Contents

List of Figures

List of Tables

Chapitre 1 | Introduction to truss

A truss is an assemblage of long, slender structural elements or beams that are connected at their ends to create a firm structure.
Trusses find substantial use in modern construction, for instance as towers, bridges, scaffolding, etc.

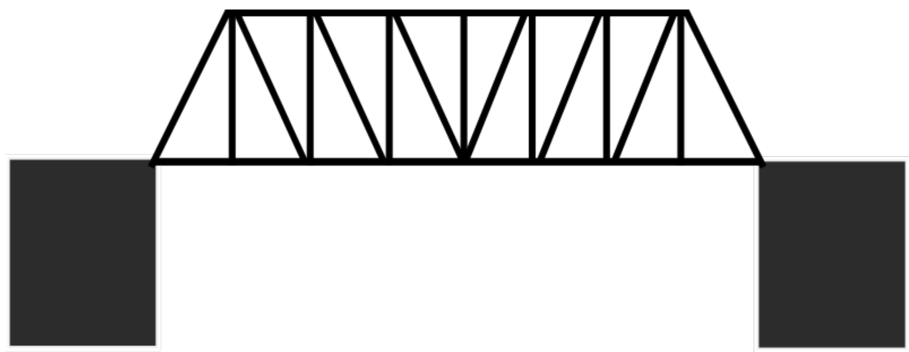

Figure 1.0.1: Truss bridge.

The scope of this project is limited to the truss contained in a plan and so does the force applied on it.

Chapitre 2 | Truss Deformation Analysis

The aim of this section is to provide an mathematical formula to calculate the deformation energy of a truss.
To do so, we will start by an elementary study of a beam delimited by two nodes. Then we'll generalize the approach to a whole truss.

2.1 Elementary Analysis

Let i and j two nodes delimiting a beam, (x_i, y_i) and (x_j, y_j) their coordinates respectively.
Let A be the cross-sectional area of the beam, E is Young's modulus and N the normal effort applied at i and at j.

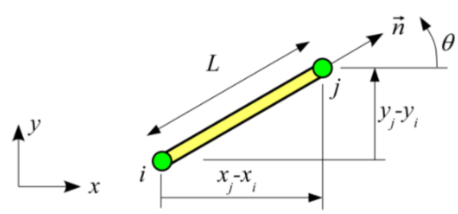

Figure 2.1.1: A beam delimited by i and j, aligned along $\vec{n}$.

We have the following formulas:

$$L^2 = (x_j - x_i)^2 + (y_j - y_i)^2 \tag{2.1.1}$$

$$\vec{n} = \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} = \frac{1}{L}\begin{Bmatrix} x_j - x_i \\ y_j - y_i \end{Bmatrix} \tag{2.1.2}$$

Being subject to the force $\vec{N} = N\vec{n}$, the two nodes i and j move slightly from their initial positions.
The new position of the node i is :

$$\begin{Bmatrix} x_i + u_i \\ y_i + v_i \end{Bmatrix}$$

Likewise, the new position of j is:

$$\begin{Bmatrix} x_j + u_j \\ y_j + v_j \end{Bmatrix}$$

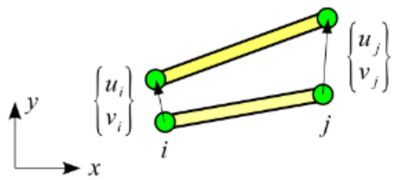

Figure 2.1.2: The new position of the beam after being subject to $\vec{N}$.

Using Hooke's law, we deduce the following formulas:

$$f_{nod} = k\,u - f_{th} \tag{2.1.3}$$

Where:

- $t = [-n_x \;\; -n_y \; n_x \; n_y]^T$
- $f_{nod} = N\,t$
- $u = [u_i \; v_i \; u_j \; v_j]^T$
- $$k = \frac{EA}{L}\,t\,t^T \tag{2.1.4}$$
- $f_{th} = EA\alpha\Delta T$

Note: In the scope of this project, we'll focus on a truss subject to mechanical forces. In other words, $\Delta T = 0$ for every beam.

Finally, the energy of deformation[1] is:

$$E_{def} \propto \frac{1}{2}u^T k u - u^T f_{th} \tag{2.1.5}$$

2.2 Truss Analysis

The truss is defined by the positions of the nodes $P_i = (x_i, y_i)$, the adjacency matrix M, where each entry is a non number denoting the cross-sectional area of the beam between the nodes i and j: $M = (a_{ij})$, E the Young's Modulus and the vector of forces applied on each node $f_i = (f_i^x, f_i^y)$.

The first step to calculate the deformation energy is to assemble the model, in other words, the *global stiffness matrix* **K** and the *global force column* **F** must be calculated. Then, we conclude the energy of deformation of the truss.

[1]The energy is proportional to that quantity and the coefficient is independent from the displacements and their derivatives.

Assembly

The following algorithm describes the necessary instructions:

1. Let F be: $F_i = (f_i^x, f_i^x)^T$. It's equivalent to vertically stacking the components of the force applied on each node, one at a time.

2. Let K be 2n by 2n zero matrix where n is the number of nodes.

3. For every beam in the truss defined by nodes i and j having $A = a_{ij} > 0$.

   - Calculate L, $\vec{n}$, k using the formulas 2.1.1, 2.1.2, 2.1.4 respectively.
   - Let t=2i, 2i+1, 2j, 2j+1 , and $K_t = (K_{ij})_{i \in t, j \in t}$ be the sub-matrix containing the intersection of the rows indexed by t and the column indexed by t.
     Finally, $K_t \leftarrow K_t + k$.

Displacements

Let **free** be a set such as:

$$\begin{cases} 2i \in free & \text{if the node i is free to move along axis x.} \\ 2i+1 \in free & \text{if the node i is free to move along axis y.} \end{cases}$$

Let U be a vector, U_{free} is the vector containing the rows existing in *free*. To calculate the vector of displacements u, we use the formula 2.1.3. Formally[2]:

$$\begin{cases} u_{free} = (K_{free})^{-1} F_{free} \\ \text{the rest are 0} \end{cases}$$

2.2.1 Deformation energy

Finally, we can approximate the energy of deformation using the formula 2.1.5.

[2] Reminder: In this project, we limit the study to $f_{th} = 0$.

Chapitre 3 | Optimization Algorithm

This project's aim is to find the optimal truss, which has the lowest energy of deformation, using the finite element method.

To do so, we will use **Genetic Algorithm**, a global optimization algorithm, to search through the the space of trusses.

3.1 Genetic Algorithm

This algorithm is inspired from the evolution of human beings (individuals).
It's mainly a four steps algorithm that get repeated a certain number of times (generations) over a set of possible solutions (population).
However, the most crucial phase is encoding/modeling of the individual. How the individuals (decision variables) are modeled and encoded can affect the overall performance of the algorithm, in terms of improvement we get after each iteration. This phase needs to be carefully processed.
Examples of modelings: binary coding, read-coding, matrix form, vector form ...etc.

Now, let's dig in the four steps of algorithm:

1. **Fitness Evaluation**: The calculation of the fitness function of each individual. It's a predefined function that depends on the optimization problem and its objective function.

2. **Selection**: It's the process of choosing individuals to perform the next operator on them. The selection operators favor fit individuals (having the greatest fitness values) over weak ones. Example of operators: Roulette Wheel Selection, Tournament Selection, Rank Selection.

3. **Crossover**: It consists of matching a couple of individuals (called parents) and transmit their characteristics to two new individuals. Some well-know crossover operators are: one-point, two-point, uniform crossover operator.

4. **Mutation**: It represents the process of randomly modifying arbitrary characteristics (could be a bit, number...) of each individual.

Note: It's worth to note that the aforementioned operators aren't universal, they need to be selected based on the problem.

These GA phases are defined with respect some hyper-parameters:

- **Crossover rate**: It's the probability of apply the crossover operator on a parent or just duplicate it.

- **Mutation rate**: It's the probability of apply the mutation operator on an individual.
- **Population size**: The total number of the individuals.
- **Stopping criterion**: Genetic algorithm is an iterative algorithm, so the definition of a stopping criteria is legitimate.
  Some stopping criteria: a *fixed number* of generations, variance of fitness is less than a *predefined* constant, stability of fitness function, convergence of individuals.

These hyper-parameters need to be fine-tuned to reach the global optimal individual in the best case, or at least avoid reaching a poor local optimum.

3.2 Case Study: Truss optimization

In this section, we'll specify how to apply the above algorithm on our optimization problem.

The decision variables are the cross-sectional area of each beam delimited by two nodes. The individual is represented by an upper triangular matrix, where each entry represents the cross-sectional area of the corresponding beam (row node, column node).
The optimization problem is defined with respect to some parameters and constraints:

- Position of nodes is fixed.
- Degrees of freedom of each node.
- Young's modulus.
- Force applied on each node.
- *Economic constraint*: since the thicker the beam the expensive it is, the volume of all the truss beams must be exactly 1.
  $\sum_i L_i A_i = 1$

The fitness function of each individual is the inverse of the energy of deformation:

$$fitness(x) = \frac{1}{E_{def}(x)}$$

The GA operators adopted in our study are the following:

Selection

The operators opted for are: Rank selection and Fitness proportionate election operators.

Crossover

Since our individuals are modelled as matrices, we define the Horizontal One-point crossover operator as: it horizontally splits an individual matrix at a given row.
In a similar way, we define the Vertical One-point crossover operator.
Moreover, the uniform crossover operator is to be used.

Mutation

For each entry matrix, with a mutation rate/probability, a standard Gaussian perturbation is applied to that entry. Formally:

$$a_{ij} \leftarrow a_{ij} + \mathcal{N}(0, 1)$$

However, if an entry becomes negative, it is shifted to zero.

Note: In selection and crossover phases, at each generation, we arbitrarily use one of the aforementioned operators.

Chapitre 4 | Experimental Results

4.1 Data

The above method and algorithm are tested on the following truss:

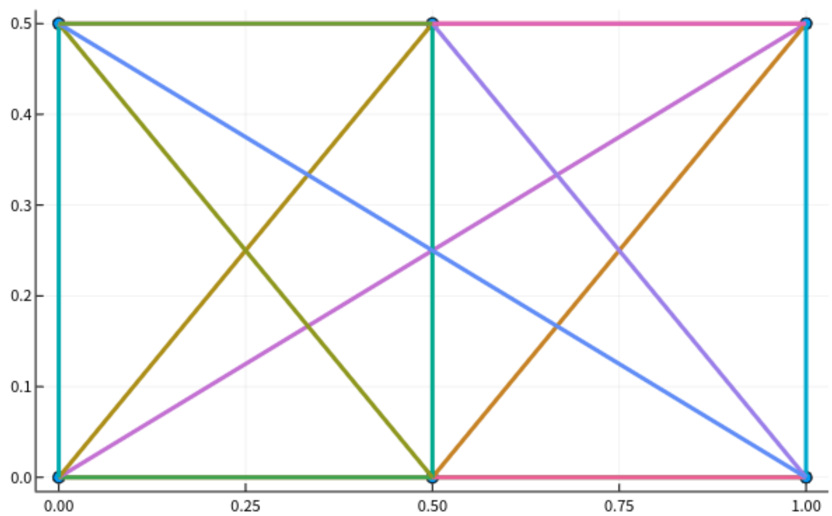

Figure 4.1.1: Test Truss.

The truss parameters are:

- *Nodes positions*: are illustrated in the figure above.
- *Degrees of freedom*: all nodes ,but those at (0,0) and (1,0), are free to move along both axes.
- *Young's Module*: E=20 000.
- *Forces*: only the node (0.5,0) is subject to the force vector (0,-2000).

4.2 GA hyper-parameters

The following table summarizes the values of each hyper-parameter:

Hyper-parameter	Value
Crossover rate	0.4
Mutation rate	0.07
Population size	12
Stopping criterion	number of generations =200 000

Table 4.1: GA hyper-parameters

4.3 Results

The following figure illustrate the evolution of the inverse of the fitness function (the energy of deformation) of the best individual in the population:

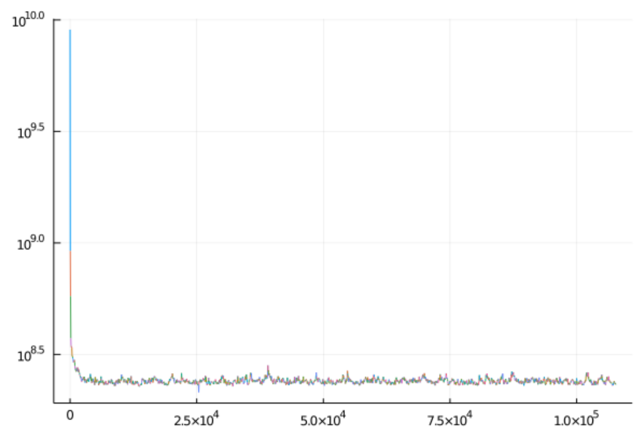

Figure 4.3.1: The evolution of GA with 12 individuals for 100K generations. The 'optimal' energy of deformation is $2.3510^8 J$.

The optimal truss is:

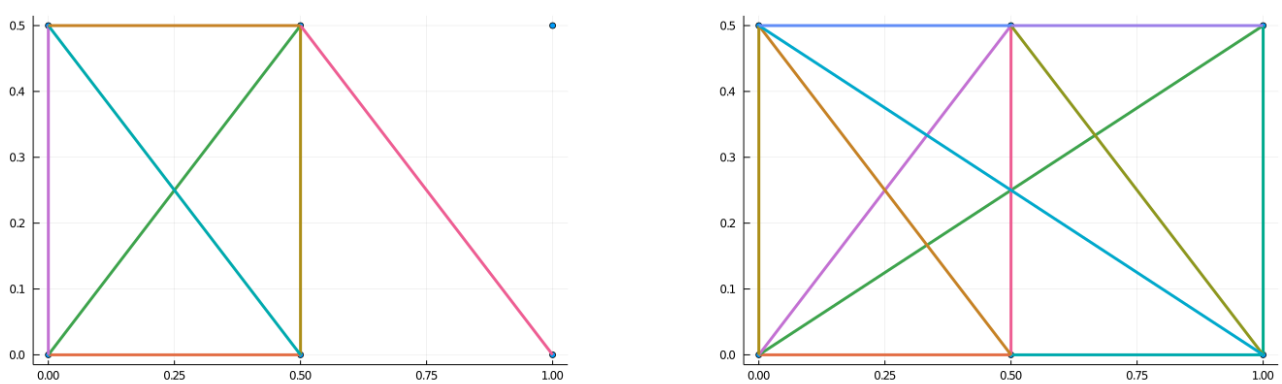

Figure 4.3.2: The optimal truss. The beams with cross section less than 0.01 are eliminated from the left figure.

It's encoded in this upper triangular matrix:

```
6×6 UpperTriangular{Float64,Array{Float64,2}}:
 0.0  0.0129559  0.0         0.000192461  0.470347    0.0215264
  ⋅   0.0        0.00294986  0.0          0.64154     0.0324343
  ⋅    ⋅         0.0         0.00293481   0.40753     0.00338905
  ⋅    ⋅          ⋅          0.0          0.00192432  0.0
  ⋅    ⋅          ⋅           ⋅           0.0         0.0207859
  ⋅    ⋅          ⋅           ⋅            ⋅          0.0
```

Figure 4.3.3:

Conclusion

Throughout this project, we defined mathematical approach to calculate the energy of deformation of a truss. Then using the genetic algorithm, we were able to find the 'best' cross-sectional area of each beam while preserving some constraints.
However, there is still a room for improvement. By personalizing the GA algorithm to add new nodes to the truss, that are free to move along both directions, then apply the previous algorithm, we can still minimize the energy of deformation.