SPECIAL ISSUE

STRUCTURAL MORPHOLOGY
Andrew Borgart and Jeroen Coenders

Vol. 53 (2012) No. 4
December n. 174
ISSN: 1028-365X
## contents

### Annual Letter from the President

*R. Motro*

199

### IASS Working Groups Reports

#### Report on ISCT 2012, Cologne 2012

*R. Harte*

201

### Technical Papers: Structural Morphology

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Morphology Special Issue</td>
<td>205</td>
</tr>
<tr>
<td><em>A. Borgart and J. Coenders</em></td>
<td></td>
</tr>
<tr>
<td>Design, Analysis and Realisation of Topology Optimized Concrete Structures</td>
<td>209</td>
</tr>
<tr>
<td><em>P. Dombernowsky and A. Søndergaard</em></td>
<td></td>
</tr>
<tr>
<td>Rigid-Foldable Cylinders and Cells</td>
<td>217</td>
</tr>
<tr>
<td><em>T. Tachi and K. Miura</em></td>
<td></td>
</tr>
<tr>
<td>Feasibility of the Universal Scissor Component (USC): Building a Full-Scale Deployable Dome</td>
<td>227</td>
</tr>
<tr>
<td><em>N. De Temmerman, L. Alegria Mira and A. Vergauwen</em></td>
<td></td>
</tr>
<tr>
<td>The Case for Subdivision Surfaces in Building Design</td>
<td>237</td>
</tr>
<tr>
<td><em>P. Shepherd and P. Richens</em></td>
<td></td>
</tr>
<tr>
<td>Shape Invariant Stiffness Optimization of Shell-Like Structures</td>
<td>247</td>
</tr>
<tr>
<td><em>M. Veltkamp, P. Basso and A. Lago</em></td>
<td></td>
</tr>
<tr>
<td>84 Ways to Manipulate a Membrane</td>
<td>257</td>
</tr>
<tr>
<td><em>A. Pronk and M. Dominicus</em></td>
<td></td>
</tr>
<tr>
<td>ParaGen: Performative Exploration of Generative Systems</td>
<td>271</td>
</tr>
<tr>
<td><em>P. von Buelow</em></td>
<td></td>
</tr>
<tr>
<td>Geometry-Based Understanding of Structures</td>
<td>285</td>
</tr>
<tr>
<td><em>T. Van Mele, L. Lachauer, M. Rippmann and P. Block</em></td>
<td></td>
</tr>
</tbody>
</table>

**COVER:** Figure from paper by T. Tachi and K. Miura
ParaGen: Performative Exploration of Generative Systems

Peter von Buelow
Assoc. Professor, University of Michigan, Ann Arbor, USA, pvbuelow@umich.edu

Editor's Note: Manuscript submitted 4 April 2012; revisions received 31 August 2012 and 3 November 2012; accepted 5 December. This paper is open for written discussion, which should be submitted to the IASS Secretariat no later than June 2013.

ABSTRACT

Spatial structures often embody generative systems. Both analog (physical modeling) as well as computational methods have been used to explore the range of design possibilities. Whereas many of the favored physical modeling techniques, such as soap films or catenary nets, inherently generate forms based on certain performative properties, many of the parametric form generating computational methods derive form based solely on geometry, detached from physical performance. ParaGen has been developed as a tool to explore parametric geometry based on aspects of performance. Within the cyclic structure of a genetic algorithm, it incorporates parametric geometry generation, simulation for performance evaluation, and the ability to sort and compare a wide range of solutions based on single or multiple objectives. The results can be visually compared by teams of designers across a graphic web interface which includes the potential for human interaction in parent selection and breeding of further designs. The result is a tool which allows the exploration of the generative design space based on performance as well as visual criteria.

Keywords: generative, parametric, evolutionary, performative, design exploration

1. INTRODUCTION

Designers of spatial structures have long benefitted from the use of modeling tools that generate form based on physical properties. Commonly referred to as “form finding” models, these include soap films used to explore membrane surfaces, catenary nets as inverted grid shells, or jersey cloth and plaster to model thin shell geometries [1]. Many of these form finding techniques do have computational counterparts, but most of the current form generation software (or associative parametric software) is based on the generation of form derived solely on the algorithmic description of the geometry. Although a wide range of possible forms are quickly generated, the designer is not informed by performative aspects of the geometries in making design decisions. Unlike the physical form finding models, form is separated from performance.

The method described in this paper, called ParaGen, combines currently available associative parametric software such as Generative Components [2], Digital Project [3] or Grasshopper [4], together with analysis tools such as finite element analysis (FEA), computational fluid dynamics (CFD) or any appropriate lighting, acoustic, thermal or behavior simulation software, to explore a range of well performing design solutions. Developed over the past three years at the University of Michigan Hydra Lab, ParaGen combines selected programs under the framework of a genetic algorithm (GA).
The entire cycle runs on a Windows cluster controlled over the internet by a web server. Figure 1 shows the cycle schematic.

Classic genetic algorithms follow a repeating cycle of selection, crossover (breeding), occasional mutation and evaluation (of the performance or fitness for selection)[5][6]. As seen in Figure 2. ParaGen incorporates and distributes these steps among a variable set of different programs, running in parallel on different machines, that each contribute to the overall cycle. Genetic algorithms (GAs) are part of the class of stochastic search methods called evolutionary computation (EC or EA for evolutionary algorithms) which include evolutionary strategies (ES), genetic programming (GP), interactive evolutionary computation (IEC), and others. In actual application EC encompasses a wide range of techniques and the boundaries between different categories are not distinct and often crossed. A good source which shows the variety of approaches and methods is the near 1000 page Handbook of Evolutionary Computation [7].

The visual images of the results are accessed through a web site which includes a complete searchable array of all solutions with both graphic and quantitative performance values (see Figure 8). By selecting preferred solutions on the screen, deeper levels of detail are immediately available – 3D geometry models and analysis models. The web interface allows the user to quickly scan through solutions which can be filtered and sorted by any of the analysis data as well as original geometry variables and parameters.

The ParaGen cycle follows five basic steps:

- **Select** - either with human interaction or by GA
- **Breed** - GA program on a web server
- **Generate** - the geometry using parametric software
- **Evaluate** - with simulation software, e.g. FEA
- **Store** - store the results in a database for selection – on web server

These steps are detailed in the next section, and examples of the process are shown subsequently.

2. THE PARAGEN CYCLE

As briefly outlined above, the ParaGen method brings together an array of programs running on both a web server and PC client machines. As is typical for genetic based search methods, the entire process is computationally intensive since many solutions need to be modeled and analyzed. Also, because the method makes use of commercial software on Windows based PCs, a single cycle can take from several minutes up to over an hour depending on the complexity of the model and the analyses being performed. In order to obtain faster run times, the ParaGen method can easily be run in parallel. Any number of client machines can attach at the same time to the web server, and process the individual solutions locally using parametric and analysis software run on the local client. The individual solutions are then uploaded to the web server, along with all of the collected analysis data, where they are stored in a SQL database along with image and data file references. Progress and results
of the exploration can be monitored by individuals or groups of designers simply by connecting to the web site where the solutions are displayed by the server. Because all of the solutions are saved in a relational database, designers can also filter, sort and retrieve the solutions by any of the parametric variables or by analysis data or by combinations of either. In this way the ParaGen method becomes even more useful as an exploration tool of the solution space.

2.1 Selection

The process begins with the selection of a pair of parent solutions from the database, which are then bred to produce a new child solution. Selection can be made either algorithmically by the program based on performance fitness values, or interactively by the designer using the web interface.

Algorithmic Selection

There are actually a few different selection algorithms depending on the stage of the run.

To jump start the beginning of a run, there needs to be an initial population before selection can begin. This initial population is normally generated randomly. The breeding program that runs on the server can produce random sets of input data. The random values are generated within maximum and minimum limits and at appropriate step size for the variables used in the parametric software. These boundary limits imposed by the parametric geometry modeler are the first level of constraints set for the ParaGen search. These constraints set in the breeding program merely reflect the constraints set in the parametric modeling software. Depending on which parametric modeler is used these constraints can have a wide range. In Digital Project, based on Catia, for example, these constraints include not just geometry but many multi-physics parameters as well. In the examples shown in this paper only geometry constraints were used. Finally, each random data set is sent to a PC client where it runs through the complete cycle, and is returned to the server with corresponding performance values.

Later in the run, after an initial population has been generated, selection is made based on better performance (better performing solutions have a higher chance of selection). This follows selection procedures commonly used in genetic algorithms. The difference in ParaGen is that the selection population is dynamic and includes all solutions generated to that point in the run. Since all solutions are maintained in a SQL database, the populations are assembled at the moment of breeding. Either the same population or two different populations are created for the selection of the two parents. In this case the population constraints (how the populations are defined) become the fitness function or objective for the run. For example, for a fitness function trying to find the solution of least weight, the population could be created with the first 30 solutions ranked by increasing weight. If the desire is to weight the pick toward the top, one could use the best of 3 random picks. This is similar in effect to penalty functions in traditional GA selection.

Also, preference can be given in the selection of one parent to recent solutions, and in the other parent to best performing solutions. Since the set of recent solutions can include both designer bred instances and occasional mutant solutions, this promotes potential infusion of new traits into subsequent generations. In effect, the population scheme used is similar to the “steady state” approach found in some evolutionary strategies [8]. The actual populations are produced dynamically during the run rather than being formed generation by generation. Nonetheless, the current best population will evolve and converge during the run as new solutions are added.

Another difference in the way ParaGen implements the selection algorithm is the fact that multiple fitness functions can be used concurrently or
progressively. Since the populations are created dynamically for each parent, the algorithm used can be chosen from a list of several different possibilities. For example there might be one algorithm that creates a population of least weight (as above) and another that creates a population of highest modal frequency, or another that creates a population of fewest members, etc. Either one or a combination of algorithms can be chosen for the selection of each parent. In this way the run can be set to explore different objectives concurrently while running in parallel on different machines. In the end all solution results are entered into the same database and viewed as one solution set. Thus a more comprehensive search is promoted which covers each objective as well as combinations.

**Interactive Selection**

There is also the possibility for designers to choose parents interactively through the web interface. Since an image of each solution as well as performance data is provided, designers are able to consider any quantitative or qualitative aspect in making a selection. In this way qualities such as aesthetics or the designer’s intuitive sense of the overall success of a solution can be taken into account. To breed two parent solutions, the designer simply selects the images from the web interface and clicks on the “breed” button. Figure 4 shows a selection ready to breed.

![Figure 4. Interactive breeding of two parent solutions by a designer.](image)

### 2.2 Breeding of New Solutions

The breeding procedure normally functions with two parents, but it can also generate a mutant variation of one parent, or a completely random parent with no prior input. There is also a check made to prevent duplication of existing children. This eliminates repeated processing of the same solution, and makes the entire ParaGen process more efficient. Since all solutions are maintained in the SQL database there is no possibility of a line dying out and also no need to hold duplicate solutions in the breeding pool. Breeding preference is determined directly by performance and can be weighted to better solutions as described above. Convergence can be determined not by numbers of like solutions, but by a lack of new better solutions.

**Two Parent Breeding**

By whichever way two parents are selected, either algorithmically or interactively, the breeding is the same. Here a variation of half uniform crossover (HUX) employed by Larry Eschelman in his CHC-GA is used [9]. Each variable value is treated as a real number allele on the genetic chromosome. After selecting which values will be crossed, care is taken to preserve the maximum, minimum and step range eventually required for each variable when applied to the parametric modeling software.

![Figure 5. Example of two parent breeding to get one child using half uniform crossover. Half of the variables are 'crossed', the other have are taken directly from parents.](image)
One Parent Breeding

If a single parent is selected, the breeding procedure produces a mutation based on that parent. A similar procedure to HUX is followed except the chosen alleles are replaced by randomly generated values rather than crossed with another parent. Mutants can be generated interactively by the designer, but also occasionally occur with algorithmic selection.

No Parent Breeding

In the case where no parents are selected, for example at the start of a run, all of the values are randomly generated. Nonetheless, even in this case, random values are constrained to fit within the range required for each variable used in the parametric modeler.

2.3 Developing the Geometries

Once a new child data set has been created, it is downloaded from the web server to a client PC which contains the parametric modeling and analysis software. In the examples which follow, Generative Components (CG) by Bentley Systems [2] is used to develop the geometry. As noted earlier, ParaGen is conceived to be able to make use of any commercial software or combination of software packages to run the various steps of the process. The values for the parametric variables (the GA child chromosome) are passed to GC in Excel format.

Figure 6. Geometry generation using parametric software (Generative Components, Bentley Systems).

The script used by the parametric modeler is of course fundamental to the ParaGen exploration since the parameters and constraints of the possible geometries that can be reached in the exploration are coded in the parametric script. Therefore, for a thorough exploration of the design space, the script needs to be written to allow the generation of a wide range of different possible solutions. At the same time it is a distinct advantage offered by ParaGen, to be able to explore large complex systems through a relatively few number of parametric variables. Geometric constraints are easily established in the parametric model, making the space open to the search more intentional. It is of course also possible to include physical constraints like minimal surfaces or catenary curves in the parametric model. The type or number of constraints is only limited by the specific parametric modeler being utilized.

Once a solution has been generated by the parametric modeler, the geometry needs to be exported in a format that can be read into the specific analysis software being used. It is also useful for the later qualitative evaluation by the designer, to capture descriptive views of the geometry which can eventually be displayed on the web interface for comparison with other solutions.

2.4 Evaluation of the Solutions

Depending on the complexity of the problem, there may be one or several evaluation steps. In this phase the solution is analyzed in some simulation software, and quantitative data is collected that will both aid in the decision process as well as direct further exploration through the fitness function set in the GA. Often the simulation can also offer visual cues to the performance. For example a finite element analysis that determines system stiffness or internal stress levels may also size members and provide a VRML or other 3D image that includes actual member sizes. Images of vibration modes, lighting conditions, acoustic reflections, heat transfer are all examples of visual data that can aid in the rapid understanding and comparison of different solutions by the designers. By saving these qualitative images along with the quantitative values found in the analysis, the designer will later be able to engage the problem at a higher level, and consider both objective quantities and subjective qualities when comparing a range of solutions with the web browser.
After all of the simulations have been run and the evaluation data has been collected, both the qualitative images (or animations or sound files) as well as a file containing all of the quantitative values, are uploaded to the web server.

2.5 Ranking the Solutions

Once all of the files and data belonging to a solution have been uploaded to the server, the data is placed in a SQL database with an identification tag that links all of the associated image and data files. The database is accessed through a PHP program which associates all of the data through the graphic interface of the ParaGen web page. Using the web page interface, the designer can both filter and sort all of the collected solutions by any single, or any combination, of either the parametric input variables and/or performance values collected from the simulation software. This effectively allows constraints be set dynamically and interactively in the form of search filters applied to the SQL database. Upper and lower limits can be set on any of the parametric or performance data in order to filter and control which solutions are displayed. Because at that point all of the solutions are contained in the SQL database, altering the constraining filter settings gives near instantaneous response, making the process ideal for interactive control. Once the solutions have been generated and stored the solution space can be truly explored through the use of the SQL filters.

As shown in Figure 8, the web page displays both a small image and key performance values for each solution. By clicking on any solution, a larger image, along with all data values, is displayed, as shown in Figure 9. In this window, links to all other associated files are also provided.

Since only one instance of each solution is entered in the database, the widest possible array of different solutions is displayed. Although duplicates are prevented, convergence can still be recognized by the slowed rate of improvement toward the fitness function (the objective), or by the visual similarity shown in a sort by best performing values.
Figure 10. Web interface showing multi-objective plotting function. Clicking the dots, gives the image.

Usually, after the procedure has generated a significant number of solutions based on the algorithmic selection, the exploration of the solution space using the filters and sorting functions will suggest possible design directions which the designer can further explore with interactive selective breeding or alterations to the algorithmic fitness function.

It is also possible to produce Pareto front plots by choosing pairs of independent variables – see Figure 10.

With the selection of a new pair of parent solutions for breeding, the procedure starts over with a new cycle.

3. EXAMPLE RUNS

To demonstrate the use of ParaGen as a design exploration tool, three examples are shown below: a tower, a bridge and a dome. Several other examples have been published as well, and the interested reader is referred to recent papers in which specific details are given [10] [11] [12].

3.1. Shukhov Water Tower

The tower example is taken from one of Vladimir Shukhov’s early hyperboloid tower designs. Originally designed as a water tower for the All-Russian Exposition in Nizhny Novgorod in 1896, it was afterwards relocated to Polibino, Lipetsk Oblast, where it still stands (see Figure 11 – left). Shukhov’s tower is 37 m high and carries a water tank and observation deck.

A parametric model based on Shukhov’s design was constructed in Generative Components (GC) using tree variable parameters: base diameter, number of diagonal members and the twist angle (Figure 11 – right). This model was able to describe the geometry of Shukhov’s tower as well as a large range of neighboring variations. A range of the initial random geometries is shown in Figure 12.

Figure 11. LEFT: Original water tower designed in 1896 by Vladimir Shukhov. (photo by Arssenev) RIGHT: the constants and variables used in the parametric model.

Figure 12. Initial random solutions generated by ParaGen.

The geometries generated in GC were then analyzed in STAAD.Pro based on a 145 km/hr (90 mph) wind speed plus the weight of the water tank.
container, 2900 kN (652 Kips). Members were sized based on the current US steel code (AISC ASD) using steel pipe sections. The dead weight of the structure was also included in the load.

![Figure 13. Comparison of least members (LEFT) with lightest weight (RIGHT) solutions.](image)

Based on the fitness function of least weight, the GA reached a plateau of 30 tonne (33 tons) after 300 unique solutions. Figure 13 shows the solution with the least members (left) next to the lightest solution found (right). The lightest case uses 10108 members, compared to the original Shukhov design with only 2920 members. Members were defined as being formed between the cross points. Ring members were also included. In achieving lighter weight, the GA found solutions with many, short members. This gave a shorter buckling length, and therefore smaller sections, but a greater number of members. Of course the greater number of members would also mean more joints and labor to construct.

![Figure 14. ParaGen display of solutions filtered for members < 1500 and sorted by ascending weight.](image)

Multiple objectives can be explored in two different ways with ParaGen. First, one objective (or actually any number) can be filtered, and then the resulting group can be sorted by the other objective. For example, Figure 14 shows solutions filtered for the number of members less than 1500, and then sorted by weight. This shows trade-off solutions, some similar to the Shukhov design.

The other way multi-objectives can be approached is by graphing the values of two objectives against each other. In the ParaGen plotting function, any pair of parameters or evaluated values can be set against each other. The number of solutions to plot can be constrained by filters as in the first approach (e.g. only plot values less than or greater than some limit). Also the order of the plot scales can be controlled (ascending or descending).
Figure 15. A plot of two objectives – weight and the number of members. Clicking points gives images.

Figure 15 shows a plot of weight vs. members. By clicking on the plotted dots, the tower thumbnails are displayed for comparison. In this way trade-offs can be made along the Pareto front.

Towers very close to the original Shukhov geometry also appeared in the solution set. Figure 16 shows a graph of deflection vs. weight. Here a solution having the same diagonal count and dimensions as the Shukhov tower is compared to others. The Shukhov tower lies close to the Pareto front, having both low weight and low displacement, in other words, a good solution.

Figure 16. A plot showing displacement (cm) vs. weight (tonne). The solution marked (id 20) is very close to the original Shukhov geometry.

3.2 Truss Road Bridge

The second example is a familiar optimization subject, a road bridge truss. The overall dimensions are based on a 36 m (118 FT) span, wrought iron bridge dating from 1879, shown in Figure 17. The parametric model used forced symmetry at the center of the 8 panel, bridge, but cross bracing and the geometry of the truss was allowed to vary. Members were not limited to tension only. The fitness function was set to minimize weight and maximize natural modal frequency (detailed below).

Figure 17. Original Pratt Truss bridge design.

The loading was based on a 20 ton moving truck (AASHTO type H-20). All members were designed by the US steel code (AISC ASD) to determine the total weight of the structure. Deflection and modal frequency data was also saved. Figure 18 shows some of the solutions with the least weight of 1.50 tonne (3.3 kips) for one truss. It is interesting to see that all of the lightest trusses share the same topology which is actually a Warren truss with verticals.

Figure 18. A selection of the lightest weight trusses.

In the interest of comparison with the original truss, a filter was used to select the lightest solutions constrained to 31 members. In Figure 19 it can be seen that no single topology dominated in this case.
In fact only two, numbers 998 and 418 share the same topology. Truss 712 (Figure 19, lower right) has the double cross bracing in the center two panels as did the original truss, but the other diagonals follow a Warren rather than a Pratt pattern. Overall this set was about 10% heavier than the lightest weight set.

A plot was also made to explore both weight and natural frequency together – Figure 20. Because both of these objectives were important, the fitness function used in the GA considered both. Alternately parents were selected in one of three ways:

1. one parent each from a population of least weight, and a population of highest natural frequency
2. one parent from a population of least weight, and one from a population of most current
3. one parent from a population of highest natural frequency, and one from a population of most current.

Selection methods 2 and 3 allow for the inclusion of possible mutations in the breeding, and work to prevent early convergence to the exclusion of a broader exploration. In a traditional GA this type of selection would be impossible, but ParaGen allows this by using dynamic breeding populations which are instantly assembled from the SQL database of all solutions thus far. The GA can be set to randomly choose from any number of fitness functions or breeding schemes concurrently.

In Figure 20, a convergence is readily seen in the cluster of results with low weight and high natural frequency (upper left corner). By inspecting these solutions visually (e.g. as in Figure 18) the convergence is confirmed. The sensitivity to changes in geometry can also easily be observed. Since a filtered sort of solutions is instantly shown on the screen (values are simply pulled from the database not recalculated) it is easy to scroll through a relatively large array to get a graphic depiction of the sensitivity of the structure to different parameters. Since the SQL post-processing offers such easy and such variable access to the solution space, the function of the GA is oriented more toward exploration of the space (e.g. with changing fitness functions) rather than a narrowly directed search based on a single criteria. The interest in ParaGen is not just a single best solution as pre-imagined when defining a fitness function, but also the other neighboring solutions that might have slightly lower performance, but with markedly different visual or other characteristics, that might make them a more attractive choice.

![Figure 20. Plot comparing natural frequency and weight of the truss. Selecting dots shows the solutions. The cluster of dots indicates the convergence of the run.](image)

![Figure 21. An example of a double-layer dome.](image)
3.3 Geodesic Domes

The last example is also familiar and also more complex that the preceding two. Designers have long been fascinated with the polyhedral geometries of what is generally called the geodesic dome – Figure 21.

“From Pythagoras to Alexander Gram Bell” [13] these forms have found numerous application in architecture. The parametric model used in this study includes a range of the double shell domes.

The parametric model used 5 parameters:

- **Primitive Type** – outer shell (inner is dual)
  - tetrahedron
  - octahedron
  - icosahedron
- **Power** - number of subdivision generations
- **Frequency** – subdivisions of primitive face
- **Trim** – cut portion from a sphere
- **Radius Multiplier** – distance between layers

The number of sub-triangles per original primitive triangle is given by frequency^2*power. With the ranges and constraints assigned to each parametric variable there are 418500 possible combinations. Figure 22 shows a small selection from the range of possibilities.

For the purpose of comparison, the largest diameter of the dome was held constant at 15.2 m (50 ft). The other parameters of the domes, were allowed to vary. This gives a wide range in both pattern complexity and member count.

Each solution was analyzed in STAAD.Pro to determine forces and member sizes based on a snow load of 2.0 kPa (42 psf) shown in Figure 23. Members were sized based on the US steel code (AISC ASD), and the total weight of the structure was determined. The natural (modal) frequency based on the final member sizes was also determined. The data recorded for each solution included: numbers of members, nodes and plates, number of different member sizes and lengths, number of supports, surface area, weight, modal frequency, total member length, maximum member length, and average member length. These data were used to both direct the search (multiple fitness functions) and to explore the database after the run.

After an initial number of randomly generated solutions were produced, breeding parents were selected from the complete database using filters. How the filters are chosen effectively defines the fitness function. When the parents are bred the children are checked with other solutions already present in the database to ensure their uniqueness. The filters can be modified over the course of the run to avoid impractical directions (e.g. too few members or plates to make a dome), and focus on more applicable solutions. Again, all of the solutions remain present in the database and can be reviewed at anytime.
In observing the results, several tendencies can be seen. In general strategies which reduce the overall number of members are generally lighter. The reduced member count is accompanied by fewer plates. Because the loading is based on snow on the plates, this reduces the load for the structure. But, as seen in Figure 24, solutions with a minimal number of members do not tend to satisfy the spatial qualities of a dome or the architectural criteria of space covering. Therefore, a filter was imposed which constrained the minimum number of members. Figure 25 shows a selection of light weight domes having at least 120 members.

![Figure 24](image1.png)

**Figure 24.** A light weight yet impractical solution with only 18 members and one plate weighing 1.39 tonne.

Although the plates were not designed in this example, the maximum dimensions of the covering plates would also be a consideration. Limiting the edge length to 2 m (6.5 ft) reveals a different set of solutions. Figure 26 shows a selection limited to 2 m edge length and sorted again by least weight.

![Figure 26](image2.png)

**Figure 26.** A selection of solutions filtered for less than 2 m panel edge length and sorted by least weight. The weights range from 6.0 tonne (upper) to 11.3 tonne (lower).

Further correlations between the parametric geometry variables and the generated domes can also be recognized through the use of the graphing function. For example by plotting least number of different length members vs. each of the geometry variables it was determined that domes based on a Primitive Type = icosahedron + dual and Power = 1 yielded the least number of different length members (6 different lengths). Figure 27 shows a set of domes having 6 to 7 different length members and weight ranging from 4.1 tonne to 10.9 tonne. This set then is based on multi-criteria, and gives further insight into the selection of the geometry variables.

4. CONCLUSIONS

This paper has given an overview of the concept and functionality of ParaGen, a conceptual design tool which allows the exploration of parametric geometries based on performance criteria. The setup which includes a web server and client PCs running various Windows programs, was discussed, and some of the options were illustrated. The
flexibility of ParaGen to run either as a pre-programmed optimization tool or as an interactive exploration tool was described.

Figure 27. Solutions constrained to only 6 or 7 different member lengths.

The examples here focus on structural performance, although that is by no means a limitation of the system, but merely the main interest of the author. In cooperation with other researchers, the tool has been used to explore thermal, solar, acoustic and other performance parameters [10] [11] [12].

From the examples shown, it can be seen that ParaGen focuses more on the exploration of different well performing solutions, rather than on finding a single ‘best’ solution. The ability to assess multiple objectives was also demonstrated by use of the interactive plotting feature or multiple fitness functions.

The process is computationally intensive, but by using parallel machines reasonable processing times are achievable. Since the ParaGen approach avoids repeating duplicate solutions, the method is more efficient than techniques based on conventional genetic algorithms. Also, search display times during post exploration of the solution database are very fast (a few seconds), which makes human interaction reasonably productive.

Further development of the system is currently underway with cooperative input of different users.

5. ACKNOWLEDGEMENTS

The author would like to thank Taubman College at the University of Michigan for support of the ParaGen project. Also the continued interest and coding support with Generative Components by Michela Turin at TU Delft has been key in moving the project forward. Theodore Hall, at the Michigan 3D Lab, developed the parametric dome model and Andreas Falk has provided much useful support and suggestions.

6. REFERENCES


