Truss Bridge System Optimization

EGR 7040 *Design Optimization*, Wright State University

Joseph Strzelecki and Admir Makas

Abstract

This report describes the process implemented to efficiently design a truss bridge for use by common foot and vehicular traffic. Design optimization techniques are used to formulate the design problem, generate a solution, and evaluate the effectiveness of the implemented solution. The project goal is to design a bridge spanning a 9 meter (~30 foot) ravine that must sustain external loads while minimizing bridge mass. By minimizing a mass cost function, monetary and material cost reductions can be achieved. The minimum mass of the bridge is constrained by the stress that each truss element of the bridge must withstand. As such an optimum design can be created for minimizing bridge mass subject to the proposed constraints. In order to calculate bridge stresses, finite element analysis (FEA) was implemented in conjunction with the optimization function FMINCON available in MATLAB. This technique was successful in generating an optimum design of the bridge system for different types of steel available in the market. Analysis shows that choosing the lightest design option does not necessarily yield the cheapest design. This result stems from the cost variability between different steel types.

1. Introduction

Trusses are widely used in civil and structural engineering to realize very complex systems. Trusses are incorporated in many structures such as bridges, buildings, airplane fuselages, and wings as seen in Figure 1. The term truss was originally derived from the French term *trousse,* which means a *collection of things bound together* [1]. In simplest terms a truss is a two-force member that allows for external load and reactions to act upon the ends of the truss members [2, 3, 4]. A truss system is comprised of a collection of individual truss elements that form the structure [2, 3, 4]. No moment loads are sustained by the truss members, thus the joints at the member ends are assumed to be revolute. This condition is especially important for straight truss members.

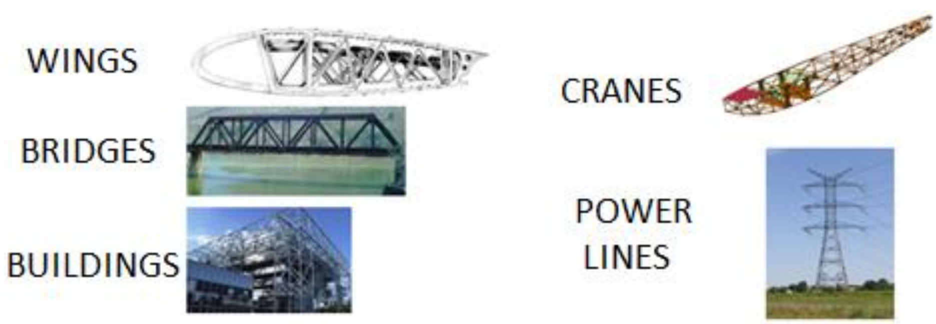
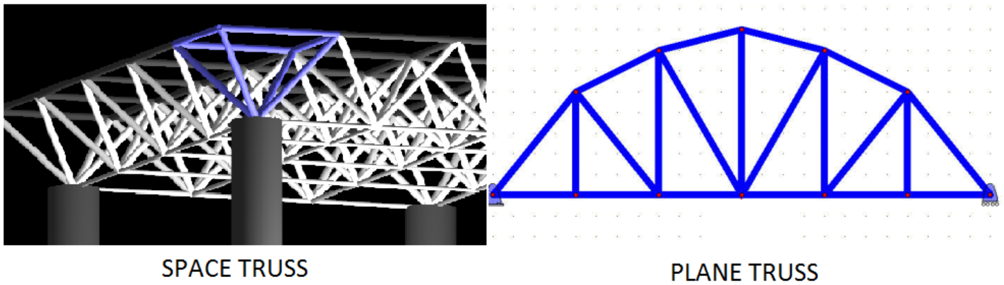


Fig.1. Visual examples of common truss applications.

All truss systems can be generally categorized into two groups: plane and space trusses. Plane trusses are defined in a two-dimensional plane. This definition is appropriate when all the loads act in the plane either horizontally or vertically, and any out-of-plane reactions are negligible. On the other hand, space trusses are defined in three-dimensional space and are necessary when out of plane loads need to be accounted for [2, 3]. An example of typical space and plane truss configurations can be seen in Figures 2a and 2b respectively.



(a) (b)

Fig. 2. Visualization of a 3-D space truss (a) and 2-D plane truss (b).

Design optimization principles can be applied to minimize the mass and cost of the structure while maintaining sufficient strength and stiffness. While simple in nature, bridges can be very complex structural systems that require a multi-disciplinary approach to achieve a successful product. In general, bridge design considerations are subject to loading requirements, thermal expansion, vibrations due to wind loading, stiffness, and environmental degradation.

2. Problem Description

Following section describes the bridge dimensions chosen for the optimization study along with loading requirements. A brief section will be devoted to outlining the finite element analysis (FEA) method, which is utilized to calculate stresses in the truss members. Next, materials chosen for the analysis are briefly discussed. Finally, the optimization problem is defined and solution methodology is identified.

2.1 Bridge Definition

The bridge chosen for the study comprises of truss members formed into equilateral triangles. This type of truss design is more commonly known as a *Warren Truss* [2, 3, 4]. The length of each truss element is set to be 3 meters long. This is a typical truss dimensions found in engineering statics texts [4]. Using the specified dimension yields a bridge of 9 meters in length. This relatively short bridge design is intended to span a small ravine or river. The intent of the proposed bridge is to allow crossing of commercial and private vehicles along with any potential foot traffic. Due to loading and stiffness requirements, the goal is to design a bridge that can hold 5000 kN applied at two locations on the mid-span. A diagram of the proposed bridge, truss elements, and loads can be seen below in Figure 3.

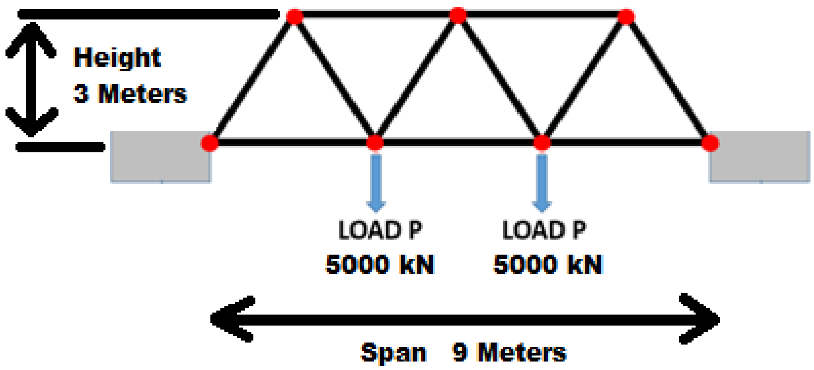


Fig. 3. Bridge dimensions

The proposed bridge consists of 11 truss members and 7 joints. Since the applied load acts vertically in the plane, it is appropriate to assume a planar truss analysis. Furthermore, the span is relatively short at 9 meters so any cross-wind loading is insignificant when compared to the vertical loading requirements. As such it is safe to assume that the out of plane loading is negligible.

2.2 Finite Element Analysis Overview

Finite element analysis is widely used across a broad range of engineering fields including but not limited to structural, acoustic, thermal, fluid, material and electrical engineering. In general, if there exist governing partial differential equations (PDE's) for a particular phenomenon then FEA technique can be applied. For the proposed problem, FEA analysis is applied using truss element formulation to calculate truss member stresses. This is a critical step since the necessary cross-sectional area of each truss element depends on the calculated stresses.

Presented next is a short overview of truss element formulation used for the FEA analysis. Since the problem is of planar truss type, the truss elements considered allow for 2-D displacements. Figure 4 illustrates a typical truss element definition.

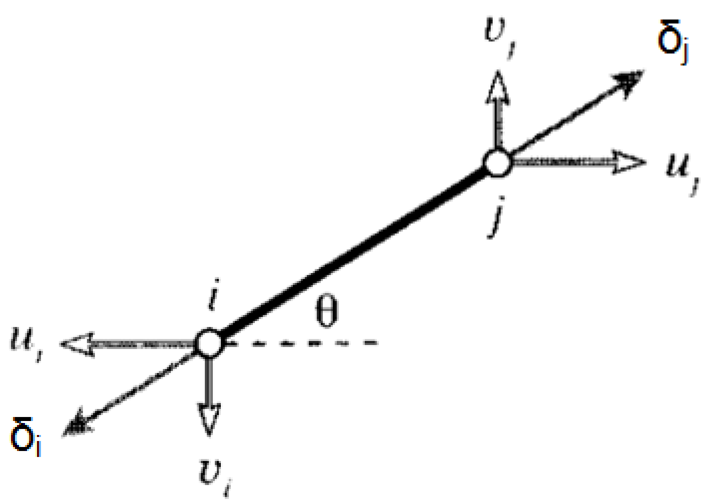


Fig. 4. Truss element definition

As seen in Figure 4, the truss element is allowed to displace at nodes *i* and *j*. The node displacements δ_i and δ_j are along the local element coordinates, requiring one of the basis vectors to be along the element length. The goal is to resolve the nodal displacement in local coordinates to global coordinates. This is done via the rotation matrix as seen in Figure 5. In other words, displacement of the nodes is resolved along directions u and v, which are defined in global coordinates.

$$\delta = (u_j \cos\theta + v_j \sin\theta) - (u_i \cos\theta + v_i \sin\theta)$$

$$\Downarrow$$

$$\delta = \begin{bmatrix} -c & -s & c & s \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

Where $\quad c = \cos\theta$ and $s = \sin\theta$

Fig. 5. Node displacement defined in global coordinates.

Once the nodal displacements are expressed in global coordinates it is possible to formulate the relationship between nodal displacements and forces as $\bar{K}\bar{U} = \bar{F}$. In this tensor notation, $\bar{K}$is denoted as the element stiffness matrix and comprises of physical properties that enable a linear mapping between displacement vector $\bar{U}$ and force vector $\bar{F}$. Explicitly this equation can be expressed as shown in Figure 6 [5, 6].

$$\begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

Fig. 6. Truss element force displacement relationship.

Terms A, E, and L correspond to cross-sectional area, Young's Modulus, and element length respectively. At this stage, it is possible to define the above expression for all the elements in the model (in this case 11 elements). Once each element is defined per equation in Figure 6, each of the 11 expressions is assembled into the global stiffness matrix. When assembling the global stiffness matrix that represents the entire structure, it is important that force equilibrium and displacement compatibilities are maintained. Finally the problem can be solved by inverting the global stiffness matrix to obtain following expression that solves for nodal displacements $\bar{U} = \bar{K}^{-1}\bar{F}$. Knowing the displacements allows for calculation of elements strains that are used to calculate element stresses via constitutive relationship (i.e. Young's Modulus, E).

The FEA method was implemented in MATLAB and the subject code can be seen in Appendix A. Before using the FEA solver for optimization, its accuracy was confirmed using commercial FEA software ABAQUS. Model geometry follows the definition in Figure 3, and Table 1 lists the material and geometrical properties used.

Parameter	**Model Properties**
Young's Modulus, E	200 GPa
Poisson's Ratio, ν	0.30
Cross-Section Area	0.020 m^2
Element Length	3 m
Load, L	5000 kN

Table 1. Model properties.

Steel material properties were obtained from a popular mechanics of materials textbook, and geometrical properties were derived from problem definition in Figure 3 [7]. Table 2 lists stress results from the MATLAB solver and ABAQUS. Letter "C" denotes truss elements in compression and "T" identifies elements in tension.

Stress Results for Matlab and ABAQUS analysis		
Element #	**Matlab Stress**	**ABAQUS Stress**
1	288.7 MPa, C	288.7 MPa, C
2	288.7 MPa, C	288.7 MPa, C
3	48.1 MPa, C	48.1 MPa, C
4	96.2 MPa, T	96.2 MPa, T
5	48.1 MPa, C	48.1 MPa, C
6	288.7 MPa, C	288.7 MPa, C
7	288.7 MPa, T	288.7 MPa, T
8	0 MPa	0 MPa
9	0 MPa	0 MPa
10	288.7 MPa, T	288.7 MPa, T
11	288.7 MPa, C	288.7 MPa, C

Table 2. FEA stress results for MATLAB and ABAQUS solvers.

From the results in Table 2, it was determined that the accuracy of the MATLAB implementation was well within an acceptable margin of error in comparison to the results from ABAQUS. To better visualize the stress results, Figure 7 illustrates node displacements and truss element stresses from ABAQUS as a 2-D plane truss diagram with each element labeled.

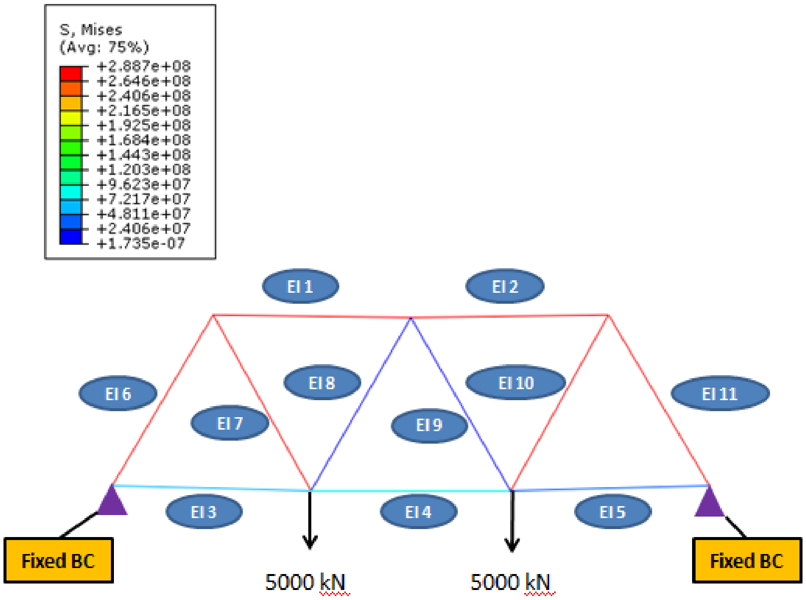


Fig 7. ABAQUS plot results.

An FEA model for the proposed bridge system is illustrated in Figure 8, which shows model nodes elements and boundary conditions (displacement, load).

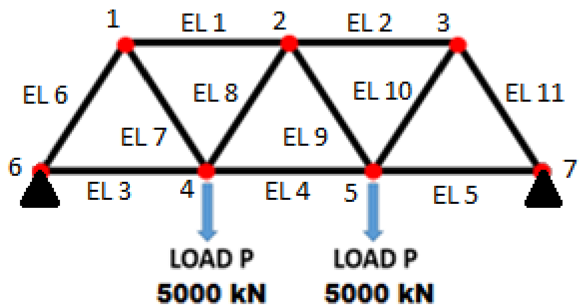


Fig 8. FEA model for bridge design optimization.

2.3 Material Selection

In civil engineering, steel is the overwhelmingly common choice of material used for truss bridge construction [8]. Typically, steel alloy grades can be specified via chemical composition or mechanical strength properties such as yield strength (σ_y). Yield strength is a material property that defines a stress limit at which permanent deformation occurs [7]. Since yield strength in this study is the only relevant material constraint on the bridge design, the following steel grades were chosen as seen in Table 3.

Steel Data		
Steel Nomenclature	**Yield Strength**	**Price per Ton**
270 Steel	270 MPa	$550.00
340 Steel	340 Mpa	$650.00
420 Steel	420 Mpa	$700.00
550 Steel	550 Mpa	$950.00

Table 3. Steel properties and costs [9, 10].

The values listed in the table above were obtained through material specifications and current steel prices listed by vendors online [9, 10]. It should be noted that steel prices can vary significantly in short period of time according to fluctuations in supply and demand so quoted prices above may not be accurate now compared to the time of this study. Other material properties required are listed in Table 4 [7, 9]. The material properties listed below are similar for the same family of steels that were chosen for this project.

	Model Properties
Young's Modulus, E	200 GPa
Poisson's Ratio, ν	0.30
Density, ρ	7850 $\frac{kg}{m^3}$

Table 4. Additional material properties [7, 9]

2.4 Optimization Problem Statement

Cost function is expressed as the minimization of the total mass of the bridge, which is the summation of each individual truss members defined below:

$$M = \rho \sum_{i=1}^{11} A_i L_i, \text{ where}$$

- M = Bridge mass, kilograms
- A_i = Cross-section area of each truss (**design variable**), *meters*^{2}
- L_i = Length of each truss element (**constant**), *meters*
- ρ = material density (**constant**), $\frac{kg}{m^3}$

Truss element lengths remain constant since the size of the bridged ravine is not changing. Therefore, the design variable in the formulation is the cross-sectional areal of each truss element. The structure is required to withstand a load of 5000 kN at two locations on the mid-span as illustrated previously in Figure 3 while requiring that material response remain elastic, ensuring that no permanent deformation is present in the bridge. As such the constraints require that each truss element not exceed yield stress σ_y in either compression or tension to avoid permanent deformation. Yield limits can be found in Table 3. Stress constraints are defined below:

$$\sigma_{T_i} = \frac{\sigma_i}{\sigma_y} \leq 1, \text{ Tensile stress constraint}$$

$$\sigma_{C_i} = \frac{-\sigma_i}{\sigma_y} \leq 1, \text{ Compression stress constraint}$$

- σ_{T_i} = Element tensile stress constraint
- σ_{C_i} = Element compressive stress constraint
- σ_i = Element stress, Pa
- σ_y = Allowable element stress, in this case yield strength, Pa

Since the structure is comprised of 11 truss elements, 22 total stress constraints (2 per element) are considered. Design variable A_i is used in the stress calculation as defined in Figure

6. Additionally, in order to have a feasible design, limits must be imposed on the design variable A_i to ensure that truss elements are not unrealistically thick. Furthermore, negative values are not allowed. This side constraint is simply expressed as:

$$LB \leq A_i \leq UB$$

Finally the problem can be stated in standard form seen below:

Minimize:

$$M = \rho \sum_{i=1}^{11} A_i L_i$$

Subject to:

$$\sigma_{T_i} = \frac{\sigma_i}{\sigma_y} \leq 1$$

$$\sigma_{C_i} = \frac{\sigma_i}{\sigma_y} \leq 1$$

$LB \leq A_i \leq UB$, A_i bounds: [0.0001 – 0.070]m^2

The minimum bound for A_i was selected to be a truss of 10x10 mm. This signifies that trusses of smaller dimension are not capable of bearing load. On the opposite end, max truss dimension is set to 265x265 mm. For this design problem, truss thickness greater than 265x265 mm represents the cross-sectional area large enough that is too difficult to handle and connect to neighboring trusses during construction.

While the cost function is linear, the corresponding stress constraints are not. Therefore, MATLAB function FCONMIN was used for the optimization procedure. This method is appropriate for the problem definition above with an acceptable computational cost. No gradient functions were provided at the initial input. Rather, it was decided to allow the FMINCON subroutine to calculate the appropriate gradients using the finite difference procedure.

3. Optimization Results

The results obtained from the FMINCON subroutine are shown in the following sections for individual truss elements and for different bridge material types.

3.1 Truss Element Results

To start the optimization procedure, an initial guess was required for the cross-sectional area of the individual truss elements. The initial A_i chosen was 0.02 m^2 for each element. The optimization subroutine was executed for each of the materials listed in Table 3. The iteration history for the four runs is summarized in Figure 9. As seen in Figure 9, each optimization run converged in 11 iterations in a smooth manner with no zig-zag patterns visible. This implies good convergence performance with respect to the defined constraints.

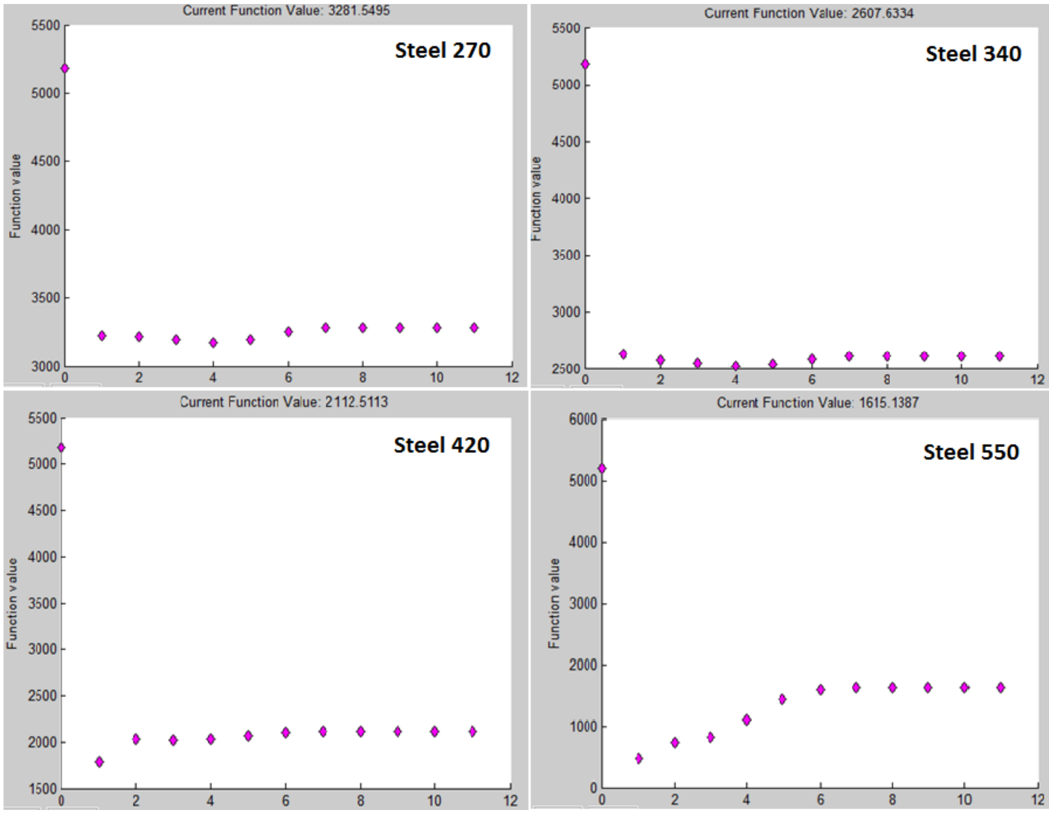


Fig 9. Optimization iteration history

Next, Table 5 shows the optimized A_i values from the FEA solver. It can be observed that as steel yield strength is increased, optimized cross-section areas decrease in value. This result is expected since stronger materials can withstand higher stresses before plasticity occurs.

Optimized Truss Element Cross-Sectional Area					
Element ID	**Initial Guess (m^2)**	**Steel 270 (m^2)**	**Steel 340 (m^2)**	**Steel 420 (m^2)**	**Steel 550 (m^2)**
1	0.020	0.021	0.017	0.014	0.011
2	0.020	0.021	0.017	0.014	0.011
3	0.020	0.0001	0.0001	0.0001	0.0001
4	0.020	0.011	0.008	0.007	0.005
5	0.020	0.0001	0.0001	0.0001	0.0001
6	0.020	0.021	0.017	0.014	0.011
7	0.020	0.021	0.017	0.014	0.011
8	0.020	0.0001	0.0001	0.0001	0.0001
9	0.020	0.0001	0.0001	0.0001	0.0001
10	0.020	0.021	0.017	0.014	0.011
11	0.020	0.021	0.017	0.014	0.011

Table 5. Optimized values for A_i for 4 steel variants.

Each of the material columns has only 3 distinct area values. This is attributed to the geometric and loading symmetry present in the bridge design. It is also important to note that while the optimized result indicates elements 3, 5, 8, and 9 have very small *A*, they cannot be removed entirely from the bridge design since undesirable asymmetries and instabilities are introduced. These elements can however be made very thin. Finally, it can be observed that optimized cross-section areas are within the defined bounds of the optimization problem. The next variable of interest is the resultant stresses in the truss members, which can be viewed in Table 6.

Truss Elements Stresses expressed as 1.0*e^8 Pa					
Element ID	**Initial Stress**	**Steel 270**	**Steel 340**	**Steel 420**	**Steel 550**
1	-2.887	-2.700	-3.400	-4.200	-5.500
2	-2.887	-2.700	-3.400	-4.200	-5.500
3	-0.481	-1.350	-1.700	-2.100	-2.750
4	0.962	2.700	3.400	4.200	5.500
5	-0.481	-1.350	-1.700	-2.100	-2.750
6	-2.887	-2.700	-3.400	-4.200	-5.500
7	2.887	2.700	3.400	4.200	5.500
8	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000
10	2.887	2.700	3.400	4.200	5.500
11	-2.887	-2.700	-3.400	-4.200	-5.500

Table 6. Stress results from the optimization output.

Results in Table 6 show that as constraint limits are increased for higher strength steel variants, the optimizer succeeds at reaching the constraint boundary for a selected group of elements (elements 1, 2, 4, 6, 7, 10, 11). In fact, the stress constraints are active for the aforementioned elements. Stress results shown in Table 6 also support results in Table 5, which show that a very small stress response in a truss element can allow a near-zero cross-sectional area (elements 3, 5, 8, 9). It should be noted that negative signs in Table 6 denote a truss element in compression.

Table 7 contains constraint values, which serve to confirm that the achieved optimum design is feasible. Constraint values are tabulated for tension and compression. The goal is to have the constraint value be ≤ 0. Any positive value signifies a violated constraint. Out of the four steel types, only steel 270 starting design violated the constraint requirements as indicated by the positive values. The remaining steel grades had no starting design constraint violations.

Stress Constraint Values								
	Steel 270 Const. Values		**Steel 340 Const. Values**		**Steel 420 Const. Values**		**Steel 550 Const. Values**	
	Initial	**Final**	**Initial**	**Final**	**Initial**	**Final**	**Initial**	**Final**
Tensile	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
	-1.18	-1.50	-1.14	-1.50	-1.11	-1.50	-1.09	-1.50
	-0.64	0.00	-0.72	0.00	-0.77	0.00	-0.83	0.00
	-1.18	-1.50	-1.14	-1.50	-1.11	-1.50	-1.09	-1.50
	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00
	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00
	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
Compression	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00
	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00
	-0.82	-0.50	-0.86	-0.50	-0.89	-0.50	-0.91	-0.50
	-1.36	-2.00	-1.28	-2.00	-1.23	-2.00	-1.18	-2.00
	-0.82	-0.50	-0.86	-0.50	-0.89	-0.50	-0.91	-0.50
	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00
	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
	-2.07	-2.00	-1.85	-2.00	-1.69	-2.00	-1.52	-2.00
	0.07	0.00	-0.15	0.00	-0.31	0.00	-0.48	0.00

Table 7. Constraint values

No constraints are violated at the final design for all steel grades. Note that values of zero indicate active constraints.

3.2 Bridge Results

Knowing that the design optimization results are feasible, cost values for the specified steel grades were used in conjunction with final mass values to calculate bridge costs. These results are tabulated in Table 8.

Bridge Optimization Summary			
Alloy Grade	**Starting Mass (kg)**	**Ending Mass (kg)**	**Cost ($)**
270	5181.0	3281.6	1805.00
340	5181.0	2607.6	1695.00
420	5181.0	2112.5	1479.00
550	5181.0	1615.0	1534.00

Table 8. Mass and cost results.

Final results show that steel grade 550 yields the lightest design. This result is expected since steel 550 has the highest yield strength and requires the least amount of cross-sectional area per truss element to resist stresses; however, a steel grade 550 bridge did not yield the most cost-efficient option. Steel 420 as a building material, while being heavier, yields the cheapest design option at $1479.00. This is a result not immediately discernable without additional cost analysis. Steel grades 340 and 270 produce the costliest design due to the mass penalty incurred in order to maintain elastic behavior.

A visual depiction of the resulting truss element cross-sectional area for the chosen steel 420 design is shown in Figure 10. Elements colored in orange indicate truss elements with active constraints. Elements colored in green have inactive stress constraints.

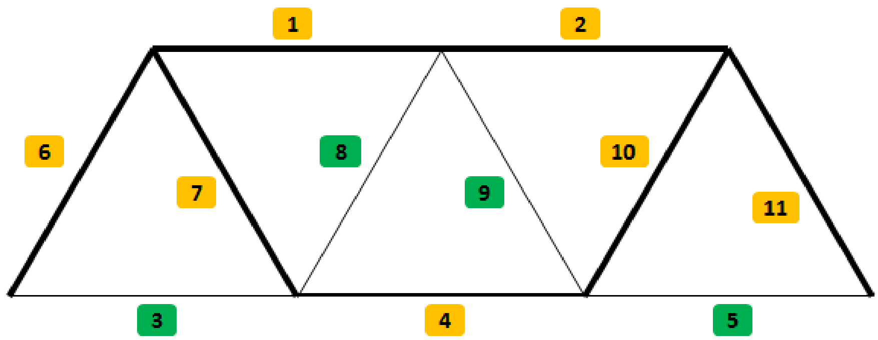


Fig. 10. Visualization of optimized truss element cross-sectional area and depiction of active and inactive constraints for the chosen steel 420 bridge.

As seen in Figure 10, those truss elements with active stress constraints tend to be thicker in cross-sectional area than their inactive stress constraint counterparts.

4. Post-optimality Analysis

In the previous section, it was shown that the constraint conditions have been met, and the optimal design is within the feasible region. This section briefly discusses post-optimality analysis performed on the results obtained.

4.1 Stress Constraint Sensitivity Analysis

For the present analysis, Lagrange multipliers are utilized to study the sensitivity of the design with respect to the constraint limits. Subroutine FMINCON has the capability of numerically approximating the Lagrange multipliers. Table 9 lists the truss element Lagrange multipliers for all the material selections studied.

Lagrange Multipliers				
Element ID	**Steel 270**	**Steel 340**	**Steel 420**	**Steel 550**
1	503.58	399.91	323.74	247.20
2	503.58	399.91	323.74	247.20
3	0.00	0.00	0.00	0.00
4	251.80	199.96	161.87	123.60
5	0.00	0.00	0.00	0.00
6	503.58	399.91	323.73	247.20
7	503.58	399.91	323.73	247.20
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	503.58	399.91	323.73	247.20
11	503.58	399.91	323.73	247.20

Table 9. Truss element Lagrange multipliers for all material cases.

Results from Table 9 are used to study the impact to bridge mass and cost if the stress constraint limits are relaxed. Table 9 shows that Lagrange multipliers exist only for elements with active constraints. This result makes sense because inactive constraint elements have enough slack that prevent any change to the cost function with respect to these truss elements. It is also noteworthy to mention that the Lagrange multipliers exhibit the same symmetry as the stress and cross-section area results. Furthermore, Table 9 shows a decrease in Lagrange multiplier values with respect to increase in material yield strength. This implies that the bridge mass is less sensitive to constraint changes as material yield strength is increased.

4.2 Yield Strength Sensitivity Analysis

Hypothetically, if steel variants 445 and 470 were available on the market, sensitivity analysis can be used to determine the impact to the bridge mass and cost. Steel 420 results are used for this sensitivity analysis although a similar procedure can be carried out for the remaining steel variants. Yield strength is increased by 25 MPa and 50 MPa with respect to the baseline yield strength of 420 MPa. The change to the cost function (bridge mass) can be calculated by the following equation [11].

$$\delta f = -\sum u_j^* \, e_j$$

- δf = Change in cost function
- u_j^* = Lagrange multiplier
- e_j = Constraint variation

Using the appropriate Lagrange multipliers along with the expression above, mass and cost impact is calculated for the optimum design using steel 420. Results are tabulated in Table 10. For comparison, optimal mass and cost for steel 420 are also tabulated in the first row of Table 10.

Sensitivity Analysis for Steel 420			
$\Delta\sigma_y$ Mpa	$\Delta Mass$ (kg)	*Mass (kg)*	*Cost ($)*
0	0	2112.5	1479.00
25	-125.3	1987.2	1391.10
50	-250.5	1862	1303.40

Table 10. Sensitivity study results for the chosen steel 420 bridge design.

Cost for the new designs were calculated assuming steel 420 pricing. If steel 445 was available on the market at the same price as steel 420, bridge mass would decrease by 125.3 kg and cost would decrease by $87.90. If steel 470 was available on the market at the same price as steel 420, bridge mass would decrease by 250.5 kg and cost would decrease by $175.60.

5. Optimization Summary

Optimization analysis of a conventional truss bridge was carried out. Stress in the truss members was calculated via FEA technique that was employed inside MATLAB. Stress results obtained from MATLAB were confirmed by ABAQUS CAE software, which is a commercial package widely used in academia and industry.

An FEA solver was used in conjunction with MATLAB function FMINCON to optimize a truss bridge design with 11 truss members as seen in Figure 10 with two 5000 kN loads as shown in Figure 3. Four steel grades were chosen for the study, each of which had its respective yield strength value. Stress constraints employed in the optimization analysis utilized individual yield strength values to define feasible design regions. Each optimization simulation required 11 iterations to converge, which was smooth in nature. Due to geometrical and loading symmetries, optimization results were also symmetric. Final optimization results have active stress constraints for certain group of elements (elements 1, 2, 4, 6, 7, 10, 11). This was true for all steel grades. Additionally, the optimized results identified truss elements with low stresses (elements 3, 5, 8, 9). In these cases, their respective cross-section area was reduced to the lower limit of the side constraint. Low stress truss elements should remain in the bridge structure in order to ensure structural stability. However, their respective cross-section area footprint should be very small. Constraint values for all the design options were ≤ 0, which implies that optimization results remained in feasible space.

Optimized bridge mass for all steel grades were used to calculate total material costs. While steel grade 550 achieved lightest design and therefore least steel material usage, due to material costs it was not the most cost efficient. A steel grade 420 bridge, while heavier than a 550 steel bridge, had the lowest material costs at $1479.00 vs $1534.00 respectively. This result is not immediately recognizable without performing additional material cost analysis. Steel grades 270 and 340 were the cheapest options in terms of material cost; however, their resultant bridge mass to compensate for element stresses was enough to make these options the costliest in terms of total monetary costs. Finally, sensitivity analysis was carried out on the steel 420 optimization results. It was determined that if yield strength is increased by 25 MPa and 50 MPa respectively, the corresponding weight reduction of the truss bridge is 125.3 kg and 250.5 kg. Bridge cost would decrease by $87.90 and $175.60 respectively if such materials were priced the same as steel 420. Sensitivity analysis also determined that bridge design is more sensitive when weaker steel variant are used.

In closing, a truss bridge optimization procedure was carried out with the aid of an FEA solver in MATLAB. The project goal was to obtain the cheapest bridge design, which was achieved successfully.

6. References

[1] D. Harper, "Online Etymology Dictionary," 2015. [Online]. Available: http://www.etymonline.com/index.php?term=truss. [Accessed 20 December 2015].

[2] M. E. Plesha, G. L. Gray and F. Costanzo, Engineering Mechanics: Statics, 2nd ed., New York: McGraw-Hill Companies, 2013.

[3] F. Ching, A Visual Dictionary of Architecture, 2nd ed., Hoboken: Wiley, 2012.

[4] R. C. Hibbeler, Engineering Mechanics-Statics, 3rd ed., New York: Macmillan Publishing Co., 1983.

[5] Y. W. Kwon and Bang Hyochoong, The Finite Element Method Using MATLAB, Boca Raton: CRC Press LLC, 1997.

[6] D. Rolyance, "Finite Element Analysis (Course Notes)," 28 February 2001. [Online]. Available: http://ocw.mit.edu/courses/materials-science-and-engineering/3-11-mechanics-of-materials-fall-1999/modules/fea.pdf. [Accessed 20 December 2015].

[7] R. C. Hibbeler, Mechanics of Materials, 6th ed., Lebanon: Prentice Hall, 2004.

[8] A. K. and L. Stonecypher, "Truss Bridge Designs," 14 July 2015. [Online]. Available: http://www.brighthubengineering.com/structural-engineering/63635-truss-bridge-designs/. [Accessed 20 December 2015].

[9] Matweb LLC, "Matweb," 20 December 2015. [Online]. Available: http://www.matweb.com/. [Accessed 20 December 2015].

[10] Alibaba Group, "Alibaba," 20 December 2015. [Online]. Available: http://www.alibaba.com/. [Accessed 20 December 2015].

[11] J. S. Arora, Introduction to Optimum Design, Waltham: Academic Press, 2011.

Appendix A - M-file to initiate optimization procedure:

```
function [] = solve(x)

clear;
clc;
%format short g;

area = 0.02.*ones(11,1); %initial area definition [11 by 1]

Lb = 0.0001 * ones (11,1); %lower bound on x-sec area
Ub = 0.070 * ones (11,1); %upper bound on x-sec area

options=optimset('Algorithm', 'interior-point', 'MaxIter', 1E5, 'MaxFunEvals', 1E5, 'TolCon', 1E5, ...
    'PlotFcns', @optimplotfval);

x0 = area; %initial truss element x-sec areas [11 by 1]

%fmincon call for optimization
[x, fval] = fmincon(@CostFunc, x0, [], [], [], [], Lb, Ub, @ConsFun, options)

%additional outputs
[area, x] %vector output of original area and optimized areas [11 by 2]
[FEA_Solve(area, 'stress'), FEA_Solve(x, 'stress')] %vector output of original truss stress and optimized stress [11 by 2]
[FEA_Solve(area, 'displacement'), FEA_Solve(x, 'displacement')] %vector output of original truss displacement and optimized displacement [11 by 2]
[CostFunc(area), CostFunc(x)] %vector output of starting and optimized costFun values
[ConsFun(area), ConsFun(x)] %vector output of stress constraints for each truss 1-11=tension 12-22=compression [22 by 2]

%COST FUNCTION DEFINITION
function f = CostFunc(x)
density = 7850; %material density definition
L = [3.0; 3.0; 3.0; 3.0; 3.0; 3.0; 3.0; 3.0; ...
    3.0; 3.0; 3.0]; %truss element lenghts
f = density * (x(1)*L(1) + x(2)*L(2) + x(3)*L(3) + x(4)*L(4) + x(5)*L(5) + ...
x(6)*L(6) + x(7)*L(7) + x(8)*L(8) + x(9)*L(9) + x(10)*L(10) + x(11)*L(11)); %cost function defined in terms of design variables

%CONSTRAINT FUNCTION DEFINITION, NO GRADIENTS SPECIFIED
function [g, h] = ConsFun(area)

stress = FEA_Solve(area, 'stress'); %element stresses [11 by 1]
sigma_max = 550e6;
g = [stress./sigma_max-1; -stress./sigma_max - 1]; %inequality constraint definitions
h = [];
```

Appendix B – FEA Solver:

```
function [output] = FEA_Solve(area, output_request)
% DEFINIG NODES AND ELEMENTS
nodeDef = [1.5 2.598; 4.5 2.598; 7.5 2.598; 3 0; 6 0; 0 0; 9 0]; %[7 by 2] matrix
elementDef = [1 2; 2 3; 6 4; 4 5; 5 7; 6 1; 1 4; 4 2; 2 5; 5 3; 3 7]; %[11 by 2] matrix

%area = 0.02.*ones(11,1); %initial area definition [10 by 1]

% MECHANICAL PROPERTIES OF ELEMENTS
youngMS = 200e9;
DOF = 2;

% DEFINE NODAL BOUNDARY CONDITIONS
% DISPALCEMENT( deltaU) = [NODE_NUMBER X_DISPLACEMENT Y_DISPLACEMENT]
% FORCE = [NODE_NUMBER F_X F_Y]
DispBC = [6 0 0; 7 0 0]; %displacement BCs [2 by 3]
force = [4 0 -5000000; 5 0 -5000000]; %force BCs [2 by 3]
numNode = size(nodeDef, 1); %scalar = 6
numEl = size(elementDef, 1); %scalar = 11
youngM = youngMS.*ones(numEl, 1); %Young's Modulus defined for all 10 elements [11 by 1]

% CALCULATING LENGTH OF EACH ELEMENT AND ALSO ITS ANGLE TO HORIZION
elLength = elementLength(nodeDef, elementDef, numEl);
elAngle = elementAngle(nodeDef, elementDef, numEl);

% CALCULATE GLOBAL STIFFNESS MATRIX
gsMat = GSM(elementDef, numNode, numEl, youngM, area, elLength, elAngle, DOF, DispBC);
% generates [8 by 8] matrix

% PUT THE FORCE MATRIX IN GLOBAL FORM
F = [];
for i = 1:size(force, 1) %goes from 1 to 2 since there are 2 loads applied
    F = [F; force(i, 2:3)']; %makes column force vector for containing applied load
end
tMat = zeros(size(gsMat, 1), size(F, 1));
for i = 1:size(force, 1) %For loop ensures that load values are in correct position withing the force matrix
    tMat(force(i ,1)*2 - 1, 2*i - 1) = 1;
    tMat(force(i ,1)*2, 2*i) = 1;
end
F = tMat*F; %creates [8 by 1] vector

% DISPLACEMENT SOLVER
nodeDisp = inv(gsMat)*F;
U = nodeDisp;
nodeDisp = [nodeDisp; zeros(4, 1)];
% STRESS SOLVER
elStress = feaStress(elementDef, numEl, nodeDisp, elLength, youngM, area, elAngle);
if(strcmp(output_request, 'stress'))
    output = elStress;
elseif(strcmp(output_request, 'displacement'))
    output = nodeDisp;
end
```

Appendix C - Additional Support Files:

```
%ASSEMBLE GLOBAL STIFFNESS MATRIX
function globalStiffnessMatrix = GSM(elementDef, numNode, numEl, youngM, area, elLength, elAngle, DOF, DispBC)
globalStiffnessMatrix = zeros(numNode * DOF, numNode * DOF); %matrix of zeros [12 by 12]
elStifness = zeros(numEl, 1); %element stiffness matrix initiated [10 by 1]
for i=1:numEl
    elStifness(i) = youngM(i) * area (i) / elLength(i);
end
for i=1:numEl %individual element global stiffness matrices calculated and added together to construct total global stiffness matrix [14 by 14]
    phi = elAngle(i);
    l2gMat = zeros(numNode * DOF, numNode * DOF);
    nodeStart = elementDef(i, 1);
    nodeEnd = elementDef(i, 2);

    rMat = [cosd(phi) sind(phi) 0 0 ; ...
    -sind(phi) cosd(phi) 0 0; ...
    0 0 cosd(phi) sind(phi); ...
    0 0 -sind(phi) cosd(phi)]; %rotation matrix defined here
    elsMat = rMat' * (elStifness(i) * [1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0 0]) * rMat; %rotated element stiffness matrix, total of 10 of them [4 by 4]

    %4 lines below add a "1" in the appropriate location within l2gMat,
    %this is used to generate a [14 by 14] matrix correspondig to a
    %specific truss element
    l2gMat(2 * nodeStart - 1, 2 * nodeStart - 1) = 1;
    l2gMat(2 * nodeStart, 2 * nodeStart) = 1;
    l2gMat(2 * nodeEnd - 1, 2 * nodeEnd - 1) = 1;
    l2gMat(2 * nodeEnd, 2 * nodeEnd) = 1;

    %reconstructs l2gMat by keeping only non-zero columns
    l2gMat(:, ~any(l2gMat, 1)) = []; %columns

    elsMat = l2gMat * elsMat * l2gMat'; %Finally generate [12 by 12] stiffness matrix for subject truss element
    globalStiffnessMatrix = globalStiffnessMatrix + elsMat; %add up all the individual global truss element stiffness matrices to create global stiffness matrix
end

tMat = eye(size(globalStiffnessMatrix, 1)); %create [124 by 14] identity matrix
for i =1:size(DispBC, 1)
    tMat(DispBC(i, 1)*2-1, DispBC(i, 1)*2-1) = 0;
    tMat(DispBC(i, 1)*2, DispBC(i, 1) * 2) = 0;
end
tMat(:, ~any(tMat, 1)) = [];
globalStiffnessMatrix = tMat' * globalStiffnessMatrix * tMat;
```

```
%CALCULATE STRESS FROM DISPLACEMENT DATA
function [elStress] = feaStress(elementDef, numEl, nodeDisp, elLength, youngM, area, elAngle)
stress = zeros(numEl, 1);
sigma = [];
for i=1:numEl
    startNode = elementDef(i, 1);
    endNode = elementDef(i, 2);
    sigma = [sigma; ...
    youngM(i) / elLength(i) * ...
    [cosd(elAngle(i)) sind(elAngle(i)) -cosd(elAngle(i)) -sind(elAngle(i))] * ...
    [nodeDisp(2*startNode-1); ...
    nodeDisp(2*startNode); ...
    nodeDisp(2*endNode-1, 1); ...
    nodeDisp(2*endNode)]];
end
elStress = -sigma;
```

```
%CALCULATE ELEMENT ANGLE
function[el_Angle] = elementAngle(nodeDef, elementDef, numEl)
el_Angle = zeros(numEl, 1);
for i=1:numEl
    pointStart = elementDef(i, 1);
    pointEnd = elementDef (i, 2);
    x1 = nodeDef(pointStart, 1);
    y1 = nodeDef(pointStart, 2);
    x2 = nodeDef(pointEnd, 1);
    y2 = nodeDef(pointEnd, 2);
    el_Angle(i, 1) = atand((y2-y1)/(x2-x1)) ;
end
```

```
%CALCULATE ELEMENT LENGTH
function [el_Length] = elementLength(nodeDef, elementDef, numEl)
el_Length = zeros(numEl, 1);
for i =1:numEl
    pointStart = elementDef(i, 1);
    pointEnd = elementDef(i, 2);
    x1 = nodeDef(pointStart, 1);
    y1 = nodeDef(pointStart, 2);
    x2 = nodeDef(pointEnd, 1);
    y2 = nodeDef(pointEnd, 2);
    el_Length(i, 1) = sqrt(power(x2-x1, 2) + power(y2-y1, 2));
end
```