Parametric exploration of discrete structures using evolutionary computation

Peter VON BUELOW

University of Michigan - TCAUP
2000 Bonisteel Blvd, Ann Arbor, Michigan, USA
pvbuelow@umich.edu

Abstract

Parametric modelling software is being employed in architectural design exploration with increasing frequency. Examples of systems used in architectural design include Generative Components (Bentley Systems), Grasshopper (Robert NcNeel), Digital Project (Gehry Technologies - Dassault Systemes) and others. All of these systems are basically geometry modellers and do not provide quantitative information regarding structural behaviour or environmental quality. This paper shows an example which couples a parametric modeller with a finite element analysis guided by a genetic algorithm to explore families of structural form. In this example, the Generative Components parametric modeller is used to generate different geometries and topologies based on a programmed set of rules. The forms are then evaluated using an FEA program to determine the overall weight. The genetic algorithm is used to guide the population of solutions in the direction of least weight. In addition, human interaction is also possible through the selection of breeding pairs. The selection process is carried out visually though a web interface which allows multiple designers to act in parallel while the program is running. As a result the analysis of the individual solutions in the population occurs in parallel over a distributed network (the internet). The parameters which act as chromosome strings in the genetic algorithm become input variables for the parametric modeller, which then generates new structural forms as the cycle repeats. The full cycle includes the following steps:

1. Selection – through a web based interface
2. Breeding – within the genetic algorithm
3. Form Generation – using the parametric modeller
4. Analysis and Member Design – with FE computation
5. Ranking in the Population – collected over the internet

Finally, advantages and limitations to the procedure are discussed. Details from an actual example are show with graphic results, and recommendations are made for continued development of the procedure.

Keywords: parametric associative design, genetic algorithm, evolutionary computation

1. Introduction

In the past decade, genetic algorithms (GAs) and other forms of evolutionary computation have found ever increasing use in the architecture and engineering communities (Besserud and Cotton [2]). In the area of geometry optimization or exploration many examples can be cited for both continuous and discrete structures (Kicinger, et al. [5]). One of the limitations of GAs is that the computational effort increases rapidly with the size of the variable string (chromosome) being optimized. Of course a greater number of variables will take more computation to evaluate. But in addition, as the length of the chromosome string increases, the size of the population pool must also be increased in order to provide sufficient opportunity for different combinations of these variables to be expressed (Goldberg [4]). As the population size increases, the evaluation time also increases since each individual produced must be evaluated. Also, larger populations generally take longer to converge which requires even more individuals to fill the additional generations of populations. As a result, topology optimization of complex geometric systems like grid shells and space frames is problematic for a GA.

Associative parametric modellers are able to describe complex geometries in a small number of variables. Since the geometric systems can be fully described in a script process, the only variables needed are the parameters being directly altered. This makes parametric models ideal for use with GAs since a very large system can be described with a very short string of variables.

This paper describes how a GA can be effectively used with a parametric modeller in order to explore complex structural systems in terms of from and efficiency. The software used is Generative Components and the analysis is preformed with STAAD.Pro.

2. Procedure for the GC-GA

The steps used in the GC-GA are fairly typical of traditional GAs. The difference has mainly to do with the incorporation of commercial software (in this case Generative Components and STAAD.Pro) to produce the specific geometry based on the variables in the chromosome string. GAs generally operate on strings of variables with no inherent knowledge of what they represent of how they are evaluated. A GA simply finds a set of values for a chromosome string that yields the best result from a fitness function. The fitness function itself is also a black box as far as the GA is concerned. So the use of Generative Components (GC) or STAAD.Pro for evaluation has very little impact on how the basic GA functions.

The general steps in the GC-GA procedure are as follows.

1. Selection for Breeding – through a web based interface
2. Breeding – within the genetic algorithm
3. Form Generation – using the parametric modeller, GC
4. Analysis and Member Design – with FEA computation, STAAD.Pro
5. Ranking and Population Selection – collected over the internet
6. Random Mutation – occasionally within population

These steps are outlined below.

2.1. Selection for breeding

Normally, breeding pairs are selected by a method that gives preference to the better performing solutions. Common solutions are Roulette Wheel, Sigma Scaling, Boltzmann Selection, Tournament Selection, etc. (Mitchell [7]). In the GC-GA there are two possible ways selection is implemented. If the exploration is being made interactively by the designer, or a group of designers, the selection is made directly by the user from the screen. Otherwise the GC-GA would make random pairings from the current population of parents.

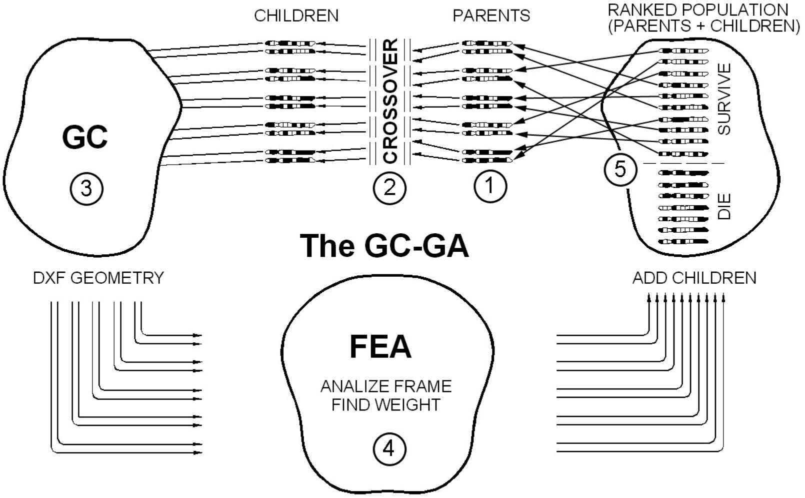

Figure 1: Diagram of the five steps in one CG-GA generation.

2.2. Breeding

Breeding combines the two chromosome strings in what is called crossover. In some way the variables are swapped between the two parent chromosomes. There are also several traditional methods for crossover, e.g. one point, two point, uniform, etc. (Mitchell [7]). The CG-GA uses a variation called half-uniform crossover. In this scheme each variable has a 50/50 chance of swapping with the corresponding variable. In the GC-GA implementation, the variables are not actually swapped but combined in a manner similar to methods used in evolutionary strategies (Bäck [1]). The value is chosen in a Gaussian distribution about the two variables. This results in one value, which is then used for the one new child chromosome.

2.3. Form Generation

The generation of the form is simply the expression of the chromosome. But with associative parametric modelling, that expression can result in very large and complex structural systems from a minimal number of variables. The examples shown below are relatively simple, but theoretically any geometry which can be generated by the parametric modelling software could be used. The variable string chromosomes produced by the breeding process can be represented by ASCII csv files. These are converted to MS-Excel format so that they can be read into GC. GC can also link to SQL or text files, but Excel affords a simple interface.

When the GC script executes it produces a screen image which can be saved as a jpg for viewing later. GC also creates a dxf version of the geometry which will be used to transfer the data to the FEA software. Of course the geometry must be compatible with the FEA software, and therein lies some difficulty. Not only must the model be well formed (complete and stable), but there needs to be additional information passed to the FEA. This includes structural information such as support location, load type and location, material and member properties, etc. This information could be included in the script of the parametric model. It could then be written to a text file which can be used to transfer the information to the FEA.

2.4. Analysis and Member Design

In this paper the GC-GA uses STAAD.Pro as the FEA software. STAAD.Pro is a production oriented office tool that incorporates many structural codes, section profiles and material property models which allows it to analyze and design a variety of systems. With the system analyzed and designed, several properties are available for use in guiding the GA selection by way of the fitness function. The weight of the system is an obvious gage of material efficiency. Stiffness in the form of deflection or modal frequency can also be used. Also, some metric to indicate erection cost, such as number of joints or members, is useful.

2.5. Ranking and Population Selection

Based on the value returned by the fitness function, each individual solution is given a score. Based on this scoring, all of the individuals in the population are rank ordered. When the population reaches full size, children = parents, the poorer performing half is deleted and the remaining half becomes the parent set for the next generation. This is strict elitist selection. As mentioned earlier other selection schemes are possible. The GC-GA is patterned after the CHC-GA by Eshelman which is designed to reduce the amount of computation and so speed the overall GA cycle (Eshelman [3]). As is generally the case with all GAs, the computational load is a problem. The GC-GA can be used with parallel processing over a distributed network which greatly speeds processing.

2.6. Random Mutation

Mutation is also a standard GA operator. In the GC-GA mutation is treated in the same way as Eshelman proposes in the CHC-GA. Within generations, as described above, there is no mutation. The generations are allowed to converge, which they do a little more quickly without the mutation. From the random start generation until the population converges (all parents about the same) is termed one cycle. The next cycle begins by taking the best performing individual form the last generation and mutating it enough times to make a new full population. This is not a "random" generation as is the case in the first cycle, but a restart based on mutants of the last converged cycle. This gives the GA a way to move off of local optima and continue to find better solutions.

2.7. The Complete GC-GA

The above steps give an overview to the GC-GA. More detailed aspects of the CHC-GA which is the bases for the GC-GA can be found in earlier publications by the author (von Buelow [6] [7]). The Cycles can continue in this automatic (non-interactive) way until a certain number end without improvement. The alternate interactive selection option runs far fewer generations and usually only one cycle at a time. This is of course due to the limited endurance of the human element. Nonetheless, since the greatest activity in terms of form variation occurs in the early generations, it is possible to get a good idea of the solution space using the interactive approach.

3. GC-GA example runs

To illustrate the process example with two runs is shown which includes the effects of different loadings and fitness criteria on the form finding. The basic example is taken from an early hyperboloid steel lattice tower by Vladimir Shukhov. It was a water tower for the All-Russian Exposition in Nijni-Novgorod in 1896. The original tower shown in Figure 2 is 37 m high and carried a water tank and observation deck.

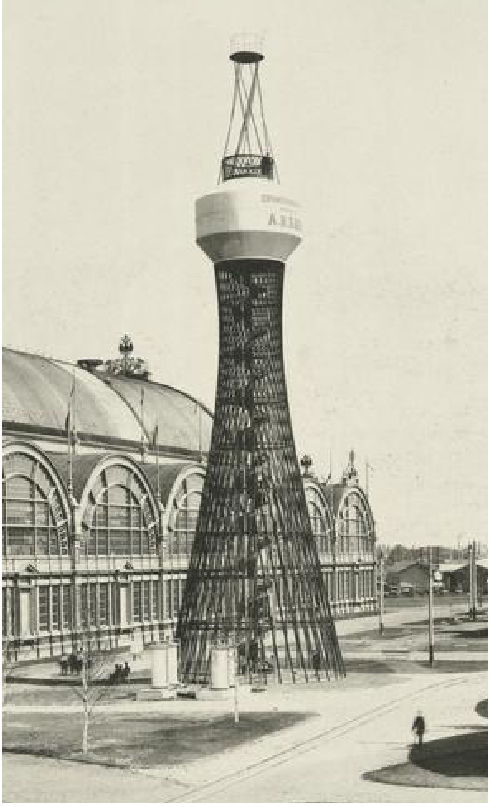

Figure 2: Shukhov's water tower used as a basis for the example problem

3.1. The parametric model in GC

Part of the reason for choosing the Shukhov tower as an example has to do with the fact that it is very effectively modelled parametrically. The model in GC contains only three variables.

- V1_DIA, the diameter of the base circle
- V2_TWIST, the angle of the lattice (twist between top and bottom rings)
- V3_DIAG_No, the number of lattice members per layer (nodes on base ring)

The above variables, along with the two dimensions held constant, are shown in Figure 3. These are sufficient to define the tower in GC. The estimated load of the water tower was 2900 kN. Since this is the dominant load it, along with self weight, was used in the structural analysis to size the members in the first trial. For simplicity, steel pipe sections were used. The fitness function in the GA was based solely on weight.

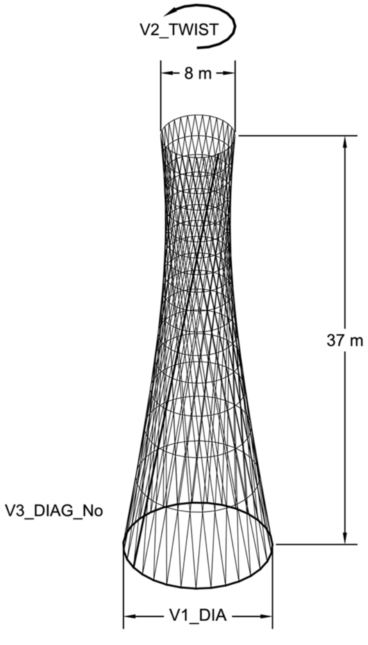

Figure 3: Basic parameters for the GC model.

3.2. Trial-A : vertical load alone

The first random generation of parents is shown in Figure 4. The size of the population was set at eight. This is a fairly small number, but with only 3 variables it works, and is easily fit on a monitor screen for viewing. More complex problems with more variables would ordinarily require larger populations, but the smaller number makes the example easier to depict. The randomly generated csv strings were run through GC to generate the geometries. Both dxf and jpg files are generated at this point in GC. Next the dxf file is read into STAAD.Pro to bring the geometry into the FEA program. A text stub is added to the end of the geometry to complete the material, support and loading information. This is fairly easy in this case because the lowest y-coordinate nodes (y is vertical) are supports and the highest are loading points (making them easy to identify). The total downward vertical load from the water tank is simply divided among the top nodes. Actual node numbers are taken from the input geometry. The total number of nodes in a ring (giving the supports and load points) is one of the original csv variables (V3_DIAG_No).

The FEA is iterated to get convergence of member size since the self weight is also calculated. STAAD.Pro is able to size members based on the AISC steel code. From the STAAD analysis the weight of the entire tower is estimated. This is the number shown under each image in Figures 5 through 9.

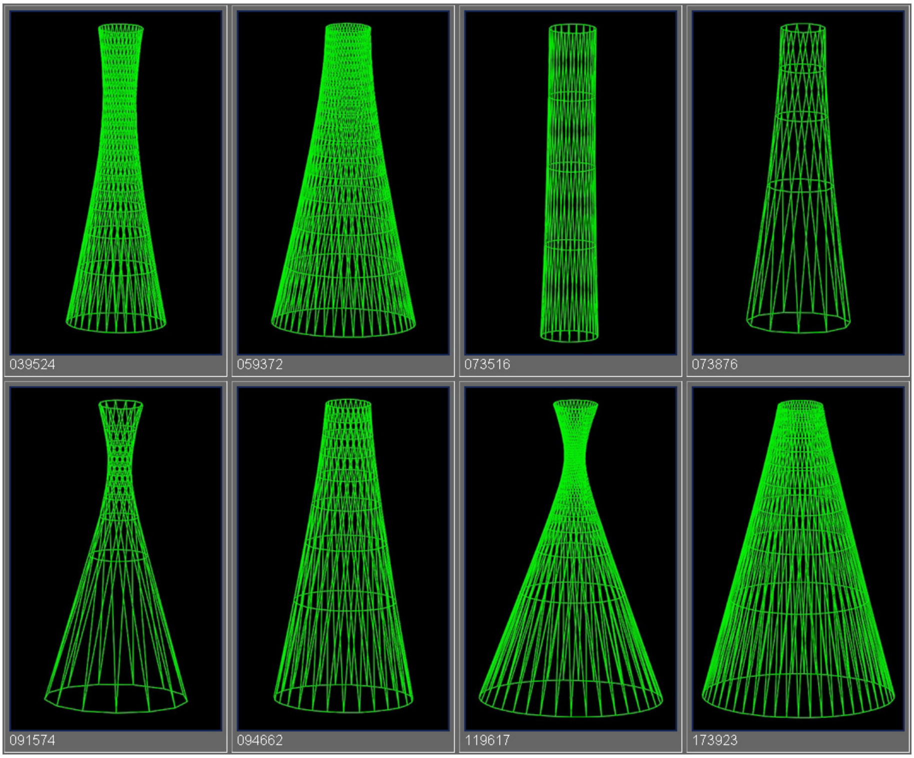

Figure 5: The first random parents, Generation 0-A from Trial-A.

The parents from the random generation in Figure 5 were bred to produce the next generation. Both children and parents of the next generation were sorted by weight and the lightest eight were retained to form Generation 1-A shown in Figure 6. In the same way the process is repeated to form subsequent generations. The next generation is shown in Figure 7. Even by Generation 2-A there is a preference for towers that are more slender than Shukhov's original form. Of course no lateral loads were considered in this first trial.

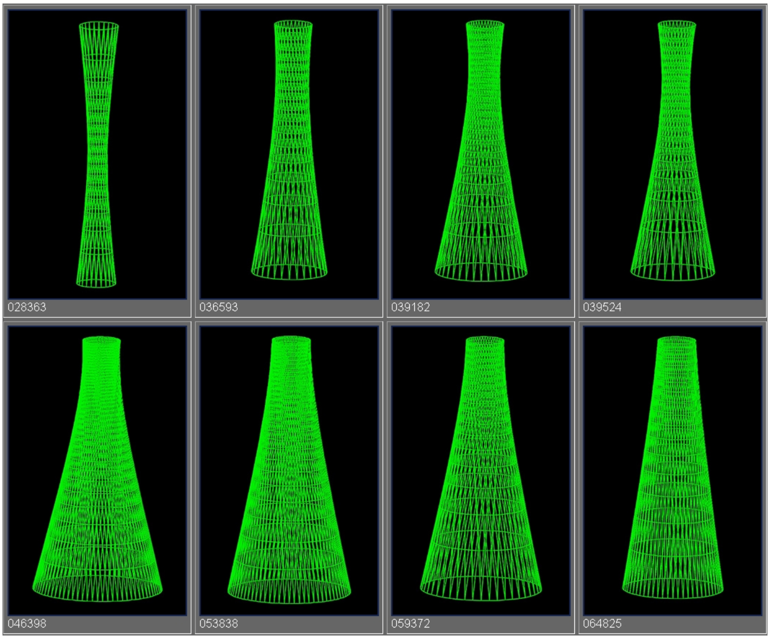

Figure 6: The first bred generation, 1-A in Trial-A after elitist selection by weight

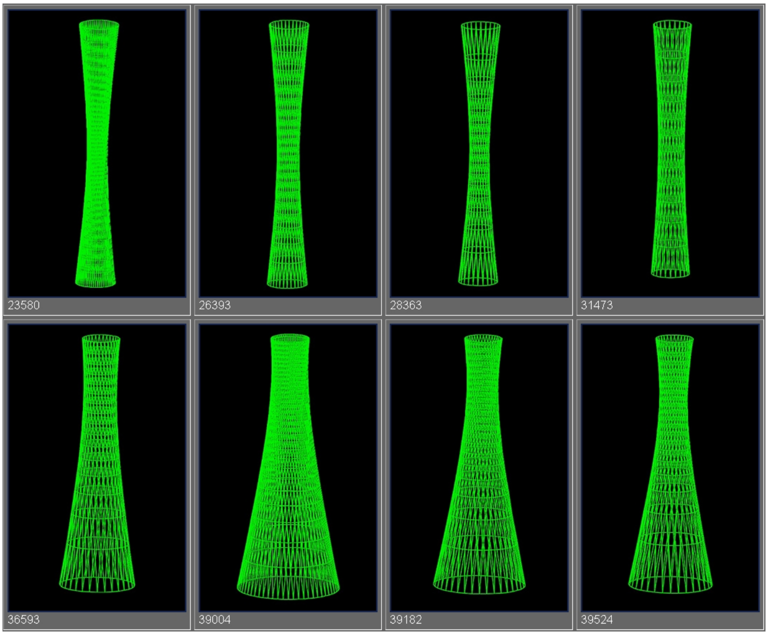

Figure 7: Generation 2-A of Trial-A, already showing a preference for more slender towers

3.3. Trial-B : vertical and transverse loads with node penalty

In the second trial a transverse wind load was added. For the purpose of form determination, the wind load is approximated by two loadings which were assumed to act in any direction. Both loads were based on a wind of 145 km/hr using methods described in the ASCE-7 [6] combining pressures of both windward and leeward sides into a single force. The first load represents the wind on the water tank. This load was simplified to a single horizontal load based on the projection of the water tank, taken as a 3 m diameter circle. The load of 80 kN was distributed evenly to the nodes at the top ring. This neglects vertical components of the load as well as the moment on the top ring, but is enough to model the effect of the transverse load. The second transverse load was the wind drag on the members themselves. This was approximate by a factored self weight (0.07 x self weight) placed in the same transverse direction as the wind. Since self weight varies with diameter there is some accounting for size of the members. The factor was chosen to be about right for 100 mm diameter pipe.

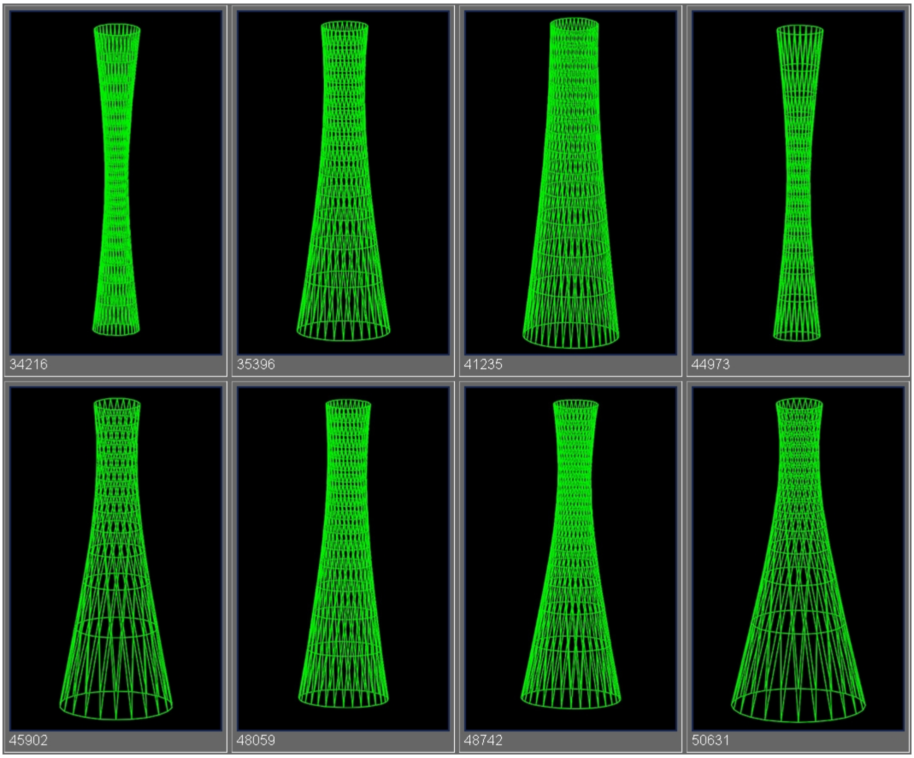

Figure 8: Generation 1-B of Trial B, including transverse loads and a joint number penalty

In order to take into account the additional cost of joints, a penalty was added to the fitness function for the total number of joints in a system. In order to combine this into one fitness value for both weight and joint cost, an additional 25 pounds was added for each joint. This is merely intended to illustrate the technique and the effect on form. No study was made to determine the weighting number based on actual cost. But again, it serves to illustrate the use of the GC-GA.

The initial random generation from the first trial, A, was used again, however the individuals were all reanalysed for the new criteria including transverse wind load and joint penalty. Figure 8 shows the first bred generation, 1-B. The second generation, 2-B, is show in Figure 9.

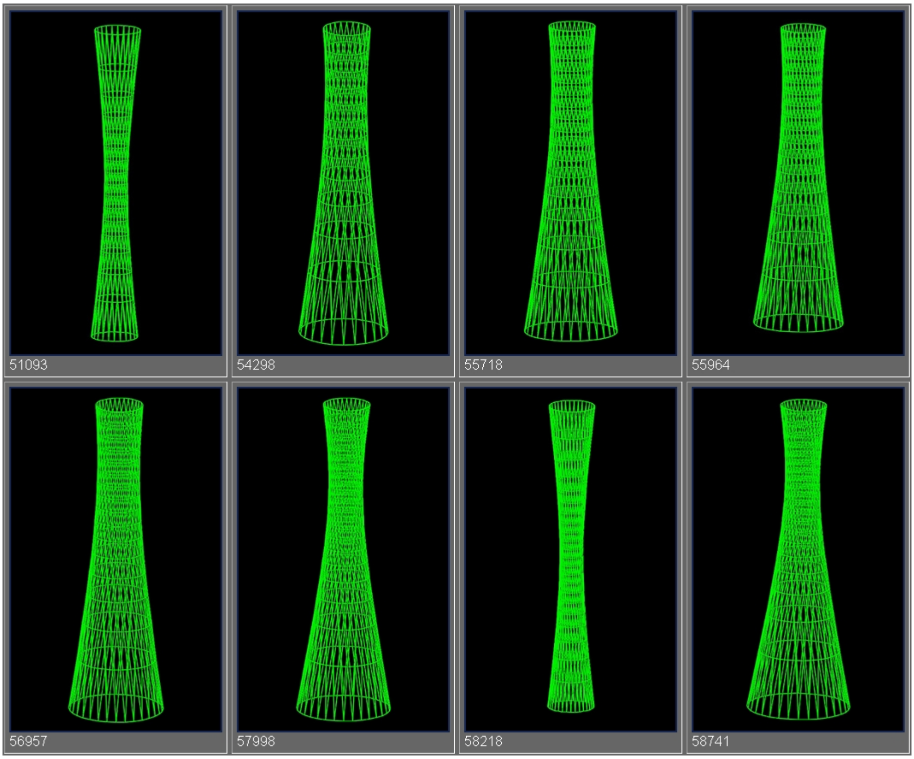

Figure 9: Generation 2-B from Trial B

In subsequent generations the more slender forms again dominated. The transverse load of the wind was not enough to have a significant effect. Lateral deflection was higher and in some of the slender cases beyond recommended limits of 1/200 of height. The penalty for high numbers of nodes did reduce the upper number on nodes which remained in the population.

4. Conclusions

The main purpose of this paper is to illustrate the application of the GC-GA for the exploration of structural form. The GC-GA combines associative parametric modelling software with finite element analysis software and a genetic algorithm. Because the parametric model is able to describe large, complex structural systems in a small number of variables, a GA can be used effectively to navigate the design space described by the model using the fitness supplied by the FEA. This greatly expands the size and complexity of structural forms that can be investigated using GAs.

A simple example based on Shukhov's water tower shows how quickly the GA is able to converge on solutions that perform well for the given criteria. The example was shown for different criteria and load conditions. The form that was found responded well to the given objectives. Other parameters, such as foundation conditions, limited member sizes or other loadings such as seismic, would also impact the form, but were not considered in this example.

The concept of the CG-GA offers much promise. It can easily be expanded to include other performance criteria that impact architectural form including lighting, thermal performance, energy utilization, etc. Research efforts at the University of Michigan are continuing with this topic.

References

[1] Bäck T. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, Oxford, England, 1996.

[2] Besserud K and Cotton J., Architectural Genomics, in *Silicon+Skin > Biological Process and Computation*. Proceedings of the 28th annual conference of ACADIA, Kudless A, Oxman N and Swackhamer M (eds.), 2008, 238-245.

[3] Eshelman L. "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", in: *Foundations of Genetic Algorithms*. Morgan Kaufmann Publ., San Mateo, California. 1991.

[4] Goldberg D.; Deb K.; and Clark J. "Genetic Algorithms, Noise, and the Sizing of Populations", IlliGAL Report No. 91010. Univ. of Illinois, Urbana-Champaign, IL. 1991.

[5] Kicinger R., Arciszewski T. and De Jong K. "Evolutionary computation and structural design: A survey of the state-of-the-art". *Computers & Structures.* vol. 83, 2005, pp 1943-1978.

[6] *Minimum Design Loads for Buildings and Other Structures- ASCE 7-02*. American Society of Civil Engineers, Reston Virginia, USA. 2002.

[7] Mitchell M. *An introduction to genetic algorithms*. The MIT Press, Cambridge, Mass., 1996.

[8] von Buelow P. "Suitability of Genetic Based Exploration in the Creative Design Process" in *Digital Creativity*. Routledge, Vol.19, No.1, pp.51-61, 2008.

[9] von Buelow P. *Genetically Engineered Architecture: design exploration with evolutionary computation,* VDM Verlag Dr. Müller, 2007.
(also at: http://elib.uni-stuttgart.de/opus/volltexte/2007/3160/pdf/PvB_diss.pdf)