3-2018

A Better Way to Construct Tensegrities: Planar Embeddings Inform Tensegrity Assembly

Elizabeth Anne Ricci
Union College - Schenectady, NY

A Better Way to Construct Tensegrities:

Planar Embeddings Inform Tensegrity Assembly

By

Elizabeth Ricci

* * * * * * * * *

Submitted in partial fulfillment

of the requirements for

Honors in the Department of Computer Science

UNION COLLEGE

May, 2018

Abstract

RICCI, ELIZABETH ANNE A Better Way to Construct Tensegrities:
Planar Embeddings Inform Tensegrity Assembly. Department of Computer Science, May, 2018.

ADVISOR: John Rieffel

Although seemingly simple, tensegrity structures are complex in nature which makes them both ideal for use in robotics and difficult to construct. We work to develop a protocol for constructing tensegrities more easily. We consider attaching a tensegrity's springs to the appropriate locations on some planar arrangement of attached struts. Once all of the elements of the structure are connected, we release the struts and allow the tensegrity to find its equilibrium position. This will allow for more rapid tensegrity construction. We develop a black-box that given some tensegrity returns a flat-pack, or the information needed to perform this physical construction.

Contents

1 Introduction

Classical robots, with wheels, are largely limited to known, regular terrains, but there is a need for more adaptable robots that can be sent out into unknown or irregular areas. For example such robots could help with search and rescue efforts following natural disasters or be used for planetary exploration. In response to this need, the area of soft robotics has emerged, and with it tensegrity robots.

A tensegrity structure is composed of rigid struts under compression and tensile strings under tension. These structures are stable despite the fact that none of the struts are directly connected. Tensegrities have an equilibrium state in which the sum of the net forces of the elements is minimal. In this state all of the tensile elements are under tension. Even after experiencing large amounts of force, such as falling from a great height, tensegrity structures will revert back to their equilibrium state. Therefore, their struts can be pushed together into a compact bundle for storage purposes. When released the tensegrity will revert back to its equilibrium state. This makes the tensegrity an appealling structure for planetary exploration; it can be compacted to fit on a spacecraft and then dropped onto a new planet, reverting back to its original shape. Additionally, there are a wide variety of tensegrity structures of varying sizes and complexities [16].

As a result of their unique structure, many researchers have begun using tensegrity structures as a basis for robots [1, 5, 7, 10, 15, 17, 18]. Due to their structure, tensegrities have a large amount of vibration and dynamic coupling, which make it difficult to model them accurately using physics simulators. Therefore much of tensegrity robotics research requires physical tensegrity structures. Unfortunately, due to the tension in the tensile elements of tensegrities, they are very difficult to construct by hand. Additionally, 3-D printers cannot be used to build a tensegrity completely, as the tensile element cannot be printed. Therefore many researchers only use small, simple and regular tensegrity structures, such as the standard 6-strut. If tensegrity structures were easier to construct, more complex structures could be used in tensegrity robotics.

We consider that by applying forces to various points on a tensegrity structure we can alter its form. When the forces are released, the tensegrity will revert back to its equilibrium position. Can we determine the proper locations to apply force such that the tensegrity's struts are planar? We call this planar configuration a "flat-packed model", as the entire tensegrity structure appears to be flat and nearly 2-dimensional. It is not necessary for a flat-packed model to have a strut configuration with no overlaps; the example in Figure 1 does have overlapping struts. Ideally, the strut overlap would be minimized while remaining within the bounds allowable by the maximum lengths of the tensile elements.

To what extent can we develop an algorithm, given a tensegrity structure and some constraints, that can be used to produce a flat-packed model of that tensegrity which meets the given constraints and will revert back to its equilibrium state upon release? When a flat-packed model of a tensegrity is known it can be

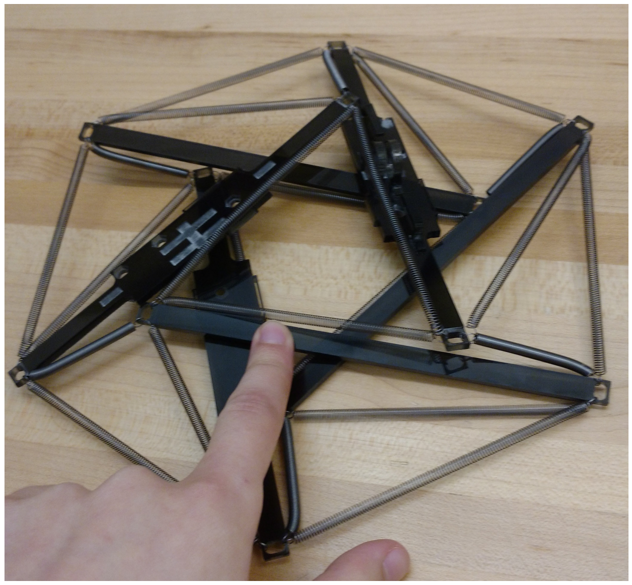


Figure 1: A flat-packed model of a regular 6-bar tensegrity.

used as a model to attach tensile elements to a flattened strut configuration. After all of the tensile elements have been attached, the forces holding the struts in 2-dimensions can be released to allow the tensegrity to spring into its 3-dimensional, low-energy, equilibrium state.

Some constraints that may be considered are: maximum length of the tensile elements, maximum area of the flat-packed form, and the allowable overlap of struts (Can two overlap? What about 3 overlapping at the same point?). This constraints can be customized to fit within the limitations of the materials available to build the tensegrity.

If an algorithm can be developed to solve our research question, we will have a much simpler way in which to build tensegrity structures. This will allow for more physical tensegrities, of various forms, to be used for further research. If an algorithm cannot be developed for an arbitrary tensegrity, limitations on the inputs should enable an algorithm to be developed for some number of tensegrity structures. Even if this is not the case, researching this topic will increase our understanding of the various forms a given tensegrity can take.

We hypothesize that for every tensegrity structure, there exists a 2-dimensional configuration of the struts with interwoven springs such that when released, the tensegrity's equilibrium state is achieved. This configuration will be determined through an algorithmic approach and verified through physics engine simulations and physical models.

2 Related Works

2.1 Tensegrity Robots

Over the past decade, tensegrity-based robots have been developed. We examine to what extent physical tensegrity structures are used when designing and developing these robots and the variety of tensegrity strucutres used.

In 2006, Paul et al. studied forward locomotion for three and four strut tensegrities [15]. The movement was developed and optimized in simulation using the open dynamics engine. At the end of the project a physical model of the 3-strut robot was created. The researchers found that the physical and simulated models did not match exactly. This highlights the fact that tensegrity structures are too complex to study in simulation alone; physical structures are needed to verify and further develop methods built in simulation. Additionally, the group only considered very simple tensegrity structures and they did not study the 4-strut tensegrity out of simulation.

Standard 6-strut tensegrities are a very popular base for tensegrity robots. Shibata et al. developed an approach for designing 6-strut tensegrity robots that move through a crawling motion [18]. Intuitively, there seem to be other tensegrity structures more suitable for crawling, such as the tower in Figure 2. The same 6-strut structure was used as a "skeletal structure" for a rolling soft robot [10].

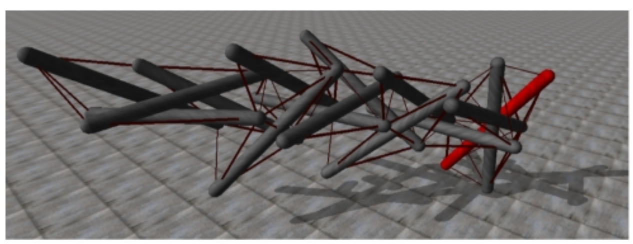


Figure 2: A 10-strut tensegrity tower. [16]

A third method of movement was developed on the same tensegrity structure by Khazanov et al. [7]. This project exploited the vibration and dynamic coupling between tensegrity components to achieve motion. The group stated that research applying these methods to more complex tensegrity structures is limited due to the inability to accurately model such complex structures in simulation and the inability to easily construct anything more than a simple tensegrity structure.

An example of a more complex tensegrity robot is the DuCTT, an 8-strut tensegrity robot, developed by Friesen et al. [5]. The researchers are developing this robot to explore duct systems. The group began by using simulations and then a physical model was built. Modifications were necessary to make up the difference between simulation and the physical world.

As the inability to accurately model tensegrities has been cited as an obstacle to studying tensegrities virtually, it should be noted that environments have been built to study tensegrity movement. Caluwaerts et al. developed two software environments to study tensegrity movement: the Euler-Lagrange simulator and the NASA Tensegrity Robotics Toolkit [1]. The researchers validated these programs against a physical 6-strut tensegrity, but not against any more complex structures. Additionally, such simulators cannot be verified without the use of physical tensegrities.

Even NASA has settled on a simple 6-strut tensegrity structure. The SUPERball is being designed specifically for space exploration [17]. It is not logical that this single structure would be ideal for all tensegrity robots, and it seems that it is the default structure to use due to its regularity, simplicity and ease of construction. If there was an easier way to construct complex tensegrity structures, it would allow for researchers to more easily experiment with various structures for their robots.

2.2 Self-Constructing and Printed Robotics

Tensegrity robotics is not the only area of robotics in need of a more efficient way to construct physical robots. In recent years, the field of self-construction robotics has emerged as a way to create robots with less human intervention. In general, these method include creating the elements of the robot through 3-D printing and/or laser-cutting, connecting the elements such that they all exist in a single plane, and using physical properties to cause the robot to transform into its desired form.

In 2013, Felton et al. used a 3-D printed circuit board and laser-cut layers to develop a robot that folded from a single plane to an inchworm shape [4]. This robot was not perfect and required some human intervention during the construction process as it had insufficient torque to flip into the proper position. This idea was improved and modified in 2015 by Onal et al. who used methods from origami to create a robot that completely self-constructed and achieved motion in its final form [14]. The entire process took less than 4 hours and $50 to create.

In addition to robots that self-construct, methods have been developed to simplify the construction process, without the robots actually constructing themselves. Trimmer et al. introduces a 3-D printing approach to create a soft robot with two different materials [22]. This resulted in a soft robot, capable of inching and crawling motions, that required much less hands-on construction than would typically be required. In response to a difficulty with adding strings and fibers to robot structures, such as hands, using typical approachs, Kimmer et al. developed a number of methods to simplify this task [8].

This work is significant as it demonstrates that new methods for robot construction are necessary, and that it is possible to simplify the construction processes of a large variety of robot structures using unconventional approaches. These ideas served as inspiration for simplifying the tensegrity construction process.

2.3 Mathematics of Tensegrities

In "Mathematics and Tensegrity: Group and representation theory make it possible to form a complete catalog of "strut-cable" constructions with prescribed symmetries" Connelly et al. provides a definition for the equilibrium state of a tensegrity structure [2]. In short, a tensegrity is in equilibrium when the form has a minimal energy state. This can be modeled through energy functions for the structure's tensile elements and the same functions, but with negative proportionality constants, for the strut elements. Additionally, the authors introduce methods for considering tensegrities in group theory. These could be be used to describe similar flat-packed forms between group elements.

Tensegrities can also be considered from a graph theory perspective. So et al. provides a graph theory definition for tensegrities:

> A tensegrity $G(\mathbf{p})$ is a graph $G = (v, E)$ together with a configuration $\mathbf{p} = (p_i) \in R^D \times ... \times R^D = R^{|V|D}$ such that each edge is labelled as a cable, strut or bar and each vertex is labelled as pinned or unpinned [19].

It remains to be shown how this definition can be used to find alternative arrangements of tensegrity struts and if this can be used in conjunction with force direct graph drawing (see section 2.5).

Linear algebra is also commonly used to model tensegrities. Lui et al. used linear algebra to represent the forces of a simple 3-strut tensegrity [11]. Once the researchers understood a simple tensegrity, components were added to create new stable tensegrity structures. The use of smaller tensegrity structures with known qualities to build up larger structures allows for the existence of large, complex structures with known components. This would allow for a piece-wise understanding and modeling of large structures. Additionally, researchers have used linear algebra to characterize the static equilibria of tensegrities with very few variables, for use in the design and simulation of tensegrities [23]. It is interesting to consider how these principles can be applied to applying forces to equilibrium tensegrity structures to find planar embeddings.

2.4 Tensegrity Structures

It is non-trivial to discover tensegrity structures beyond the small regular ones that are commonly used, such as the regular 6-strut tensegrity. Rieffel et al. developed an evolutionary algorithm to represent and evolve large tensegrity structures [16]. This algorithm is able to produce a large variety of tensegrity structures and describe classes of tensegrities. This work will be used to provide arbitrary tensegrities to flat-pack. Additionally, we may be able to develop general flat-packed forms for tensegrity classes. This work also describes a rapid prototyping machine for creating tensegrities. This machine 3-D printed tensegrities in their equilibrium state with solid elements in the place of tensile elements. These elements were then replaced, by hand, with rubber bands. This approach for tensegrity construction did not end up being much easier than constructing them fully by hand, and is no longer being used. Another example of alternative methods for tensegrity construction is presented in "A Review of a Family of Self-Deploying Tensegrity Structures with Elastic Ties" [3]. The approach presented in this paper has a very limited scope, as it can only be applied to a specific tensgrity class.

Finally, in the past year 3-D printing has been used in the construction of tensegrity structures. Lui et al. constructed a range of tensegrity structures through the use of shape memory polymers [12]. This method allows for tensegrities to be constructed fairly quickly with minimal human intervention. Zappetti et al. present a similar method for tensegrity construction, but the shape shifting struts are replaced with a folding network of elastic cables to which ridged struts are attached [24]. We also note that currently this method has only been applied to constructing the regular six strut tensegrity structure. Furthermore, both of these methods construct tensegrities with some elastic printed material acting as the tensile element, thus the elasticity of the structure has limited variation as opposed to one that can be constructed with any variety of extension spring.

Due to their ability to compress into much smaller forms, tensegrities are of interest for activities such as deploying satellites. The mechanics behind tensegrity based satellite deployment is detailed by Tibert et al., from which we can see that such methods are what Tur et al. would call passive deployment, in that external forces are required to transition the structure from its compact configuration to its deployed position [20, 21]. This differs from our approach which utilizes the energy present in the tensegrity's springs from being compressed to allow it to revert to its equilibrium position upon release, without any external assistance.

2.5 Force Directed Graph Drawing

A graph is simply defined by vertices and edges, the actual positioning of these elements in space is relative, as such there are an infinite number of ways in which any given graph can be drawn in the plane. Force directed graph drawing is a method for drawing graphs through the use of physical properties in order to achieve some simplest drawing. For example, the vertices of a graph can be modeled as elements with repulsive forces and the edges of a graph can be modeled as springs with spring forces. Forces of each

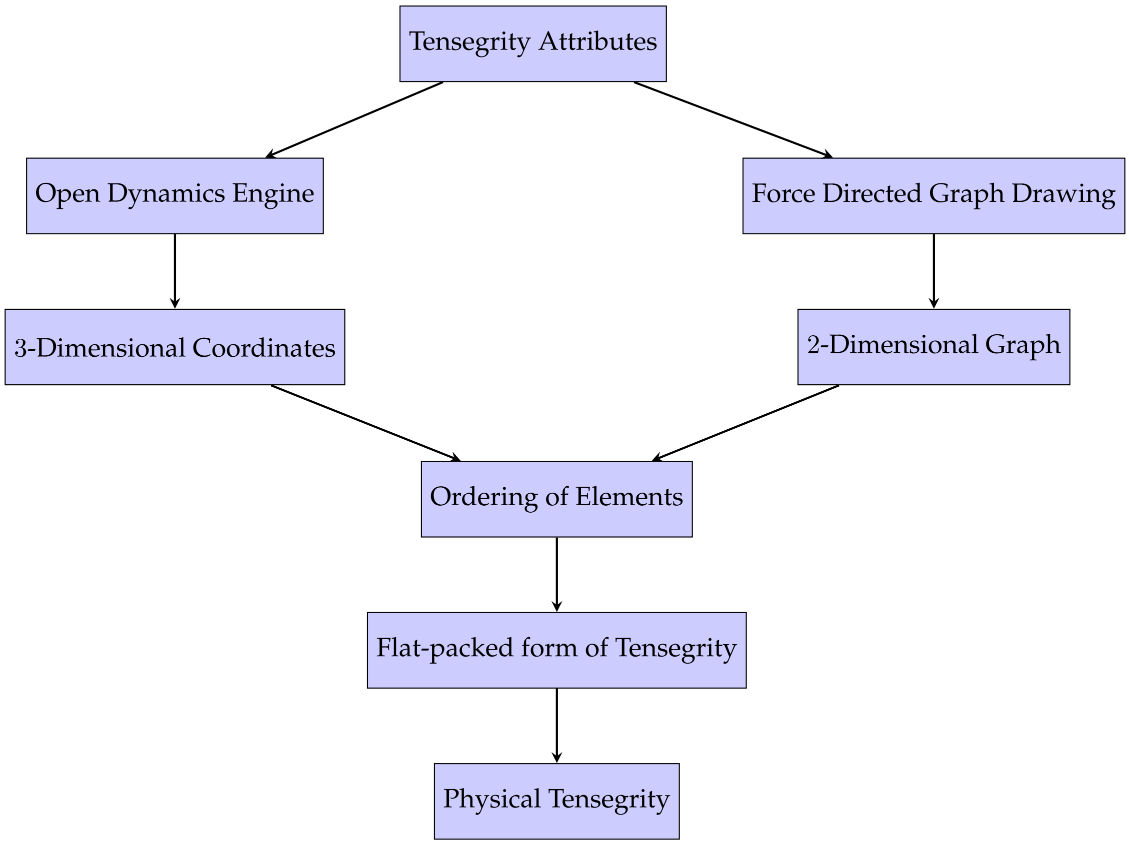


Figure 3: A High Level Solution.

vertex in the graph are calculated over the course of time until an equilibrium state is reached, at which point the graph should have minimal edge overlap. Algorithms for this process are introduced in a number of papers [6, 9, 13]. Such algorithms have not been used to draw tensegrity graphs, but we hope that they can be applied to this field to find planar embeddings of tenesgrity struts.

3 Methods

At a high level I have created a black-box that takes a tensegrity structure as input and outputs the flat-packed form that can be used to physically construct the physical tensegrity. The flat-packed form consists of a two-dimensional graph of the tensegrity and a text file detailing the ordering of each overlap present in the graph. Figure 3 shows the steps that occur within the computational black-box, each step is detailed below. For the purposes of this section all of the example images will be of the three-bar tensegrity.

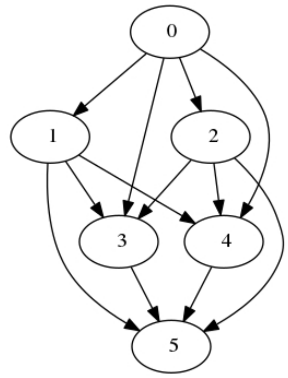


Figure 4: A graph generated from the input graphviz .dot file for the 3-bar tensegrity.

3.1 Tensegrity Attributes

The black-box program takes a tensegrity represented as a graphviz dot file as input. I choose this format as this is how tensegrities are returned from Rieffel et al's tensegrity generation program [16]. In the input file the tensegrity is modeled as a graph were each strut endpoint is a vertex and the struts and springs are represented by edges, see Figure 4. Each endpoint of a strut is assigned an index, ranging from zero to twice the number of struts. These indices are assigned such that the top vertex of each strut is even and the corresponding bottom vertex is one greater than the index of the top. This enables the struts and springs to be represented in the same way, while still being distinguishable. By parsing this dot file the number of struts, number of springs and an array of all the struts and springs in the given tensegrity are determined and written to a text file. This is all the information that is necessary to progress to the next stage, I will refer to this data collectively as the tensegrity attributes.

3.2 Three-Dimensional Model

The tensegrity attributes are used to generate a three-dimensional model of the given tensegrity. This model is created through the Open Dynamics Engine physics simulator. This program takes the text file created in the previous step (section 3.1) as a parameter. The information in the text file is used to automatically model the given tensegrity. Once all of the struts and springs are added to the simulation, the simulation form-finds an equilibrium state for the tensegrity, see Figure 5. Once an equilibrium has been reached the three-dimensional coordinates of each strut endpoint are written to a text file, to be used in a later step (section 3.4). It should be noted that some amount of control is lost by creating a program robust enough to generate any tensegrity in this fashion, but as our work does not rely on the actual image generated this should not impact our results.

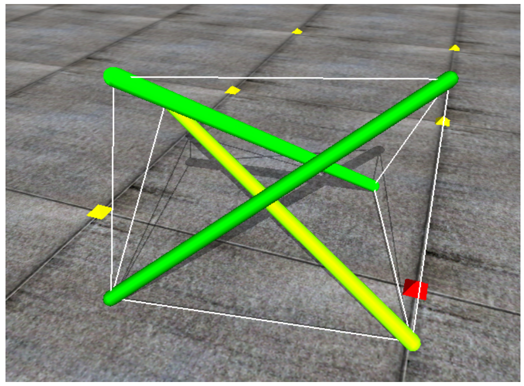


Figure 5: The 3-dimensional model for the 3-bar tensegrity.

3.3 Two-Dimensional Model

Once the three-dimensional model has been used to calculated the coordinates of the equilibrium state tensegrity, the final program can be run. This program computes the information necessary for the two-dimensional model and to generate the flack-packed form (covered in section 3.4). This C program takes the file containing the tensegrity attributes (from section 3.1) and the file containing the three-dimensional coordinates (from section 3.2). The graph representing the flat-packed form and a text file containing the ordering data are produced.

The first step in this program is generating a two-dimensional model of the tensegrity, which is done through force directed graph drawing, see section 2.5. Before the graph drawing process can begin, the graph must be modeled. This is done by transferring the information from the tensegrity attributes to a graph object, with the number of vertices equal to twice the number of struts, and an edge between all the pairs of indices listed in the array of struts and springs. The forces and two-dimensional coordinates of each vertex are stored in the graph object. I have chosen the set the initial positions of each vertex such that the struts are parallel to the x-axis and slightly offset in the y direction. This provides a starting configuration with no strut overlaps, which is important for the graph drawing and selection process. The graph can be drawn to a PPM file, when this is done the struts are drawn in blue, while the springs are green. An initial graph drawing for the three-bar is shown in Figure 6a.

After the initial graph is created a graph drawing program, based on [13], is applied to the graph. For some constant number of times, I have been using 60000, the forces of each vertex are updated based on the repulsive forces between each pair of vertices and the spring forces on all the edges that represent springs. For each iteration the coordinates of each vertex is updated based on the updated forces. When physically constructing the tensegrity each strut will be the same length, I have incorporated this limitation into the program by adjusting the position the second vertex in each strut is updated to. The line that the strut

(a) The initial drawing of the 3-bar tensegrity.

(b) The final drawing of the 3-bar tensegrity.

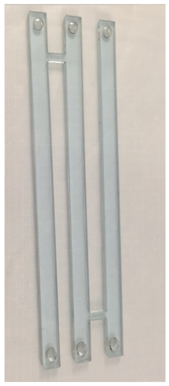
Figure 6: Two-dimensional models of the 3-bar tensegrity.

would fall on is not altered, only the length of the line segment. Note that in some of the example graphs the strut lengths do not appear to be equal, this is as a result of scaling that occurred when writing the graph to an image file; when calculated from the coordinates of the vertices each strut is 50 units long.

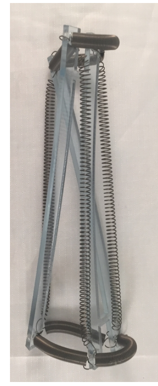
Ideally graph drawing programs halt when the system's equilibrium state is reached, unfortunately there is no way to guarantee that this occurs with a single set of constants and an arbitrary graph. In order to circumvent this issue, while still choosing some "best" drawing, I compare the graphs generated throughout the process and choose the one that aligns best with our needs. Our ideal graph would have no strut overlaps, and minimize both the number of spring crossings and the total energy of the system. Since strut-over-strut crossings would prevent the struts to be laser cut and connected in a single sheet of material during the physical construction process, see section 3.5, I require the final graph to have no such overlaps. This is possible to ensure due to how I create the initial graph drawing. A new graph is deemed better than a prior one if the number of intersections did not increase and the total energy of the system did not increase. As the graph drawing runs, the attributes of the best drawing are updated as a better graph is created, after the program has been run for the specified number of iterations the attributes of the current best graph are used to recreate that drawing. This final graph drawing is written to a PPM file and is the graph portion of the flat-pack. The best drawing for the three-bar can be seen in Figure 6b. We note that the third strut is difficult to see as it is intersecting with an adjacent spring.

3.4 Flat-Packed Form

The second step of the program introduced in section 3.3 determines the ordering of each intersection present in the final graph to be used in the flat-pack. This is done by first locating add of the intersections present in the graph, achieved by simple linear methods. Next for each intersection present in the graph the three-dimensional layering information is determined. This can be done by considering the three-dimensional coordinate pairs for each line involved in the intersection. Using these coordinates the program finds the x-interception of these lines projected in the xz-plane. Lastly the canonical equation of a line is



(a) The laser cut struts.



(b) The constructed tensegrity before the struts are separated.

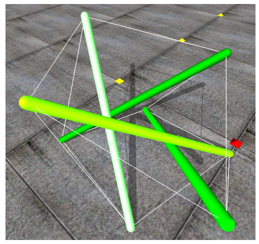
(c) The final form of the constructed tensegrity.

Figure 7: Physical construction of 3-bar tensegrity using graph from Figure 6a.

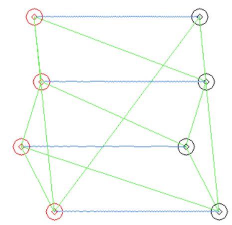
used to find the y value that corresponds to the x-intercept value calculated. The line with the larger y value should be on the top. This information is written to a text file. At this point the flat packed form is complete and is composed of the PPM file of the graph from 3.3 and the text file containing the pairs of edges that intersect and the ordering of the elements involved.

3.5 Physical Construction

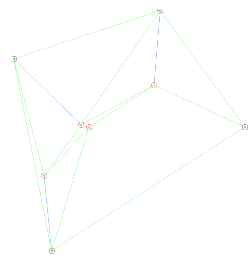
Once the flat-packed form is known, a physical construction of the tensegrity can be attempted. Note that the images include are not representative of a construction from the graph from Figure 6b, instead the naive drawing shown in Figure 6a was used. First struts should be laser-cut out of acrylic material according to the arrangement in the flat-packed graph. Connections should be included, such that all of the struts remain slightly connected, see Figure 7a. Next springs or rubber bands should be attacked to the strut endpoints as specified by the flat-packed form, see Figure 7b. Finally once all of the springs are attached the struts can be separated, allowing the structure to find the desired equilibrium position, Figure 7.



(a) The 3-dimensional model.



(b) The initial graph drawing.



(c) The best found graph drawing.

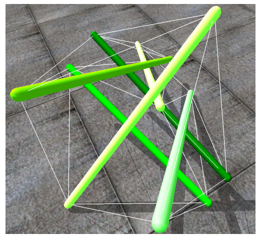
Figure 8: Example of 4-bar tensegrity.

4 Results

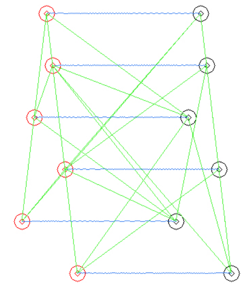
The goal of this project was to "develop an algorithm, given a tensegrity structure and some constraints, that can be used to produce a flat-packed model of that tensegrity which meets the given constraints and will revert back to its equilibrium state upon release." We have successfully developed and implemented an algorithm that given a tensegrity produces a flat-pack model of that tensegrity. The algorithm is robust enough to work on any tensegrity, although we are not guaranteed to generate a flat-packed form that is feasible to physically construct. Included are three more examples of the information produced by this process, see Figures 8, 9, 10. In addition this method is time efficient, at least for relatively small tensegrities. The 15-strut example took about a minute and a half total to run through the black-box. It is significant that a black-box has been created that accept any tensegrity, represented in the correct format, and output, relatively quickly, a graph and overlap data, even if the given solution is not physically feasible. This being said, my program has a number of limitations.

The graph drawing program is not as effective as it should be. While it is able to select the best graph generated, it is likely that the ideal arrangement, that minimizes the tensegrity's energy, is never being generated. In addition this program does not consider the limitations of a spring; a spring has a maximum length it can stretch before it deforms. By ignoring this constraint it is likely the program generates final graphs that can not actually be physical constructed.

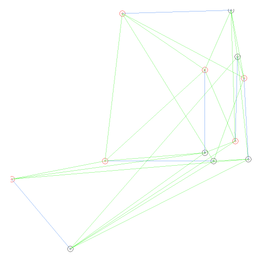
A physical construction of a four strut tensegrity using this process, shown in Figure 11 shed light on some issues in translating this computational work to the physical world. An initial difficulty is translating



(a) The 3-dimensional model.

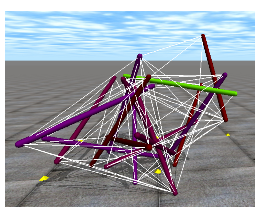


(b) The initial graph drawing.

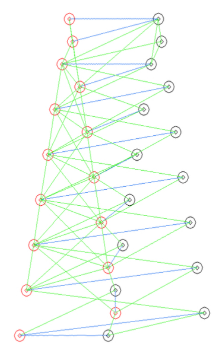


(c) The best found graph drawing.

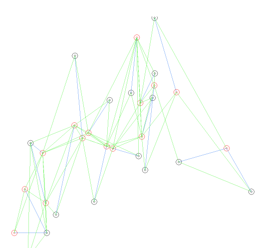
Figure 9: Example of 6-bar tensegrity.



(a) The 3-dimensional model.

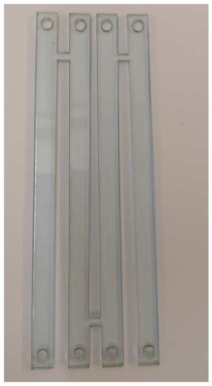


(b) The initial graph drawing.



(c) The best found graph drawing.

Figure 10: Example of 15-bar tensegrity, from Rieffel et al [16]. Note: all of the struts (in blue) are all the same length, they appear distorted in these images.



(a) The laser cut struts.

(b) The constructed tensegrity before the struts are separated.

(c) The final form of the constructed tensegrity.

Figure 11: Physical construction of 4-bar tensegrity using graph from Figure 8b.

the graph produced into struts laser cut out of acrylic. A further difficulty is the limitations of the materials used. In the examples included in this paper wire springs were used, these proved difficult to securely fasten to the struts which lead to further complications during construction. Finally, in order for the structure to take the desired form the length of the struts and springs, as well as the spring constant used, must be in the proper proportions. This being said, I cannot make any claims about the extent to which the flat-pack generated for any given tensegrity will translate into the physical construction process, but the foundations to test and improve this area have been put in place.

5 Future Work

Our future work will work on addressing the limitations mentioned in section 4. Specifically the graph drawing process will be improved to generate even lower energy graph systems. In addition the ability to consider more constraints, namely the maximum spring length, will be added. A further limitation in our results is the lack of an automated way to go from the final graph to a file that can be given to a laser-cutter and produce the struts in the position indicated by the graph. In the future we will add this to the black-box, this will enable physical verification of this process to be faster and easier. Once this is complete, more physical construction tests should be performed to further verify the method and to identify physical limitations of the process. Finally we consider altering the materials used so that springs do not need to be added by hand, but instead an elastic material can be positioned with the acrylic sheets in such a way that both the struts and the springs can be laser-cut.

References

[1] Ken Caluwaerts et al. "Design and control of compliant tensegrity robots through simulation and hardware validation". In: *Journal of The Royal Society Interface* 11.98 (2014), p. 20140520.

[2] Robert Connelly and Allen Back. "Mathematics and Tensegrity: Group and representation theory make it possible to form a complete catalogue of "strut-cable" constructions with prescribed symmetries". In: *American Scientist* 86.2 (1998), pp. 142–151. ISSN: 00030996. URL: `http://www.jstor.org/stable/27856980`.

[3] J Duffy et al. "A review of a family of self-deploying tensegrity structures with elastic ties". In: *Shock and Vibration Digest* 32.2 (2000), pp. 100–106.

[4] Samuel M Felton et al. "Robot self-assembly by folding: A printed inchworm robot". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 277–282.

[5] Jeffrey Friesen et al. "DuCTT: A tensegrity robot for exploring duct systems". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 4222–4228.

[6] Thomas MJ Fruchterman and Edward M Reingold. "Graph drawing by force-directed placement". In: *Software: Practice and experience* 21.11 (1991), pp. 1129–1164.

[7] Mark Khazanov et al. "Exploiting Dynamical Complexity in a Physical Tensegrity Robot to Achieve Locomotion." In: *ECAL*. Citeseer. 2013, pp. 965–972.

[8] CJ Kimmer and CK Harnett. "Combining Strings and Fibers With Additive Manufacturing Designs". In: *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2016, V004T05A014–V004T05A014.

[9] Stephen G Kobourov. "Spring embedders and force directed graph drawing algorithms". In: *arXiv preprint arXiv:1201.3011* (2012).

[10] Yuusuke Koizumi, Mizuho Shibata, and Shinichi Hirai. "Rolling tensegrity driven by pneumatic soft actuators". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 1988–1993.

[11] Heping Liu, Jinsong Geng, and Ani Luo. "Tensegrity configuration method for connecting tensegrity units along their axes". In: *Composite Structures* 162 (2017), pp. 341–350.

[12] Ke Liu et al. "Programmable Deployment of Tensegrity Structures by Stimulus-Responsive Polymers". In: *Scientific Reports* 7.1 (2017), p. 3511.

[13] Michael J McGuffin. "Simple algorithms for network visualization: A tutorial". In: *Tsinghua Science and Technology* 17.4 (2012), pp. 383–398.

[14] Cagdas D Onal et al. "Origami-inspired printed robots". In: *IEEE/ASME Transactions on Mechatronics* 20.5 (2015), pp. 2214–2221.

[15] Chandana Paul, Francisco J Valero-Cuevas, and Hod Lipson. "Design and control of tensegrity robots for locomotion". In: *IEEE Transactions on Robotics* 22.5 (2006), pp. 944–957.

[16] John Rieffel, Francisco Valero-Cuevas, and Hod Lipson. "Automated discovery and optimization of large irregular tensegrity structures". In: *Computers & Structures* 87.5-6 (2009), pp. 368–379. ISSN: 0045-7949. DOI: DOI:10.1016/j.compstruc.2008.11.010. URL: http://www.sciencedirect.com/science/article/B6V28-4V995R7-3/2/f539af780b2794a82f889965cd292e68.

[17] Andrew P Sabelhaus et al. "System design and locomotion of SUPERball, an untethered tensegrity robot". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 2867–2873.

[18] Mizuho Shibata, Fumio Saijyo, and Shinichi Hirai. "Crawling by body deformation of tensegrity structure robots". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 4375–4380.

[19] Anthony Man-Cho So and Yinyu Ye. "A semidefinite programming approach to tensegrity theory and realizability of graphs". In: *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics. 2006, pp. 766–775.

[20] AG Tibert and S Pellegrino. "Deployable tensegrity reflectors for small satellites". In: *Journal of Spacecraft and Rockets* 39.5 (2002), pp. 701–709.

[21] Josep M Mirats Tur and Sergi Hernàndez Juan. "Tensegrity frameworks: Dynamic analysis review and open problems". In: *Mechanism and Machine Theory* 44.1 (2009), pp. 1–18.

[22] Takuya Umedachi, Vishesh Vikas, and Barry A Trimmer. "Highly deformable 3-d printed soft robot generating inching and crawling locomotions with variable friction legs". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 4590–4595.

[23] Darrell Williamson, Robert E Skelton, and Jeongheon Han. "Equilibrium conditions of a tensegrity structure". In: *International Journal of Solids and structures* 40.23 (2003), pp. 6347–6367.

[24] Davide Zappetti et al. "Bio-inspired Tensegrity Soft Modular Robots". In: *Conference on Biomimetic and Biohybrid Systems*. Springer. 2017, pp. 497–508.