

Use of Genetic Algorithms to Optimize Efficiency in Small-Scale Structures

Aaron Chew
 Valencia High School
 Yorba Linda, California
 aaronchewbani@gmail.com
 March 11, 2020

Abstract—The prevalence of algorithms and computational tools in the modern-day has intersected with nearly every field. Generative design, specifically those using genetic algorithms, is an increasingly effective, yet cost efficient way to generate architectural designs in modern engineering. Thus, we adopt a genetic algorithm model in pursuit of maximizing the durability of a structure when it is stressed while minimizing the material cost. After the model is formulated, the algorithm is able to approximate with high accuracy the load a small-scale structure is able to bear, as well as iterate upon its designs to maximize a fitness function.

I. INTRODUCTION

In recent years, rising costs of management and safety guidelines have placed upward pressure on prices of architecture design and have posed new challenges to architects in developing blueprints in a safe and cost-effective manner. We aim to assist the process of design, maximizing safety, and minimizing material and management costs of framing buildings throughout architecture.

Generative design, a process of generating solutions to design problems with specific constraints, is becoming an increasingly used method in architecture. New methods of architecture implement generative design to harness the power of artificial intelligence and advanced computing algorithms to automatically construct designs that satisfy specific criteria. Recently, Volkswagen incorporated generative design in the backfitting of the 1962 microbus. In doing so, Volkswagen's design team lowered the weight of the wheels by 18%, highlighting the efficacy of optimization that generative design provides. Even more impressive was the acceleration of development, reducing times from the typical 1.5 years to a few months [1]. Furthermore, generative design has already been implemented in high energy usage buildings as a means to increase energy efficiency. [4] Layout of energy usage systems, including lighting or air conditioning, can be optimized with generative designs and thus reduce net energy usage. The increasing prevalence of artificial intelligence in technological processes offers the prospect of further improving upon generative design, generating the potential to

revolutionize the design and manufacturing of products and the focus of this paper: structures.

In this work, we explore generative design as a method to produce highly optimized small-scale structures, specifically, frames of buildings. Utilizing mechanics formulas modeling how structures perform under stress tests, we implement generative design to efficiently select new and modify old structures. Specifically, we try to maximize the ratio of weight held via vertical compression to the cumulative weight of building frame material. By comparing the blueprints of theoretically optimized structures, we execute small-scale experiments implementing virtual structures chosen by the algorithm to test the method's efficiency and practicality in the real world.

II. DATA

A. Constraints

For technical constraints, the base of the structure must span a minimum of a 20cm by 20cm square. Using the Pythagorean theorem, we can easily minimize the base of the tower to a $\sqrt{(10^2 + 10^2)} \approx 15$ cm base. A 15 cm base rather than a $\sqrt{200}$ cm base is used to account for margin of error. The structure must be constructed with a minimum of 50cm in vertical height, with a 5cm by 5cm gap at the top.

The testing site will be comprised of a block that is laid upon the top of the structure with a chain spanning the interior of the structure. At the bottom of the chain will be attached a bucket, which will slowly be loaded with sand until the tower breaks due to vertical compression. The intent is to maximize the load beared by the tower.

Thus, our design will be comprised of four legs oriented 15 cm apart at the base, and 5 cm apart at the top, with the vertical height spanning 50 cm. The length of each leg is thus approximately 53cm.

B. Data Source

In this study, we utilize a dataset [3] of pretested buckling strength for various densities of balsa wood. A column buckling test machine was used to determine the buckling strength of 1/8'' by 1/8'' by 32'' columns.

We utilize these columns as our four main structural legs. The data here helps us in our simulation to determine the buckling point of each leg, given the densities. We can use the densities of each leg as a parameter of optimization in our algorithm. Note that the length of each leg will be cut to approximately 53cm to fit the constraints of the problem. Furthermore, smaller pieces of bracing material, namely 3/32", 1/16", 1/8", and 1/32" are used as braces. The orientation and density of these braces are additionally used as parameters in our training algorithm. Data for buckling strength of bass and balsa wood sticks can be requested via email.

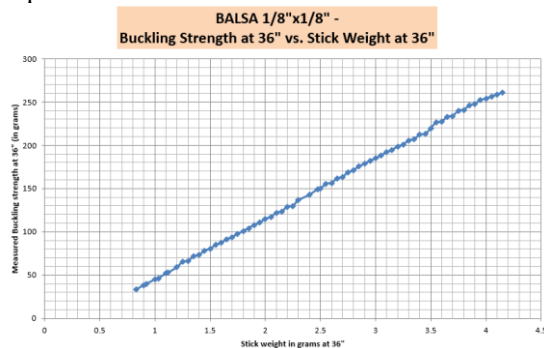


Figure 1. Buckling capacity of a 1/8" by 1/8" by 36" balsa sticks at varying weight in grams. Weight is proportional with density.

III. USE OF GENERATIVE DESIGN

A. Variables in Consideration

Generally, frames of buildings or structures have main vertical axially loaded columns (known simplistically as "legs") that are the essential backbone and bear most of the weight on the building.

Additionally, braces are used in the frames of buildings to sturdy the main legs. Braces come in many different shapes and formations; perhaps the most well-known is the "X", or cross brace. Other useful braces used prominently in frame building include diagonals and horizontal ladders. The use of braces provides a low-cost solution to prevent local buckling upon main legs.

Intuitively, stronger, denser materials hold more weight than weaker and less dense materials. Therefore, it is a challenge design for architects and constructors to select a myriad of variables:

- Material of braces
- Material of legs
- Density of braces
- Density of legs
- Type of bracing (ladders, crosses, diagonals)
- Placement of bracing

For simplicity, these are only variables of structures in the most condensed form. Therefore, the generative design algorithm attempts to identify the optimal combination of the variables mentioned above.

B. Simulating the collapse of a Structure

To solve the problem of maximizing ratio of weight held/weight of the structure, we first attempt to simulate the weight that a structure can hold given its blueprint. This is primarily done through Euler's critical load as given by the equation:

$$F = \frac{\pi^2 EI}{(KL)^2}$$

Briefly, F is the force at which a column buckles. E is the 'modulus of elasticity', a measurement based on the strength of the material the structures uses. I is the 'cross-sectional moment of inertia', simply the width and length of a leg. KL is the 'effective length'; for our purposes, K is constant. Once the force on the columns or bracing exceeds F, columns will buckle, and shortly after it will lead to collapse. For sake of simplicity, the point at which the column buckles will be the point at which the tower collapses.

Bracing effectively reduces the buckling threshold of a column under vertical compression exponentially. If a column were to be braced at its midway point, its critical load would quadruple, due to L halving. This is easily scalable, as a column divided into 10 sections using bracing would lead to a critical load 100 times less. The point of collapse will be the minimum force applied to break either the bracing or the legs. Because the legs of the tower are at a lean, the amount of force required to buckle the vertical columns is less. Computation of the critical on the legs is simply the *largest subsection height * sin(lean angle)*. Similarly, the bottleneck for bracing durability is defined $\min(\{all\ bracing\ length * \sin(bracing - leg\ angle)\})$. The minimum bottleneck of both above values will be the maximum weight the structure can support.

Using data [3] and the theorem mentioned above, we can effectively and accurately simulate the breaking of a balsa wood structure. We can easily modify E and I provided the modulus of elasticity and cross section size, respectively, for any other type of substance.

C. Formulating into a Computational Problem

Testing of every possible framing combination for a structure is infinite and thus infeasible. Therefore, we propose the use of a genetic algorithm to reduce the search space of optimal framing designs. To implement this, we define the fitness score of a design by $\frac{TotalWeightHeld}{WeightOfStructure}$. At first, genes are randomly selected in accordance with the variables mentioned in section A. With the "blueprint" of each structure generated and the

collapse simulation mentioned in section B, we receive the maximum amount of weight that a structure can hold. Through the fitness function, we can determine the score of this tower. Its design and score are added to the population of already existing designs. Then the genetic algorithm uses this to select, crossover, and mutate genes. Designs with a smaller fitness value will not be selected to reproduce, while those with high fitness values will reproduce more often. These genes then translate back into the variables mentioned in section A and a new, more optimal tower is created. In selection, the best characteristics of the previous generations are retained. In crossover, the genetic algorithm attempts to mix multiple already strong genes. Mutation is the occasional randomization of genes, as searching using only selection and crossbreeding can lead to local minima. This process, inspired by evolutionary biology, is repeated millions of times. Eventually, the program arrives at a design solution or blueprint with the least amount of materials used to hold a threshold of weight.

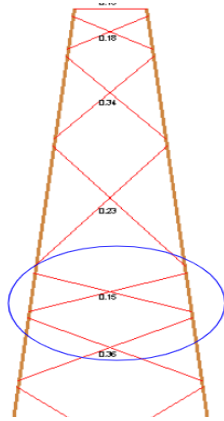


Figure 2: Uneven and unoptimized bracing patterns in earlier generations

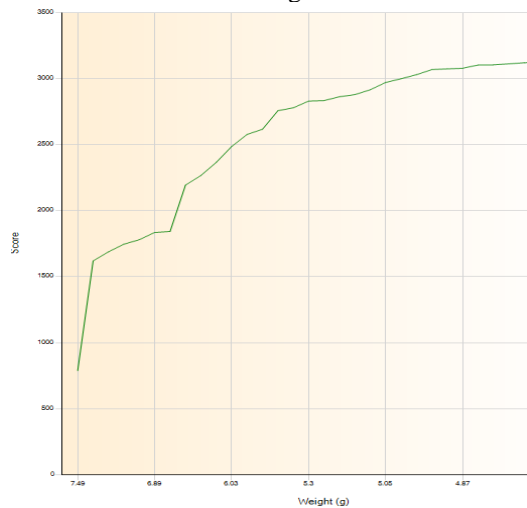


Figure 3: Score of structures through 1.4 million generations of evolution. The strongest towers are selected through mutation at the start but eventually rely on crossbreeding.

D. Particle Swarm

To assist in the optimization of blueprints, we also utilize particle swarm optimization (PSO). Particle swarm utilizes a population of candidate solutions, shown below as dots, that move systemically around the search space. When an optimal solution is encountered, PSO reproduces more generally near the search space of the optimal solution. Furthermore, PSO occasionally assigns candidates to random solutions across the search space to reduce the possibility of convergence on a local optimum.

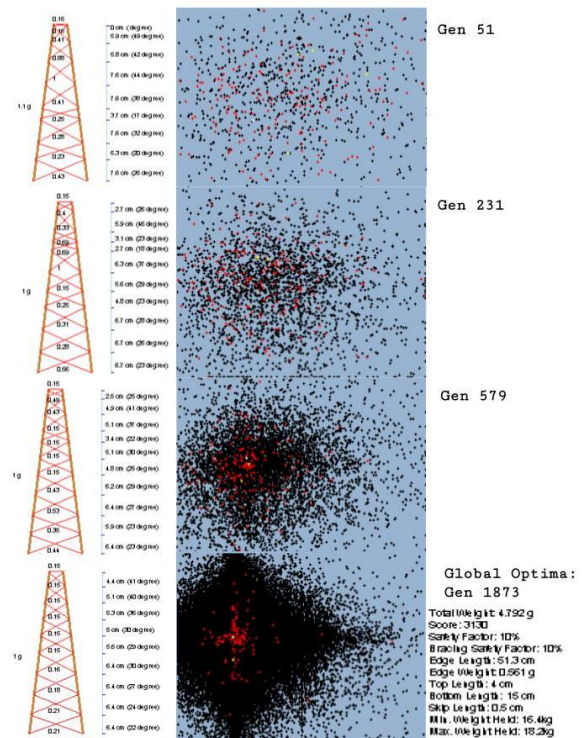


Figure 4: Particle swarm where each black dot represents a previously generated tower. Red dots represent the current population. The yellow dot is the current, most optimal tower. Observe that the optimal solution lies within the leftward region, where many of the current candidates (red) are swarming.

E. Results

Our final results yield the lightest structure to hold a certain amount of weight. To verify accuracy of the program in practice. Because of potential error in building, the results yielded in practicality are slightly below the maximum theoretical amount generated by the program. Expanding the search space as well as incorporating additional variables in the fitness function could yield a more efficient tower.

Algorithm Results

Weight Held (Kg)	5.0	10.0	15.0	20.0	25.0	30.0
Tower Weight (g)	3.30	4.62	4.98	5.19	5.43	5.57
Score	1.51	2.16	3.01	3.85	4.60	5.39

Tested Results

Weight Held (Kg)	5.1	9.8	15.1	19.9	24.9	30.2
Tower Weight (g)	3.46	4.67	4.94	5.12	5.40	5.51
Score	1.47	2.10	3.06	3.89	4.61	5.48

Figure 5: Data table of lightest structures at given weights needing to be held

Generally, all optimal blueprints generated were symmetrical with consistent pattern bracing. Towards the top of the tower, gaps between each cross brace increased as the width between the two legs decreased. Additionally, ladders were rarely used as a result of the “twisting” of an axial column. Although twisting was greatly minimized when paired with cross bracing, the use of a ladder was inefficient as it contributed weight to the overall tower weight. Overwhelmingly, 1/16” braces were chosen as opposed to 3/32”, 1/8”, and 1/32”. The density of the braces generally remained low, and the density of legs was higher. This is likely because the buckling load on braces exponentially decreases with its reduced length and cross pattern.

The genetic algorithm was also able to show that some counter-intuitive features of structure building can actually be most efficient. One prominent example of this is the implementation of gaps between braces. These small gaps, illustrated in Figure 6, have insignificant repercussions on the overall buckling strength of the tower but can reduce the amount of material used slightly.

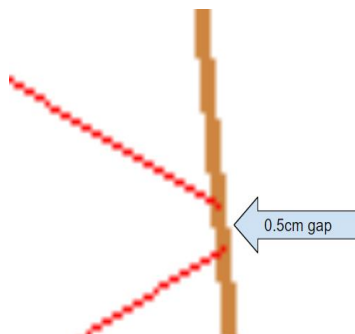


Figure 6: A gap is used between bracing as a method of optimization. The software has discovered the 0.5cm gap can be unbraced, as buckling will occur minimally in the area.

One other interesting feature that the program was able to produce was the layout of the tower in general. The right side of Figure 7 pictured below was the most efficient solution generated by my program. Some non-trivial logic reveals the reason for such a design. The high

vertical angle at which braces are jointed has significant ramifications on the buckling threshold. Therefore, by splitting X braces into diagonals, we can reduce buckling while keeping the cost of material constant.

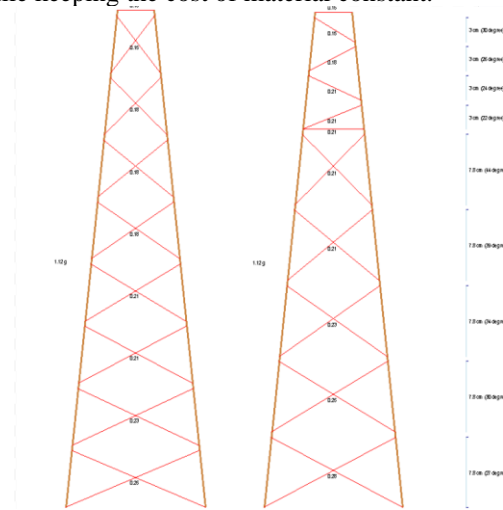


Figure 7: Left: Normally X-Braced Tower, Right: Optimized X-Braced Tower

IV. CONCLUSION AND FURTHER RESEARCH

In this experiment, we explore the viability of generative design in the creation of structures to carry critical loads. Importantly, this paper investigates the extent to which genetic algorithms can be used to minimize costs of framing structures with similar safety indices. Using Euler's Buckling Theorem, we can approximate the strength of different structures. Optimizing framing techniques can be quite cost efficient as it accounts for the second highest building cost after interior design [2]. Several architectural challenges exist today, such as minimizing the distance between a room to a fire exit in a hotel or maximizing a building's resistance to natural disasters such as earthquakes. Generative design can offer a holistic solution to these safety considerations, as shown. Future research options would include developing more accurate methods of evaluating a structural design, incorporating elements of the mentioned safety parameters, and testing other configurations of the apparatus. In particular, future works can explore testing generative structures in relation to a plethora of different materials - this current experiment assumes a static material of wood for experimentation due to the lack of a variety of testing data. By adjusting for different material combinations, the experiment can offer even more practical analysis on the application of generative design to architecture.

REFERENCES

- [1] R. Spiegel “The VW Microbus Gets Revamped As an EV with Generative Designed Parts”, designnews.com, July 31, 2019
- [2] Shaiful Amri Mansur, Rahim, A., Yusof, N. A., & Penerbit Akademia Baru. (2016). *Rising Trend in Construction Cost and*

Housing Price. ResearchGate; unknown.
https://www.researchgate.net/publication/330672601_Rising_Trend_in_Construction_Cost_and_Housing_Price

- [3] L. Joeris, "Balsa Buckling Strength Analysis," unpublished.
- [4] Touloupaki, E., & Theodosiou, T. (2017). Energy Performance Optimization as a Generative Design Tool for Nearly Zero Energy Buildings. *Procedia Engineering*, 180, 1178–1185. <https://doi.org/10.1016/j.proeng.2017.04.278>
- [5] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on

Neural Networks. IV. 1948. doi:10.1109/ICNN.1995.488968

pp. 1942–

ACKNOWLEDGMENTS

I would like to thank the late Len Joeris for providing the data used in this study. I would also like to thank Aqil Naeem in assisting with the editing of this paper.