

Truss Topology Using Linear Programming

Pawan S. Sawant



University of Colorado
Boulder

MCEN 5125: Optimal Design
Prof. Shalom D. Ruben

March 4, 2024

Abstract

In this report, we will be optimizing the truss topology problem using linear programming(LP) using two different constraints, first by minimizing the force and second by minimizing the length. Since the given problem has 11 rows and 20 columns, it has a lot of complexity, we will first try to find the solution for a 2x4 grid problem, to reduce the complexity and then we will use the pattern that we will be getting in this small problem to solve the complex problem. The solution we get in this optimization problem shows the best geometry for the given problem that we found by solving using *linprog()* to find the internal forces of all the possible truss members.

Contents

1	Introduction	4
2	Problem Statement	4
3	Methodology:	5
3.1	Simplifying the Problem	5
3.2	Finding Internal Forces	5
3.2.1	Finding the Angular Relation	6
3.3	Equilibrium of Forces	8
3.4	Given Constraints	10
3.5	Cost Function	10
3.5.1	Optimization by Minimizing Total Internal Forces	10
3.5.2	Optimization by Minimizing Length and Total Internal Forces	11
3.6	Generating the A_{eq}, A_{ineq}, b_{eq} and b_{ineq} matrices	12
3.6.1	Equality Constraints: A_{eq} and b_{eq}	12
3.6.2	Inequality Constraints: A_{ineq} and b_{ineq}	13
4	Solution	14
4.1	Solution for Total Internal Forces	14
4.2	Solution for Length and Total Internal Forces	15
5	Conclusion	16

List of Figures

1	Truss System with 11 rows and 20 columns and given anchors.	4
2	simplified Problem.	5
3	Force resolution.	8
4	Optimization solution for Total Internal Forces	14
5	Optimization solution for Total Internal Forces	15

1 Introduction

Linear Programming is an optimization method which is used to find the optimal solution for the linear function, this linear function is called the cost function. Linear Programming is used to solve complex real-world problems by defining the required cost function and the given constraints. In this project, we will be solving two truss topology problems with two specific cost functions. In the first method, we will be minimising the total internal forces of all the truss members and in the second method, we will be minimizing the length of all the possible truss members. Now that we have cost functions, we have to define the constraints given in the problem.

2 Problem Statement

The given Truss system has 11 rows and each row has 20 columns. The grid can be shown in Figure 1 as we can see the force is acting on the node of 5th row and 20th column with a magnitude of 4. It is also given that the cross-section of all the truss members is a unit area and yield strength of 8. Also, given truss anchors are $[4,0]$, $[5,0]$, $[6,0]$.

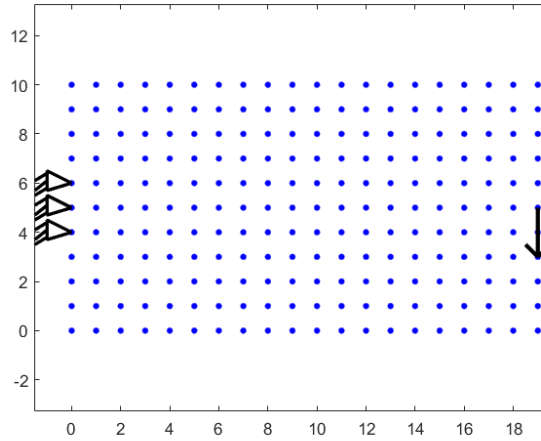


Figure 1: Truss System with 11 rows and 20 columns and given anchors.

For the given truss system, we will have to use linear programming to optimize the truss topology problem. We will be using l_1 norm for optimization.

3 Methodology:

3.1 Simplifying the Problem

Since the given problem is extremely complex, we will first simplify the given problem to find the necessary pattern in formulating the given problem. Then we will apply this formulation to the bigger and complex problem. For this reason 2 shows the simplified problem that we we first be analysing.

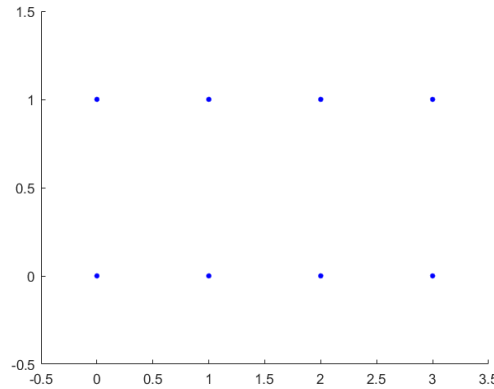


Figure 2: simplified Problem.

As you can see in the 2, the simplified problem has a truss system with 3 rows and 4 columns. This simplification will significantly reduce the complexity of the bigger problem and we will be able to find the required pattern to solve this problem. One way to show the complexity is that the larger problem has a total of 220 nodes (11x20 row x columns), therefore there are $\binom{n}{k}$ possible combinations of truss members in this truss system. Therefore there are a total of 24090 possible truss members in this truss system. But if we simplify the given problem into 2 rows and 4 columns, there are 8 total nodes in this truss system and a total possible truss members of 28.

3.2 Finding Internal Forces

Now that we have simplified the problem, we will be finding the internal forces in all the truss elements. To find the forces, first, we need to find the angles of every possible truss member. For formulating the matrices, let m be the total number of rows and n be total number of columns in the truss system.

3.2.1 Finding the Angular Relation

In order to find the angles of each possible truss member, we will first have to find the y-direction and x-direction distances of the truss member's starting node and respective ending node. To find this distance, we will need two matrices, first, we will need a coordinate matrix and another will subtracting matrix we we will subtract from the coordinate matrix to get the corresponding x and y distances. Therefore the coordinate matrix for x and y position are as follows:

$$x_{coordinates} = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \end{bmatrix}_{8 \times 8} ; y_{coordinates} = \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}_{8 \times 8}$$

The matrix of $x_{coordinates}$ has values from 1 to 4, which corresponds to 4 columns in the truss systems. The appending this 1 to 4 pattern two times corresponds to the 2 rows in the truss system. Let us now generalise this formulation.

$$x_{temp} = \begin{bmatrix} 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 2 & \cdots & n \end{bmatrix}_{(m*n) \times n}$$

$$y_{temp} = \begin{bmatrix} 1 & \cdots & 1_{n-times} & 2 & \cdots & 2_{n-times} & n & \cdots & n_{n-times} \\ 1 & \cdots & 1_{n-times} & 2 & \cdots & 2_{n-times} & n & \cdots & n_{n-times} \\ 1 & \cdots & 1_{n-times} & 2 & \cdots & 2_{n-times} & n & \cdots & n_{n-times} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \cdots & 1_{n-times} & 2 & \cdots & 2_{n-times} & n & \cdots & n_{n-times} \end{bmatrix}_{(m*n) \times n}$$

Now we can formulate the generalised for for $x_{coordinates}$ and $y_{coordinates}$ as follows:

$$x_{coordinates} = \begin{bmatrix} x_{temp} & x_{temp} & \cdots & x_{temp} \end{bmatrix}_{(m*n) \times (m*n)}$$

$$y_{coordinates} = \begin{bmatrix} y_{temp} \\ y_{temp} \\ \vdots \\ y_{temp} \end{bmatrix}_{(m*n)x(m*n)}$$

Now that we have the coordinate matrix, to find the distances we need to subtract a subtracting matrix from the coordinate matrix. For the smaller problem, the matrices are formulated as follows:

$$x_{subtract} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}_{8x8} ; y_{subtract} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}_{8x8}$$

As we can see from the $x_{subtract}$ and $y_{subtract}$, they are transpose of the $x_{coordinates}$ and $y_{coordinates}$ respectively. Therefore they can be written as follows:

$$x_{subtract} = x_{coordinates}^T \quad (1)$$

$$y_{subtract} = y_{coordinates}^T \quad (2)$$

Now to find the x-distances and y-distance, we just subtract the subtract matrices from the coordinate matrices. This can be formulated as follows:

$$x_{distance} = x_{coordinates} - x_{subtract} \quad (3)$$

$$y_{distance} = y_{coordinates} - y_{subtract} \quad (4)$$

Now, using the inverse tangent function we can calculate the angle of each truss member with its starting and ending node. This matrix of angles can be used to find the force equation. The resultant theta matrix can be written as follows:

$$\begin{bmatrix} \theta_{1,2} & \theta_{1,2} & \cdots & \theta_{1,mn} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,mn} \\ \theta_{3,1} & \theta_{3,2} & \cdots & \theta_{3,mn} \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{m,1} & \theta_{m,2} & \cdots & \theta_{m,mn} \end{bmatrix}_{mn \times mn}$$

3.3 Equilibrium of Forces

In order for a truss system to be in equilibrium, it is essential for the sum of all the forces in the x-direction and y-direction to be zeros. The resolution of forces in the x and y directions can be seen in Figure 3.

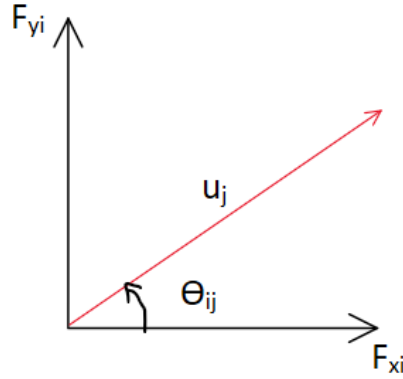


Figure 3: Force resolution.

This equilibrium condition is applied to every node of the truss system, except the anchor point as it is assumed that the anchor node provides necessary restoring forces to achieve the equilibrium. The equilibrium condition can be written as follows:

$$\sum_{i=1}^{mn} \underbrace{u_j \begin{bmatrix} \cos \theta_{ij} \\ \sin \theta_{ij} \end{bmatrix}}_{\text{internal forces}} + \underbrace{\begin{bmatrix} F_{xi} \\ F_{yi} \end{bmatrix}}_{\text{external load}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

or

$$\sum_{i=1}^{mn} \underbrace{u_j \begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\text{if bar } j \text{ is not connected}} + \underbrace{\begin{bmatrix} F_{xi} \\ F_{yi} \end{bmatrix}}_{\text{external load}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In the equation above, $\theta_{i,j}$ represents the angle of the truss member from node i to j.

Since, the external forces is only applied to a single point, the force equation can be written as follows:

$$\sum_{i=1}^{mn} u_j \cdot \cos \theta_{ij} = 0 \quad (5)$$

$$\sum_{i=1}^{mn} u_j \cdot \sin \theta_{ij} = 0 \quad (6)$$

Now let us consider the smaller problem that we formulate earlier, and find the theta matrix for it then we can formulate the generalised matrix for the bigger problem. The equation of force With respect to node 1, can be written as follows:

$$u_1 [\cos \theta_{1,2} + \cos \theta_{1,3} + \cdots + \cos \theta_{1,8} + \underbrace{\cos \theta_{2,1} + \cos \theta_{2,3} + \cdots + \cos \theta_{8,8}}_{\text{all these terms are zero w.r.t node 1}}] = 0 \quad (7)$$

Similarly, we can do it for all the nodes. We can generate the theta matrix as follows:

$$\theta = \begin{bmatrix} \theta_{1,2} & \theta_{1,3} & \cdots & \theta_{1,8} & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \theta_{2,1} & 0 & \cdots & 0 & \theta_{2,3} & \cdots & \theta_{2,8} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \theta_{3,1} & 0 & \cdots & 0 & 0 & 0 & \theta_{3,2} & \cdots & \theta_{3,8} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & \theta_{8,1} & 0 & \cdots & \theta_{8,2} & 0 & 0 & \theta_{8,3} & 0 & \cdots & 0 & \theta_{mn,mn-1} \end{bmatrix}_{220 \times 24090} \quad (8)$$

Now that we have the theta matrix, we can generate this matrix for the sin and cosine matrix. Since we won't be considering the anchor point forces, we will be deleting the corresponding rows for the anchors for their x and y direction forces. Therefore the resultant theta matrix will be of the size 217x24090.

3.4 Given Constraints

Since the resultant internal forces in the truss member are either in tension or compressions. We can formulate this constraints as follows:

$$u_i \geq 0 \therefore -u_i \leq 0 \quad (9)$$

$$u_i \leq 0 \quad (10)$$

We are also been provided with the yield strength of the material that we will be using hence we can use this as a constraint in the problem.

$$-S_y.A \leq u_i \leq S_y.A \quad (11)$$

Since the given area of the member is a unit area, we can write the constraints as follows:

$$u_i \leq S_y \quad (12)$$

$$-u_i \leq S_y \quad (13)$$

3.5 Cost Function

3.5.1 Optimization by Minimizing Total Internal Forces

To minimize the total internal forces, we can write it as follows:

$$\min ||u_i||_1$$

$$\min ||u_1|| + ||u_2|| + \dots + ||u_i||$$

$$\max(u_1, -u_1) + \max(u_2, -u_2) + \dots + \max(u_i, -u_i)$$

Therefore, this can be formulated as follows:

$$\begin{array}{ll} u_1 \leq t_1 & -u_1 \leq t_1 \\ u_2 \leq t_2 & -u_2 \leq t_2 \\ \vdots & \vdots \\ u_i \leq t_i & -u_i \leq t_i \end{array}$$

Moving knowns to RHS and Unknowns to LHS,

$$\begin{array}{rcl} u_1 - t_1 \leq 0 & - & u_1 - t_1 \leq 0 \\ u_2 - t_2 \leq 0 & - & u_2 - t_2 \leq 0 \\ & & \vdots \\ u_i - t_i \leq 0 & - & u_i - t_i \leq 0 \end{array}$$

From the above equations, we can write them in inner product form as follows:

$$\underbrace{\begin{bmatrix} \vec{1} & \vec{1} \end{bmatrix}}_{\hat{c}} \underbrace{\begin{bmatrix} \vec{x} \\ \vec{t} \end{bmatrix}}_{\hat{x}} \leq \underbrace{\begin{bmatrix} 0 \downarrow \\ 0 \downarrow \end{bmatrix}}_{\hat{b}} \quad (14)$$

Therefore the problem is formulated as follows:

$$\begin{array}{ll} \text{minimize} & \hat{c}^T \hat{x} \\ \text{subject to} & \hat{A}_{eq} \hat{x} = \hat{b}_{eq} \\ & \hat{A}_{ineq} \hat{x} \leq \hat{b}_{ineq} \end{array} \quad (15)$$

3.5.2 Optimization by Minimizing Length and Total Internal Forces

To minimize the total internal forces, we can write it as follows:

$$\min ||l_i u_i||_1$$

$$\min ||l_1 u_1|| + ||l_2 u_2|| + \dots + ||l_i u_i||$$

$$\max(l_1 u_1, -l_1 u_1) + \max(l_2 u_2, -l_2 u_2) + \dots + \max(l_i u_i, -l_i u_i)$$

Therefore, this can be formulated as follows:

$$\begin{array}{rcl} l_1 u_1 \leq t_1 & - & l_1 u_1 \leq t_1 \\ l_2 u_2 \leq t_2 & - & l_2 u_2 \leq t_2 \\ & & \vdots \\ l_i u_i \leq t_i & - & l_i u_i \leq t_i \end{array}$$

Therefore the problem is formulated as follows:

$$\begin{aligned}
& \text{minimize } \hat{c}^T \hat{x} \\
& \text{subject to } \hat{A}_{eq} \hat{x} = \hat{b}_{eq} \\
& \quad \hat{A}_{ineq} \hat{x} \leq \hat{b}_{ineq}
\end{aligned} \tag{16}$$

Moving knowns to RHS and Unknowns to LHS,

$$\begin{aligned}
l_1 u_1 - t_1 &\leq 0 & -l_1 u_1 - t_1 &\leq 0 \\
l_2 u_2 - t_2 &\leq 0 & -l_2 u_2 - t_2 &\leq 0 \\
&\vdots & &\vdots \\
l_i u_i - t_i &\leq 0 & -l_i u_i - t_i &\leq 0
\end{aligned}$$

l_i is the length of each possible truss element. We can calculate each length using the Pythagorean theorem. This can be formulated as $l_i = \sqrt{x_{\text{distance}}^2 + y_{\text{distance}}^2}$.

From the above equations, we can write them in inner product form as follows:

$$\underbrace{\begin{bmatrix} \vec{l}_i & \vec{1} \end{bmatrix}}_{\hat{c}} \underbrace{\begin{bmatrix} \vec{x} \\ \vec{t} \end{bmatrix}}_{\hat{x}} \leq \underbrace{\begin{bmatrix} 0 \downarrow \\ 0 \downarrow \end{bmatrix}}_{\hat{b}} \tag{17}$$

Therefore the problem is formulated as follows:

$$\begin{aligned}
& \text{minimize } \hat{c}^T \hat{x} \\
& \text{subject to } \hat{A}_{eq} \hat{x} = \hat{b}_{eq} \\
& \quad \hat{A}_{ineq} \hat{x} \leq \hat{b}_{ineq}
\end{aligned} \tag{18}$$

3.6 Generating the A_{eq}, A_{ineq}, b_{eq} and b_{ineq} matrices

3.6.1 Equality Constraints: A_{eq} and b_{eq}

Since the given internal forces equilibrium is not an inequality constraint, it is an equality constraint. We can get this constraint for the force balance equation. Therefore the constraint is written as follows:

$$\underbrace{A_{eq}}_{\text{all the angles}} \underbrace{x}_{\text{all the internal forces}} = \underbrace{b_{eq}}_{\text{external forces}} \tag{19}$$

Where,

$$A_{eq} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \quad (20)$$

The equations above represent the equilibrium state of the truss system in x and y direction. The above matrix is of size 434x48180, where the $\sin \theta$ and $\cos \theta$ are of size 434x24080 and zero matrix of 434x24090.

$$b_{eq} = \begin{bmatrix} 0 \\ F_y \\ 0 \end{bmatrix} \quad (21)$$

Where the b_{eq} matrix is of size 434x1 and the force F_y is applied at the given location.

3.6.2 Inequality Constraints: A_{ineq} and b_{ineq}

From the given inequality constraints in equations (9),(10),(12) and (13), we can form the inequality constraints. Thus, by formulating the given constraint we can write it as:

$$A_{ineq} = \begin{bmatrix} I & -I \\ -I & I \\ I & 0 \\ -I & 0 \end{bmatrix} \quad (22)$$

$$b_{ineq} = \begin{bmatrix} 0 \\ 0 \\ S_y \\ -S_y \end{bmatrix} \quad (23)$$

Where A_{ineq} has a size of 96360x48180 and b_{ineq} has a size of 96360x1.

4 Solution

To solve this optimization problem, we will use Matlab's inbuilt function *linprog()*. The important thing about the *linprog()* function is that it can solve equality and inequality constraints.

4.1 Solution for Total Internal Forces

By solving optimization using *linprog()* for minimizing the total internal forces, we got a result with total truss elements of 14 members. Figure 4 shows the optimal solution for the total internal forces.

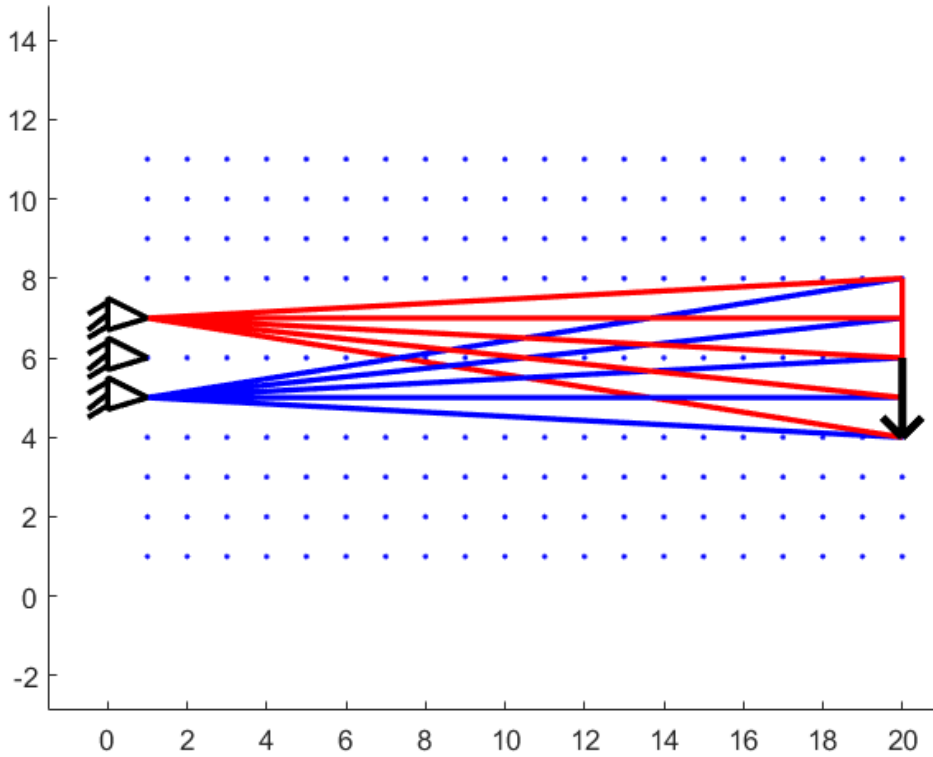


Figure 4: Optimization solution for Total Internal Forces

4.2 Solution for Length and Total Internal Forces

By solving optimization using *linprog()* for minimizing the length and total internal forces, we got a result with total truss elements of 189 members. Figure 5 shows the optimal solution for the total internal forces. This solution look more interesting as it look like a teardrop shape. In this solution, more members are generated near the anchor points.

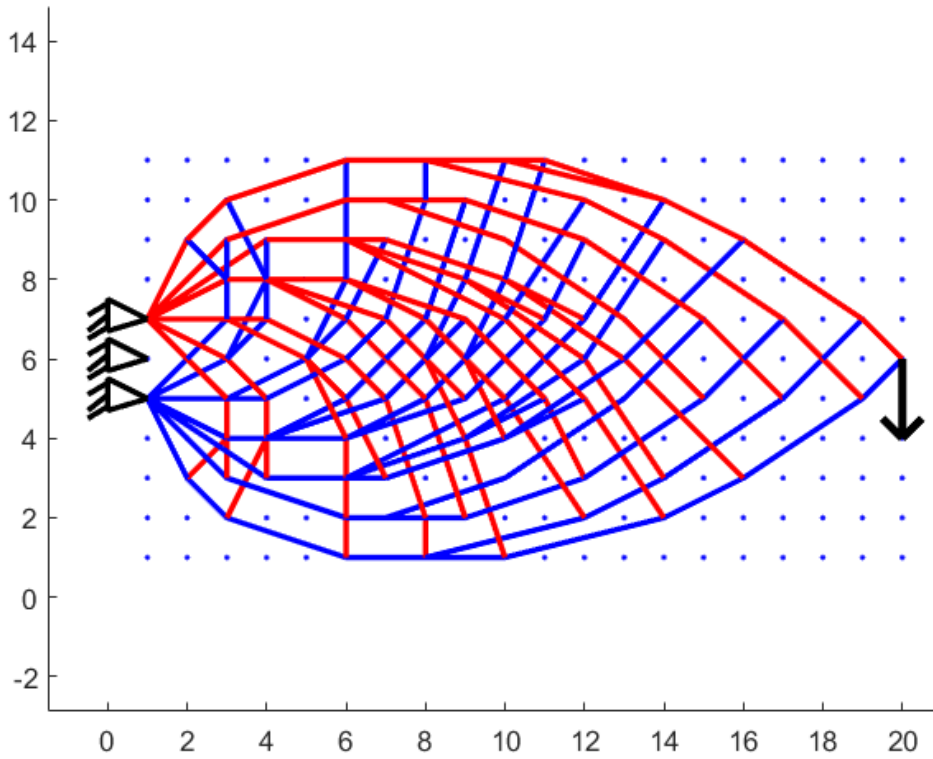


Figure 5: Optimization solution for Total Internal Forces

5 Conclusion

The presented problem was a complex structural optimization problem that optimizes the truss topology problem. The given complex problem has a total of 24090 possible truss elements, There are two types of constraints given in this problem, the first is the equality constraint which we can formulate using the equilibrium of forces in x-direction and y-direction. The second constraint is inequality constraint, which is formulated from the tension and compression of the internal forces and the yield strength. We have to find the optimization using two methods using l_1 norm, first is the minimizing the total weight, where we minimise the sum of all the internal forces, second method is minimizing the sum of the length and internal forces. The length of all the possible elements is calculated using the Pythagorean theorem. By solving using Matlab's inbuilt function *linprog()*, we get an optimized solution of 14 members in weight and 189 in optimizing the length and weight.