Proceedings of the IASS-SLTE 2014 Symposium
"Shells, Membranes and Spatial Structures: Footprints"
15 to 19 September 2014, Brasilia, Brazil
Reyolando M.L.R.F. BRASIL and Ruy M.O. PAULETTI (eds.)

Using Database Storage to Improve Explorative Optimization of Form Critical Structures

VON BUELOW, Peter

University of Michigan
Taubman College of Architecture and Urban Planning
Ann Arbor, Michigan U.S.A.
pvbuelow@umich.edu

Abstract

Heuristic methods, including genetic algorithms (GAs) have been used and improved for many years in the design and optimizations of complex structural systems. In recent years an additional modification has been developed at the University of Michigan which offers several advantages over traditional GA implementations. This paper outlines specific innovations to GA based design tools, and describes a method which includes both multi-objective optimization as well as more open exploration, with particular application in the early phases of a design project.

The ParaGen, method has been presented previously in several articles (von Buelow [4][5]). This paper looks in more detail at the advantages offered through a non-destructive dynamic population GA (NDDP GA) coupled with a relational database. These two components are combined with associative parametric software for form generation, and simulation and analysis software for performance evaluation in what we call the ParaGen cycle. By storing all generated solutions in a database, solutions can not only be retrieved on demand, but can be retrieved in a programed way based on structured queries. ParaGen formulates fitness or objective functions using a structured query language (SQL). SQL queries can describe very specific sets of data which can be extracted from the entire data set. In the ParaGen method, breeding populations are dynamically created using SQL sorts and queries at the moment each new child is formulated. Unlike traditional populations, which are generated based only on the preceding population, dynamic populations in ParaGen are assembled from the entire range of solutions evaluated up to that point in the process. Since a relational database management system (RDMS) can easily store all solutions, there is no need to ever drop a solution from the potential range of selections. So the NDDP GA is able to make breeding selections from an ever increasing base of potential parents.

This paper describes in detail a list of advantages made possible by coupling an NDDP GA with a relational database. These include the ability to:

- Store all solutions without duplicates
- Use multi-objective fitness functions with SQL queries
- Create dynamic parent populations
- Change search direction instantaneously
- Explore solution space with interactive search
- Graph parameters to determine Pareto trade-off sets
- Utilize parallel hardware for efficient computation

These points are particularly useful in determining good solutions early in the design process when there is still both the freedom to significantly alter the design direction and the potential to benefit most from the improved path.

Keywords: conceptual design, morphology, form finding, optimization, exploration, generative, performance.

1. Introduction – the ParaGen cycle

Before discussing the specific advantages listed above which can be obtained by combining a GA with a database, a brief overview of the ParaGen cycle is given. The steps of the cycle have been described in other publications in more detail (von Buelow [4][5]), so only a brief outline will be given here. There are two parts to the cycle: Server Side and Client Side and these parts are comprised of five basic steps:

- **Select** – using dynamically created populations from the database
- **Breed** – selected parents on the web server and download child to the client
- **Generate** - the geometry using parametric software
- **Evaluate** - with simulation software, e.g. FEA or multiple others
- **Store** - the results in a database on web server for selection and exploration

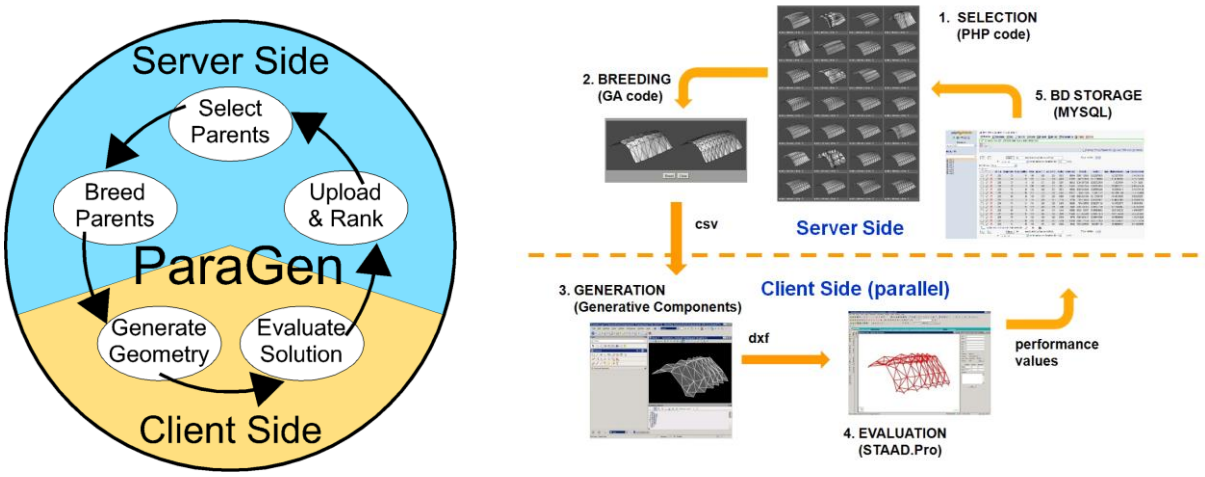

Figure 1: The ParaGen cycle showing two parts comprised of five steps.

Each time the cycle runs, one child solution is created, analyzed and placed in the database with all related values describing both geometry and performance. As the process proceeds, the database builds a description of the solution space in a way that can be directly accessed and explored by designers in a variety of ways. This is what makes the ParaGen method useful in early design phases: it allows the designer to both visually and quantitatively explore and compare a wide range of solutions based on both performance and geometry.

2. Advantages of combining an RDBMS with an NDDP GA

2.1. Storing all solutions without duplicates

The last step in the ParaGen cycle described above is to upload all of the given and collected information about an individual solution to the master database. Modern relational database management systems (RDBMS) are capable of storing and retrieving data in the terabyte range. For example comparing the two common MySQL5 databases – MyISAM has a table size limit of 256 TB while Innodb has a maximum size of 64 TB [2]. In either case, considering that only a few bytes of plain text data are stored in the database table for each solution, the potential database table size is far in excess of amount of data generated in even the most comprehensive design exploration. All saved images are linked to the database entries through ID numbers and are simply stored in the server file system. So there is no physical reason why all solutions cannot be stored.

Beyond the mere possibility, there are several advantages in retaining a database of all analyzed solutions. First, it is the only way to avoid reanalyzing a previous solution. A list of all analyzed solutions is needed to verify that a current solution is in fact a new solution. This is part of the reason traditional GAs are so computationally inefficient – they often reanalyze a solution from a previous generation because there is no retained memory of which solutions have already been tried. In fact, as a population converges on a "best" solution, duplicates generally occur within the same generation. That is basically the definition of a converged problem. But besides the computational inefficiency, looking at multiple copies of the same solution is not helpful to the designer. In

order to aid in the design process the system needs to supply the designer with a set of different solutions that nonetheless all have acceptable performance characteristics. This is what ParaGen is able to do by storing all of the analyzed solutions. A database of solutions also makes many other features possible. These are expanded below.

2.2. Using multi-objective fitness functions with SQL queries

ParaGen uses a Structured Query Language (SQL) in order to access and search the database of solutions. SQL provides a full featured utility to pull targeted information out of a database. In an NDDP GA, SQL queries are used to dynamically build breeding populations (see Fig. 2). The SQL queries are used as a fitness function to select individuals for breeding from the total population. The queries can be either very simple sorts – for example choose the 20 solutions with the lightest weight; or more complex multi-objective filters – for example choose the 20 lightest weight solutions where the modal frequency is greater than 2.0 and where the number of elements is greater than 50 but less than 100 and where the maximum daylight radiation is less than 300. Either of these queries results in a set of 20 individuals which become the breeding population from which parents are selected. The first case represents a single variable fitness function, were as the second query is a much more complex, multi-objective function.

Most design problems are in fact more complex, multi-objective problems that even include qualitative objectives such as aesthetic appearance. In these cases it is essential to be able to formulate more complex search directives. Using the SQL queries, complex fitness functions can be formulated and applied to each parent population individually which not only allows for multi-objective criteria, but also allows for the selection of each parent based on different criteria. This makes it possible to directly combine attributes from individuals which perform well for different criteria.

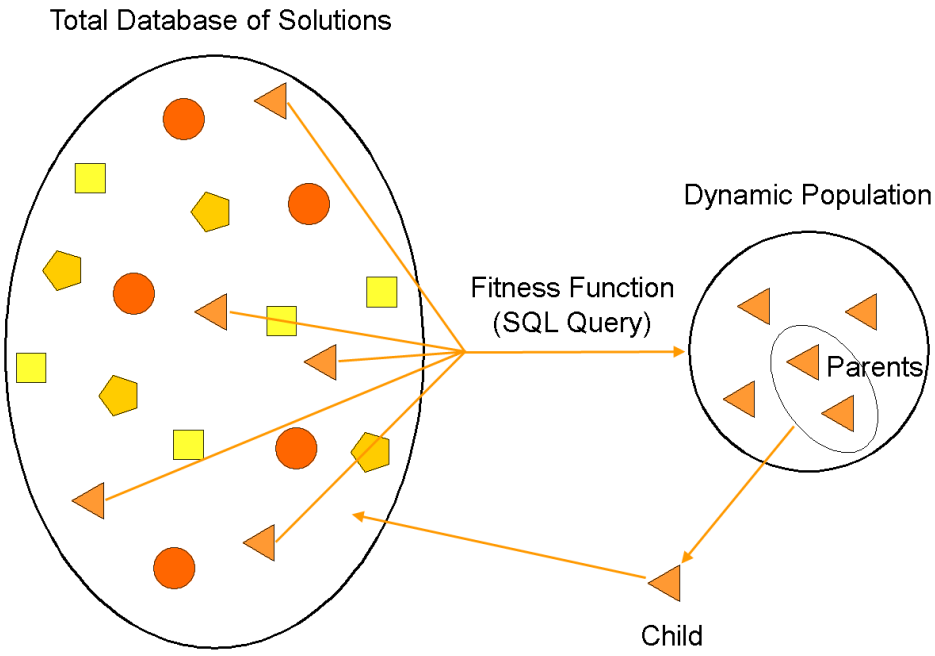

Figure 2: Diagram of selection and breeding using non-destructive dynamic populations.

2.3. Creating dynamic parent populations

In ParaGen the selection of the parent population is completely separate from the breeding routine. At the moment a child solution is required, a new, dynamic population is created for each parent. The populations are created using SQL queries as described in section 2.2 above. In this way the populations are continuously updated as new individuals are added to the total population database. This also removes the requirement for a traditional generational structure where a new population evolves out of the previous population by removing (permanently) less fit solutions. The use of non-generational breeding populations has been used in the past

(Bäck [1]). In the case of static populations, a pool of parents is continually bred and the better performing children replace the worst performing parents. In such a case, solutions are lost as they are replaced by new, better performing solutions The dynamic populations used in ParaGen are a little different since they are created new as required for each breeding by pulling solutions out of the permanent database. As new solutions are created and added to the database they may potentially be included in future dynamic populations, but no solutions are ever lost because all solutions are retained permanently in the main database. Even if the SQL fitness query remains unchanged, the dynamic population will evolve due to the new solutions being added. But the further advantage is that the fitness query *can* in fact be changed as discussed in section 2.4 below.

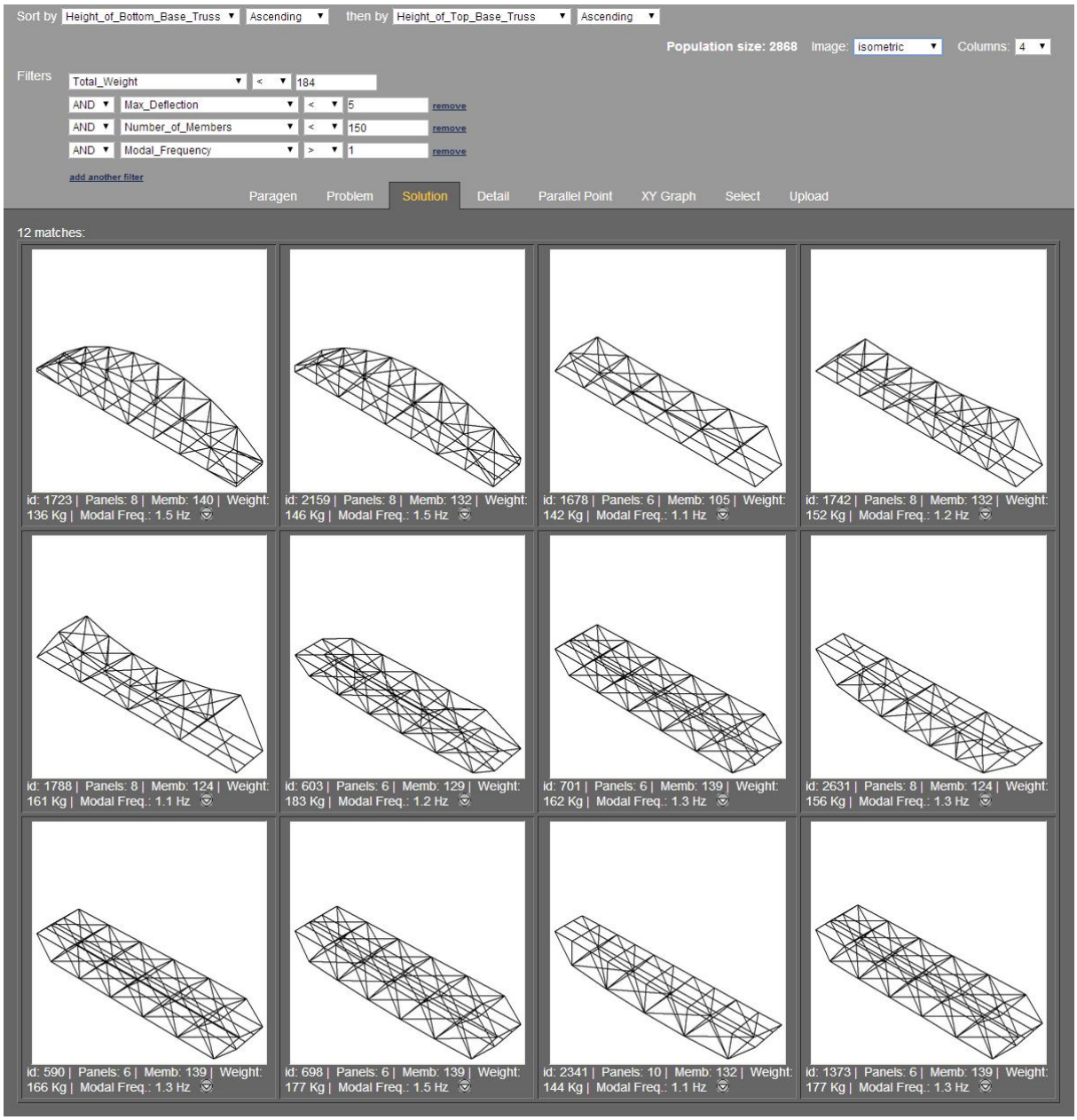

Figure 3: An example of the designer exploration interface on the ParaGen web site for a simple truss bridge design. Filters and Sort pull-downs are show at the top left. By using the settings show, 12 solutions are chosen from the total database of 2868. Display settings include number of columns and the image type displayed.

2.4. Changing search direction instantaneously

The fact that the populations are created new for each breeding instance also makes it feasible to change the fitness function at any point without producing an ill fit population. Since the SQL query which creates the population *is* the fitness function, the current population is always the best suited based on all of the runs to that point. This is not the case with a traditional GA. In the traditional GA the breeding population evolves and cannot be changed except by further evolution – which is slow. The ability to instantly change the fitness function is much more accommodating to design exploration where the object initially is to understand the bounds of the solution space rather than immediately focus on one direction. One might initially set the fitness function to a simple parameter to learn what the better solutions for each parameter look like. For example one might first set the fitness to least weight, and then later to least deflection or any other max/min type parameter. Then subsequently one might devise more complex SQL queries as described in section 2.2 above. Each time the fitness function is changed the breeding population is instantly reconfigured to respond to the new query. By shifting the fitness function, different areas of the solution space are searched in more detail, which actually provides the designer with a better view and understanding of the entire solution space. Also, there is no danger that the range of solutions will become too narrow or get stuck in "hill climbing" since no solutions are ever lost. Different local optima can be found and "climbed" without danger of losing a diverse population base for further exploration in other directions.

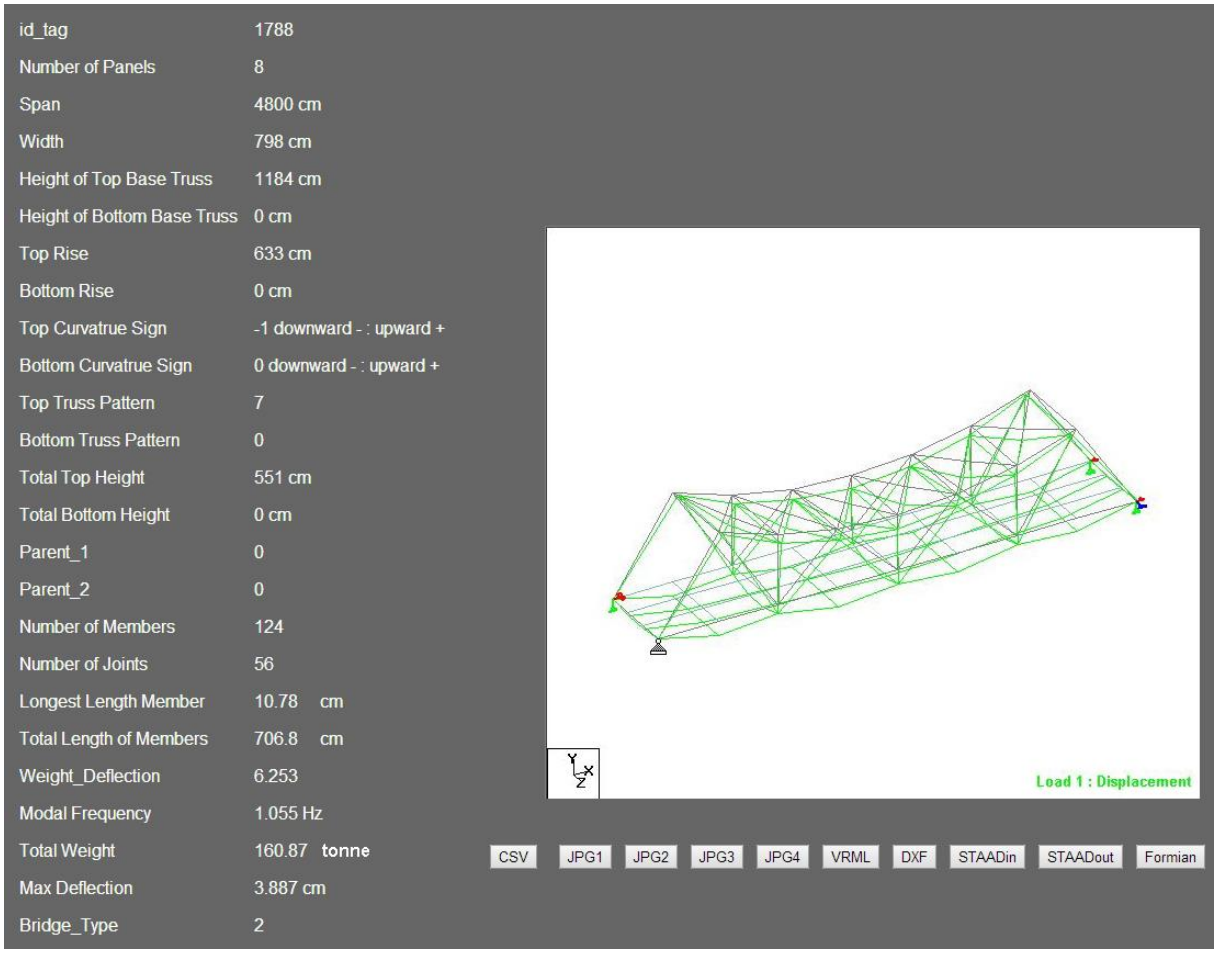

Figure 4: An example of the detail page from ParaGen showing bridge ID 1788 from Fig. 2. The bottom buttons allow access to further downloadable files describing the solution.

2.5. Exploring solution space with interactive search

In the same way that breeding populations can be instantly created with SQL queries, viewable population sets can also be created and viewed through the web interface. This can be accomplished at any time including during

the run, but it is of course more productive after a number of solutions have been created and placed in the database. The web interface contains all of the same SQL sorts and filters in a simple pull-down menu system. Fig. 3 shows the query pull-down and the resulting pallet of solutions. By adjusting the limits of the parameters, the number of solutions displayed can be controlled. The example shown in this paper is from a two lane bridge design using AASHTO HL-93 moving truck loads. (Kodadadi, [3]).

Viewing images of the solutions is generally a very effective way to assess the results and explore the solution space. This is especially true when making qualitative assessments of characteristics which involve aesthetics or other practical considerations which rely on designer experience. For that reason several options are available in the ParaGen interface which allow control over the image display. Besides the SQL sorts and filters that control which solutions are shown, display characteristics such as number of columns and image type (different images potentially captured during the run) can be set. Below each image, key values are displayed which can be expanded by clicking the arrow below each image. Fig. 4 shows an expanded view of bridge ID 1788.

Linked data and image files can be accessed through the buttons below the image (see Fig. 4). Any number of supplemental files of any type can be collected during the analysis process and stored on the server for later access. Fig. 5 shows an example of a VRLM 3D file from bridge ID 1788.

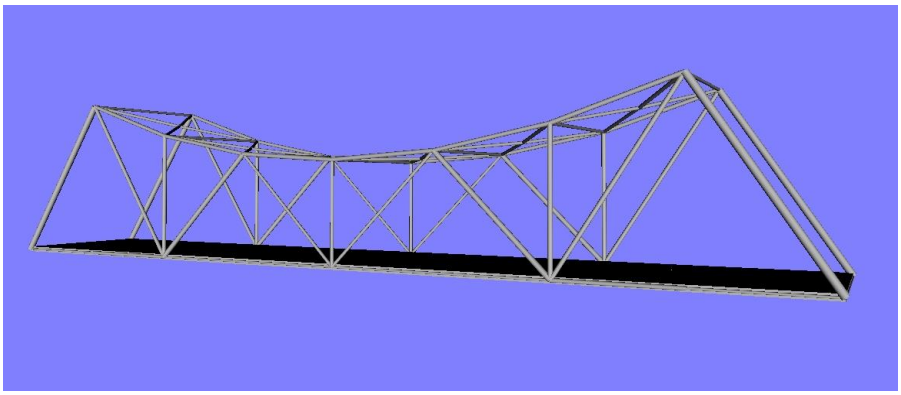

Figure 5: An example of a VRML image downloaded through the interface shown in Fig. 4.

Designer interaction is also possible in the selection/breeding phase. In this way the designer can actually influence or guide which new solutions are created. Using the "Select" tab (see Fig. 3) the designer has three options: create a random mutant child, mutate one parent to get a new child, or breed two parents to get a new child. It is also possible to select an entire population (a user specified group of any size) which can then interbreed and self-generate a larger number of child solutions. In any case, the designer makes selections by clicking images on the "Solution" page (Fig. 3) which then appear on the "Select" page shown in Fig. 6. Each time the user clicks the "Breed" button, the data for a new child solution is generated. In this way the designer can guide the breeding process based on any visual assessment.

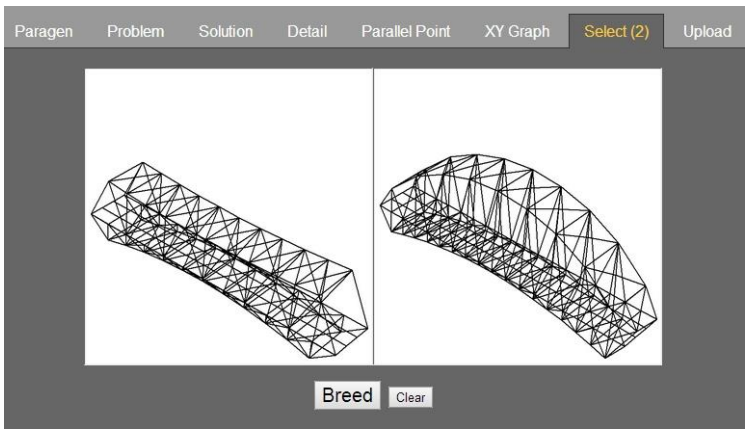

Figure 6: The Select page used for interactive breeding

2.5. Graphing parameters to determine Pareto trade-off sets

Because all of the solutions are recorded in the database with values for all parameters, it is easily possible to interactively plot any pair of parameters. In addition, using the SQL queries it is possible to selectively plot any portion of the solution space. This effectively gives the designer a viewing scope that can zoom in on any portion of the solution space and also filter out any unwanted solutions. So following the example of the bridge design from above, an example might be to set the filter to obtain a limited range of weight and deflection values and further, for example, filter out any trusses with sections below the deck. In this way the designer can quickly target certain sets of solutions, and plot comparative performance values on the graph. Fig. 7 shows a filtered graph of weight vs. deflection of solutions from the bridge problem. A mouse-over of any of the dots reveals the performance values, while clicking on the dots brings up an image of that solution. In this way key performing solutions can be quickly pinpointed.

By graphing opposing parameters Pareto front trade-off sets are easily displayed, which can be further explored by clicking the dots along the front.

2.5. Utilize parallel hardware for efficient computation

As discussed in Section 1, the ParaGen method has two parts - one part runs on a web server while the other part runs on a PC client. Because the breeding populations are not generational, multiple solutions can be processed simultaneously on different machines running in parallel. As each of the solutions that are running on different client PCs are completed, the data and images are simply uploaded to the server database, and the input data for a new child solution is returned to the client PC for the next cycle. Any number of client PC machines can be added to the cluster by simply logging onto the ParaGen website and downloading a child to process. The speed of course scales by the number of machines employed.

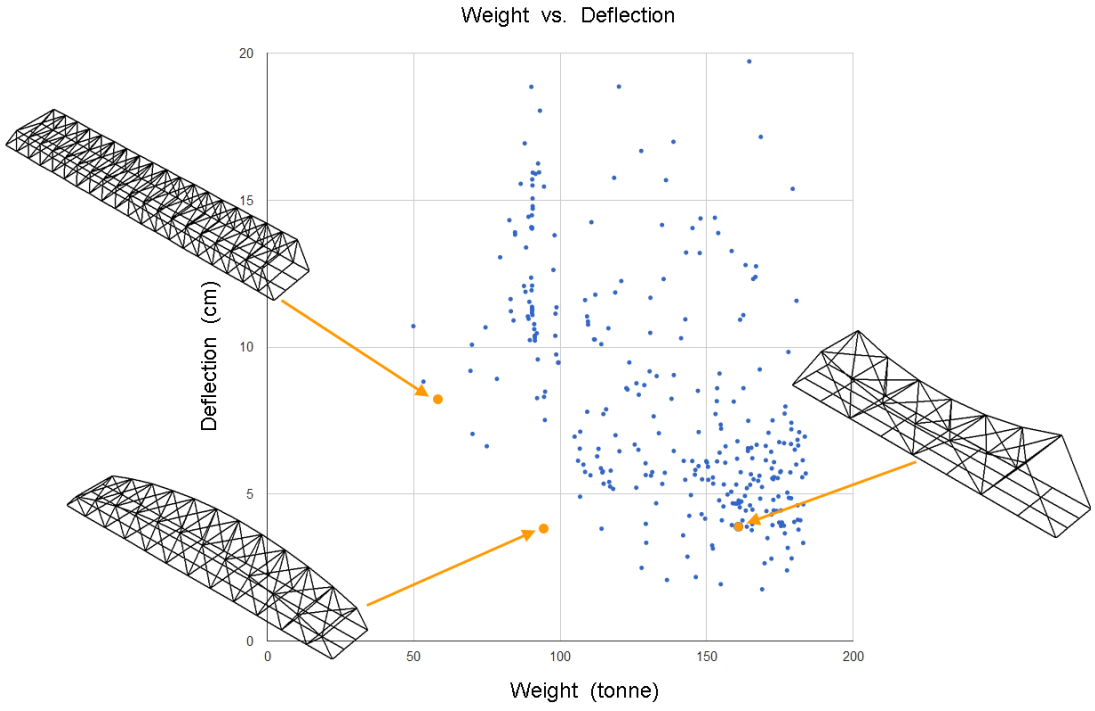

Figure 7: A graph of weight vs. deflection for a selected set of bridge solutions. By clicking dots data and images of the solutions are revealed.

3. Conclusion

This paper outlines advantages that can be attained by combining non-destructive dynamic population genetic algorithms (NDDP GA) with relational database management systems (RDBMS). The advantages are illustrated

by examples, and show there is a significant enhancement in the level of information that can be supplied to the designer when implementing such a system. The development of this method at the University of Michigan is called ParaGen, and has been successfully employed over the past several years to approach a wide range of integrated design problems. The method offers a primary advantage in the early phases of design when the exploration and comparison of different approaches to a problem are most needed. Further development continues with expanded comparative assessment tools.

Acknowledgements

The author is most appreciative to Taubman College for ongoing support of the Hydra Lab and the ParaGen project.

References

[1] Bäck T., et al., *Handbook of Evolutionary Computation*, Oxford Univ. Press, 1997.

[2] *MySQL 5.6 Reference Manual*, Sebastopol, CA ; Cambridge : O'Reilly Community Press, 2002.

[3] Kodadadi, A.; von Buelow, P. Dynamic Configuration Processing and Optimization of Forms (Exploring Truss-bridges). 1st International Conf. on Engineering and Applied Sciences Optimization (OPT-i 2014).

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

[5] von Buelow, P., Techniques for More Productive Genetic Design: Exploration with GAs using Non-Destructive Dynamic Populations, in *Adaptive Architecture*, Proceedings of ACADIA 2013, Cambridge, Ontario, Canada.