Discrete Optimization of Trusses using Genetic Algorithms

Carlos Artemio Coello Coello
Department of Computer Science
Tulane University
New Orleans, LA 70118

Abstract

During the last few years, several methods have been developed for the optimal design of structures. However, most of them, because of their calculus-based nature, treat the search space of the problem as continuous, when it is really discrete. This leads to unrealistic solutions and complex processes, and therefore, they are not used in industry, which still prefers to rely on the more traditional iterative methods. This paper proposes the use of genetic algorithms for this task. The results obtained show how good this technique behaves, even when compared to more specialized and sophisticated optimization methods.

Keywords : Genetic Algorithms, Structural Optimization, Discrete Optimization, Structural Design, Truss Design.

Introduction

Structural design is a traditionally sub-optimal process, because it normally relies on the experience of an engineer who uses a computer to iterate through several possible choices of shapes and sizes for each one of the elements of a certain structure. This is just a fast variation of the same computations performed by engineers of the last century, but not a real improvement to the design process.

During the last few years, a lot of work has been done to fully automate the design of structures. Nevertheless, most of the new methods developed have a common problem: they are based on linear programming techniques, and therefore tend to treat structural optimization as a problem in which the search space is continuous, when it's really discrete, because there is only a small number of structural shapes available in the market. In the other hand, some recent structural optimization techniques can deal with discrete search spaces, but they have an inherent lack of generality and therefore can't be readily extended to other kind of structures.

This paper focuses on the use of a search technique called Genetic Algorithm (GA) to optimize the design of plane and space trusses. This technique considers a discrete seach space, yielding more realistic results than linear programming methods, and it's problem independent. This means that the code developed for designing trusses can be reused to solve the remaining framed structures (plane and space frames, plane grids and frames) with little change.

Related Work

Goldberg and Samtani [1] appear to have first suggested the use of GAs for structural optimization. In their paper, they use a GA to optimize a 10-bar plane truss. Deb [2] applied the technique to the design of welded beams, Jenkins to plane frames [3], a trussed-beam roof structure and a thin-walled cross-section [4]; and Rajeev and Krishnamoorthy [5] to generalized trusses.

Pham and Yang [6] have done interesting work on the optimization of multi-modal discrete functions using GAs. Powell [7] described a domain independent design optimization tool for engineers involved with iterative design called EnGENEous. This program uses expert systems and GAs to move from a domain independent system with no knowledge to a domain dependent system with knowledge. It has been used in the design of aircraft engine turbines, cooling fans and molecular electronic structure, and their authors claim that this system has increased engineers productivity tenfold.

Schoenauer and Xanthaquis [8] presented a general method of handling constraints in genetic optimization, based on the Behavioural Memory paradigm. They applied this scheme to test problems of truss structure optimization: a 10-bar (2D) and a 25-bar (3D) truss.

Louis and Rawlins [9] discussed the application of GAs to design structures, but focusing on combinatorial circuit design problems: given a set of logical gates, we want to design a circuit that performs a specified function.

Optimization of a Plane Truss

First, let's define a *plane truss* [10]:

> "A plane truss is idealized as a system of members lying in a plane and interconnected at hinged joints. All applied forces are assumed to act in the plane of the structure, and all external couples have their moment vectors normal to the plane. The loads may consist of concentrated forces applied to the joints, as well as loads that act on the members themselves. For purposes of analysis, the latter loads may be replaced by statically equivalent loads acting at the joints. Then the analysis of a truss subjected only to joint loads will result in axial forces of tension and compression in the members. In addition to these axial forces, there will be bending moments and shear forces in those members having loads that act directly upon them. The determination of all such stress resultants constitutes the complete analysis of the forces in the members of a truss."

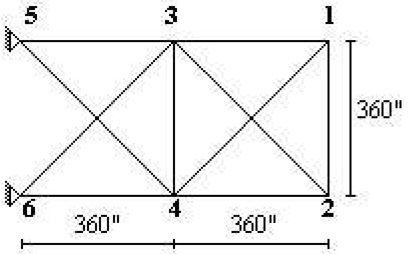

Fig. 1 : 10-bar plane truss used for the first example. Taken from [5].

Now, let's consider a 10-bar truss optimization problem taken from Rajeev and Krishnamoorthy [5], shown in Fig. 1. The objective function of the problem is to minimize the weight of the structure, *f(x)*,

$$f(x) = \sum_{i=1}^{10} \rho A_i L_i \tag{1}$$

where x is the candidate solution, A_i is the cross-sectional area of the ith member, L_i is the length of the ith member, and ρ is the weight density of the material. The assumed data are: modulus of elasticity, E = $1\text{x}10^4$ ksi ($6.89 \text{ x } 10^4$ MPa), $\rho = 0.10$ lb/in^3 (2,770 kg/m^3), and vertically downward loads of 100 kips (445.374 kN) at nodes 2 and 4. Additionally, the truss is subject to the following set of constraints

$$\sigma_i \leq \sigma_a, \text{ for } i = 1 \text{ to } 10$$
$$u_i \leq u_a$$

where σ_i is the stress in member i, σ_a is the maximum allowable stress for all members, u_i is the displacement of each node (horizontal and vertical), and u_a is the maximum allowable displacement for all nodes. These constraints can be expressed in normalized form as

$$\frac{\sigma_i}{\sigma_a} - 1 \leq 0 \; ; \; \frac{u_i}{u_a} - 1 \leq 0$$

The fitness function used was

$$F(x) = \frac{1}{f(x)[1000\, v + 1]} \tag{2}$$

where v is a counter that keeps track of the number of constraints violated by a given solution. We can easily see how when there is no violation to the constraints, the fitness function returns simply the inverse of the weight (this is necessary because the GA only maximizes). As constraints are violated, the fitness is lowered correspondingly. The constant 1000 was determined experimentally.

The additional constraints for this problem are the following: the maximum displacement is 2 inches (50.8 mm) and the stresses are limited to ±25 ksi (172.25 MPa). The list of possible cross sections, taken from the American Institute of Steel Construction Manual [11], is S={1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5} (in^2).

Since there are 10 design variables, and each can take any of the 42 values of the list S, the intrinsic size of the search space is 42^{10} ($\cong 10^{16}$). Six bits are required to represent the 42 available sections ($2^6 = 64$), assigning random values from S to the extra codes. Thus, each chromosome is 60 bits long (6 bits/bar x 10 bars) as shown in Fig. 2.

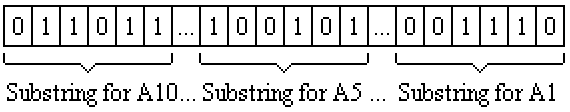

Fig. 2 : Binary representation scheme used to encode a solution.

Comparison of Results

The results produced with the GA are compared with several other structural optimization methods in Table 1. For more detailed information about them, see [12] and [5]. Most of these methods are calculus-based, and therefore their results have to be rounded to the closest available section (this implies by itself a suboptimal solution). However, it's important to point out that Rajeev and Krishnamoorthy [5] use also a GA to optimize the plane truss of this example, with the fitness function

$$F(x) = [\phi(x)_{max} + \phi(x)_{min}] - \phi(x) \tag{3}$$

where $F(x)$ is the fitness of a given solution x, $\phi(x)_{max}$ and $\phi(x)_{min}$ are respectively the maximum and minimum $\phi(x)$ over the population. $\phi(x)$ is

$$\phi(x) = f(x)[1 + KC] \tag{4}$$

$$C = \sum_{i=1}^{n} c_i \tag{5}$$

where n is the number of constraints, the values of c_i are the amounts by which each constraint is violated, $f(x)$ is the weight function as defined in eq. (1) and K is a constant that weights the constraint violations. They found that a value of 10 was appropriate for this example. The results of Table 1 show that a simpler penalty function like the one proposed in this paper can do the work. In fact, it beats the solution produced by Rajeev and Krishnamoorthy [5], even without using lower and upper bounds on each member's values as they propose in order to reduce the size of the search space.

Method	**Weight**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**
OPTDYN	5472.00	25.70	0.10	25.11	19.39	0.10	0.10	15.40	20.32	20.74	1.14
CONMIN	5563.00	25.20	1.89	24.87	15.83	0.10	1.75	16.76	19.73	20.98	2.51
GENETIC	5586.59										
Rajeev	5613.84	33.50	1.62	22.00	15.50	1.62	1.62	14.20	19.90	19.90	2.62
M-3	5719.00	25.84	3.07	26.42	12.77	0.10	3.43	19.34	19.17	18.76	4.42
M-5	5725.00	25.83	2.88	26.45	12.75	0.10	3.77	19.37	19.18	18.77	4.38
GRP-UI	5727.00	24.78	4.17	24.78	14.45	0.10	4.17	17.46	19.26	19.27	5.26
SUMT	5932.00	30.69	2.37	31.62	11.66	0.10	3.71	21.71	20.90	13.97	3.26
LINRM	6249.00	21.57	10.98	22.08	14.95	0.10	10.98	18.91	18.42	18.40	13.51

Table 1 : Comparison of our results (GENETIC) with other techniques reported in the literature for the example of Fig. 1

Optimization of a Space Truss

The second example chosen was a 25-bar truss taken from Rajeev and Krishnamoorthy [5], and shown in Fig. 3. A *space truss* is defined in [10] as follows:

> "A space truss is similar to a plane truss except that the members may have any directions in space. The forces acting on a space truss may be in arbitrary directions, but any couple acting on a member must have its moment vector perpendicular to the axis of the member. The reason for this requirement is that a truss member is incapable of supporting a twisting moment."

Loading conditions for this space truss are given in Table 2, member groupings are given in Table 3, and node coordinates are given in Table 4. The assumed data are: modulus of elasticity, E = 1×10^4 ksi (6.89×10^4 MPa), $\rho = 0.10$ lb/in^3 (2,770 kg/m^3); $\sigma_a = \pm 40{,}000$ psi, $u_a = \pm 0.35$ in. The set of available areas is [5] S = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4} (in^2).

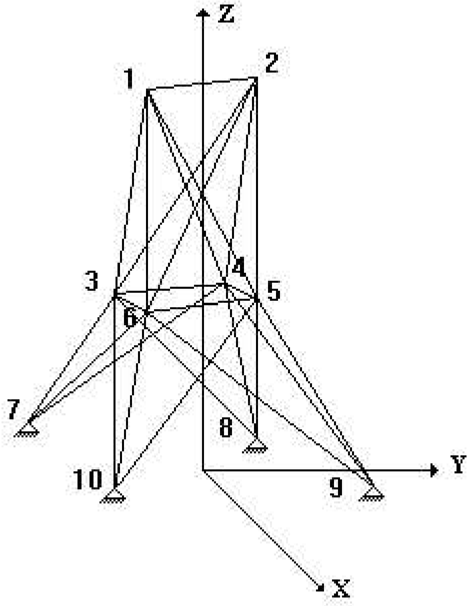

Fig. 3 : 25-bar space truss used for the second example. Taken from [5].

Node	Fx (lbs)	Fy (lbs)	Fz (lbs)
1	1000	-10000	-10000
2	0	-10000	-10000
3	500	0	0
6	600	0	0

Table 2 : Loading conditions for the space truss of Fig. 3. Taken from [5]

Group Number	Members
1	1-2
2	1-4, 2-3, 1-5, 2-6
3	2-5, 2-4, 1-3, 1-6
4	3-6, 4-5
5	3-4, 5-6
6	3-10, 6-7, 4-9, 5-8
7	3-8, 4-7, 6-9, 5-10
8	3-7, 4-8, 5-9, 6-10

Table 3 : Goup membership for the space truss of Fig. 3. Taken from [5].

Node	X	Y	Z
1	-37.50	0.00	200.00
2	37.50	0.00	200.00
3	-37.50	37.50	100.00
4	37.50	37.50	100.00
5	37.50	-37.50	100.00

6	-37.50	-37.50	100.00
7	-100.00	100.00	0.00
8	100.00	100.00	0.00
9	100.00	-100.00	0.00
10	-100.00	-100.00	0.00

Table 4 : Coordinates of the nodes of the space truss of Fig. 3.

Comparison of Results

The results produced with the GA are compared with some other structural optimization methods in Table 5. For more detailed information about them, see [5] and [13]. Rajeev and Krishnamoorthy use a GA as explained before, but get a poorer solution that the one obtained with our suggested fitness function. In fact, our GA even beats continuous methods like the ones from Rizz and Schmit. Finally, it should be pointed out that Zhu' s method deals directly with discrete member sizes, but it can' t be generalized to other framed structures.

Method	Weight	A1	A2	A3	A4	A5	A6	A7	A8
GENETIC	539.78	1.500	0.700	3.400	0.700	0.400	0.700	1.500	3.200
Rizz	545.16	0.010	1.988	2.991	0.010	0.010	0.684	1.676	2.662
Schmit	545.22	0.010	1.964	3.033	0.010	0.010	0.670	1.680	2.670
Rajeev	546.01	0.100	1.800	2.300	0.200	0.100	0.800	1.800	3.000
Zhu	562.93	0.100	1.900	2.600	0.100	0.100	0.800	2.100	2.600

Table 5 : Comparison of our results (GENETIC) with other methods reported in the literature.

Implementation Details

A customized version of the SGA (Simple Genetic Algorithm) presented in [14] was used for the examples of this paper. However, instead of using static arrays, a dynamic memory management technique proposed by Porter [15] was used. Also, binary tournament selection and two-point crossover were preferred over roulette-wheel selection and one-point crossover. The stopping criteria was always a maximum number of generations, and the crossover and mutation probabilities always oscillated around 0.80 and 0.01, respectively. The analysis of each truss was performed using the programs from [16].

The basic operation of the program is the following: the values of the list S are fed into the program (or read from a file). Then, the constraints on maximum allowable stress and deflection are provided. The module that does the structural analysis is executed and the stiffness matrix and its results are stored in a file. The GA is then executed, and the user provides the size of population, number of generations, etc. After that, the program starts iterating, reading and rewriting the file that contains the results of the analysis, and modifying it with values from the chromosomes at each generation. As progress is made, a simplified report is sent to the output device (a laser printer in out case) showing the current generation and the best solution found so far.

Future Work

The final goal of this work is to produce a fully automated structural design system that uses GAs. However, there is still a long way to go. Right now, a solid implementation in Turbo Pascal 7.0 is available for PCs, and a C version is being prepared. Due to the generality of the technique, it' s trivial to extend it to the remaining framed structures (we only have to fit the appropriate structural analysis module into the program). In fact, the technique can also be easily extended to other design tasks, and even to other engineering areas.

Conclusions

The GA seems to be a very good choice for discrete structural optimization, because of its generality and its ability to deal directly with discrete search spaces. Furthermore, the GA operates with several partial solutions simultaneously (this is called implicit parallelism), in contrast with the traditional sequential search of the other methods. Our results show how well they perform even when compared with methods that use continuous search spaces, and are not portable and extremely complex. This does mean, however, that a program that uses this technique will fully replace human engineers in the design process, because a lot of common sense is still required in such a complex task. Nevertheless, GAs should be expected to play a main role in the structural design software of the future.

References

[1] Goldberg, David E. and Samtani, Manohar P. Engineering optimization via genetic algorithm, in *Ninth Conference on Electronic Computation*, New York, N.Y. ASCE, 1986, pp. 471-82.

[2] Deb, Kalyanmoy. Optimal design of a welded beam via genetic algorithms, in *AIAA Journal* 29, 1991, pp. 2013-5.

[3] Jenkins, W. M. Plane frame optimum design environment based on genetic algorithm, in *Journal of Strctural Engineering* 118(11), 1992, pp. 3103-13.

[4] Jenkins, W. M. Towards structural optimization via the genetic algorithm, in *Computers and Structures* 40(5), 1991, pp. 1321-7.

[5] Rajeev, S. and Krishnamoorthy, C. S. Discrete optimization of structures using genetic algorithms, in *Journal of Structural Engineering* 118(5), 1992, pp. 1233-50.

[6] Pham, D. T. and Yang, Y. Optimization of multi-modal discrete functions using genetic algorithms, in *Institute of Mechanical Engineers (Part D)*, 1993, pp. 53-9.

[7] Powell, David; Skolnick, Michael and Tong, S. EnGENEous: domain independent, machine learning for design implementation, in *Third International Conference on Genetic Algorithms*, Schaffer, J. David (Editor), Morgan Kauffman Publishers, 1989, pp. 151-9.

[8] Schoenauer, Marc and Xanthakis, Spyros. Constrained GA optimization, in *Fifth International Conference on Genetic Algorithms*, Morgan Kauffman Publishers, 1993, pp. 573-80.

[9] Louis, Sushil and Rawlins, Gregory. Designer genetic algorithms: Genetic Algorithms in structure design, in *Fourth International Conference on Genetic Algorithms*, Belew, Richard K. and Booker, Lashon B. (Editors), 1991, Morgan Kauffman Publishers, pp. 53-60.

[10] Gere, James M. and Weaver, William. **Analysis of Framed Structures**, D. Van Nostrand Company, Inc., 1965.

[11] Arora, Jasbir S. **Introduction to Optimum Design**, McGraw-Hill Book Company, 1989.

[12] Belegundu, Ashok Dhondu. **A Study of Mathematical Programming Methods for Structural Optimization**, Ph.D. Dissertation, University of Iowa, Dept. of Civil and Environmental Engineering, 1982.

[13] Zhu, D. M. An improved Templeman's algorithm for optimum design of trusses with discrete member sizes, in *Engrg. Opt.* 9, 1986, pp. 303-12.

[14] Goldberg, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**, Addison-Wesley Publising Co., 1989.

[15] Porter, Kent. Handling huge arrays, in *Dr. Dobb's Journal of Software Tools for the Professional Programmer* 13(3), 1988, pp. 60-3.

[16] Coello, Carlos A. **Análisis de estructuras reticulares por computadora (método de rigideces)**, Tesis de Licenciatura, 1991 (in Spanish).