Further Exploration of Evolutionary Algorithms to Generate Large Irregular Tensegrities

Daniel H. Casper

June 12, 2024

Abstract

Tensegrity form-finding is a complex field relating to the discovery and creation of new tensegrity structures. This process has been approached and practiced from many different angles ranging from mechanical, to algorithmic. When researching the use of evolutionary algorithms to evolve irregular structures, the experimenters were met with success, however the line of interest ended there. I propose recreating the algorithms explored in old studies and attempting to match or improve upon the results presented prior. Through the application of newer studies as well as a fresh perspective on the problem, I want to explore what changes can be made to the original algorithms to produce new tensegrities.

1 Introduction

A tensegrity robot is a type of soft robot that consists of rigid elements called struts, and elastic or otherwise tensile but stable elements which are referred to as cables. When connected, these struts and cables are put under compression and tension respectively, creating a stable yet resilient structure which is a tensile-integrity (tensegrity). In the world of tensegrity robotics, one of the most prevailing problems facing researchers is that of designing new unique structures upon which to model their robot. The process of designing and creating a novel tensegrity structure is known as morphological discovery. Taking that new design and finding the best resting configuration for a new robot is called form-finding which in itself is a substantial field of study. In the years since tensegrity robotics began to grow as an increasingly researched topic, there have been many methods and approaches to the problems of both morphological discovery and form-finding that have been explored.

As tensegrity structures and robotics exist at a unique intersection between mechanical and structural engineering and computer science, researchers from both backgrounds have looked into numerous strategies through which to discover new and unique stable structures. Ranging from strictly mechanical and kinematic approaches [9, 4, 2, 7] to algorithmic or machine learning methods [6, 8, 11, 10].

For this project I aim to explore a method of morphological discovery and form-finding for large scale, unique, complex structures by employing evolutionary algorithms. The goal would be to develop an algorithm that, when given a simple 2d graph input, will output a representation of a new large, irregular tensegrity structure for a robot. Using the final resulting structure from this I would then be able to simulate the proposed design and determine the optimal configuration for it with a physics simulator.

While this line of study has been explored before by Rieffel et al. [6], the code that was used for that and any further studies has fallen into disuse and is no longer able to compile. I plan to recreate the work that was done in that study and in doing discover two things. First, I of course want to see if this is still a viable method of generating unique tensegrity structures. On top of that however, I also hope to determine ways to improve on the original work both by implementing discoveries that were made following the initial paper's publishing, as well as learning from the original experiment and making more educated decisions in the algorithmic design.

In order to create a system that generates these new structures there would be two distinct components. The first component would apply a set of graph grammars to systematically grow a simple input graph into a more complex graph to be used in an evolutionary algorithm. This would be done in a two step process which would first use a series of transformations to generate new edges and nodes, and then follow

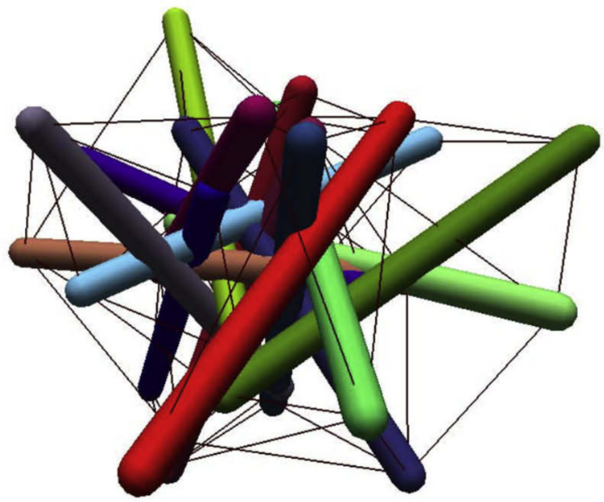

Figure 1: A large, irregular 15-strut tensegrity structure that was generated in [6]'s paper using the methods that I'm proposing to revisit.

that with a series of secondary changes that would further change the graph and make it translatable into a tensegrity. The second component would then take a set of these complex graphs and use an evolutionary algorithm to selectively mutate them and determine which structures developed desirable traits for a tensegrity.

2 Background and Related Work

The work that I am proposing is in essence a re-evaluation of the work done by Rieffel et al. [6], exploring methods of automated generation of large scale, unique, complex tensegrity structures. As seen in Figure 1 In their research, they created a set of grammars to dictate the evolutionary "growth" of a 2d graph into a fully realized tensegrity structure. This work in itself built upon work done by Paul et al. [5], attempting to find solutions that were able to adapt and perform well at a larger scale as well as at the smaller scale which had been successfully done. While they found success in their experiments, there was not much follow up on the methods they explored and the study of morphology and form-finding through grammar based evolutionary algorithms has been more or less static for the last decade.

There have been further recent explorations of other automated methods for new tensegrity form-finding however, that could be useful to consider when re-exploring an evolutionary method. Recently, in 2020 Zalyaev et al. [11] explored using a machine learning pipeline to so as a method of form-finding. Another route that has been looked into, that is more closely linked to topics I intend to research is the implementation of genetic algorithms towards irregular form finding Xu and Luo [10]. In this study, a genetic algorithm was proposed, that while similar to the work done by Rieffel et al. [6] and Paul et al. [5], used some differing optimization models. This study also saw promising results, but nonetheless the study of irregular tensegrity form-finding has not been touched very much since then.

Along with this study, Koohestani [3] also explored using genetic algorithms to solve the form-finding problem for new, although regular, tensegrity structures. It was concluded that while the methods proved promising for regular structures, it would require more development for applications to irregular one. While there have been many other experiments with algorithmic solutions for form-finding, many of those studies as well as the one in this paper were found to not perform particularly well when applied to irregular tensegrities. As with other studies mentioned above, it could prove useful to review the methods, strategies, and parameters used to produce tensegrity structures in this study to inform the parameters I elect to apply in my own algorithm.

Besides automated methods of form-finding in more computer-science related fields, there is also much

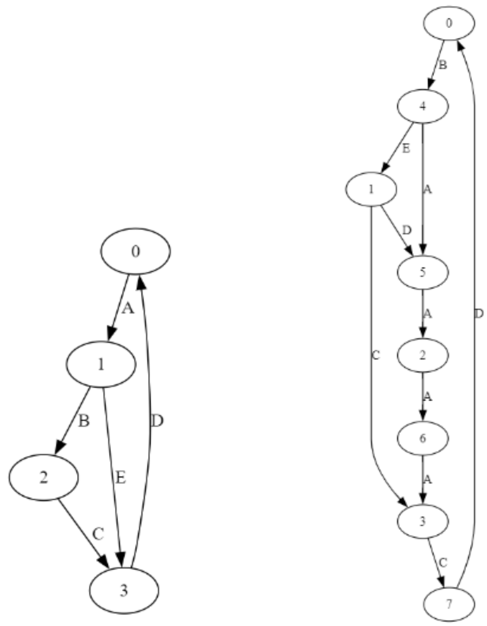

Figure 2: The result of a simple graph being transformed by a map L-system

research into methods of form finding through more direct approaches from mechanical fields. Methods of static Juan and Mirats Tur [2] and algebraic Masic et al. [4] analysis have been explored as well as kinematics driven methods [7, 8]. While not directly tied to the methods I plan to use, I think it could be useful to explore some of the strategies used in traditional engineering solutions to form-finding to better inform optimization choices or even the graph-grammars that would be used in my study. Even if I ultimately do not use any significant aspects from the research done in these papers, I believe that developing a deeper and more rounded understanding of the form-finding problem for tensegrities should prove useful regardless. At the very least, having taken the time to understand different approaches to this problem will provide me with a greater perspective of the field as a whole.
The benefit of returning to this line of work over a decade after the initial paper was published is the ability to draw on new studies as well as the results of the original study for better optimization. Not only will I be able to utilize on all of the same resources that were cited in the original paper, but there is now a vast pool of resources and studies that have conducted in the time since to draw on as well.

3 Methods and Design

For this research I plan to mostly follow the approach taken in the original study to generate new tensegrities. I intend to mimic the processes through which the original paper outlined graphs were grown and then transformed into translatable tensegrity structure designs. The methods that will be used to generate a simple 2-d graph into a representation of a new tensegrity will be broken into two phases as follows.

3.1 Phase 1: Graph Transformation and Growth

In the first step there will be a series of transformations applied to a graph to grow it into a larger, more complex representation of a new tensegrity structure. This process in itself will consist of two distinct components. Each of these components will employ a unique set of grammars, or transformational rules, to use as the parameters by which to transform the graph. The first component would use node and edge identification on a directed graph and evolve it a predetermined number of times according to the set of rules given.

At the end of this first phase we will have a significantly more complex directed graph as shown in Figure 2. This graph on its own however, would not lend itself to creating a new tensegrity based on the paths between nodes. In order to do this, the second set of grammars would evaluate the new graph and indicate which an edge is to be strut, a cable, or extraneous and could be disregarded in the overall final design. The output after applying this second component to the process would outline a new, unique complex tensegrity structure to be constructed.

3.2 Phase 2: Evolution and Mutation

Following the transformation step, the second component will use an evolutionary algorithm to mutate the resulting graphs. There will be two types of mutations that can be applied to the graphs as well as crossover that two graphs can undergo. A primary mutation occurs when a change is made to the grammar that is used to transform and grow the graph. As mentioned in the original paper, Hornby et al. [1] showed that even a slight change in the map L-system grammars can drastically change the resultant graph and create new and unique potential structures. A crossover will take two parent sets of rules and create a child by selecting rules from both sets to create a new set of grammars.
A secondary mutation directly mutates the final state of a graph. There are two types of secondary mutation, either swapping the destination nodes of two cable, or the destination nodes of two struts. Both of these types of secondary mutations allow for a structure to change slightly to better explore possible forms without straying too far from it's initial state unlike a primary mutation would.

3.3 Testing Results

After being optimized by the evolutionary algorithm, results will be tested using a simulator that allows for modeling representations of the evolved designs that the above processes produces. Additional modifications in this stage could be made to account for edges that theoretically intersect each other not at a node, so that it may be realistically fabricated for practical applications. The original study did this analysis using Open Dynamics Engine, however it is a C/C++ based simulator as the original study was almost entirely done in, while I am programming mine in Python. One option is to use PyODE, a python binding of ODE, although that has been untouched since 2013, while another option is pybullet. Ultimately however, I don't plan on committing to any one simulator until later in the process when I am able to throughouly assess which one will serve my purposes best.

4 Thesis Timeline

As of right now there are still some aspects of the final part of this project that I have yet to figure out, but I intend to iron out those details before starting this project in full. As such, the later part of this timeline is a little muddy but I expect it to crystallize a great deal once I have these details more set.

4.1 Winter 2024

Despite only now proposing this as a thesis project, I initially started working on this project at the end of 2023 as a part of Union's Evolutionary Robotics Lab. As a result I already have a notable head start on developing the programs that I detailed the behavior of in the methods section above. Thus far I have more or less fully developed the map L-system that does the heavy lifting of transforming and growing a resultant graph from the initial input. As a result of this early work I have also begun developing a familiarity with the graphviz package for python.

4.2 Fall 2024

One of the first things I plan to do after re-familiarizing myself with the code this fall is to reassess the one discrepancy between my version of the transformations compared to the original one. In both the original study as well as my own program, the graph in Figure 3 is evolved according to the same set of grammars.

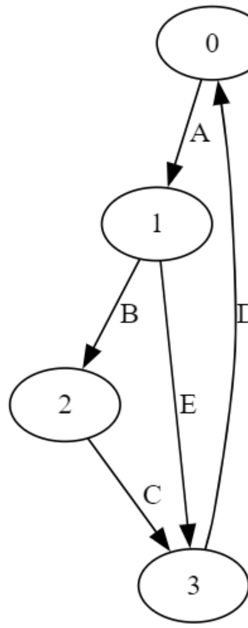

Figure 3: The graph that was created manually before growth with the map L-system.

Despite all the conditions being the same, the original study's result is different from the version that my program created as shown in Figure 4. I want to go back and reevaluate my transformer class to ensure it works as intended. I can also redo my manual by hand transformation of the graph to double check the work.

In the first week or so I also intend to do a much more in depth review of the code base that I've been provided by John. My plan is to really identify where exactly the methods that I plan to implement on my own are and use those as a point of reference for my own work.

Following that, the fall term will primarily consist of implementing the rest of the program's functionality and methods as I described above. Having fully reviewed my work so far in the first week, I will be able to hit the ground running and begin implementing the next component which would be what allows me to take a fully grown 2d graph and turn it into a comprehensible design for a new tensegrity structure. Doing this will require me to determine how exactly the program will decide whether a given edge will end up becoming a strut, cable, or discarded. The details of how it will do this is not yet ironed out and will be explored over the first few weeks of the fall while I implement it.

The next step would be to create methods for each of the possible mutations that could occur during the second phase of the program. This shouldn't prove too hard as it won't need to be making too many choices on its own and will just simply be performing a task verbatim.

The final piece of the puzzle that I would need to implement is the actual evolutionary algorithm that it uses to evolve and optimize the set of graphs that are output. For this I will have to determine by what parameters the algorithm selects optimal graphs, as well as determining what evolutionary algorithm I want to use in the first place. I wouldn't necessarily use the same algorithm that the original experiment uses and I hope to be able to explore other potential algorithms that could be used here.

I am unsure exactly how long designing all the pieces of this project will take, but I hope that I can finish this last component by week 8 so that I can dedicate some time this term to beginning experimentation and begin working on simulating and analyzing the tensegrity structures that the algorithm produces.

4.3 Winter Break 2024

In the worst case scenario that I am not quite done with developing the algorithm and other programs, I intend to dedicate as much time as needed during my winter break to finish that up. However, assuming that I will have already begun experimentation at the end of the fall term I anticipate not being able to do

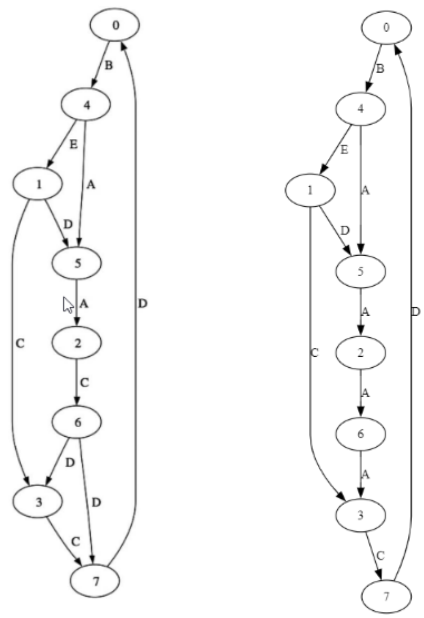

Figure 4: *Left:* The grown version of the graph in Figure 3 from [6]. *Right:* My version of the same initial graph grown using the same map L-system.

much over break as I likely will not have a machine that is capable of performing at the level that will be needed.

4.4 Winter 2025

With the start of the winter term I will begin the last leg of this project. At this point I will have at minimum, just finished designing the programs and algorithms to be used for the experiments. Regardless of how I ended the fall term I will be able to begin this term with data collection and analysis in earnest. This will be the last leg of this project and I will be primarily concerned with analyzing the results of the algorithm, making any adjustments to the program as necessary and completing work on writing the thesis in full. As I stated earlier, the finer details for the end of this research project are not quite clear yet but I aim to be finishing up with the end of this term.

5 Why Me?

I believe that I am a good candidate for working on this project for my thesis for many reasons, chief among them being that I am passionate and excited to research and explore the topic further. When I initially joined the Evolutionary Robotics Lab I knew very little about soft robotics, but as I've spent more time as part of the lab I have grown interested in the different applications of these kinds of robots. When the opportunity to begin working on this as a project was given over this past winter term I quickly requested to take it on. As both a computer science major and a former mechanical engineering major I am thoroughly invested in this project in all its facets. During my time as an engineering student I took both PHY-120 as well as MER-201, providing me a strong base of knowledge in structural principles and equilibrated systems from which I can draw on while working on this.

Following my change to being a computer science major I immediately joined the Evolutionary Robotics Lab and enthusiastically began doing what I could to learn as much as possible to participate and contribute to the lab quickly. I've enthusiastically taken on this project already and feel that the work I've done both in and outside the lab in my classes at Union have prepared me for continuing this project as my senior thesis.

References

[1] Greg Hornby, Hod Lipson, and Jordan Pollack. "Generative Representations for the Automated Design of Modular Physical Robots". In: *Robotics and Automation, IEEE Transactions on* 19 (Sept. 2003), pp. 703–719. DOI: 10.1109/TRA.2003.814502.

[2] Sergi Hernàndez Juan and Josep M. Mirats Tur. "Tensegrity frameworks: Static analysis review". In: *Mechanism and Machine Theory* 43.7 (2008), pp. 859–881. ISSN: 0094-114X. DOI: https://doi.org/10.1016/j.mechmachtheory.2007.06.010. URL: https://www.sciencedirect.com/science/article/pii/S0094114X07001218.

[3] K. Koohestani. "Form-finding of tensegrity structures via genetic algorithm". In: *International Journal of Solids and Structures* 49.5 (2012), pp. 739–747. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2011.11.015. URL: https://www.sciencedirect.com/science/article/pii/S0020768311003970.

[4] Milenko Masic, Robert E. Skelton, and Philip E. Gill. "Algebraic tensegrity form-finding". In: *International Journal of Solids and Structures* 42.16 (2005), pp. 4833–4858. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2005.01.014. URL: https://www.sciencedirect.com/science/article/pii/S0020768305000351.

[5] Chandana Paul, Hod Lipson, and Francisco J. Valero Cuevas. "Evolutionary form-finding of tensegrity structures". In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. GECCO '05. Washington DC, USA: Association for Computing Machinery, 2005, pp. 3–10. ISBN: 1595930108. DOI: 10.1145/1068009.1068011. URL: https://doi.org/10.1145/1068009.1068011.

[6] John Rieffel, Francisco Valero-Cuevas, and Hod Lipson. "Automated discovery and optimization of large irregular tensegrity structures". In: *Computers & Structures* 87.5 (2009), pp. 368–379. ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2008.11.010. URL: https://www.sciencedirect.com/science/article/pii/S0045794908002575.

[7] Sergei Savin, Oleg Balakhnov, and Alexandr Klimchik. "Energy-based local forward and inverse kinematics methods for tensegrity robots". In: *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. 2020, pp. 280–284. DOI: 10.1109/IRC.2020.00051.

[8] Sergei Savin, Oleg Balakhnov, and Alexandr Klimchik. "Workspace Analysis of Tensegrity Structures with an Iterative Energy-based Algorithm". In: Dec. 2020, pp. 1–4. DOI: 10.1109/NIR50484.2020.9290196.

[9] Meijia Wang, Yafeng Wang, and Xian Xu. "Form-Finding and Evaluation of Spherical Tensegrity Toward Applying in Locomotive Robots". In: *Journal of Mechanisms and Robotics* 16.12 (Mar. 2024), p. 121004. ISSN: 1942-4302. DOI: 10.1115/1.4065072. eprint: https://asmedigitalcollection.asme.org/mechanismsrobotics/article-pdf/16/12/121004/7323314/jmr_16_12_121004.pdf. URL: https://doi.org/10.1115/1.4065072.

[10] Xian Xu and Yaozhi Luo. "Form-finding of nonregular tensegrities using a genetic algorithm". In: *Mechanics Research Communications* 37.1 (2010), pp. 85–91. ISSN: 0093-6413. DOI: https://doi.org/10.1016/j.mechrescom.2009.09.003. URL: https://www.sciencedirect.com/science/article/pii/S009364130900130X.

[11] Eduard Zalyaev, Sergei Savin, and Lyudmila Vorochaeva. "Machine Learning Approach for Tensegrity Form Finding: Feature Extraction Problem". In: *2020 4th Scientific School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR)*. 2020, pp. 265–268. DOI: 10.1109/DCNAIR50402.2020.9216799.