Genetically Enhanced Parametric Design in the Exploration of Architectural Solutions

P. von Buelow
University of Michigan, Ann Arbor, USA

ABSTRACT: ParaGen is a method for exploring solution options in the early phases of design. The method is intended to aid the designer by exposing solutions which perform well against an array of performance criteria. ParaGen makes use of a variety of commercial simulation software to determine the desired performance parameters in relevant areas such as structures, lighting, acoustics or energy. Based on the values of these parameters a multi-objective search is carried out using a Non-Destructive Dynamic Population Genetic Algorithm (NDDP GA). In this approach all solutions, including relevant performance parameters as well as geometric data are retained in a relational database both as quantitative values and as qualitative images. SQL queries are used both by the NDDP GA to guide the search and by the designer directly to explore the solutions. A number of tools are also included to aid the designer in finding well performing solutions. These include graphing functions to find Pareto sets of conflicting performance criteria and parallel coordinate graphs to examine the interrelation of model parameters and performance values.

1 INTRODUCTION

1.1 *Application to Early Design*

In the development of any design, the initial form giving decisions have the greatest impact on the direction of continued development and the ultimate outcome of the solution. This is particularly true in architectural problems where many, and often conflicting, factors influence the chosen form. ParaGen is a method developed at the University of Michigan to help designers navigate the early phases of design by both exploring the range of the design space (the multiplicity of solutions) and also determining the set of well performing solutions for each performance criteria and for combinations of criteria (Pareto sets).

ParaGen is particularly well matched for the characteristics common to early phases of a design problem. First it is fundamentally explorative. Unlike deterministic optimization techniques which are more appropriate to later stages of problem solving, ParaGen makes use of a Non-Destructive Dynamic Population Genetic Algorithm (NDDP GA) to build a database of solutions. Like any GA, ParaGen is always working with populations of solutions rather than targeting a single best solution. Because the designer is always working with sets of solutions, there is less danger of design fixation as when shown a single “best” solution. Second, it functions in cycles. This also mirrors the early design process. Third, through the GA mechanics of crossover and mutation there is an element of chance or serendipitous discovery. Finally, it is also possible to describe the fitness function independently from how the performance values are obtained. Also the use of the NDDP GA allows multiple, conflicting fitness values to be used either together, in parallel (on parallel machines) or in serial (progressively changing the function). Because the populations for the GA are created dynamically from the entire database of solutions, the creation of each child solution occurs independent of the generations which immediately proceed it. This allows for greater latitude in exploration than in traditional GAs.

1.2 *Overview of the ParaGen procedure*

The ParaGen method makes use of parametric modeler to describe and access a range of geometries which respond to a given architectural problem. The values of the parametric variables are chosen using a Non-Destructive Dynamic Population Genetic Algorithm (NDDP GA). The dynamic populations are drawn from a database of solutions constructed by ParaGen as the run proceeds. The database contains an array of performance data gathered from each parametric solution by using a suite of simulation software chosen to model key aspects of the design – e.g. structural performance, energy use, lighting, acoustics, etc. Thus an integrative exploration is possible by using multi-objective fitness functions based on the various performances.

ParaGen is non-destructive in that it maintains all geometric and performance data as well as links to chosen images of both geometry and performance (e.g. deflection plots or lighting levels) in a relational database. The NDDP GA is then able to formulate fitness or objective functions using a structured query language (SQL). SQL queries are very flexible in defining and extracting desired sets of solutions from a larger database. ParaGen uses these queries as fitness functions which define a certain subset of solutions with desired geometric and performance characteristics. These subsets are dynamically created at the moment a child is needed, and act as the breeding population from which parents are selected. Unlike traditional GA populations, which are generated based only on the preceding population, dynamic populations in an NDDP GA are assembled from the entire range of solutions evaluated up to that point in the process. This allows much more variation in how the populations are formed. In fact, there is no difficulty in having several fitness functions in use simultaneously – either in parallel or serially, because each fitness function creates its own dynamic population using the SQL query.

ParaGen also includes several tools which enhance the exploration of the solution space interactively through a web interface. Because images are linked to every solution, SQL queries can also result in a display of a pallet of solution images. By submitting sorts and queries to the web interface, the designer can very effectively explore the solution space both visually and through performance values. In addition there are multiple graphing functions which allow the construction of Pareto fronts and other techniques to explore the solutions by graphing their performance. Several examples using ParaGen have been presented in earlier work which also demonstrate the operational procedure as well as the post analysis data exploration. This paper describes the overall method and advantages it can offer in early phases of design exploration.

2 THE PARAGEN METHOD

2.1 *The ParaGen Cycle*

For the user, the main interface with the ParaGen database is through a website (http). The relational database which collects all data particular to each solution and displays it on the website is located on the same web server that hosts the website. The first step in the cycle is to create the dynamic population and select two parents. Unlike traditional GA populations which are derived through progressive generations of parents and children, a dynamic population is pulled from the database using a SQL query as a fitness function. The SQL queries provide a great deal of latitude in how this population is formulated. This can include multiple criteria either as simple max/min sorts, or more complex definitions of ranges of combinations of values. From the dynamic population either one parent can be selected for mutation or two can be selected for breeding.

In the second step the breeding algorithm is employed. Half uniform crossover (HUX) (Eshelman, 1991) is used for combining the parent data into one child. The child data is then downloaded to a client machine (a machine connected to the host through the website) in the form of a csv file. This file contains the values for parametric variables that will produce the geometry of the child solution.

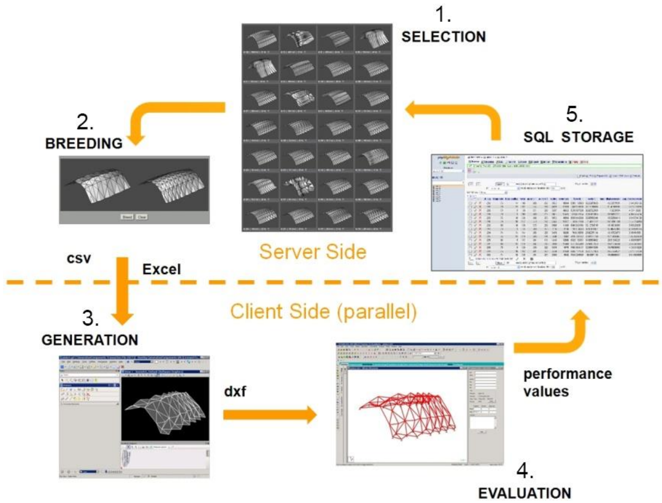

Figure 1. The five steps in the ParaGen cycle. 1. Selection of parents, 2. GA breeding through crossover, 3. Generation of the child solution, 4. Evaluation of the solution, 5. Saving all data in a relational database.

Step three is the generation of this child solution. Any parametric modeler may be used for this task, e.g. Generative Components (by Bentley Systems), Grasshopper for Rhino (by Robert McNeel & Assoc.), Digital Project (by CATIA), Inventor (by Autodesk), etc. At this stage some data and images may also be collected from the geometry. This can include any renderings of images that will help the designer visualize the solution as well as any quantitative data such as floor area, volume or key dimensions. This data is collected in a csv file, which will eventually be uploaded to the database on the server along with the images.

In step four the geometry model is passed to any relevant simulation program where it can be analyzed to obtain the desired performance values. This can be a structural FEM program, or daylighting analysis, or an energy assessment, etc. In each instance the model is prepared by pre-coded scripts which run the analysis and gather the desired performance values. Again any relevant images (such as deflections plots or lighting graphs) can be captured for uploading to the server.

In the five and last step all of the collected performance values, combined with the original child description data is uploaded and recorded in a relational database. The image files are also uploaded to the server and keyed to the record in the database. Before the record is entered in the database, it is checked to be unique so that no duplicates are contained in the data set. This maximizes the exploration of different solutions by the designer.

2.2 *Non-Destructive Dynamic Population GAs*

As mentioned above, ParaGen makes use of a nontraditional GA called a Non-Destructive Dynamic Population GA (NDDP GA) (von Buelow, 2012). There are several advantages that are offered by this method. The key characteristic of the NDDP GA is the use of a relational database to store all solutions rather than retaining only a current generation. This one change makes possible several functionalities that make the method particularly well suited to exploration in the early phases of design.

- Prevention of duplicate solutions
- Enables multi-objective search
- Makes possible dynamic population
- Allows changeable search direction
- Allows interactive search
- Finds Pareto trade-off sets
- Suitable for parallel computation

Duplicate solutions occur frequently in the course of a traditional GA run. In fact as the generations of populations converge on a solution, there are generally more and more duplicates.

This may provide an indication of convergence, but it is also a reason why GAs tend to be so computationally intensive. Since all solutions are saved (non-destructive) in an NDDP GA, there is no need or value in having duplicate solutions in the database. This both increases the computational efficiency and the speed of the run. It also makes interactive exploration much more effective since the designer only sees one instance of each solution. In a SQL database it is easy to set a unique index that can be used to insure each child generated is indeed a new solution. Also there is effectively no limit to the amount of data that can be stored (actually about 256 TB in a MySQL database). So in some cases it may even be reasonable to perform a systematic sweep of the solution space rather than a limited sampling guided by the GA.

Multi-objective fitness functions as well as more complex specifications of portions of ranges or interrelationships of ranges can all be specified using SQL queries. These “fitness functions” are applied to the entire database of solutions, not just the current generation and as such are always able to reach any individuals found to date as candidates in breeding new solutions.

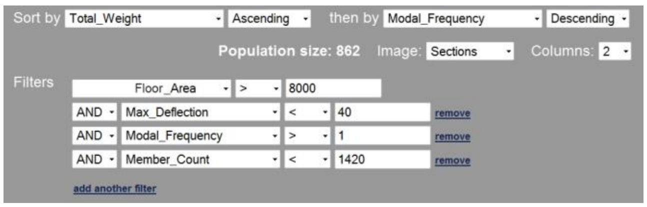

Figure 2. An example of a SQL query used for parent selection or post-processing

A population of solutions used in selecting parents is chosen dynamically during a specific breeding event. The SQL query described above results in a limited population – a subset of the full database – from which two parents are randomly chosen. Through the SQL query, the size and composition of the breeding population can be deliberately chosen. Although there is generally no need for it, it is even possible to use different SQL queries on different machines in parallel. A schematic of the process of creating a dynamic population is shown in Figure 3.

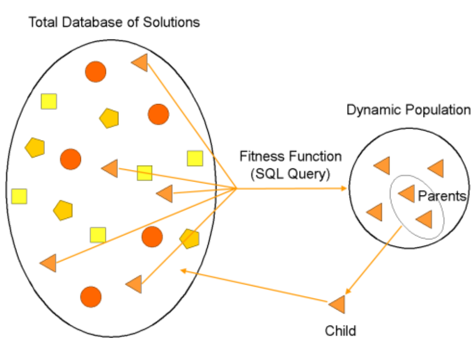

Figure 3. A schematic showing the use of a dynamic population in breeding.

Just as the populations are formed dynamically, they can be changed at any time in the run. Normally this would not be done using a traditional GA since changing the fitness function would interfere with convergence of the populations. But in the case of the NDDP GA the fitness function can be changed at any time allowing more detailed exploration of different areas of the solution space. One could see an analogy in searching a solution space as one searches a database mapping system such as Google Maps. Just as one scrolls to different parts of the map with a mouse, one can “pan” to different areas of the solution space map by changing the fitness function. Figure 4 shows how different areas of the solution space can be pin pointed by using the SQL queries. Based on the same set of data, the three views show the effect of shifting the criteria which define the darker black section.

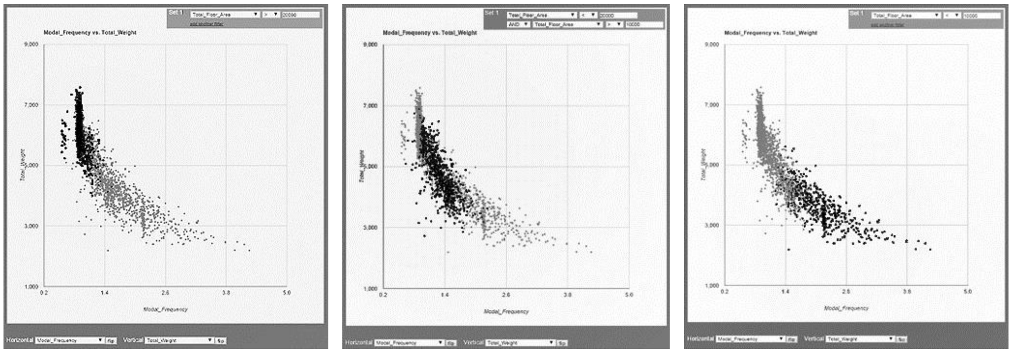

Figure 4. Selection of areas of the solutions space based on performance criteria

It is also possible to allow the designer to interactively select either parents or a breeding population during a run. In this way such qualities as aesthetic or characteristics which are easily defined through visual inspection can be included. The designer may either pick parents individually to breed one child or when a population is chosen ParaGen will continue to choose random parents from this set until the breeding in canceled. Figure 5 shows the selection of two parents by the designer for breeding.

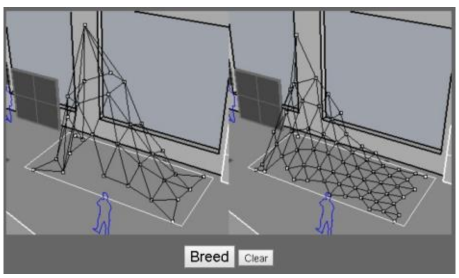

Figure 5. An example of the interactive breeding menu.

In working with conflicting objectives, defining Pareto sets is a well know technique to find possible trade-off solutions. ParaGen offers graphing functions coupled with the standard SQL sort and query functions to generate x-y scatter plots of any geometry or performance variables. The plotted solutions can also be further differentiated by colorization controlled by a second set of SQL queries. Figure 6 shows an example of an x-y scatter plot. In addition filtered solutions can also be depicted in the form of parallel coordinate graphs. These plots are particularly useful in exploring interrelationships of different parameters and performance values. Figure 7 shows and example of a parallel coordinate graph.

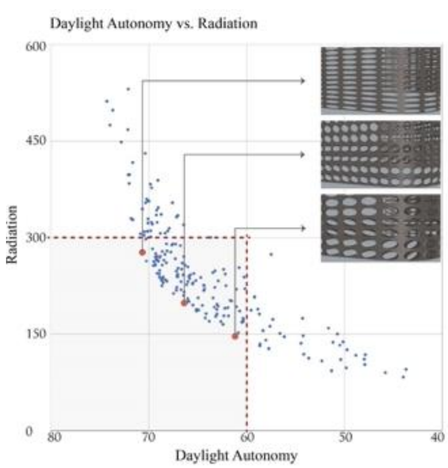

Figure 6. An x-y scatterplot showing a Pareto front of trade-off solutions. (Omidfar, 2014)

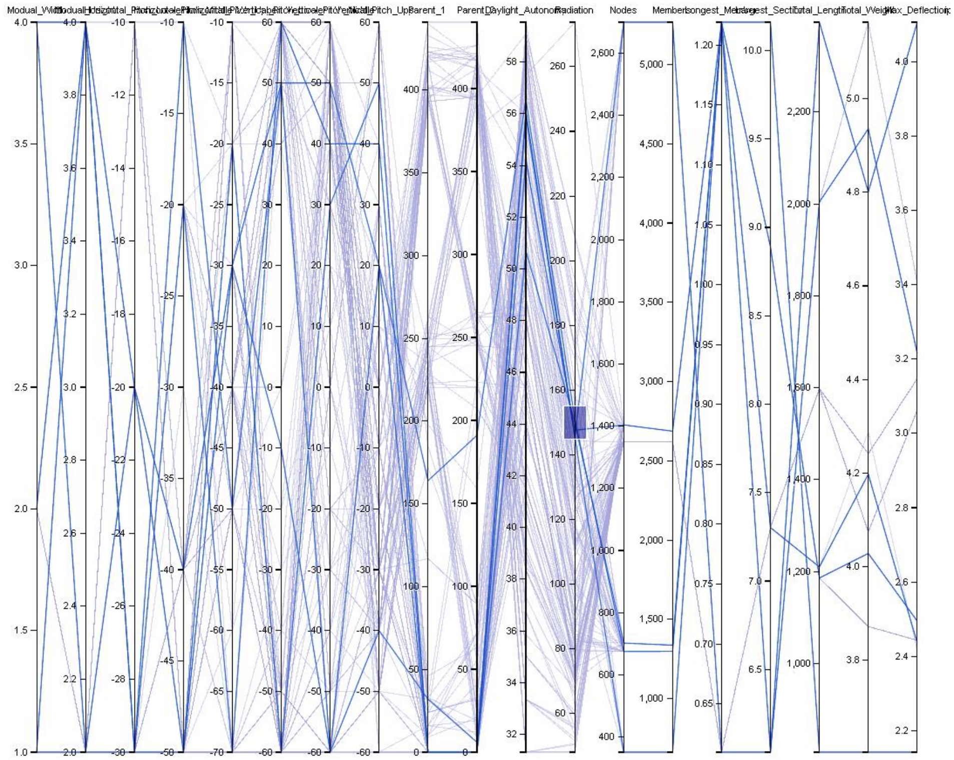

Figure 7. A parallel coordinate plot showing variable values for solutions from Figure 6.

Finally it is also an easy matter to make ParaGen runs in a parallel environment. Since solutions are processed on any machines which can be connected to the web server, there may be any number of machines acting as parallel clients. This can greatly increase the speed of computation although it may initially take some time to set each client up for the run.

3 EXPLORATION WITH POST-PROCESSING

3.1 *Visualizing the solution space*

A major advantage of the ParaGen method is that it allows the visualization of the solution space. This can be done at any time as soon as there are solutions to see. In other words as client machines are cycling through the five step process shown in Figure 1, a designer can view the progress from any other machine through the web interface. The main view through the web interface is the Solution page. On this page the designer can choose:

- What criteria to filter solutions by
- What image to view solutions with
- How many solutions to view at once

What criteria or parameters to use in creating a set of solutions to view is again controlled by the SQL query interface shown in Figure 2. This gives a great deal of latitude in choosing what solutions are displayed. Selections may be made based on any of the original geometry parameters or using any of the performance values found during the simulation and analysis of the solution. The solution can also be sorted using multi parameter sorts. For example Figure 8 compares two sets of solutions: one simply random and the other filtered by height and orientation. This makes an easily observable example, but of course it would be more informative for the designer to filter the solutions by performance criteria.

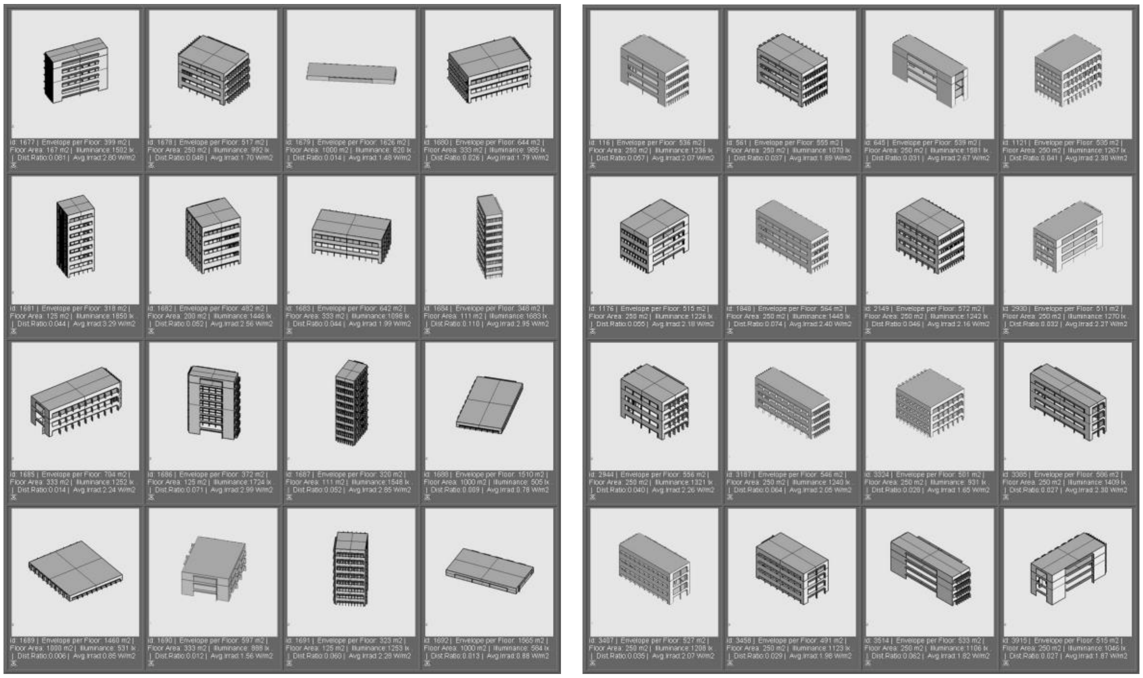

Figure 8. A comparison of random (on left) vs. filtered (on right) solutions

The solutions may also be viewed using any set of images that were collected during the run cycle. Images can of course simply depict the geometry or show some aspect of performance that informs the design decision. Since most modern simulation software includes various graphic displays, a variety of informative images are certainly possible. Figure 9 shows a set of four solutions using three different images. The display can be switched to the various images by simply clicking the pull-down menu. This allows the designer to consider different aspects of a solution using visual characteristics.

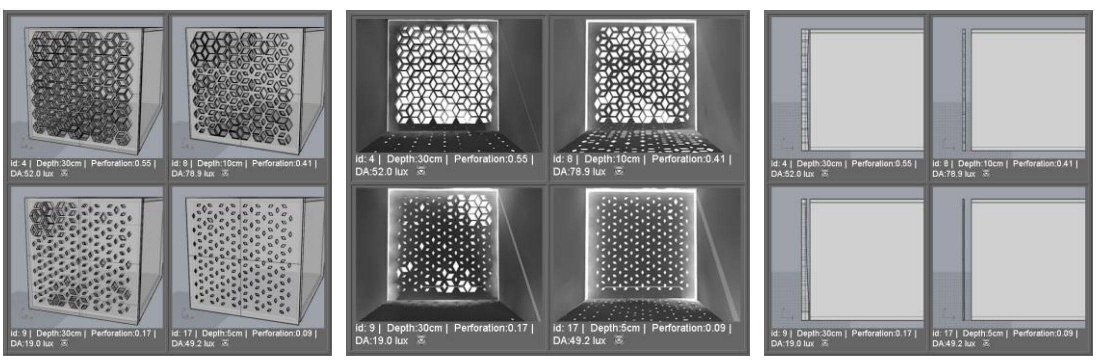

Figure 9. Three image views of the same set of four solutions from a shade screen design: exterior, interior and section views. (Emami, 2014)

Finally it is also possible to control the number of images to be displayed in a row. This gives the effect of a zoom by allowing the images to be displayed at different sizes.

3.2 *Solution analysis through graphing*

In studying and comparing the results of the performance analysis, graphing of the different performance values can be very informative. The techniques and graphic depictions are well known tools in the analysis of data. Two implementations used in ParaGen at this point are scatter plots and parallel coordinate graphs. In the implementation of both of these techniques, ParaGen makes available the same SQL filters used in the Solution page. In this way any specific range or set of data can be plotted using any combination of geometric parameters or performance values. In addition a second level of SQL filters can control coloration of the glyphs in the plot to bring a third dimension to the plots. Figure 10 shows an example of a scatter plot taken from the analysis of a folded plate barrel vault structure. The lower left edge forms a Pareto front giving the best trade-off solutions for the two parameters considered (modal frequency and weight). By clicking on any of the dots the associated image of the solution is displayed.

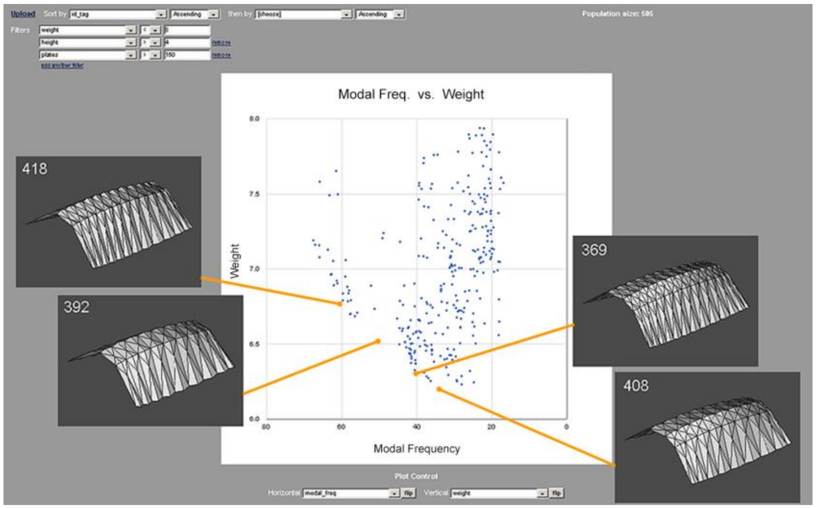

Figure 10. An example of a scatter graph for conflicting parameters. SQL filters at the top left limit the data set. The lower left edge in this case forms a Pareto front. (Falk, 2011)

The other tool used in data analysis is a parallel coordinate graph (see fig. 7). These plots can be useful in understanding relationships between various geometric parameters and performance values. Each vertical line in the plot represents the range of values for that particular parameter. By boxing in certain values, all solutions containing those values are immediately highlighted. This makes tracing relationships visually accessible.

4 CONCLUSIONS

This paper has given an overview of the ParaGen method with examples of how it can be used by designers to explore a multiplicity of solutions in the early phases of a design project. More detailed descriptions of specific applications have been published in recent years (see References). This paper gives an overview of the method with descriptions of individual features.

REFERENCES

Emami, N.; Khodadadi, A.; von Buelow, P. 2014. Design of a Shading Screen Inspired by Persian Geometric Patterns: An Integrated Structural and Daylighting Performance Evaluation. In R. Brasil and R. Pauletti (eds), *Shells, Membranes and Spatial Structures: Footprints. Proceedings of the IASS-SLTE 2014 Symposium. 15 - 19 September 2014*, Brasilia, Brazil.

Eshelman, L. 1991. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, in Gregory J. E. Rawlins (ed), *Proceedings of the First Workshop on Foundations of Genetic Algorithms.* pages 265-283.

Falk, A.; von Buelow, P. 2011. Form Exploration of Folded Plate Timber Structures based on Performance Criteria. In Nethercot, D. and Pellegrino, S. (eds). *Taller, Longer, Lighter: Meeting growing demand with limited resources. Proceedings of the 35th Annual Symposium of IABSE and the 52nd Annual Symposium of IASS. 20 Sept. – 23 Sept. 2011.* London, UK.

Omidfar, A.; Oliyan Torghabehi, O.; von Buelow, P. 2014. Performance-based design of a self-standing building skin; A methodology to integrate structural and daylight performance in a form exploration process. In R. Brasil and R. Pauletti (eds), *Shells, Membranes and Spatial Structures: Footprints. Proceedings of the IASS-SLTE 2014 Symposium. 15 - 19 September 2014*, Brasilia, Brazil.

von Buelow, P. 2012. ParaGen: Performative Exploration of Generative Systems. In *Journal of the International Association for Shell and Spatial Structures.* Vol. 53 No. 4, n. 174. pp. 271-284.