REDIRECTING THE GENERATIVE DESIGN PROCESS: DEVELOPING A COMPUTATIONAL TOOL TO EXPLORE MATERIAL OPTIMISATION FOR SPACE FRAME STRUCTURE

Comparative structural analysis driven by a generative design process

R. JOHANN,
University of New South Wales, Sydney, Australia
ryan_johann@hotmail.com

Abstract. Generative design (GD) is the process of defining high-level goals and constraints and then being able to use the power of computation to automatically explore a wide range of solutions that all meet the desired requirements. Due to the speed and iterative nature, it is proposed that GD can be a solution to fast-track the early-stage development process. The GD outcomes generated are analyzed simultaneously to expedite the design process for Architects and Engineers. Whilst previously material properties have being defined as a driving agent within the generative system, they currently fail to identify the material performance and structural capacity. This research paper proposes that a constrained approach, exploring traditional and non-traditional building materials can further validate the feasibility of a structure in the generative process. The tool will be developed within Grasshopper using C# programming, Karamaba3D (structural analysis plugin), Galapagos (evolutionary solver) and various engineering formulas (Utilization of Safety). The result of the research will be to create a generative script which prioritizes the structural characteristics of a material as one of the driving factors within the generative system. This in return will produce and analyze results that will aid engineers and architects in their early stage development process by using a generative design method.

Keywords. Generative design, material properties, utilization of safety, evolutionary algorithms

1. Introduction: Research Aims and Motivations

Generative design defines a process which incorporates a series of specific principles which range from genetic algorithms to topology optimization. The combination of these systems drives the process of the forms generation. In general, integrative design processes emphasize natural morphogenesis in which the pattern formation is contiguous to the process of materialization. (Baharlou. 2015). Much like the concept of morphogenesis, form can be defined as an interaction between internal components and external forces (Kwinter, 2008). Baharlou and Kwinter both describe unique components that would act as separate classes to be synthesized into the generative design tool. These classes then interact with each other within the virtual environment driven by their unique set of rules and algorithms as one complete system. Due to the nature of the way this system works, the results can be extremely superfluous, thus, the proper generative computational framework includes both mechanisms to generate possibilities and constraints to limit the range of possibilities (Holland, 2000). It is this approach that Holland describes in which this paper will be addressing.

In response to Hollands approach, this paper will take a constrained approach to the generative framework. This research aims to develop a material-based generative design system which can optimize for traditional and non-traditional based materials. By using an action research methodology, the system will be built iteratively, whereby each iteration will progressively build upon the previous one to reach the final outcome.

2. Research Observations and Objectives

The purpose of this research paper is to investigate the possibility of integrating material-based constraints as the driving factor within the generative framework. Different structural patterns will be explored using real-time computational form finding. The development of the generative system will be investigated through constraints generating procedures. This will aid in understanding the variety of constraints already explored as well as the possibility to simultaneously linking these constraints, allowing for the exploration in engineering solutions. Understanding the bottom up methodology of behavioral - based systems will be useful to understand and organize the complexity of the emergence and its process. Linking to the generative design approach, investigating the mechanical properties of a material and their formation within a structural pattern, allows the form to emerge based on this class of rules. The rules will be gathered by investigating data specific to the geometrical behavior of the desired material, as well as the structural analysis formulas used in the optimization process. The structural formation will use planar truss system as the case study.

3. Research Questions

How will the generative design process of a built form be challenged through its algorithmic pattern as well as its material properties?

Generative design follows rules and algorithms to build upon itself with limitations. The process of applying the sectional and material properties to the beginning of this process will challenge the design outcome but in return aims to produce more feasible result.

4. Methodology

The objective for this study was to explore the possibilities of using an GD system which would be driven by the material properties and physical forces being applied to them. An action research methodology was adopted for the processes “Implicit in the term action research is the idea that teachers will begin a cycle of posing questions, gathering data, reflection, and deciding on a course of action” (Ferrance E 2000). This approach allowed initial research to the direct generative solution, which subsequently could then be tested, analysed and improved upon.

This method maintains a generative computational framework to generate all the future possibilities, while maintaining specific constraints and/or limitations. The mechanisms of this generative system are bound to material properties, physical forces and construction constraints. The material properties have the ability to characterize geometrical behaviour mechanisms. In addition, motion behaviour mechanisms can perform as consequence for each parameter, where if the desired conditions are not being met, then the responsible mechanism will release an appropriate response to change the systems behavior.

Due to the nature of the research topic, it required a strong foundation of data and background research into the mechanical properties of the desired material being optimized. As there are an endless number of rules and algorithm’s that engineers use for material and force calculations, it was essential to choose the most influential one in relation to the chosen case study, as it would yield the most accurate results for real-life scenarios.

A planar space truss system was agreed upon to be the structural form used as the case study. C# was used to create this due to various advantages it had over Grasshopper scripting. A structural analysis plugin (Karamba3D) was also used to extract the desired forces needed for the analysis algorithms. The mechanical properties of Steel, Wood and Bamboo were used in the material properties component from Karamba, these were to be compared during optimization stage. A utilization of safety formula would then finally be used to help validate the generative design outcomes against all the internal properties. As this methodology is constantly tested, analyzed and validated, each iteration of the GD system could then progressively build upon the previous, applying more constraints and limitations to further validate the feasibility of the generated outcomes.

5. Background Research

Generative based design approaches the challenge of physical limitations within a virtual environment. In an architecture context, Terzidis states this distinction between computerization and computation. Whereby the translation from the computer processing of gathering data to conclude with a derived result is a static analog approach, implying a making of and from computing processes, that executes relational means which inductively and deductively inform specific solutions (Terzidis, 2003). However, the most basic understanding of a functioning system is one which operates through levels of development, feedback and redevelopment (Bertalanffy, 1969) in search for the perfect balance. Or even, the idea of a systems behavior-based approach where the form is realized in the process of interrelating material and spatial conditions (Weinstock, 2004). The early idea of a behavioral based generative system which works to highly complex rules is better known as an agent-based system, which sets a strong framework to this generative design workflow.

An agent-based system consists of large number of agents that follows simple local rules and interacts within an environment (Gilbert, 2008). The agent-based generative system helps inspire a material-based approach, whereby it draws from a bottom-up approach and takes advantage of using all the properties of a design to build it up, rather then having a design and breaking it down. In Architecture, creating a generative model ~~would be~~ associated ~~with different~~ methods to establish ~~a~~ unique complexity, however in engineering it would closely encompass with a more physical set of properties and rules. One of the features of such adaptation in complex system is emergent properties, which can be obtained through Constrained Generating Procedures (CGP's) (Holland, 2000).

CGP's systems prove to be a strong driver with agent-based systems, whereby each agent can be defined to the unique set of constraints (material properties) as desired into one bottom up approach system. As each agent has a unique set of data and rules, they are critical in defining the outcome which should be both generative and work within their specific constraints. This real-time interaction is relied on the agents' data structure; the agents perceive the environment within the coding environment as well as other agents, and based on their defined ontology compute the proper response to any stimuli (Pfeifer and Scheier, 2001).

In 2004 Martin Tamke, David Stasiuk, Mette Ramsgard Thomsen led a generative material-based concept of a growing architecture able to sense and dynamically adapt to its environment as it grows into form while continuously reacting to its own material performance and behavioral constraints. Even though it did prove the concept of a material-based generative system, its fabrication was limited to a top-down approach where by each strut was made of a tightly packed bundle of variable-sized rattan elements, clashing with the bottom – up approach for generative design. They even concluded that “There is significant space to better code local physical characteristics into the simulation system as a means to further refine the relationship between digital feedback and anticipated physical performance”.

In 2013 Ehsan Baharlou and Achim Menges investigated integrating material formation and construction constrains to plate-like structures; whereby, this mechanism was limited to the planes geometry. They even concluded that “it is also discernible that the lack of construction mechanisms (which naturally has been used in the plate structure), along with insufficient construction constraints caused the initial result to be far from what was expected.” (Baharlou, 2013) Given the evidence of the material influence of a generative form, it is extremely important to establish whether or not structurally an investigation like this could have resulted in a feasible outcome, ultimately resulting in a possible design to be fabricated whilst still holding true to its class principles.

In engineering projects, the early design stage is decisive in determining a building’s eventual performance (Kimpian, 2009). A structures performance is directly proportional to the fundamental principles of its physical properties. We make the mistake by basing design on geometrical manipulation only, rather than the physical environment we build for. The agent-based approach to these generative design process’ missed a fundamental structural agent which can be firmly reiterated by Pugnale et al. (2011) “The principle of reciprocity in structural design and construction refers to the use of load bearing elements to compose a spatial configuration in which they mutually support one another” (Pugnale, 2001). This has been known since the antiquity (Baverel, 2000).

Material properties such as stiffness and elasticity play an important role in the generative design process, this set of data is a driving factor to generate the emergence within the virtual environment. However, structurally the emerging phenomenon of the materials form must be constrained with a new agent which includes the mechanical properties of those materials working under examples of tension or compression. A set of parameters responding to dynamic, material and variable contextual forces over time (Kolarevic, 2003). A new agent which not only complies with material properties but also complies to the material’s structural capacity.

6. Case Study

The research project originally focused on creating a generative train station canopy as the case study ~~for the experiments,~~ however this was quickly changed to a truss system to cater for the accuracy of the structural analysis. In collaboration with Aurecon, an engineering company based in Australia and South Africa, the aim was to create a generative script for a structural system which could optimize for traditional and non-traditional materials. The generative framework was driven by four stages, the first being the component for the generative truss system, secondly was the structural analysis, thirdly was the optimization algorithm and the final was optimization process using an evolutionary solver.

6.1 GENERATIVE COMPONENT

A generative component needs to be highly diverse and parametric. This system must be able to produce a countless number of results. This way the computer can automatically change all the variables whilst simultaneously analyzing them.

6.1.1 C# Component

Creating a generative space truss system required a robust and complex structure to achieve a high level of parametricisim. A C# component within the grasshopper environment was chosen as the driving factor to create this truss system. There were three main reasons as to why C# was used; the first and most important was because by using a raw coding language such as C#, the ability to 'loop' within the same set of data and perform the same function multiple times was crucial to create the generative nature needed. Grasshopper could not have been used because it is a linear coding system which can only execute a function once. The second reason was for speed and efficiency. Grasshopper exponentially requires more computing power with each component added, which resultantly increases the time for each calculation to complete. C# programming counteracts this problem, thus allowing for a faster and more flexible system, whereby the efficiency of the computing power can be preserved as much as possible, saving it for the optimization stage. Finally, was because the analysis of trusses often assumes that loads are applied to the joints only and not at intermediate points along the members. This means every member of the truss is then subjected to pure compression or tension forces only. Whereby shear, bending moment and other more-complex forces are all practically zero. This ensured the analysis followed a more direct path and stayed within the scope.

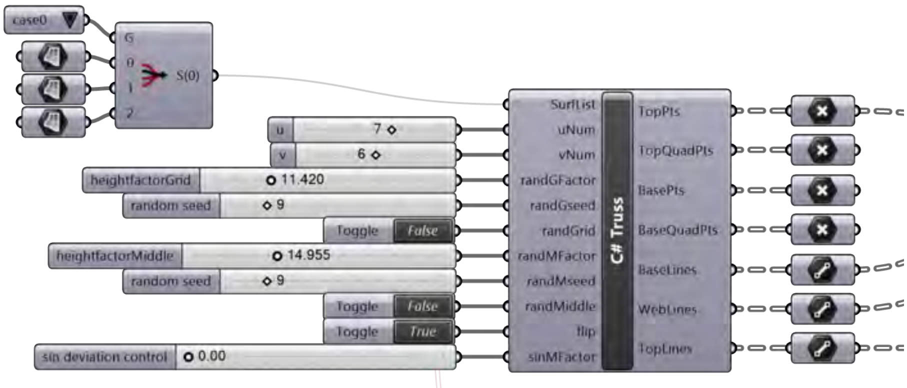


Figure 1. Planar Space Truss System C# Component

6.2 STRUCTURAL ANALYSIS

A grasshopper plugin called Karamba3D is a parametric structural engineering tool which provides accurate analysis of spatial trusses, frames and shells. It played a key role in acquiring the data needed to execute the structural calculations.

The Karamba analysis tool was able to output the results for all the various forces being applied to each beam, dependent on the types of loads applied to them.

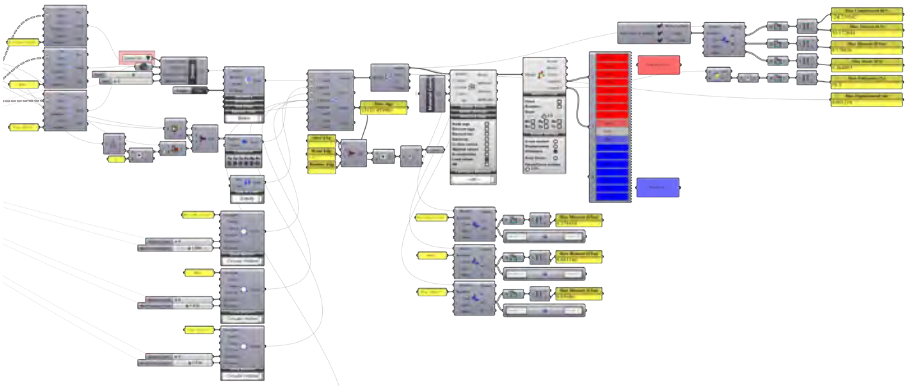


Figure 2. Karamba3D setup within Grasshopper

6.2.1 Normal force

The main forces extracted from the structural analysis plugin were the resultant normal forces. The normal forces are the component of the contact force that is perpendicular to the object and in this case is it the truss beams.

These forces were critical to complete the next step in developing the optimization formula.

6.3 OPTIMISATION FORMULA

Using the normal forces extracted from the Karamba analysis plugin, these forces were needed to develop a factory of safety (FoS) or utilization of safety (UoS) algorithm which will be the resulting number used by the evolutionary solver.

6.3.1 Utilization of Safety

In engineering a UoS is a number range which expresses how much stronger a structural system needs to be to withstand the load it will be undertaking.

The UoS is defined by the following formula:

Utilization of safety = Design Stress / Yield Stress

6.3.2 Design Stress

The design stress is what the object is required to be able to withstand. This is where the resultant normal forces from Karamba were used. The design stress is calculated in Grasshopper using following equation:

Design Stress = Applied Load / Area Cross Section

6.3.3 Yield Stress

Yield stress is the material property. It is the maximum stress up to which a body undergoes elastic deformation. The yielding point determines the limits of performance for mechanical components, since it represents the upper limit to forces that can be applied without permanent deformation.

This calculation was done within the grasshopper environment and used the yield strength for steel which was already in Karamba3d's data base.

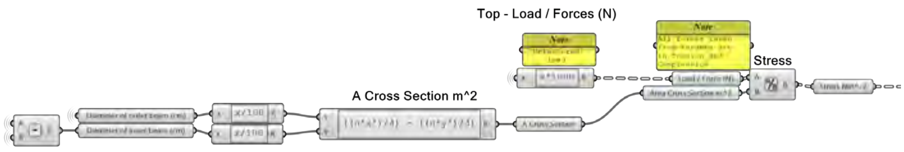


Figure 3. Yield Stress formula using the resultant normal forces

6.3.4 'If' Statement

To be able to start the optimization stage, the resulting utilization of safety result needed to be run through an 'If' statement. This was critical to correctly filter for the results that fell within and the results that fell outside of the UoS threshold.

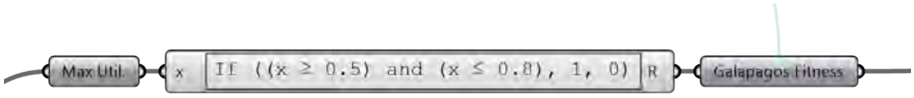


Figure 4. Original ‘If’ Statement rule

More specifically the way the statement works is that if the resulting UoS fell within the set range of 0.5 and 0.8, the resulting number will equal to 1. However, if the resulting number fell either above or below the range the resulting number would be a ‘false’ which in computer terms will be a 0. This way any results that resulted in a 1 could be categorized as ‘safe’ and any results that equaled to 0 would be considered ‘un-safe’.

6.4 EVOLUTIONARY ALGORITHM

The geometry for the truss system is iterated from the external optimization loop so that the optimum solution can be discovered. To achieve this looping an evolutionary/genetic algorithm was used. There are a range of evolutionary algorithms that are available to be used within Grasshopper, these include tools such as Galapagos, Goat and Octopus. Each solver has its pros and cons in terms of flexibility and adaptability for its intended use. A user could decide which algorithm is adequate for each optimization problem based on the number of objectives to be fulfilled or the necessity of finding a global or a local optimum.

6.4.1 Galapagos

In this research, the Galapagos evolutionary solver was considered adequate for the main optimization algorithm. The external input required from the solver is the genes that form the genome and the fitness value. The possible assigned genes are floating point numbers that can accept all the values between two numerical boundaries (Galapagos, 2015).

6.4.2 Solver Settings

In the Galapagos editor some further adjustments from the user is necessary before the optimization could commence. Firstly, whether the fitness value that is being optimized for should be a minimum or maximum. The number of outcomes that form a population as well as the multiplication factor for the first populations being optimized. Furthermore, the percentage of designs of a population that can be transferred in the next generation should be decided as well as the inbreeding percentage. The value of -100% is used for fully zoo-philic and the value of 100% for fully incestuous (Galapagos, 2015). The algorithm terminates when the maximum assigned duration defined from the user is reached or when the number of consecutive generations without finding an improved solution exceeds the maximum stagnant limit.

Through testing of various optimization of the planar truss system, some of the solver settings were changed to ensure a more reliable workflow. The multiplication factor had to be increased from 2x to 3x, this ensured Galagoes was exploring a wide-solution space to begin with. Also, the original inbreeding variable of +75% was changed to +25% to acquire a more zoo-philic approach, which ensured a greater deviation each time galagoes jumped to create a new population of results.

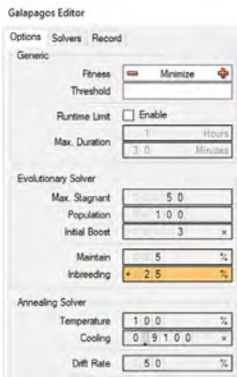


Figure 5. Galapagos Solver Settings

6.4.3 Penalties

The original 'If' statement which filtered result that fell within or fell outside the UoS threshold was changed to be able to optimize for a minimum mass. A penalty value had to be present in the output of the grasshopper 'if' component, since these values will be used as a fitness for Galapagos. The first change was instead of the 'if' statement returning a 1 if the result fell within the UoS threshold, it would now return the resulting mass. However, if the result fell outside the range, they could no longer result in a 0, because Galagoes would consider all those results to be most optimal as the mass it is solving for is 0. Therefore, for all the structures that fell outside of the UoS threshold, a mass penalty was implemented to ensure they were always exponentially greater than the correct results that fell within the threshold. This ensured Galagoes was always optimizing for the correct results. The penalty factor within the 'if' statement can be seen below within the Grasshopper environment.

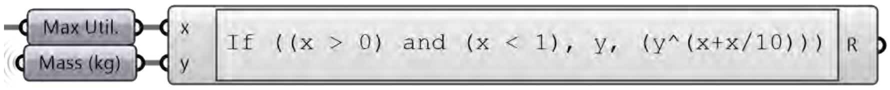


Figure 6. 'If' Statement with Penalty Factor

6.5 GENERATIVE FRAMEWORK

The GD design framework has produced promising yet conceptual results. The ability to be able to optimise both traditional and non-traditional materials such as steel, wood and bamboo has shown that the resulting planar space truss system can optimise to adhere between the same utilisation of safety threshold regardless of the material chosen. This has resulted in bamboo structures that can be far more optimal in terms of their total mass and cost compared to steel. (Refer to Appendix A for results)

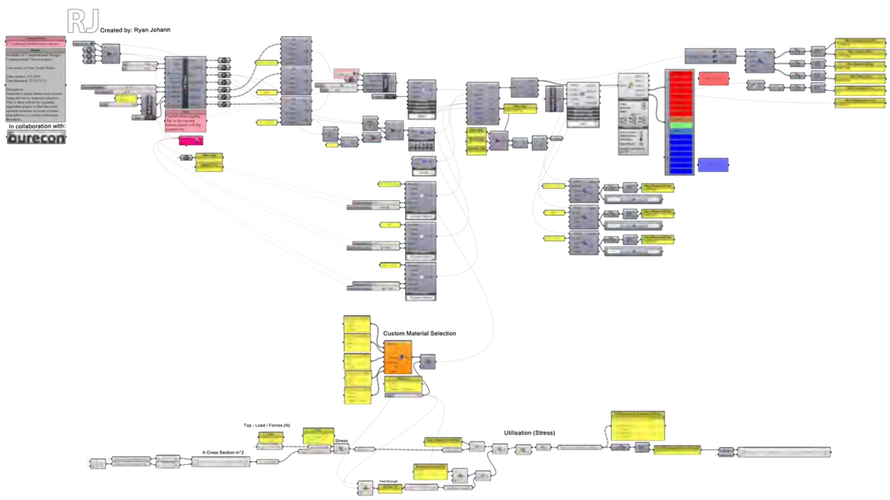


Figure 7. Complete generative framework (see Appendix C for higher resolution)

7. Significance of Research

The intention of my research paper was to explore new possibilities to design that were not previously available, and which might have offered new ways to overcome current limitations of existing design methods. The argument imposed by the generative design process is that neither the current post-rationalized nor the pre-rationalized approach to a project aims to resolve the optimal solution in such an early stage of the development.

For a post-rationalized method, various changes that might have been made to the overall configuration could result in a form/geometry that is no longer suitable for the job, whilst for a pre-rationalized method, the design process may feel limited/constrained to rules within the system. By meeting in the middle of the two rationalized methods, this research has added a new generative design process which can aid engineers to fast track the early stage development process for a traditional and non-traditional material-based structure.

Also, it allows engineers and architects to explore a wide solution space within minutes, so when they want to go into the design phase, they already know which solutions are optimal. This would help eliminate the need to manually tweak a solution until it became optimal.

Finally, it allows for the inclusion of configuring and exploring the consequences to build upon each other simultaneously, a new design approach aimed to be at the forefront within the built environment.

8. Evaluation of Research Project

This research paper has presented a material-driven generative solver as an extensive design process, which can produce robust solutions that adhere to a utilization of safety. By comparing traditional and non-traditional structural materials, this system has distinguished the use of a hypothetical framework for generating iterations that may 'fast-track', and/or inspire the early stage development of a structural system without necessarily conflicting with desired design directions, all while driven by a material.

Unlike previous research papers which lack the exploration of material emphasis in a generative design method, the results provided agreed with outcomes that would have been expected if they were developed and analyzed using traditional methods. The expectation of a bamboo structure to withstand the same loads of a steel system while still adhering to the same utilization of safety only becomes more apparent with the use of the material-driven generative method.

However, even though the importance of early stage performance-driven evaluations is becoming more valued as the power of generative computation grows, this method and its outcomes still encounter much resistance.

The proposed generative process in turn will require a substantial investment of time and collaboration between different disciplines and will inherently need to redesign the way in which the design process is perceived. This would regard engineers, architects, computational designers and material scientists to imply a new framework in which each aspect of their contribution would drive the other, and without a digital collaborative platform to accommodate for this workflow, the proposed generative design method, will always remain conceptual.

Secondly, in the current alpha stage of development, the outcomes cannot be deemed as feasible as proposed due to its limited number of structural analysis formulas integrated. A utilization of safety is only one generalized formula to deem how suitable the resulting structure could be in terms of how safe it is. The results would still need to be furthered analyzed in a reputable structural analysis software such as SAP2000 & ETABS.

This is not to say a generative design method could not withhold a quality assurance to a professional standard, because due to the framework's nature, new design conditions and materials can be easily integrated to further refine the performance, allowing an exploration of structural possibilities which would have otherwise been ignored.

9. Conclusion

In this paper we have presented a generative framework which can optimize traditional and non-traditional materials by applying rules and algorithms as a set of constraints resulting in novel outcomes. By providing two different test examples, we have produced promising yet conceptual results for the purpose of creating structural geometry that could 'fast-track' and be a 'source of inspiration' during the design phase according to real world configurations and materials.

The example of the planar space truss system being optimized by different materials addresses the capabilities of exploring new design options through a generative approach, veering from traditional analysis tools. The ability to simultaneously evaluate options creates a bidirectional relationship between scripting potential, parametric environments and traditional analyses for the development stage. Therefore, the design development of a structural system should not be limited to traditional building materials only because as examined through this research, the same design requirements can be achieved through non-traditional materials.

Given the flexibility and principle of the generative framework, the overarching approach of this constrained system should be further considered within the greater 'generative' and 'analysis' ecosystem. By starting with a material-based approach, the framework is already in place for migrating design concepts into design developments and analysis. Collaboration between industries needs to be promoted to further develop a generative design workflow, whereby the system can then withhold a professional quality assurance approved by each profession.

Finally, we strongly feel that by taking a more constrained approach to the generative design framework, feasible results can be achieved.

Advancing the future of computer aided design for both engineers and architects.

Acknowledgements

This research was supported by the faculty of the Built Environment at UNSW and Aurecon. In particular I would like to thank my tutors Alessandra Fabbri, Nicole Gardiner and M. Hank Hauesler. As well as my supervisors from Aurecon, Michael Chernyavsky and Michael Broderick for all their dedicative assistance and feedback.

References

Schoina, S.: 2016. “Performance-based form-finding and material distribution of free form roof structures: Implementation in the Post Rotterdam Case Study”. Available from: repository.tudelft <https://repository.tudelft.nl/islandora/object/uuid%3A69c21b76-847d-4eca-9732-99799361299a> (accessed

Baharlou, E., Menges, A.: 2013. “Rethinking Prototyping, Proceedings of the Design Modelling Symposium”. Available from: Google books < https://books.google.com.au/books/about/Rethinking_Prototyping.html?id=Da0TAQAAQBAJ > (accessed

Baharlou, E., Menges, A.: 2008. “Generative Agent-Based Design Computation”. Available from: cumincad <https://cumincad.architexturez.net/system/files/pdf/ecaade2013_081.content.pdf> (accessed

Bertalanffy, L.: 1968. “The meaning of general system theory”. Available from: semanticschholar <https://pdfs.semanticscholar.org/75b8/dce92efbc16498bb435455189c941c041c7a.pdf> (accessed

Holland, JH.: 2000. “Emergence: From chaos to order, Perseus Books, Cambridge, Mass”. Available from: web.mit.edu <http://web.mit.edu/esd.83/www/notebook/EmergenceReview.PDF> (accessed

Menges, A.: 2008. “Integral Formation and Materialization: Computational Form and Material Gestalt”. Available from: Research Gate <https://www.researchgate.net/publication/285886477_Integral_formation_and_materialisation_Computational_form_and_material_gestalt> (accessed

Pantazis, E., Gerber, D.: 2014. “Material Swarm Articulations New View Reciprocal Frame Canopy”. Available from: Research Gate <https://www.researchgate.net/publication/267695406_Material_Swarm_Articulations_New_View_Reciprocal_Frame_Canopy> (accessed

Schlueter, A. and Bonwetsch, T.: 2008. “Design Rationalization of irregular Cellular Structures”. Available from: Sage journals <https://journals.sagepub.com/doi/10.1260/147807708785850104> (accessed

Terzidis, K. 2003. “Algorithmic form. Expressive Form: A Conceptual Approach to Computational Design”. Available from: < https://epdf.tips/expressive-form-a-conceptual-approach-to-computational-design.html> (accessed

Weinstock, M. 2004. “Morphogenesis and the mathematics of emergence”. Available from: EPDF <https://www.researchgate.net/publication/233868005_Morphogenesis_and_the_Mathematics_of_Emergence> (accessed

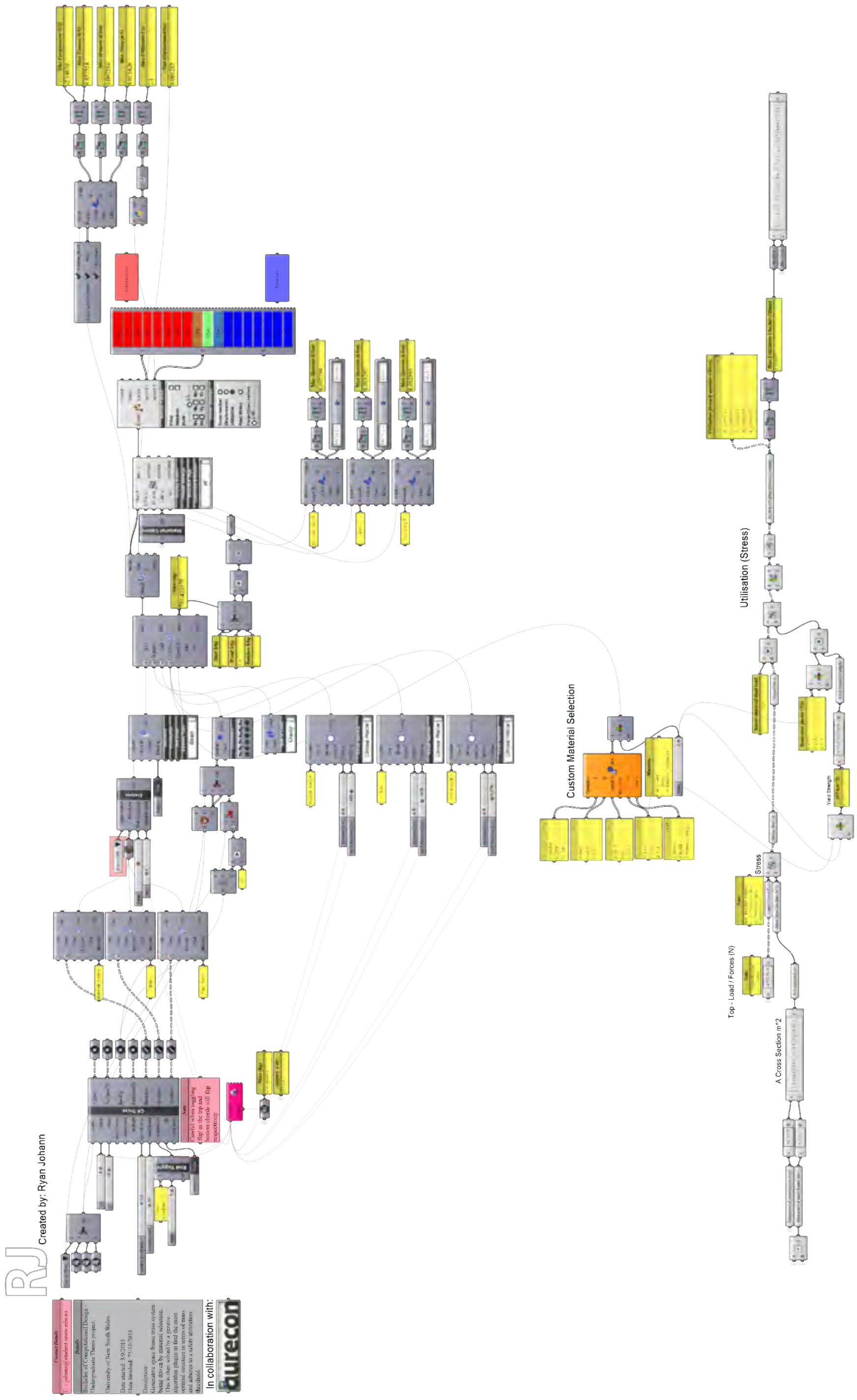
Appendix A: Test Results

		Steel		Wood		Bamboo	
		Traditional	Random Dev. & delete	Traditional	Random Dev. & delete	Traditional	Random Dev. & delete
Flat Surface	Mass (kg)	1452.52	951.85	123.74	78.56	133.21	67.52
	Cost (Aud)	7263	4759	309	196	266	135
	Height (m)	0.5	0.5	0.5	0.5	0.5	0.5
	Total num. beams	32	18	32	23	32	19
	Max Utiliszation	0.001679	0.019856	0.003857	0.024985	0.000674	0.009917
	Bottom chord diameter (cm)	5	5	5	5	5	5
	Bottom chord wall thickness (cm)	1	1.453	1.483	1.65	1.559	1.241
	Web diameter (cm)	5	5	5	5	5	5
	Web wall thickness (cm)	1.65	1.957	1.528	1	1.47	1.137
	Top chord diameter (cm)	5	5	5	5	5	5
	Top chord wall thickness (cm)	1.34	1.706	1.715	1.776	1.579	1.478
Gable Roof	Mass (kg)	1624	966.47	108.25	83.666	104.69	62.56
	Cost (Aud)	8120	4832	271	209	209	125
	Height (m)	0.5	0.5	0.5	0.5	0.5	0.5
	Beam Total	32	19	32	21	32	19
	Max Utiliszation	0.001685	0.019931	0.004381	0.0474	0.000926	0.002694
	Bottom chord diameter (cm)	5	5	5	5	5	5
	Bottom chord wall thickness (cm)	1.593	1.1614	1.368	1.767	1	1.501
	Web diameter (cm)	5	5	5	5	5	5
	Web wall thickness (cm)	1.458	1.432	1	1.471	1	1.708
	Top chord diameter (cm)	5	5	5	5	5	5
	Top chord wall thickness (cm)	1.379	1.672	1.442	1.621	1.276	1.738

Appendix B: Optimization video

https://youtu.be/8pvvSxAX8NQ

Appendix C: Complete Script in Grasshopper



Appendix D: Various Optimization Results for Planar Space Truss Systems