BILKENT UNIVERSITY MECHANICAL ENGINEERING DEPARTMENT

ME 362 - FINITE ELEMENTS

PROJECT 2: DESIGNING OPTIMAL TRUSS-STRUCTURED BRIDGE

Group 4 Matrix Members: Aysel Güleç 21602702
İpeksu Tutsak 21501551
Muhammed Burak Görmüş 21602797
Mustafa Burak Can 21703073
Simay Cankurtaran 21602596

Submitted to Assist. Prof. Ali JAVILI

06.04.2021

1.0 Introduction

Bridge optimization is one of the interesting topics in finite element method. The bridge optimization problem has been one of the topics that the researchers have been conducting research and proposing different methods since the 1950s and it is still progressing [1]. The aim of the optimization may involve decreasing the total weight of the structure while trying to keep the structure as stiff as possible since an excessive amount of deflection is not desired while the structure is under the applied force.

In this paper, a bridge that has a length of 10m and is fixed at its ends is studied. A Guided User Interface is created using MATLAB to test different bridge topologies to get an optimum design concerning the system's total weight and deflection. The load has a magnitude of 100000 N is applied to the structure at the center. The elements in the bridge are constrained to have a maximum length of 3m, and they are imagined to be made of steel having a density value of 8000 kg/m^3. Besides, the cross-sectional area of the elements is assumed to 0.01 m^2. This paper aims to obtain an optimal bridge design by comparing the optimization objective function results which is multiplication of the weight and the deflection of the structure. In order to satisfy this aim, the user may delete the elements or nodes based on the stress and displacement figures provided or automatic optimization tools can be used.

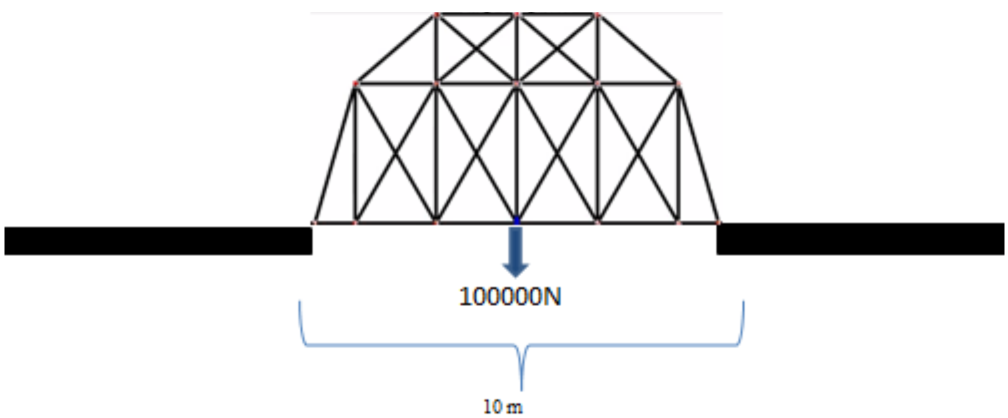

Figure 1: The Bridge Structure Studied

2.0 The Algorithms Implemented

In the literature, there are several numerical optimization algorithms proposed employing deterministic and stochastic algorithms [1]. Deterministic algorithm employs gradient based objective function during the optimization process and Newton-Rhapson algorithm can be given as an example of one of the well-known deterministic algorithms [1]. In detail, Newton-Rhapson algorithm is used to find an approximate solution for the roots of a real valued function by implementing a straight tangent line near the root value [2]. However, implementing a deterministic algorithm may be a challenge to real life problems because finding the global optimum is not easy [1]. Therefore, implementing a metaheuristic algorithm, which is one of the stochastic algorithms, could be useful because it uses randomization [1]. One of the other algorithms for the bridge optimization problem is evolutionary structural optimization, and it employs extracting the inefficient elements from the domain based on the stress values [3].

For this project, the combination of metaheuristic like algorithms and evolutionary structural like algorithms are implemented to obtain a better structure. Obtaining a lower objective function result value ($W \times \delta$) means obtaining a better structure. In the created GUI, three options, namely Element Optimization, MergeNodeOptimization and Optimize All, can be found. The pseudo code for each option can be found below.

Element Optimization is based on the principle of evolutionary structural optimization and it tests the structure while throwing away the elements having a low stress value.

The Pseudo Code for Element Optimization

- *Sort the elements in the structure based on their stress levels.*
- *From the element having the least stress value to the element having the highest stress value, elements are thrown away from the structure.*
- *Then, the optimization objective function result is compared to the old design.*
- *Until obtaining a better structure, throw the elements away from the structure.*
- *If a better structure is obtained, update the structure as well as the element list, stiffness etc.*
- *If not, raise an error message saying "There is no element to be thrown away and give a better optimization objective function."*

MergeNodeOptimization is based on the principle of the Metaheuristic Algorithm because it implements randomness. The user enters a pair of nodes and a value of increment. Then, the algorithm moves the "Optimize Node" with respect to other nodes and tries to obtain a better structure.

The Pseudo Code for MergeNodeOptimization

- *Find the angle between the nodes.*
- *Move the "Optimize Node" closer to the other node with a given increment value. (Manipulate the element between the nodes.)*
- *If the obtained structure is better, keep continuing until obtaining a structure worse.*
- *If moving the "Optimize Node" closer to the other node does not work, move the "Optimize Node" further away from the other node.*
- *If the obtained structure is better, keep continuing until obtaining a structure worser.*

Optimize All employs the same principle, but it looks for each node for every possible direction until finding a better structure.

3.0 The Structures Test Run

In this section, 20 iterations are implemented to reach an optimum structure in terms of optimization objective function(W×δ). In each iteration, it is expected to have a lower value of optimization objective function.

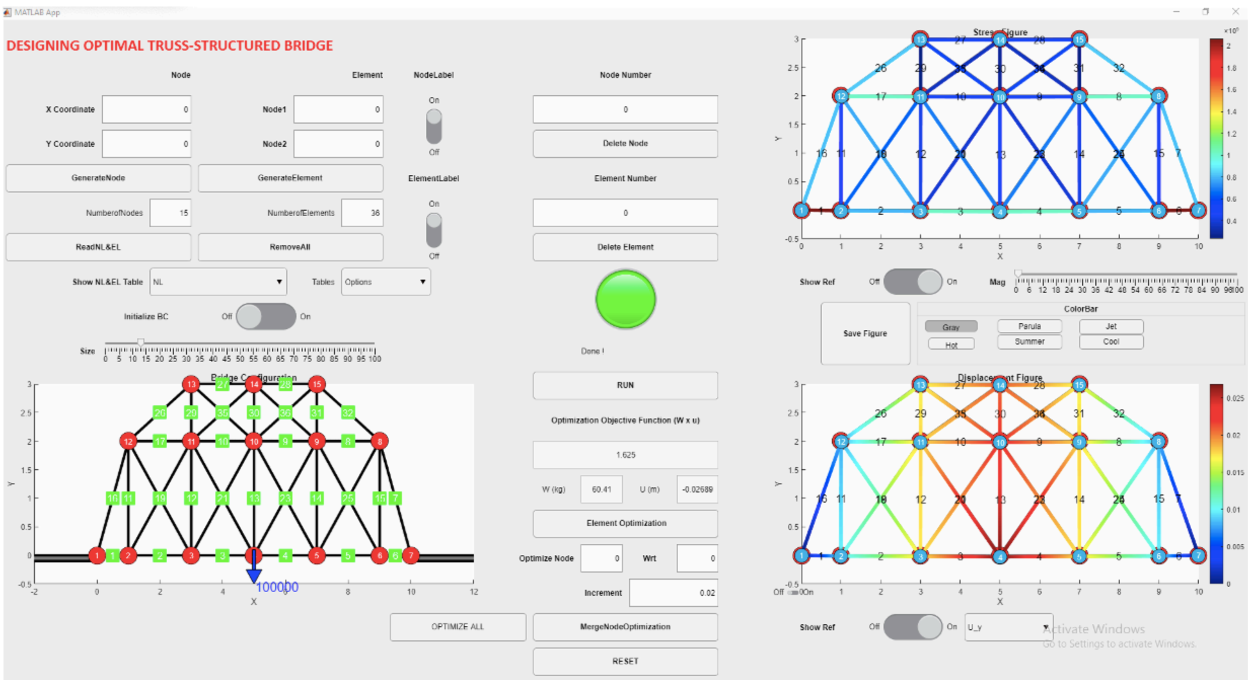

Figure 2 : First Structure

First structure is shown in the figure above. For this structure objective optimization function(W×δ) is calculated as 1.625 kg.m. To optimize this bridge configuration, an element optimization button is used. This button enables us to remove the elements so that the objective optimization reduces.

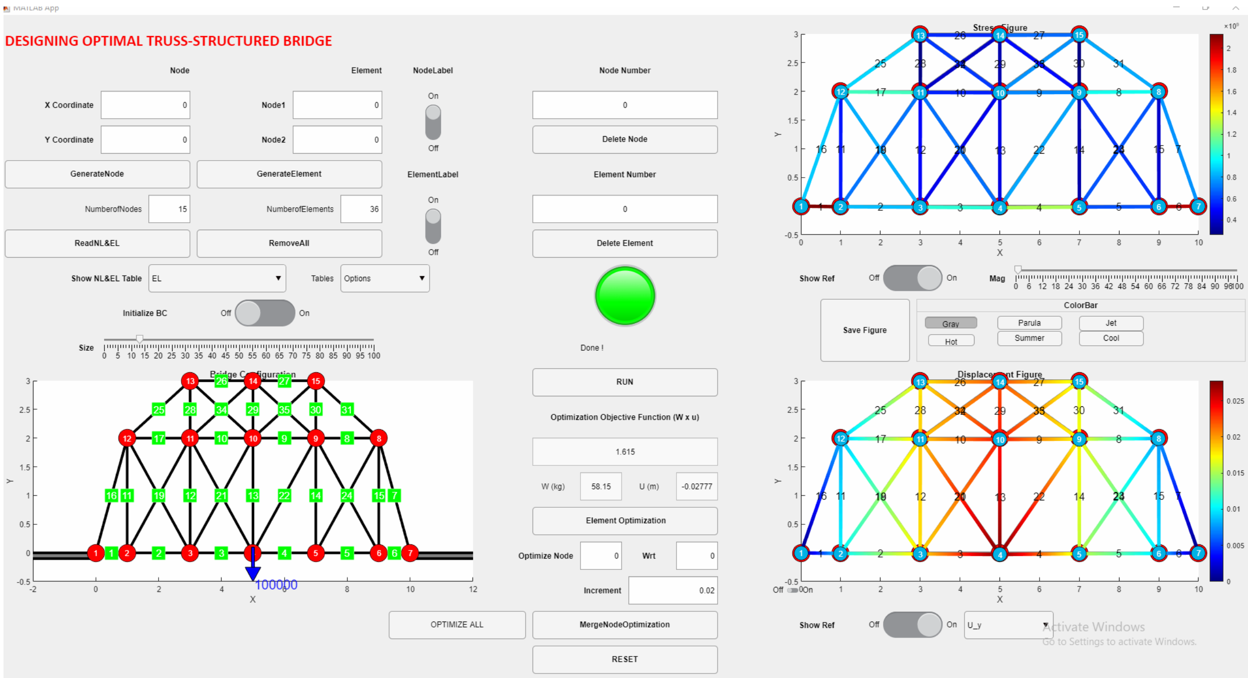

Figure 3: Iteration Number 1

We used the Element Optimization button in order to remove the redundant element, and the element between the nodes 10 and 5 has been removed.

Optimization Objective Function is found as 1.615 kg.m.

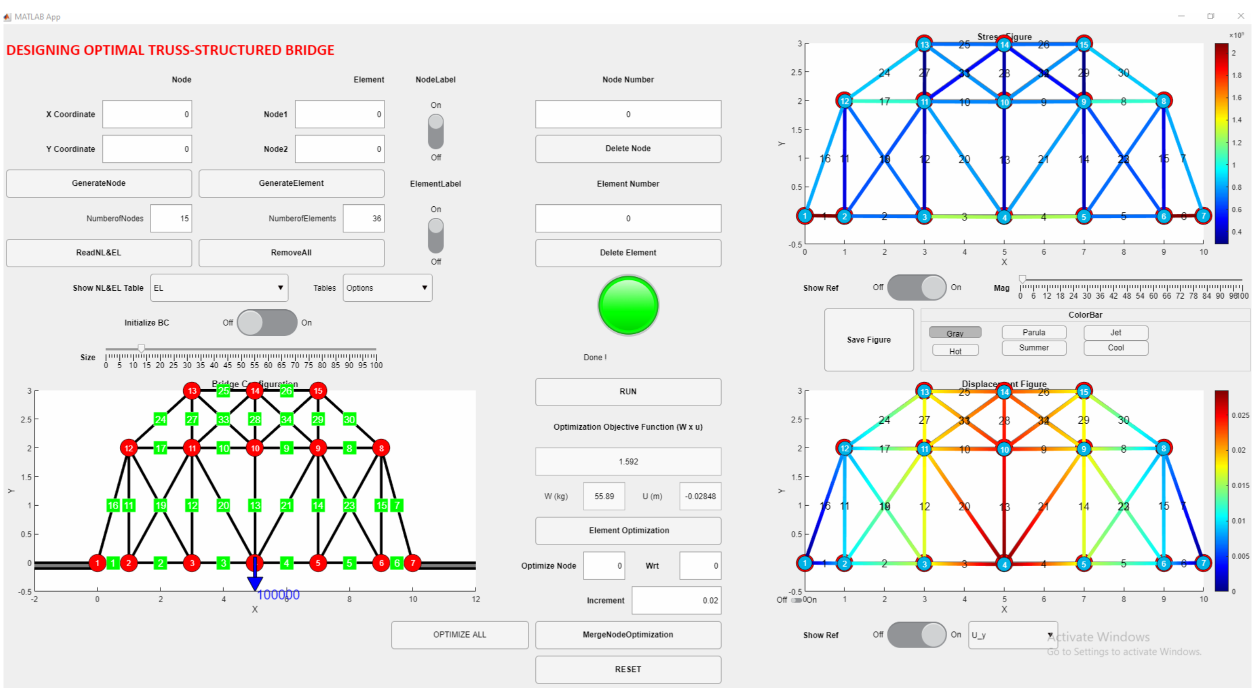

Figure 4: Iteration Number 2

We used the Element optimization, and the element between nodes 3 and 10 has been removed.

Optimization Objective Function is found as 1.592 kg.m.

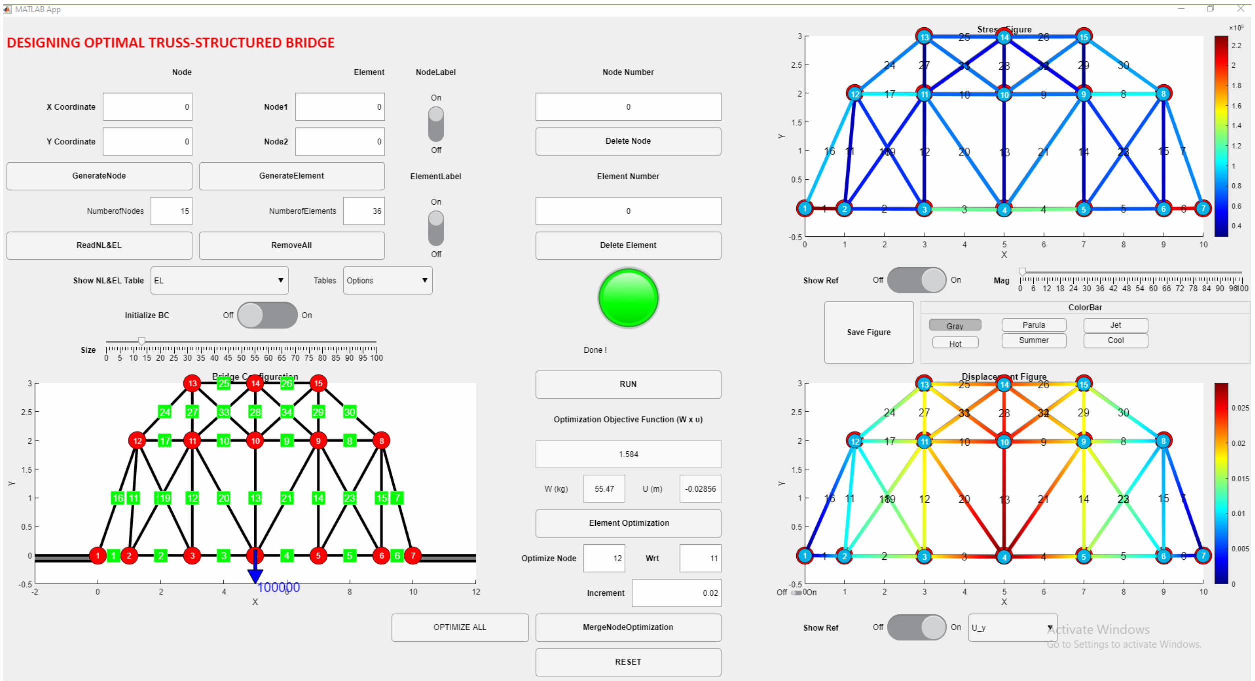

Figure 5: Iteration Number 3

Here, we used the MergeNodeOptimization button because throwing away an element did not produce a better result. We optimize node 12 with respect to node 11 expecting to have a lower stress value at the elements 17 and 16.

Optimization Objective Function is found as 1.584 kg.m.

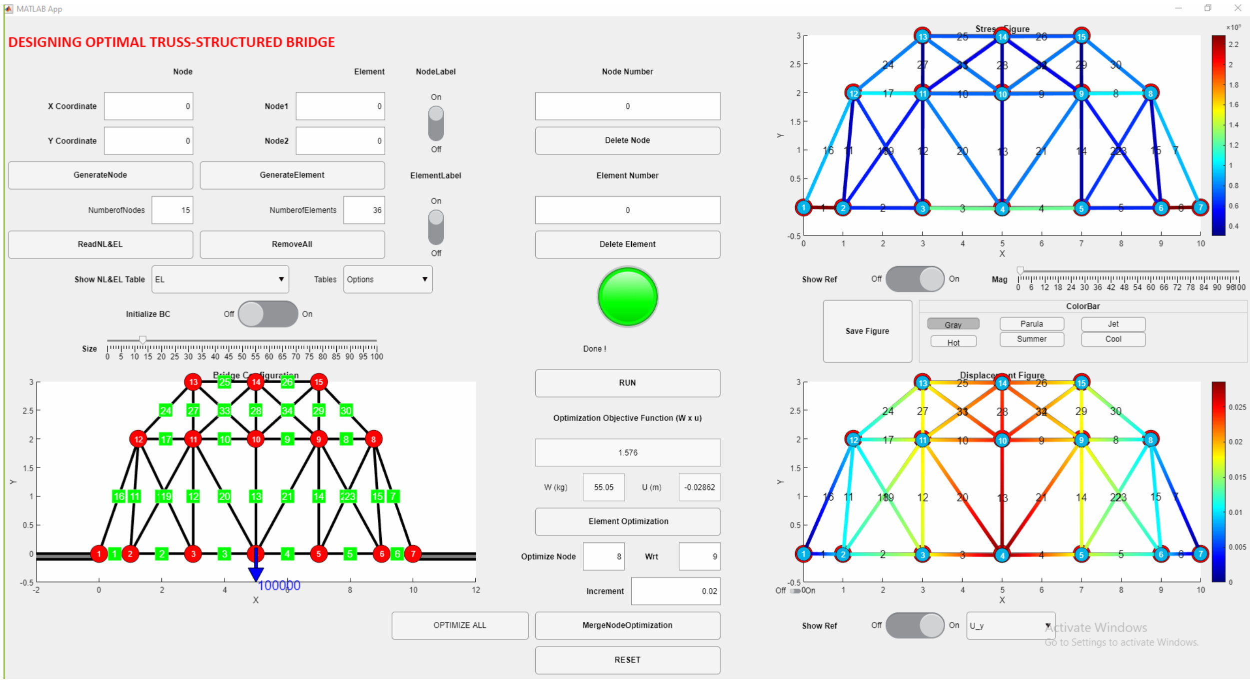

Figure 6: Iteration Number 4

By using MergeNodeOptimization, node 8 was optimized with respect to node 9 because keeping the symmetry is intended.

Optimization Objective Function is found as 1.576 kg.m.

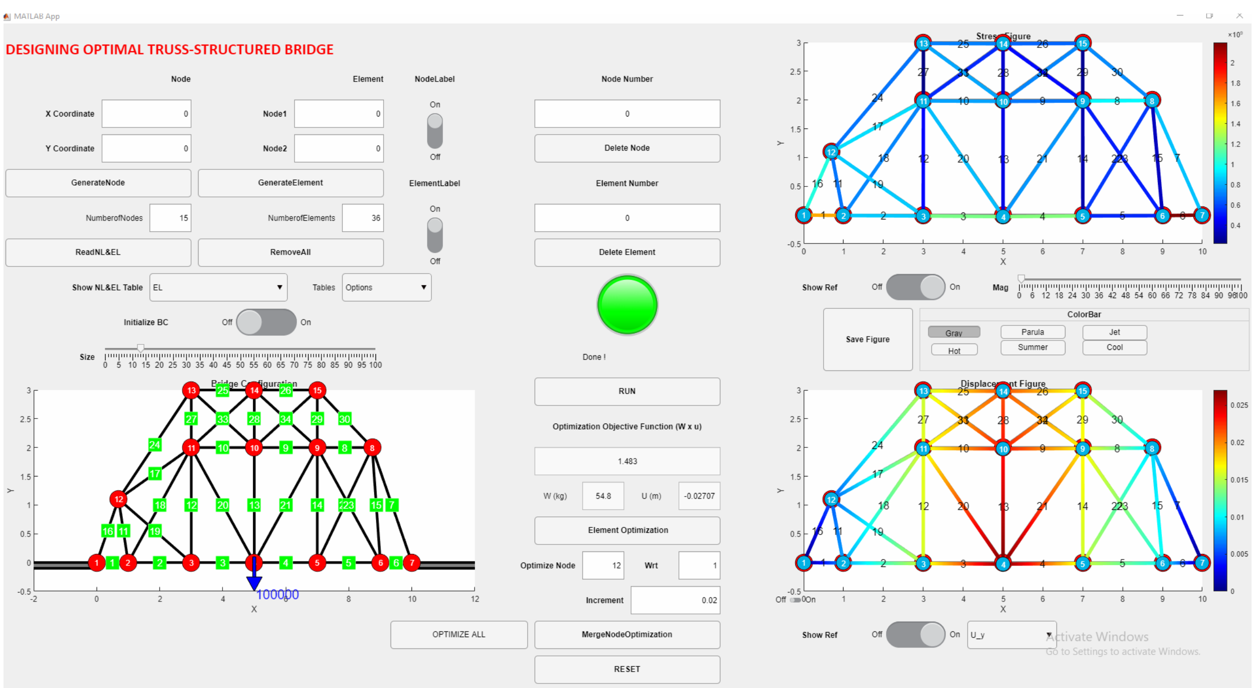

Figure 7 : Iteration Number 5

By using MergeNodeOptimization, node 12 was optimized with respect to node 1 expecting to lower the stress at the ground level.

Optimization Objective Function is found as 1.483 kg.m.

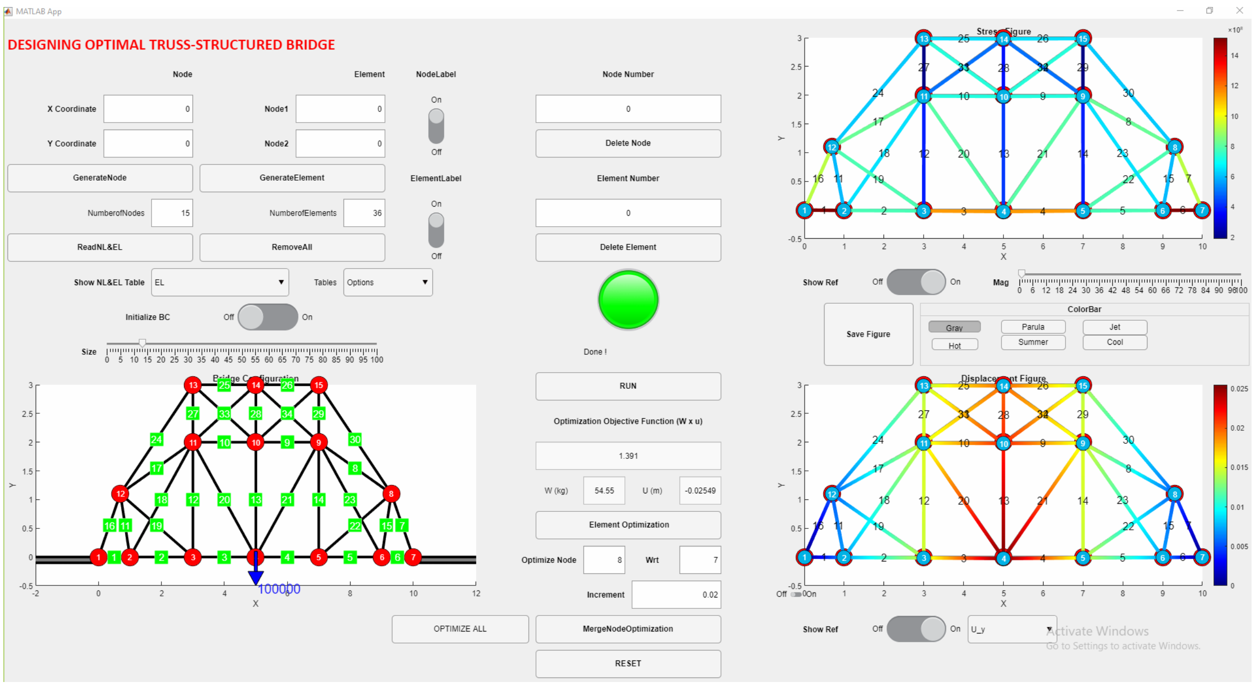

Figure 8: Iteration Number 6

By using MergeNodeOptimization, node 8 was optimized with respect to node 7 due to symmetry reasoning.

Optimization Objective Function is found as 1.391 kg.m.

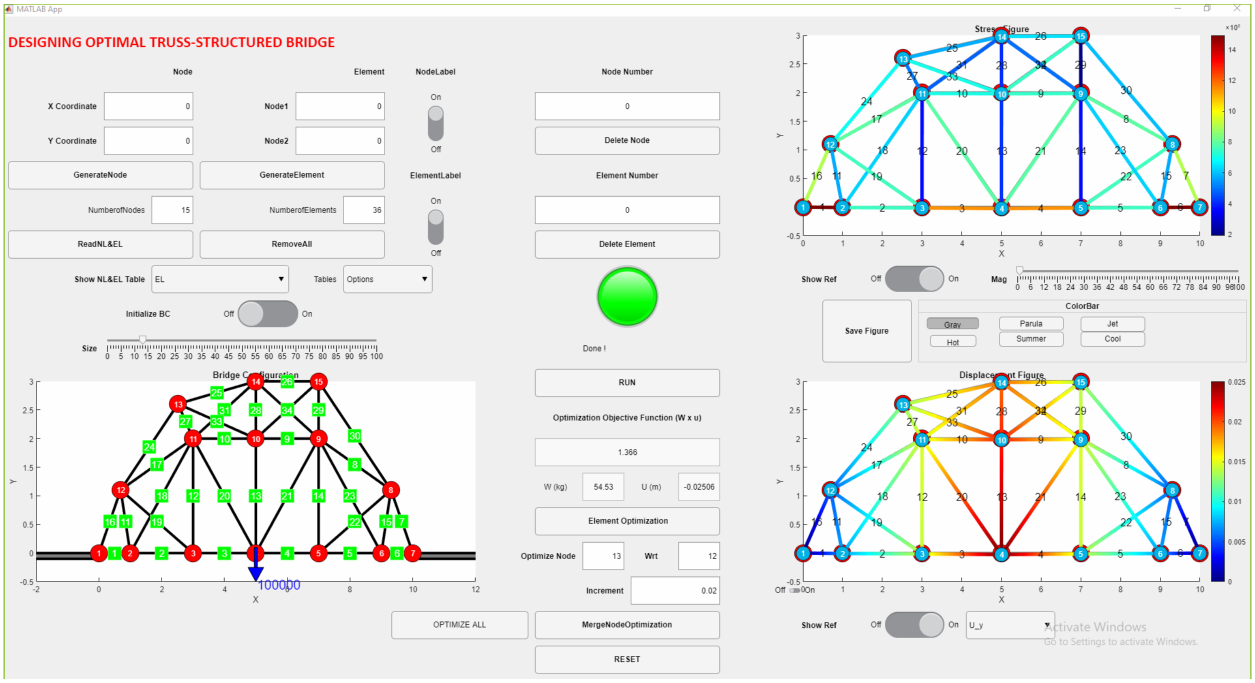

Figure 9: Iteration Number 7

By using MergeNodeOptimization, node 13 was optimized with respect to node 12 expecting to distribute the stress more evenly throughout the structure.

Optimization Objective Function is found as 1.366 kg.m

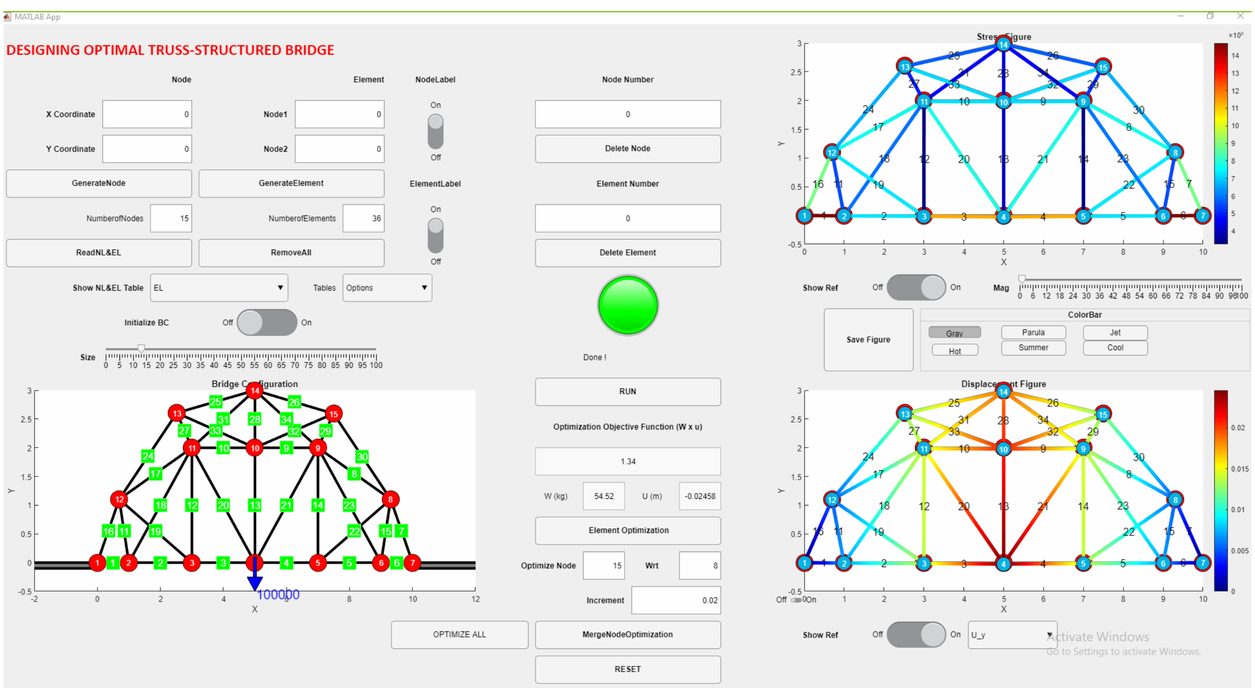

Figure 10: Iteration Number 8

By using MergeNodeOptimization, node 15 was optimized with respect to node 8 expecting to distribute the stress more evenly throughout the structure.

Optimization Objective Function is found as 1.340 kg.m.

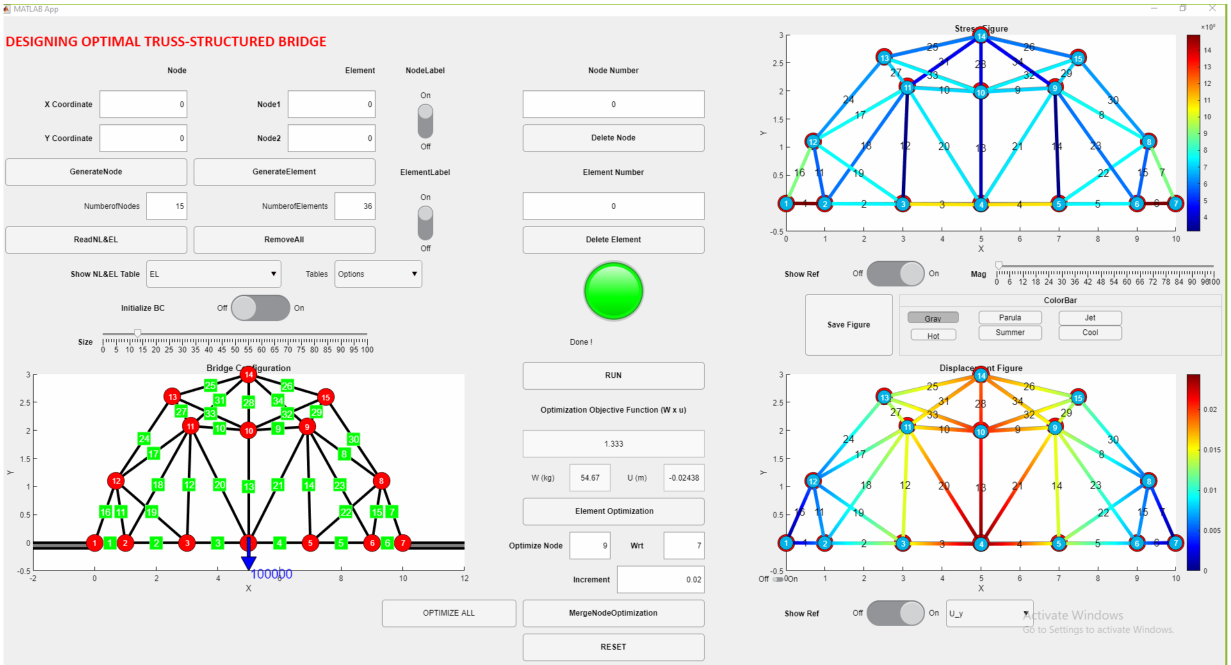

Figure 11: Iteration Number 9

By using MergeNodeOptimization, node 11 was optimized with respect to node 1. In order to keep the symmetry, node 9 was also optimized with respect to node 7. For this step, we tried to distribute the stress more evenly throughout the structure.

Optimization Objective Function is found as 1.333 kg.m.

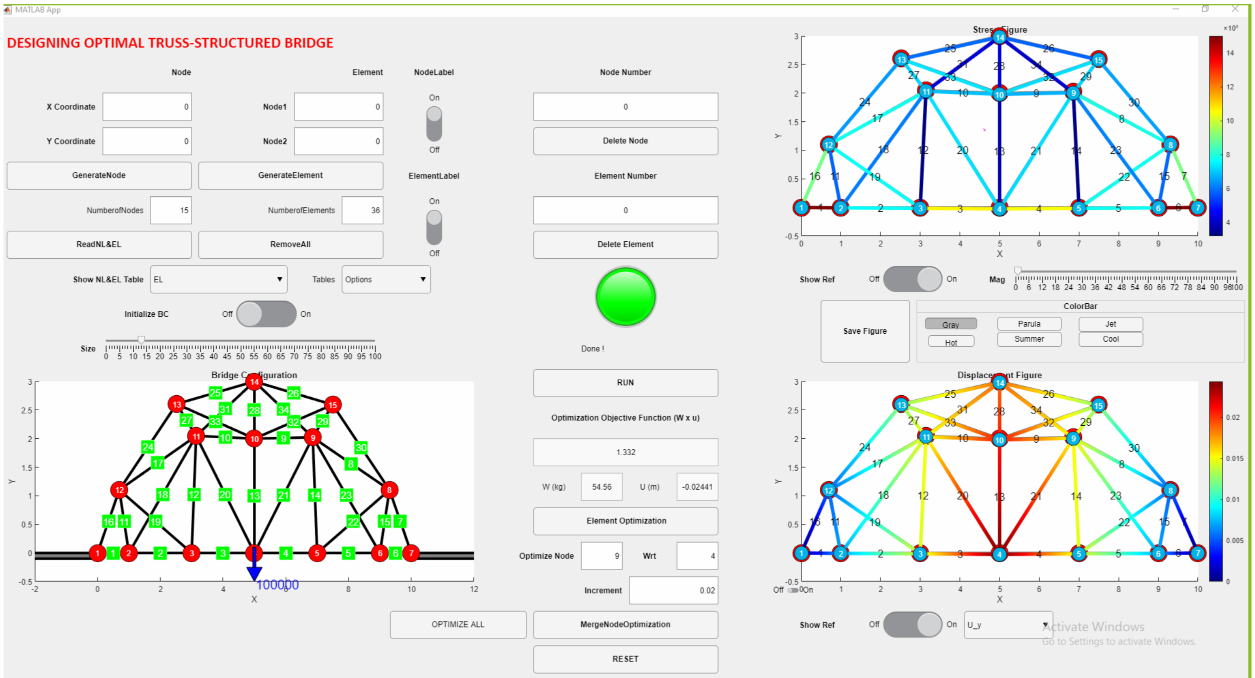

Figure 12: Iteration Number 10

By using MergeNodeOptimization, node 11 was optimized with respect to node 4. In order to keep the symmetry, node 9 was also optimized with respect to node 4.

Optimization Objective Function is found as 1.332 kg.m.

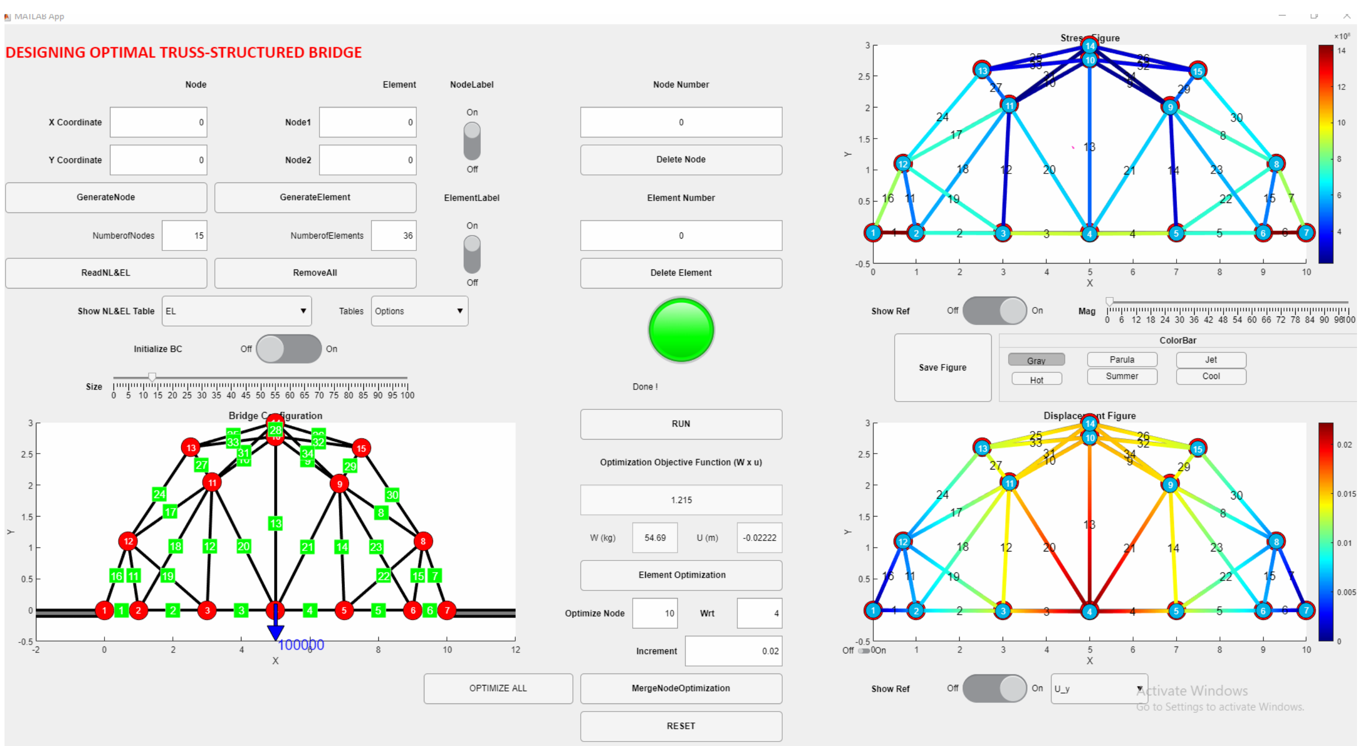

Figure 13: Iteration Number 11

By using MergeNodeOptimization, node 10 was optimized with respect to node 4 trying to lower the stress for the elements above the structure.

Optimization Objective Function is found as 1.215 kg.m.

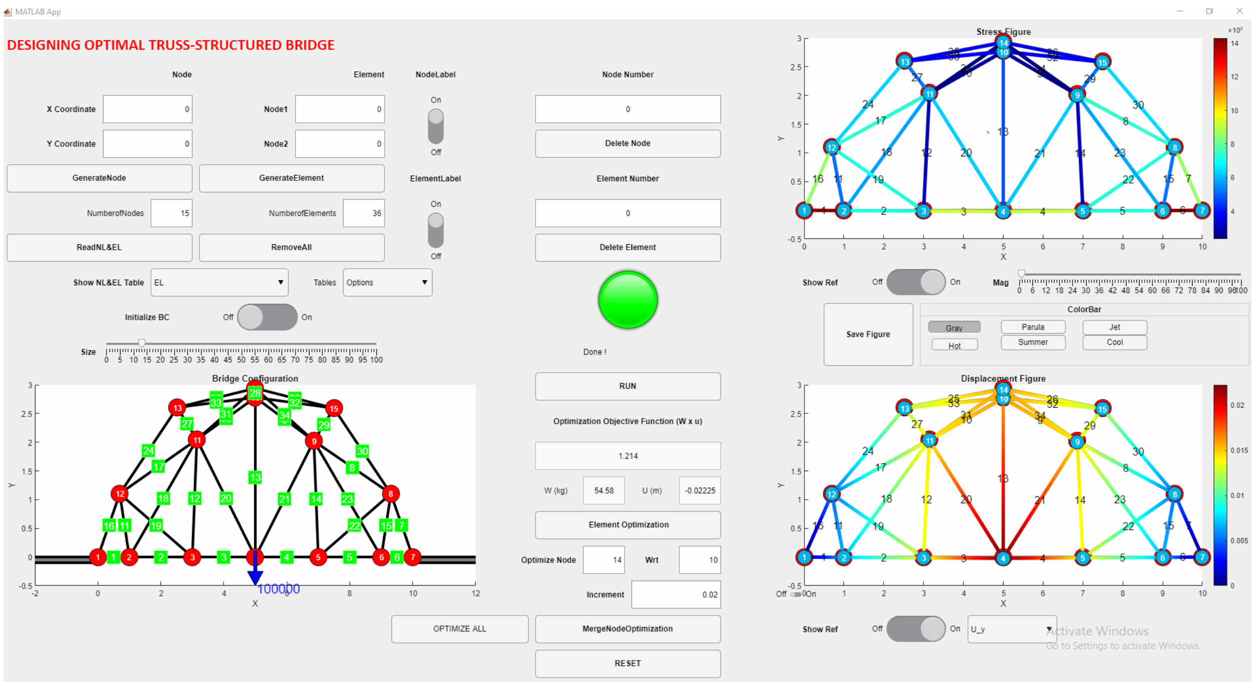

Figure 14: Iteration Number 12

By using MergeNodeOptimization, node 14 was optimized with respect to node 10 trying to lower the stress for the elements above the structure.

Optimization Objective Function is found as 1.214 kg.m.

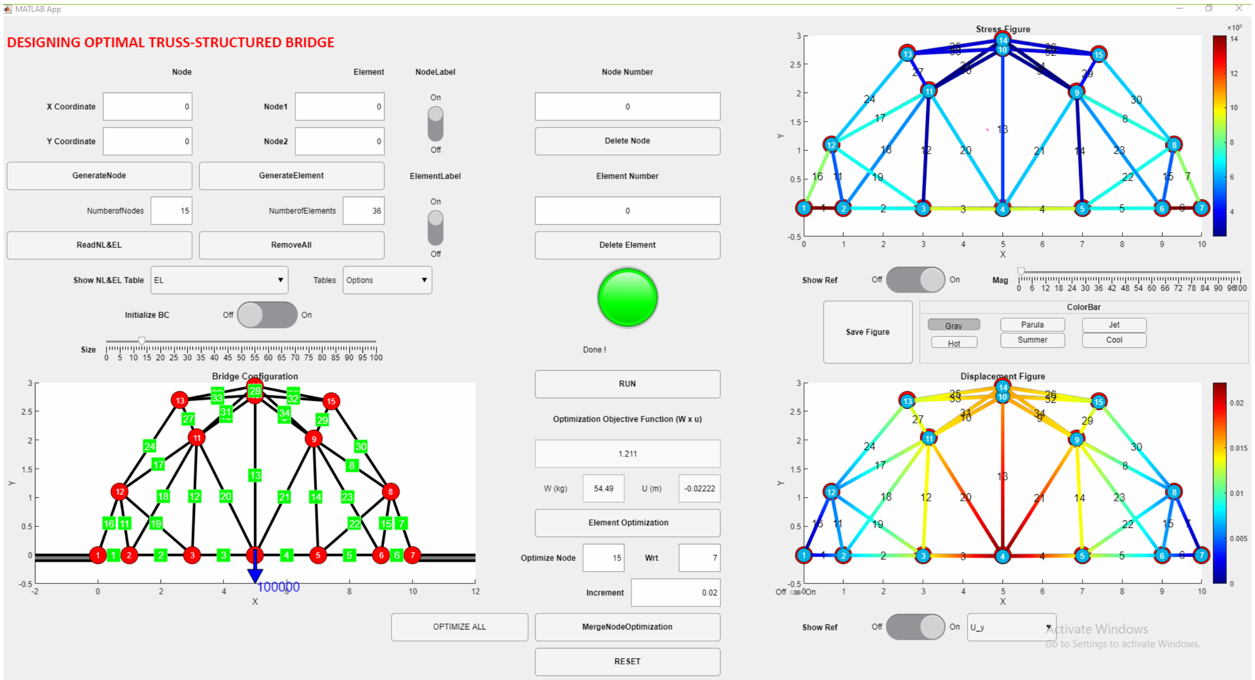

Figure 15: Iteration Number 13

By using MergeNodeOptimization, node 13 was optimized with respect to node 1. In order to keep the symmetry, node 15 was also optimized with respect to node 7 because we tried to decrease the stress by carrying the elements 25 and 26 to a lower level.

Optimization Objective Function is found as 1.211 kg.m.

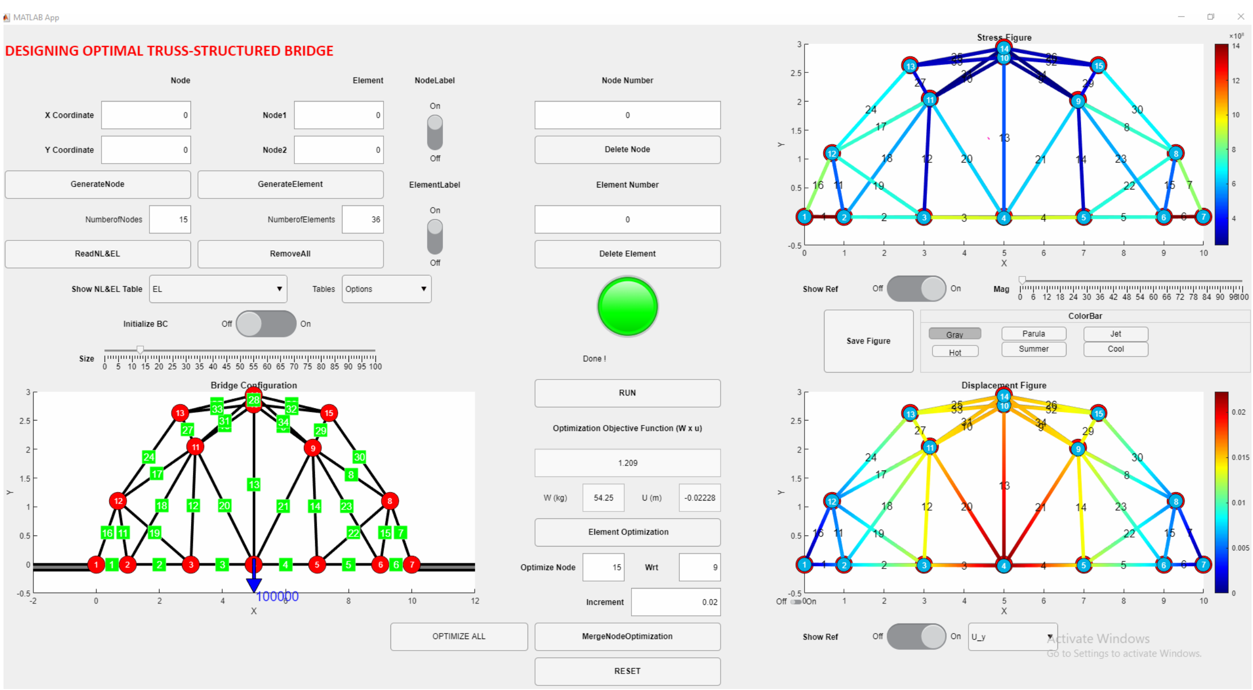

Figure 16: Iteration Number 14

By using MergeNodeOptimization, node 13 was optimized with respect to node 11. In order to keep the symmetry, node 15 was also optimized with respect to node 9.

Optimization Objective Function is found as 1.209 kg.m.

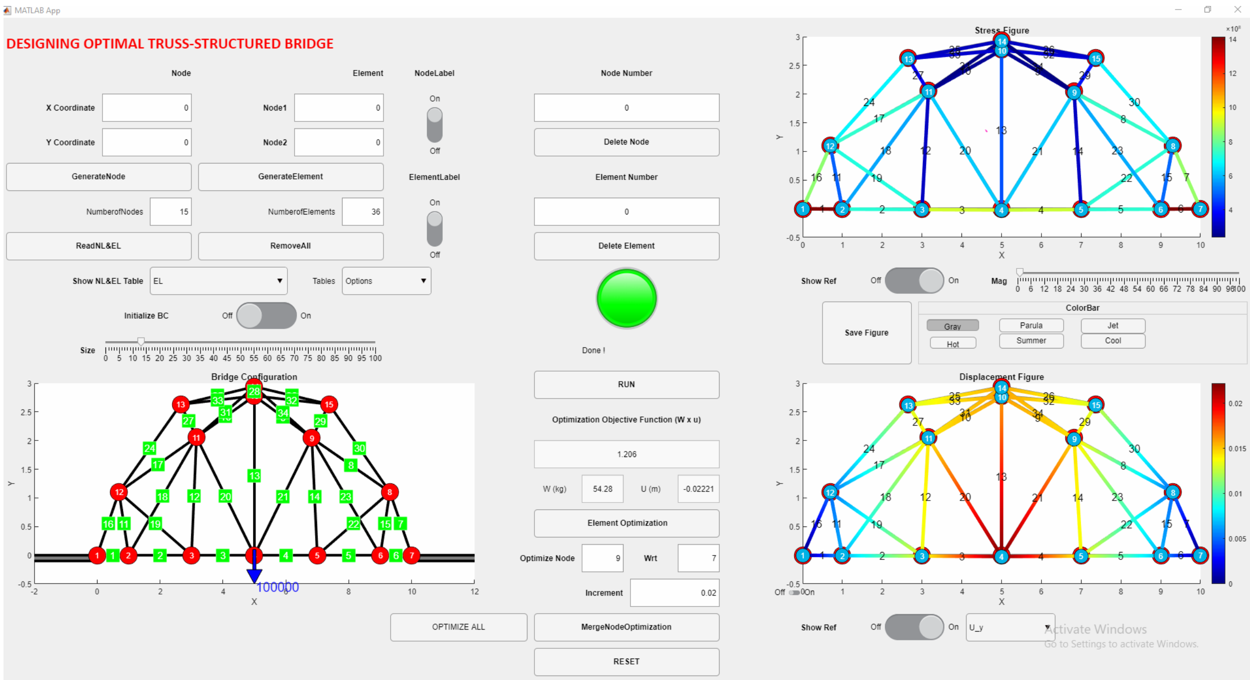

Figure 17: Iteration Number 15

By using MergeNodeOptimization, node 11 was optimized with respect to node 1. In order to keep the symmetry, node 9 was also optimized with respect to node 7.

Optimization Objective Function is found as 1.206 kg.m.

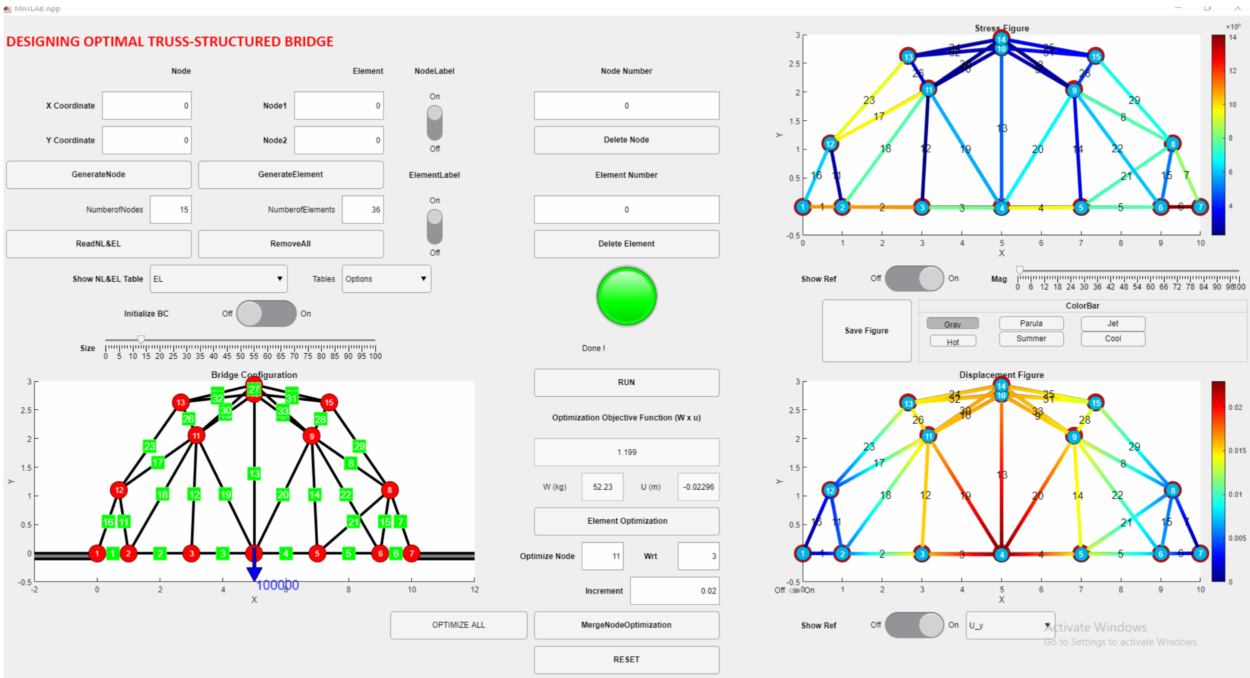

Figure 18: Iteration Number 16

We used the Element optimization, and the element between nodes 3 and 12 has been removed. Optimization Objective Function is found as 1.199 kg.m.

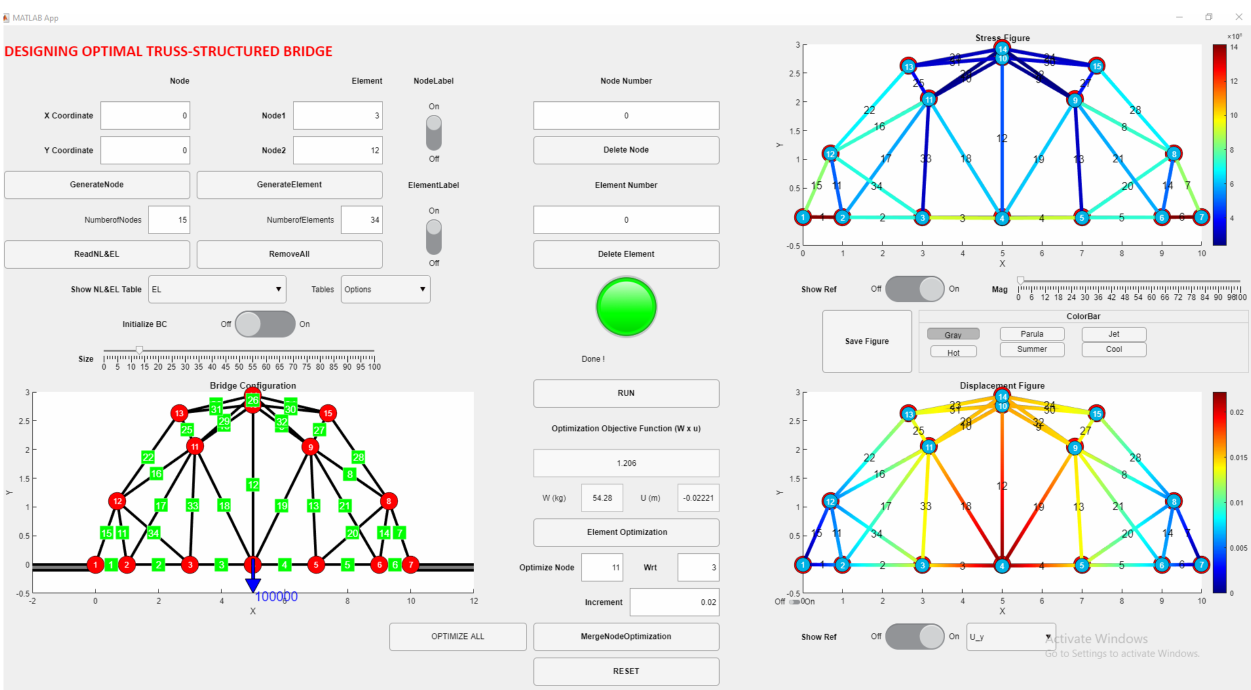

Figure 19: Iteration Number 17

Here, we have added the element between node 3 and node 11, because the stresses have gone up, and in order to reduce the stresses, we added the element.

Optimization Objective Function is found as 1.206 kg.m.

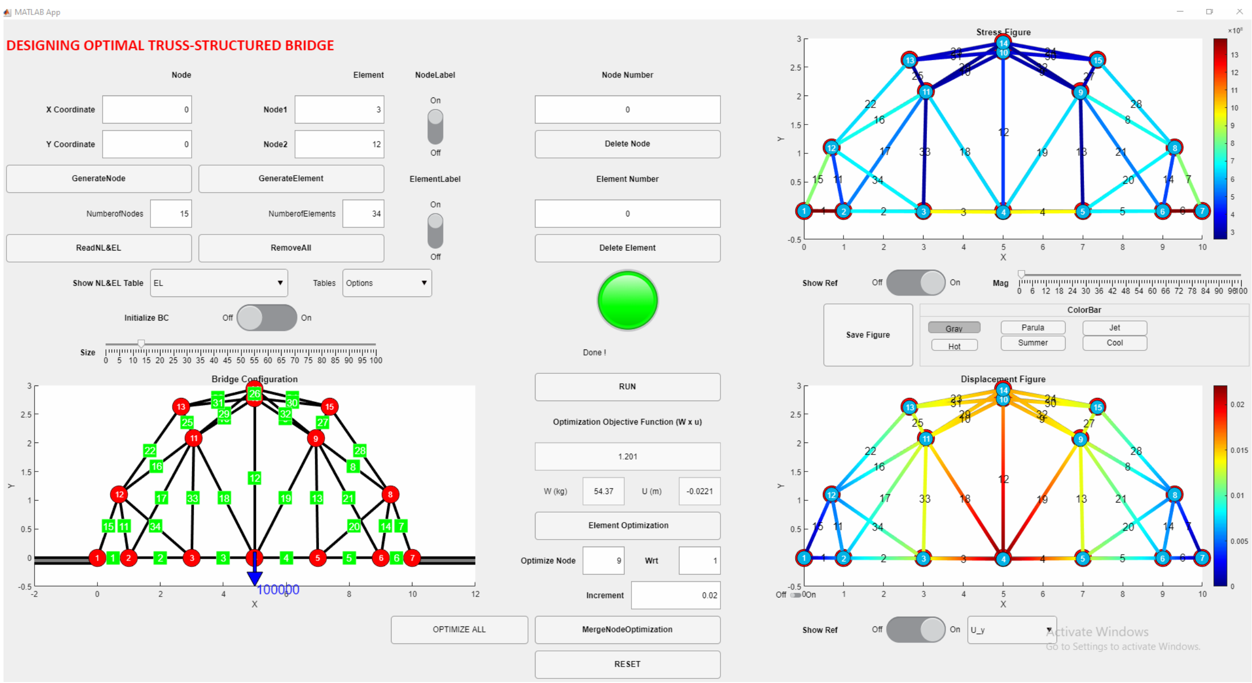

Figure 20: Iteration Number 18

By using MergeNodeOptimization, node 11 was optimized with respect to node 7. In order to keep the symmetry, node 9 was also optimized with respect to node 1.

Optimization Objective Function is found as 1.201 kg.m.

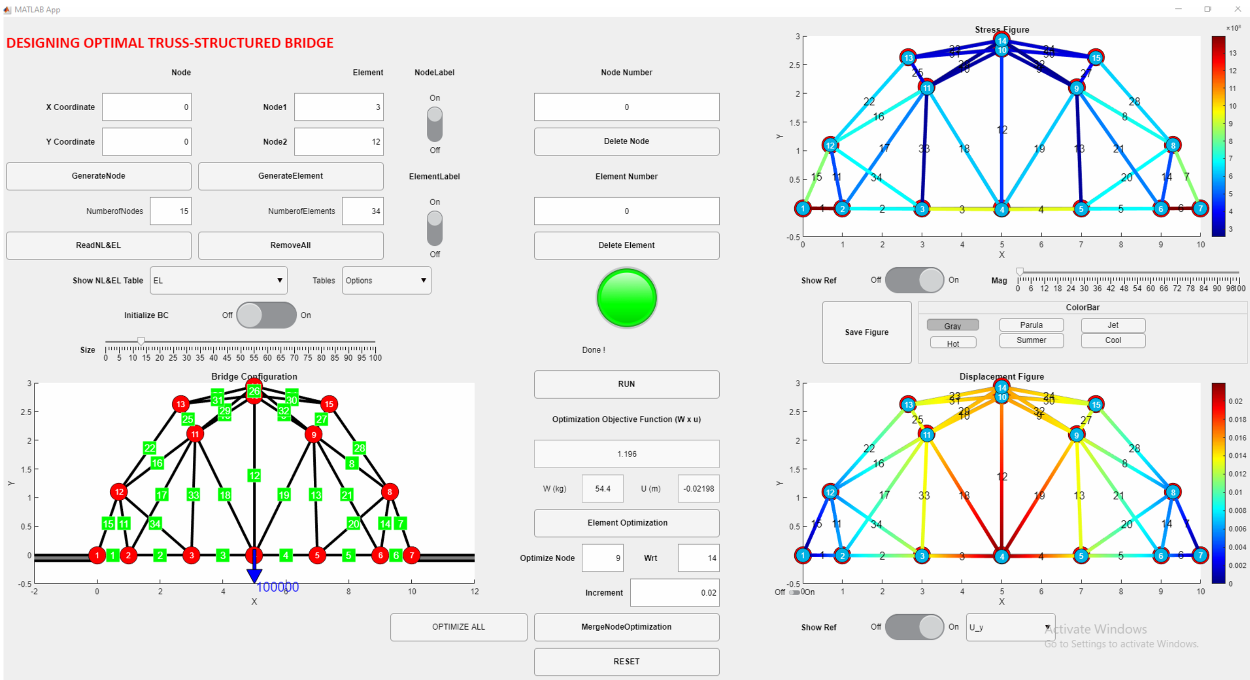

Figure 21: Iteration Number 19

The MergeNodeOptimization button is used in order to optimize node 11 with respect to node 14. Also, by symmetry it is used to optimize node 9 with respect to 14.

Optimization Objective Function is found as 1.196 kg.m.

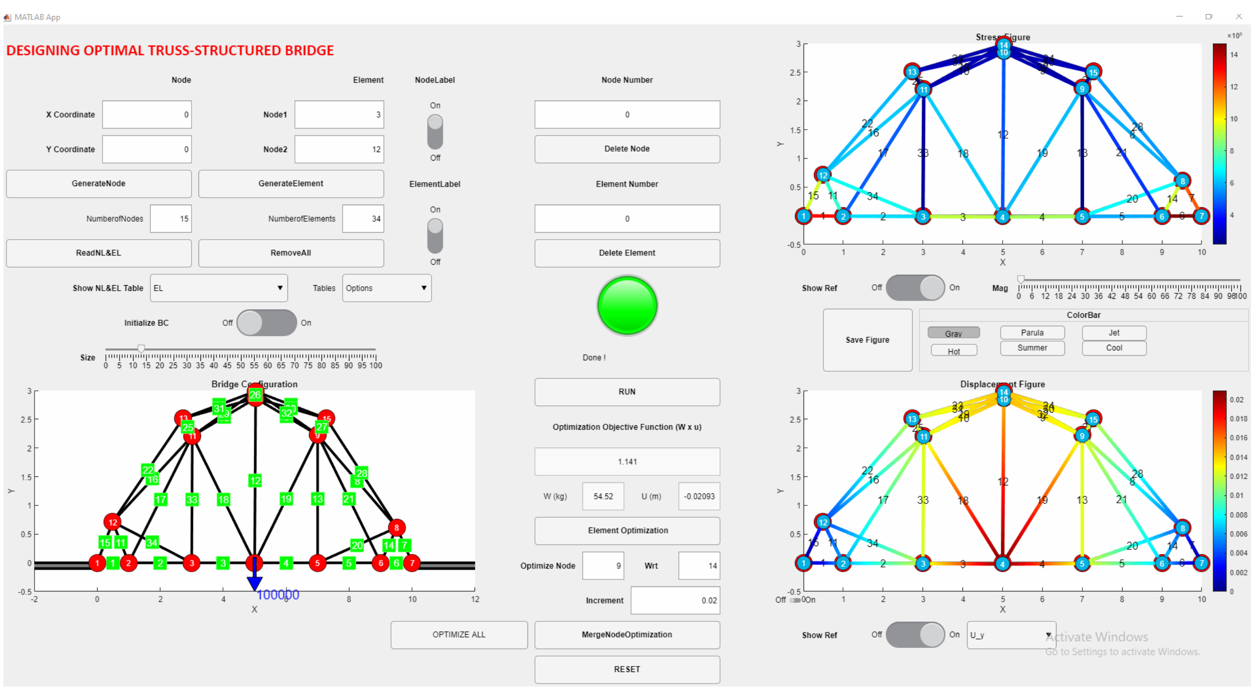

Figure 22: The Final Step

Here, we used the Optimize All button in order to let the algorithm decide on the suitible spots for the nodes. The algorithm moves the nodes around and finds the optimal spot for the least amount of stress.

Optimization Objective Function is found as 1.141 kg.m.

4.0 Conclusion

The total results of weight, deflection and optimization objective function value are shown in Table 1.

Table 1: Iteration Results

Iteration Number	Weight (kg)	Deflection (m)	Optimization Objective Function Value
The First Structure	60.41	-0.02689	1.625
1	58.15	-0.02777	1.615
2	55.89	-0.2848	1.592
3	55.47	-0.02856	1.584
4	55.05	-0.02862	1.576
5	54.80	-0.02707	1.483
6	54.55	-0.02549	1.391
7	54.53	-0.02506	1.366
8	54.52	-0.02458	1.340
9	54.67	-0.02438	1.333
10	54.56	-0.02441	1.332
11	54.69	-0.02222	1.215
12	54.58	-0.02225	1.214
13	54.49	-0.02222	1.211
14	54.25	-0.02228	1.209
15	54.28	-0.02210	1.206
16	52.23	-0.02296	1.199
17	54.28	-0.02221	1.206
18	54.37	-0.02210	1.201
19	54.40	-0.02198	1.196
The Final Structure	54.52	-0.02093	1.141

REFERENCES

[1]Mourabit, S., 2016. *Optimization of Concrete Beam Bridges*. [online] Diva-portal.org. Available at: <http://www.diva-portal.org/smash/get/diva2:945092/FULLTEXT01.pdf> [Accessed 6 April 2021].

[2]"Newton Raphson Method," *Brilliant Math & Science Wiki*. [Online]. Available: https://brilliant.org/wiki/newton-raphson-method/. [Accessed: 06-Apr-2021].

[3]P. Tanskanen, "The evolutionary structural optimization method: Theoretical aspects," 07-Nov-2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782502004644?casa_token=DjgDNAfD9swAAAAA%3AsoNgk_KouaAGgeLobFIdbOIt5LAu6PSHVKDcuI__jLJFX5ljZWd6kvWKjXrmdQeDWdK8Q5htmg. [Accessed: 06-Apr-2021].