Proceedings of the IASS-SLTE 2014 Symposium
"Shells, Membranes and Spatial Structures: Footprints"
15 to 19 September 2014, Brasilia, Brazil
Reyolando M.L.R.F. BRASIL and Ruy M.O. PAULETTI (eds.)

Optimizing a Trussed Frame Subjected to Wind Using Rhino, Grasshopper, Karamba and Galapagos

Evan J. GERBO*, Edmond P. SALIKLIS[a]

*Former Graduate Student California Polytechnic State University
Current PhD Student University of Notre Dame
Dept of Civil Engr. Notre Dame, IN 46556
EGerbo@gmail.com

[a] Associate Professor
California Polytechnic State University
Dept. of Architectural Engineering

Abstract

This paper presents an approach to optimization of efficiency in a building trussed frame. The process of optimization was the repeated reconstruction of models in an attempt to create a structure that has minimal weight in its columns and in its diagonal braces, coupled with minimal lateral deflection at the roof level, for a given applied lateral wind load on one side of the trussed frame.

The optimization was handled entirely in the virtual environment provided by Rhino, Grasshopper, Karamba and the evolutionary algorithm Galapagos. This parametric model created a loop that automatically redesigned columns, and braces as well as the width of the bay at each story level. This process is completed with an Evolutionary Solver that uses genetic fitness to eliminate unwanted characteristics (here larger weight and larger top deflection) and to select genes (here member sizes and bay width per story) that evolved towards genetic success (here minimum weight as well as minimum lateral roof movement).

A unique feature of this optimization is the efficient reduction of variables that were genetically modified. This reduction greatly enhanced the speed of the process and made the optimization so efficient that 40 or 50 generations of modifications can occur in a few minutes on a laptop.

Several braced frames were used in the parametric model, these were subjected to ASCE 7 Wind Loads and the results of the structural analysis performed by Karamba on the optimized structures were compared to results from a commercially available traditional structural analysis package. The comparisons were excellent, giving us further confidence into delving deeper into the physics engine of Karamba, apart from traditional finite element programs.

Keywords: optimization, genetic algorithm, structural analysis, form finding

1. Introduction

The term "paradigm shift" is sometimes loosely used today to describe phenomena that are new, but do not fundamentally alter the way people work and do research. Yet *paradigm shift* may be an appropriate image for the new world of parametric modeling in architectural engineering and architecture. The shift that is occurring is due to the vast power unleashed by combining 3D rendering programs such as Rhinoceros along with physics engines such as Karamba and now the genetic algorithm engine Galapagos. The relationship between the various software utilized in this study is shown in Figure 2. Within this virtual environment, and separate from traditional structural analysis programs of the past few decades, engineers and architects can optimize structural form for a given set of criteria. Whether or not this is truly a paradigm shift is certainly debatable, but Pugnale (Pugnale [2]) articulates and summarizes some current thinking on this topic by differentiating between technologies that are used to ease the burden of complex tasks and technologies that fundamentally alter the way

we design. It is reasonable to conclude that parametric modeling via genetic algorithms falls into this second category. It potentially leads to a new way of thinking about form finding.

The definition for genetic algorithms provided by Koza (Koza [1]) is pertinent to this paper. Koza states that a genetic algorithm is a series of mathematical operations that transform individual objects of a given population into a subsequent new population, by selecting a certain percentage of objects according to a fitness criteria. The robustness of a genetic algorithm is defined as the balance or interplay between the algorithm's efficiency and its effectiveness in adaptive, changing environments. The built-in genetic algorithm in Galapagos allows for some user defined control, but in general it is closed to the user and it can be qualified as being robust because it is very fast and it can be applied to a myriad of problems, not only architectural engineering. Ohmori has stated (Ohmori [2]) that genetic algorithms are one of the most effective optimization methods to handle changes of discrete variables. Consequently, designers can handle standardized structural elements from which they can search for the optimal combination for a given set of constraints. Ohmori also points out that since genetic algorithms provide a variety of optimum solutions rather than a single solution, such a technique lends itself naturally to the world of structural design, with its infinite variety of viable solutions.

Pugnale (Pugnale [3]) has produced a helpful flowchart to explain the overall process of a conventional genetic algorithm. A modified version of Pugnale's chart is shown in Figure 1 and it applies to the algorithm within Galapagos.

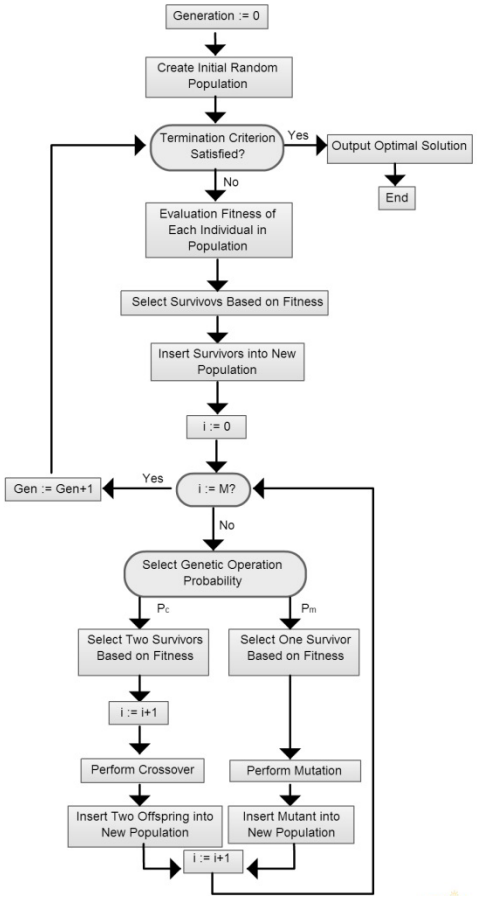

Figure 1. Genetic Algorithm Flowchart

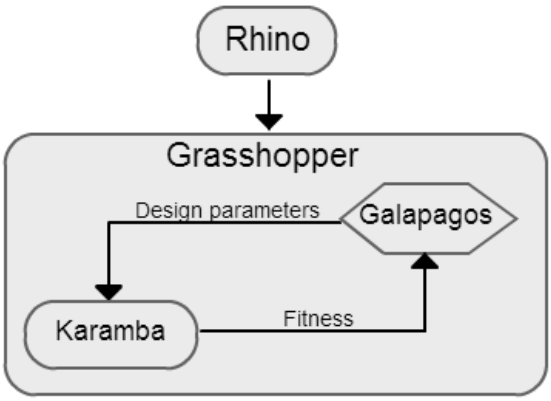

Figure 2. Software Interaction

2. Details of the Optimization

The structure looks like a standard braced frame, but in order to simplify our initial efforts, we chose to model the entire structure with pin-ended elements, consequently it looks like a frame but behaves like a truss, thus the name *trussed frame*. This trussed frame is very similar to a truss, with the difference being that the trussed frame is used as the lateral system for a building, versus a truss that is normally used to support vertical loads. Originally we intended to have the genetic algorithm select and modify the width of each bay as well as the size of each column and the size of each bay at a given story, in other words to have three genes to modify. The beam sizes were held constant in this optimization. The width of the ground floor was also held constant to better reflect the realities of grounding a building at a fixed site. The height of the building was held at a constant ten stories. Soon we realized that even for these three independent variables (size of columns, size of braces, width of bays), the genetic algorithm struggled with the wide variety of combinations possible as the structure approached ten stories. This approach required 30 (3genes * 10 stories) variables to define a unique structure. Consequently, a creative solution was found to streamline the selection process. This will be described shortly.

The Grasshopper program starts with standard input as shown in Figures 3, note that we show the base width as a slider input, but in this problem we kept it fixed width to make the problem realistically grounded.

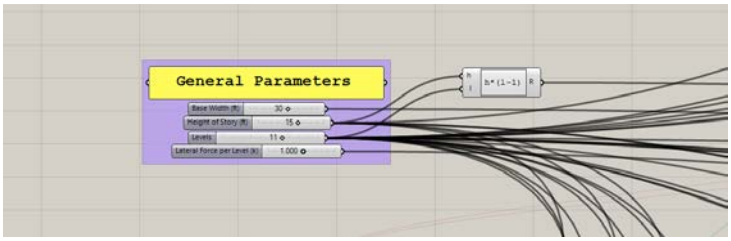

Figure 3. Initial steps of program

Figure 4 shows the reduction of computational complexity we invoked through a clever shift of gene parameters to undergo evolution.

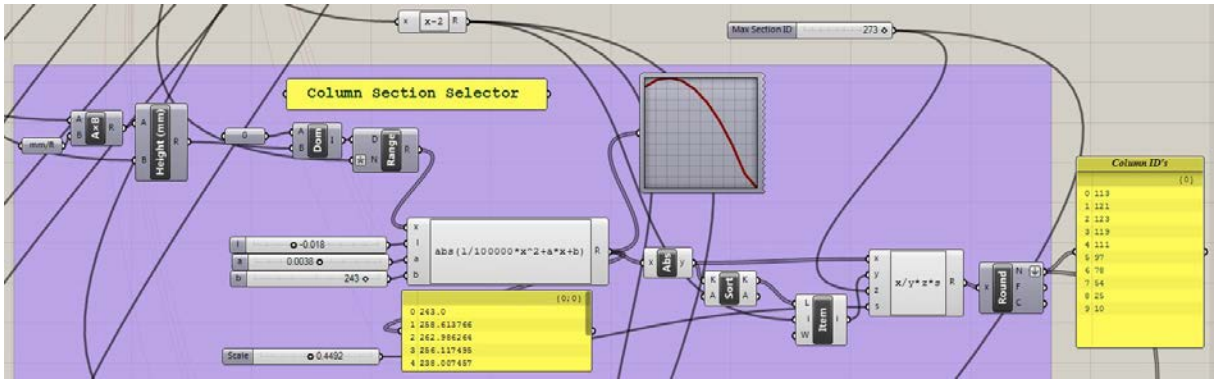

Figure 4. Switching control of particular genes to a polynomial function

The heart of Figure 4 is the polynomial as a function of *x*, where *x* is the height at each floor level of the building

$$output = abs\{(gene_1 \cdot x^2 + gene_2 \cdot x + gene_3) \cdot gene_4\} \quad \text{.....(1)}$$

Recall that each column, each brace and each bay width per floor were tasked to evolve until minimum weight AND minimum top deflection were obtained. For a structure with three of four stories this is not overwhelming. But the problem grows exponentially more complicated as the number of stories increases. In order to reduce the number of genes to evolve, we tasked the algorithm to evolve only the output of the polynomial shown in Equation 1. Notice the parabolic graph depicted in Figure 4. The abscissa of this graph is story height *x*, the ordinate is the evolved output. After evolutionary success has been achieved (defined later), the program recognizes that more structure (large width, larger sections) needs to be provided just above the base of the trussed frame (hence the parabolic rise) and it is most efficient to provide less structure at the top, which essentially becomes pointed (narrow width, section sizes become small). This is shown in Figure 9.

The use of this parabolic function as the central item to evolve as opposed to the myriad of choices presented in a multi-story frame greatly increased the efficiency and the elegance of this program. It decreases the number of parameters from 30 to 10, drastically reducing the number of possible combinations. Yet it also created a minor challenge in that it required going to a pre-defined table to pull out cross sectional properties or bay widths in accordance to the output of the evolved Equation 1. We manually created this table of standard cross sectional properties of wide flange beams. The output of the polynomial function was then scaled and rounded to provide an integer section ID within the available range. See equation (2) for this calculation. All dimensions were converted to metric for the Karamba analysis.

$$section(level) = round(\frac{output(level)}{\max(output)}(Scale)(Max\ Section\ ID))\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

The applied wind loading arises from ASCE 7 Building Code, and it correctly captures the wind variation over the height of the building, but it uses only a single exposure as shown in Figure 3.

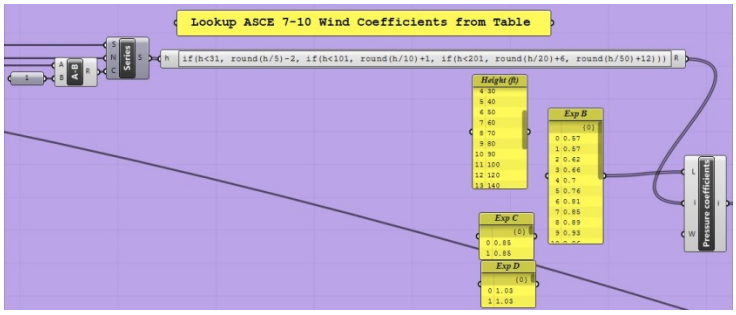

Figure 5. Applying correct ASCE wind loads via table lookup

As noted in the Introduction, all the structural analysis was performed using Karamba. The images were continuously drawn in Rhino throughout the evolution (Rhino running in the background), and to make the Rhino lines perform as structural elements in Karamba the Line to Beam function was used as shown in Figure 4. Notice that Karamba allows bases to be fixed or pinned. All points were restrained in the *y* direction, this creates stability for this 2D truss in the program's 3D workspace.

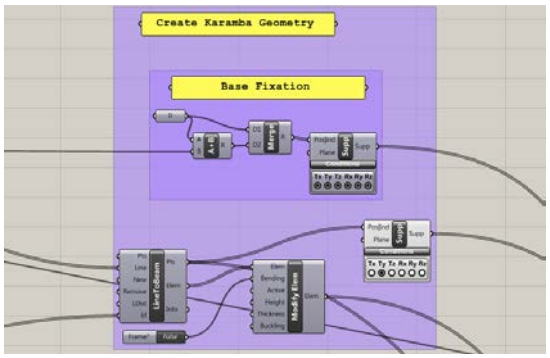

Figure 6. Changing drawn elements to structural entities

After Karamba performs its structural analysis on the trussed frame, all the information is sent to the genetic algorithm in Galapagos where several decisions can be controlled by the user. These highlighted wires indicate parameters that are allowed to be modified. This highlighted wire shows the fitness value being fed back into Galapagos.

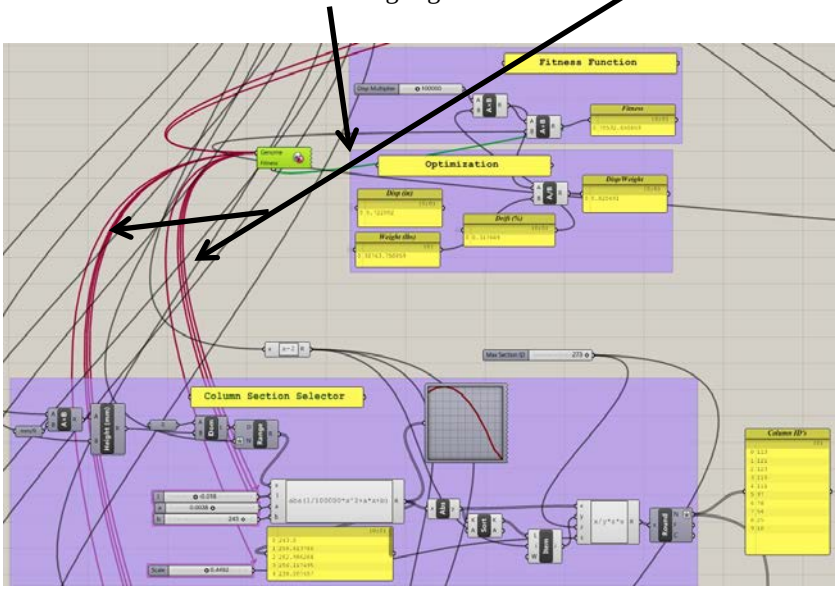

Figure 7. Structural analysis complete, regenerate gene pool

2.1. Decisions within the genetic algorithm

The overarching evolution which occurs within the genetic algorithm is based on a "fitness function". Fitness here is defined as the combination of two somewhat opposing ideas: minimum weight of the structure as well as minimum deflection at the top of the frame. Obviously a lighter frame will deflect more, thus the fitness function attempted to find a balance between these two disparate entities, even the units differ. Thus we applied a multiplier to the top deflection to roughly make their magnitudes equivalent. At the end of the optimization, we can check this fitness to see how the weight influence compared to the deflection influence. This multiplier can be adjusted by the user to change the relative importance of drift vs weight. A small multiplier value will make the algorithm focus more on optimizing weight, and a large value will put more emphasis on drift. In the end, we achieved near balance between these two parameters as shown in Figure 8.

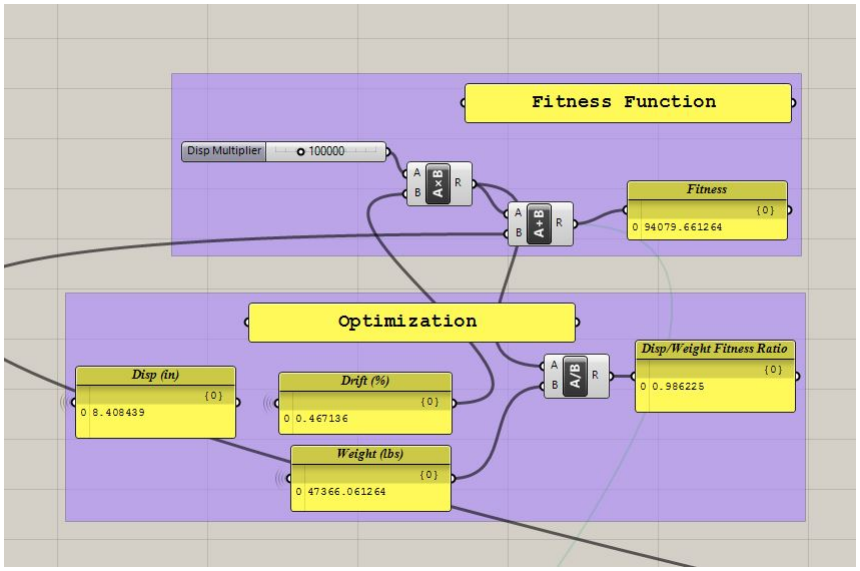

Figure 8. Fitness ratio exhibits near balance of Displacement vs. Weight

The genetic algorithm originally creates a random set of numbers, this is the initial gene pool. The user has control over the range of these numbers. The fact that the starting point is random sometimes leads to ineffective starts to the evolution process; occasionally it simply won't improve if the starting data set performs exceptionally poorly. Each gene represents one parameter that is fed into the genetic algorithm. The collection of genes that make up a single structure are referred to as an individual. The population is then filled with a variety of individuals.

The genetic algorithm then evaluates the fitness of each individual and it keeps a certain percentage for reproduction, known as survivors. This user controlled percentage of the population being retained is known as the "maintenance rate". In this study we used a maintenance rate of 5%, meaning that 5% of the individuals are retained and 95% get permutated. The maintenance rate also controls the number of permutations each survivor experiences. In our study, 95/5 = 19 means each surviving individual experiences 19 permutations. If for example we would have chosen a maintenance rate of 20%, then 80% of the spots remain open for permutation and 80/20 =4 means each gene would experience 4 permutations. The higher maintenance rate number thus means larger genetic diversity, but it also means longer computational effort. Note that making the maintenance rate extremely small can lead to unexpected consequences such as a false optimum, (or a local optimum) that may not be the best answer.

In our study at a 5% maintenance rate, we fill up the remaining 95% of the population with variants of the 5% that performed best. Ideally, this would mean a variation of structural section properties and bay width, but as explained earlier, in our study we streamlined the evolution process by having a new parameter in the polynomial that guides the selection of section sizes and bay widths.

The genetic algorithm in Galapagos allows for a "maximum stagnation" value which we set to 40. This means that if the algorithm sees no improvement over 40 generations it automatically stops. Conversely, if the process continues to improve over each generation, the program will continue to run and the genes will continue to evolve.

2.2. Results

The trussed frame has a fixed width at its base as described previously, and for this study the height of each floor was kept constant. The loads are applied to one side of the frame and they are determined from the ASCE 7 Building Code. The program creates an near optimal solution for not only the width of the bays at each floor level as is shown in Figure 9, but also by selecting the lightest columns and braces.

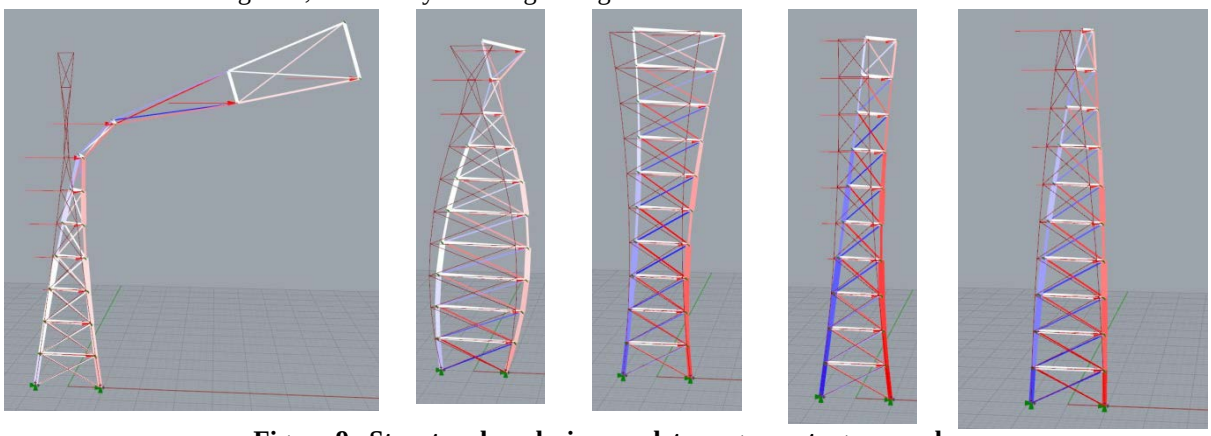

Figure 9. Structural analysis complete, regenerate gene pool

Once the optimal structure was found, we recreated this structure in a commercially available structural analysis program (SAP) to verify the validity of Karamba's structural analysis engine. The comparisons were excellent as shown in Table 1. This validates the Karamba engine and gives us encouragement to explore more complicated structural problems in this environment.

Table 1. Comparison between Karamba and SAP

	Karamba	SAP	Error
Roof Disp.	141.05mm	141.28mm	0.16%
Weight	17789kg	17835kg	0.26%

The roof displacement and the weight are used to evaluate the efficiency of the frame, as Table 2 below displays.

Table 2. Improvement due to Genetic Algorithm

	Intial	Evolved	Improvement
Roof Disp.	154.69mm	141.28mm	8.9%
Weight	32692kg	17835kg	45.5%

2.3. Conclusions

Although the use of genetic algorithms as a form finding tool is not a new idea, the present study is innovative for several reasons. First, it uses state of the art tools completely outside of the realm of traditional structural engineering software to perform structural analyses. The polynomial based selection scheme, which constrains the number of variables regardless of the size of the frame, provides an innovative approach to maintain the efficiency of the genetic algorithm. This can be thought of as a transfer function. Rather than evolving the genes, the transfer function evolves and then the transfer function is used to pick out pre-written items such as wide flange section or bay width. Even the selection process itself is novel, as very little documentation exists for these types of steps within Karamba. We created a method ourselves to perform this selection which is efficient and transparent.

For the problem at hand, a trussed frame subjected to ASCE 7 wind loads, the optimized shape makes structural sense and is similar to a Michell Truss solution which exists for a sole point load at the top of the building. The resulting form is aesthetically compelling as well as structurally sound.

Finally, we intuit that we are at the dawn of a new way of thinking about structural problems and form finding. This study, as well as others around the world, are using new parametric design tools to explore new shapes that are efficient, economical and elegant.

3. Referencing literature

[1] Koza, J.R., Genetic Programming: On the programming of computers by means of natural selection, 1st edition, Cambridge, The MIT Press, p. 18, 1992.

[2] Ohmori, H., Computational Morphogenesis: Its Current State and Possibility for the Future, International Journal of Space Structures Vol. 25 No. 2 2010.

[3] Pugnale, A., Engineering Architecture: Advances of a Technological Practice, Ph.D. Dissertation in Architecture and Building Design, Politecnico di Turino, 2010.