IASS 2020/21
SURREY 7
International Conference on
Spatial Structures 2020/21

Proceedings of the IASS Annual Symposium 2020/21 and
the 7th International Conference on Spatial Structures
Inspiring the Next Generation
23 – 27 August 2021, Guilford, UK
S.A. Behnejad, G.A.R. Parke and O.A. Samavati (eds.)

Computational form exploration of branching columns using concepts of formex algebra and the ParaGen method

Peter VON BUELOW*, Anahita KHODADADI[a]

*University of Michigan
Ann Arbor, Michigan, USA
pvbuelow@umich.edu

[a] Portland State University, USA

Abstract

This study addresses the relationship between the geometry and structural performance of branching columns using an example case based on a square grid shell supported by four branching columns. The branching columns are configured with three levels of members, bifurcating into four members as each member branches upwards. A range of solutions is parametrically generated using the concepts of formex algebra and its associated software system, Formian 2.0. The form exploration uses the GA-based method, ParaGen which incorporates both quantitative structural performance and qualitative architectural considerations in the exploration process. Certain design constraints, as well as multiple objectives, are established including minimizing structural weight and deflection, and increasing vibration stiffness, in addition to the designer's satisfaction with the visual appearance of the columns. Within the iterative process of form generation, the structural performance of the branching columns under a combination of self-weight and snow load is evaluated using the Finite Element Analysis (FEA) software, STAAD.Pro. The branching members are sized based on the AISC LRFD steel code. ParaGen creates a database of suitable solutions, which can be explored by filtering and sorting based on a variety of performance parameters. Different techniques are demonstrated in the exploration of good solutions including scatter point graphs, Pareto front analysis, and images of the design alternatives.

Keywords: branching columns, tree columns, exploration, form finding, genetic algorithm, formex algebra, Formian, ParaGen

1. Introduction

Branching structures are attractive to architects and engineers because of their novel appearance and significant structural performance. Since Frei Otto's fundamental studies on branching structures in the early 1960s [1], many researchers have investigated various model-based and numeric methods to conduct the form-finding analysis of these structures. Jörg Schlaich's office, Schlaich, Bergermann Partner (SBP), engineered several small bridges as well as work on the much larger supports for Terminal 3 at the Stuttgart Airport (architect von Gerkan, Marg und Partner) [2].

The geometry of a branching column might be termed "form active" in that the geometry is determined by the forces present. Normally the best forms would be ones which placed the members in compression only (no bending). Frei Otto used hanging string models to find geometries acting in tension only. When the geometry is then flipped upside down, it becomes a pure compression system. For a given start (base) and end points (tops of upper branches) there is actually a whole array of possible compression only geometries. These can be easily seen using string models with sliding nodes as shown in Figure 1.

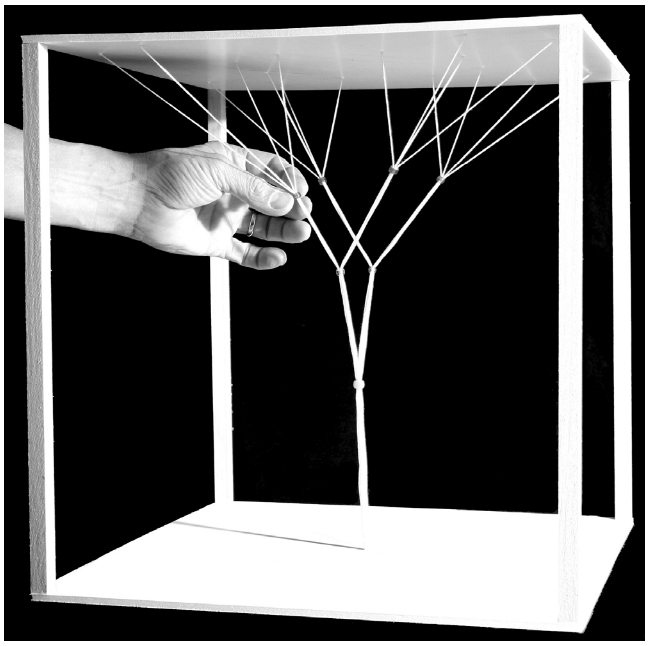


Figure 1: A tension thread model with adjustable nodes used to study compression only geometries.

But the problem with the string models is knowing which geometry is actually the most structurally efficient. With the members acting in compression, buckling becomes a controlling factor and so there is a relationship between the force in the member and its length. This means that under different levels of loading, the ideal lengths and thus the branching points for the members will be different. The effect of differing load levels on the least weight branching geometry is discussed in more detail in an earlier IASS paper by von Buelow [3].

In this study, the parametric model of four branching columns and the single-layer grid shell is created in Formian 2.0 using the concepts of formex algebra. In this mathematical system, the topology of the spatial structures can be formulated through some geometrical concepts including, movement, propagation, deformation, and curtailment [4].

An interactive GA-based form exploration method, called ParaGen [5], is then used to explore the design alternatives generated by Formian. Each alternative is analysed with the FEA STAAD.Pro [6] to determine the required members sizes and performance values of deflection, stiffness and weight. The performance values as well as the overall form are then used to search for the desired solution. The following sections describe the configuration processing of the tree columns to shed light on opportunities and challenges that the use of available editions of the Formian software system may bring to the design process. In the end, the outcome of the form exploration process is discussed.

2. Configuration of the branching columns

In this form exploration project, a set of four branching columns along with a single-layer grid shell are modeled parametrically to cover a 40 m × 40 m area. This is a similar area per column as in the Stuttgart airport mentioned above , however in Stuttgart the columns have 4 levels whereas in this example there are only 3 levels (base column, first-level branches, and second-level branches). The curved grid shell is 12 m at the corners and rises to 14 m on the edges and 16 m at the center of the grid as shown in Figure 2. The branching columns in this study are defined with two levels of branches. As a convention, branches that sprout from the trunk are denoted as first-level branches, and in a similar fashion, components that sprout from the first-level branches are called second-level branches [7].

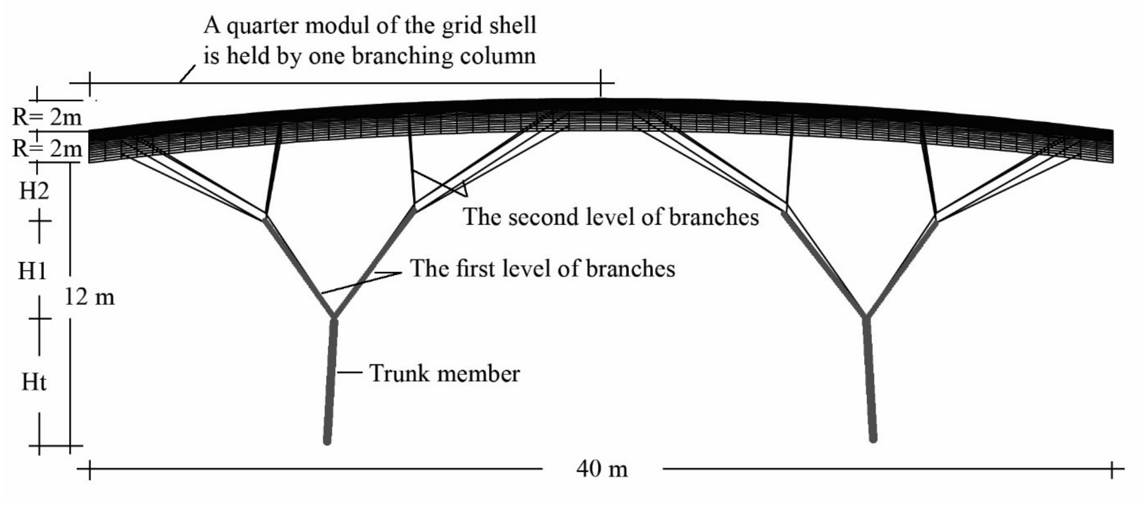

A quarter modul of the grid shell is held by one branching column
R= 2m
R= 2m
H2
H1
12 m
Ht
The second level of branches
The first level of branches
Trunk member
40 m

Figure 2: The curved grid shell held by the four branching columns is configured by the application of the Pellevation function in Formian.

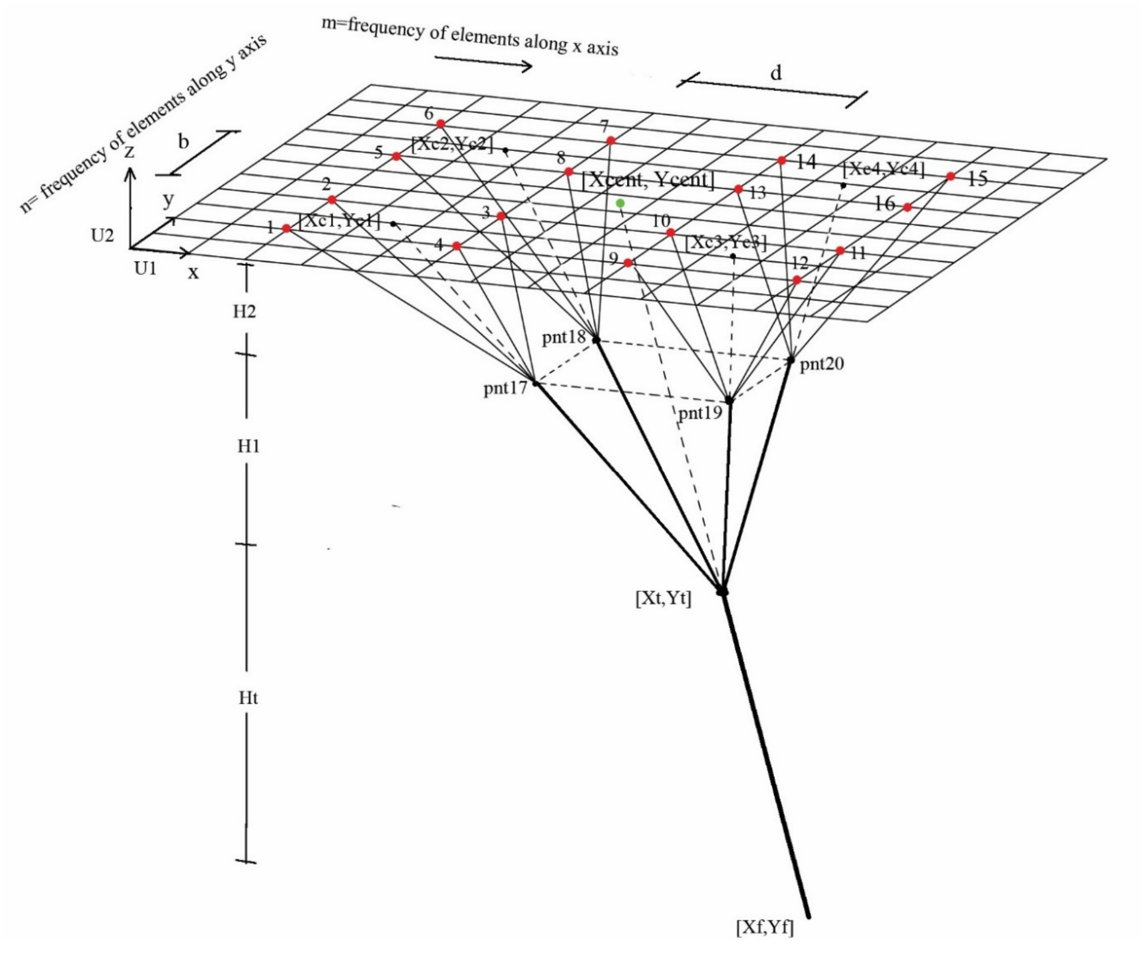


Figure 3: Configuration of a typical branching column studied in this paper

To replicate a string model and minimize the bending moment in the structural components, the topology of the branching columns should be defined such that the axial direction of each branch passes through the centroid of the area created by its upper-level nodes. For example, Figure 3 shows that the axial direction of the trunk passes the centroid of the area created by its upper-level nodes, including points 17, 18, 19, and 20. Or the axial direction of the branch passing through point 17 is extended to the centroid of the area created by the upper-level nodes at point c1.

The entire structure is modeled using the concepts of formex algebra via its associated software system, Formian 2.0. Further information about configuration processing of diverse spatial structures can be found in [8, 9, 10, 11, 12, 13, 14, 15]. Application of the latest edition of the Formian software system, Formian K, could be helpful for configuration processing of the branching structures and the grid shell and visualization of the output. However, the number of elements in this parametric model exceeds the maximum number of structural elements in Formian K, and the Pellevation function is not enabled. Thus, in this study, the earliest edition of Formian 2.0 is used for the configuration processing.

Within the Formian model, the topology of the four branching columns and the grid shell is defined in the global XYZ Cartesian coordinate system. The base point of the trunk, the height of each set of branches along the Z-axis, and the closeness of each set of second-level branches are the variables to which the designer should assign the appropriate values. The list of geometrical parameters, along with their acceptable intervals, are given in Table 1.

Table 1: Geometrical parameters of the folded plate diamatic dome

Parameters	**Sign**	**Constant/ Variable**	**Acceptable interval/ value**
Coordinates of the base point of the trunk	X_f, Y_f,	Variable Variable	[1, L_slab/4] = [1, 10] [1, W_slab/4] = [1, 10]
Length and width of the grid	L_slab, W_slab	Constant Constant	40m , 40 m
Rise of the curved vault	R	Constant	2 m at edges, 2×2 m at the center
frequency of elements along X and Y axes	M_slab N_slab	Constant Constant	40 40
frequency of elements between each set of branches at the highest level along X and Y axes	b d	Variable Variable	$1 \leq b \leq 8$ $1 \leq d \leq 8$
Height of the trunk along the Z-axis	H_t	Variable	$H_t+H_1+H_2 = 12$ m
Height of the 1st level of branches along the Z-axis	H_1	Variable	
Height of the 2nd level of branches along the Z-axis	H_2	Variable	

The configuration processing of the branching columns begins with the parametric modeling of a 40 m × 40 m flat grid shell, its centroid, the joints where the branching columns touch the grid, and the relevant centroid of each set of four branches. Then, the flat grid shell and essential points are projected onto a curved surface by applying the Circular Barrel Pellevation function twice. Once the barrel cap is situated along the X axis and once along the Y axis. Further information about the Pellevation function can be found in reference [4]. After the centroids are projected onto a curved surface, their new z coordinates are obtained by executing a "Give" statement (see Table 2). When the coordinates of the centroids of the grid shell and the second level of branches are identified, the configuration of the branching columns is processed using the concept of a tension string model to minimize the bending moment in columns. The appropriate tilt angles of the trunk and branches are computed by using the equation of a straight line in three dimensions in a Cartesian coordinate system presented in Table 3. In the end, the four branching columns are obtained by reflectional replication of a trunk.

Table 2: Processing the topology of the curved grid shell, pntcent, and points C1, C2, C3, C4

```
d=6;                      (*)Frequency of elements between the set of 4 highest level of branches along X axis (*)
b=4;                      (*)Number of elements between the highest nodes along width (Y axis)(*)
L_slab= 40;               (*) Span of the vault OR the flat grid slab in m. (*)
W_slab= 40;               (*) Width of the vault OR the width of the grid slab in m. (*)
M_slab= 40;               (*)Frequency of element along length (X axis) of the flat grid (*)
N_slab= 40;               (*)Frequency of element along width (Y axis) of the flat grid (*)
m= M_slab/2;              (*)Frequency of element along the tributary area of a tree column (*)
n=N_slab/2;               (*)Frequency of element along the tributary area of a tree column(*)
U1=L_slab/M_slab;         (*) X coordinate unit equals to 1 m (*)
U2=W_slab/N_slab;         (*) Y coordinate unit equals to 1 m (*)
R=2;                      (*)Rise of the curved vault at the edges equals 2 m(*)

(*) Coordinates of the centroid of the flat grid (*)
Xcent=U1*m/2;             (*) X coordinate of the centroid of the tributary area (*)
Ycent=U2*n/2;             (*) Y coordinate of the centroid of the tributary area (*)
Pnt_cent=bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|bapel
(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[Xcent,Ycent,0];

(*) Coordinates of the 4 centroids on the second level of branches on a curved grid (*)
stepx= (m-2-2-d)/2;   (*)The step between the four branches in each set along x axis(*)
stepy= (n-2-2-b)/2;   (*)The step between the four branches in each set along y axis(*)

pnt_c1= bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|bapel
(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+U1*stepx/2, 2*U2+U2*stepy/2, 0];
pnt_c2= bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|bapel
(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+U1*stepx/2, 2*U2+1.5*stepy*U2+b*U2, 0];
pnt_c3= bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|bapel
(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+1.5*stepx*U1+d*U1, 2*U2+U2*stepy/2, 0];
pnt_c4= bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|bapel
(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+1.5*stepx*U1+d*U1, 2*U2+1.5*stepy*U2+b*U2,0];

Use &, vm(2), vt(2), c(1,1), vh(L_slab,0,100000, 0,0,0, 0,0,20);
Clear;

Give pnt_c1#pnt_c2#pnt_c3#pnt_c4;
```

Table 3: Computation of the appropriate tilt angles of the trunk and the first level of branches. To minimize the bending moment in the structural components, the topology of the branching columns is defined such that the axial direction of each branch passes the centroid of the area created by its upper-level nodes.

Tilt angle of the trunk:

$$\frac{X_t-X_{cent}}{X_{f-}X_{cent}} = \frac{Y_t-Y_{cent}}{Y_{f-}Y_{cent}} = \frac{Z_t-Z_{cent}}{Z_{f-}Z_{cent}} \quad \rightarrow \quad \frac{X_t-X_{cent}}{X_{f-}X_{cent}} = \frac{Y_t-Y_{cent}}{Y_{f-}Y_{cent}} = \frac{-H1-H2-Z_{cent}}{H_{Total\_}Z_{cent}}$$

Z_{cent} is obtained by using the Give function in Formian after the Pellevation function of the flat grid shell is applied and the grid is suitably curved.

Tilt angle of the first level of branches:

$$\frac{X_{17}-X_{c1}}{X_{t-}X_{c1}} = \frac{Y_{17}-Y_{c1}}{Y_{t-}Y_{c1}} = \frac{Z_{17}-Z_{c1}}{Z_{t-}Z_{c1}} \quad \rightarrow \quad \frac{X_{17}-X_{c1}}{X_{t-}X_{c1}} = \frac{Y_{17}-Y_{c1}}{Y_{t-}Y_{c1}} = \frac{-H_2-Z_{c1}}{-H_2-H_1-Z_{c1}},$$

$Z_{c1}, Z_{c2}, Z_{c3}, Z_{c4}$ are obtained by the Give function in Formian after the Pellevation function of the flat grid shell is applied and the grid is suitable curved.

Each branch of the tree structure could be generated by either producing the cantle from the signets of the two endpoints using the Tignum function or directly echoing the coordinates of the two endpoints in brackets. The former method does not always generate the expected branch, and the software system occasionally crashed due to some internal problems. Also, the corresponding DXF file of branched will not be exported since the Tignum function seems to yield a numeric definition of the coordinates rather than a geometrical output. Meanwhile, the latter method seems convenient only for producing the trunk and the first level of branches, where the exact coordinates of the two endpoints of an element are computed in the parametric model. However, after projecting the original flat grid onto a curved surface, identification of the coordinates of the top points of the second level of the branches is not straightforward.

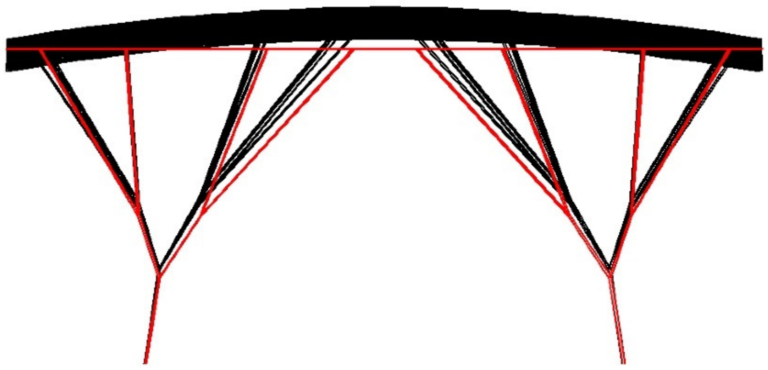


Figure 4: Displacement of nodes and branch elements when a Pellevation function is applied to grid shell and the branching structure. The branching columns in red hold a flat grid, while the branching columns and the curved grid shell in black are created by the application of a Pellevation function.

Table 4: Definition of 4 branches at the second level using the novation function.

```
(*) Slope of the first level of branches between point17 and point c1 (*)
lxb17= Xc1-Xt;
lyb17=Yc1-Yt;
lzb17=Zc1+H1+H2;

(*) Coordinates of upper point of the first level of branches between point17 and point c1 (*)
X17=lxb17*(-H2-Zc1)/lzb17 + Xc1;
Y17=lyb17*(-H2-Zc1)/lzb17 + Yc1;
pnt_17=[X17, Y17, -H2];

(*) Coordinates of 4 upper points of the second level of branches on a flat grid (*)
pnt_1= bapel (1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|
bapel (1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1, 2*U2, 0];
pnt_2=bapel(1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|
bapel(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1, 2*U2+stepy*U2, 0];
pnt_3=bapel(1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|
bapel(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+stepx*U1, 2*U2+stepy*U2, 0];
pnt_4=bapel(1,L_slab/2,0,L_slab/2,W_slab,L_slab,R)|
bapel(1,0,W_slab/2,L_slab,W_slab/2,W_slab,R)|[2*U1+stepx*U1, 2*U2, 0];

(*) Second level of branches on a flat grid (*)
Brch2= [2*U1, 2*U2, 0; X17, Y17, -H2]#[2*U1, 2*U2+stepy*U2, 0; X17, Y17, -H2]#[2*U1+stepx*U1, 2*U2+stepy*U2,
0; X17, Y17, -H2]#[2*U1+stepx*U1, 2*U2, 0; X17, Y17, -H2];

(*) Using Novation function to generate the second level of branches on a curved grid (*)
P1= [2*U1, 2*U2, 0]#[2*U1, 2*U2+stepy*U2, 0]#[2*U1+stepx*U1, 2*U2+stepy*U2, 0]#[2*U1+stepx*U1, 2*U2, 0];
P2=pnt_1#pnt_2#pnt_3#pnt_4;
Branch2= nov(1,P1,P2)|Brch2;
```

In each generated tree structure, there are 16 points in the second level of branches that touch the grid shell. The coordinates of these 16 points on the curved grid shell can be obtained by executing a Give statement. But using the Give function for several points will make the iterative process of form exploration extremely complicated, long, and more probable to become flawed.

Another way of generating the second level of branches seems to be the application of the Pellevation function. However, this procedure displaces the joints and branches that are expected to pass through the centroid of the area created by its upper-level nodes (see Figure 4). Thus, in this study, the only suitable method of defining the second level of branches was to apply a sharp Novation function on the 16 points connecting the first level of branches to the original flat grid shell and transforming them to the 16 corresponding points on the curved grid shell (see Table 4).

3. ParaGen exploration

ParaGen is an exploration method that was first devised by the research team at Taubman College at the University of Michigan. It is termed a "method" rather than a software "application" because it incorporates several different in-house and commercial applications to perform the exploration. This gives ParaGen considerable latitude in how and what it is able to explore. Several presentations have been given over the past dozen years at IASS symposia that illustrate details of the method through examples, but von Buelow [16] gives a good description of the method and the advantages of a Non-destructive Dynamic Population GA (NDDP-GA) in exploration.

The ParaGen method is a cyclic process that combines a GA to generate input to a parametric model. The model is then evaluated for parameters that can be best used to describe it. The results are then stored along with images and the model itself in an online database. This database of all evaluated solutions can be searched (explored) in a variety of ways (graphs, sets of sorted images, etc.) to interactively find a suitable solution to the problem. Because the database and the graphic interface are accessed through a website, any number of people can collaborate and explore the solutions together. Also, because the database storage can be accessed by multiple machines at the same time through the internet, parallel processing of the entire cycle becomes relatively easy which speeds up the creation and storage of the solutions. In addition, because the capacity of SQL database storage is so large, there is no need to delete any solutions – hence "Non-destructive". Instead, every unique solution generated by the GA is retained. This also speeds up the GA since no duplicate solutions ever need to be analysed. It also makes possible the changing of the objective function without restarting the process. This can in fact be done interactively as the process is being run to better or alternately define the search direction. In this way, ParaGen can offer more toward "exploration" than traditional "optimization" tools can do.

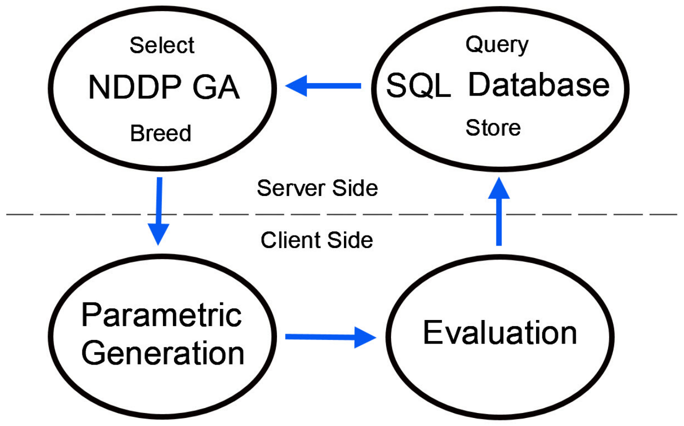


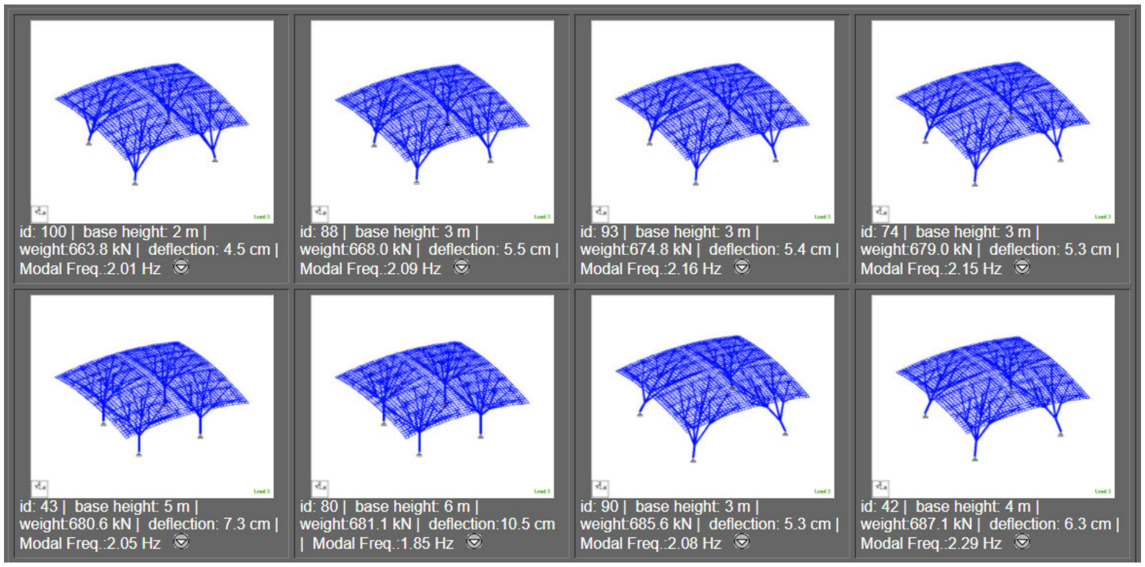
Figure 5: The ParaGen exploration cycle

In this project, the focus is on the structural performance of the system, and so the evaluation is limited to an FEA using STAAD.Pro. Nonetheless, visual images of the structures were also captured and can be used to include qualitative assessment based on aesthetic or other non-structural qualities. In addition, all of the files used to perform the assessment are retained and can easily be accessed for more detailed evaluation in the exploration process. As is usually the case the exploration run started by generating an initial number of solutions based on a random selection of the defining variables. After this initial population was produced, selective breeding began. Any of the measured performance parameters or combinations of them can be chosen to guide the selection of "parents". In this case each of the key parameters were successively selected to created a breeding pool of parents. For example, a query was made to select the 10 lightest weight solutions and then from that pool of 10 to randomly select two parents for breeding the next solution. This process was continued with the other performance parameters as a way of generating a range of well performing solutions that could be explored.

4. Results

Each solution was designed using Hollow Structural Sections (HSS) pipe following the American Institute of Steel Construction (AISC) LRFD code. The load on the structure was given as its own selfweight + 0.125 kN/m^2 applied dead load and an assumed snow load of 1 kN/m^2. The design objectives were to minimize the structural material required (selfweight) and to maximize the overall stiffness (minimize deflection and maximize modal frequency).

A simple sort of the results by each of these parameters gives an initial impression of performance. Figure 6 shows a simple sort to find the 8 solutions with the lightest weight, while Figure 7 shows a simple sort to find the least deflection. It will be noted however, that there is no overlap between these two sets.



id: 100 | base height: 2 m | weight:663.8 kN | deflection: 4.5 cm | Modal Freq.:2.01 Hz

id: 88 | base height: 3 m | weight:668.0 kN | deflection: 5.5 cm | Modal Freq.:2.09 Hz

id: 93 | base height: 3 m | weight:674.8 kN | deflection: 5.4 cm | Modal Freq.:2.16 Hz

id: 74 | base height: 3 m | weight:679.0 kN | deflection: 5.3 cm | Modal Freq.:2.15 Hz

id: 43 | base height: 5 m | weight:680.6 kN | deflection: 7.3 cm | Modal Freq.:2.05 Hz

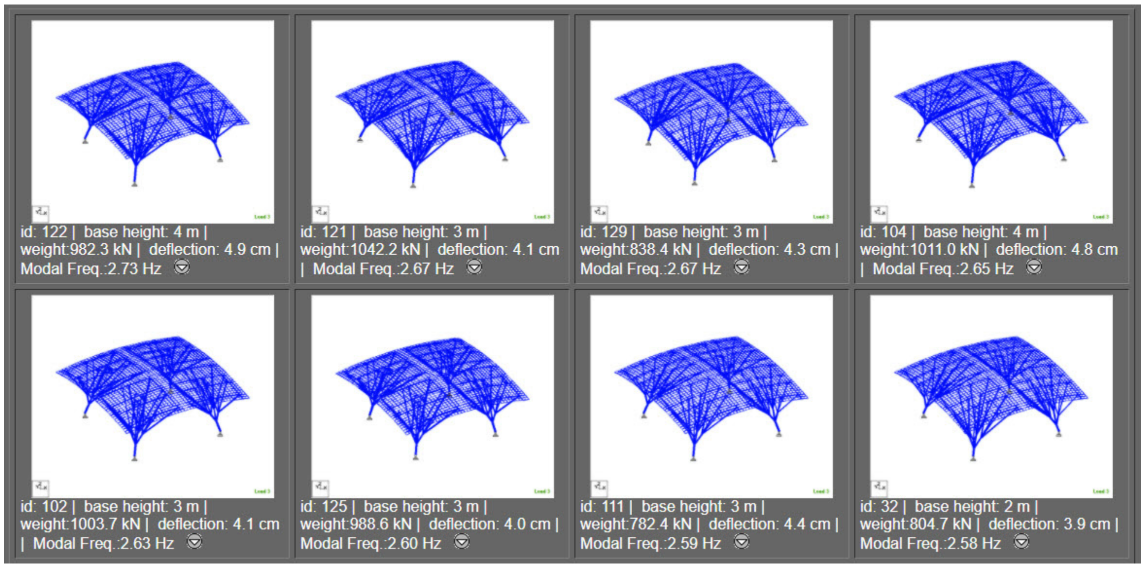
id: 80 | base height: 6 m | weight:681.1 kN | deflection:10.5 cm | Modal Freq.:1.85 Hz

id: 90 | base height: 3 m | weight:685.6 kN | deflection: 5.3 cm | Modal Freq.:2.08 Hz

id: 42 | base height: 4 m | weight:687.1 kN | deflection: 6.3 cm | Modal Freq.:2.29 Hz

Figure 6: A set showing solution sorted by least weight first (8 lightest solutions).

In fact, this is an example of conflicting design criteria. A common approach to finding a good solution in this case is to find the Pareto set of solutions. A Pareto (level 1) set is composed of solutions which are non-dominated for the given criteria by any other solutions in the population. These are commonly known as best trade-off solutions. It is possible to generate a Pareto set for any conflicting criteria. A plot can be shown for two or three criteria (beyond that a set can be calculated and shown but not plotted). Figure 8 shows a plot with the first level Pareto solutions for modal frequency vs. weight. One can also click on any dot on the graph to see an image of that solution. Generally, in Pareto optimization the extreme ends of the graph are less useful since they show only the best for one or the other of the criteria. In viewing the results on the ParaGen website the range of results depicted can be restricted to show only the central part of the curve. Figure 9 shows the four best trade-off solutions for modal frequency vs. least weight.



id: 122 | base height: 4 m | weight:982.3 kN | deflection: 4.9 cm | Modal Freq.:2.73 Hz

id: 121 | base height: 3 m | weight:1042.2 kN | deflection: 4.1 cm | Modal Freq.:2.67 Hz

id: 129 | base height: 3 m | weight:838.4 kN | deflection: 4.3 cm | Modal Freq.:2.67 Hz

id: 104 | base height: 4 m | weight:1011.0 kN | deflection: 4.8 cm | Modal Freq.:2.65 Hz

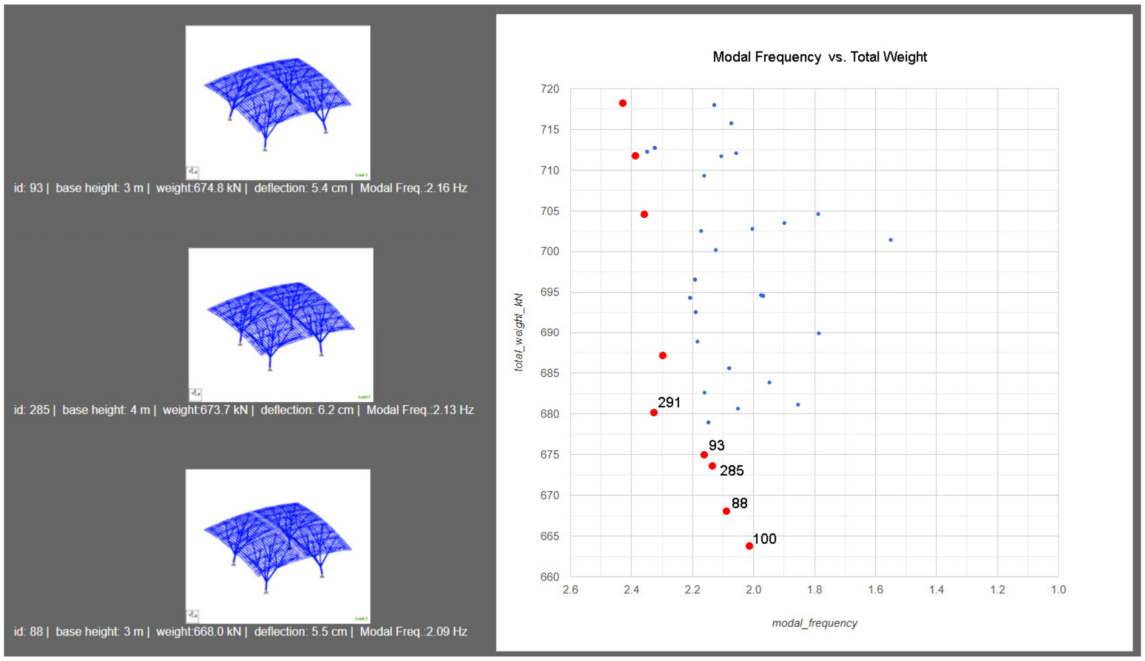
id: 102 | base height: 3 m | weight:1003.7 kN | deflection: 4.1 cm | Modal Freq.:2.63 Hz

id: 125 | base height: 3 m | weight:988.6 kN | deflection: 4.0 cm | Modal Freq.:2.60 Hz

id: 111 | base height: 3 m | weight:782.4 kN | deflection: 4.4 cm | Modal Freq.:2.59 Hz

id: 32 | base height: 2 m | weight:804.7 kN | deflection: 3.9 cm | Modal Freq.:2.58 Hz

Figure 7: A set showing solution sorted by greatest modal frequency first (8 stiffest solutions).



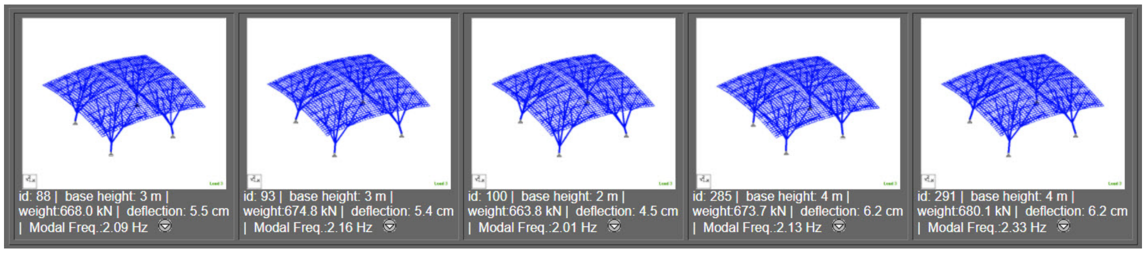
id: 93 | base height: 3 m | weight:674.8 kN | deflection: 5.4 cm | Modal Freq.:2.16 Hz

id: 285 | base height: 4 m | weight:673.7 kN | deflection: 6.2 cm | Modal Freq.:2.13 Hz

id: 88 | base height: 3 m | weight:668.0 kN | deflection: 5.5 cm | Modal Freq.:2.09 Hz

Figure 8: Pareto frontier of maximum modal frequency vs. minimum weight.

One can also click on any dot on the graph to see an image of that solution. Generally, in Pareto optimization the extreme ends of the graph are less useful since they show only the best for one or the other of the criteria. In viewing the results on the ParaGen website the range of results depicted can be restricted to show only the central part of the curve. Figure 9 shows the five best trade-off solutions for modal frequency vs. least weight sorted by least weight. Solution 285, shown in Figure 10, combines good stiffness and low weight as well as a 4 m base, which could be a practical advantage.


id: 88 | base height: 3 m | weight:668.0 kN | deflection: 5.5 cm | Modal Freq.:2.09 Hz

id: 93 | base height: 3 m | weight:674.8 kN | deflection: 5.4 cm | Modal Freq.:2.16 Hz

id: 100 | base height: 2 m | weight:663.8 kN | deflection: 4.5 cm | Modal Freq.:2.01 Hz

id: 285 | base height: 4 m | weight:673.7 kN | deflection: 6.2 cm | Modal Freq.:2.13 Hz

id: 291 | base height: 4 m | weight:680.1 kN | deflection: 6.2 cm | Modal Freq.:2.33 Hz

Figure 9: Five solutions taken from the central Pareto frontier
of maximum modal frequency vs. minimum weight.

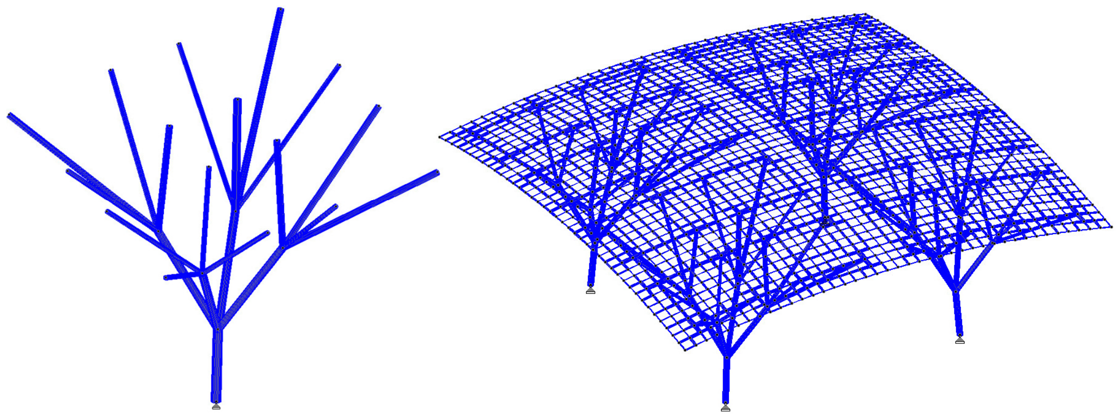


Figure 10. Detailed view of solution 285.

5. Conclusions

The study did show that Formian could be used effectively as a parametric form generator. It worked well together with the ParaGen method as a design exploration tool. Because of its simplicity and speed it performed well where 100's of iterations were required. Application of Formian in form exploration projects would be more convenient if it were possible to execute a Give statement within the same Formian script that is to generate the forms and execution of the Tignum function which yields the geometric outputs of expected cantles. Formian K seems to have a more user-friendly interface and operates more suitably regarding visualization of the generated form. However, its current edition restricts the number of generated elements to l000. Such a limitation will hopefully be improved in future editions.

References

[1] *Verzweigungen, Natürliche Konstruktionen - Leichtbau in Architektur und Natur*, Vol. 4, SFB 230, Universität Stuttgart, 1992.

[2] Schlaich J. and Schüller M., *Ingenieurbauführer Baden-Württemberg*. 1st edition, Bauwerk Verlag, Berlin (Germany), pp. 334-336.

[3] von Buelow P., A Geometric Comparison of Branching Structures in Tension and Compression versus Minimal Paths. in *Shell and Spatial Structures: Structural Architecture – Towards the future looking to the past*. Venice, Italy. 3-6 December 2007.

[4] Nooshin H., Disney P. and Champion O., Computer-Aided Processing of Polyhedric Configurations. in *Beyond the Cube: The Architecture of Space Frames and Polyhedra*. Gabriel J. (ed.), Wiley, 1997, pp. 343-384.

[5] von Buelow P., ParaGen: Performative Exploration of generative systems. *Journal of the International Association for Shell and Spatial Structures*, 2012; **53**; 271-284.

[6] Bentley Systems. STAAD.Pro. https://www.bentley.com/en/products/product-line/structural-analysis-software/staadpro

[7] Zhao Z., Liang B. and Liu H., Topology Establishment, Form Finding, and Mechanical Optimization of Branching Structures. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2018, **40**.

[8] Nooshin H. and Disney P., Formex Configuration Processing I, *International Journal of Space Structures*, 2000, **15**, no. 1.

[9] Nooshin H. and Disney P., Formex Configuration Processing II, *International Journal of Space Structures*, 2001, **16**, no. 1.

[10] Nooshin H. and Disney P., Formex Configuration Processing III, *International Journal of Space Structures*, 2002, **17**, no. 1.

[11] M. Moghimi, Formex Configuration Processing of Compound and Freeform Structures, University of Surrey, 2006.

[12] O. C. Champion, Polyhedric Configurations, University of Surrey, 1997.

[13] Nooshin H. and Samavati O., Exploring the Concept of Novation, *Asian Journal of Civil Engineering*, 2014, **15**, no. 6, pp. 869-895.

[14] Khodadadi A., Modern Lattice Domes Based on the Traditional Iranian Masonry Domes, *International Journal of Space Structures*, 2012, **27**, no. 4, pp. 231-245.

[15] Nooshin H., Kamyab R. and Ali S., Exploring Scallop Forms, *International Journal of Space Structures*, 2017, **32**, no. 2, pp. 84-111.

[16] von Buelow, P. Choosing parents to produce better preforming children: a comparison of selection methods used for evolutionary search. in *Interfaces: architecture . engineering . science*, (IASS 2017). Sept. 25-28, 2017. Hamburg, Germany.