BREEDING BRIDGES: GENETIC BASED FORM EXPLORATION

Peter VON BUELOW
Assistant Professor, Taubman College of Architecture and Urban Planning, University of Michigan, Ann Arbor, MI 48109 USA pvbuelow@umich.edu

ABSTRACT

This paper considers the distinction between exploration and traditional optimization as methods of form finding in structural design applications. Although optimization is usually a component of exploration, a distinction is made between the goals of the two. Optimization usually implies the search for one single *best* solution, were as exploration seeks a set or range of *good* solutions.

A topology exploration method is described based on optimization using Genetic Algorithms (GAs). The implementation is called an Intelligent Genetic Design Tool (IGDT). The advantages offered by GAs in terms of exploration are discussed, and the mechanics of finding a set of significantly different, good solutions is outlined. In the IGDT, GAs are used for both geometry and topology optimization of truss structures. The solutions discovered are passed through a filter to produce a set which represents good, and also different solutions. The size of the set of solutions produced for exploration is determined by setting a passing threshold. The solution set is further filtered to show only significantly different geometries within this set. This limits the number of solutions show to a size that is useful to a designer performing exploration.

An example is give based on the design of a simple truss bridge. Two techniques are demonstrated. The first technique runs unattended, in the automatic mode, to produce a pallet of possible solutions to explore. The second technique involves interactive selection to explore different topologies in more detail through genetic breeding and mutation.

INTRODUCTION

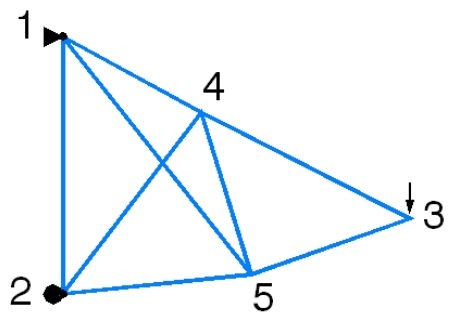
Exploration is not the same as optimization, although it usually requires searching for optimal or near optimal solutions. Exploration in the sense of design or form determination requires searching for a set of significantly different "good" solutions [1]. Exploration might proceed a more traditional optimization procedure, and be used to determine the significant design objectives. Multi-objective optimization is also not the same as exploration. Pareto Set solutions are similar to sets of solutions found by exploration, but Pareto Sets are derived using definite, predetermined variables and yield only a limited view of the range of the solutions beyond those variables. Exploration is concerned with exposing the wider range of good solutions.

Genetic Algorithms (GAs) have demonstrated success in a wide range of optimization problems [2]. GAs, and Evolutionary Computation methods in general are particularly well suited to exploration because they use sets of solutions as a fundamental component of the method. Exploration has as a goal the discovery of a set of solutions, were as traditional optimization has the goal of finding one "best" solution. In early stages of form determination, exploration may discover new and better solutions by searching a wider range of possibilities for reasonably "good" solutions. The other quality GAs share with exploration is the discovery of unexpected solutions [3]. In terms of structural form, these may be solutions that contain aspects not driven by the stipulated design variables. The great value of exploration is that it expands one's view of the problem itself. Using mechanisms like mutation and recombination, GAs have the ability to find solutions that may not be anticipated by the designer. In this way exploration using GA based search techniques can help the designer find a better solution by considering possibilities outside of what is expected.

FORM OPTIMIZATION USING GENETIC ALGORITHMS

Use of GAs or other Evolutionary Computation techniques in the area of structural optimization is well established [4]. Typically chromosome strings are used to define the on/off settings of members based on a predetermined ground structure. One drawback in this approach is that the ground structure has a limiting effect on the possible topologies that will be explored. Another problem in terms of the GA coding is that the chromosomes are generally longer than necessary which raises the computational cost. The method used by the

IGDT is based on the incidence matrix (Figure 1.) as described in a previous paper.[5] Also, the breeding method used to cross topology types is able to operate with unequal length chromosomes (Figure 3.). The result is that any structure that can be described with an incidence matrix (any truss) can potentially be found in the exploration process.



X	1	2	3	4	5
1	\	1	0	1	1
2		\	0	1	1
3			\	1	1
4				\	1
5					\

Joints 5 Members 8

Incidence Matrix

Chromosome

10110 11111

Figure 1. An example of a truss with corresponding incidence matrix

Optimization is carried out in stages: member sections, geometry and topology. Topology is the highest level. Each topology is optimized in terms of its geometry and individual members. Topology and geometry are each optimized using nested GAs, that is each topology in the GA population spawns in turn another GA which optimizes the geometry. The member optimization used is simply the selection of the lightest member cross section based on tension or compression criteria of the US steel code (AISC ASD 1991). This is performed using the controlling load case (or load position in the case of moving loads) for each member.

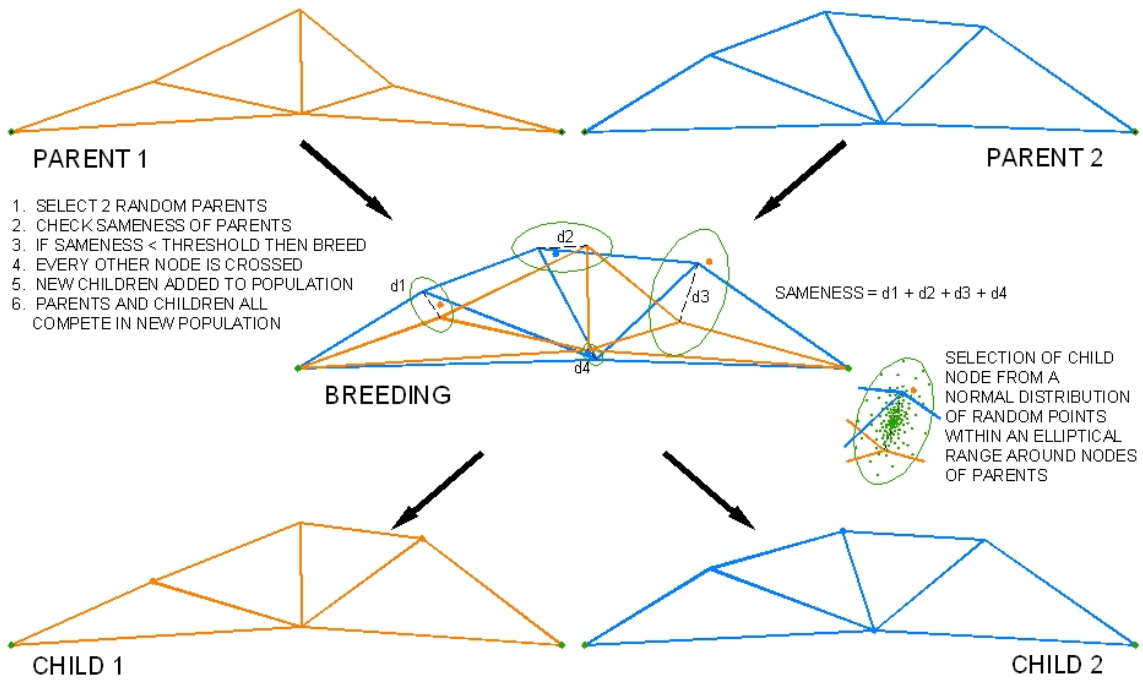


Figure 2. An example of geometry breeding using half-uniform crossover.

Geometry Optimization

The geometry optimization employs a CHC-GA in which randomly paired parents having a passing difference are bred to produce two children (Figure 2.). The population size for the example shown in this paper was 50 parents in a generation. The CHC runs in cycles with a restart between each cycle. At restart the best individual

is retained and mutated 49 times to begin a new cycle. A cycle runs between 100-200 generations and 3 cycles are used for each optimization run. The CHC-GA is described in more detail in reference [5].

Topology Optimization

In the IGDT, another GA is used to optimize the topology. The individuals generated in this part are filtered and selected for the exploration process. The topology GA is more traditional except that it makes use of chromosomes of different length as shown in Figure 3. The population size used for the automatic mode was set at 50 parents (altogether 100 for parents and children), although smaller sizes have also given satisfactory results. The interactive mode uses smaller populations of 10 parents, since the selection of each population is made visually on the monitor screen.

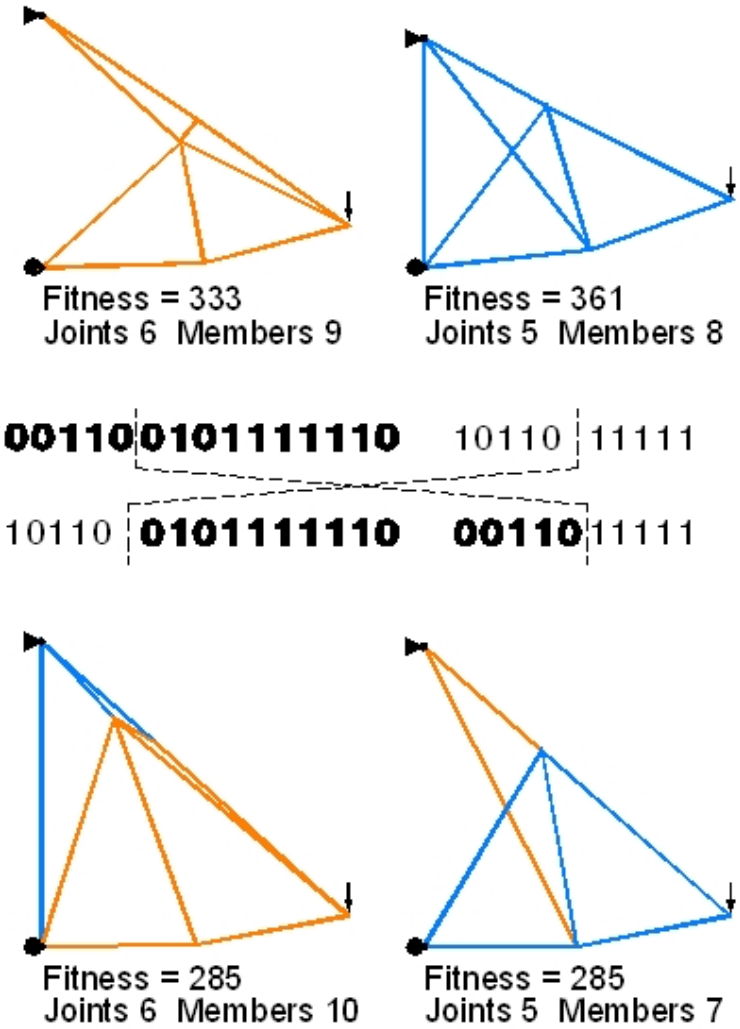


Figure 3. The breeding of two trusses of different length to produce two children.

FORM EXPLORATION

For exploration to be productive, the number of solutions that are offered by the program for review by the designer has to be limited to a manageable quantity. If one regards the number of possible solutions that the program generates in the course of topology optimization, one quickly sees the futility of attempting to even to glance at each of the many solutions. Using population sizes indicated above, 150 geometry generations of 50 parents each (these represent the 50 best selected from a larger total population) would give 7500 individuals for each geometry cycle. Using 3 cycles in each geometry optimization, this produces 22500 individuals for each optimization of a single topology. If the topology population is again 50, and cycles run for 5 cycles with about 20 generations each, one would have around 100 million individuals. Printing 24 individuals on a page for viewing one would have about 4.6 million pages to peruse. And if one could view about 2 pages a minute,

inspecting a new truss almost each second, the task could be completed in about 39000 hours or 4800 days or every day for 13 years. So the obvious problem is to filter the solutions down from 4.6 million pages to about 10 or so. At the same time, for effective exploration, one wants to see not the 100 best, which would likely all look the same, but 100 *good but different* solutions.

Automatic Mode (Non-interactive Solution Filtering)

In order for exploration to be practical, the overall number of viewable solutions has to be limited to a number that one can realistically inspect. In order for exploration to be effective, the solutions that are presented need to be significantly different. The overall number of solutions is limited by using a fitness limit, where only solutions scoring better than the limit will be saved. The significant difference is maintained by keeping only the best from each topology type. Making distinctions regarding the latter criterion is a bit more difficult. In the present algorithm, an attempt is made to number the joints in the same order (supports first, then either left to right or bottom to top, depending on the general geometry and which ordering would be less ambiguous). Then only the best of each topology type, as defined by the incidence matrix, is kept. Since significantly different geometries of the same topology will generally have different joint ordering, they will be regarded as different types, which is desirable for exploration. Finally the selected set is sorted by fitness before printing, so that if it comes up a bit long, the last pages can be either brushed over more quickly or just skipped. In this way the goal is achieved to distill 100 million solutions down to not simply the best 100, but hopefully instead, the most interesting 100.

Interactive Mode (User Selection)

Exploration in the interactive mode has the advantage that one can somewhat steer the direction of search. Rather than using elitist selection based on a fitness function, selection is made by the designer directly. The program allows interaction in the following ways. Firstly, in establishing the initial generation, there are three possibilities:

- User supplied initial population – any combination of repeated or different trusses.
- User supplied single "progenitor" which is mutated to build the initial population.
- Randomly generated initial population.

In all subsequent generations one can choose from among five possibilities:

- Choose any combination if individuals from the current generation to breed as parents for the next generation.
- Pick single individual from the population to act as a new progenitor, which will then be mutated to build a new generation based just on that single individual.
- Insert a new individual (possibly modified by had) to act in combination with either of the previous possibilities.
- Automatically select parents based on fitness (same as automatic mode, but just for one generation).

Using the above selection mechanisms, the process can be continued in any order as long as is useful. In addition to the screen output, files are created which record each generation for plotting (AutoCAD format) or input into further FEA programs. Also a summary file is created which removes duplicates and individuals below a given level of fitness.

Although the interactive mode gives the designer some degree of control, as compared to the automatic mode, much less of the solution space is covered. As suggested above, the nature of the GA is such that viewing all of the solutions generated is generally not practical. What can be useful is to further explore topologies generated in the automatic mode by means of the interactive mode. Either individual solutions or pairs or groups of solutions can be selected from the automatic mode for further development in the interactive mode.

EXAMPLE APPLICATION: TRUSS BRIDGES

This section gives examples of solutions that the program generates in both the automatic and interactive modes. The problem shows the exploration of alternative solutions to an existing single lane truss bridge with a simple span of 36.2 m. Figure 4. shows the original truss bridge. Two loadings are considered, a moving AASHTO HS 20, 2-axle truck load (14.2 kN axle loads placed on 6 successive pairs of deck joints), and a uniform lane load of 9.4 kN/m. Dead load was left out of this example to decrease the run time. Members are sized using the 1991 AISC-ASD steel code using circular tubing (pipe) sections. Figure 5. shows the loading cases considered. The fitness function used for the GA was least weight. In addition a progressive penalty is applied for deflections greater than L/120.

Figure 4. The Foster Bridge, Ann Arbor, Michigan. Alternative designs are based on this case study.

Automatic Mode Solutions

Using a passing fitness filter of 2570 kg for a single truss the program found 550 different solutions in 5 cycles from a total of 2350. This was more than enough, but the fitness was deliberately set a bit low to include the original topology of the bridge at 2348 kg. From this set the 21 trusses shown in Figure 6. were visually selected. As one can see, the topologies differ quite a bit, and yet the fitness function of weight, is often very similar. Topo 3120 is actually the lightest at 1895 kg, and the heaviest was Topo 27 at 2378 kg (actually a bit more that the original design of 2348 kg). As mentioned, this set was simply selected on appearance, certainly a criterion that an architect might consider. But other, more quantitative values (maximum deflection, modal frequency, numbers of joints or members, clear height) might also certainly be considered. The point is that a selection can be made that might include something other than such quantifiable parameters. For instance Topo 1819 is very regular and visually rather calm, while those in the row above it (Topo's 27, 755 and 954) are all visually more active and dynamic. Because of the symmetry, what happens on the center line is important because it provides a natural focus for ones gaze. Topo 2701 is interesting in this regard by leaving the center completely void.

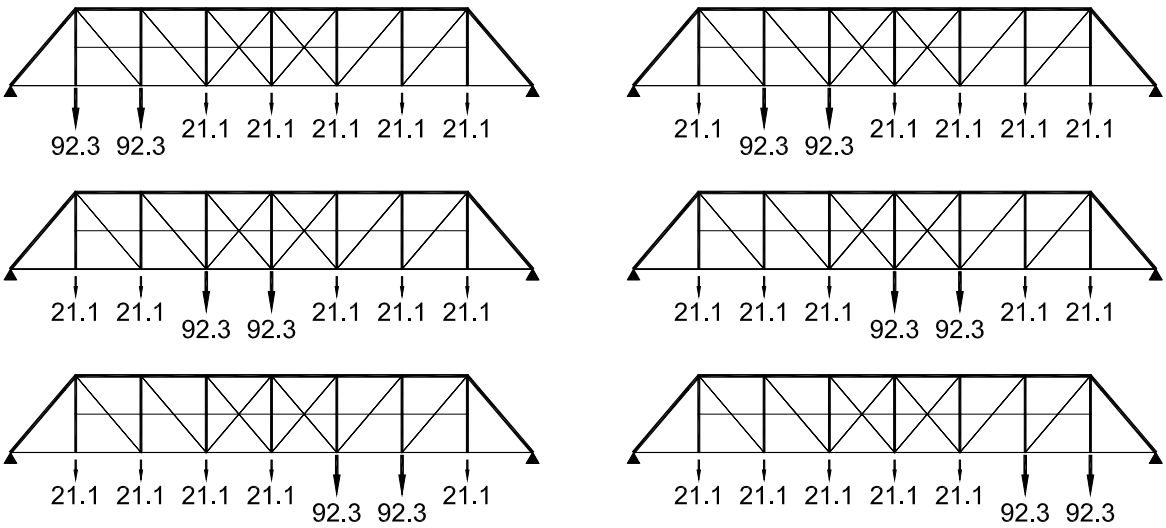


Figure 5. Moving load case used in the bridge example. All values shown are in kN.

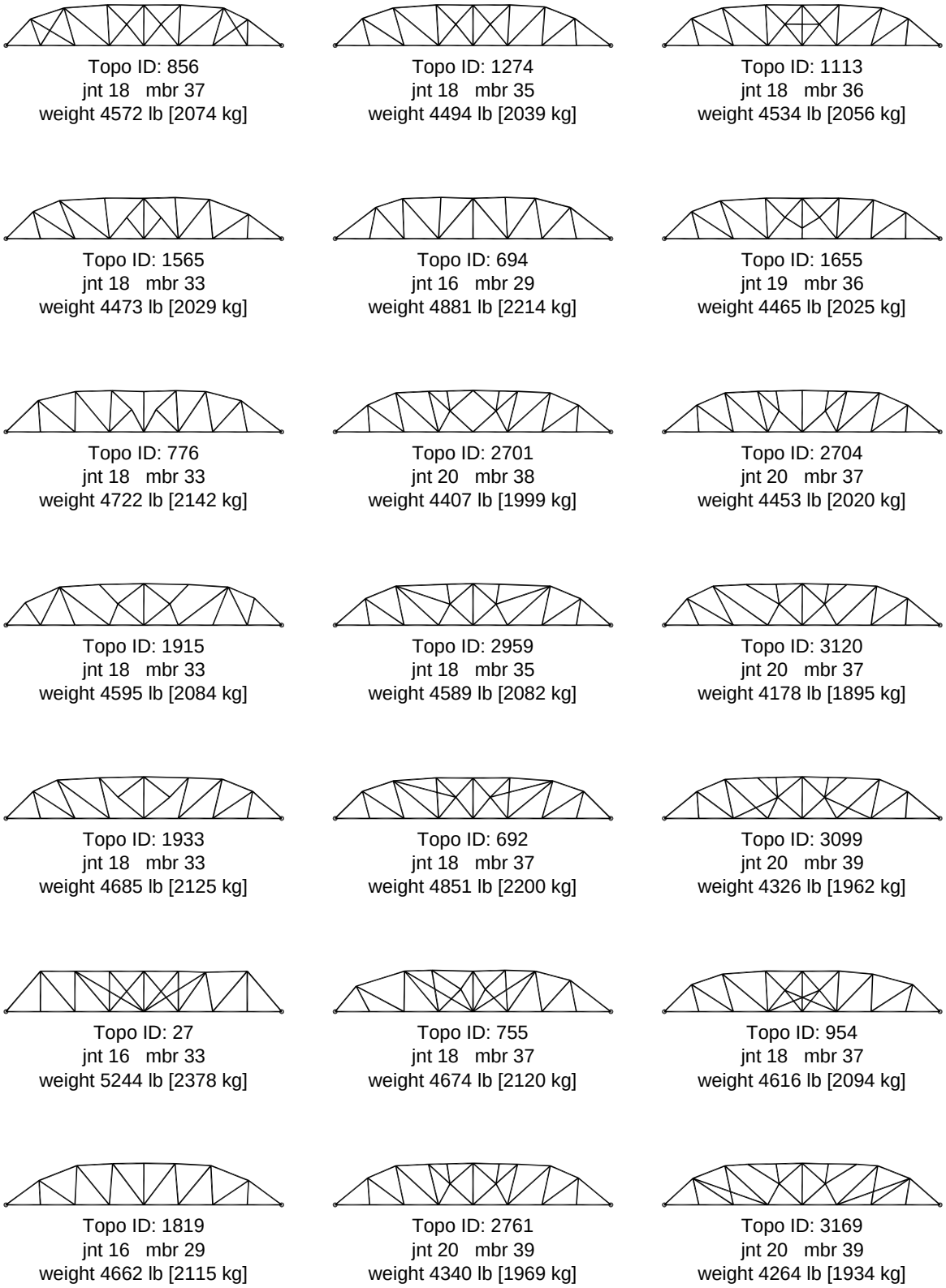


Figure 6. Selections from the automatic mode based on the Foster Bridge.

Interactive Mode

For the interactive mode a parent population of 10 was used. This is a number that can comfortably be viewed on the monitor screen (4 x 5 for the full population). Figure 7. shows this layout used for interactive selection. The text window provides for simple typed input. Although any of the topologies show from the automatic mode in Figure 6. would provide workable solutions, the interactive mode can be used to explore a particular

direction in more detail that may not be in the direction defined by the coded fitness function of the automatic mode alone. In this example a conscious effort was made to explore trusses that have a double arch form with a slight depression in the center. This was chosen primarily for aesthetic reasons. The desire was to explore topologies that contained this visual element, but at the same time provided a good solution functionally in terms of load support.

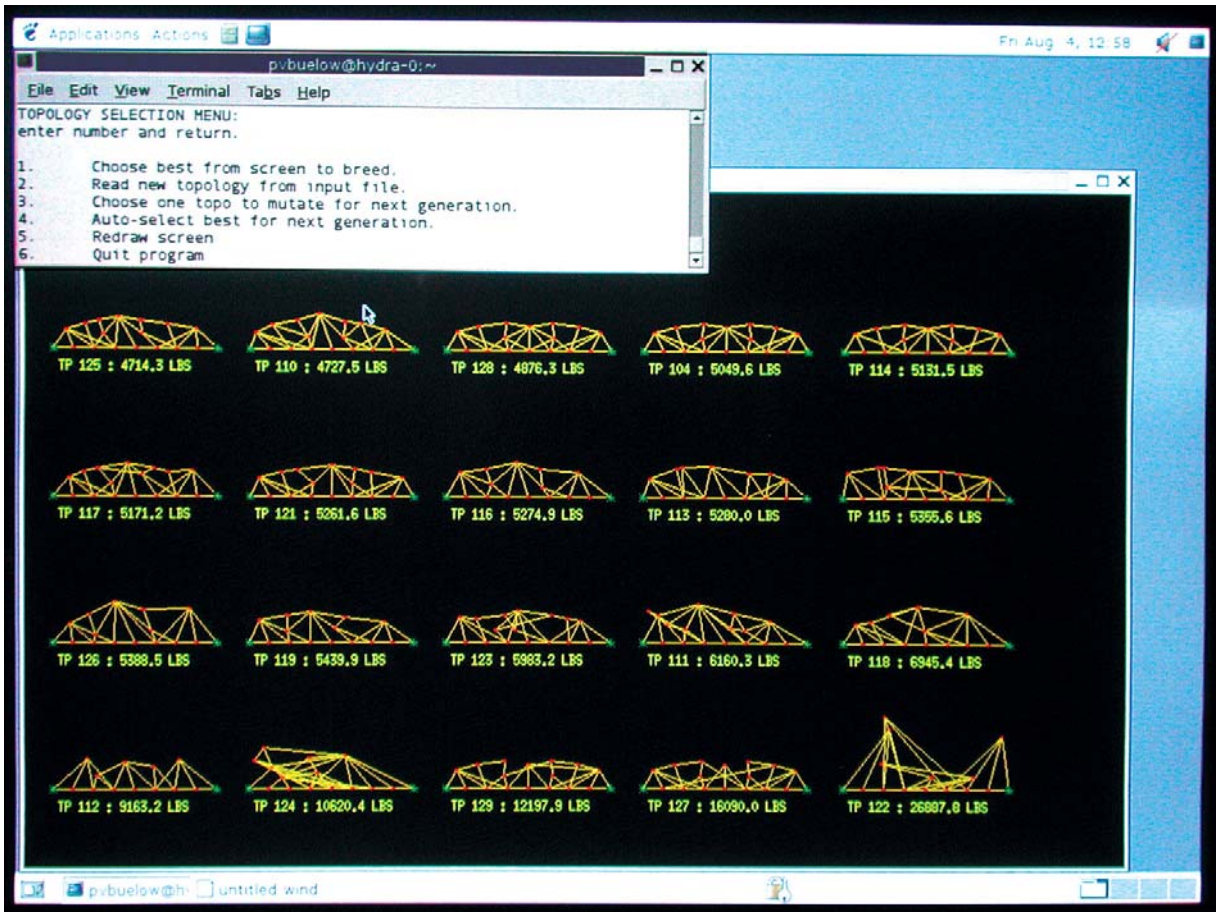
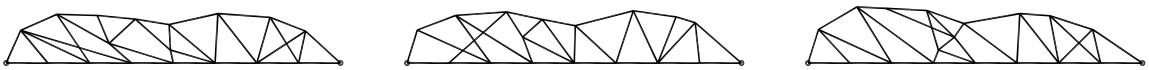


Figure 7. The screen layout of the interactive mode allowing user selection.

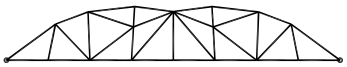
As a point of departure for the interactive mode, several trusses were found from the automatic runs that exhibited the desired profile. These trusses had been discarded earlier because of their poorer performance, but the one visual element of the profile was seen as worth exploring. Figure 8. shows 3 of this type.



Topo ID: 1152	Topo ID: 1639	Topo ID: 1640
jnt 19 mbr 37	jnt 18 mbr 33	jnt 19 mbr 36
weight 5375 lb [2438 kg]	weight 5559 lb [2522 kg]	weight 5430 lb [2463 kg]

Figure 8. Three examples found from earlier automatic runs with the desired profile shape.

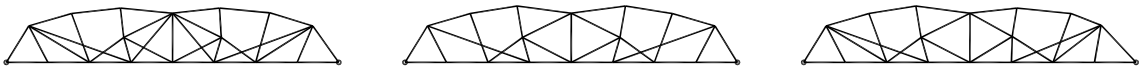
The goal was to explore trusses with this profile that might also provide better performance under the truck loading. Using the interactive mode as described above a population was generated based on the truss forms in Figure 8. Actually, 4 runs were made with different choices made in each. In the course of the exploration all of the options listed above were used. After viewing some of the IGDT solutions, the author added his own design into the breeding pool as a progenitor on one of the runs. This truss is show in Figure 9. It eventually evolved into forms like Topo 190 in Figure 10.



Topo ID: 0
jnt 18 mbr 35
weight 5394 lb [2447 kg]

Figure 9. A suggested topology made by the designer.

Figure 10. shows three examples of trusses that were found during the interactive exploration. They all have the desired profile and also represent good solutions to the structural conditions. Each generations required about 5 minutes to run if based on mutations and about 10 minutes if based on breeding. This was running parallel on a Linux cluster of 60 class P-III machines.



Topo ID: 162
jnt 18 mbr 37
weight 4823 lb [2188 kg]

Topo ID: 179
jnt 18 mbr 33
weight 4802 lb [2178 kg]

Topo ID: 190
jnt 18 mbr 35
weight 4723 lb [2142 kg]

Figure 10. Three solutions selected from the interactive mode.

CONCLUSIONS AND FURTHER DEVELOPMENT

As a method of form exploration the approach described in this paper is very successful. It finds solutions which are both know and found by other optimization methods. What is particularly significant is the additional capability of exploration of *good* solutions rather than a concentration on finding one *best* solution. This approach is more helpful in early design phases in which many parameters are not yet determined. In promoting creative or innovative solutions, methods like the IGDT, based on evolutionary computation have an advantage.

One disadvantage to this method is that it is very computationally intensive. The IGDT is currently programmed for operation on Linux cluster architecture which greatly enhances the speed of operation. The Hydra cluster in Building Technology Lab at the School of Architecture, University of Michigan was recently expanded to 100 processors. With this increase in size, we hope to be able to expand the IGDT program to include 3D frames in the near future.

REFERENCES

[1] J. S. Gero (1994), "Towards a model of exploration in computer-aided design", J. S. Gero and E. Tyugu (eds.), *Formal Design Methods for CAD*. North-Holland, Amsterdam, 315-336.

[2] Mitsuo Gen and Runwei Cheng (1997), Genetic Algorithms and Engineering Design, Wiley, New York.

[3] David E. Goldberg (2002), The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Kluwer, Boston.

[4] Leite and Topping (1995), "Improved Genetic Operators for Structural Engineering Optimization". Topping, B.H.V. (ed.). Developments in Neural Networks and Evolutionary Computing for Civil and Structural Engineering. Civil-Comp Press, Edinburgh, UK.

[5] Peter von Buelow (2002), "Using Evolutionary Algorithms to Aid Designers of Architectural Structures", Bentley and Corne (eds), Creative Evolutionary Systems, Morgan Kaufmann, 315-336.