Bagh-e Nazar

Bagh-e Nazar, *21*(134), 39-46 / Aug. 2024
DOI: 10.22034/BAGH.2024.429271.5549

Persian translation of this paper entitled:
فرم‌یابی دیجیتال سازه‌های درختی براساس آزمایش رشته‌های خیس
published in this issue of journal

Original Research Article

Digital Form-finding of Tree-like Structures Based on Wet Threads Experiment

Mona Alibabaye Lavasani[1*], Mohammad Reza Matini[2], Saeed Khaghani[3]

1. M.A. in Architectural Technology, School of Architecture, College of Fine Arts, University of Tehran, Iran.
2. Associate Professor in Architecture, Department of Architecture , Faculty of Architecture and Urban Design, Iran University of Art,Tehran.
3. Assistant Professor, School of Architecture, College of Fine Arts، University Of Tehran, Iran.

Received: 25/01/2023; accepted: 19/06/2024; available online: 22/07/2024

Abstract

Problem statement: Self-organizing particles in nature have long inspired structural form-findings. These forms shaped to transfer forces efficiently use minimal materials and are light-weight. Physical models have been used to explore these self-organizing particles and served as a basis for design and calculation. However, making, measuring, and scaling these models is tedious and hard, especially for complex geometries such as tree-like structures. Nowadays, computer simulations can apply nature's logic to create digital models. These models simulate form-finding and scaling faster and easier.

Research objective: The purpose of this research is to present a digital tool derived from algorithmic design for the digital form finding of branching structures based on the physical testing of a wet thread model.

Research method: This research was first formed through the study of available resources and scientific articles in this field, and then the results were used to design digital tools using computational methods.

Conclusion: Algorithmic design based on the wet thread model simplifies the optimal design of tree-like structures. It optimizes both the design outcome and the design process. Physical form-finding often faces difficulties in converting models into construction plans. By digitizing this process, the measurement of the final form becomes faster and easier. This enhances the constructability of these forms.

Keywords: *Self-Organizing Patterns, Digital Form-finding, Algorithmic Design, Tree-like Structures.*

Introduction

Throughout history, architects have been looking at natural shape formation as a model for the architectural organization. (Pathak, 2019). Natural forms emerge from the evolutionary process of placing their particles in an optimal arrangement (Otto & Rasch, 2001). However, these naturally complex structures are not the result of complex processes. They come from organizing patterns under suitable conditions (Isaacs, 2008). Based on this, Frei Otto (1925-2014), the German Architect, built physical models for form-finding efficient and lightweight structures (Spuybroek, 2005). And the "Wet Threads Experiment" was one of those models. In this model, the threads restrained between the supports stick together due to the surface tension of the water to form an optimal network that can be used to find the efficient form of tree-like structures (Otto & Rasch, 200). However, the form-finding process through these models appears easy, but its measuring and extrapolation to a 1:1 scale is very challenging (Kilian, 2004). This problem could be solved by creating the physical model in the computer and simulating how it works.

Research Problems

The main question of this research is "How we can

---

*Corresponding author: +989102019627,mona.lavasani.92@gmail.com use design algorithms to simulate the organizing process of wet threads in the computer for form-finding of optimal tree-like structure?". Many studies have been done on the role of physical models in design, structural calculation of architectural forms, and how to create digital models based on the logic of their physical counterparts. However, the focus of this research is on the "Wet Threads" model built by Frei Otto to design optimal branching structures. Besides the question above, there are two sub-questions here:

1. How the emergent process in the wet threads model could end up with optimized branching structures?
2. How mathematical algorithms could be used to create a simulating model through algorithmic design?

Methodology

This research has two sections. The first one allowed us to study the previous research on this field and how the current algorithms could be developed to be used as a simulation tool in mimicking the process of the wet threads experiment. And section two is about writing the algorithm in Grasshopper which is a parametric plugin for the Rhino software and asses the accuracy and correction of its result in form-finding of tree-like structures.

Theoretical Foundations and Research Background

- **Biomimetic design**

Nature is a database of versatile strategies that can be used in problem-solving. "Biomimicry" or "Biomimetic Design" literally means to mimic or imitate nature. However, in this research, it is defined as a design based on the way nature responds to problems (Pathak, 2019). Natural form formation is a dynamic process and should be considered as a moving picture, not a static and instantaneous one. Examining this process reveals the underlying logic of natural organization (Otto & Rasch, 2001). Inspired by nature, architects can achieve optimal design by making logical decisions and eliminating superfluous geometries at the onset of the design process. Both natural and man-made structures arise from mathematical principles, shaped by the analysis of physical forces and the minimization of excess geometries. Such an approach enhances both the aesthetics and the functional efficiency of form-finding in architecture (Dixit et al., 2020)

- **Self-organizing systems**

These systems are dynamic ones that change themselves through the interaction between their consisting particle to achieve better sustainability (Banzhaf, 2003). The 18th-century philosopher Kant was the first to define this phenomenon. He said that in an organism, each part owes its existence to the characteristics and function of other parts that form the whole system (Karsenti, 2008). Natural examples of this pattern include group swimming of fishes, the movement of a column of ants, the formation of termite mounds, or the growth pattern of lichen (Camazine et al., 2020), And the urban traffic network and the Internet are its man-made examples (Banzhaf, 2009).

- **Physical form-finding**

The form of a structure is effective in its structural performance. Before the age of computers, the design of complex forms was based on the construction of physical models that showed the optimal arrangement of force transmission (Isaacs, 2008). The forces in such forms are pure and axial (fully tensile or fully compressive), and due to the elimination of bending stresses, the consumption of materials and finally the dead load of the structure is minimized. Therefore, it can be said: "The form follows the forces." (Veenendaal & Block, 2012). The forming force in these models was the weight of the material and gravity. Antonio Gaudi's (1852-1926) hanging chains model, Felix Candela's (1910-1997) thin skins, Heinz Isler's (1926-2009) ice models Sergio Musmeci's (1926-1981) soap film models, and Fry Otto's cable networks are examples of these models. And the history of these models goes back to Robert Hook's (1635-1703) chain model (Li et al., 2017).

- **Analog computer**

The basis of these models is a concept called "Graphic Statics", which was expressed by Simon Stevin (1548-1602) in 1586. His two-dimensional model was able to check the balance of forces visually and without the need for complex calculations (Block et al., 2006). Physical form-finding models were three-dimensional ones that were used for structural optimization and calculations so that mathematical calculations were no longer needed.

But the final design should also be drawn by studying and measuring the exact same models. However, the precise interpretation of these models was not simple and error-free. Isler and Otto both offered solutions to solve this problem. Isler designed a three-dimensional grid table to capture the coordinates of the points that make up the model. After form-finding and measuring the forces at each point, the coordinates of each of these points had to be taken carefully, which according to Isler himself was a very exhausting task. But Otto offered a faster and more accurate solution, which was taking pictures from different angles of the model (Fig. 1.). By putting these photos together, an accurate and measurable model was quickly obtained (Whitehead, 2016). Otto called this way of finding the optimal form by physical models and its formation process "analog computer" (Fabricius, 2016). And in 1970, he introduced a standard for comparing models called "Bic". Bic's criterion was derived from the comparison of the relationship between the three factors of form, force, and mass. In this way, he was able to provide a quantitative criterion for accurate measurement of light structures (Spuybroek, 2005). But in this method, time, cost, manpower, and even a lot of space were required to build the model, which made the design method out of optimal condition. In addition, the smallest error in the model would magnify on the real scale, and it would have a destructive effect on the structural performance of the form (Kilian, 2004).

- **Tree-like structures**

The complex shape of trees is formed in response to their structural and biological needs. These forms appeared in architecture since the Middle Ages, in the ribbed vaults and fan vaults of Gothic architecture, like the Kings Cambridge College church. And in the 19th century, it reached its peak during the "Art Nouveau", in the Grand Palace building located in Paris. Because the engineers of that time had become skilled in using cast iron (Md Rian & Sassone,2014). The form of trees has two distinct characteristics in terms of its mechanical behavior: first, the diameter of the branches is proportional to the amount of load they bear, due to hierarchical load transfer. Second, the length and angle of the tree branches depend directly on the force to be transferred. Therefore, this system does not favor long or horizontal branches that bend and break due to gravity. (Mattheck, 1991). In addition, the thin branches of the tree can resist the lateral forces of wind, water flow, or even earthquakes as a damper (Kovacic et al., 2018). The most important feature of tree structures is the short paths of load transmission from the point of force application to the supports. This feature inspired Frei Otto's experiments on form finding of tree structures. In the early 1990s, he and his group at the Institute of Lightweight Structures in Stuttgart researched "Optimized Path Systems " and built several models for designing tree structures: the suspended model (based on Gaudi's model), the dry threads model, and the wet thread model (Ahmeti, 2007).

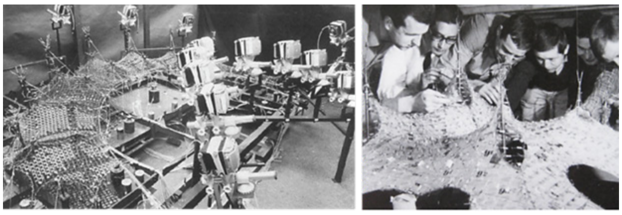

Fig. 1. The process of calculating and drawing the design of the German pavilion at the Montreal Expo 1967, by the Frei Otto team at IL. Source: Whitehead, 2016.

- **Shortest path networks**

Shortest-path networks represent the shortest connections between a set of nodes. In nature, river deltas, lightning, and even tree-leaf veins are examples of these networks. But Fry Otto had several methods to find these optimal networks, one of which was using soap film. He connected two transparent plates to each other with some short cylinders, each cylinder representing the position of that node on the plate, and by immersing this set in a mixture of soap and water, The soap film created a path and formed with the minimum energy expenditure. This showed the shortest possible path between the nodes. In another experiment, he simulated urban sprawl with magnetic needles, iron particles, and their magnetic force (Burkhardt, 2016). These two-dimensional patterns were used to find optimal urban paths and traffic patterns. However, the wet threads model could provide a three-dimensional pattern of optimal paths.

- **Wet Thread experiment**

In this experiment, first, the support points or nodes are identified and strings of thread are tied between them, the length of each string is equal to 8% more than the straight distance between the nodes. So, the strings are not fully stretched, and they have a sag. This sag or increased length may vary slightly from this number for various reasons and provides the threads flexibility and the possibility of movement (Spuybroek, 2005). However, the wet threads experiments differ from the other two models (suspended model and dry threads) in that they can optimize the branch lengths of tree structures (Fig. 2). The result of these two models was the optimization of the branches' angles, not their lengths. However, the wet threads model, similar to the soap film experiment, can merge the initial paths to obtain a set of new shortest paths.

Discussion

• Digital form-finding

At the Institute of Structural Mechanics, at the University of Stuttgart, some scientists created computational tools for designing and optimizing tree structures. However, none of them could simulate the wet threads experiment process (Von Buelow, 2007). Because the programs relied on mathematical algorithms and did not account for the internal forces that shaped the structure.

• Finding suitable algorithm

Computer programs operate on algorithms. Therefore, designing an algorithm is the first step in creating a digital tool. Unlike mathematical algorithms that use mathematical formulas and produce immediate output, physical algorithms simulate a process by following the behavior of materials and the laws of physics (Lopes et al., 2014). In data visualization, some algorithms work like a wet threads model. They use graphs, which have nodes and edges, to show a set of data and their connections. But if there are too many nodes and edges, the graph becomes messy and unclear. The solution is to make the edges curved instead of straight (Holten & Van Wijk, 2009). Similar to the threads in a water container that stick together to reduce the system's complexity and create an optimal structure, data visualization uses three algorithms: 1. Geometry-based Edge Bundling (GBEB) (Cui et al., 2008), 2. Hierarchical Edge Bundles (HEB) (Holten, 2006), and 3. Force-Directed Edge Bundling (FDEB). The first two algorithms are mathematical, but the FDEB algorithm is considered physical due to the presence of forces. In this algorithm, each of the straight lines between the nodes is divided into several parts, these divisions turn the system into smaller components, to provide the necessary flexibility to respond to applied forces (Fig. 3.). In fact, the initial lines in this method are considered as springs. The spring-like behavior is achieved by increasing the length of the line segments that make up each line. There are two factors for the changes in the length of each line segment, the first is the force that creates repulsion between the nodes and the points resulting from the division and increases the length of the line segments, and the second is the coefficient that determines the stiffness of the spring which control the amount of length change. After this stage, it is necessary that the line segments, which now become longer due to the spring-like behavior, should be attracted to the corresponding line segments in their neighboring lines and merged so that the desired optimal curved network emerges (Holten & Van Wijk, 2009). The general method for determining the form of tree-like structures is illustrated in (Fig. 4).

• Designing the digital tool

The behavior of the system in the wet threads model is very similar to the FDEB algorithm. So, it can be used as the basis of the simulation algorithm. The designing steps of the wet threads model algorithm which is done in the Grasshopper plugin are as follows:

- Part 1: The initial graph includes support points and straight lines representing how these points are related to each other.

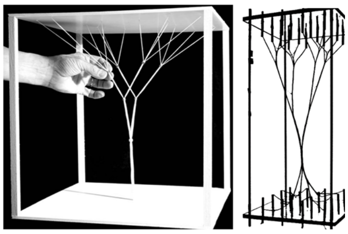

Fig. 2. Left: dry threads model, Right: wet threads model. Source: Von Buelow, 2007.

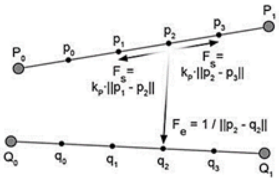

Fig. 3. The straight lines are divided into several segments. The resulting points are the system's components. (Fs): repulsive force between points on each line, and (Fe): attractive force between matching points on a set of lines. (Kp): a coefficient that regulates the springiness of the lines. Source: Holten & Van Wijk, 2009.

- Part 2: Dividing straight lines into shorter segments, which can be based on the number of divisions or the length of divisions. It is better to make this division based on the length of the resulting line segments, because if the lines are not the same size, the shorter lines will have much smaller line segments than the longer lines, and this may cause errors in the answer.

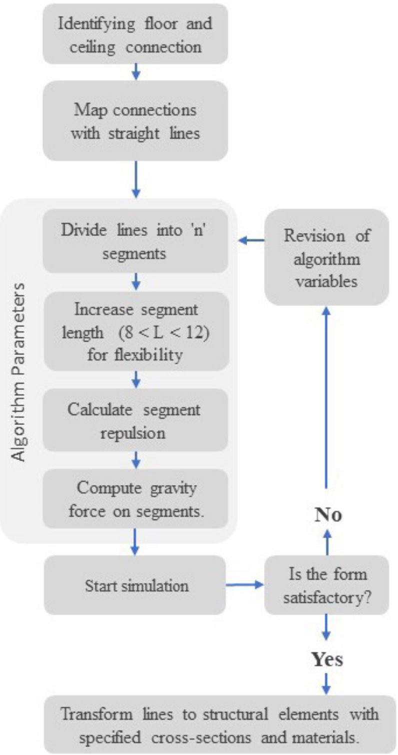

Fig. 4. Flowchart of form finding based on wet threads algorithm. Source: Authors.

- Part 3: After dividing and producing new line segments, it is time to turn the lines into springs, which is done by increasing the length of the new line segments, and by applying repulsive force from the points resulting from the division. This increase in length can vary between 8% and 12% of the original length of the line segments. Changes in this value affect the final form.

- Part 4: After turning the lines into springs, the necessary force must be produced to attract the line segments to each other. This force will be applied correspondingly between each point resulting from the division of the second stage. It is important to control the intensity of this force because different results can be obtained by changing it.

- Part 5: In this section, the Kangaroo physical simulation engine receives the system forces, which is the repulsion force between the springs and the attraction force, besides anchor points and the overall geometry of the system as input. By performing the simulation in the form of an animated image, the final result will emerge after a few seconds. The algorithm and its different steps can be seen in Figs. 5 &.6.

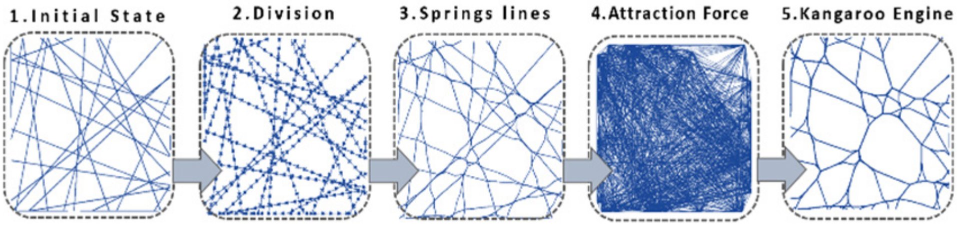

Fig. 5. Displaying the shape transformation in each step of the wet threads algorithm. Source: Authors.

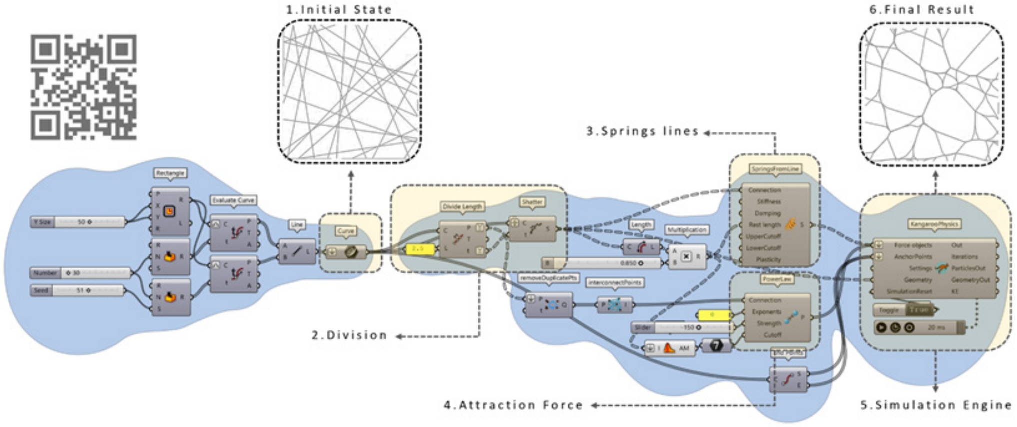

Fig. 6. Algorithm of wet threads model in Grasshopper plugin, along with the simulation display QR-code. Source: Authors.

- **Evaluating the responsiveness of digital tools**

The algorithm was applied to two examples (two-dimensional and three-dimensional) to find and optimize their forms, before designing the tree-like structure with it. This was to demonstrate its accuracy and performance.

- Steiner Tree problem (2D)

Steiner's mathematical algorithm finds an optimal path between multiple points, such as a traffic route in a city. The optimal path is not any of the initial lines, but a new and curved one. Steiner's Algorithm can also be verified with a soap film shape between two transparent plates pined with some small cylinders (Lopes et al., 2014). As can be seen in (Fig. 7), the results of the soap film and the wet threads algorithm are the same. But this is a two-dimensional example, and the response of the algorithm should be tested in a three-dimensional example as well.

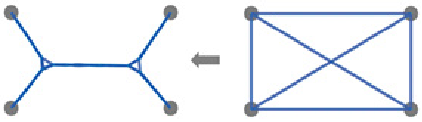

Fig. 7. All the possible direct paths, then the optimal path resulted from digital form-finding of the wet threads in the Grasshopper, and its simulation QR-code. Source: Authors.

- Gothic tree columns (ribbed vaults)

The ribbed vaults that support the ceiling of Westminster Abbey Hall are also a kind of tree-like structure. Therefore, the wet threads algorithm was used to find the shape of these arches, to test the algorithm's accuracy in three dimensions (Fig. 8).

- **Digital form-finding of tree-like structures**

Tree columns distribute the load to the foundations more evenly by spreading their branches across the roof, and thus have finer dimensions. However, the connection points of the roof to each support must be properly divided. Moreover, the branch-to-roof connection point is also crucial. The design assumed four supports on the ground and divided the roof into four sections. Each section had ten random connection points, with correct locations optimized using the genetic algorithm. By connecting the points of each roof section to its respective support point, four groups of straight lines are formed, which take up a lot of space. Therefore, the arrangement of these lines needs to be optimized. This could be done by the wet threads' algorithm. The result is a complex and unpredictable form (Fig. 9).

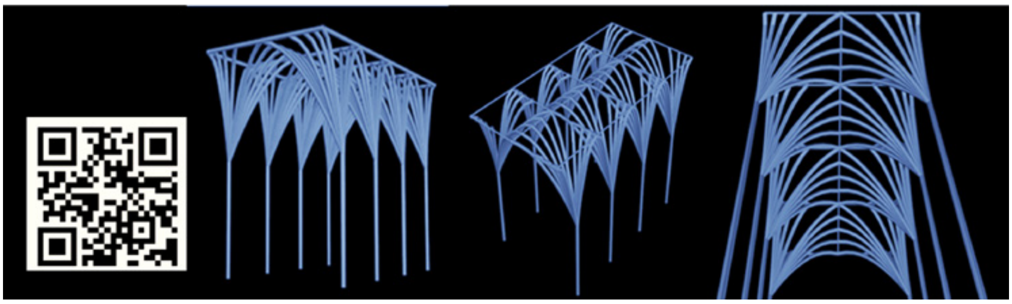

Fig. 8. Digital form-finding of the ribbed vaults of the Westminster church, using wet threads algorithm, and its simulation QR-code. Source: Authors.

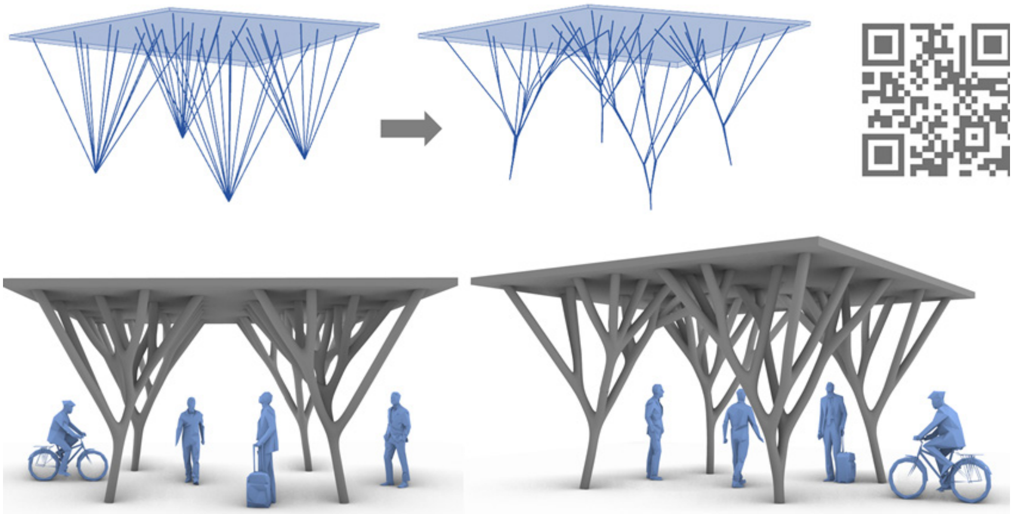

Fig. 9. Digital form-finding a tree-structure pavilion with wet thread algorithm in Grasshopper. Source: Authors.

As mentioned, the algorithm depends on variables such as the length increase of the line segments after splitting the lines, and the attraction force intensity between the matching points based on distance, for optimizing such structures. However, this research did not find a specific relationship for the proper adjustment of this coefficient in the algorithm. If the computer simulation speed can be improved with a better algorithmic design, these values can be optimized for each design (Fig. 10). by using multi-objective optimization, and thus, more refined and optimal forms can be found. For instance, if the attraction force between the points, which makes the lines bundle, is considered lower, the structure will have severe bending stresses as the length of the resulting branches increases. Too much of this force will also make the branches more horizontal and cause bending and beam-like behavior in the structure. It should be noted that the mentioned algorithm is only able to optimize the correct extension of tree structures and the diameter and diameter change of structural members cannot be determined at this stage. These could be topics for future research.

Conclusion

In the wet thread algorithm, the forces optimize the system components to achieve minimum disturbance and minimum material consumption. This feature can be very useful in finding the form of tree structures, especially their complex types. The use of these algorithms in the design process turns the computer into a virtual laboratory. A laboratory that conducts experiments faster than its real type and requires less cost and time from the designer. Although the digital tool introduced in this research has limitations, and the forces in the tree-like structures resulting from this algorithm may not be pure, or the thickness of the structural elements may not be determined, it can be used in the design process to facilitate the designing of complex tree-like structures.

References list

- Ahmeti, F. (2007). *Efficiency of Lightweight Structural Forms: The Case of Tree-like Structures-A comparative Structural Analysis* (Unpublished master's thesis in Science). Building Science & Technology, Vienna University of Technology, Austria. https://papers.cumincad.org/data/works/att/bsct_ahmeti.content.pdf .16.
- Banzhaf, W. (2003). Self-Organizing Systems. In R. A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology.* Academic Press. 590.
- Banzhaf, W. (2009). Self-organizing Systems. In R. A. Meyers (Ed.), *Encyclopedia of Complexity and Systems Science.* Springe, 8040–8050.
- Block, P., DeJong, M. & Ochsendorf, J. (2006). As Hangs the Flexible Line: Equilibrium of Masonry Arches. *Nexus Network Journal, 8*(2), 13–24. https://doi.org/10.1007/s00004-006-0015-9
- Von Buelow, P. (2007). A Geometric Comparison Of Branching Structures In Tension And Compression Versus Minimal Paths. *IASS Conference: International Association of Shell and Spatial Structures*, Venice, Italy. https://www.academia.edu/6725876/A.
- Burkhardt, B. (2016). Natural structures - the research of Frei Otto in natural sciences. *International Journal of Space Structures, 31*(1), 9. https://doi.org/10.1177/0266351116642060
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G. & Bonabeau, E., (2020). *Self-Organization in Biological Systems.* Princeton University Press.
- Cui, W., Zhou, H., Qu, H., Chung Wong, P. & Li, X. (2008). Geometry-Based Edge Clustering for Graph Visualization: IEEE. *Transaction on Visualization and Computer Graphics, 14*(6), 84-1277. http://.doi.org/10.1109/TVCG.2008.135
- Dixit, S., Stefańska, A. & Musiuk, A. (2020). Architectural form finding in arboreal supporting structure optimisation. *Ain Shams*

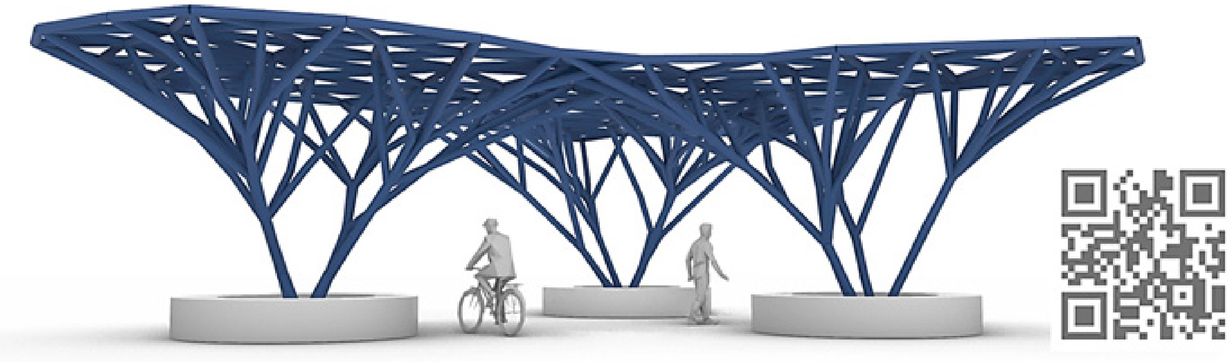

Fig. 10. Another form of branching structure designed by this algorithm, along with a simulation of its QR code. Source: Authors.

Engineering-Journal, 12(2), 2321-2329. doi:https://doi.org/10.1016/j.asej.2020.08.022.

- Fabricius, D. (2016). Architecture before architecture: Frei Otto's 'Deep History'. *The Journal of Architecture, 21*(8), 1253-1273. https://doi.org/10.1080/13602365.2016.1254667
- Holten, D. (2006). Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *The 11th Eurographics/ IEEE Transaction on Visualization and Computer Graphics, 15*(5), 741-748. http://dx.doi.org/10.1109/TVCG.2006.147
- Holten, D. & van Wijk, J.J. (2009). Force-Directed Edge Bundling for Graph Visualization. *The 11th Eurographics/ IEEE-VGTC Symposium on Visualization, 28*(3), 983-990. https://doi.org/10.1111/j.1467-8659.2009.01450.x
- Isaacs, A.J. (2008). *Self-Organizational Architecture: Design Through Form-Finding Methods* (Unpublished master thesis's in Architecture). The Academic Faculty, Georgia Institute of Technology.
- Karsenti, E. (2008). Self-organization in cell biology: a brief history. *Nature Reviews Molecular Cell Biology, 9*(3), 255-262. https://doi.org/10.1038/nrm2357
- Kilian, A. (2004). Linking Hanging Chain Models to Fabrication: *The 24th Annual Conference of the Association for Computer Aided Design in Architecture*. ACADIA. https://doi.org/10.52842/conf.acadia.2004.110
- Kovacic, I., Miodrag, Z. & Dragi, R. (2018). Sympodial tree-like structures: from small to large amplitude vibrations. *Bioinspiration & Biomimetics, 13*(2), 026002–026002. https://doi.org/10.1088/1748-3190/aa9d1c
- Li, Q., Su, Y., Wu, Y., Borgart, A. & Rots, J. G. (2017). Form-finding of shell structures generated from physical models. *International Journal of Space Structures, 32*(3), 11–33. https://doi.org/10.1177/0266351117696577.
- Lopes, J.V., Paio, A.C. & Sousa, J.P. (2014). Parametric Urban Models Based on Frei Otto's Generative Form-Finding Processes. *Proceedings of the 19th Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*, 595-604. https://doi.org/10.52842/conf.caadria.2014.595
- Mattheck, G. C. (1991). *Trees: The Mechanical Design*. Springer-Verlag. https://books.google.com/books/about/Trees.html?id=i8PqCAAAQBAJ
- Md Rian, I., & Sassone, M. (2014). Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview. *Frontiers of Architectural Research, 3*(3), 298–323. https://doi.org/10.1016/j.foar.2014.03.006
- Otto, F. & Bodo R. (2001). *Finding form: towards an architecture of the minimal*. Edition Axel Menges.
- Pathak, SH. (2019). Biomimicry: (Innovation Inspired by Nature). *International Journal of New Technology and Research, 5*(6). https://doi.org/10.31871/ijntr.5.6.17
- Spuybroek, L. (2005). The Structure of Vagueness. *Textile, 3*(1), 6-19. https://doi.org/10.2752/147597505778052620
- Veenendaal, D. & Block, P. (2012). An overview and comparison of structural form finding methods for general networks. International *Journal of Solids and Structures, 49*(15), 3741-3753. https://doi.org/10.1016/j.ijsolstr.2012.08.008
- Whitehead, R. (2016). Model Behavior: The Evolving Use of Physical Prototypes in Structural Shell Design, 1959-1974. In R. Corser & Sh. Haar (eds.), *Shaping new knoeledges*. ACSA , 114. https://www.acsa-arch.org/chapter/model-behavior-the-evolvinguse-of-physical-prototypes-instructural-shell-design-1959-1974/

HOW TO CITE THIS ARTICLE

Alibabaye Lavasani, M., Matini, M. R. & Khaghani, S. (2024). Digital Form-finding of Tree-like Structures Based on Wet Threads Experiment. *Bagh-e Nazar*, *21*(134), 39-46.

DOI: 10.22034/BAGH.2024.429271.5549
URL: https://www.bagh-sj.com/article_198609.html?lang=en