Optimization of Trusses via Evolution for the ASCE Steel Bridge Competition

Herman Li

December 7, 2018

Abstract

The ASCE Steel Bridge Competition, also known as the Student Steel Bridge Competition (SSBC), is an annual event where civil engineering students from colleges and universities around the country compete to design and build a steel truss bridge most optimized for given specifications and limitations. This paper will discuss an evolutionary approach to optimize a starting truss design based upon this current year's specifications and rules in SSBC 2019. Using foundational equations for calculating deflection taught in first and second year civil engineering curriculum, through evolutionary processes with University of Manoa's current bridge design for the 2019 competition, it can hopefully optimize the bridge design for better performance.

1 Introduction

1.1 ASCE Steel Bridge Competition

The ASCE Steel Bridge Competition is an annual event for civil engineering college students to design and build a bridge. In the competition, there are three sections and penalties which will contribute to the score. The first is the aesthetics of the bridge which points are given depending on the quality of the poster discussing the design of the bridge. The second is the construction of the bridge where points are given depending on the time it takes to construct it. Penalties can be applied if tools, bolts, screws, or other things are dropped from the people piecing together the bridge. The last, is the structural efficiency of the bridge, the focus of the optimization via evolution. With these values, the “cost” or score of the bridge will be calculated with the school with the lowest cost winning.

This year the competition has a challenge where every team has to construct a bridge 23 feet long with the right side’s support offset by 3 to 4 feet from the end and the left side’s supports on the ends of the bridge. There will be two loads on the bridge with 6 configurations on their placements on the bridge. The first load is 1400 lbs. and the second will be 900 lbs. There is also a limit on the size of the bridge given in page 43 of the SSBC 2019 rulebook.

1.2 Previous and Similar Work

Currently, there have been no schools which have approached designing and optimizing their truss design via evolutionary computation. In the engineering literature, there have been multiple papers related to the optimization of trusses via multiple evolution methods including the design of a bridge via cellular growth done by Tschida and Silverberg (2013). There have also been multiple papers with an approach similar to the one used here but with much more complexity (the limit being time constraints of this project) including one using up to twenty-five members to generate a truss with no limit in position and member locations (Azid et al. 2002).

2 Approach

The implementation of the project was done via C++ and simplified the 3D truss into a single truss. Although there would be two trusses for the bridge's design, the single truss will be replicated on both sides of the bridge with the truss with the longer span tested since that span will deflect more than the other. Since the maximum span will be counted, this means the other side's truss can be replicated and should not affect the result too much.

2.1 Representation

The truss is represented by joints and members. A joint is simply two values representing an X and Y coordinate in inches. The coordinate (0, 0) represents the start of one side with (276, 0) representing the other side. A member was represented by two joints showing that there is a connection between them. A bridge will be represented by a list of joints and members. Since there is a limit on where joints can be placed, there is a check on whether it exceeds the limit. Similarly, members cannot cross each other or it will result in an indeterminate truss, therefore, there is a check and limit on how the members and joints can be moved and modified.

2.2 Search Space

The search space for the truss will be large and will vary depending on the initial configuration and design of the bridge. With R as all locations a joint can be located j as joints and m as members, the search space can be represented the equation $(R \wedge j) * m$.

2.3 Population Size

Currently, every generation will have 100 trusses to be tested and analyzed with a variable number of generations although generally it will be run from 250 to 1000 iterations.

2.4 Selection

Currently, selection is done by removing a certain percentage of bridges dependent on their performance level. The best performing bridge of a generation will never be removed. The first third after this will have a 10% chance of being replaced. The next third will have a 30% chance of being replaced. The last third will have a 90% chance of being replaced. The reasoning behind this chance of removal is to add some possible variation allowing for a possibly better optimization to a truss than what the current focus is on.

2.4 Mutation

Mutation is done in a similar staging as selection. The removed first third will use another random bridge other than itself in the first third to be copied over. The next third will use any bridge within the top two-thirds. The last third will just use any bridge that is currently better than the bridge it will be replacing. This in a form is an extended part of the selection process.

After copying the bridge over, for points where the Y value is not equal to 0, the point will be moved in increments of 0.5 inches using a random number generator generating a number based on a Gaussian curve with a standard deviation of 0.5 inches. This means, most of the time, the joint will be unlikely to change location but has a very small chance to be modified by more than two inches. If the joint is modified to have a Y value lower or equal to 0, then it will add 0.5 inches until it is not. For joints already having a Y value of 0, the mutation will only allow modification of the joint on the X-axis using the same system of random number generation. After modification, joints are checked if the position they are in already has a joint. If it is, then it will be changed again until it is not.

Since there are mandatory joints where the supports of the bridge are, joints already in those locations will not be moved and will remain in their original positions.

2.5 Fitness Equation

The fitness equation just uses the structural efficiency equation provided by the competition rules on page 15 section 6.2.6. This equation is in the form of a stepwise function dependent on weight with the lower the cost calculated from the equation, the fitter the bridge. The only values required will be the aggregate deflection, the summation of all deflection measured at the defined points on the bridge, and the total weight, calculated from the type of steel and amount used.

If weight does not exceed 120 lbs.:

$$Cost = Aggregate\ Deflection * 3{,}250{,}000$$

If weight is between 120 lbs. and 200 lbs.:

$$Cost = (Weight - 120) * 5{,}000 + Aggregate\ Deflection * 3{,}250{,}000$$

If weight is greater than 200 lbs.:

$$Cost = (Weight - 184) * 25{,}000 + Aggregate\ Deflection * 3{,}250{,}000$$

Deflection is calculated via the method of joints to determine the forces in each member and the axial deformation equation.

3 Results

After running a test bridge for 500 generations, the bridge resulted in a significantly lower deflection than what is originally started with. The fitness was strongly correlated with the deflection mostly because the fitness equation is mostly affected by the deflection once it hits a certain weight threshold.

The deflection values are generated by the algorithm as it processed each generation and therefore might be inaccurate since the deflection algorithm might not be correctly implemented. At around generation 90, there is a steep drop in cost which might be related to how a certain member might have switched from tension to compression resulting in a change in bridge dynamics.

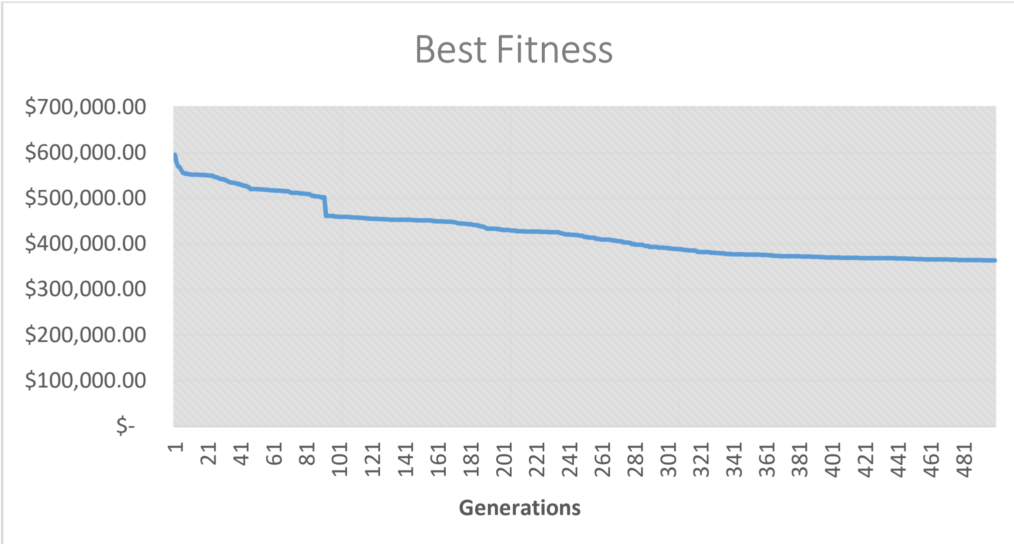

Figure 1.

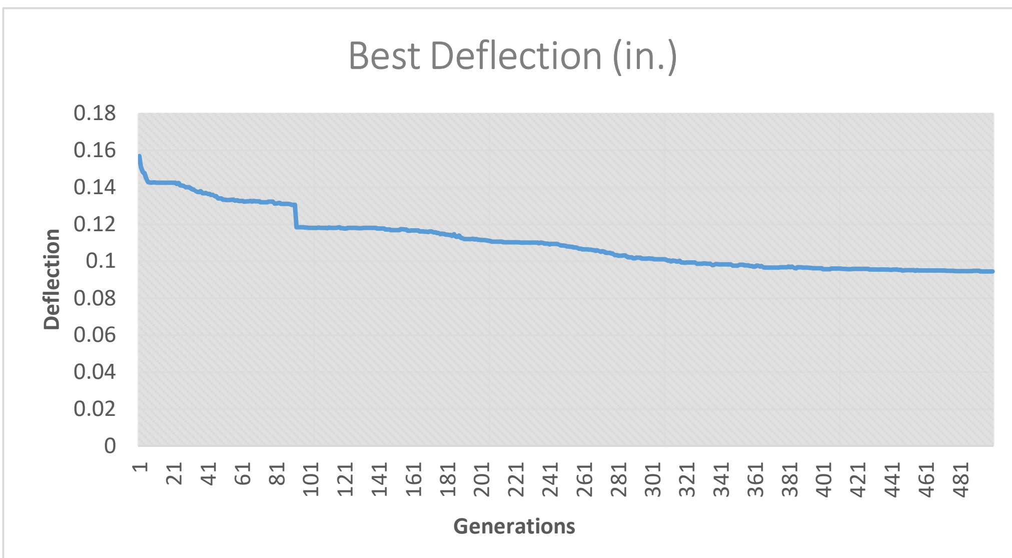

Figure 2.

In other tests, the truss somehow generated a deflection of 0 inches which seems highly unreasonable given the design of the truss. This issue will be discussed later.

When running this upon the current bridge design by the University of Hawaii's ASCE steel bridge team, the initial design and optimized output was tested using SAP2000. SAP2000 is the structural analysis program used by civil engineers in a professional capacity to give a relatively accurate estimate on what to expect on the performance of different structures. The initial truss (Figure 3.) was modeled and tested to an estimate of 0.006375 inches of deflection. After optimization via the program, the truss (Figure 4) now has only 9.956×10^{-8} inches of deflection. This was very similar to what the deflection equation implemented within the evolutionary algorithm had estimated. Before the test within SAP2000, it was believed to be an error in the project's implementation of method of joints and deflection equation. Since SAP2000 provided similar values to what was calculated, the 0 deflection calculated in other runs of the project seems to be the result of something else which is affecting both this project and the SAP2000 calculations. Due to lack of time to complete the project, the reasoning behind this anomaly was not discovered.

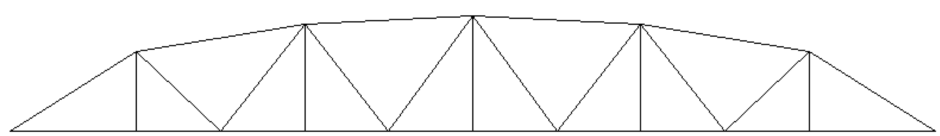

Figure 3.

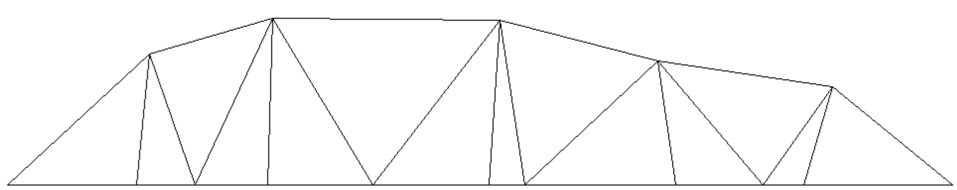

Figure 4.

4 Conclusion

The original idea of the project was to assist the steel bridge team in optimizing their design for the bridge to better perform in the competition. Although the result seems to be successful, it likely is a flaw in both the approach used in the project's calculation of deflection and SAP2000's approach in deflection calculation. Since some civil engineering calculations might be limited in scope on where they can be applied, it might be the case in this scenario. This likely means the evolutionary algorithm just found the optimal way to optimize the bridge using an equation which cannot be applied in the bridge design resulting in no deflection.

5 Final Remarks

Code used to optimize the bridge using evolution can be found here at this link.

https://github.com/lihongman/BridgeEvolutioner

Also SAP2000 modeling and calculations are done by Elmo Gonzales, a civil engineering major at the University of Hawaii at Manoa.

References

[1] Student Steel Bridge Competition 2019 Rules
https://www.aisc.org/globalassets/aisc/university-programs/ssbc/ssbc-2019-rules.pdf

[2] C. Tschida and L. Silverberg "Cellular growth algorithms for shape design of truss structures" in *Computers & Structures Volume 116, 2013*

[3] Azid, I., Kwan, A. & Seetharamu, K. Struct Multidisc Optim (2002) 24: 333.
https://doi.org/10.1007/s00158-002-0244-9

[4] Anthony Bedford and Wallace Fowler "Engineering Mechanics: Statics & Dynamics (5th Edition)