Proceedings of the IASS Annual Symposium 2019 – Structural Membranes 2019
Form and Force
7 – 10 October 2019, Barcelona, Spain
C. Lázaro, K.-U. Bletzinger, E. Oñate (eds.)

The combination of SQL database queries and stochastic search methods used to explore generative design solutions

Peter VON BUELOW*

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

Abstract

ParaGen is a method that has been developed over the past several years at the University of Michigan's Taubman College. ParaGen combines both a GA and a SQL database for solution exploration and performance optimization. Multi-performance optimization has been successfully demonstrated on numerous problems using stochastic algorithms like GAs. But for many problems which include visual aspects of form, like architectural design, quantitative performance values alone are not the only criteria needed to search for a good design. In architecture as in other design fields with a visual component, aesthetics, human comfort and other qualitative aspects need to be considered. By allowing designers to make visual comparisons of solution forms that have been found through performance driven search, ParaGen is able to respond to both quantitative and qualitative aspects of design. In this paper an example is given where a combination of quantitative performance criteria along with qualitative criteria based on the designer's visual selection are used to search for better performing structural forms.

Keywords: performance optimization, generative design, parametric, data-driven optimization, genetic algorithms

1. Introduction

The exploration of a range of possible solutions as a means of searching for new solutions has long been seen as a valuable tool in the design process. Numerous studies in the latter half of the 20th century attempt to describe or even formularize this process. Some form of exploration is generally common to these methods. Osborn describes "ideation, piling up alternatives by way of ideas"[1]. Koberg and Bagnall describe ideation as a way to "generate options" and "search out all possible ways of realizing the major goals."[2] Most stochastic optimization techniques, such as genetic algorithms (GAs), include a search operator which essentially explores the solution space in search of good solutions. Unfortunately most GAs do not maintain a history of the search progress and only have a 'memory' of the current generation. So although these techniques do "generate options" the do not really "pile up alternatives" since they lack any recorded memory of the search. Of course one solution is to simply store the solutions as they are generated, but a more useful approach is to actually incorporate a database with a structured query language (SQL) into the search process so that solutions are not only saved, but more importantly can later be explored interactively. This is much closer to Osborn's concept of ideation.

The ParaGen method links both a SQL database and a GA to form a tool which can both search using performance criteria and explore the broader scope of possible solutions [3]. It is intended for use in the early stages of design development when form exploration is desired. But at the same time ParaGen is able to bring a higher level of performance information into consideration which allows for more effective form exploration at the early stages of design development.

2. The ParaGen method

ParaGen is a method rather than a specific piece of software. It combines four basic components:

1. A genetic algorithm to search for new solutions
2. A parametric form generator to realize the solutions
3. Analysis software to find performance characteristics
4. A SQL database to store solution characteristics and performance data

In addition, descriptive images are saved and linked to the data. This enables the solutions to be searched using both quantitative performance values and qualitative images. It is also possible to guide the GA using either a fitness function based on SQL queries of the performance values or by using breeding populations selected based on user (visual) preference. The procedure is repeated in cycles until good solutions are found. Figure 1 shows the procedure.

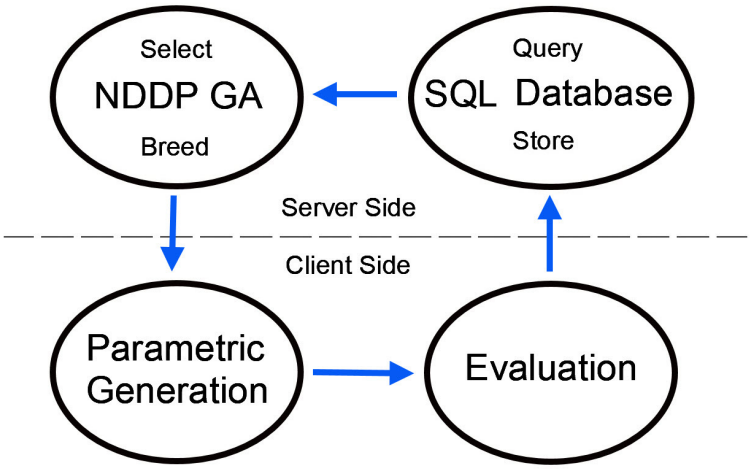

Figure 1: An outline of the ParaGen method

The ParaGen method is conceived to run in a parallel processing environment using a single web server and some number of clients in parallel. In this way longer evaluations such as daylighting analysis or complex wind analysis can run in parallel for greater efficiency. ParaGen is based on a Non-Destructive Dynamic Population GA (NDDP GA) [4]. Rather than traditional generational structures which eliminate less fit solutions from the breeding population, ParaGen stores all solutions in a database and dynamically creates a breeding population based on a SQL query at the moment of breeding. Parents from this population are then bred using half-uniform crossover (HUX) [5]. In the first step, the NDDP GA produces a new solution either by selection and breeding or by selection and mutation or by random generation (usually to start the process). This new 'child' is then downloaded in the form of a CSV file to a client machine where in step 2 it is read into some parametric software which produces the geometry based on the values supplied by the CSV file. In the third step, this geometry is analyzed for whatever performance characteristics are relevant. This is often possible within the parametric software using

plugins, or it may require the geometry to be passed (usually in DXF format) to some specialized software. In either case the resulting performance values along with descriptive images are uploaded in the fourth step to the SQL database on the server. The GA then accesses the database in selecting parents to breed the next child and the cycle continues [6].

Initially, the GA code generates some number of random solutions to populate the database. Once the data and images for solutions are stored in the database, designers can browse through them using the web interface and SQL queries. Both the performance data and the images are then linked, enabling both quantitate and qualitative information to be viewed together. In an NDDP GA the breeding population is

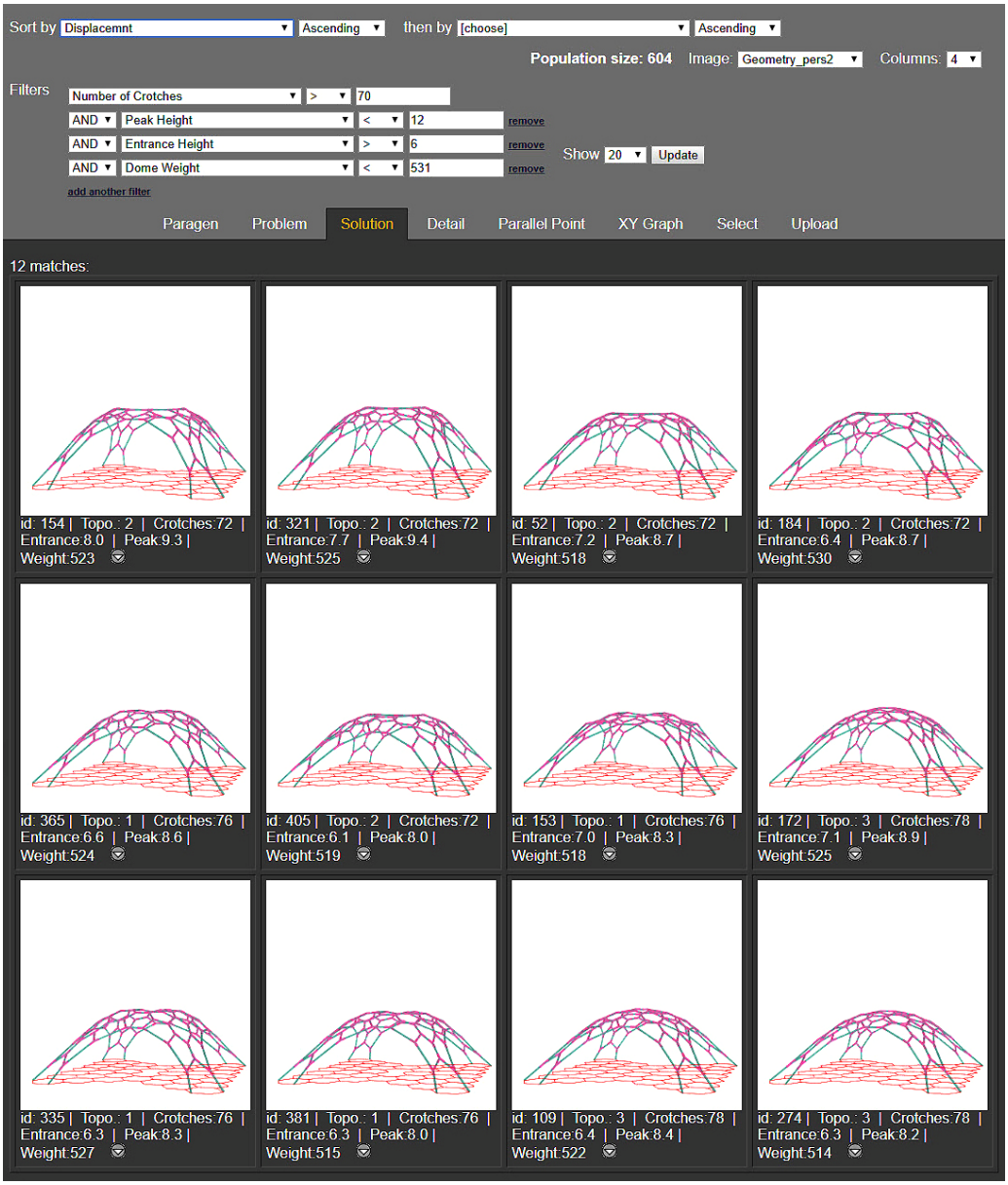

Figure 2: A SQL query defined in the ParaGen interface with the resulting set of solutions

selected from the database using a SQL query each time a new child is bred. The query contains both soft (flexible) design constraints as well as optimization fitness criteria. In the ParaGen method, constraints can be seen as having two forms – hard and soft. The hard constraints are coded into the definition of the parametric model and cannot be changed without altering that base model definition. The soft constraints, on the other hand, are used to focus or limit the scope of the design space being searched. In other words the full design space is defined by the parametric model, but the search space used for breeding is restricted to a part of the full design space limited by the SQL query. The intent of limiting the breeding population is to focus on a particular (optimal) area of the design space. Initially, by using either randomly generated solutions or by systematically scanning the design space, the search space equals the full design space. But the goal is to search that space for the "better" solutions as defined by the SQL query. This search is made flexible in order to follow the exploratory nature of early design by altering the SQL query. In this way the designer can steer the search procedure to explore different areas of the design space. SQL queries are designed to be efficient in searching a database for certain desired items based on multiple attributes. Figure 2 shows an example of a SQL query given through the ParaGen website and the resulting set of solutions.

3. Selection using SQL

ParaGen is able to make use of SQL queries in different ways is searching the design space. Three ways which are illustrated below are:

1. Quantitative values for geometry and performance
2. Conflicting performance values using Pareto sets
3. Qualitative considerations combined with other values

3.1. Selection based on quantitative values

In the example shown in Figure 2, designs for an irregular wooden reticulated dome are searched based on five criteria or constraints:

1. the number of joints is greater than 70
2. the peak center height is less than 3.7 m. (12 ft.)
3. the entrance (arch) height is greater than 1.8 m. (6 ft.)
4. the material weight is less than 240 kg. (530 lbs.)
5. the deflection is minimized

The first four constraints can be seen listed under "Filters" and the last criterion which minimizes displacement is show at the top as "Sort by…Ascending". The total selection is then limited in number (in this case 12). This in effect produces a breeding population from which a pair of parents are randomly chosen and bred in the GA. In this way the SQL query becomes the fitness function for the GA. The size of the breeding population is controlled by the query limit parameter, set in this example to 12. So after meeting the first four constraints, the 12 with the least deflection (sort by displacement ascending) become the breeding population. As new solutions are found that meet the first four constraints and have less deflection, they enter into the breeding population. It would also be possible not to set the population limit criterion (12 in this case) or perhaps replace it with a set constraint like the other four – for example require deflection less than some certain amount. In such a case the breeding population would gradually increase but could be restricted (re-focused) periodically by adjusting any of the five constraints or add additional constraints. Unlike generational GAs that build a breeding population through successive generations, the fitness function for the NDDP GA can be changed at any time without restarting the

whole process. Because all solutions are maintained in the SQL database, changing the GA fitness function merely shifts the focus of search in the database.

3.2. Selection based on Pareto sets

Architectural design problems often contain conflicting parameters. For example one might want to increase span to give more column-free open space, but at the same time reduce structural weight in order to reduce cost. These would be conflicting parameters. A common solution is to seek some trade-off solution whereby the two (or could be more) parameters are adjusted to give a balanced performance. GAs have for some time used Pareto sets as the breeding populations to search for the best (non-dominated) trade-off solutions [7]. ParaGen also uses Pareto sets to explore conflicting multi-objective goals [6]. Figure 3 shows a graph of a set of solutions from the wood dome example. The Pareto frontier (level 1) is highlighted with the red dots. By clicking on the dots, images of the solutions are shown. This can quickly show the relationship between the form and the graphed quantities and allow the designer to choose among the options based on visual qualities.

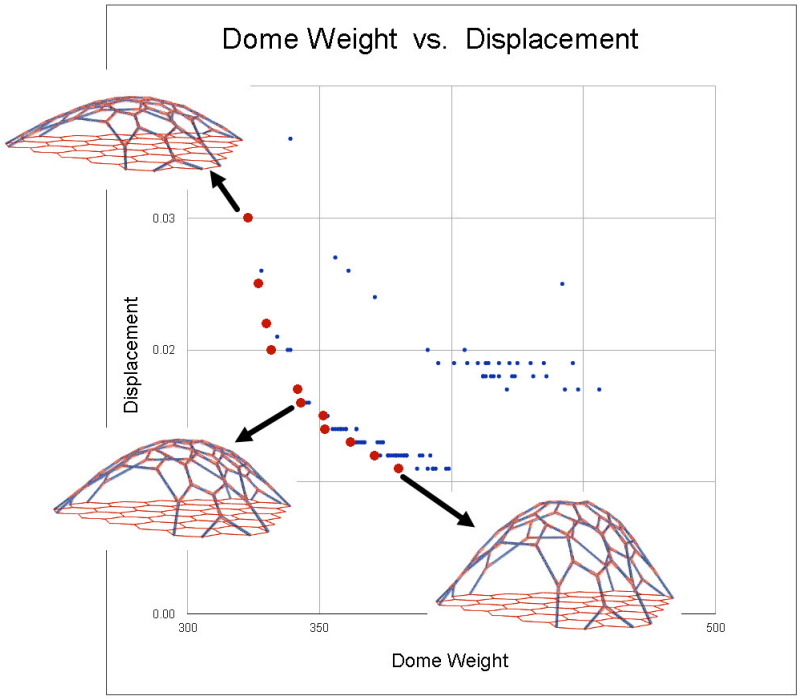

Figure 3: A graph showing the first level Pareto set. Selected images show the range of form.

The database can also store the Pareto level number for each solution. From any pair of conflicting parameters the Pareto frontier as shown in Figure 3 can be calculated and stored. The frontier as shown is considered level 1. The next level behind this (one level less optimal) is level 2. Behind level 2 is level 3 and so on. Having more than one level adds depth to the set being considered either for breeding or simply viewing more options. Any number of sets of parameters can be chosen for Pareto calculations. Each Pareto set is listed in the database as a column (or field) which holds the Pareto level number for each solution. Pareto sets can also be based on more than two parameters although beyond three is hard to visualize or graph. Again, by using the SQL filters on the solution array page, the different visual images

of the Pareto solutions can be scrolled through. Figure 4 shows the full set of images from the Pareto level 1 set shown in Figure 3.

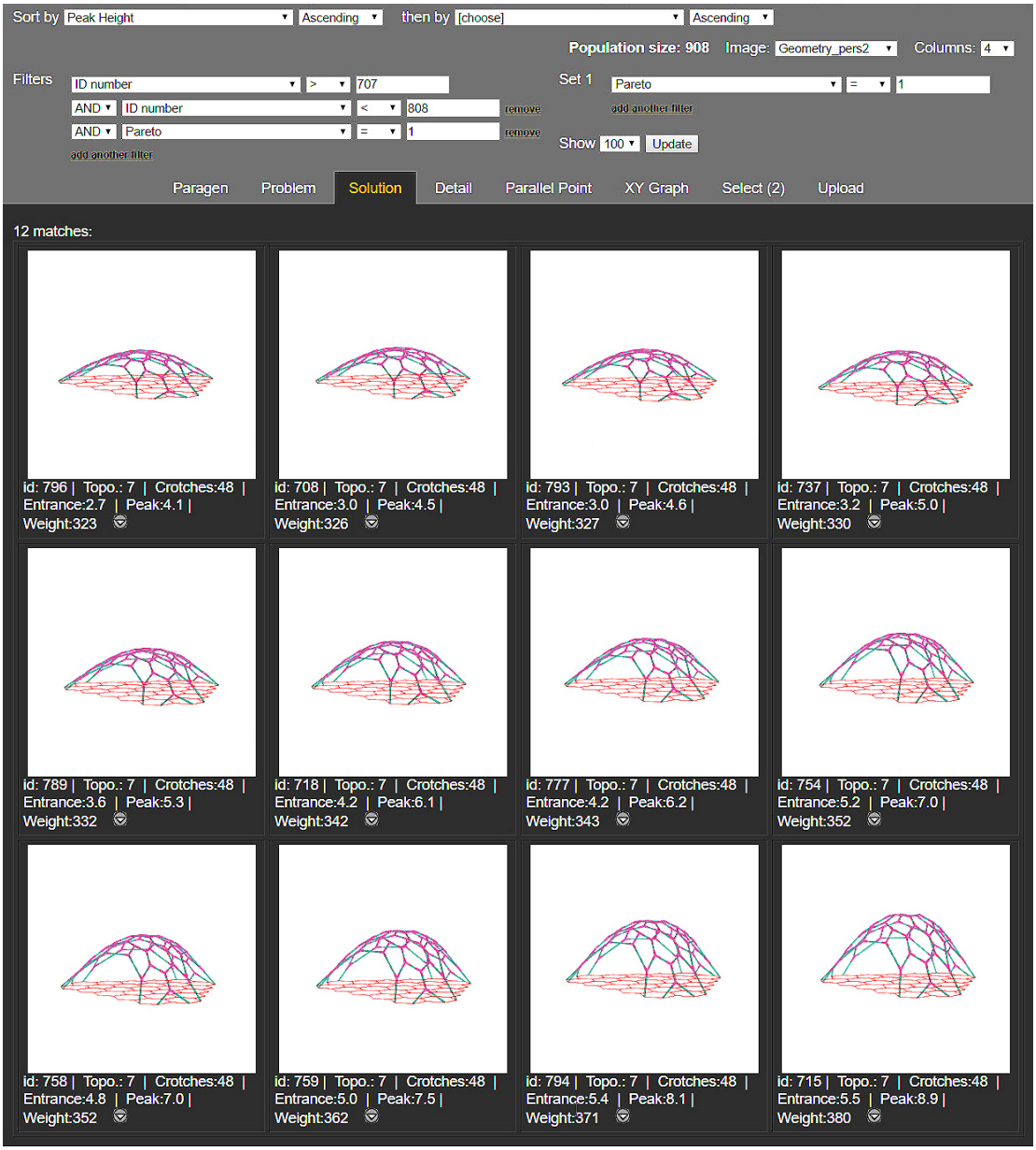

Figure 4: Solutions from the Pareto frontier shown in Figure 3.

3.3. Selection based on user selection

The breeding population may also be selected from the database based on qualitative visual criteria of the designer. Any desired combination of filters and sorts can be chosen to allow the designer to visually explore the solutions in the database. Each solution has a unique identifying number and by specifying a list of these solution numbers chosen by the designer, the GA can use this list as the breeding population.

As an example, using a full set of Pareto level 1 solutions from the wood dome problem (a total of 68 solutions), 20 solutions were selected by the designer based on visual criteria of aesthetics and usability. These 20 solutions were then taken as a breeding set to produce 100 further solutions. From these new solutions the dome shown in Figure 5 was chosen as meeting the aesthetic interest of the designer as well as having fairly good performance.

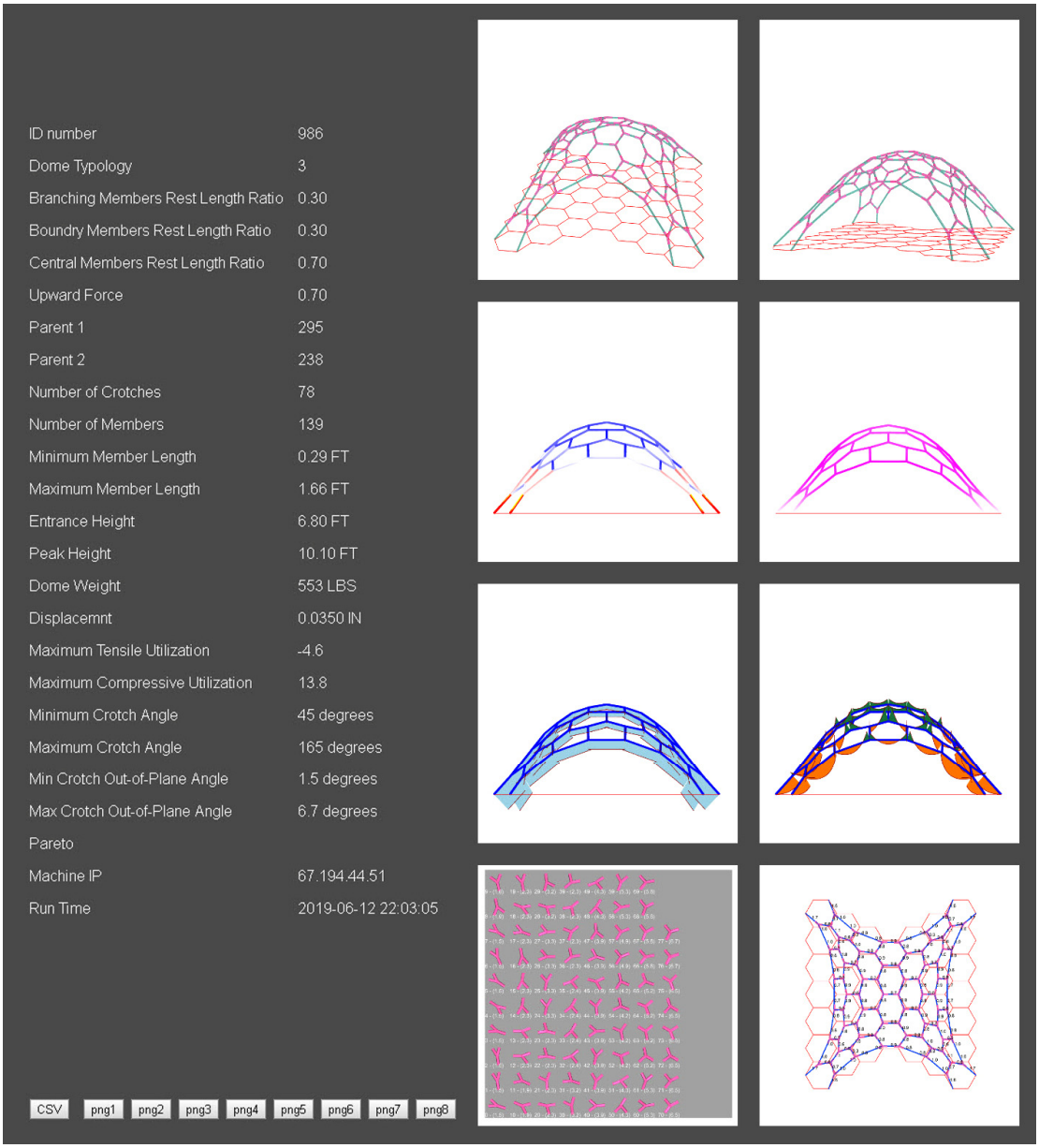

Figure 5: A chosen design generated from the designer selected population. ParaGen can show an overview page of any selected solution which includes all images along with all generating and performance data.

4. Conclusion

Through a simple design example of a reticulated wood dome, some advantages were demonstrated in the use of a SQL database together with a genetic algorithm to search for good solutions. ParaGen, developed at the University of Michigan, combines both an optimizing GA and a SQL database for solution exploration in the early phases of design. ParaGen uses SQL queries to both focus the GA search direction and to provide the designer with post processing, interactive search of the generated solutions. Because a variety of images can also be linked to the stored data, visual and qualitative aspects of the design can be considered in the evaluation process. Using selections made from these visual images, it is also possible to further direct the search for new solutions. In this way ParaGen is able to include both quantitative and qualitative aspects of a design in the exploration process. The result is a more complete consideration of the problem and ultimately a better solution.

Acknowledgements

The author wishes to thank the University of Michigan Taubman College for support over the years of the ParaGen project and the Hydra Lab for parallel computing.

References

[1] A. Osborn, *Applied Imagination – Principles and Procedures of Creative Writing*, Scribner's, Oxford, England, 1953.

[2] D. Koberg and J. Bagnall, *The Universal Traveler,* William Kaufmann, Inc., Los Altos, USA, 1972.

[3] P. von Buelow, "ParaGen: Performative Exploration of Generative Systems" in *Journal of the International Association for Shell and Spatial Structures*. Vol. 53 (2012) No. 4, pp. 271-284.

[4] P. von Buelow, . "Techniques for More Productive Genetic Design: Exploration with GAs using Non-Destructive Dynamic Populations" in *Adaptive Architecture* eds. P. Beesley, O. Khan and M. Stacey. Proceedings of the Association for Computer-Aided Design in Architecture 2013 International Conference. October 24-26, 2013. Cambridge, Ontario, Canada.

[5] L. Eshelman, "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. pp. 265-283

[6] P. von Buelow, "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.

[7] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. Wiley, 2001.