Proceedings of the 6th International Conference on
Computation of Shell and Spatial Structures
IASS-IACM 2008: “Spanning Nano to Mega”
28-31 May 2008, Cornell University, Ithaca, NY, USA
John F. ABEL and J. Robert COOKE (eds.)

Using Evolutionary Computation to explore geometry and topology without ground structures

Peter VON BUELOW

University of Michigan, TCAUP
2000 Bonisteel Blvd, Ann Arbor, MI USA
pvbuelow@umich.edu

Abstract

Over the past two decades there has been an increasing interest in using what has come to be called Evolutionary Computation (EC) in the analysis and optimization of structural systems. These methods include Genetic Algorithms (GA), Evolution Strategies (ES), Simulated Annealing and other stochastic based numerical methods. Each of these methods shares the drawback that they are very computationally intensive compared to deterministic methods. Furthermore, the computational burden can rapidly increase as the size of the analyzed structure increases. This paper suggests that the level of computation can be significantly reduced by avoiding the common practice of using ground structures in coding the topology. Additionally, comparative examples show that a broader range of good solutions can be reached when the use of ground structures is avoided.

1. Introduction

The problem of topology and geometry optimization of discrete structures like trusses using EC is often hindered by conditions associated with the use of a ground structure. Ground structures are used to define both topology and geometry by providing a raster of admissible nodes between which members can either be present or not present. This on-off character of ground structures makes them well suited for description as binary strings used by conventional EC methods like Genetic Algorithms. But the use of ground structures produces a computational hindrance in that the length of the binary string (chromosome) impacts the level of computational intensity needed for the EC. This is because the size of the EC population is related to the length of the chromosome string used to describe the solutions. Since each individual in the population needs to be analyzed and breed with other individuals, the level of computation quickly escalates as the size of the ground structure, and thereby the population, increases, even though the total number of members present may not be so high.

In addition, a coarse ground structure can negatively impact the quality of the solutions found. The dilemma is that to find better solutions a finer ground structure is needed, but by using a finer ground structure the computational intensity makes the method impractical for larger systems. This paper hopes to offer some suggestions to code the topology and geometry without the use of ground structures, and thereby reduce the computational intensity of the problem and enhance the quality of the solution.

2. Ground structures

Ground structures have been used since Dorn et al. [1] proposed their use for topology optimization of plane trusses. Dorn describes a ground structure as the set of all admissible bars which can be considered to connect a set of admissible joints. Ground structures were first applied in direct computational methods like Linear Programming. Figure 1 shows an example by Klarbring et al. [2] of a ground structure used with a Linear Programming solver to design a bridging structure. Klarbring selected 8767 members in the 210 node ground structure shown on the left in Figure 1. All possible members for a 210 node ground structure would total 21945. On the right side only the members selected for the solution are depicted. There are of course many more members on the left of Figure 1, than actually selected in the solution shown on the right.

EC applications use coded (usually binary) strings, analogous to chromosomes, to describe individual solutions. In this paper the solutions are structures or trusses. The "genetic" description of the structure is represented by the binary string. When ground structures are used, this coded string takes the form of an on-off list of all admissible members. In this binary string, member connectivity is represented by 1's and no connectivity (no member between nodes) is represented by 0's. So in the Klarbring example, the string would need to be 21945 bits long in order to include all possible members of the 210 node ground structure. In fact it makes no difference whether all of the members are present as on the left or just selected members as on the right. In either case the binary string would have the same length or 21945. Only the ratio of 1's to 0's would change.

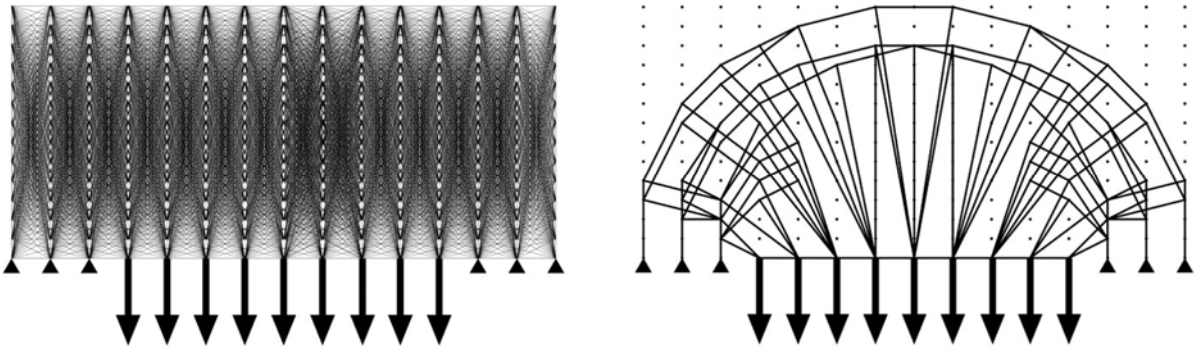


Figure 1: Left - Ground structure from Klarbring (1995) showing 8767 of 21945 possible members.
Right - Ground structure nodes and optimized topology from Klarbring (1995).

Unfortunately, in EC, the longer the length of the genetic string, the larger will be the population needed to evolve towards better solutions, see Goldberg [3]. And the larger the population size, the more generations will be needed to move the population towards convergence. The size of the population and the number of generations are what make EC so computationally demanding. Each solution must be bred, new solutions generated and then each new solution evaluated. So the coding efficiency of the genetic string becomes critical to the performance of the EC.

2.1 An example using a ground structure

The example shown in Figure 2 is by Deb & Gulati [4] and uses a Genetic Algorithm to produce optimal geometry and topology for a bridge truss based on a ground structure. Deb comments that most examples of truss topology optimization which use GA's find first a topology while holding all member cross sections constant, then once a topology is chosen the members are sized. Of course, in using a ground structure, the selection of topology and geometry are linked. That is, when a topology is chosen from the ground structure, the geometry is not altered from that ground structure. These two points certainly limit the quality of the results one can obtain. Deb corrects the short coming of the first point by optimizing each stable topology found in order to determine each member size. In this way the fitness can be based on actual member sizes. But like all methods based on ground structures, it is still limited by the second problem of linked topology and geometry.

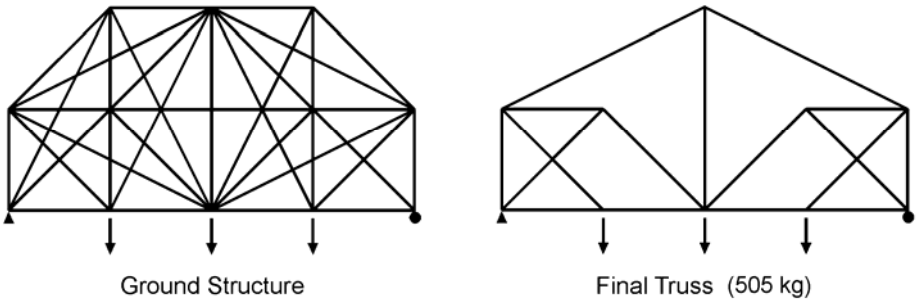


Figure 2: The ground structure (left) and the optimized truss (right) used by Deb & Gulati (1999)

The effect of linking topology with geometry is made apparent in comparing Figure 2 with Figure 3. Figure 3 uses the same topology and load conditions as used in Figure 2, but selects a geometry without the constraints of Deb's ground structure.

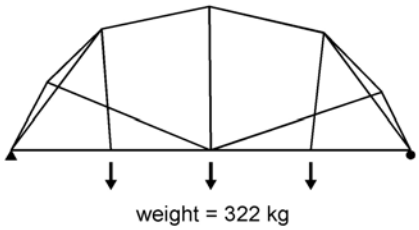


Figure 3: A lighter weight geometry based on the same topology as the one found by Deb & Gulati (1999) in Figure 2. This solution was found without using the ground structure.

Furthermore, Figure 4 shows three other topology/geometry combinations for the same problem, each of which performs better than the solution found in Figure 2. The solutions in Figure 4 were all found without the use of ground structures. It might be added that although buckling is considered in the sizing of the members, each member is assumed to be braced at the joints. This makes the evolution toward many short compression members, as can be seen in the in the arch in the Topo ID 18 of Figure 4, and the longer tension members readily understandable.

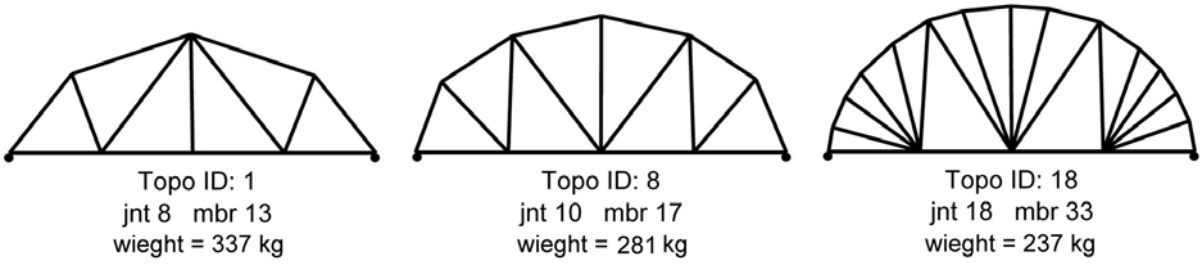


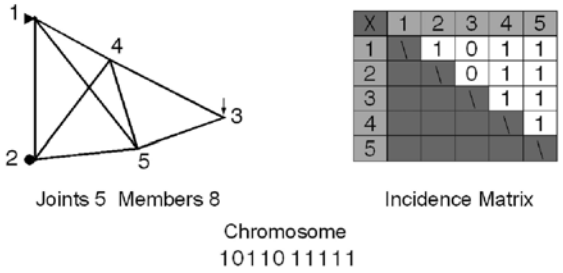
Figure 4: A selection of three other topologies based on the same criteria as used in Figure 2, but without the constraints of a ground structure. A larger set of solutions is shown in reference [5] by von Buelow.

2.2 The computational cost of ground structures in EC

Another aspect of ground structures is the effect the resolution has on the solution. Compare Deb's result using low resolution (12 nodes) with that of Klarbring in Figure 1, which uses the much higher resolution of 210 nodes. Then compare this with the results shown in Figure 4 that do not use a ground structure at all. As the resolution of the ground structure increases, the solutions become more similar to those without a ground structure. Of course the reason Deb chose such a low resolution has to do with the GA coding. The length of the chromosome is based on the total number of admissible members in the ground structure. In Deb's ground structure this number is 39 as counted from the ground structure in Figure 2. The chromosome coding used by the author is based on the upper triangular portion of the incidence matrix. That number equals nodes(nodes-1)/2, or in this case 10(10-1)/2=45. If Deb were to increase the resolution of the ground structure to something on the order of Klarbring's 210, the number of admissible members in his ground structure would go way up to 210(210-1)/2 = 21945. Since the population size in a GA is linked to the chromosome length, an increase from 39 to 21945 would require a significant increase in the population size as well. Both the longer chromosome and the larger population size would slow the computation of the GA to an unacceptable level, and the size of the problem quickly becomes unmanageable. However, since the chromosome coding method used by the author is determined by the actual incidence matrix size, and not the ground structure size, the chromosome remains 45 even when using real numbers for joint locations. In other words, by not using ground structures as a basis for chromosome coding, it is possible to get much higher resolution than Klarbring's example, at about the same cost computationally as Deb's unsatisfactorily low resolution GA.

3 EC chromosome coding without ground structures

Figure 5 shows how the chromosome can be coded based on the incidence matrix. Joints are sorted by location before the chromosome string is extracted, with fixed and loaded joints numbered first. See von Buelow [6].



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

Figure 5: Extraction of the EC chromosome from the incidence matrix

Of course by this method the chromosome lengths for different topologies are often different. In traditional GA crossover breeding this would be a problem. It is easily overcome, however, by simply choosing a cut and crossover point taken from the shorter chromosome string. Figure 6 shows this procedure.

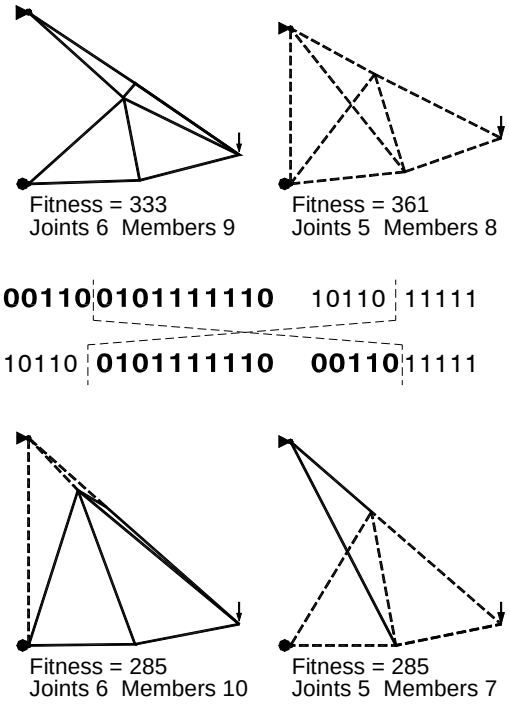


Figure 6: Breeding of two truss topologies using the chromosomes from the incidence matrices

Summary

This paper has shown two major advantages which can be achieved by basing EC chromosome structure on the member incidence matrix rather than a traditional ground structure. These are:

- Refinement and quality of the solution
- Reduction of chromosome length leading to reduction of computation

Further details can be found in the recently published dissertation by the author [7] under the direction of Prof. Werner Sobek (ILEK, Universität Stuttgart). See:
http://elib.uni-stuttgart.de/opus/volltexte/2007/3160/pdf/PvB_diss.pdf

References

[1] Dorn WS, Gomory RE, Greenberg HJ. Automatic design of optimal structures. In Journal de Mecanique 1964; Vol 3, No 1: 25-52.

[2] Klarbring A, Petersson J, Rönnqvist M. Truss Topology Optimization Involving Unilateral Contact - Numerical Results. In: *WCSMO-1: Proceedings of the First World Congress of Structural and Multidisciplinary Optimization*, Olhoff N, Rozvany G (eds). Pergamon: 1995; 129-134.

[3] Goldberg DE. Sizing populations for serial and parallel genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms, Schaffer JD (ed). Morgan Kaufmann, 1989.

[4] Deb K. and Gulati, S. *Design of truss-structures for minimum weight using genetic algorithms*. KanGAL Report No. 99001. Kanpur: Kanpur Genetic Algorithms Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India, 1999.

[5] von Buelow P. Suitability of Genetic Based Exploration in the Creative Design Process. In *Digital Creativity*. Routledge, 2008; Vol. 19, No.1, 51-61.

[6] von Buelow P. Using Evolutionary Algorithms to Aid Designers of Architectural Structures. In *Creative Evolutionary Systems*, Bentley PJ and Corne DW. (eds). Morgan Kaufmann, 2002; 315-335.

[7] von Buelow P. Genetically Engineered Architecture: design exploration with evolutionary computation VDM Verlag Dr. Mueller, 2007.