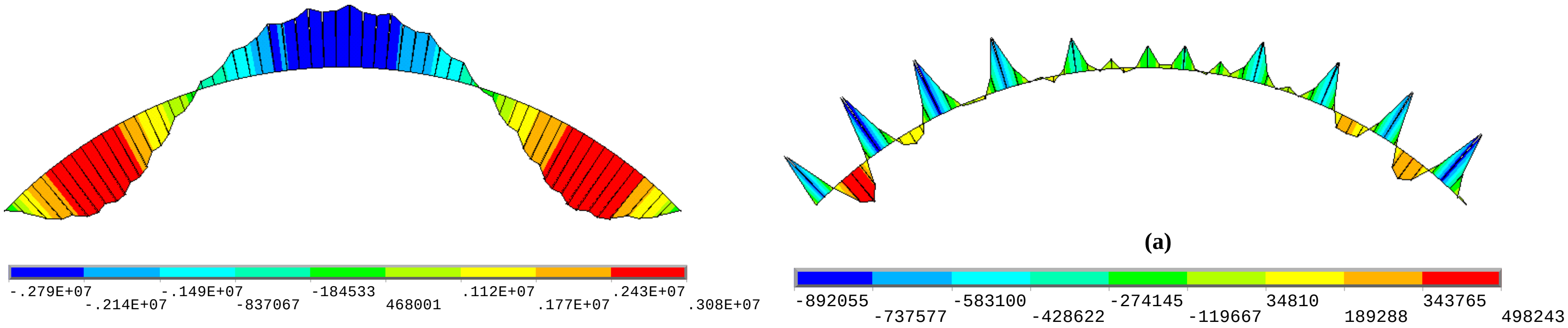


Figure 6.17: Bending Moment in N-m for Dead Load only (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

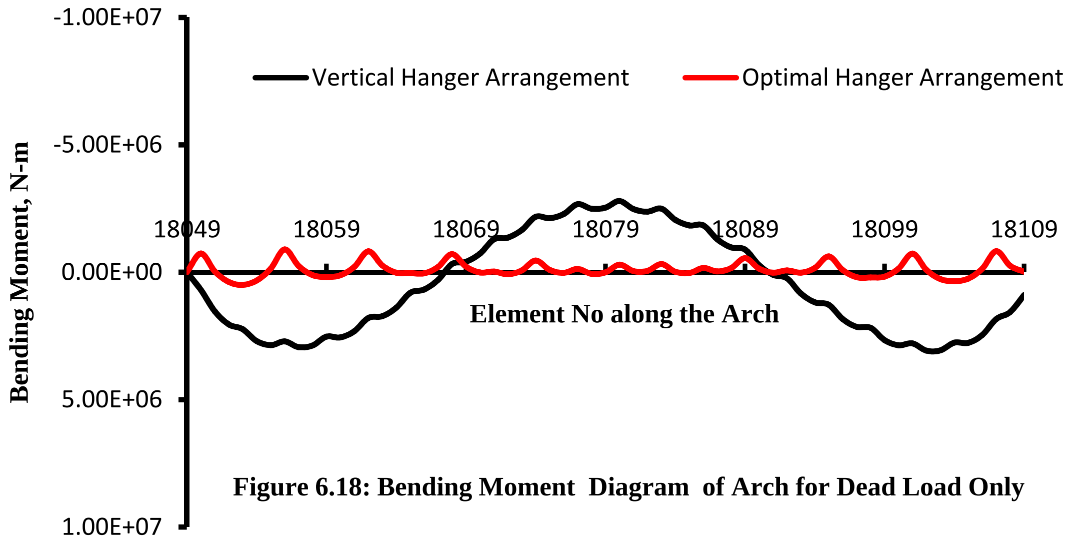


Figure 6.18: Bending Moment Diagram of Arch for Dead Load Only

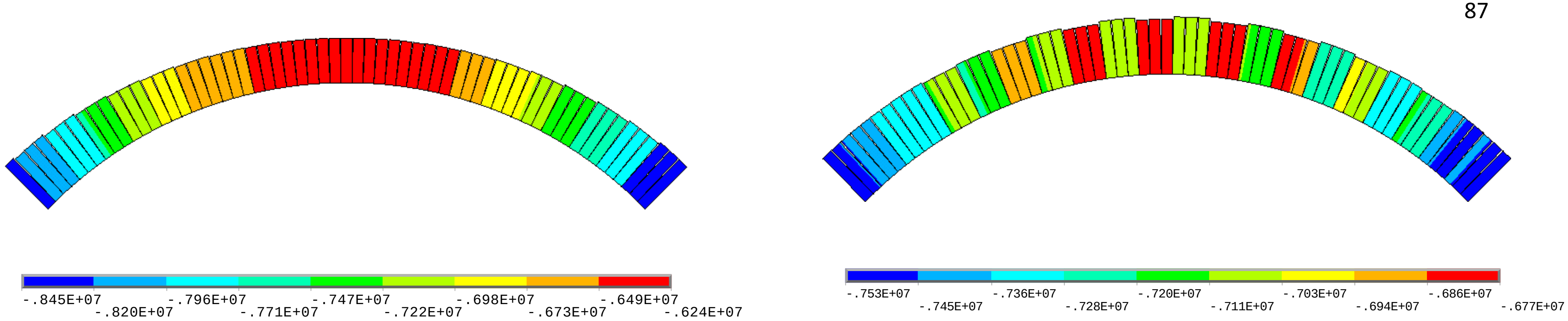


Figure 6.19: Axial force in N for Dead Load only (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

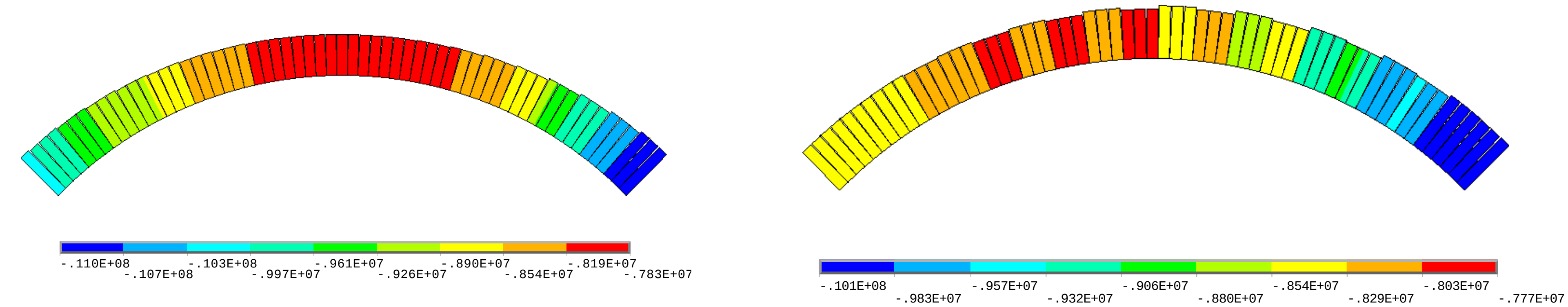


Figure 6.20: Axial Force in N for Load Step 47 (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

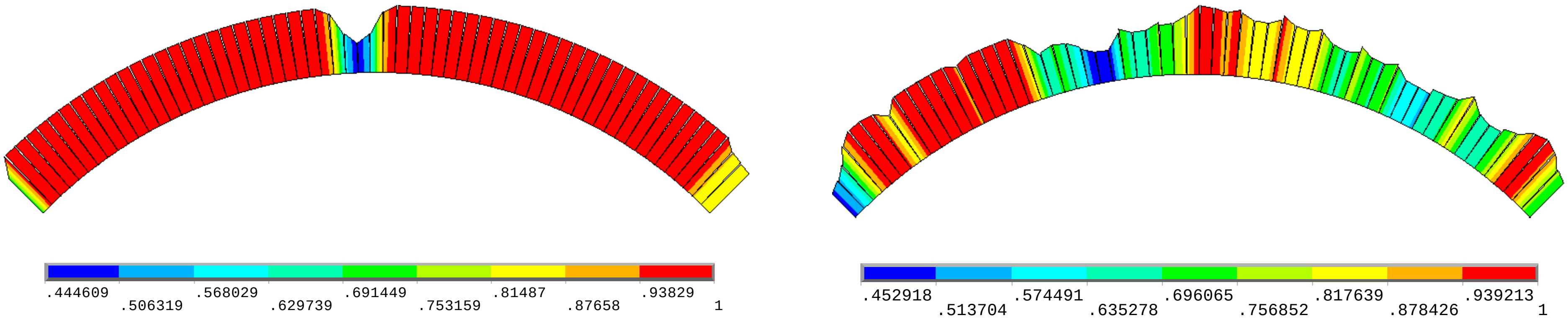


Figure 6.21: Strength Criteria for Load Step 47 (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

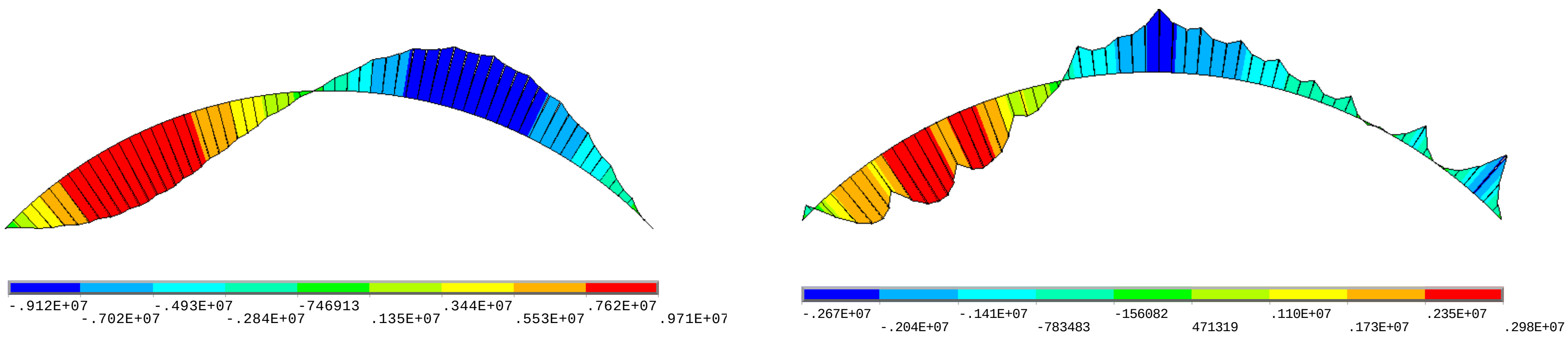


Figure 6.22: Bending Moment in N-m for Load Step 47 (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

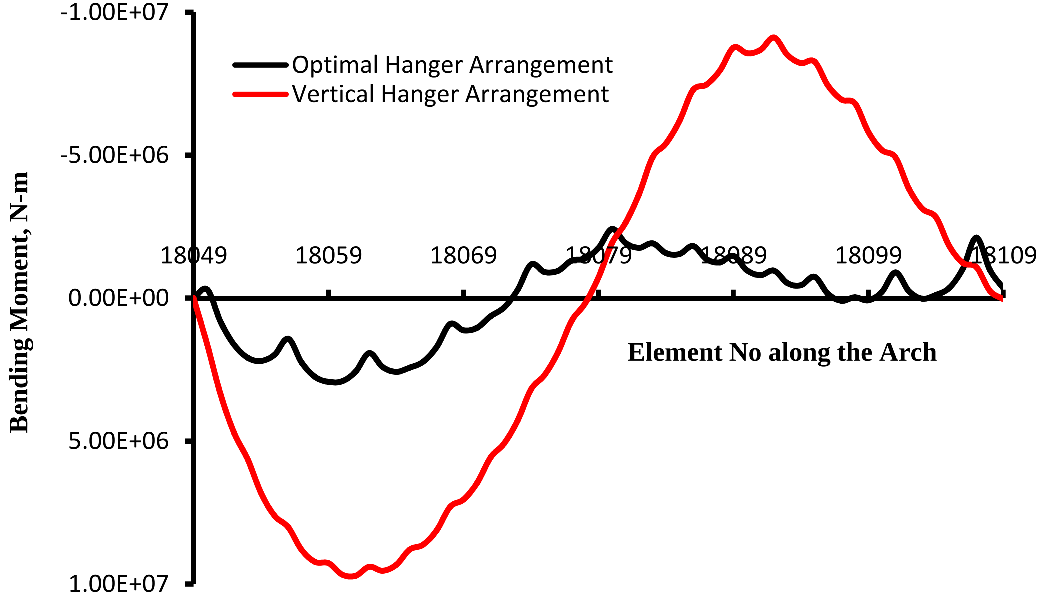


Figure 6.23: Bending Moment Diagram of Arch for Load Step 47

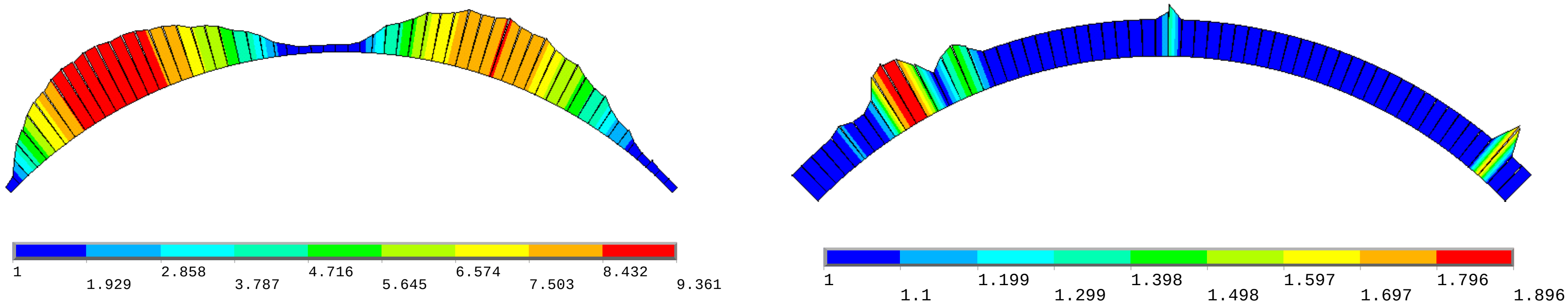


Figure 6.24: Reinforcement Percentage for Load Step 47 (a) With Vertical Hanger Arrangement (b) With Optimal Hanger Arrangement

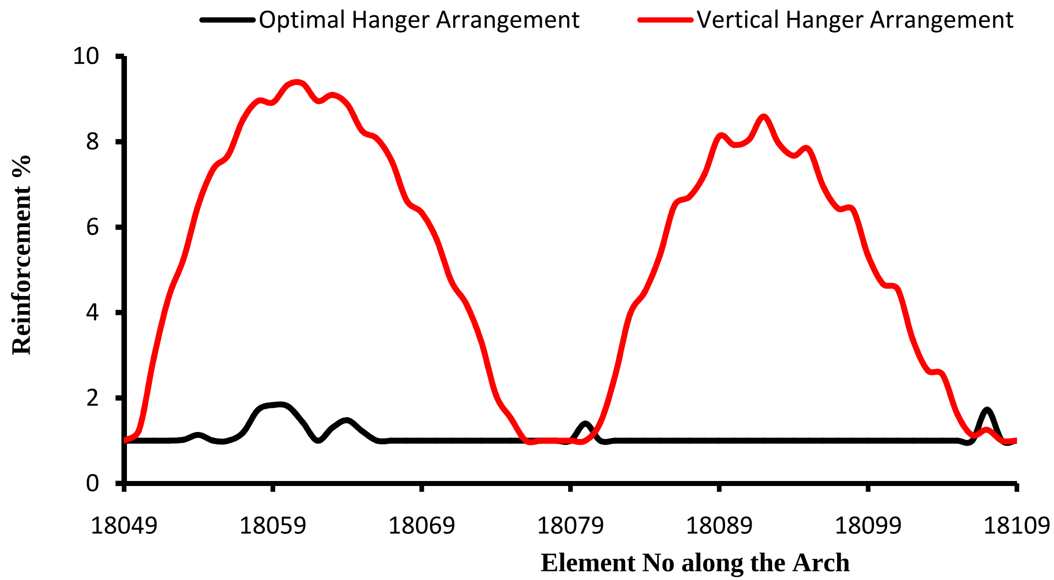
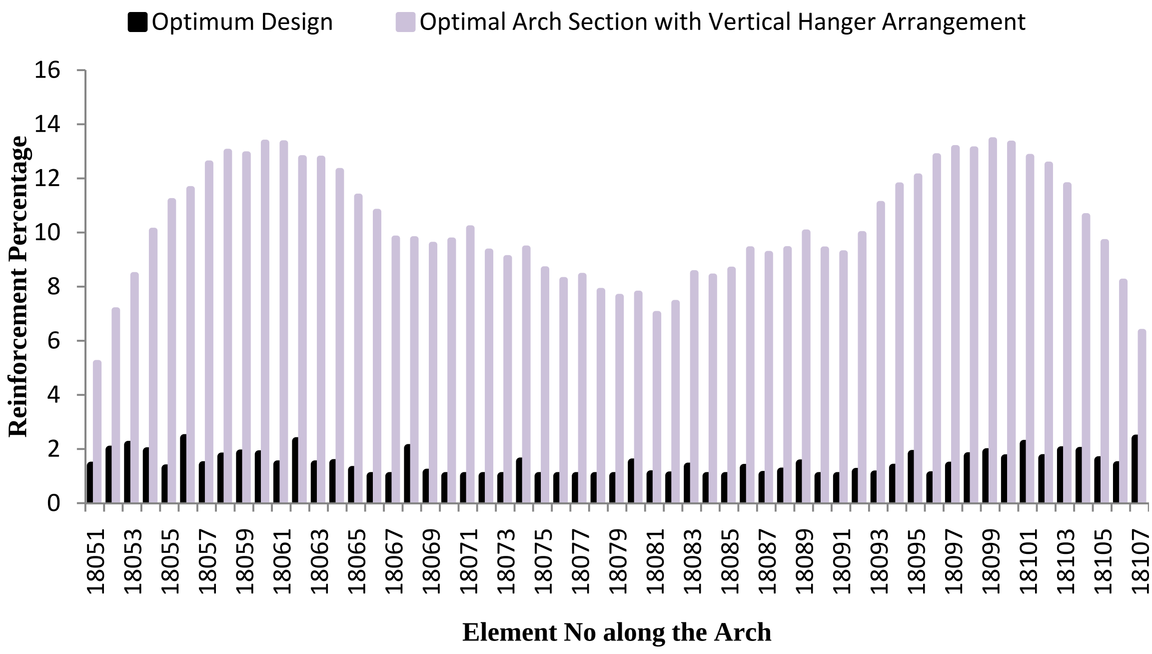


Figure 6.25: Reinforcement Percentage Envelope of Arch at load Step 47

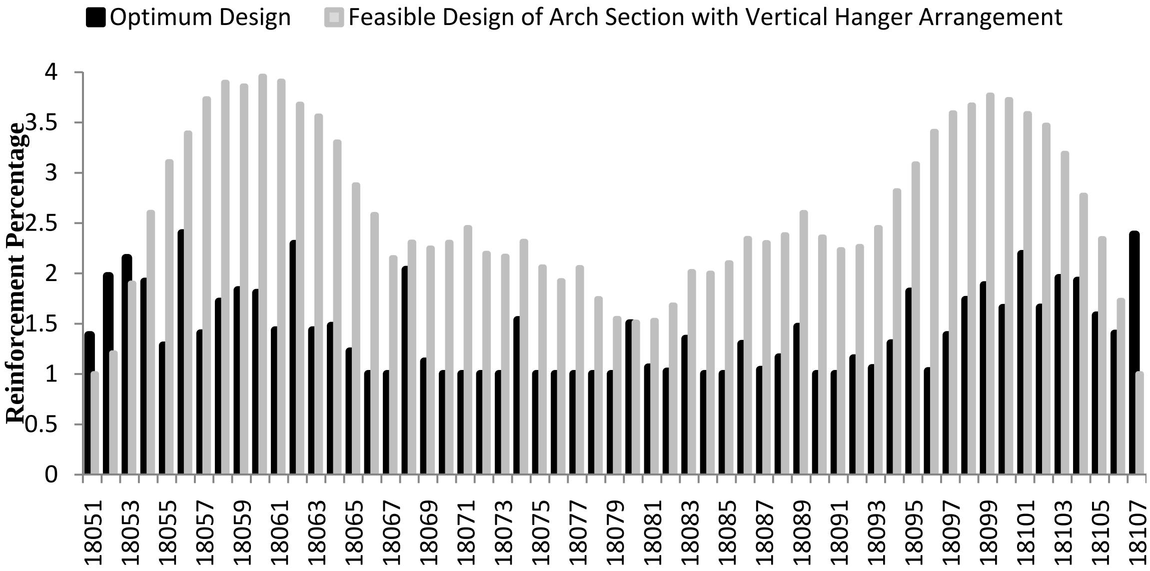
Result shows for dead load only that bending moment in the arch with optimal design variables is negligible than that of arch with vertical hangers. Axial force for both cases remains similar.

For load step 47 results show that bending moment in the arch with optimal design variables is 3 to 4 times less than the arch with vertical hangers. Axial forces for both cases have similar values. Due to large bending moment in arch with vertical hangers the arch either requires large amount of steel or large arch section. By considering constant arch section it has been shown in Figure 6.25 that arch with vertical hanger requires a maximum 9% steel which is not feasible but signifies virtues of optimal hanger arrangement.

Reinforcement percentage envelope required along the arch for all load steps is shown in Figure 6.26. Reinforcement percentage is drawn in the Figure 6.26(a) by considering same arch section for vertical hanger and optimal hanger arrangements. Reinforcement percentage envelope for feasible design of arch section with vertical hanger arrangement is also plotted in the Figure 6.26(b). Results show that maximum amount of steel is required at quarter span for vertical hanger arrangements. Again vertical hanger arrangement requires larger amount of steel compared with optimal hanger arrangements. Optimum design requires steel same to all over the arch whereas vertical hanger arrangement requires a variation of steel along the arch.



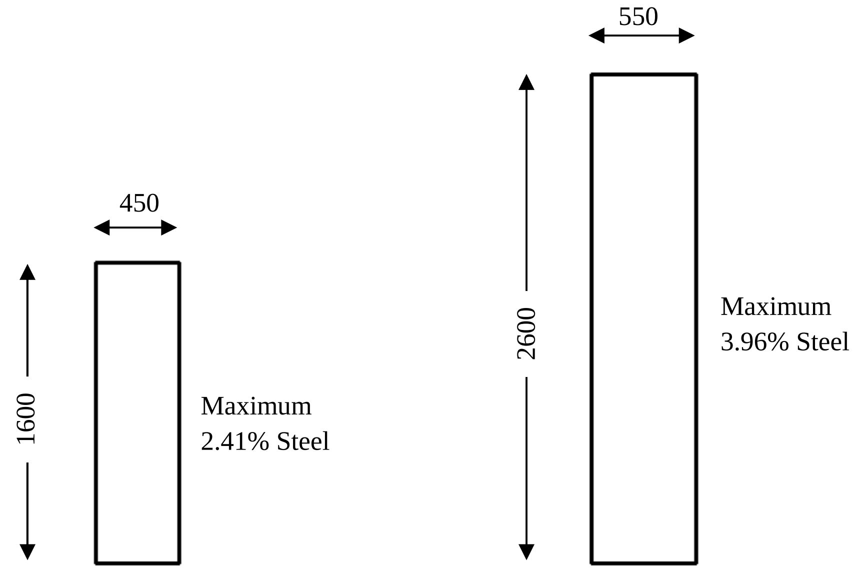
(a)



(b)

Figure 6.26: Reinforcement Percentage Envelope along the Arch of Circular Arch Geometry

By considering feasible design of arch of less than 4% of reinforcement with vertical hangers but with other optimal variables, larger arch section is required. A comparison between optimum arch section and a feasible design with vertical hanger arrangement is depicted in Figure 6.27.



(a) **(b)**

Figure 6.27: (a) Optimal Arch Section (b) Feasible Design of Arch Section of Vertical Hangers of Circular Arch Geometry

6.3.3 Comparison Optimum Values of Objective Function

Optimal hanger arrangement of network arch bridge using global optimization technique shows significant amount of cost saving over the bridges with vertical hangers. A cost breakdown for hanger, concrete of arch and arch reinforcement is outlined in Table 6.2 for initial and final optimal design. Again cost required for optimal design and final design but with vertical hangers is compared in Table 6.3.

Table 6.2: Result Comparison for Cost of Initial and Optimal Design of Circular Arch Geometry

Cost Breakdown	Initial	Final
Cost of Cable (BDT)	0.100001412E+07	0.199471517E+07
Cost of Concrete of Arch (BDT)	0.417691526E+07	0.180762830E+07
Cost of Reinforcement in Arch (BDT)	0.345129619E+07	0.134102902E+07
Total Cost (BDT)	0.862822557E+07	0.514337249E+07
		40.38% Save

Table 6.3: Result Comparison for Cost of Optimal Design and Final Design with Vertical Hangers of Circular Arch Geometry

Cost Breakdown	Final	Vertical Hangers with Final Design Variables
Cost of Cable (BDT)	0.199471517E+07	0.156038691E+07
Cost of Concrete of Arch (BDT)	0.180762830E+07	0.180762830E+07
Cost of Reinforcement in Arch (BDT)	0.134102902E+07	0.960399891E+07
Total Cost (BDT)	0.514337249E+07	0.129720141E+08
		75.63% Save of Reinforcement

Optimization approach developed in the study shows 40.38% cost can be saved for circular arch if design is optimized. Again it has been shown that reinforcement required in arch with optimal hanger saves 75.63 % steel than that of vertical hanger.

Optimum design for the circular arch is compared with traditional design by Brunn et al. [7] in Figure 6.28. Network arch bridge design by Tveit requires 48 no of hangers per arch for 100 m long bridge and hanger inclination varies from 75 to 7 degree with vertical. Brunn [7] does not optimize the cost but used trial and error approaches to reduce bending moment in arch. Optimization of the arch bridge considering cost of both arch and hangers reduces no of hangers to 20 no per arch. A cost breakdown of optimal design and traditional design of network arch bridge is enlisted in Table 6.4. Result shows optimum design by using global optimization technique can save 40% cost than the traditional method of design prescribed by Brunn et al. [7].

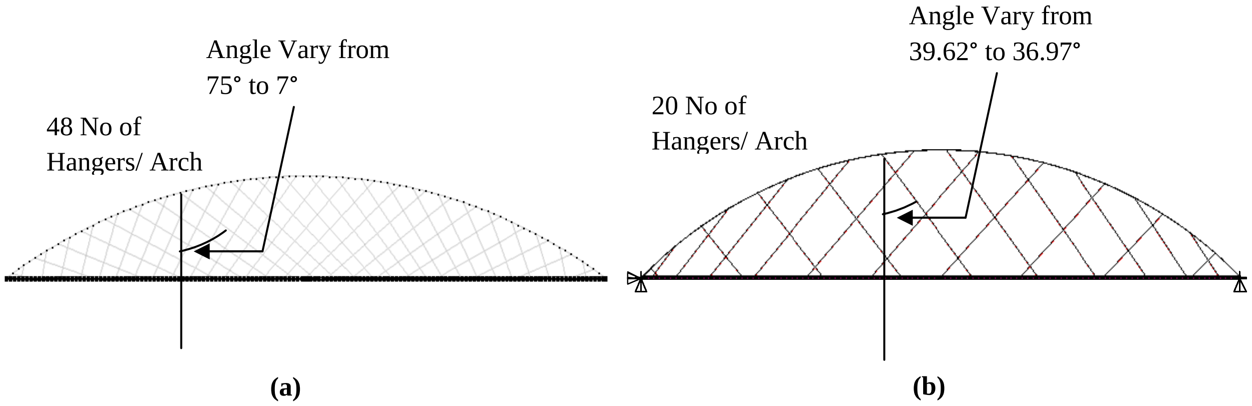


Figure 6.28: Circular Arch Geometry with (a) Traditional Design by Brunn et al. [7] (b) Optimum Design

Table 6.4: Result Comparison for Cost of Traditional (Brunn et al.) and Optimal Design of Circular Arch Geometry

Cost Breakdown	Traditional (Brunn and Tveit)	Final
Cost of Cable (BDT)	0.392999278E+07	0.199471517E+07
Cost of Concrete of Arch (BDT)	0.171805804E+07	0.180762830E+07
Cost of Reinforcement in Arch (BDT)	0.292411092E+07	0.134102902E+07
Total Cost (BDT)	0.857216174E+07	0.514337249E+07
		40% Save

6.4 Optimum Design for Parabolic Arch

Optimum design for the parabolic geometry of arch is compared with initial feasible design in Figures 6.29 and 6.30. Optimization process for the arch geometry for a typical two lane 100 meter roadway bridge results in 22 no of hangers per arch. Optimal hanger arrangement exists if hanger inclination with vertical varies from 32.74 to 31.55 degree. Zinc coated bridge wire of ASTM A586 of cross sectional area 1025.8 mm^2 is found optimum for the present design constants. Again arch rise span

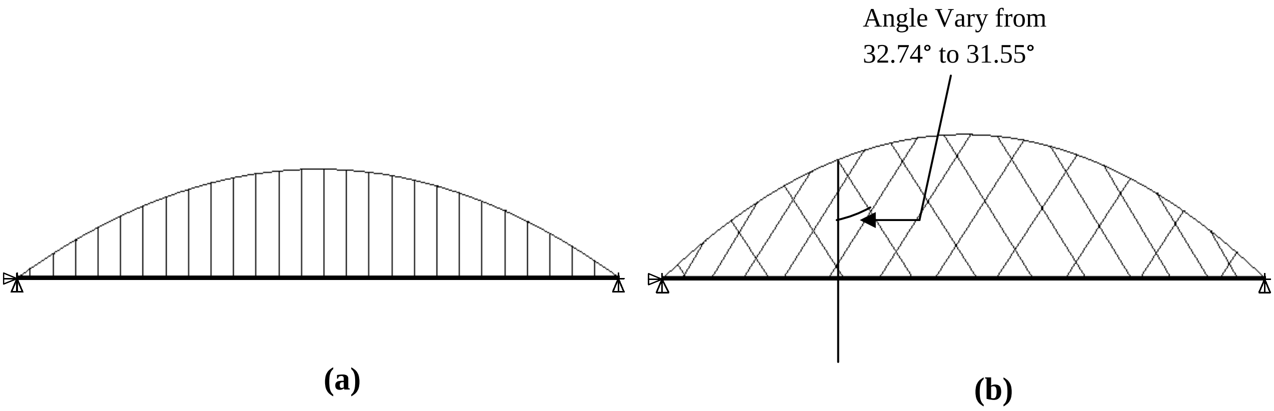


Figure 6.29: Parabolic Arch Geometry with (a) Initial Design Variables (b) Final Design Variables

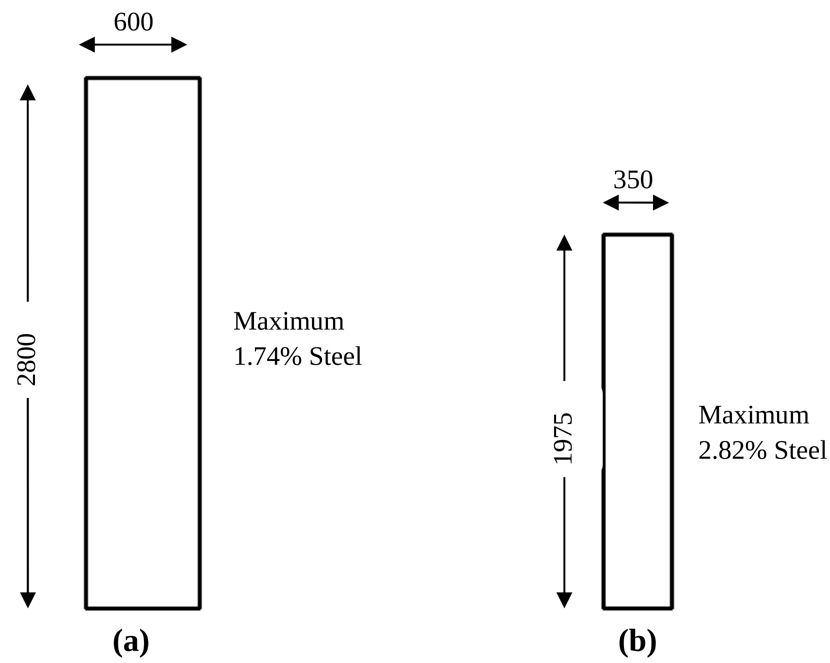


Figure 6.30: Arch Section of Parabolic Arch Geometry (a) Before Optimization (b) After Optimization

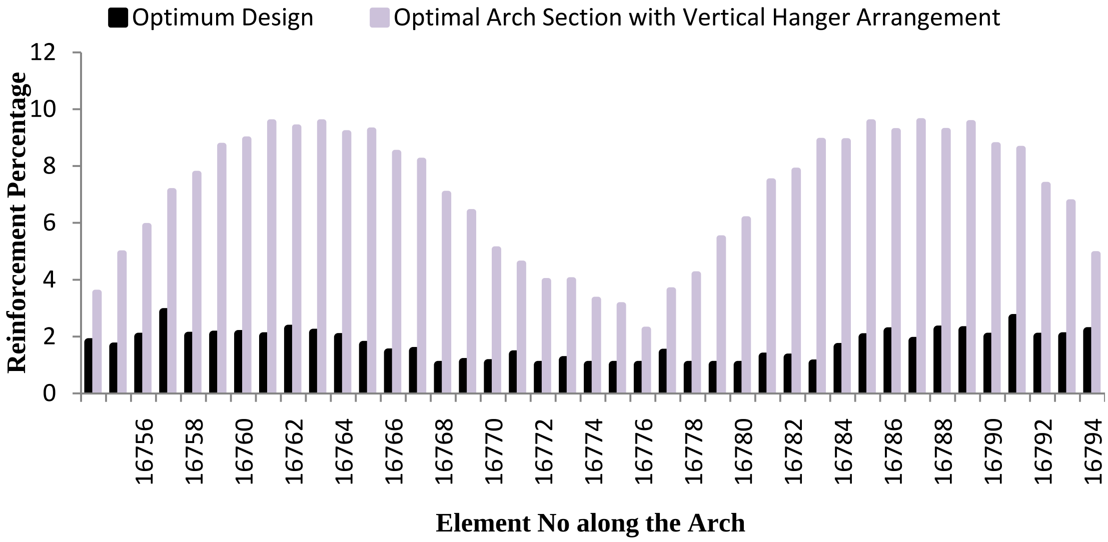
ratio of 0.2386 is found optimum and rectangular arch section of 350 mm by 1975 mm is the most economical design. After design, maximum arch reinforcement percentage is found 2.82%.

Comparison of results for influence line, bending moment, axial force and reinforcement percentage in arch for circular geometry is described in above articles. For parabolic geometry of arch, the comparison is similar, and only deign variables, response parameters and objective function is compared in the following articles.

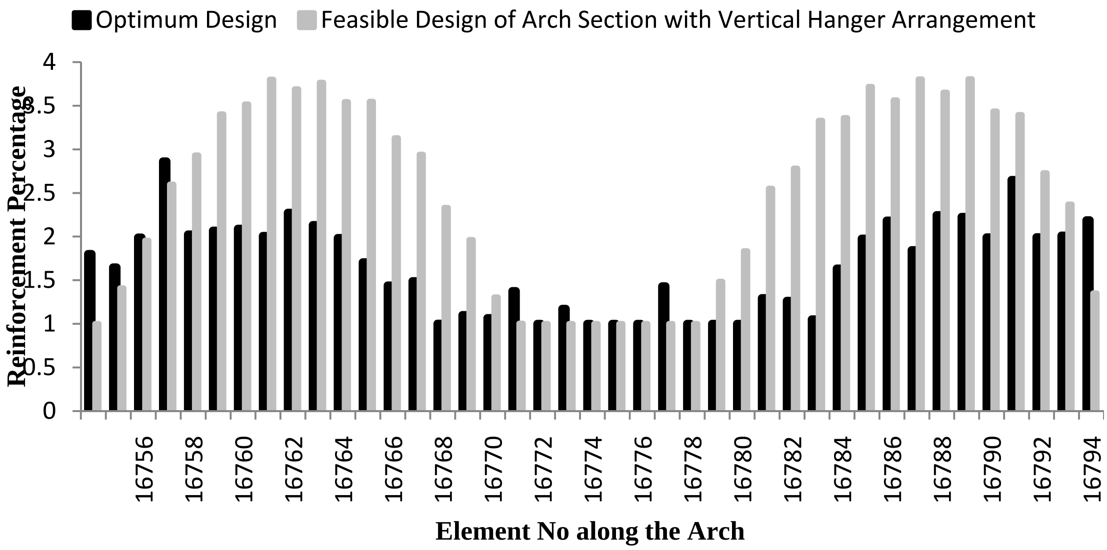
The result comparison of traditional and optimal design for parabolic arch geometry is enlisted in Table 6.5.

Table 6.5: Result Comparison of Design Variables and Response Parameters of Parabolic Arch Geometry

	Parameter	Unit	Before optimization	After optimization
Explicit Constraints (Design Variables)	Number of Hangers, N_h	-	26	22
	Start Angle for Hanger Inclination Set1, $\varphi 1$	Degree	0	32.74
	Start Angle for Hanger Inclination Set2, $\varphi 1$	Degree	0	31.55
	Angle Change for $\varphi 1$, $\Delta\varphi 1$	Degree	0	-0.11
	Angle Change for $\varphi 2$, $\Delta\varphi 2$	Degree	0	0.11
	Cross Sectional Area of Cable of Hanger, A_h	mm^2	800	1025.8
	Arch Width, B_a	mm	600	350
	Arch Depth, H_a	mm	2800	1975
	Rise to Span Ratio, R_a	-	0.18	0.2386
Implicit Constraints	Extreme Hanger Stress, σ_{max}	MPa	823.578	1125..36
	Strength Criteria of Arch, CRT	-	1	1
	Maximum Design Reinforcement Factor in Arch, RNR	%	1.74	2.82
	End Angle for Hanger Inclination Set1, $\varphi 1_n$	Degree	0	31.55
	End Angle for Hanger Inclination Set2, $\varphi 2_n$	Degree	0	32.74
	Slenderness Ratio	-	7.095	13.01
	Arch Depth to Width Ratio		4.67	5.64



(a)



(b)

Figure 6.31: Reinforcement Percentage Envelope along the Arch of Parabolic Arch Geometry

Reinforcement percentage envelope required along the arch for all load steps is shown in Figure 6.31. Reinforcement percentage is drawn in the Figure 6.31 (a) by considering same arch section for vertical hanger and optimal hanger arrangements. Reinforcement percentage envelope for feasible design of arch section with vertical hanger arrangement is also plotted in the Figure 6.31(b). Results show that maximum amount of steel is required at quarter span for all cases. Again vertical hanger arrangement requires larger amount of steel compared with optimal hanger arrangements. Variation of reinforcement percentage envelope along the arch for vertical hanger arrangement is higher whereas for optimum design it is lesser.

By considering feasible design of arch of less than 4% of reinforcement with vertical hangers but with other optimal variables, larger arch section is required. A comparison between optimum arch section and a feasible design with vertical hanger arrangement is depicted in Figure 6.32.

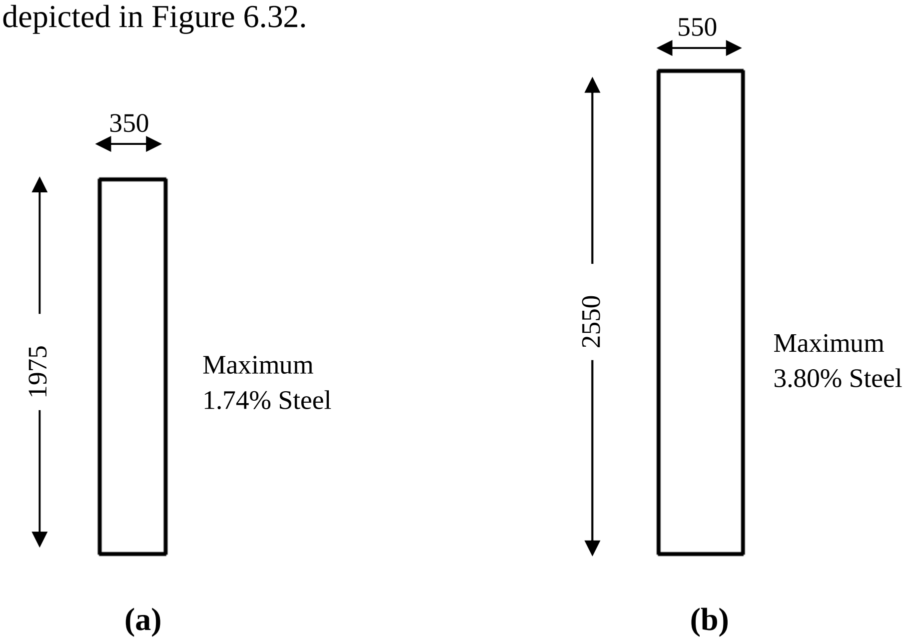


Figure 6.32: (a) Optimal Arch Section (b) Feasible Design of Arch Section of Vertical Hangers of Parabolic Arch Geometry

6.4.1 Comparison Optimum Values of Objective Function

A cost breakdown for hanger, concrete of arch and arch reinforcement is outlined in Table 6.6 for initial and final optimal design. Again cost required for optimal design and final design but with vertical hangers is compared in Table 6.7.

Table 6.6: Comparison for Cost of Initial and Optimal Design of Parabolic Arch Geometry

Cost Breakdown	Initial	Final
Cost of Cable (BDT)	0.908678753E+06	0.154834904E+07
Cost of Concrete of Arch (BDT)	0.409068897E+07	0.176910130E+07
Cost of Reinforcement in Arch (BDT)	0.269647296E+07	0.145578683E+07
Total Cost (BDT)	0.769584069E+07	0.47650856E+07
		38.1% Save

Table 6.7: Result Comparison for Cost of Optimal Design and Final Design but with Vertical Hangers of Parabolic Arch Geometry

Cost Breakdown	Final	Vertical Hangers with Final Design Variables
Cost of Cable (BDT)	0.154834904E+07	0.131112083E+07
Cost of Concrete of Arch (BDT)	0.176910130E+07	0.176890557E+07
Cost of Reinforcement in Arch (BDT)	0.145578683E+07	0.618068340E+07
Total Cost (BDT)	0.47650856E+07	0.926076980E+07
		76.44% Save of Reinforcement

Optimization approach developed in the study shows 38.1% cost can be saved for parabolic arch if design is optimized. Again it has been shown that reinforcement required in arch with optimal hanger saves 76.44 % steel than that of vertical hanger.

6.5 Comparison between Circular and Parabolic Geometry of Optimal Arch

Parabolic arch geometry is more economic than circular arch geometry considering optimum design. Total cost required for optimum design of arch is enlisted in Table 6.8 and sown in Figure 6.33.

Table 6.8: Total Cost Required for Optimal Design for Various Arch Geometry

Optimal Cost	Circular Geometry	Parabolic Geometry
Total Cost (BDT)	5.14E+06	4.77E+06

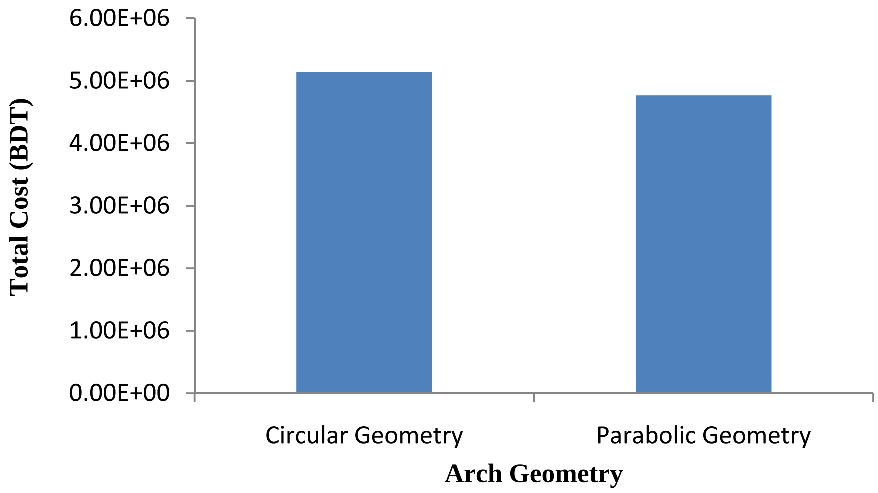


Figure 6.33: Total Cost Required for Optimal Design for Various Arch Geometry

Optimum Design parameters are enlisted in Table 6.9 for circular and parabolic geometry of arch and shown in Figures 6.34 to 6.36. It has been observed that optimum number of hangers for circular arch requires lower than that required for parabolic arch. Again hanger inclination with vertical is higher for circular arch geometry than that of parabolic arch. Arch depth required for parabolic arch is found larger than that required for circular arch. Again larger rise to span ratio is better for parabolic arch than that of circular arch.

Table 6.9: Optimum Design Variables for Various Arch Geometry

Optimal Parameters	Circular Geometry	Parabolic Geometry
Number of Hanger in Each Arch	20	22
Hanger Inclination (Degree)	39.62-36.97	32.74-31.55
Change of Angle (Degree)	0.265	0.11
Cable Area (mm^2)	1548.4	1025.8
Arch Width (mm)	450	350
Arch Depth (mm)	1600	1975
Rise to Span Ratio	0.2105	0.2386

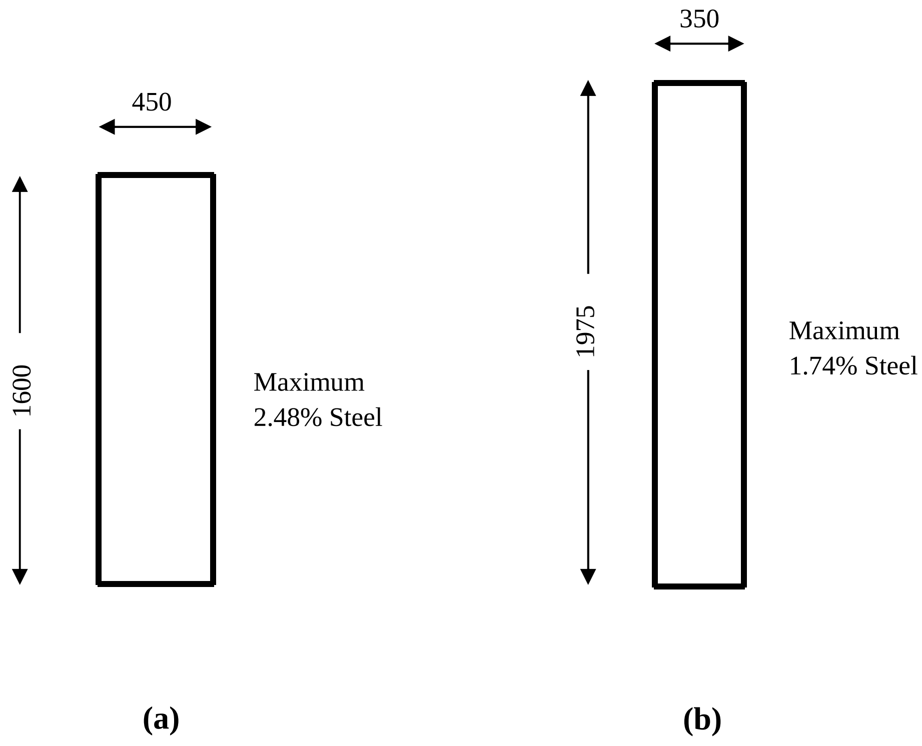


Figure 6.34: Optimal Arch Section for (a) Circular Arch (b) Parabolic Arch

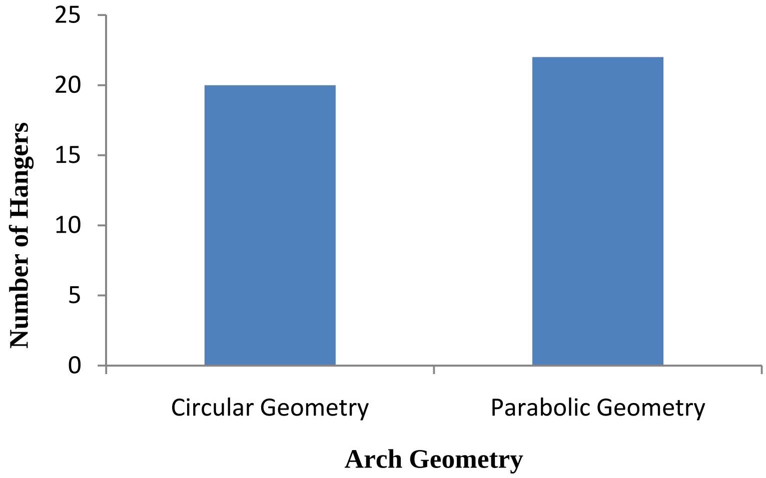


Figure 6.35: Optimal Number of Hanger for Various Arch Geometry

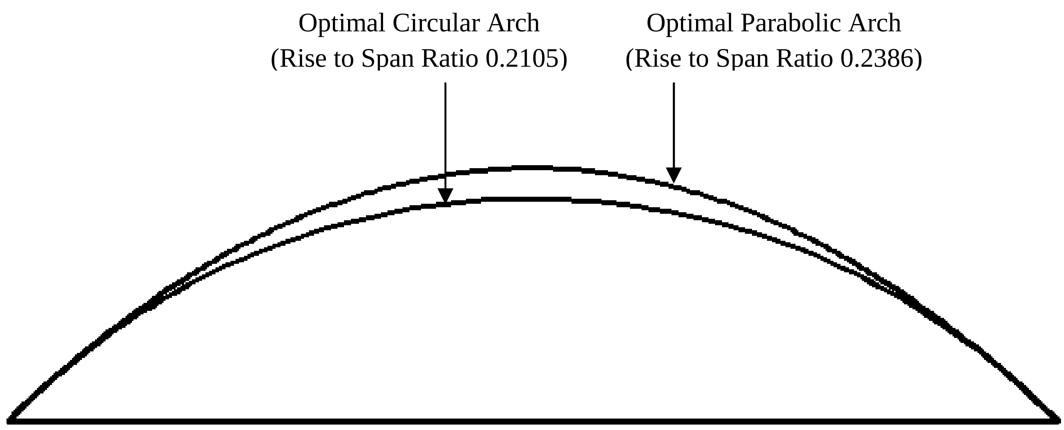


Figure 6.36: Optimum Rise to Span Ratio for Various Arch Geometry

6.6 Conclusion

Results from optimization process for various design constant parameters are described in this chapter. It has been observed that optimization approach results optimum design variables for most economic design of arch bridges of circular and parabolic arch geometry. Furthermore arch bridges with optimal design variables using global optimization technique shows significant improvement over the arch bridges designed traditionally.

CHAPTER 7

CONCLUSION

7.1 General

Simulation driven structural optimization using finite element analysis of network arch bridges is performed in the study. Optimization has been achieved through execution of a simulator, evaluation of a constrained objective function and adjustment of system parameters in an iterative and directed way. The structural analysis of virtual prototype of network arch bridges is performed by finite element simulator, ANSYS. Optimization of structural design of the bridge is performed through a global optimization algorithm EVOP. Objective function considered in the optimization problem is the total material cost required for hanger, arch concrete and arch reinforcement. A program written in Visual C++ has been developed which works as a platform for interfacing EVOP with the simulator, ANSYS. The interfacing has been verified through some benchmark problems of optimization. Finally a traditionally designed tied arch bridge has been invoked in the optimization process as the initial design and after completion of optimization process, optimal design variables i.e. hanger arrangement, no of hangers, cross section required for hanger, arch section and rise to span ratio of arch, are obtained within a range of design constant parameters. Response parameters of the arch bridge with optimum design variables are analyzed regarding bending moment, axial force, and influence line for bending moment and amount of reinforcement required in the arch. Results are compared with the initially designed arch. Arches of final design variables, but with vertical hangers, and the arches of vertical hanger arrangement, but with similar amount of steel required in arch of optimal hanger arrangement, are also studied for comparison with optimal design. Optimal design variables for circular and parabolic geometry of arch are analyzed and findings are reported. A summary of findings is outlined in the following article.

7.2 Conclusions

Based on the numerical analysis within the range of parameters examined, the following conclusions can be drawn for structural optimization of network arch bridges.

i) Network arch bridge is more economical than vertical arch bridges.

ii) It is found that optimal hanger arrangement is attained for hangers placed at some inclinations with vertical. Hanger inclination is found same to all hangers for parabolic arch geometry whereas for circular arch geometry the inclination varies from one hanger to another in a small range. Under the scope of study, hanger inclination for parabolic arch geometry is found 32 degree whereas it varies from 36 to 39 degree from first hanger to the last for circular arch geometry.

iii) Parabolic arch geometry is found more economical than circular arch geometry considering optimum design.

iv) Under the scope of study, it is observed that circular arch geometry requires lesser no of hangers than parabolic arch geometry.

v) Under the scope of study, it is found that circular arch geometry requires shallower arch than that required for parabolic arch.

7.3 Limitations and Scope of Future Study

Findings outlined in the previous article are limited for an arch bridge of fixed span and lane, and for optimum design cost of deck is not considered. Therefore it is recommended that the study can be extended further in the following fields:

i) Optimization of network arch bridges including constraints for deck section can be investigated in further studies.

ii) Optimization can be performed for different span and lane of bridge.

iii) Constraints regarding bracing spacing and arrangement for lateral load analysis may be incorporated in the optimization process.

iv) Local buckling effect in slender arch may be investigated in further studies.

v) Dynamic properties of the bridge with optimum design variables may be investigated further.

REFERENCES

[1] Pauling, L., *The Nature of the Chemical Bond*, Cornell Univ. Press, Ithaca, New York. ISBN: 0-8014-0333-2, 1960.

[2] Neumaier, A., “Global Optimization and Constraint Satisfaction”, the Computational Mathematics group. In I. Bomze, I. Emiris, Arnold Neumaier, and L. Wolsey, editors, *Proceedings of GICOLAG workshop (of the research project Global Optimization, Integrating Convexity, Optimization, Logic Programming and Algebraic Geometry)*, University of Vienna, Austria, 2006. http://www.mat.univie.ac.at/~neum/glopt.html.

[3] Wilde, D, J., *Globally Optimal Design*, John Willey & Sons, New York, USA, 1978.

[4] Nadir, W., Kim, Y. and Weck, O. L., “Structural Shape Optimization Considering Both Performance and Manufacturing Cost”, *10th AIAA/ ISSMO Multidisciplinary Analysis and optimization Conference*, Albany, New York, 2004.

[5] Haftka, R. T. and Kamat, M. P., *Elements of Structural Optimization*, Martinus Nijhoff Publishers, Dordrecht, The Netherlands, 1985.

[6] Tveit, P., “Network arches.” *11th IABSE Congress, held in Vienna*, Austria, Final Report, IABSE, ETH- Höngerberg, CH-8039, Zürich, Switserland, IBSN 3 85748 0254, pp. 817-818, 1980.

[7] Brunn, B.,Grabe W., Tveit, P. and Teich, S., *Calculation of a Double Track Railway Network Arch Bridge Applying the European Standards*, Master of Science Thesis, University of Dresden, 2003.

[8] Tveit, P., *The Network Arch, Bits of Manuscript in September 2008 after Lectures in 50 Countries*, 2008. http://pchome.grm.hia.no/~ptveit/.

[9] Ghani, S. N., “A Versatile Algorithm for Optimization of a Nonlinear Non differentiable Constrained Objective Function”, *UKAEA Harwell Report Number R-13714*, ISBN 0-7058-1566\8, HMSO Publications Centre, PO Box 276, London, SW8 5DT, 1989.

[10] Parcel, J. I. and Moorman, R. B. B. “The Elastic Arch”, *Analysis of Statically Indeterminate Structures*, John Wiley & Sons, Inc., New York, London, 1955.

[11] Fox, G.F. "Arch Bridges", *Bridge Engineering Handbook,* Ed. Wai-Fah Chen and Lian Duan, Boca Raton: CRC Press, 2000.

[12] O’Conor, C., *Design of Bridge Superstructures*, John Wiley & Sons, New York, 1971.

[13] Tveit, P., *Preliminary Design of Network Arch Road Bridges*, edition 19.12.03, Grimstad, 2003. http://pchome.grm.hia.no/~ptveit/.

[14] Herzog, M., “Steel Weights of Modern Rail- and Road-bridges”, *Der Stahlbau 9/1975*, Berlin, 1975.

[15] Weise, T., *Global Optimization Algorithms - Theory and Application*, 2nd edition, 2008. http://www.it-weise.de.

[16] Box, G., E., P., “Evolutionary Operation: A Method for Increasing Industrial Productivity”, The Imperial Chemical Industry Limited, Dyestuffs Division Headquarters, Manchester, *Applied Statistics*, 6(2):81–101, June 1957. ISSN: 00359254.

[17] Chen, S-Y., “Integrating ANSYS with Modern Numerical Optimization Technologies”, Accepted by *ANSYS Solutions Magazine*, to be published in Winter Issue of 2001, Oct 2000.

[18] Cristello, N. and Kim, I. Y., “Design Optimization of an Automotive Universal Joint Considering Manufacturing Cost”, *Modelling and Simulation,* 17th IASTED International Conference, Montreal, Canada, 2006.

[19] Roman, G., Uebersax, M. and Kaonig, O., “Structural Optimization Tool using Genetic Algorithms and Ansys”, *CAD-FEM User's Meeting Schweiz 2000,* Zurich, 2000.

[20] Hepworth, A. I., Jensen, C. G. and Roach, J. T., “Methods to Streamline Laminate Composite Analysis and Optimization”, *Computer-Aided Design and Applications*, CAD Solutions, LLC, 2010.

[21] ANSYS Inc., *Ansys Basic Analysis Procedure Guide,* Release 12.0, Canonsburg, PA, USA, 2009.

[22] Gallagher, R. H. and Zienkiewicz, O. C., *Optimum Structural Design : Theory and Application***,** John Wiley & Sons Ltd, 1973.

[23] Pahl, G. and Beitz, W., *Engineering Design*. Springer-Verlag, The Design Council, Berlin, London, 1984.

[24] Beazley, P. K., *Ansys Revision 4.3 Tutorial - Design Optimization*, Swanson Analysis, Inc., Houston, Pa., 1987.

[25] Kiemele, M. J., “How to Improve the Effectiveness, Efficiency and Integration of Test & Evaluation (T&E) and Modeling and Simulation (M&S)”, *NDIA Conference on T&E and M&S,* Jacksonville, FL, 2006.

[26] AASHTO, “American Association of State Highway and Transportation Officials”, *Standard Specifications for Highway Bridges*, 17th edition, Washington, DC, 2002.

[27] AISC, “American Institute of Steel Construction, Inc”, *Specification for Structural Steel Buildings,* ANSI/AISC 360-05, March 9, 2005.

[28] RHD, *Schedule of Rates,* Roads and Highway Department, Dhaka, Bangladesh, 2008.

[29] ASTM A 586, “Standard Specification for Zinc-Coated Parallel and Helical Steel Wire Structural Strand and Zinc Coated Wire for Spun-In-Place Structural Strand”, American Society for Testing and Materials, *Annual book of ASTM Standards*, Volume 01.06, West Conshohocken, PA, 19428, 1998.

[30] Nettleton, D. A. and Torkelson, J. S., *Arch Bridges*, Bridge Division, Office of Engineering, Federal Highway Administration, U. S. Department of Transportation, Washington, D. C., 1977.

[31] CSI, *CSI Concrete Frame Design Manual ACI 318-02/ IBC 2003 for SAP 2000*, Computers and Structures, Inc., Berkeley, California, 2008.

[32] ANSYS CivilFEM Inc., *CivilFEM Theory Manual 300609,* Release 12.0, Ingeciber, S. A., 2009.

[33] ACI, “Building Code Requirements for Structural Concrete (ACI 318-02) and Commentary”, (ACI 318R-02), *American Concrete Institute*, P.O. Box 9094, Farmington Hills, Michigan, 2002.

[34] DAKOTA, *DAKOTA User’s Manual*, Version 4.2, Sandia National Laboratories, Livermore, CA, 2008.

[35] ANSYS Inc., *Verification Manual of Ansys*, Release 12.0, Canonsburg, PA, USA, 2009.

[36] Prasad B., Haftka R. T., "Optimal Structural Design with Plate Finite Elements", *ASCE, J. Structural Div.*, Vol. 105, No. ST11, 1979.

[37] LCB Public Meeting, *Lake Champlain Bridge Project*, Public Meeting on New Bridge Design Concepts and Commemorating the old Bridge, New York, 2009.

[38] Coates, R. C., Coutie M. G., Kong F. K., "*Basic Structural Concepts*", 2nd edition, English Language Book Society, Van Nostrand Reinhold, UK, 1986.

[39] Aswani, M. G., Vazirani V. N., Ratwani M. M., *Design of Concrete Bridges*, 2nd edition, Khanna Publishers, New Delhi, India, 1996.

[40] Salvadori M., Levy M., *Structural Design in Architecture*, Prentice Hall, Inc., Englewood Cliffs, N. J., 1967.

[41] Rana, S, *Cost Optimization of Post-Tensioned Prestressed Concrete I-Girder Bridge System*, M. Sc. Engg. Thesis, Department of Civil Engineering, Bangladesh University of Engineering and Technology, 2010.

APPENDIX A

OPTIMIZATION PROCESS OF EVOP

The optimization method, EVOP developed by Ghani [9] is subdivided into six fundamental processes which are as follows.

1. Generation of a 'Complex'
2. Selection of a 'Complex' vertex for penalization
3. Testing for collapse of a 'Complex'
4. Dealing with a collapsed 'Complex'
5. Movement of a 'Complex' and
6. Convergence tests.

Flow chart of these fundamental processes described by Ghani [9] which is further formulated by Rana [41] is shown in Figure A.1.

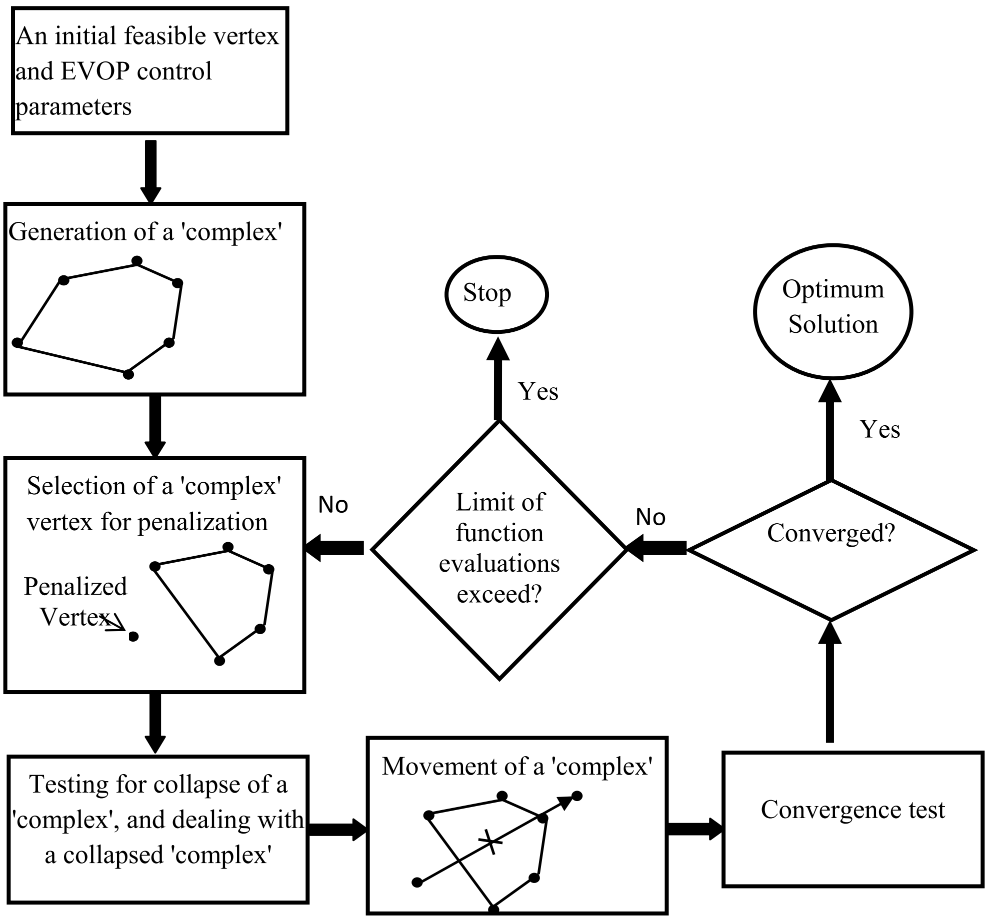


Figure A.1: General Outline of EVOP Algorithm [41]

Ghani described the 'Complex' as a 'living' object spanning an n-dimensional space defined by $k \geq (n+1)$ vertices inside the feasible region. It has the intelligence to move towards a minimum located on the boundary or inside the allowed space. It can rapidly change its shape and size for negotiating difficult terrain. Figure A.2 shows a 'Complex' with four vertices in a two dimensional parameter space. The 'Complex' vertices are identified by lower case letters 'a', 'b', 'c' and 'd' in an ascending order of function values, i.e. f(a)<f(b)<f(c)<f(d). Straight line parallel to the co-ordinate axes are explicit constraints with fixed upper and lower limits. The curved lines represent implicit constraints set to either upper or lower limits. The hatched area is the two dimensional feasible search spaces.

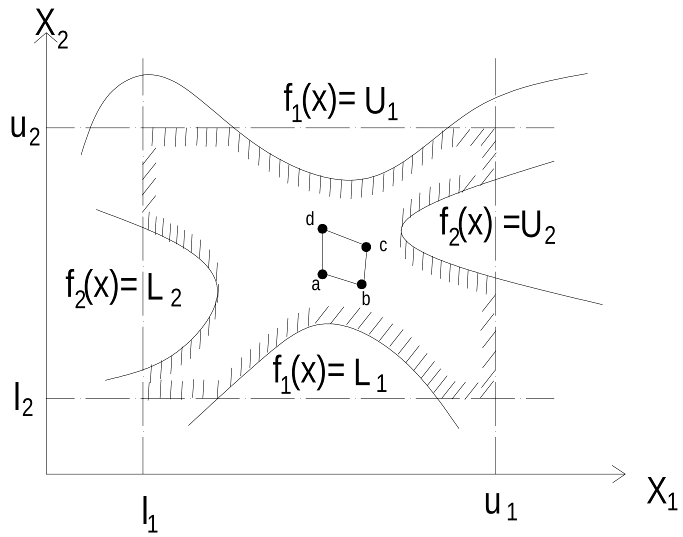


Figure A.2: “Complex" with Four Vertices Inside a Two Dimensional Feasible Search Space [9]

A.1 Generation of a 'Complex'

For generation of a ‘Complex’, a starting point is required that satisfies all explicit and implicit constraint sets. A second point is randomly generated within the bounds defined by the explicit constraints. If this second point also happens to satisfy all implicit constraints, the algorithm proceeds to the next step. Centroid, ***C*** of the two feasible points is determined. If it satisfies all constraints, then the ‘Complex’ is updated with this second point. If, however, the randomly generated point fails to satisfy implicit constraints it is continually moved half way towards the feasible starting point till all constraints are satisfied. Feasibility of the centroid of the two points is next checked. If the centroid satisfies all constraints, then the ‘Complex’ is accepted, and the third point is generated for the ‘Complex’. If, however, the centroid fails to satisfy any of the constraints this second point is randomly once again generated in the space defined by the explicit constraints. Initial complex generation in evolutionary operation process is depicted in Figure A.3 where a is the feasible starting point, $d^{/}$ is the first random point which moves toward centroid a at d and $c^{/}$ and $b^{/}$ are random points which move to c and b for satisfying violated implicit constraints.

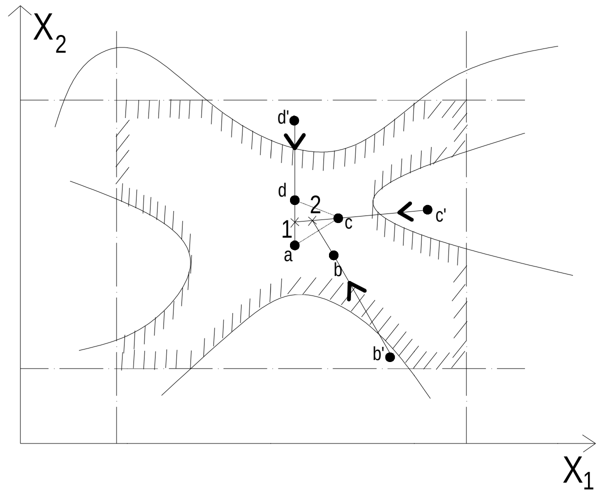


Figure A.3: Generation of Initial "Complex" [9]

For a convex feasible parameter space the above method would, without fail, succeed in generating a 'Complex' with k vertices. If the parameter space is non-convex and the centroid happens to lie in the infeasible area, there is every chance that a 'Complex' cannot be generated. Figure A.4 shows inappropriate Complex generation. Here X is the centroid which lies in the infeasible area and the parameter space abc is non-convex.

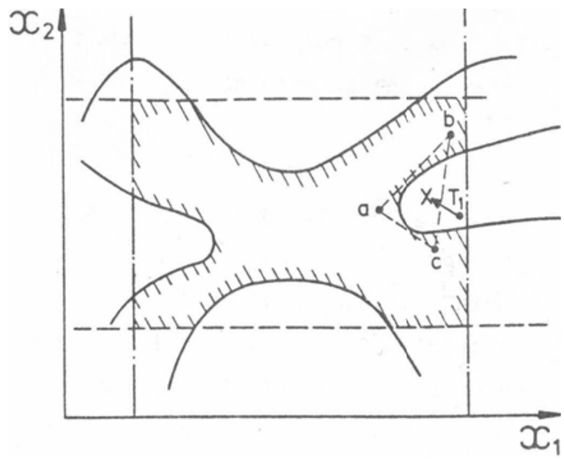


Figure A.4: No Generation of 'Complex' with Four Vertices [9]

In such case if a new feasible 'Complex' vertex results in the new centroid to lie in the infeasible area, that new vertex is discarded and another is generated until a feasible centroid is obtained.

A.2 Selection of a 'Complex' Vertex for Penalization

The 'Complex' vertices having highest, second highest and lowest function values are identified, and labelled as 'ng', 'nh' and 'ns' respectively. The centroid of all vertices except 'ng' is calculated. For penalization of a Complex vertex, the worst vertex of a 'Complex' with the highest function value is penalized by over-reflecting on the centroid. Again if the centroid itself is in the infeasible region, repeated movement of point 'T_l' halfway towards the centroid would result in collapse of the point on the centroid which is shown in Figure A.5.

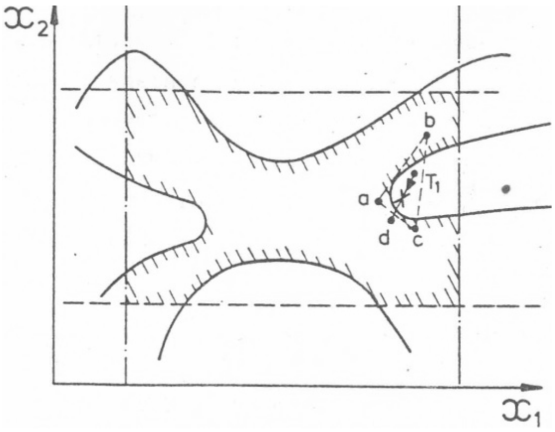


Figure A.5: Possibility of Collapse of a Trial Point onto the Centroid [9]

One more such collapse would result in complete collapse of the 'Complex' because an object with two vertices cannot span a two dimensional space.

A.3 Testing for Collapse of a 'Complex'

A 'Complex' is said to have collapsed in a subspace if the i^{th} coordinate of the centroid is identical to the same of all 'k' vertices of the 'Complex'. This is a sufficiency condition and detects collapse of a 'Complex' when it lies parallel along a coordinate axis as shown in Figure A.6.

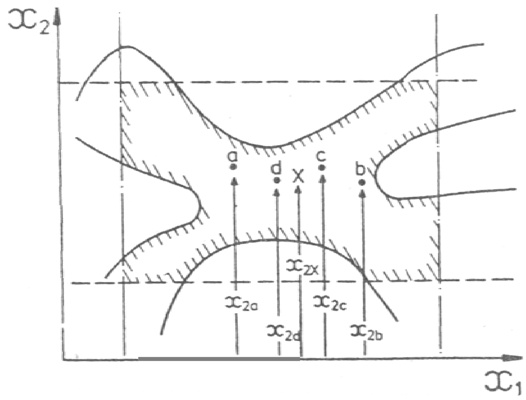


Figure A.6: Collapse of a 'Complex' to a One Dimensional Subspace [9]

A.4 Dealing with a Collapsed 'Complex'

On detecting collapse of a 'Complex' some actions are taken such that a new 'Complex' is generated within the full feasible space defined by the explicit and implicit constraints or a 'Complex' spanning smaller feasible space. The process for movement of a 'Complex' as explained below is continued.

A.5 Movement of a 'Complex'

The process begins by over-reflecting the worst vertex 'n_g' of a 'Complex' on the feasible centroid 'C' of the remaining vertices to generate a new trial point, x_r.

$$\boldsymbol{x_r} = (1+\alpha)\boldsymbol{C} - \boldsymbol{\alpha x_g} \tag{4.1}$$

where α is reflection coefficient and x_g is location of vertex n_g.

A check is then made to determine whether the trial point violates any constraints. If an explicit constraint is violated, the trial point is moved just inside the boundary by a small amount Δ called the explicit constraint retention coefficient. If any implicit constraint is violated the trial point is repeatedly moved halfway towards the centroid until the constraint is satisfied. Figure A.7 shows the reflected point T_1 which violates an explicit constraint, moved just inside the boundary by a retention coefficient to feasible point T_2. Again Figure A.8 shows that infeasible trial point T_1 which violates implicit constraint is moved to feasible T_2.

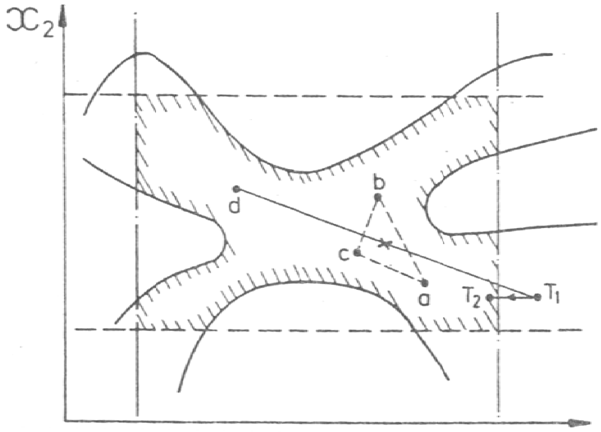


Figure A.7 Reflected Point Violating an Explicit Constraint [9]

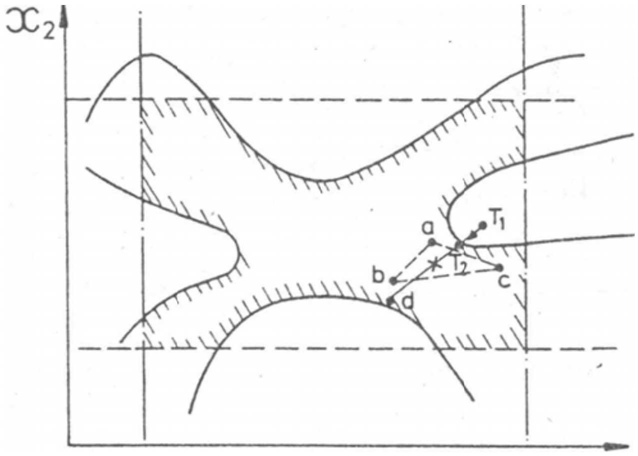


Figure A.8 Reflected Trial Point Violating an Implicit Constraint [9]

The function value at the feasible trial point is then evaluated. The reflection step is considered successful if the function value at this new trial point is lower than that at vertex 'nh', and the vertex 'ng' is replaced by the trial point. If, however, the function value at the trial point is greater than that at vertex of the current 'Complex', the trial point would be the worst vertex in the new 'Complex' configuration. The reflection step is, therefore, considered unsuccessful and contraction step applied as described by Ghani [9].

Depending on situation anyone of the three stages (Stages 1-3) of the contraction step can be called. If the function value at the feasible trial point after over-reflection is less than that at vertex 'ng' but equal to or greater than that at vertex 'nh', Stage 1 of contraction step is applied. This is essentially an under-reflection, and the co-ordinates of the new trial point **x** is given by

$$\boldsymbol{x} = (1+\beta)\boldsymbol{C} - \beta\boldsymbol{x_g} \tag{4.2}$$

where β is contraction coefficient.

Stage 2 of contraction step is applied if at the end of over-reflection the function value at the feasible trial point is equal to or greater than that at the worst vertex 'ng' of the current Complex. The co-ordinates of the new trial point are given by:

$$\boldsymbol{x} = \beta\boldsymbol{x_g} + (1-\beta)\boldsymbol{C} \tag{4.3}$$

Stage 3 of contraction step is called only after Stages 1 and/or 2 have been previously applied consecutively for more than '2k' times. A small 'Complex' is generated using vertex 'ns' as the starting point. If on over-reflection the trial point has not violated any constraints, has a function value lower than the lowest function value at vertex 'ns' of the current 'Complex', and the previous move was not a contraction step, this over-reflection is considered over-successful. Expansion step is then applied to generate a new trial point further away from the feasible centroid along the same straight line used for over-reflection. The co-ordinates of this accelerated trial point is given by

$$\boldsymbol{x} = \gamma\boldsymbol{x_r} + (1-\gamma)\boldsymbol{C} \tag{4.4}$$

Where, γ is expansion coefficient.

Feasibility of the accelerated trial point is next checked. If any constraint is violated, the acceleration step is considered unsuccessful, and a new 'Complex' is formed with the over-reflected feasible trial point as the updated vertex replacing the worst vertex of the current 'Complex'. Otherwise the function value at the feasible accelerated trial point is evaluated. If it is lower than that at x the acceleration step is considered successful. The accelerated point then replaces the worst vertex to form the new 'Complex'. Else, if the function value at the accelerated point equals or exceeds that at the over-reflected point the acceleration step is also considered unsuccessful. The accelerated point is rejected in favor of the over-reflected point to form the new 'Complex'.

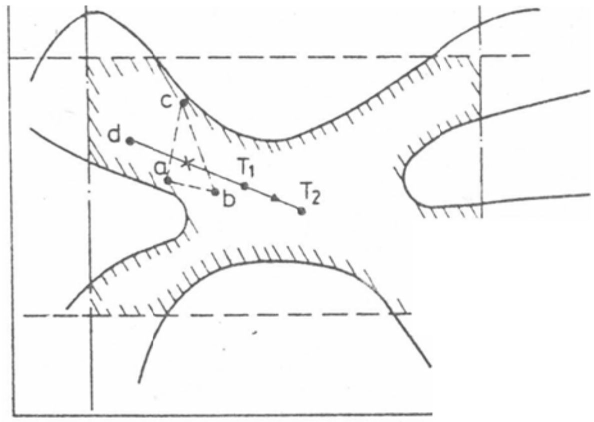


Figure A.9 Successful Acceleration Steps [9]

Figure A.9 shows a successful acceleration step. The over-reflected trial point 'T_1' does not violate any constraint, and has a function value lower than the lowest function value at vertex 'a' of the current 'Complex' 'abcd'. Since contraction step was not applied previously acceleration step is called to obtain the trial point 'T_2'. 'T_2' does not violate any constraint and yet has a function value lower than that at 'T_1 '. Trial point 'T_2' replaces the worst vertex 'd' of the current 'Complex' to form the updated 'Complex' 'abcT_2'.

A.6 Convergence tests

While executing the process of movement of a ‘Complex’, test for convergence are made periodically after certain preset number of calls to the objective function. There are two levels of convergence tests. The first convergence test would succeed only if a predefined number of consecutive function values are identical within the resolution of convergence parameter Φ, which should be finer than another parameter, Φ_{cpx}. Φ_{cpx} is a parameter for determining collapse of a ‘Complex’ in a subspace The second convergence test is attempted only if the first convergence test succeeds. This second test for convergence verifies whether function values at all vertices of the current 'Complex' are also identical within the resolution of Φ.

APPENDIX B

COMPUTER PROGRAM OF INTERFACING PLATFORM

```
!--------------------------------------------------------------------------
!Author                                                 Nazrul Islam
!                                                       Lecturer
!                                                       Dept. of Civil Engineering
!                                                       BUET, Dhaka 1000.
! -------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!**************************************************************************
!The following script is the Interfacing Platform of Optimization written
!in Visual C++ for data input output operation between ANSYS and EVOP.
!File Name is                    arch11.cpp
!**********************************BEGIN***********************************

#include <iostream>
#include <fstream>
#include <stdio.h>
#include <cmath>
#include <string>
#include <windows.h>
#include<cstdlib>
#define MAX_CHAR 1024

using namespace std;
extern "C"
{
        void __stdcall
EVOP(double*,double*,double*,double*,double*,double*,double*,

int*,int*,int*,int*,int*,int*,int*,int*,int*,int*,double*,double*,double*,

double*,double*,double*,double*,double*,double*,double*,double*,double*);
        void __stdcall DINTG2(int*,double*,double*,double*,double*);

        void __stdcall
DISCR2(double*,int*,int*,double*,double*,double*,double*);
        void __stdcall EXPCON(int*,int*,int*,int*,double*,double*,double*);
        void __stdcall FUNC(double*,int*,int*,int*,double*);
        double __stdcall RNDOFF(double*);

        void __stdcall IMPCON(int*,int*,double*,double*,double*,double*);
}

double cost_cable, cost_arch, cost_reinforce, cost_total, Hsmax, Hsmin,
ARCH_INTERACT_MAX_lcase_i, ARCH_REINFACT_MAX_lcase_i, SLENDER_FACTOR;

double DX0[28],DX5[35],DX6[95],DX7[115];


void __stdcall EXPCON(int *IFLG,int *ISKP,int *KKT,int *KOUNT,double
XMAX[9],double XMIN[9],double XT[9])
{
        double STRIP;
        int x0 = 24,x5 = 30,x6=91 ,x7=111 ;
```

```
        *KOUNT = *KOUNT+1;
        *KKT = *KKT+1;
        XMIN[0]=3.9888;
        XMIN[1]=0;
        XMIN[2]=0;
        XMIN[3]=-1;
        XMIN[4]=-1;
        XMIN[5]=96.77;
        XMIN[6]=249.9;
        XMIN[7]=249.9;
        XMIN[8]=0.14;

        XMAX[0]=50.1;
        XMAX[1]=80;
        XMAX[2]=80;
        XMAX[3]=1;
        XMAX[4]=1;
        XMAX[5]=2071.1;
        XMAX[6]=2500.1;
        XMAX[7]=3000.1;
        XMAX[8]=0.25;

        if(*IFLG == 0)
      {
            STRIP=1e-2;

            DISCR2(DX0,ISKP,&x0,&STRIP,&XT[0],&XMAX[0],&XMIN[0]);
            DISCR2(DX5,ISKP,&x5,&STRIP,&XT[5],&XMAX[5],&XMIN[5]);
            DISCR2(DX6,ISKP,&x6,&STRIP,&XT[6],&XMAX[6],&XMIN[6]);
            DISCR2(DX7,ISKP,&x7,&STRIP,&XT[7],&XMAX[7],&XMIN[7]);
      }
}

void __stdcall FUNC(double *F,int *KOUNT,int *KUT,int *N,double XT[9])
{
double N_hanger, theta_set1, theta_set2, delta_ANSYS_set1,
delta_ANSYS_set2, cable_area_astm_A586, arch_width, arch_depth,
rise_span_ratio;

      *KOUNT=*KOUNT+1;
      *KUT=*KUT+1;

      N_hanger  = XT[0];
      theta_set1 = XT[1];
      theta_set2 = XT[2];
      delta_ANSYS_set1 = XT[3];
      delta_ANSYS_set2 = XT[4];
      cable_area_astm_A586 = XT[5];
      arch_width = XT[6];
      arch_depth = XT[7];
      rise_span_ratio = XT[8];

      ofstream fout("ansys_func.txt");

      fout<<"N_hanger = "<<N_hanger<<endl;
      fout<<"theta_set1 = "<<theta_set1<<endl;
      fout<<"theta_set2 = "<<theta_set2<<endl;
      fout<<"delta_ANSYS_set1 = "<<delta_ANSYS_set1<<endl;
      fout<<"delta_ANSYS_set2 = "<<delta_ANSYS_set2<<endl;
      fout<<"cable_area_astm_A586 = "<<cable_area_astm_A586<<endl;
      fout<<"arch_width = "<<arch_width<<endl;
```

```
      fout<<"arch_depth = "<<arch_depth<<endl;
      fout<<"rise_span_ratio = "<<rise_span_ratio<<endl;

      fout.close();
      system ("batch_ansys_func.bat");
//    Sleep(10000);
        ifstream fin("func.txt");
        fin>>cost_cable;
      fin>>cost_arch;
      fin>>cost_reinforce;
      fin>>cost_total;
      fin.close();

      cout<<""<<endl;
      cout<<"******ANSYS**************OUTPUT**************EVOP*******"<<e
ndl;
      cout<<""<<endl;
      cout<<"N_hanger = "<<RNDOFF(&N_hanger)*2<<endl;
//    cout<<"N_hanger = "<<N_hanger<<endl;
      cout<<"theta_set1 = "<<theta_set1<<endl;
      cout<<"theta_set2 = "<<theta_set2<<endl;
      cout<<"delta_ANSYS_set1 = "<<delta_ANSYS_set1<<endl;
      cout<<"delta_ANSYS_set2 = "<<delta_ANSYS_set2<<endl;
      cout<<"cable_area_astm_A586 = "<<cable_area_astm_A586<<endl;
      cout<<"arch_width = "<<arch_width<<endl;
      cout<<"arch_depth = "<<arch_depth<<endl;
      cout<<"rise_span_ratio = "<<rise_span_ratio<<endl;

      cout<<"                          "<<endl;

      cout<<"Cost of Cable = "<<cost_cable<<endl;
      cout<<"Cost of Arch = "<<cost_arch<<endl;
      cout<<"Cost of Reinforcement = "<<cost_reinforce<<endl;
      cout<<"Total Cost = "<<cost_total<<endl;
      cout<<""<<endl;
cout<<"******ANSYS**************OUTPUT**************EVOP*******"<<endl;
      *F=cost_cable + cost_arch + cost_reinforce;
 }

void __stdcall IMPCON(int *KOUNT,int *M,double XT[9],double XX[6],double
XXMAX[6],double XXMIN[6])
{
      double N_hanger, theta_set1, theta_set2, delta_ANSYS_set1,
delta_ANSYS_set2, cable_area_astm_A586, arch_width, arch_depth,
rise_span_ratio;

      double Fu_cable,Fy_con;

      Fu_cable=1520E6;
      Fy_con=24.137E6;

      *KOUNT = *KOUNT + 1;
      *M = *M + 1;
      N_hanger  = XT[0];
      theta_set1 = XT[1];
      theta_set2 = XT[2];
      delta_ANSYS_set1 = XT[3];
      delta_ANSYS_set2 = XT[4];
      cable_area_astm_A586 = XT[5];
      arch_width = XT[6];
      arch_depth = XT[7];
```

```
	rise_span_ratio = XT[6];

	ofstream fout("ansys_imp.txt");

	fout<<"N_hanger = "<<N_hanger<<endl;
	fout<<"theta_set1 = "<<theta_set1<<endl;
	fout<<"theta_set2 = "<<theta_set2<<endl;
	fout<<"delta_ANSYS_set1 = "<<delta_ANSYS_set1<<endl;
	fout<<"delta_ANSYS_set2 = "<<delta_ANSYS_set2<<endl;
	fout<<"cable_area_astm_A586 = "<<cable_area_astm_A586<<endl;
	fout<<"arch_width = "<<arch_width<<endl;
	fout<<"arch_depth = "<<arch_depth<<endl;
	fout<<"rise_span_ratio = "<<rise_span_ratio<<endl;

	fout.close();
	system ("batch_ansys_imp.bat");
//	Sleep(10000);
	ifstream fin("imp.txt");
	fin>>Hsmax;
	fin>>Hsmin;
	fin>>ARCH_INTERACT_MAX_lcase_i;
	fin>>ARCH_REINFACT_MAX_lcase_i;
	fin>>SLENDER_FACTOR;
	fin.close();

	cout<<""<<endl;

cout<<"******ANSYS**************OUTPUT**************EVOP*******"<<endl;
	cout<<""<<endl;
	cout<<"N_hanger = "<<RNDOFF(&N_hanger)*2<<endl;
///	cout<<"N_hanger = "<<N_hanger<<endl;
	cout<<"theta_set1 = "<<theta_set1<<endl;
	cout<<"theta_set2 = "<<theta_set2<<endl;
	cout<<"delta_ANSYS_set1 = "<<delta_ANSYS_set1<<endl;
	cout<<"delta_ANSYS_set2 = "<<delta_ANSYS_set2<<endl;
	cout<<"cable_area_astm_A586 = "<<cable_area_astm_A586<<endl;
	cout<<"arch_width = "<<arch_width<<endl;
	cout<<"arch_depth = "<<arch_depth<<endl;
	cout<<"rise_span_ratio = "<<rise_span_ratio<<endl;

	cout<<"                          "<<endl;

	cout<<"Maximum Hanger Stress = "<<Hsmax<<endl;
	cout<<"Minimum Hanger Stress = "<<Hsmin<<endl;
	cout<<"Extreme Capacity Ratio of Arch=
"<<ARCH_INTERACT_MAX_lcase_i<<endl;
	cout<<"Maximum Percentage of Steel=
"<<ARCH_REINFACT_MAX_lcase_i<<endl;
	cout<<"Slenderness Ratio= "<<SLENDER_FACTOR<<endl;
	cout<<"Theta End Set1= "<<theta_set1 +
N_hanger*delta_ANSYS_set1<<endl;
	cout<<"Theta End Set2= "<<theta_set2 +
N_hanger*delta_ANSYS_set2<<endl;
	cout<<""<<endl;
cout<<"******ANSYS**************OUTPUT**************EVOP*******"<<endl;

	XX[0]=  Hsmax;
	XXMAX[0]= 0.75*Fu_cable;
	XXMIN[0]= 0;

	XX[1]=  theta_set1 + N_hanger*delta_ANSYS_set1;
```

```
        XXMAX[1]= 80;
        XXMIN[1]= 0;

        XX[2]=  theta_set2 + N_hanger*delta_ANSYS_set2;
        XXMAX[2]= 80;
        XXMIN[2]= 0;

        XX[3]= ARCH_INTERACT_MAX_lcase_i;
        XXMAX[3]= 1.1;
        XXMIN[3]= 0;

        XX[4]= ARCH_REINFACT_MAX_lcase_i;
        XXMAX[4]= 4.1;
        XXMIN[4]= 0.999;

        XX[5]=  SLENDER_FACTOR;
        XXMAX[5]= 22;
        XXMIN[5]= 0.1;
}

void main()
{
double theta_set1, theta_set2, delta_ANSYS_set1, delta_ANSYS_set2,
cable_area_astm_A586, arch_width, arch_depth, rise_span_ratio;
double ALPHA,BETA,DEL,GAMA,PHI,PHICPX;
int ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC;
double N_hanger;

            N_hanger = 26;
            theta_set1 = 0;
            theta_set2 = 0;
            delta_ANSYS_set1 = -0.1;
            delta_ANSYS_set2 = 0.1;
            cable_area_astm_A586 = 1025.8;
            arch_width  = 1000;
            arch_depth  = 2000;
            rise_span_ratio  = 0.18;

double C[9], FF[10], H[90], OLDCC[9], XDN[9], XG[9], XMAX[9], XMIN[9],
XUP[9], XX[6], XXMAX[6], XXMIN[6], XT[9];

            XT[0]= N_hanger;
            XT[1]= theta_set1;
            XT[2]= theta_set2;
            XT[3]= delta_ANSYS_set1;
            XT[4]= delta_ANSYS_set2;
            XT[5]= cable_area_astm_A586;
            XT[6]= arch_width;
            XT[7]= arch_depth;
            XT[8]= rise_span_ratio;

            for(int i=0;i<=23;i++)
            {
                  DX0[i]=4 + 2*i;
            }

            DX5[0] = 96.8;
            DX5[1] = 122.6;
            DX5[2] = 151;
            DX5[3] = 183.2;
            DX5[4] = 218.1;
```

```
            DX5[5] = 255.5;
            DX5[6] = 296.1;
            DX5[7] = 340;
            DX5[8] = 387.1;
            DX5[9] = 436.8;
            DX5[10] = 489.7;
            DX5[11] = 545.8;
            DX5[12] = 605.6;
            DX5[13] = 664.5;
            DX5[14] = 729;
            DX5[15] = 800;
            DX5[16] = 871;
            DX5[17] = 948.4;
            DX5[18] = 1025.8;
            DX5[19] = 1103.2;
            DX5[20] = 1187.1;
            DX5[21] = 1271;
            DX5[22] = 1361.3;
            DX5[23] = 1451.6;
            DX5[24] = 1548.4;
            DX5[25] = 1645.2;
            DX5[26] = 1748.4;
            DX5[27] = 1851.6;
            DX5[28] = 1961.3;
            DX5[29] = 2071;
            //DX5[30] = 2180.7;
            //DX5[31] = 2303.2;
            //DX5[32] = 2419.4;
            //DX5[33] = 2541.9;
            //DX5[34] = 2664.5;
            //DX5[35] = 2793.6;
            //DX5[36] = 2929;


            for(int j=0;j<=90;j++)
            {
                  DX6[j]=250 + 25*j;
            }

            for(int k=0;k<=110;k++)
            {
                  DX7[k]=250 + 25*k;
            }



//************************************************************************

//  CONTROL PARAMETERS FOR "EVOP"
            ALPHA=1.2;
            BETA=0.5;
            GAMA=2.0;
            DEL=1e-12;
            PHI=1e-14;
            PHICPX=1e-16;
            ICON=5;
            LIMIT=20000;
            KNT=25;
            N=9;
            NIC=6;
            K=10;
```

```
            IPRINT=2;
            NRSTRT=2;
            IMV=0;
            IJK=1;

line1:
EVOP(&ALPHA,&BETA,C,&DEL,FF,&GAMA,H,&ICON,&IJK,&IMV,&IPRINT,&K,&KNT,&LIMIT,
&N,&NRSTRT,

&NIC,OLDCC,&PHI,&PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,XX,XXMAX,XXMIN);
            if (IJK < 9)
            goto line1;
}
```

APPENDIX C

Finite Element Simulation Script

```
!-------------------------------------------------------------------------
!Author                                              Nazrul Islam
!                                                    Lecturer
!                                                    Dept. of Civil Engineering
!                                                    BUET, Dhaka 1000.
! ------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!************************************************************************
!The following script is finite element simulation file for analysis and
!design of network arch bridges which is used by objective function
!evaluation of global optimization algorithm EVOP.
!File Name is                 arch_func_final.txt
!*********************************BEGIN**********************************

/BATCH
FINISH
/CLEAR
!/TRIAD,OFF
!/PLOPTS,LOGO,0
/COM,ANSYS RELEASE 12.0
~CFACTIV,CIVILFEM,Y
~UNITS,SI
~CFACTIV,NLBR,Y
~CFACTIV,PRSC,Y
~CODESEL,LRFD13,AASHTOHB,ACI,,AASHTO
~CFCONFG,RCV,RESMAX,100000

/PREP7

!~CFSAVE
!/NOPR
!PARSAV,ALL,_CFTEMP.DAT

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ANSYS_FUNC.TXT
/INPUT,'E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_MODEL_FINAL.MAC'
/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_PARAM.TXT
/INPUT,'E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_BRIDGE_RADIAL_FINAL.MAC
'
!PARRES,NEW,_CFTEMP.DAT
!/GO

FINISH

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\INFLUENCE_LINE_FUNC.TXT

/SOLU
ALLSEL,ALL
CSYS,0

!************************************************************************

ANTYPE,STATIC
!EQSLV,PCG,1E-8
!SSTIF,ON
!NSUBST,3
```

```
!OUTPR,ALL,ALL

!**********************************************************************

!! VEHICLE LOAD

~CFLSWRT,INIT
/TITLE,VEHICLE LOAD
~BLVLIB,1,AASHTO,HS20,1
~BLVMDF,1,NCOL,4
~BLVMDF,1,COLSP,4.267,1,2
~BLVMDF,1,COLSP,4.267,2,3
~BLVMDF,1,COLSP,10,3,4
~BLVMDF,1,LOC,4.267,1
!~BLVMDF,1,LOADS,0,1,4
!~BLVMDF,1,LOADS,0,2,4

!! FAMILY DEFINITION
~BLFDF,1001,SELECT,1,0,2
!! ASIGNS VEHICLE 1 TO FAMILY 1001
~BLVR,1001,,1,,,-2.0,,,0,5
~BLVR,1001,,1,,,2.0,,,0,5
!~BLVR,1001,,1,,,,,,0,5

~BLSOLVE
FDELE,ALL,ALL ! DELETE LOADS

*GET,LSTEP_LAST_1,ACTIVE,0,SET,LSTP

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\LANE_LOAD_SOLU.TXT

ALLSEL,ALL
!ESEL,U,MAT,,4
! SELF WEIGHT
/TITLE,SELF WEIGHT
!TIME,21
 ACEL,,,9.81
!~CFLSWRT,,LSTEP_LAST_1+19
~CFLSWRT
ACEL,

ALLSEL,ALL
! DEAD LOAD
/TITLE,WEARING SURFACE
!TIME,22
!SFBEAM,ALL,2,PRES,33200
ESEL,S,ENAME,,154
WEARING=1436      !WEARING = 30 PSF
SFE,ALL,1,PRES,0,WEARING
ALLSEL,ALL
!~CFLSWRT,LSTEP_LAST_1+20
~CFLSWRT
SFEDELE,ALL,ALL,ALL
LSTEP_LAST_LANE=LSTEP_LAST_1+5*2*1+2

ALLSEL,ALL
!LSSOLVE,LSTEP_LAST+1,LSTEP_LAST+4,1
~CFLSSLV,LSTEP_LAST_1+1,LSTEP_LAST_LANE,1
!SOLVE
FINISH
```

```
/POST1

RSYS,SOLU

LSTEP_LAST=LSTEP_LAST_1+5*2*1
DL_FACT=1.3
LL_FACT_TRUCK=2.17*(1+50/(125+3.28*SPAN_ARCH))
LL_FACT_LANE=2.17*(1+50/(125+3.28*SPAN_ARCH))

*DO,ICASE_1,1,LSTEP_LAST
LCDEF,(ICASE_1-1)+2,ICASE_1,1

*IF,ICASE_1,LE,LSTEP_LAST_1,THEN
LCFACT,(ICASE_1-1)+2,LL_FACT_TRUCK
*ELSEIF,ICASE_1,GT,LSTEP_LAST_1
LCFACT,(ICASE_1-1)+2,LL_FACT_LANE
*ENDIF
*ENDDO

*DO,ICASE_2,LSTEP_LAST+1,LSTEP_LAST+2
LCDEF,ICASE_2+1,ICASE_2,1
LCFACT,ICASE_2+1,DL_FACT
*ENDDO

!/TITLE,1.30 DC
LCASE,LSTEP_LAST+1+1
LCOPER,ADD,LSTEP_LAST+2+1
/TITLE,1.30 DC
RAPPND,LSTEP_LAST+3+1,26
SET,LSTEP_LAST+3+1
~MOD_SF
!LCDEF,LSTEP_LAST+3,LSTEP_LAST+3,1
LCDEF,1,LSTEP_LAST+3+1,1

*GET,LSTEP_INI_1,ACTIVE,0,SET,LSTP

!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
*DO,ICASE_3,2,LSTEP_LAST+1
LCASE,ICASE_3
!LCOPER,ADD,LSTEP_LAST+3
LCOPER,ADD,1
/TITLE,1.30 (DC+DW) + 2.17 (L+I)
RAPPND,LSTEP_LAST+3+1+ICASE_3-1,26
SET,LSTEP_LAST+3+1+ICASE_3-1
~MOD_SF
*ENDDO
*GET,LSTEP_FIN_1,ACTIVE,0,SET,LSTP


!*********************************2ND LANE********************************

!*GET,LSTEP_SLANE,ACTIVE,0,SET,LSTP
!LSTEP_LAST_2=(LSTEP_LAST_1)/2
!LSTEP_SLANE_1=0
!*DO,ICASE_4,2,(LSTEP_LAST_2)/2+1
!*DO,ICASE_5,LSTEP_LAST_2+1+1,LSTEP_LAST_1+1
!LCASE,ICASE_4
!LCOPER,ADD,ICASE_5
!LSTEP_SLANE_1=LSTEP_SLANE_1+1
!RAPPND,LSTEP_SLANE+LSTEP_SLANE_1,26
!SET,LSTEP_SLANE+LSTEP_SLANE_1
```

```
!!~MOD_SF
!*ENDDO
!*ENDDO

!LCDEF,ERASE
!LCDEF,1,LSTEP_LAST+3+1,1

!*DO,ICASE_6,LSTEP_SLANE+1,LSTEP_SLANE+LSTEP_SLANE_1
!!LCDEF,ICASE_6,ICASE_6,1
!ICASE_8_DEMO=1+(ICASE_6-(LSTEP_SLANE+1)+1)
!LCDEF,ICASE_8_DEMO,ICASE_6,1
!*ENDDO

!*GET,LSTEP_INI_2,ACTIVE,0,SET,LSTP

!!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
!*DO,ICASE_7,LSTEP_SLANE+1,LSTEP_SLANE+LSTEP_SLANE_1
!ICASE_9_DEMO=1+(ICASE_7-(LSTEP_SLANE+1)+1)
!LCASE,ICASE_9_DEMO
!LCOPER,ADD,1
!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
!LSTEP_CURRENT=(LSTEP_SLANE+LSTEP_SLANE_1)+(ICASE_7-(LSTEP_SLANE+1))+1
!RAPPND,LSTEP_CURRENT,26
!SET,LSTEP_CURRENT
!~MOD_SF
!*ENDDO

!*GET,LSTEP_FIN_2,ACTIVE,0,SET,LSTP

!*********************************2ND LANE********************************

LOADSTEP_SET1=LSTEP_FIN_1-LSTEP_INI_1
!LOADSTEP_SET2=LSTEP_FIN_2-LSTEP_INI_2
!TOTAL_LOADSTEP=LOADSTEP_SET1+LOADSTEP_SET2

ESEL,S,MAT,,2
*GET,N_ELEM_ARCH,ELEM,0,COUNT
N_COMB_OTHER=1
N_COMB_VEH=LSTEP_LAST
N_LCASE=N_COMB_OTHER+N_COMB_VEH
!TOTAL_N_ELEM_ARCH=N_ELEM_ARCH*N_LCASE

!***********************ENVELOPED REINFORCEMENT IN ARCH*******************

ESEL,S,MAT,,2
*GET,ARCH_ELEMENT_MIN,ELEM,0,NUM,MIN
ARCH_ELEMENT_MAX=ARCH_ELEMENT_MIN+NINT(N_ELEMENT)

!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST+2)+1,(LSTEP_LAST+2)+N_LCASE,1
!~CFSET,,I_ARCH_MAX_LCASE_I,1

!!!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)/2+1,TOTAL_LOADSTEP,1

*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)+1,LOADSTEP_SET1,1

!*IF,I_ARCH_MAX_LCASE_I,LE,LOADSTEP_SET1,THEN
~CFSET,,LSTEP_INI_1+I_ARCH_MAX_LCASE_I,1
!*ELSEIF,I_ARCH_MAX_LCASE_I,GT,LOADSTEP_SET1
!~CFSET,,LSTEP_INI_2+(I_ARCH_MAX_LCASE_I-LOADSTEP_SET1),1
!*ENDIF
```

```
~DIMCON,2DB,,1,1,1000

!******************************ELEMENT CONTROL**************************

*IF,N_HANGER,GT,25,THEN
INCR_ELEM=2
*ELSE
INCR_ELEM=1
*ENDIF

*DO,I_ARCH_MAX_I,ARCH_ELEMENT_MIN+2,ARCH_ELEMENT_MAX-2,INCR_ELEM

!******************************ELEMENT
CONTROL*****************************

~CFGET,ARCH_REINFACT_MAX_%I_ARCH_MAX_I%_%I_ARCH_MAX_LCASE_I%,ELEMENT,I_ARC
H_MAX_I,RESULT,REINFACT,I
*ENDDO

*ENDDO

ARCH_REIN_PER_I=1
ARCH_REIN=(ARCH_REIN_PER_I/100)*((ARCH_WIDTH/1000)*(ARCH_DEPTH/1000))
ARCH_REIN_FACE=ARCH_REIN/((ARCH_WIDTH/1000)*2+(ARCH_DEPTH/1000)*2)

ARCH_REIN_TOTAL=0

!******************************ELEMENT CONTROL**************************

*IF,N_HANGER,GT,25,THEN
INCR_ELEM=2
*ELSE
INCR_ELEM=1
*ENDIF

*DO,I_ARCH_MAX_I,ARCH_ELEMENT_MIN+2,ARCH_ELEMENT_MAX-2,INCR_ELEM

!******************************ELEMENT CONTROL**************************

*GET,ARCH_LENGTH_ELEM_%I_ARCH_MAX_I%,ELEM,I_ARCH_MAX_I,LENG

ARCH_REINFACT_%I_ARCH_MAX_I%=0

!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST+2)+1,(LSTEP_LAST+2)+N_LCASE,1
!!!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)/2+1,TOTAL_LOADSTEP,1

*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)+1,LOADSTEP_SET1,1

*IF,ARCH_REINFACT_MAX_%I_ARCH_MAX_I%_%I_ARCH_MAX_LCASE_I%,GE,ARCH_REINFACT
_%I_ARCH_MAX_I%,THEN
ARCH_REINFACT_%I_ARCH_MAX_I%=ARCH_REINFACT_MAX_%I_ARCH_MAX_I%_%I_ARCH_MAX_
LCASE_I%
*ELSE
ARCH_REINFACT_%I_ARCH_MAX_I%=ARCH_REINFACT_%I_ARCH_MAX_I%
*ENDIF

*ENDDO

ARCH_REIN_ELEM_%I_ARCH_MAX_I%=ARCH_REIN*ARCH_LENGTH_ELEM_%I_ARCH_MAX_I%*AR
CHREINFACT_%I_ARCH_MAX_I%
ARCH_REIN_TOTAL=ARCH_REIN_TOTAL+ARCH_REIN_ELEM_%I_ARCH_MAX_I%
```

```
*ENDDO

!***********************ENVELOPED REINFORCEMENT IN ARCH*******************

ESEL,S,TYPE,,10
ETABLE, VOLUME, VOLU
SSUM
*GET,VOLUME_CABLE,SSUM,,ITEM,VOLUME

ESEL,S,MAT,,2
ETABLE, VOLUME, VOLU
SSUM
*GET,VOLUME_ARCH,SSUM,,ITEM,VOLUME

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

UNIT_PRICE_CABLE=223.303 !TAKA/KG
UNIT_PRICE_CON_ARCH=10527.25 !TAKA/M3
UNIT_PRICE_REIN=80.04 !TAKA/KG

COST_CABLE=VOLUME_CABLE*8000*UNIT_PRICE_CABLE
COST_ARCH=VOLUME_ARCH*UNIT_PRICE_CON_ARCH

!*******************************ELEMENT CONTROL***************************

*IF,N_HANGER,GT,25,THEN
ELEM_MULTIPLIER=2
*ELSE
ELEM_MULTIPLIER=1
*ENDIF

!*******************************ELEMENT CONTROL***************************

COST_REINFORCE=(ARCH_REIN_TOTAL*2)*7833.4*UNIT_PRICE_REIN*ELEM_MULTIPLIER
TOTAL_COST=(COST_CABLE)+(COST_ARCH)+(COST_REINFORCE)
FINISH

!*************************************************************************

*CFOPEN,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\FUNC,TXT
*VWRITE,COST_CABLE
(E16.9,1X)
*VWRITE,COST_ARCH
(E16.9,1X)
*VWRITE,COST_REINFORCE
(E16.9,1X)
*VWRITE,TOTAL_COST
(E16.9,1X)
*CFCLOS
/EXIT

!**********************************END************************************
```

```
!-------------------------------------------------------------------------
!Author                                            Nazrul Islam
!                                                  Lecturer
!                                                  Dept. of Civil Engineering
!                                                  BUET, Dhaka 1000.
! ------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!*************************************************************************
!The following script is finite element simulation file for analysis and
!design of network arch bridges which is used by implicit function
!evaluation of global optimization algorithm EVOP.
!File Name is                 arch_imp_final.txt
!**********************************BEGIN**********************************

/BATCH
FINISH
/CLEAR
!/TRIAD,OFF
!/PLOPTS,LOGO,0
/COM,ANSYS RELEASE 12.0
~CFACTIV,CIVILFEM,Y
~UNITS,SI
~CFACTIV,NLBR,Y
~CFACTIV,PRSC,Y
~CODESEL,LRFD13,AASHTOHB,ACI,,AASHTO
~CFCONFG,RCV,RESMAX,100000

/PREP7

!~CFSAVE
!/NOPR
!PARSAV,ALL,_CFTEMP.DAT

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ANSYS_IMP.TXT
/INPUT,'E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_MODEL_FINAL.MAC'
/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_PARAM.TXT
/INPUT,'E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ARCH_BRIDGE_RADIAL_FINAL.MAC
'
!PARRES,NEW,_CFTEMP.DAT
!/GO

FINISH

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\INFLUENCE_LINE_IMP.TXT

/SOLU
ALLSEL,ALL
CSYS,0

!*************************************************************************

ANTYPE,STATIC
!EQSLV,PCG,1E-8
!SSTIF,ON
!NSUBST,3
!OUTPR,ALL,ALL

!*************************************************************************
```

```
!! VEHICLE LOAD

~CFLSWRT,INIT
/TITLE,VEHICLE LOAD
~BLVLIB,1,AASHTO,HS20,1
~BLVMDF,1,NCOL,4
~BLVMDF,1,COLSP,4.267,1,2
~BLVMDF,1,COLSP,4.267,2,3
~BLVMDF,1,COLSP,10,3,4
~BLVMDF,1,LOC,4.267,1
!~BLVMDF,1,LOADS,0,1,4
!~BLVMDF,1,LOADS,0,2,4

!! FAMILY DEFINITION
~BLFDF,1001,SELECT,1,0,2
!! ASIGNS VEHICLE 1 TO FAMILY 1001
~BLVR,1001,,1,,,-2.0,,,0,5
~BLVR,1001,,1,,,2.0,,,0,5
!~BLVR,1001,,1,,,,,,0,5

~BLSOLVE
FDELE,ALL,ALL ! DELETE LOADS

*GET,LSTEP_LAST_1,ACTIVE,0,SET,LSTP

/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\LANE_LOAD_SOLU.TXT

ALLSEL,ALL
!ESEL,U,MAT,,4
!SELF WEIGHT
/TITLE,SELF WEIGHT
!TIME,21
ACEL,,,9.81
!~CFLSWRT,,LSTEP_LAST_1+19
~CFLSWRT
ACEL,

ALLSEL,ALL
!DEAD LOAD
/TITLE,WEARING SURFACE
!TIME,22
!SFBEAM,ALL,2,PRES,33200
ESEL,S,ENAME,,154
WEARING=1436      !WEARING = 30 PSF
SFE,ALL,1,PRES,0,WEARING
ALLSEL,ALL
!~CFLSWRT,LSTEP_LAST_1+20
~CFLSWRT
SFEDELE,ALL,ALL,ALL

LSTEP_LAST_LANE=LSTEP_LAST_1+5*2*1+2

ALLSEL,ALL
!LSSOLVE,LSTEP_LAST+1,LSTEP_LAST+4,1
~CFLSSLV,LSTEP_LAST_1+1,LSTEP_LAST_LANE,1
!SOLVE
FINISH

/POST1

RSYS,SOLU
```

```
LSTEP_LAST=LSTEP_LAST_1+5*2*1
DL_FACT=1.3
LL_FACT_TRUCK=2.17*(1+50/(125+3.28*SPAN_ARCH))
LL_FACT_LANE=2.17*(1+50/(125+3.28*SPAN_ARCH))

*DO,ICASE_1,1,LSTEP_LAST
LCDEF,(ICASE_1-1)+2,ICASE_1,1

*IF,ICASE_1,LE,LSTEP_LAST_1,THEN
LCFACT,(ICASE_1-1)+2,LL_FACT_TRUCK
*ELSEIF,ICASE_1,GT,LSTEP_LAST_1
LCFACT,(ICASE_1-1)+2,LL_FACT_LANE
*ENDIF
*ENDDO

*DO,ICASE_2,LSTEP_LAST+1,LSTEP_LAST+2
LCDEF,ICASE_2+1,ICASE_2,1
LCFACT,ICASE_2+1,DL_FACT
*ENDDO

!/TITLE,1.30 DC
LCASE,LSTEP_LAST+1+1
LCOPER,ADD,LSTEP_LAST+2+1
/TITLE,1.30 DC
RAPPND,LSTEP_LAST+3+1,26
SET,LSTEP_LAST+3+1
~MOD_SF
!LCDEF,LSTEP_LAST+3,LSTEP_LAST+3,1
LCDEF,1,LSTEP_LAST+3+1,1

*GET,LSTEP_INI_1,ACTIVE,0,SET,LSTP

!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
*DO,ICASE_3,2,LSTEP_LAST+1
LCASE,ICASE_3
!LCOPER,ADD,LSTEP_LAST+3
LCOPER,ADD,1
/TITLE,1.30 (DC+DW) + 2.17 (L+I)
RAPPND,LSTEP_LAST+3+1+ICASE_3-1,26
SET,LSTEP_LAST+3+1+ICASE_3-1
~MOD_SF
*ENDDO
*GET,LSTEP_FIN_1,ACTIVE,0,SET,LSTP

!*********************************2ND LANE*******************************

!*GET,LSTEP_SLANE,ACTIVE,0,SET,LSTP
!LSTEP_LAST_2=(LSTEP_LAST_1)/2
!LSTEP_SLANE_1=0
!*DO,ICASE_4,2,(LSTEP_LAST_2)/2+1
!*DO,ICASE_5,LSTEP_LAST_2+1+1,LSTEP_LAST_1+1
!LCASE,ICASE_4
!LCOPER,ADD,ICASE_5
!LSTEP_SLANE_1=LSTEP_SLANE_1+1
!RAPPND,LSTEP_SLANE+LSTEP_SLANE_1,26
!SET,LSTEP_SLANE+LSTEP_SLANE_1
!!~MOD_SF
!*ENDDO
!*ENDDO
```

```
!LCDEF,ERASE
!LCDEF,1,LSTEP_LAST+3+1,1

!*DO,ICASE_6,LSTEP_SLANE+1,LSTEP_SLANE+LSTEP_SLANE_1
!!LCDEF,ICASE_6,ICASE_6,1
!ICASE_8_DEMO=1+(ICASE_6-(LSTEP_SLANE+1)+1)
!LCDEF,ICASE_8_DEMO,ICASE_6,1
!*ENDDO

!*GET,LSTEP_INI_2,ACTIVE,0,SET,LSTP

!!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
!*DO,ICASE_7,LSTEP_SLANE+1,LSTEP_SLANE+LSTEP_SLANE_1
!ICASE_9_DEMO=1+(ICASE_7-(LSTEP_SLANE+1)+1)
!LCASE,ICASE_9_DEMO
!LCOPER,ADD,1
!/TITLE,1.30 (DC+DW) + 2.17 (L+I)
!LSTEP_CURRENT=(LSTEP_SLANE+LSTEP_SLANE_1)+(ICASE_7-(LSTEP_SLANE+1))+1
!RAPPND,LSTEP_CURRENT,26
!SET,LSTEP_CURRENT
!~MOD_SF
!*ENDDO

!*GET,LSTEP_FIN_2,ACTIVE,0,SET,LSTP

!*********************************2ND LANE********************************

LOADSTEP_SET1=LSTEP_FIN_1-LSTEP_INI_1
!LOADSTEP_SET2=LSTEP_FIN_2-LSTEP_INI_2
!TOTAL_LOADSTEP=LOADSTEP_SET1+LOADSTEP_SET2

ESEL,S,MAT,,2
*GET,N_ELEM_ARCH,ELEM,0,COUNT
N_COMB_OTHER=1
N_COMB_VEH=LSTEP_LAST
N_LCASE=N_COMB_OTHER+N_COMB_VEH

!********************************MAXIMUM HANGER STRESS*******************

ESEL,S,TYPE,,10


HSMAX=10
!*DO,I_HANG_MAX,(LSTEP_LAST+2)+1,(LSTEP_LAST+2)+N_LCASE,1

!*DO,I_HANG_MAX,1,TOTAL_LOADSTEP,1
!!!*DO,I_HANG_MAX,(LSTEP_LAST_1)/2+1,TOTAL_LOADSTEP,1

*DO,I_HANG_MAX,(LSTEP_LAST_1)+1,LOADSTEP_SET1,1

!*IF,I_HANG_MAX,LE,LOADSTEP_SET1,THEN
~CFSET,,LSTEP_INI_1+I_HANG_MAX,1
!*ELSEIF,I_HANG_MAX,GT,LOADSTEP_SET1
!~CFSET,,LSTEP_INI_2+(I_HANG_MAX-LOADSTEP_SET1),1
!*ENDIF

ETABLE,HANG_AXIAL_STRESS,LS,1
ESORT,ETAB,HANG_AXIAL_STRESS
!PRETAB,HANG_AXIAL_STRESS
!PLETAB,HANG_AXIAL_STRESS
*GET,HSMAX_%I_HANG_MAX%,SORT,,MAX
```

```
*GET,HSMIN,SORT,,MIN

*IF,HSMAX_%(I_HANG_MAX)%,GE,HSMAX,THEN
HSMAX=HSMAX_%(I_HANG_MAX)%
*ELSE
HSMAX=HSMAX
*ENDIF

*ENDDO

!*******************************MAXIMUM HANGER STRESS*******************

!****************MAXIMUM DESIGN INTERACTION VALUE IN ARCH*****************

ESEL,S,MAT,,2
*GET,ARCH_ELEMENT_MIN,ELEM,0,NUM,MIN
ARCH_ELEMENT_MAX=ARCH_ELEMENT_MIN+NINT(N_ELEMENT)
ARCH_INTERACT_MAX_LCASE_I=0
ARCH_REINFACT_MAX_LCASE_I=0
!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST+2)+1,(LSTEP_LAST+2)+N_LCASE,1

!*DO,I_ARCH_MAX_LCASE_I,1,TOTAL_LOADSTEP,1
!!!*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)/2+1,TOTAL_LOADSTEP,1

*DO,I_ARCH_MAX_LCASE_I,(LSTEP_LAST_1)+1,LOADSTEP_SET1,1

!*IF,I_ARCH_MAX_LCASE_I,LE,LOADSTEP_SET1,THEN
~CFSET,,LSTEP_INI_1+I_ARCH_MAX_LCASE_I,1
!*ELSEIF,I_ARCH_MAX_LCASE_I,GT,LOADSTEP_SET1
!~CFSET,,LSTEP_INI_2+(I_ARCH_MAX_LCASE_I-LOADSTEP_SET1),1
!*ENDIF

~DIMCON,2DB,,1,1,1000

ARCH_INTERACT_MAX_I=0
ARCH_REINFACT_MAX_I=0

!******************************ELEMENT CONTROL**************************

*IF,N_HANGER,GT,25,THEN
INCR_ELEM=2
*ELSE
INCR_ELEM=1
*ENDIF

*DO,I_ARCH_MAX_I,ARCH_ELEMENT_MIN+2,ARCH_ELEMENT_MAX-2,INCR_ELEM

!******************************ELEMENT CONTROL**************************

~CFGET,ARCH_INTERACT_MAX_%I_ARCH_MAX_I%,ELEMENT,I_ARCH_MAX_I,RESULT,CRT_TO
T,I
~CFGET,ARCH_REINFACT_MAX_%I_ARCH_MAX_I%,ELEMENT,I_ARCH_MAX_I,RESULT,REINFA
CT,I

*IF,ARCH_INTERACT_MAX_%I_ARCH_MAX_I%,GE,ARCH_INTERACT_MAX_I,THEN
ARCH_INTERACT_MAX_I=ARCH_INTERACT_MAX_%I_ARCH_MAX_I%
*ELSE
ARCH_INTERACT_MAX_I=ARCH_INTERACT_MAX_I
*ENDIF

*IF,ARCH_REINFACT_MAX_%I_ARCH_MAX_I%,GE,ARCH_REINFACT_MAX_I,THEN
```

```
ARCH_REINFACT_MAX_I=ARCH_REINFACT_MAX_%I_ARCH_MAX_I%
*ELSE
ARCH_REINFACT_MAX_I=ARCH_REINFACT_MAX_I
*ENDIF

*ENDDO

ARCH_INTERACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%=ARCH_INTERACT_MAX_I
ARCH_REINFACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%=ARCH_REINFACT_MAX_I

*IF,ARCH_INTERACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%,GE,ARCH_INTERACT_MAX_LC
ASE_I,THEN
ARCH_INTERACT_MAX_LCASE_I=ARCH_INTERACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%
*ELSE
ARCH_INTERACT_MAX_LCASE_I=ARCH_INTERACT_MAX_LCASE_I
*ENDIF

*IF,ARCH_REINFACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%,GE,ARCH_REINFACT_MAX_LC
ASE_I,THEN
ARCH_REINFACT_MAX_LCASE_I=ARCH_REINFACT_MAX_LCASE_I_%I_ARCH_MAX_LCASE_I%
*ELSE
ARCH_REINFACT_MAX_LCASE_I=ARCH_REINFACT_MAX_LCASE_I
*ENDIF

*ENDDO

!*************MAXIMUM DESIGN INTERACTION VALUE IN ARCH******************


*GET,UNSUPPORT_LENGTH_ELEM,ELEM,I_ARCH_MAX_I,LENG
K_UNBRACE=1.323

*IF,N_HANGER,GT,20,THEN
SLENDER_FACTOR=K_UNBRACE*UNSUPPORT_LENGTH_ELEM*2/(0.3*ARCH_DEPTH_M)
*ELSE
SLENDER_FACTOR=K_UNBRACE*UNSUPPORT_LENGTH_ELEM*3/(0.3*ARCH_DEPTH_M)
*ENDIF

FINISH

!*************************************************************************

*CFOPEN,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\IMP.TXT
!*VWRITE,COST_CABLE
!(E16.9,1X,'COST_CABLE')
*VWRITE,HSMAX
(E16.9,1X)
*VWRITE,HSMIN
(E16.9,1X)
*VWRITE,ARCH_INTERACT_MAX_LCASE_I
(E16.9,1X)
*VWRITE,ARCH_REINFACT_MAX_LCASE_I
(E16.9,1X)
*VWRITE,SLENDER_FACTOR
(E16.9,1X)
*CFCLOS
/EXIT

!*********************************END*************************************
```

```
!----------------------------------------------------------------------------
!Author                                          Nazrul Islam
!                                                Lecturer
!                                                Dept. of Civil Engineering
!                                                BUET, Dhaka 1000.
! ---------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!****************************************************************************
!The following script is input file of FE simulation code and is generated
!by EVOP and optimization engine.
!File Name is        ansys_func.txt or ansys_imp.txt
!***********************************BEGIN************************************

N_hanger = 20.008
theta_set1 = 40.871221
delta_ANSYS_set1 = -0.018262933
cable_area_astm_A586 = 1548.39
arch_width = 474.99
arch_depth = 1625.01
rise_span_ratio = .21238095

!***********************************END**************************************

!----------------------------------------------------------------------------
!Author                                          Nazrul Islam
!                                                Lecturer
!                                                Dept. of Civil Engineering
!                                                BUET, Dhaka 1000.
! ---------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!****************************************************************************
!The following script is material, geometry and element definition file
!required for finite element analysis and design of network arch bridges
!File Name is                arch_model_final.mac
!***********************************BEGIN************************************

/TITLE, 'ARCH CONCRETE BRIDGE'

!ARCH DIMENSIONS
SPAN_ARCH=100
!MATERIALS
ECC = 2.482E10    ! CONCRETE ELASTICITY MODULUS

FAC_INTERACT=0.9
FC_PSI=3500
FC_N_M=FC_PSI/145*1E6
FC=FC_N_M*FAC_INTERACT

FY_PSI=60000
FY_N_M=FY_PSI/145*1E6
FY=FY_N_M*FAC_INTERACT

~CFMP,1,LIB,CONCRETE,ACI,FC_3500
~CFMP,1,USER
!~CFMP,1,CONCR,EXLN  ,,ECC
!~CFMP,1,ACI_C,EC   ,,ECC,3
~CFMP,1,DATGEN,GAM  ,,24500
```

```
~CFMP,1,ACI_C,EC   ,,ECC,7,0,0
~CFMP,1,ACI_C,FC   ,,FC

!~CFMP,2,LIB,STEEL,ASTM,A36 ! HA-35 CONCRETE

~CFMP,2,LIB,CONCRETE,ACI,FC_3500
~CFMP,2,USER
!~CFMP,2,CONCR,EXLN  ,,ECC
!~CFMP,2,ACI_C,EC   ,,ECC,3
~CFMP,2,DATGEN,GAM  ,,24500
~CFMP,2,ACI_C,EC   ,,ECC,7,0,0
~CFMP,2,ACI_C,FC   ,,FC

~CFMP,3,LIB,PREST,ASTMA416,GR270
~CFMP,3,USER
~CFMP,3,PREST,MU,,0.21
~CFMP,3,PREST,K ,,1.26E-002
~CFMP,3,PREST,A ,,5E-003
~CFMP,3,DATGEN,GAM  ,,78453
!~CFMP,3,DATGEN,GAM  ,,0000
~CFMP,3,ACI_PRES,FPU,,1520E6
~CFGET,FY_CABLE,MATERIAL,3,ACI_PRES,FPU

MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,4,,2.482E10
MPDATA,PRXY,4,,0.2
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,DENS,4,,0
MPTEMP,,,,,,,,
MPTEMP,1,0
UIMP,4,REFT,,,
MPDATA,ALPX,4,,1E-005
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,DAMP,4,,0

~CFMP,5,LIB,REINF,ACI,FY_60000
~CFMP,5,USER
~CFMP,5,ACI_S,FY   ,,FY

!ELEMENT TYPE
ET,1,BEAM44
ET,2,SOLID45

! BRIDGE SECTIONS DEFINITION
!~BRSSLAB,1,TF,1,8.5,9.5,12,3.0,2.2,0.5
~BRSSLAB,1,TF,1,8,9,10,0.52,0.4,0.1
!~BRSSLAB,1,RS,1,12,0.52

! ARCH
ET,3,BEAM4
!~CSECDMS,1,CIRC,2,0.90
ARCH_WIDTH_M=ARCH_WIDTH/1000
ARCH_DEPTH_M=ARCH_DEPTH/1000
~CSECDMS,1,REC,2,ARCH_WIDTH_M,ARCH_DEPTH_M
!~RNFDEF,1,1,4,1,0,0,0.038,0.0013
!~RNFDEF,1,1,5,1,0,0.04,,,16,,,0.15,4 ! GROUP 1

ARCH_STEEL_PER_I=1
```

```
ARCH_STEEL=(ARCH_STEEL_PER_I/100)*(ARCH_WIDTH_M*ARCH_DEPTH_M)
ARCH_STEEL_FACE=ARCH_STEEL/(ARCH_WIDTH_M*2+ARCH_DEPTH_M*2)
~RNFDEF,1,1,5,1,0,0.04,,ARCH_STEEL_FACE,,,,0.15,4 ! GROUP 1
~RNFDEF,1,2,5,2,0,0.04,,ARCH_STEEL_FACE,,,,0.15,4 ! GROUP 1
~RNFDEF,1,3,5,3,0,0.04,,ARCH_STEEL_FACE,,,,0.15,4 ! GROUP 1
~RNFDEF,1,4,5,4,0,0.04,,ARCH_STEEL_FACE,,,,0.15,4 ! GROUP 1

!~RNFMDF,1,1,FI,20
!~RNFMDF,1,1,S,0.1333
!~RNFDEF,1,2,5,3,0,0.04,,,16,,,0.15,4 ! GROUP 2
!~RNFMDF,1,2,FI,20
!~RNFMDF,1,2,N,6
~SECMDF,1,STRPROP,ASEC,,7
~BMSHPRO,1,BEAM,1,1,,,4,1,0,,ARCH

!PILE
~CSECDMS,2,BOX,1,0.6,2,0.2,0.2,
~BMSHPRO,2,BEAM,2,2,,,4,1,0,,PIER

!CABLE
ET,10,LINK10,,,0
!ET,10,LINK8,,,0
!R,10,10E-4
!PI_=4*ATAN(1)
!CABLE_AREA=PI_*(CABLE_DIA**2)/4
CABLE_AREA=(CABLE_AREA_ASTM_A586)*1E-6
R,10,CABLE_AREA,0
KEYOPT,10,3,0

!BRACING
~CSECDMS,3,CIRC,1,0.15
~BMSHPRO,3,BEAM,3,3,,,4,1,0,,BRACING

!**********************************END***********************************

!--------------------------------------------------------------------------
!Author                                              Nazrul Islam
!                                                    Lecturer
!                                                    Dept. of Civil Engineering
!                                                    BUET, Dhaka 1000.
! -------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!*************************************************************************
!The following script is parameter file required for finite element
modeling !of network arch bridges
!File Name is                 arch_param.txt
!**********************************BEGIN*********************************

*SET,PR1_ ,
*SET,ZR1_ , 0.000
*SET,ZR2_ ,
*SET,NR_ , 0
*SET,DR_ ,
*DIM,DR_,ARRAY,NR_
! DR_(1) =
*SET,NPILDT_R_ , 0
*SET,ZTR_ ,
*DIM,ZTR_,ARRAY,NPILDT_R_
! ZTR_(1) =
```

```
*SET,PC1_ , 0.000
*SET,ZC1_ , 0.000
*SET,ZC2_ , 0.000
*SET,ZAL_ , -20.00
*SET,ZAR_ , -20.00

RISE_VAL=(1-(2*RISE_SPAN_RATIO)**2)/(1+(2*RISE_SPAN_RATIO)**2)
THETA_CIRC=ACOS(RISE_VAL)
PER_THETA_CIRC=100*(TAN(THETA_CIRC))
*SET,PAL_ , PER_THETA_CIRC
*SET,BEAMYES_ , 0
*SET,SUPPORT_ , 1

*IF,N_HANGER,GT,20,THEN
N_ELEMENT=NINT(N_HANGER)*2
*ELSE
N_ELEMENT=NINT(N_HANGER)*3
*ENDIF

*SET,NC_ , NINT(N_ELEMENT)+1
!*SET,NC_ , 5

*SET,DC_ ,
*DIM,DC_,ARRAY,NC_
! DC_(1) = 20@15.8

*DO,III,1,NC_,1
!RISE_CHECK=RISE_SPAN_RATIO*SPAN_ARCH
DC_(III)=SPAN_ARCH/NC_
!DC_(III)=5
*ENDDO

*SET,NUMARCH_ , 2
*SET,DA_ ,
*DIM,DA_,ARRAY,NUMARCH_
! DA_(1) = 3.036, -3.036
*SET,DA_(1) , 4
*SET,DA_(2) ,-4
*SET,PL1_ , 0.000
*SET,ZL1_ , 0.000
*SET,ZL2_ , 0.000
*SET,NL_ , 3
*SET,DL_ ,
*DIM,DL_,ARRAY,NL_
!DL_(1) = 11.8,15.8,15.8
*SET,DL_(1) , 0.1
*SET,DL_(2) , 0.1
*SET,DL_(3) , 0.1
*SET,NPILDT_L_ , 0
*SET,ZTL_ ,
*DIM,ZTL_,ARRAY,NPILDT_L_
! ZTL_(1) = -10,-15
!*SET,ZTL_(1) , -10
!*SET,ZTL_(2) , -15
*SET,XLI_ , 0.000
*SET,YLI_ , 0.000
*SET,EPS_ , 0.5000
*SET,EPS2_ , 0.3000
!DECK MESH SIZE
*SET,SIZEV_ , 0.250
*SET,SIZET_ , 2.000
```

```
!*SET,SIZEL_ , 0.5
*SET,SIZEL_ , 1.0
!
*SET,SIZEH_ , 0.000
*SET,PESIZE_ , 3.000
*SET,SECTR_ , 1
*SET,ELTYPE_ , 2
*SET,ELMAT_ , 1
*SET,ARELTYPE_ , 3
!!!*SET,ARMAT_ , 1
*SET,ARMAT_ , 2
!
*SET,BSARCH_ , 1
*SET,TPIL_ , 3
*SET,MPIL_ , 1
*SET,BSPIL_ , 2
*SET,SYMYES_ , 1
*SET,CABLEYES_ , 1
*SET,TIEDYES_ , 1
!CABLE PROPERTY
*SET,THT_ , 10
*SET,MHT_ , 3
*SET,RHT_ , 10
!
*SET,BRACINGYES_ , 1
*SET,TBRAC_ , 3
*SET,MBRAC_ , 4
*SET,BSBRAC_ , 3
*SET,SL1_ , 1
*SET,SLM_ , 1
*SET,SL2_ , 1
*SET,SC1_ , 1
*SET,SCM_ , 1
*SET,SC2_ , 1
*SET,SR1_ , 1
*SET,SRM_ , 1
*SET,SR2_ , 1
*SET,SINT_ , 1

!**********************************END************************************
**

!--------------------------------------------------------------------------
!Author                                           Nazrul Islam
!                                                 Lecturer
!                                                 Dept. of Civil Engineering
!                                                 BUET, Dhaka 1000.
! ------------------------------------------------------------------------
!As a part of the thesis work "Global Optimization of Design Parameters of
!Network Arch Bridges" for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!*************************************************************************
!The following script is basic finite element modeling file of network
!arch bridges
!The file is very long. Some scripts are given
!File Name is                arch_bridge_radial_final.mac
!*********************************BEGIN**********************************

*CREATE,MI_PARAB1,MAC
ALFA_=X2_ - X1_
BETA_=Y2_ - Y1_
```

```
A_   =(BETA_ - ALFA_*Y1P_)/(ALFA_*ALFA_)
B_   = Y1P_
Y2P_=2*A_*ALFA_+B_
AA_=A_
BB_=-2*A_*X2_
CC_=Y2_+A_*X2_**2+B_
*END
*CREATE,MI_PARAB2,MAC
Y_ = Y1_ + A_*(X_-X1_)*(X_-X1_) + B_*(X_-X1_)
YP_= 2*A_*(X_-X1_) + B_
*END

*CREATE,MI_CIRCLE,MAC
Y_ = Y1_ + BB + SQRT(R_**2-(X_ - X_MID)**2)
!RISE=R_ - (-BB)
YP_= 2*A_*(X_-X1_) + B_
*END

*CREATE,MI_FUNCION,MAC
X_=ARG1
FX_=(A1_-A2_)*X_**2+(B1_-B2_)*X_+(C1_-C2_)
*END
*CREATE,MI_PARAB3,MAC
*DO,II,1,1000
XINT_=(X1_+X2_)/2
MI_FUNCION,XINT_
*IF,ABS(FX_),LT,0.01,*EXIT
MI_FUNCION,X1_    $  FX1_=FX_
MI_FUNCION,XINT_  $  FX_= FX_
*IF,FX1_*FX_,LT,0,THEN
X1_=XINT_
*ELSE
X2_=XINT_
*ENDIF
*ENDDO
*END
*CREATE,MODIFNOD_1,MAC
NUMITE_=ARG1
!
!
!
!
!
!
!
*CREATE,CREA_ARCO,MAC
      NIN_=NL_ $ NTRM_=NC_
      X1_=DT_(NIN_) $ Y1_= ZAL_ $ Y1P_=PAL_/100 $ X2_=DT_(NIN_+NTRM_) $
Y2_=ZAR_
      MI_PARAB1
TAN_THETA_I=Y1P_
THETA_I=ATAN(Y1P_)
SPAN=ALFA_
X_MID=X1_+SPAN/2
R_= (SPAN/2)/(SIN(THETA_I))
YR=(SPAN/2)/(TAN(THETA_I))
BB=-YR
RISE=R_ - YR
PI_=4*ATAN(1)
THETA_X=PI_/2-THETA_I
THETA_XX=THETA_I*2/NC_
```

```
      YY_=DA_(I) $ Y_=ZAL_
      TYPE,ARELTYPE_ $ MAT,ARMAT_ $ REAL,BSARCH_
      NSEL,ALL  $ ESEL,NONE
      *GET,NUDMX_,NODE,,NUM,MAX
*END

*CREATE,NODE_CIRCLE,MAC
X_I=ARG1
THETA_II=THETA_X+THETA_XX*(X_I-1)
N,NUDMX_+II,R_,THETA_II*180/PI_,YY_
PC_(II)=NUDMX_+II
!WPSTYL,DEFA

*END
!
!
!
!
!
!

~BRGEN,A,SIZEV_,SIZET_,SIZEL_,SIZEH_


*GET,N_SYS_MAX2_,CDSY,,NUM,MAX
CMSEL,S,CF_BR_SECS_NODE_12
*GET,ZMIN_,NODE,,MNLOC,Z
NSEL,S,LOC,Z,ZMIN_
*GET,YMAX_,NODE,,MXLOC,Y
BWIDTH_=2*YMAX_
/VIEW,1,1,1,1
/VUP,ALL,Z
EPLOT
/PSYMB,CS,0
CM,DECK_ELEM,ELEM
/PSYMB,CS,0
*IF,CABLEYES_,EQ,1,THEN
*DO,I,1,NUMARCH_
*IF,TAB_,EQ,1,THEN

*IF,TIEDYES_,EQ,1,THEN

                    CREA_ARCO
N,NUDMX_+1,DT_(NL_),YY_,Y_
                  PC_=
                  *DIM,PC_,ARRAY,NC_+1
                  PC_(1)=NUDMX_+1
                  BASE%I%_L=PC_(1)

WPSTYL,,10,,R_,,1,0,1,10
WPROTA,,90
WPOFFS,X_MID,BB,0
WPROTA,,,180
CSYS,4
                  *DO,II,2,NC_+1
                     *IF,II,LT,NC_+1,THEN
                  !X_I=II
                  !*USE,NODE_CIRCLE.MAC,X_I
THETA_II=THETA_X+THETA_XX*(II-1)
N,NUDMX_+II,R_,THETA_II*180/PI_,YY_
```

```
PC_(II)=NUDMX_+II
                     *ELSE
WPSTYL,DEFA
                   X_= MP_(NL_+II)
                    CREANOD,X_
                      *ENDIF
                    E,PC_(II-1),PC_(II)
                  *ENDDO
              BASE%I%_R=PC_(II)
                  CM,ARCH%I%,ELEM
                  *DIM,RON%I%_,ARRAY,NC_+1

          *DO,II,1,NC_+1
                       RON%I%_(II)=PC_(II)
                    *ENDDO
                  ESEL,NONE

!***************************HANGER****************************************
                  TYPE,THT_ $ MAT,MHT_  $ REAL,RHT_

                  N_CONNECT=3
                  !N_CONNECT_NC_L=N_CONNECT+2+2-(N_CONNECT-3)
                  !N_CONNECT_NC_R=N_CONNECT+2-2-(N_CONNECT-3)
N_CONNECT_NC_L=1
N_CONNECT_NC_R=1
!***********************CONTROL HANGER CONNECTIVITY***********************

*IF,N_HANGER,GT,20,THEN
INI_L=0
INI_INCR_L=4
*ELSE
INI_L=0
INI_INCR_L=6
*ENDIF
                  *DO,II,INI_L,NC_-N_CONNECT_NC_L,INI_INCR_L
!                 *DO,II,0,NC_-N_CONNECT_NC_L,6
!***********************CONTROL HANGER CONNECTIVITY***********************

                  NSEL,S,NODE,,PC_(II+2)
PI_=4*ATAN(1)
ANGLE_DEG1=THETA_SET1
ANGLE_RAD1=ANGLE_DEG1*(PI_)/180
THETA_ARC1=ANGLE_RAD1

                  ANGLE_DEG_DEL1=
                  *DIM,ANGLE_DEG_DEL1,ARRAY,NC_+1
                  ANGLE_RAD_DEL1=
                  *DIM,ANGLE_RAD_DEL1,ARRAY,NC_+1
                  THETA_ARC_DEL1=
                  *DIM,THETA_ARC_DEL1,ARRAY,NC_+1
                  THETA_ARC_TOT1=
                  *DIM,THETA_ARC_TOT1,ARRAY,NC_+1

!ANGLE_DEG_DEL1(II+2)=DELTA_ANSYS_SET1(II+2)
ANGLE_DEG_DEL1(II+2)=DELTA_ANSYS_SET1
ANGLE_RAD_DEL1(II+2)=ANGLE_DEG_DEL1(II+2)*(PI_)/180
THETA_ARC_DEL1(II+2)=ANGLE_RAD_DEL1(II+2)
THETA_ARC_TOT1(II+2)=THETA_ARC1+THETA_ARC_DEL1(II+2)*(II-2)

                  X1_=NX(PC_(II+2)) $ Y1_=NY(PC_(II+2)) $ Z1_=MPY_(II+2) $
Z1_ARC=NZ(PC_(II+2)) $ X2_=DT_(NL_+NC_)
```

```
                  X1_DECK=NX(PC_(II+2))+Z1_ARC*TAN(THETA_ARC_TOT1(II+2))

                  *IF,X1_-EPS_,LE,0,OR,ABS(X1_-X2_),LE,EPS_,*CYCLE
                  CMSEL,S,CF_BR_DECK_NODE
                  N2_=NODE(X1_DECK,Y1_,Z1_)
                  NSEL,A,NODE,,PC_(II+2)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                  !FILL,PC_(II+2+N_CONNECT),N2_,5,5000,1
                  !E,5000,5001
E,N2_,PC_(II+2)

!************************CONTROL HANGER CONNECTIVITY***********************

*IF,N_HANGER,GT,20,THEN
INI_R=2
INI_INCR_R=4
*ELSE
INI_R=3
INI_INCR_R=6
*ENDIF
                  *DO,II,INI_R,NC_-N_CONNECT_NC_R,INI_INCR_R

!************************CONTROL HANGER CONNECTIVITY***********************

                  NSEL,S,NODE,,PC_(II+2)

PI_=4*ATAN(1)
!ANGLE_DEG2=THETA_SET2
ANGLE_DEG2=THETA_SET1 + NINT(N_HANGER)*DELTA_ANSYS_SET1
ANGLE_RAD2=ANGLE_DEG2*(PI_)/180
THETA_ARC2=ANGLE_RAD2

                  ANGLE_DEG_DEL2=
                  *DIM,ANGLE_DEG_DEL2,ARRAY,NC_+1
                  ANGLE_RAD_DEL2=
                  *DIM,ANGLE_RAD_DEL2,ARRAY,NC_+1
                  THETA_ARC_DEL2=
                  *DIM,THETA_ARC_DEL2,ARRAY,NC_+1
                  THETA_ARC_TOT2=
                  *DIM,THETA_ARC_TOT2,ARRAY,NC_+1

!ANGLE_DEG_DEL2(II+2)=DELTA_ANSYS_SET2(II+2)
!ANGLE_DEG_DEL2(II+2)=DELTA_ANSYS_SET2
ANGLE_DEG_DEL2(II+2)=-DELTA_ANSYS_SET1

ANGLE_RAD_DEL2(II+2)=ANGLE_DEG_DEL2(II+2)*(PI_)/180
THETA_ARC_DEL2(II+2)=ANGLE_RAD_DEL2(II+2)
THETA_ARC_TOT2(II+2)=THETA_ARC2+THETA_ARC_DEL2(II+2)*(II-2)


                  X1_=NX(PC_(II+2)) $ Y1_=NY(PC_(II+2)) $ Z1_=MPY_(II+2) $
Z1_ARC=NZ(PC_(II+2)) $ X2_=DT_(NL_+NC_)
                  X1_DECK=NX(PC_(II+2))-Z1_ARC*TAN(THETA_ARC_TOT2(II+2))

                  *IF,X1_-EPS_,LE,0,OR,ABS(X1_-X2_),LE,EPS_,*CYCLE
                  CMSEL,S,CF_BR_DECK_NODE
                  N2_=NODE(X1_DECK,Y1_,Z1_)
                  NSEL,A,NODE,,PC_(II+2)


E,N2_,PC_(II+2)
```

```
                  *ENDDO


CM,CABLES%I%_E,ELEM
                  ANODE1_= BASE%I%_L
                  ANODE2_= BASE%I%_R
                  MODIFNOD_2,2
*ELSE
!
!
!
!
!
*IF,BRACINGYES_,EQ,1,AND,NUMARCH_,GE,2,THEN
ESEL,NONE
TYPE,TBRAC_$ MAT,MBRAC_  $ REAL,BSBRAC_
ESIZE,,BNDIV_
            *IF,TAB_,EQ,1,THEN
                        *IF,TIEDYES_,EQ,1,THEN,

                                    *DO,I,1,NUMARCH_-1

!*************************CONTROL BRACING CONNECTIVITY*******************
*IF,N_HANGER,GT,20,THEN
INI_BRAC=2
INI_INCR_BRAC=2
INI_GAP_BRAC=4
*ELSE
INI_BRAC=6
INI_INCR_BRAC=3
INI_GAP_BRAC=6
*ENDIF
*DO,II,1+INI_BRAC+1,NC_+1-INI_GAP_BRAC,INI_INCR_BRAC

!*************************CONTROL BRACING CONNECTIVITY*******************

E,RON%I%_(II),RON%I+1%_(II)

*ENDDO
*ELSE


!**************************************DOF******************************

ALLSEL
*DO,I,1,NUMARCH_
*IF,SUPPORT_,EQ,0,THEN
D,BASE%I%_L,,,,,,ALL,
D,BASE%I%_R,,,,,,ALL,
*IF,NPILDT_L_,GT,0,THEN
*DO,II,1,NPILDT_L_
D,NPL%I%_%II%,,,,,,ALL,
*ENDDO
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
*DO,II,1,NPILDT_R_
D,NPR%I%_%II%,,,,,,ALL,
*ENDDO
*ENDIF
*ELSE
```

```
!D,BASE%I%_L,,,,,,UX,UY,UZ,
!D,BASE%I%_R,,,,,,,,UZ,
!D,BASE%I%_R,,,,,,UX,UY,UZ,
!!D,BASE%I%_L,,,,,,,,UZ,
!!D,BASE%I%_R,,,,,,,,UZ,

*IF,NPILDT_L_,GT,0,THEN
*DO,II,1,NPILDT_L_
D,NPL%I%_%II%,,,,,,UX,UY,UZ,
*ENDDO
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
*DO,II,1,NPILDT_R_
D,NPR%I%_%II%,,,,,,UX,UY,UZ,
*ENDDO
*ENDIF
*ENDIF
*ENDDO
CMSEL,S, CF_BR_SECS_NODE_%N_SYS_MAX1_+1%
*GET,Z_MIN_,NODE,,MNLOC,Z
NSEL,R,LOC,Z,Z_MIN_-0.05,Z_MIN_+0.05 $ D,ALL,UY,0 $ D,ALL,UZ,0
NSEL,U,LOC,Y,-3.8,3.8 $ D,ALL,UX,0 $ D,ALL,UY,0 $ D,ALL,UZ,0
!NSEL,R,LOC,Y,-4,-4 $ D,ALL,UX,0 $ D,ALL,UY,0 $ D,ALL,UZ,0
!NSEL,R,LOC,Y,4,4 $ D,ALL,UX,0 $ D,ALL,UY,0 $ D,ALL,UZ,0
NSEL,ALL
CMSEL,S, CF_BR_SECS_NODE_%N_SYS_MAX2_-1%
*GET,Z_MIN_,NODE,,MNLOC,Z
NSEL,R,LOC,Z,Z_MIN_-0.05,Z_MIN_+0.05 $ D,ALL,UY,0 $ D,ALL,UZ,0
!NSEL,U,LOC,Y,-3.8,3.8 $ D,ALL,UZ,0
NSEL,ALL
EPLOT
*IF,TAB_,EQ,1,THEN
*IF,TIEDYES_,EQ,1,THEN
CMSEL,S,DECK_ELEM
*DO,II,1,NUMARCH_
CMSEL,A,ARCHS_
CMSEL,A,CABLES%II%_E
*IF,BRACINGYES_,EQ,1,THEN
CMSEL,A,BRACING_ELEM
*ENDIF
*IF,NPILDT_L_,GT,0,THEN
CMSEL,A,TPIERSL_
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
CMSEL,A,TPIERSR_
*ENDIF
*ENDDO
*ELSE
CMSEL,S,DECK_ELEM
*DO,II,1,NUMARCH_
CMSEL,A,ARCHS_
CMSEL,A,CABLES%II%_E
CMSEL,A,PIERSL_
CMSEL,A,PIERSR_
*IF,BRACINGYES_,EQ,1,THEN
CMSEL,A,BRACING_ELEM
*ENDIF
*IF,NPILDT_L_,GT,0,THEN
CMSEL,A,TPIERSL_
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
```

```
CMSEL,A,TPIERSR_
*ENDIF
*ENDDO
*ENDIF
*ELSEIF,TAB_,EQ,0,THEN
CMSEL,S,DECK_ELEM
*DO,II,1,NUMARCH_
CMSEL,A,ARCHS_
CMSEL,A,PIERS%II%_E
*IF,BRACINGYES_,EQ,1,THEN
CMSEL,A,BRACING_ELEM
*ENDIF
*IF,NPILDT_L_,GT,0,THEN
CMSEL,A,TPIERSL_
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
CMSEL,A,TPIERSR_
*ENDIF
*ENDDO
*ELSE
CMSEL,S,DECK_ELEM
*DO,II,1,NUMARCH_
CMSEL,A,ARCHS_
CMSEL,A,PIERSL_
CMSEL,A,PIERSR_
*IF,BRACINGYES_,EQ,1,THEN
CMSEL,A,BRACING_ELEM
*ENDIF
*IF,NPILDT_L_,GT,0,THEN
CMSEL,A,TPIERSL_
*ENDIF
*IF,NPILDT_R_,GT,0,THEN
CMSEL,A,TPIERSR_
*ENDIF
*ENDDO
*ENDIF
/NUMBER,1
/PBC,ALL,1
CMPLOT
/PLOPTS,INFO,3
/VIEW,1,1,1,1
/ANG,1
/REP,FAST
ALLSEL

!**********************************END************************************
```

```
!-------------------------------------------------------------------------
!Author                                        Nazrul Islam
!                                              Lecturer
!                                              Dept. of Civil Engineering
!                                              BUET, Dhaka 1000.
! ------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!*************************************************************************
!The following script is the file for generation of influence line in arch
!of network arch bridges.
!File Name is                 influence_line_func.txt
!*********************************BEGIN**********************************

/SOLU
ALLSEL,ALL
CSYS,0

ANTYPE,STATIC
!EQSLV,PCG,1E-8
!SSTIF,ON
!NSUBST,3
!OUTPR,ALL,ALL

!! VEHICLE LOAD
/TITLE,VEHICLE LOAD
~BLVLIB,2,AASHTO,H20,1
~BLVMDF,2,COLSP,SPAN_ARCH/10,1,2
~BLVMDF,2,LOADS,0,1,1
~BLVMDF,2,LOADS,0,2,1
~BLVMDF,2,LOADS,-0.5,1,2
~BLVMDF,2,LOADS,-0.5,2,2
~BLVMDF,2,LOC,0,1

!! FAMILY DEFINITION
~BLFDF,1001,SELECT,1,0,1
!! ASIGNS VEHICLE 1 TO FAMILY 1001
~BLVR,1001,,2,,,,,,0,5
~BLSOLVE
FDELE,ALL,ALL ! DELETE LOADS

FINISH

/POST1
/INPUT,E:\MSC\THESIS\ARCH_EVOP_ANSYS_WORKING\ANSYS_FUNC.TXT

!***************************INFLUENCE LINE*******************************
ESEL,S,MAT,,2
*GET,ARCH_ELEMENT_MIN,ELEM,0,NUM,MIN
ARCH_ELEMENT_MAX=ARCH_ELEMENT_MIN+NINT(N_ELEMENT)

ARCH_ELEMENT_I1=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10)
ARCH_ELEMENT_I2=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*2)
ARCH_ELEMENT_I3=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*3)
ARCH_ELEMENT_I4=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*4)
ARCH_ELEMENT_I5=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*5)
ARCH_ELEMENT_I6=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*6)
ARCH_ELEMENT_I7=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*7)
ARCH_ELEMENT_I8=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*8)
ARCH_ELEMENT_I9=ARCH_ELEMENT_MIN+NINT(N_ELEMENT/10*9)
```

```
*DO,I_IL,1,10,1
~CFSET,,I_IL,1
~CFGET,ARCH_MY_ARCH_ELEMENT_I1_%I_IL%,ELEMENT,ARCH_ELEMENT_I1,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I2_%I_IL%,ELEMENT,ARCH_ELEMENT_I2,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I3_%I_IL%,ELEMENT,ARCH_ELEMENT_I3,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I4_%I_IL%,ELEMENT,ARCH_ELEMENT_I4,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I5_%I_IL%,ELEMENT,ARCH_ELEMENT_I5,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I6_%I_IL%,ELEMENT,ARCH_ELEMENT_I6,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I7_%I_IL%,ELEMENT,ARCH_ELEMENT_I7,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I8_%I_IL%,ELEMENT,ARCH_ELEMENT_I8,FORCE,MY,I
~CFGET,ARCH_MY_ARCH_ELEMENT_I9_%I_IL%,ELEMENT,ARCH_ELEMENT_I9,FORCE,MY,I

*IF,ARCH_MY_ARCH_ELEMENT_I1_%I_IL%,LT,0,THEN
NEG_LENG_I1_%I_IL%=1
POS_LENG_I1_%I_IL%=0
*ELSE
POS_LENG_I1_%I_IL%=1
NEG_LENG_I1_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I2_%I_IL%,LT,0,THEN
NEG_LENG_I2_%I_IL%=1
POS_LENG_I2_%I_IL%=0
*ELSE
POS_LENG_I2_%I_IL%=1
NEG_LENG_I2_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I3_%I_IL%,LT,0,THEN
NEG_LENG_I3_%I_IL%=1
POS_LENG_I3_%I_IL%=0
*ELSE
POS_LENG_I3_%I_IL%=1
NEG_LENG_I3_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I4_%I_IL%,LT,0,THEN
NEG_LENG_I4_%I_IL%=1
POS_LENG_I4_%I_IL%=0
*ELSE
POS_LENG_I4_%I_IL%=1
NEG_LENG_I4_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I5_%I_IL%,LT,0,THEN
NEG_LENG_I5_%I_IL%=1
POS_LENG_I5_%I_IL%=0
*ELSE
POS_LENG_I5_%I_IL%=1
NEG_LENG_I5_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I6_%I_IL%,LT,0,THEN
NEG_LENG_I6_%I_IL%=1
POS_LENG_I6_%I_IL%=0
*ELSE
POS_LENG_I6_%I_IL%=1
NEG_LENG_I6_%I_IL%=0
*ENDIF
```

```
*IF,ARCH_MY_ARCH_ELEMENT_I7_%I_IL%,LT,0,THEN
NEG_LENG_I7_%I_IL%=1
POS_LENG_I7_%I_IL%=0
*ELSE
POS_LENG_I7_%I_IL%=1
NEG_LENG_I7_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I8_%I_IL%,LT,0,THEN
NEG_LENG_I8_%I_IL%=1
POS_LENG_I8_%I_IL%=0
*ELSE
POS_LENG_I8_%I_IL%=1
NEG_LENG_I8_%I_IL%=0
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I9_%I_IL%,LT,0,THEN
NEG_LENG_I9_%I_IL%=1
POS_LENG_I9_%I_IL%=0
*ELSE
POS_LENG_I9_%I_IL%=1
NEG_LENG_I9_%I_IL%=0
*ENDIF

*ENDDO

!***********************CONCENTRATED LOAD LOCATION************************

*DO,I_CON_LOC,1,9,1

MAX_POS_MY_I%I_CON_LOC%=0
MAX_NEG_MY_I%I_CON_LOC%=0
MAX_POS_MY_LOC_I%I_CON_LOC%=0
MAX_NEG_MY_LOC_I%I_CON_LOC%=0

*DO,I_IL,1,10,1

*IF,ARCH_MY_ARCH_ELEMENT_I%I_CON_LOC%_%I_IL%,GE,MAX_POS_MY_I%I_CON_LOC%,TH
EN
MAX_POS_MY_I%I_CON_LOC%=ARCH_MY_ARCH_ELEMENT_I%I_CON_LOC%_%I_IL%
MAX_POS_MY_LOC_I%I_CON_LOC%=I_IL*SPAN/10
*ELSE
MAX_POS_MY_I%I_CON_LOC%=MAX_POS_MY_I%I_CON_LOC%
MAX_POS_MY_LOC_I%I_CON_LOC%=MAX_POS_MY_LOC_I%I_CON_LOC%
*ENDIF

*IF,ARCH_MY_ARCH_ELEMENT_I%I_CON_LOC%_%I_IL%,LE,MAX_NEG_MY_I%I_CON_LOC%,TH
EN
MAX_NEG_MY_I%I_CON_LOC%=ARCH_MY_ARCH_ELEMENT_I%I_CON_LOC%_%I_IL%
MAX_NEG_MY_LOC_I%I_CON_LOC%=I_IL*SPAN/10
*ELSE
MAX_NEG_MY_I%I_CON_LOC%=MAX_NEG_MY_I%I_CON_LOC%
MAX_NEG_MY_LOC_I%I_CON_LOC%=MAX_NEG_MY_LOC_I%I_CON_LOC%
*ENDIF

*ENDDO

*ENDDO

!***********************CONCENTRATED LOAD LOCATION************************
*DO,I_NODE_ARCH,1,9,1
```

```
NEG_LENG_I%I_NODE_ARCH%_COUNT1=0
POS_LENG_I%I_NODE_ARCH%_COUNT1=0
NEG_LENG_I%I_NODE_ARCH%_COUNT2=0
POS_LENG_I%I_NODE_ARCH%_COUNT2=0
NEG_LENG_I%I_NODE_ARCH%_COUNT3=0
POS_LENG_I%I_NODE_ARCH%_COUNT3=0
NEG_LENG_I%I_NODE_ARCH%_COUNT4=0
POS_LENG_I%I_NODE_ARCH%_COUNT4=0

NEG_LENG_I%I_NODE_ARCH%_COUNT_1=0
POS_LENG_I%I_NODE_ARCH%_COUNT_1=0
NEG_LENG_I%I_NODE_ARCH%_COUNT_2=0
POS_LENG_I%I_NODE_ARCH%_COUNT_2=0
NEG_LENG_I%I_NODE_ARCH%_COUNT_3=0
POS_LENG_I%I_NODE_ARCH%_COUNT_3=0
NEG_LENG_I%I_NODE_ARCH%_COUNT_4=0
POS_LENG_I%I_NODE_ARCH%_COUNT_4=0


*IF,NEG_LENG_I%I_NODE_ARCH%_1,EQ,1,THEN
I_I%I_NODE_ARCH%_NEG_COUNT1=0
I_I%I_NODE_ARCH%_POS_COUNT1=0
I_I%I_NODE_ARCH%_NEG_COUNT2=0
I_I%I_NODE_ARCH%_POS_COUNT2=0
I_I%I_NODE_ARCH%_NEG_COUNT3=0
I_I%I_NODE_ARCH%_POS_COUNT3=0
*DO,I_COUNT_LENG,1,10,1
*IF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_LENG_I%I_NODE_ARCH
%_COUNT1,EQ,0,THEN
NEG_LENG_I%I_NODE_ARCH%_COUNT1=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG_
LENG_I%I_NODE_ARCH%_COUNT1
NEG_COUNT1=1
I_I%I_NODE_ARCH%_NEG_COUNT1=I_COUNT_LENG
*ELSEIF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_COUNT1,EQ,1
POS_COUNT1=1
I_I%I_NODE_ARCH%_POS_COUNT1=I_COUNT_LENG
POS_LENG_I%I_NODE_ARCH%_COUNT1=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS_
LENG_I%I_NODE_ARCH%_COUNT1
*ELSEIF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_COUNT1,EQ,1
NEG_LENG_I%I_NODE_ARCH%_COUNT2=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG_
LENG_I%I_NODE_ARCH%_COUNT2
NEG_COUNT1=
NEG_COUNT2=1
I_I%I_NODE_ARCH%_NEG_COUNT2=I_COUNT_LENG
*ELSEIF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_COUNT2,EQ,1
POS_COUNT1=
POS_COUNT2=1
I_I%I_NODE_ARCH%_POS_COUNT2=I_COUNT_LENG
POS_LENG_I%I_NODE_ARCH%_COUNT2=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS_
LENG_I%I_NODE_ARCH%_COUNT2
*ELSEIF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_COUNT2,EQ,1
NEG_LENG_I%I_NODE_ARCH%_COUNT3=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG_
LENG_I%I_NODE_ARCH%_COUNT3
NEG_COUNT2=
NEG_COUNT3=1
I_I%I_NODE_ARCH%_NEG_COUNT3=I_COUNT_LENG
*ELSEIF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_COUNT3,EQ,1
POS_COUNT2=
POS_COUNT3=1
I_I%I_NODE_ARCH%_POS_COUNT3=I_COUNT_LENG
```

```
POS_LENG_I%I_NODE_ARCH%_COUNT3=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS_
LENG_I%I_NODE_ARCH%_COUNT3

*ENDIF
*ENDDO
LENG_NEG_I%I_NODE_ARCH%_COUNT0=0
LENG_NEG_I%I_NODE_ARCH%_COUNT1=I_I%I_NODE_ARCH%_NEG_COUNT1*10
LENG_POS_I%I_NODE_ARCH%_COUNT1=I_I%I_NODE_ARCH%_POS_COUNT1*10
LENG_NEG_I%I_NODE_ARCH%_COUNT2=I_I%I_NODE_ARCH%_NEG_COUNT2*10
LENG_POS_I%I_NODE_ARCH%_COUNT2=I_I%I_NODE_ARCH%_POS_COUNT2*10
LENG_NEG_I%I_NODE_ARCH%_COUNT3=I_I%I_NODE_ARCH%_NEG_COUNT3*10
LENG_POS_I%I_NODE_ARCH%_COUNT3=I_I%I_NODE_ARCH%_POS_COUNT3*10

*ELSE

I_I%I_NODE_ARCH%_NEG_COUNT1=0
I_I%I_NODE_ARCH%_POS_COUNT1=0
I_I%I_NODE_ARCH%_NEG_COUNT2=0
I_I%I_NODE_ARCH%_POS_COUNT2=0
I_I%I_NODE_ARCH%_NEG_COUNT3=0
I_I%I_NODE_ARCH%_POS_COUNT3=0
*DO,I_COUNT_LENG,1,10,1
*IF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_LENG_I%I_NODE_ARCH
%_COUNT_1,EQ,0,THEN
POS_LENG_I%I_NODE_ARCH%_COUNT_1=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS
_LENG_I%I_NODE_ARCH%_COUNT_1
POS_COUNT1=1
I_I%I_NODE_ARCH%_POS_COUNT1=I_COUNT_LENG
*ELSEIF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_COUNT1,EQ,1
NEG_COUNT1=1
I_I%I_NODE_ARCH%_NEG_COUNT1=I_COUNT_LENG
NEG_LENG_I%I_NODE_ARCH%_COUNT_1=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG
_LENG_I%I_NODE_ARCH%_COUNT_1
*ELSEIF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_COUNT1,EQ,1
POS_LENG_I%I_NODE_ARCH%_COUNT_2=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS
_LENG_I%I_NODE_ARCH%_COUNT_2
POS_COUNT1=
POS_COUNT2=1
I_I%I_NODE_ARCH%_POS_COUNT2=I_COUNT_LENG
*ELSEIF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_COUNT2,EQ,1
NEG_COUNT1=
NEG_COUNT2=1
I_I%I_NODE_ARCH%_NEG_COUNT2=I_COUNT_LENG
NEG_LENG_I%I_NODE_ARCH%_COUNT_2=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG
_LENG_I%I_NODE_ARCH%_COUNT_2
*ELSEIF,POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,NEG_COUNT2,EQ,1
POS_LENG_I%I_NODE_ARCH%_COUNT_3=POS_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+POS
_LENG_I%I_NODE_ARCH%_COUNT_3
POS_COUNT2=
POS_COUNT3=1
I_I%I_NODE_ARCH%_POS_COUNT3=I_COUNT_LENG
*ELSEIF,NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%,EQ,1,AND,POS_COUNT3,EQ,1
NEG_COUNT2=
NEG_COUNT3=1
NEG_LENG_I%I_NODE_ARCH%_COUNT_3=NEG_LENG_I%I_NODE_ARCH%_%I_COUNT_LENG%+NEG
_LENG_I%I_NODE_ARCH%_COUNT_3
I_I%I_NODE_ARCH%_NEG_COUNT3=I_COUNT_LENG
*ENDIF
*ENDDO

LENG_POS_I%I_NODE_ARCH%_COUNT0=0
```

```
LENG_POS_I%I_NODE_ARCH%_COUNT1=I_I%I_NODE_ARCH%_POS_COUNT1*10
LENG_NEG_I%I_NODE_ARCH%_COUNT1=I_I%I_NODE_ARCH%_NEG_COUNT1*10
LENG_POS_I%I_NODE_ARCH%_COUNT2=I_I%I_NODE_ARCH%_POS_COUNT2*10
LENG_NEG_I%I_NODE_ARCH%_COUNT2=I_I%I_NODE_ARCH%_NEG_COUNT2*10
LENG_POS_I%I_NODE_ARCH%_COUNT3=I_I%I_NODE_ARCH%_POS_COUNT3*10
LENG_NEG_I%I_NODE_ARCH%_COUNT3=I_I%I_NODE_ARCH%_NEG_COUNT3*10

*ENDIF
*ENDDO

FINISH

!***********************************END************************************

!--------------------------------------------------------------------------
!Author                                          Nazrul Islam
!                                                Lecturer
!                                                Dept. of Civil Engineering
!                                                BUET, Dhaka 1000.
! -------------------------------------------------------------------------
!As a part of the thesis work “Global Optimization of Design Parameters of
!Network Arch Bridges” for the partial fulfillment of M. Sc. in Civil
!Engineering (Structural).
!**************************************************************************
!The following script is the file for lane load generation on deck of
!network arch bridges.
!File Name is                 lane_load_solu.txt
!***********************************BEGIN**********************************

UNIFORM_LANE=9300 !LANE = 0.64 K/FT
CONC_LOAD=-80357  !CONC = 18 KIP

*DO,I_NODE_ARCH,1,9,1

ALLSEL,ALL
/TITLE,LANE LOAD I%I_NODE_ARCH% NEG MOMENT
TIME,19

*IF,NEG_LENG_I%I_NODE_ARCH%_1,EQ,1,THEN

NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT0,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,S
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT1,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT2
ESLN,A
      *ELSE
      *ENDIF
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A
      *ELSE
      *ENDIF
*ELSE

NSEL,S,LOC,Z,0,0
```

```
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,S
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT2
ESLN,A
      *ELSE
      *ENDIF
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT3,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT3,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A
      *ELSE
      *ENDIF
*ENDIF

ESEL,R,ENAME,,154
SFE,ALL,1,PRES,0,UNIFORM_LANE

SELTOL,0.5
NSEL,S,LOC,X,MAX_NEG_MY_LOC_I%I_NODE_ARCH%
NSEL,R,LOC,Z,0,0
!NSEL,R,LOC,Y,1.5,1.5
NODE_CONC_NEG_1=NODE(MAX_NEG_MY_LOC_I%I_NODE_ARCH%,1.5,0)
NSEL,S,NODE,,NODE_CONC_NEG_1
SELTOL,
F,ALL,FZ,CONC_LOAD

ALLSEL,ALL
 ~CFLSWRT
  SFEDELE,ALL,ALL,ALL
FDELE,ALL,ALL

ALLSEL,ALL
/TITLE,LANE LOAD I%I_NODE_ARCH% POS MOMENT
TIME,20

*IF,NEG_LENG_I%I_NODE_ARCH%_1,EQ,1,THEN

NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,S
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A
      *ELSE
      *ENDIF
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT3,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT3,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A
      *ELSE
      *ENDIF
*ELSE
```

```
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT0,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,S
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A
      *ELSE
      *ENDIF
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,2.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A
      *ELSE
      *ENDIF
*ENDIF

ESEL,R,ENAME,,154
UNIFORM_LANE=3050.4     !LANE = .64 K/FT
SFE,ALL,1,PRES,0,UNIFORM_LANE

SELTOL,0.5
NSEL,S,LOC,X,MAX_NEG_MY_LOC_I%I_NODE_ARCH%
NSEL,R,LOC,Z,0,0
!NSEL,R,LOC,Y,1.5,1.5
NODE_CONC_NEG_1=NODE(MAX_NEG_MY_LOC_I%I_NODE_ARCH%,1.5,0)
NSEL,S,NODE,,NODE_CONC_NEG_1
SELTOL,
F,ALL,FZ,CONC_LOAD

ALLSEL,ALL
  ~CFLSWRT
  SFEDELE,ALL,ALL,ALL
FDELE,ALL,ALL

ALLSEL,ALL
/TITLE,LANE LOAD I%I_NODE_ARCH% NEG MOMENT ***BOTH LANE***
!TIME,19

*IF,NEG_LENG_I%I_NODE_ARCH%_1,EQ,1,THEN

NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT0,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,S
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT0,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,A

      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT1,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT2
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT2
```

```
ESLN,A

      *ELSE
      *ENDIF
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A

      *ELSE
      *ENDIF
*ELSE

NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,S
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT1,LENG_NEG_I%I_NODE_ARCH%_COUNT1
ESLN,A

      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT2
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT2,LENG_NEG_I%I_NODE_ARCH%_COUNT2
ESLN,A

      *ELSE
      *ENDIF
      *IF,LENG_POS_I%I_NODE_ARCH%_COUNT3,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT3,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT3,LENG_NEG_I%I_NODE_ARCH%_COUNT3
ESLN,A

      *ELSE
      *ENDIF
*ENDIF

ESEL,R,ENAME,,154

SFE,ALL,1,PRES,0,UNIFORM_LANE

SELTOL,0.5
NSEL,S,LOC,X,MAX_NEG_MY_LOC_I%I_NODE_ARCH%
NSEL,R,LOC,Z,0,0
!NSEL,R,LOC,Y,1.5,1.5
```

```
NODE_CONC_NEG_1=NODE(MAX_NEG_MY_LOC_I%I_NODE_ARCH%,2.0,0)
NODE_CONC_NEG_2=NODE(MAX_NEG_MY_LOC_I%I_NODE_ARCH%,-2.0,0)
NSEL,S,NODE,,NODE_CONC_NEG_1
NSEL,A,NODE,,NODE_CONC_NEG_2
SELTOL,
F,ALL,FZ,CONC_LOAD

ALLSEL,ALL
~CFLSWRT
SFEDELE,ALL,ALL,ALL
FDELE,ALL,ALL

ALLSEL,ALL
/TITLE,LANE LOAD I%I_NODE_ARCH% POS MOMENT ***BOTH LANE***
!TIME,20

*IF,NEG_LENG_I%I_NODE_ARCH%_1,EQ,1,THEN

NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,S
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,A

      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A

      *ELSE
      *ENDIF
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT3,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT3,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT3,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A

      *ELSE
      *ENDIF
*ELSE
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT0,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,S
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_POS_I%I_NODE_ARCH%_COUNT0,LENG_POS_I%I_NODE_ARCH%_COUNT1
ESLN,A
```

```
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT1,LENG_POS_I%I_NODE_ARCH%_COUNT2
ESLN,A

      *ELSE
      *ENDIF
      *IF,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LT,SPAN_ARCH,THEN
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,0.5,3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A
NSEL,S,LOC,Z,0,0
NSEL,R,LOC,Y,-0.5,-3.5
NSEL,R,LOC,X,LENG_NEG_I%I_NODE_ARCH%_COUNT2,LENG_POS_I%I_NODE_ARCH%_COUNT3
ESLN,A

      *ELSE
      *ENDIF
*ENDIF

ESEL,R,ENAME,,154
SFE,ALL,1,PRES,0,UNIFORM_LANE

SELTOL,0.5
NSEL,S,LOC,X,MAX_POS_MY_LOC_I%I_NODE_ARCH%
NSEL,R,LOC,Z,0,0
!NSEL,R,LOC,Y,1.5,1.5
NODE_CONC_POS_1=NODE(MAX_POS_MY_LOC_I%I_NODE_ARCH%,2.0,0)
NODE_CONC_POS_2=NODE(MAX_POS_MY_LOC_I%I_NODE_ARCH%,-2.0,0)
NSEL,S,NODE,,NODE_CONC_POS_1
NSEL,A,NODE,,NODE_CONC_POS_2
SELTOL,
F,ALL,FZ,CONC_LOAD

ALLSEL,ALL
~CFLSWRT
SFEDELE,ALL,ALL,ALL
FDELE,ALL,ALL

*ENDDO

!**********************************END************************************
```