Proceedings of the International Association for Shell and Spatial Structures (IASS)
Symposium 2015, Amsterdam
Future Visions
17 - 20 August 2015, Amsterdam, The Netherlands

Potentials of branching algorithms for the generation of spaceframes in the context of high-rise perimeter structures

Samar MALEK[*], Maximilian THUMFART[a], Kavin HORAYANGKURA[a], Moritz FLEISCHMANN[a]

[*]United States Naval Academy
590 Holloway Road, Annapolis, MD 21402
smalek@alum.mit.edu

[a] HENN GmbH
Augustenstraße 54, 80333 Munich, Germany

Abstract

In this paper, we develop an innovative computer-assisted design strategy for a high-rise structure based on branching. Branching describes the splitting of one element (the seed) into two, while both, so called – children, change the direction of growth symmetrically. However, changing the angle of growth, the length of the members and the number of seeds can result in different, non-symmetrical topologies. The Grasshopper component that is created simulates branching through a series of rules that describe the varying direction of growth, and the merging of the branches. The key parameters that influence the growth of the topology include: the branch iterations, the angle between each branching pair, the length of the members and the number and location of seeds and attraction points. Through branching, three kinds of structures can be created: a diagrid, a space frame and a hybrid of the diagrid and space frame. While the user explores the design space and options, the tool conducts a preliminary structural analysis using Karamaba3D. It is found that there are many architectural and structural advantages to using branching including creating unique spaces, providing redundancy in the structure, alleviating vortex shedding with irregular facades and forming a triangular grid, the most stable topology.

Keywords: branching, diagrid, spaceframe, topologies.

1. Problem Statement

Driven by the increasing demand for super tall buildings, and the limiting supply of resources, there is a need to create sustainable structures. The objective of this work is to derive an innovative, computer assisted design strategy for a 450m high-rise building. The computational tool must integrate structure, space and program and account for real-world loadings and targeted structural weights.

2. Methodology

A prototypical 450m tall building is designed using a branching algorithm that "grows" structural members that also define the architectural space. This algorithm and computational tool is created through an iterative process that requires evaluating structural performance and spatial quality while updating the algorithm. The computational tool is a component coded in C#.NET and implemented in Grasshopper, a graphical algorithm editor used for parametric modeling in Rhino. While the user explores different designs, Karamba3D, a finite element tool-kit for Grasshopper, simultaneously performs a structural analysis designing for strength, stiffness and stability (Preisinger [3]). We will first explain the structural boundary conditions and then the branching algorithm. Because this is an iterative process, a set of structural case studies are proposed and their results are incorporated into the final design.

2.1. Loading Conditions

The 450m tall building is subjected to only dead loads and wind loads in both directions. As this is a preliminary design tool, we adhere to a few building codes, but not all. We consider a load combination of 1.2DL+1.2W where the dead load includes the self-weight of the structure plus an additional 8kN/m^2 to account for the self-weight of the floor, its finishes and the live load. The wind load is uniformly distributed at 1.5 kN/m^2. We neglect the additional strength and stiffness provided by floor slabs and assume that 100% of the lateral load goes to the branching structure. A core is assumed to be symmetrically located and takes 50% of the vertical load while the remaining 50% is taken by the branching structure. The displacement tolerance is $\Delta = H/500$ (4.5m) where H is the height of the building.

2.2. Boundary Conditions and Member Properties

The structure is simply supported at the foundation. The intersections of all members are modeled as rigid joints. The members are steel with a Young's Modulus of Elasticity of 21,000 kN/cm^2 and yield strength of 23.5 kN/cm^2. Each member is a hollow tube with an initial diameter of 87 cm and thickness of 9 cm; they are later optimized for material efficiency.

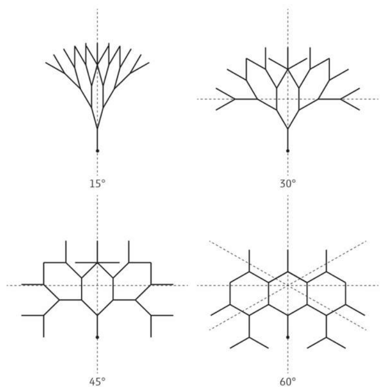


Figure 1: Illustration of varying topologies generated by branching

2.3. Proposed Branching Algorithm

Branching describes the splitting of one element into two, while both – so called - children change the direction of growth symmetrically. Figure 1 illustrates a simple tree that was generated by branching with a fixed number of iterations and member length. As can be seen in Figure 1, different topologies can result by changing the angle. Interestingly, at an angle of 60° a diagrid is formed.

Previous structures have used branching both aesthetically and structurally. For example, the Flughafen Stuttgart uses columns shaped like trees, the IBM building in Pittsburgh uses a uniform diagrid, and lastly, at Tod's Omotesando in Tokyo, the façade is a non-uniform branched structure that wraps around the building (Figure 2).

Figure 2: Tod's, Flughafen Stuttgart, IBM Building (clockwise from top left)

In architecture, branching began as a space making process as seen in the Flughafen Stuttgart (Otto [2]). However branching also has a potential for high-rise design. Our objective is to grow a 450m high-rise by a branching algorithm. Figure 3 illustrates the concept where a seed grows towards an attraction point, mimicking the behavior of a tree growing towards the sun. Starting at the seed the tree splits after a certain height into two branches, it continues this splitting process until it reaches the height of the attraction point. The attraction point helps us to focus the growth of our structure.

2.3.1 2-Dimensional Branching

In 2D, the parameters that influence the growth of our topology include: the number and location of seeds and attraction points, the number of branch iterations, the angle between each branching pair and the length of the members (Figure 4). Each branch pair tries to focus its symmetry axis to the attraction point along the focus line.

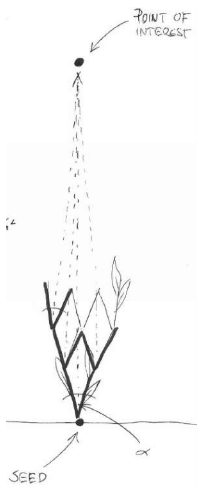


Figure 3: Illustration of the seed and attraction points

The algorithm allows the branches to "re-connect" and form a connected structure like a diagrid; instead of a branched structure with singular branches. Unlike an actual tree, our branches must re-connect and form a connected structure like a diagrid. By merging the branches, the structure itself becomes more stable. Around each member's tip there is a merging tolerance radius, which can force the tips to join if their tolerance circles are intersecting. We create two modes for merging: angle preservation or angle domination. The angle preservation mode tries to keep both angles at the same value while the domination mode keeps the left angle and shrinks the right.

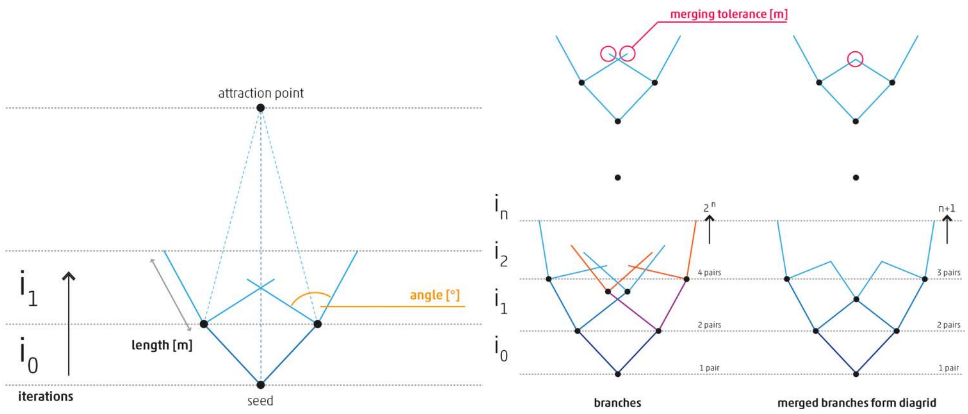


Figure 4: 2D Branching parameters

2.3.1 3-Dimensional Branching

In order to create a 3D structure, new rules are implemented to allow the structure to grow spatially. We define two 3D branching modes: two-branches or four-branches (Figure 5). The two-branch mode

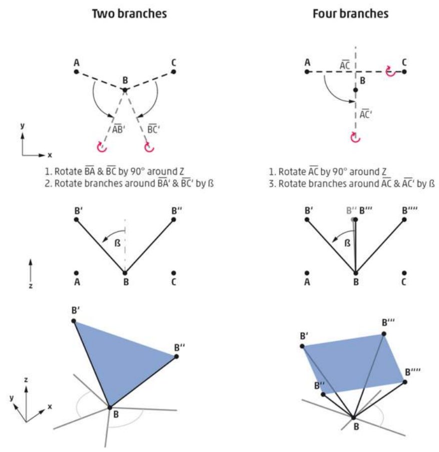


Figure 5: 3D Branching modes

creates two branches where each branch grows into the direction of the nearest neighbor. The four-branch mode creates four branches of which two are growing in one plane between the closest neighbors to both sides of the starting point. The other two branches are branching perpendicular to this plane.

To guide the 3D growth, the user can define "checkpoints" or "tree-zones" (Figure 6). The checkpoints force the structure to pass a certain predefined point if the branches are within the attraction radius of the checkpoint. The tree-zones can overwrite the growth parameters for certain areas to influence the structure due to program. For example, in a high-rise building there could be retail, office, or residential zones which would all require different spaces.

The aforementioned parameters and rules are all implemented into the Grasshopper component. The user can explore different structural topologies through the manipulation of the seeds, attraction points, checkpoints and treezones.

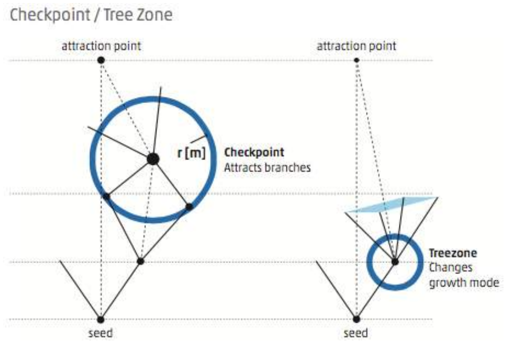


Figure 6: Branching checkpoints and tree-zones

2.4. Structural Case Studies

The branching algorithm can create two types of structures: a diagrid or a spaceframe. Several case studies are performed to determine how the topology and layout affect the structural performance. In all these studies we design for strength, stiffness and stability. The case studies include:

- Diagrid versus Spaceframe: Two towers of equal height and member lengths are created where one is a diagrid and the other a spaceframe. Both towers' member sizes are optimized for displacement and checked for strength (Figure 7).
- Location of Diagrid within a Hybrid Structure: The spaceframe has a more interesting spatial potential than the diagrid, while the diagrid is structural more efficient. We investigate the structural performance of a hybrid tower that is composed of both a diagrid and a spaceframe. We test the structural performance of three towers where the diagrid transitions to a spaceframe at 33%, 50% and 66% of the tower's height (Figure 7).
- Diagrid Topology – Member Angles: The topology of the diagrid is also governed by the angle of the members. We use Galapagos, an evolutionary algorithm within Grasshopper, to find the optimal angle to minimize the displacement at the top.
- Diagrid Topology – Member Lengths: The structural performance of three different diagrid topologies is studied. Three diagrid towers are created with different member lengths. The first tower has shorter members at the bottom growing to longer members at the top. The second tower considers the reverse, longer members are at the bottom and shorter are at the top. And the third tower has uniform member lengths (Figure 8).

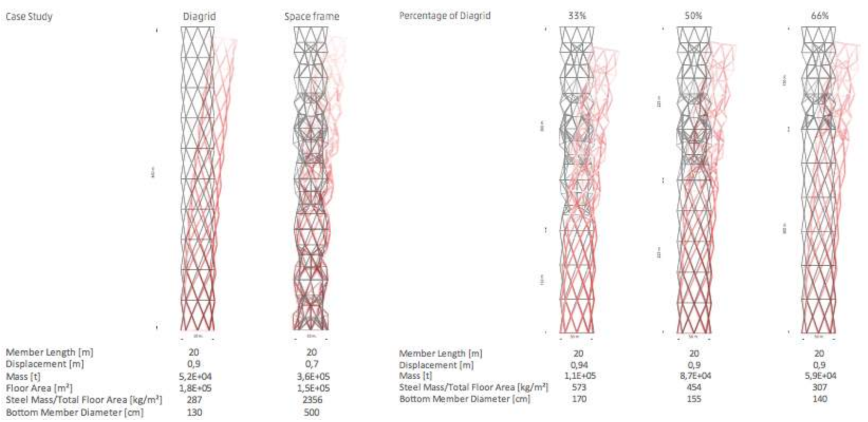


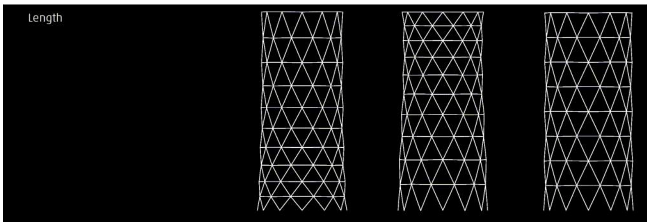
Figure 7: Comparing the structural performance of a diagrid to a spaceframe

3. Results

3.1 Structural Case Studies

The structure is designed for displacement, strength and stability. The structural case studies laid out in the Methodology Section yielded the following results.

- Diagrid versus Space Frame: the diagrid is very efficient. We compare the steel mass for the total floor area for the diagrid to the spaceframe and found that the diagrid is 10 times more efficient. As a result, our final tower design will include some portion of a diagrid.
- Location of Diagrid within a Hybrid Structure: For our hybrid tower, we place the diagrid at the bottom of the tower. It is best to put structure at the perimeter of the tower and especially at the base for structural stability. For architectural programming, the spaceframe is placed at the top because it created more unique spaces and in tall buildings the prime real estate is located at the top. Also for architectural programming, placing the diagrid at the bottom provided a column free floor plan which allows retailers to customize their spaces as they see fit. We found that a diagrid at 33% of the tower's height did not meet the displacement tolerance. In the next two towers where the diagrid was at 50% and 66% respectively, the 50% and 66% towers had the same displacement though the 66% tower is 32% lighter in steel weight than the 50% tower (Figure 7).
- Diagrid Topology (Member Angles and Lengths): The optimization analysis yielded an optimal angle of 70 degrees which agrees with the findings in Moon *et al.* [1]. We found that longer members at the bottom minimized the tower's displacement. In addition, the axial forces in the members are less which meant section sizes could be reduced (Figure 8).



	Short to Long	Long to Short	Uniform Length
Displacement [m]	0,21	0,17	0,2
Mass [t]	9,7E+03	9,1E+03	1,1E+04
Steel Mass/Total Floor Area [kg/m²]	108	101	114
Maximum Tension [kN]	2,5E+04	1,1E+04	1,5E+04
Maximum Compression [kN]	-4,5E+04	-3,3E+04	-4,3E+04
Max Member Diameter at Bottom [cm]	95	80	95

Figure 8: The effect of the diagrid topology

3.2 Final Tower Design

Given the lessons learned from our structural and spatial case studies, we designed a tower using our branching algorithm with the following characteristics: from the bottom to the top of the tower the member lengths decrease; the member angles are kept at approximately 70 degrees; and the diagrid is used for the bottom half of the structure before transitioning to a spaceframe.

The final design (Figure 9) was then structurally analyzed with the loading and boundary conditions outlined in the Methodology section. A member size optimization tool within Karamba3D was also used to reduce the member sizes. In summary, the final tower had: a displacement of 0.82m at the top, a ratio of steel mass (vertical structural members) to floor weight of 354 kg/m², and 8 unique section sizes that ranged from a diameter of 155cm and thickness of 40cm to a diameter of 50cm and thickness of 5cm.

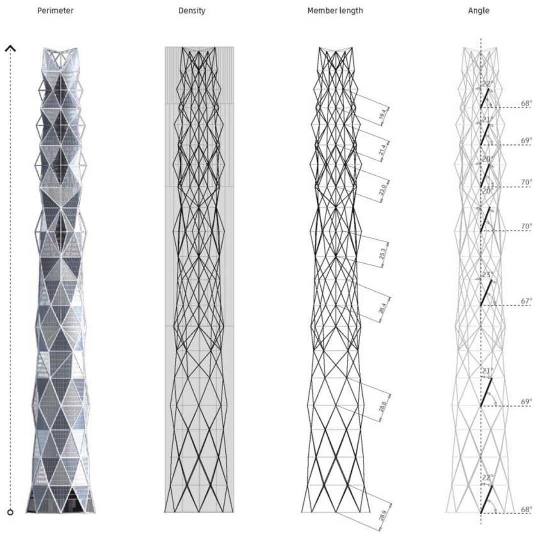


Figure 9: Key tower characteristics

4. Discussion

The branching algorithm can create many unique structures all while incorporating structure, space and program. Some structural advantages to our algorithm include its ability to create irregularities on and along the façade. The irregularities create disturbances to wind loading. In addition, the use of attraction points ensures that the structure will taper at the top, creating an aerodynamic shape. Branching also avoids structural frame discontinuity because all members must grow from each other. Lastly, due to the flexibility of placing seed points and attraction points, our algorithm can "grow" a tall building from any given footprint.

5. Conclusions

We created and tested a branching algorithm that could integrate structure, space and program into the design of a high-rise building. Our computational tool also allows the user to explore a large design space while accounting for structure, space and program. Our future work includes testing, modifying and adapting the algorithm for the design of other structures such as gridshells.

Acknowledgements

The authors would like to thank the Design Research Exchange (DRX) at HENN, created by Martin Henn and Moritz Fleischmann. The DRX brings together academics and practitioners in architecture, engineering, computer science and mathematics to derive innovative computer-assisted design strategies for high-rise structures. We would like to thank Clemens Preisinger and Moritz Heimrath of Bollinger Grohmann for their karamba workshops and advice.

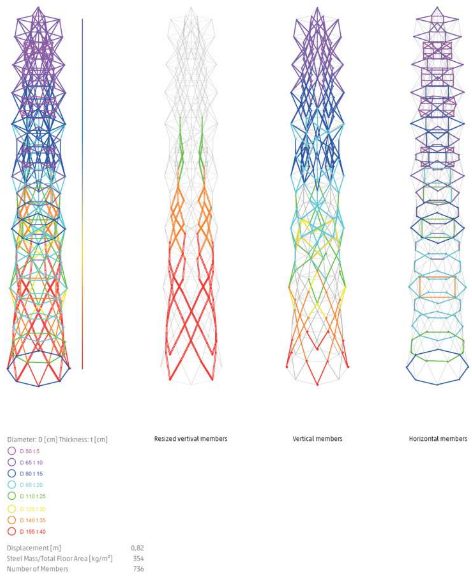


Figure 10: Final tower design

References

[1] Moon, K., Connor, J., and Fernandez, J., Diagrid Structural Systems for Tall Buildings: Characteristics and Methodology for Preliminary Design. *The Structural Design of Tall and Special Buildings*, 2007 16, 205–230.

[2] Otto, F., *Occupying and Connecting: Thoughts on Territories and Spheres of Influence with Particular*, 2009.

[3] Preisinger, C., Linking Structure and Parametric Geometry. *Archit Design*, 2013; **83**; 110-113. doi: 10.1002/ad.1564.