www.lajss.org
Latin American Journal of Solids and Structures 3 (2006) 107–123

Minimum weight design of framed structures using a genetic algorithm considering dynamic analysis

Li Chong Lee Bacelar de Castro* and Paul William Partridge

Dept. Eng. Civil e Ambiental - FT, Universidade de Brasília, Campus Darcy Ribeiro – Asa Norte, Brasília, DF - Brazil

Abstract

Genetic Algorithms can be considered as a family of computational models inspired on the principals of evolution. These algorithms start from a set or population of chromosomes in which each chromosome contains a possible solution to the specific problem. Genetic operators, (selection, crossover and mutation) are applied which generate a new population preserving and improving the information considered in the original chromosomes.

Here the optimization of the cross-sectional area of frame structures is considered in order to obtain the minimum weight for dynamic analysis. The optimization is obtained through the minimization of an objective function, which, for the examples considered is written using the cross-sectional area and the length of the bars and the specific weight of the material from which the total weight of the structure can be obtained. The cross-sectional areas are the project variables which should be within certain limits in order to obtain realistic solutions. Restrictions are applied on the natural frequencies. Results are obtained for some test problems.

Keywords: Genetic Algorithm, framed structures, dynamic analysis, minimum weight.

1 Introduction

A large part of the engineers who design framed structures employ robust commercial software for the structural analysis such as ANSYS or SAP2000N. In general, such users do not have access to the source code or detailed knowledge of the structural analysis algorithms employed in this software. On the other hand, the biggest challenge for researchers in structural optimization is the development of methods which are appropriate for use with the commercial codes mentioned above. Another important challenge is the high computational cost associated with the more complicated optimization problems encountered. This is all the more serious given that the project engineer who designs the structure, does not have time to analyse it exhaustively. Given the above it is of practical interest to find techniques which minimise the need for commercial software and require as few iterations as possible.

*Corresp. author Email: lichonglee@unb.br Received 10 Nov 2005; Revised: 24 Feb 2006

An interesting approach to such problems is the use of a Genetic Algorithm (GA) for the optimisation combined with the finite element method for evaluating the configurations encountered. In this way an open code can be written. Genetic Algorithms have the advantages of not needing derivatives or other auxiliary information, starting only from the initial population. The use of the binary coding of the parameters rather than the parameters themselves improves accuracy. In addition, using the GA, there is a better chance of obtaining a true global minimum.

The use of Genetic Algorithms for optimization in science and engineering is widespread, applications having been made for example to problems such as the design of turbines for aircraft and even recently in anlyses of the structure of proteins. A review paper, considering 4300 articles is given in reference [13].

The GAs have been applied to static structural weight optimization of frames since 1989, [1, 10–12, 19], and more recently some applications to dynamic analysis have been considered, including damage detection, [5, 16, 23, 30], support location and the location of sensors and actuators, [20–22].

In structural design for which the comfort of users is to be taken into account, it is necessary that the structure has a specific dynamic response, for example a limitation on the range of the natural frequencies. This is an important area of research particularly in relation to weight optimization, [28].

Here, after a description of the algorithm used, numerical results are given for some examples employing different frequency constraints.

2 Description of the computer code employed

The computer code considered here was implemented using FORTRAN 90. The Genetic Algorithm employed is described below, followed by some considerations on dynamic analysis and the penalty function used to impose the frequency constraints. A block flow diagram is given in Fig. 1.

2.1 Encoding of possible solutions

There are several different alternatives for mapping a finite length string to the parameters of an optimization problem. Here a binary coding is employed, though real numbers may also used. An important consideration is that the coding be capable of representing the entire search space being investigated, that is, any solution should be capable of being represented by a chromosome, and reciprocally, any chromosome generated by the GA should be associated with a valid solution for the problem under consideration. The chromosome contains a discrete or continuous value of one or more parameters, each of which is called an allele.

The number of bits to be used to represent each variable defines the size of the allele, and depends on each problem. For a search space with continuous variables, one has

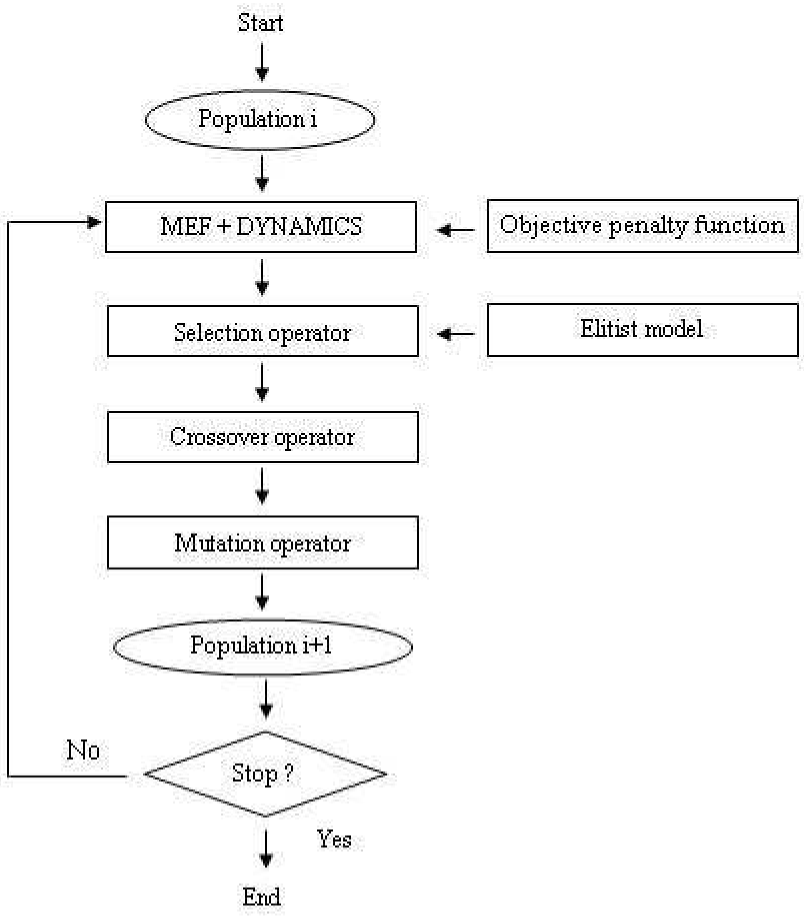

Figure 1: Block flow diagram for Genetic Algorithm employed

$$\ell \geq \log_2 \frac{x^U - x^L}{\epsilon} \tag{1}$$

where ℓ is the length of the string, or the number of bits used for each allele in the chromosome. One allele in the chromosome with this length is capable of representing 2^ℓ values uniformly distributed over the range $\left[x^L, x^U\right]$ with a interval ϵ between each value.

2.2 Mapping of variables

The chromosome as shown in Fig. 2, consists of a set of alleles. In order to recuperate the value contained in an allele with binary coding a decoding procedure is necessary. In the case of continuous values, Eq. (2) is employed.

$$x_i = x^L + \mathrm{ID}\frac{x^U - x^L}{2^\ell - 1} \tag{2}$$

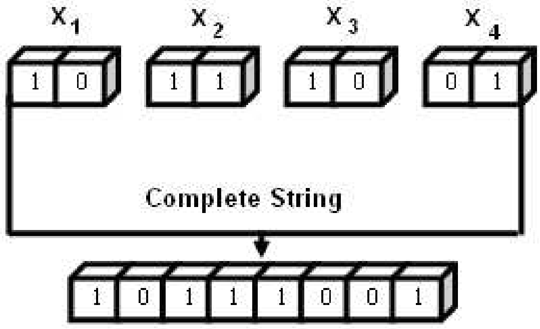

Figure 2: Binary code string for chromosome

Consider for example the first allele in Fig. 2, the value of which is $x_1 = 10$ in binary notation, with $\ell = 2$. Consider also that the search space for this variable has lower limit $x^L = 20$ and upper limit $x^U = 50$, respectively. Considering ID to be the value of the binary code, in this case $ID = 1{\times}2^1 + 0{\times}2^0 = 2$. If one substitutes the values given above in Eq. (2), one obtains $x_i = 40$ which is the decoded value of the first allele in the interval $[20; 50]$. For this example, as $\ell = 2$ the binary code may take 4 values, 00, 01, 10 and 11 corresponding to the decoded values 20, 30, 40, 50. The value of ϵ in Eq. (1) is thus 10.

2.3 Population size

The population size (n_{pop}) indicates the total number of chromosomes to be considered for a given problem. With an insufficient population size, the GA will probably not perform satisfactorily, because a too small population is unable to represent the problem search space correctly. On the other hand, a large population will be able to represent a large and diverse number of solutions, however as the population size increases so does the computer cost necessary to carry out the repeated evaluations of the fitness function. In this way the population size has a direct influence on the overall performance and efficiency.

There are as yet no well defined criteria for fixing the population size for a given case, the choice will depend on the experience acquired in the optimization process and on the problem itself, [12].

Khan, [17] suggests that the population size for the great majority of problems should be between ℓ and 2ℓ

$$\ell \leq n_{pop} \leq 2\ell \tag{3}$$

2.4 Evaluation of the chromosomes

According to Ochi and Rocha, [27], the evaluation of the chromosomes is the most important part of the process, it is here that one finds the connection between the algorithm and the problem. In order to evaluate a chromosome in a GA, it is necessary to calculate its fitness for survival in relation to the other individuals in the population. At this stage the first step in the selection process is carried out: each individual is associated with a numerical value which represents its degree of fitness, obtained using the objective function (OF).

The value of the objective function is not always adequate for use as a fitness parameter. If the OF produces values which are very close to each other, the selection process becomes random. On the other hand, a very high value in relation to the rest of the population causes the algorithm to converge too soon. For this reason the fitness is obtained by ordering the values calculated by the objective function. Amongst the methods available, the linear scale introduced by Baker, [2], is most often encountered in the GA literature. The modified objective function (AF_i) is given by

$$AF_i = \min + (\max - \min)\frac{n_{pop} - i}{n_{pop} - 1} \tag{4}$$

The value of i is the position of the chromosome in decreasing order of the value of the objective function and n_{pop} is the number of individuals in the population. Here, the values min = 1 and max = 2 were taken.

2.5 Selection of chromosomes

At this stage the fittest individuals from the present generation are selected. These individuals are used to generate a new population by crossover and each individual has a probability of being selected which is proportional to its fitness. This method is known as roulette wheel selection.

Given a population of n_{pop} individuals, the probability of selection p_i in Eq. (5), for each chromosome i with fitness f_i, is given by Goldberg, [12], as:

$$p_i = \frac{f_i}{\sum_{j=1}^{n_{pop}} f_j} \tag{5}$$

In order to visualise roulette wheel selection, consider the circle shown in Fig. 3 representing a population of size $n_{pop} = 4$.The circle is divided into n_{pop} regions, in such a way that the area of each region is proportional to the fitness of each individual.

The individuals in regions with larger areas will have a larger probability of being selected more than once. As a consequence, the selection of individuals may contain several copies of the same high fitness individual, while others will disappear.

It is possible that in the process of evolution a high fitness individual is eliminated in the process of crossover or mutation. For this reason the elitist model is used, in which it is guaranteed that the best individual passes to the next generation without modification.

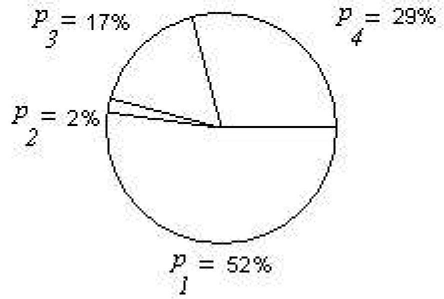

Figure 3: Roulette wheel: probabilities of the selection of each chromosome

2.6 Probability of crossover

This parameter indicates the probability of an individual being mated with another. After the selection of the new set of chromosomes by the roulette wheel described above, pairs of individuals are chosen randomly for mating. This is done by exchanging part of the two strings to form a new pair, a process known as crossover. The probability of an individual being chosen is p_c.

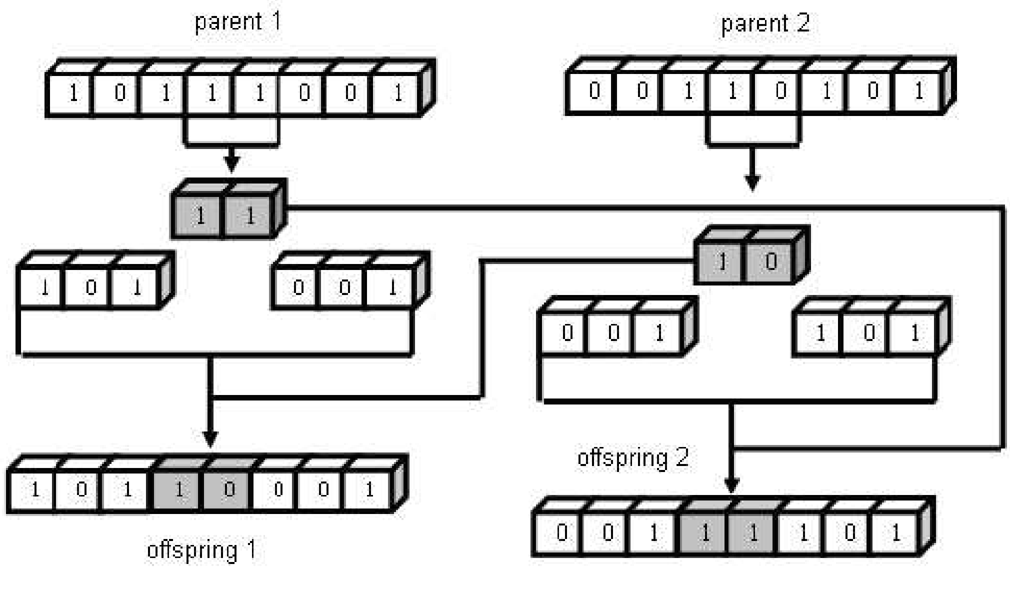

Figure 4: Two point crossover

Crossover is an important mechanism of the GA since it permits the exchange of genetic material between the two individuals involved producing two new points in the optimization space. The two point crossover, shown in Fig. 4, works as follows. Two random positions, the same for each of the two strings, are chosen. The part of the string between these two positions is then exchanged to form two new individuals which will then receive a value for individual fitness.

The higher the probability p_c, the more rapidly new individuals are introduced into the population. On the other hand, if this probability is too high, individuals with high levels of fitness will be removed quickly from the population, which represents a loss of good solutions. Low values can make the algorithm slow to converge. Usually a probability of crossover between 0.50 and 0.95 is used. These numbers are only a guideline, given that there are many types of crossover, limited only by the creative capabilities of the researcher. Here $p_c = 0.80$.

2.7 Probability of mutation

The probability of mutation p_m, is the chance that the binary value of a given position in the chromosome be changed. Mutation is employed in order to introduce new information into the population. Without this feature, after many crossovers, the population may be saturated with similar chromosomes. Employing mutation the diversity of the population increases and new points in the search space are found. High probabilities of mutation make the strings become random. For this reason some researchers recommend the choice of a probability of mutation related to the size of the chromosome and to the total population. Hesser and Manner, [15], suggest that an optimum probability of mutation can be found using the expression

$$p_m = \frac{1}{n_{pop}\sqrt{\ell}} \tag{6}$$

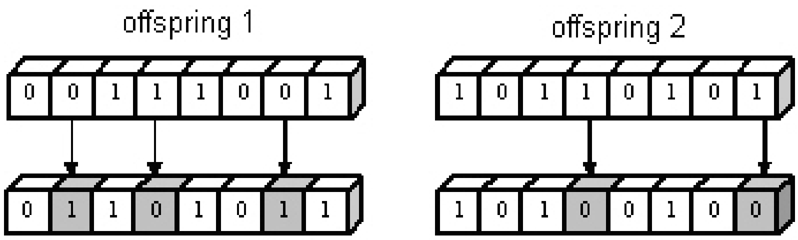

Figure 5: Mutation operator

Mutation is carried out on the individuals which result from the crossover process (offspring). Each bit of the chromosome is considered with a probability of change of p_m. If the change is carried out, a bit originally 0 will become 1 and vice versa. Fig. 5 illustrates the basic procedure adopted for the mutation operator. As is the case for the other parameters, the probability of mutation depends on the problem being considered.

2.8 Stopping criterion

At this stage, the individuals resulting from the process of crossover and mutation are inserted into the new population. The population continues to be of the same size; the same number of individuals were created as in the previous population, substituting this completely, [25], in such a way that this algorithm can be classified as generational.

The process is stopped by a test to establish if the GA has reached a predetermined stopping point. Here this point is given when 98% of the chromosomes have the same value.

2.9 Dynamic analysis

In order to evaluate each of the possible solutions obtained by the Genetic Algorithm a dynamic analysis using finite elements is carried out as shown in the block flow diagram given in Fig. 1. This analysis has the basic objective of determining the natural frequencies, the vibration modes may be calculated optionally. The damping which normally exists in a structure is considered to be relatively small and does not practically effect the calculation of the natural frequencies or the vibration modes, [4, 6, 7, 29]. The consistent mass and stiffness matrices used here were obtained from [32]. Here the generalized Jacobi method is employed given that both the mass and stiffness matrices are symmetric, [32]. Here in the optimisation problem, natural frequency constraints are considered as explained below.

2.10 Penalty methods

A problem with constraints can be efficiently transformed into a problem without constraints using a penalty function, [9,14,31]. The penalty function is used to modify the objective function in the following way:

$$OF(A_i) = f(A_i) + \text{Penal}(x) = \sum_{i=1}^{n} \rho A_i L_i + \text{Penal}(x) \tag{7}$$

Classical applications of penalty functions to impose constraints can be classified as direct or interior, in which only feasible solutions are considered, and indirect or exterior in which any solution is used in the optimization process. However since the 1990s these techniques have been classified as static, dynamic and adaptive penalty methods, [18, 26].

Here the static penalty method as considered below is employed for simplicity. In the static method the objective function is defined as a fixed part plus the product of a constant chosen by the designer and the sum of the residuals of each violated constraint.

$$OF(A_i) = f(A_i) + \alpha \cdot \sum_{i=1}^{n} S_i \tag{8}$$

For the present case including natural frequency limitations, the penalty function took the form given below:

$$F(A_i) = \sum_{i=1}^{n} \rho A_i L_i + \alpha \sum_{j=1}^{m} \left[\frac{|w_j|}{w_{\max}} - 1 \right] \tag{9}$$

where n is the number of bars considered, m is the number of frequency restrictions and ω is a natural frequency.

3 Numerical examples

Three examples are considered, the first two being traditional frame structures and the third being a real structure, [8], represented by a frame with some pin joints

Each problem was solved four times, once for each of the frequency constraints shown in Table 1. The frequency constraints considered are based on Castro, [3].

Table 1: Natural Frequency constraints for problems considered

Case	Characteristics
I	First natural frequency $\omega_1 = 5$Hz
II	Second natural frequency $\omega_2 = 15$ Hz
III	Third natural frequency $\omega_3 = 30$ Hz
IV	First natural frequency $\omega_1 \geq 5$ Hz Second natural frequency $\omega_2 \geq 15$ Hz Third natural frequency $\omega_3 \geq 30$ Hz

Table 2 shows the material parameters for all of the cases considered. MacGee and Phan, [24] propose that for structural problems which involve continuous sections, the moment of inertial should be a function of the area of the section.

Table 2: Material Parameters for all of the problems considered

Parameter	Value
Cross-sectional area:	Lower limit $x^L = 0.1\text{in}^2$ Upper limit $x^U = 100.0\text{in}^2$
Youngs Modulus (E)	3 $\times 10^7$ Ksi
Density (ρ)	0.283lb/in^3
Moment of Inertia (I) (McGee and Phan, [35])	for $0.1 \leq A_i \leq 44.2\text{in}^2$ $I = 4.62 * A_i^2$ for $44.2\text{in}^2 \leq A_i \leq 100.0\text{in}^2$ $I = 256 * A_i - 2300$

3.1 Problem 1: plane 6-bar frame

The structure considered is shown in Fig. 6. In this case, the cross-sectional areas of each of the six bars of which the frame is composed are the project variables, in such a way that the chromosomes posses 6 alleles.

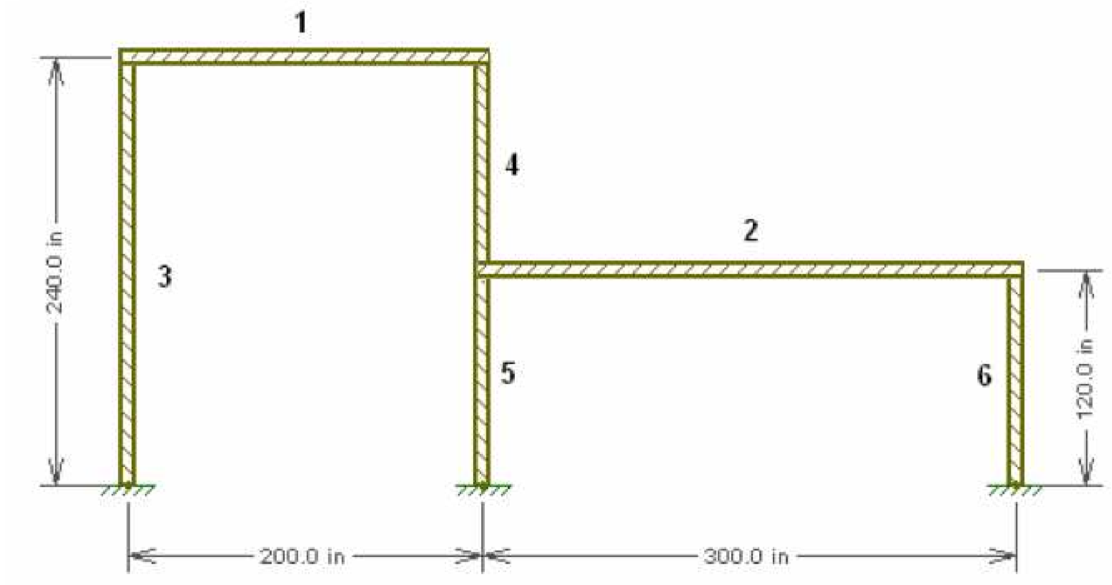

Figure 6: Problem 1: 6-bar plane frame

The size of the alleles of the chromosomes is determined using Eq. (1). Considering $\epsilon = 0.1$, and the maximum and minimum values for the cross-sectional area as given in Table 2, each variable cross-sectional area is represented using 10 bits, given that there are six bars, and thus 6 alleles, the chromosome is of size 60 bits.

Using Eq. (3) and Eq. (6) one obtains the total number of individuals in the population, n_{pop} and the probability of mutation p_m respectively,

$$n_{\text{pop}} = 100$$

$$p_m = 0.001$$

The results obtained by the program for the first three natural frequencies for each of the four cases of frequency constraints given in Table 1 are shown in Table 4. The GA solution, show in Table 3, gives the areas of the bars for each of the four constraints considered. A solution is obtained for the natural frequencies using the SAP2000N commercial code. The cross-sectional areas of the bars considered are those given by the GA, the other information required is input

Table 3: Results for Cross-sectional areas for Problem 1, (in^2)

	Case 1	Case 2	Case 3	Case 4
Section 1	5.90	6.63	6.12	10.60
Section 2	5.90	6.58	6.70	10.80
Section 3	19.20	18.00	13.00	10.50
Section 4	19.20	18.20	12.20	10.50
Section 5	5.60	6.80	5.88	2.90
Section 6	2.80	1.50	10.90	3.00
Total Weight (lb)	3079.95	3056.40	2783.71	2791.17

data. From this point the graphical capabilities of this program can be taken advantage of. The solutions obtained for the four cases considered converged at generation numbers 756, 1032, 1422 and 823, respectively. Table 3 also presents the minimum weight of the structure for each of the four cases considered. The best solution was found for case 3 in which a constraint is placed on ω_3.

Table 4: Results for natural frequencies for Problem 1, (Hz)

	Case 1		Case 2		Case 3		Case 4	
frequency	GA	SAP	GA	SAP	GA	SAP	GA	SAP
ω_1	5.00	4.96	9.72	9.51	22.38	21.97	7.21	6.99
ω_2	7.44	7.12	14.87	14.20	25.87	25.06	22.11	21.92
ω_3	10.48	10.01	20.20	20.01	29.53	28.99	32.28	31.02

3.2 Problem 2: plane 14-bar truss

The structure considered is shown in Fig. 7. In a similar way to the previous problem, there are now 14 alleles per chromosome. The number of bits in each allele is found in the same way as in the previous problem, giving a chromosome of 140 bits. The population size n_{pop} and the probability of mutation, p_m will then be

$$n_{\text{pop}} = 200$$
$$p_m = 0.0004$$

The results obtained by the program for the first three natural frequencies are shown in Table 6 for the four different frequency constraint cases considered. Comparison is made with

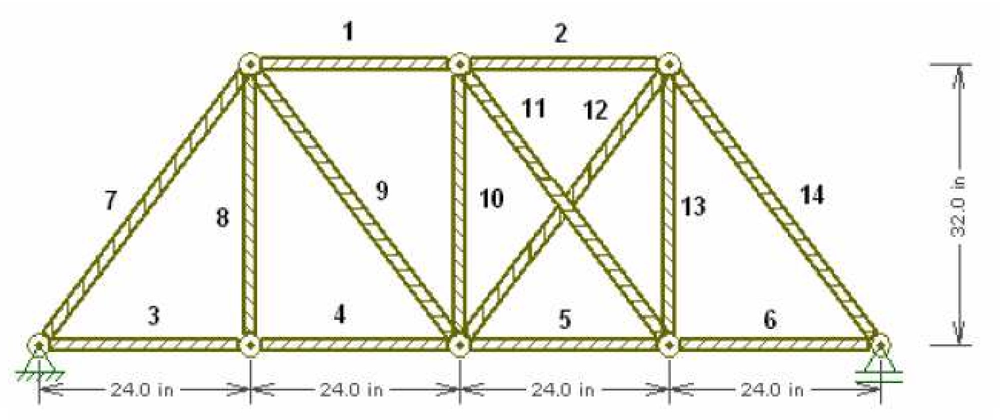

Figure 7: Problem 2: 14-bar plane truss

results obtained using the commercial software SAP2000N, as in the previous problem. For this problem, the minimum weight solutions for the four cases considered were obtained at generation numbers 926, 1256, 1223 and 1822. The values obtained for the cross-sectional areas of the bars are given in Table 5, for each case, together with the minimum weight of the frame. The best solution was found for case 2 in which a constraint is placed on ω_2.

3.3 Problem 3: industrial building

The structure considered in this case is shown in Fig. 8 and is a real problem analysed statically by Erbatur, [8]. The frame comprises 29 bars with pin-joints at the places indicated. For this problem the bars are considered to be grouped together, each group shown in Table 7 has the same cross-sectional area.

Given this, only 12 independent sections need to be obtained, and the chromosome has 120 bits, less than in the previous problem. Values for population size, n_{pop} and the probability of mutation, p_m are then given by

$$n_{\text{pop}} = 200$$
$$p_m = 0.004$$

The solutions for the four sets of frequency constraints considered were given at generation numbers 1052, 1832, 1364 and 2051 and the results for the cross-sectional areas of the bars, the total weight of the structure and the first three natural frequencies are given in Tables 8 and 9 respectively in the same way as in the previous cases. It is observed that if several frequencies are taken into consideration, as in case IV, a lighter structure is obtained than if only one frequency is considered.

Table 5: Results for cross-sectional areas for Problem 2, (in^2)

	Case 1	Case 2	Case 3	Case 4
Section 1	12.56	8.33	3.51	4.33
Section 2	1.23	7.00	2.78	4.00
Section 3	8.33	5.60	11.65	3.20
Section 4	9.25	1.38	5.23	3.18
Section 5	9.66	5.28	3.28	2.28
Section 6	9.51	2.31	8.54	3.30
Section 7	0.42	0.88	3.48	6.89
Section 8	0.63	1.62	3.02	4.89
Section 9	6.21	0.79	4.06	1.89
Section 10	9.57	6.33	1.32	3.23
Section 11	3.58	4.66	0.98	0.68
Section 12	6.32	0.98	8.23	0.54
Section 13	0.62	1.64	6.12	4.66
Section 14	0.22	2.62	3.55	6.66
Total Weight (lb)	630.86	402.34	562.17	442.14

4 Conclusions

The minimum weight of frame structures for free vibration analysis is considered using a Genetic Algorithm of the generational type using binary coding for the chromosomes and employing the elitist model. The constraints on natural frequency are considered using a modified objective function in which a penalty term is added. The penalty function employs parameters which remain constant during the evolutionary process.

The minimum weight for the structure was obtained by the algorithm for three problems, for each problem four different cases of frequency constraints are considered. The natural frequencies

Table 6: Results for natural frequencies for Problem 2, (Hz)

	Case 1		Case 2		Case 3		Case 4	
frequency	GA	SAP	GA	SAP	GA	SAP	GA	SAP
ω_1	4.81	4.31	12.13	11.95	19.89	19.23	7.03	6.57
ω_2	7.65	7.25	14.53	14.10	24.68	23.88	15.53	14.99
ω_3	13.07	12.89	20.45	19.11	30.00	29.60	31.66	29.86

Table 7: Groups of sections for Problem 3

$A_1 = A_{24}$
$A_2 = A_{23}$
$A_3 = A_{26}$
$A_{28} = A_{29}$
$A_4 = A_5 = A_{22} = A_{25}$
$A_7 = A_8 = A_9 = A_{10}$
$A_{11} = A_{12} = A_{13} = A_{14}$
$A_{15} = A_{21}$
$A_{16} = A_{20}$
$A_{17} = A_{19}$
A_{18}

Table 8: Results for cross-sectional areas for Problem 3, (in^2)

	Case 1	Case 2	Case 3	Case 4
Sections 1,24	10.61	4.33	11.20	8.30
Sections 2,23	13.70	20.50	14.30	10.10
Sections 3,26	4.74	1.11	3.10	2.56
Sections 28,29	8.00	5.24	9.80	13.00
Sections 6,27	5.55	3.30	2.03	3.55
Sections 4,5,22,25	6.59	3.28	2.55	3.25
Sections 7,8,9,10	2.24	4.89	3.20	0.68
Sections 11,12,13,14	6.21	4.66	1.00	1.35
Sections 15,21	0.92	0.38	1.23	0.98
Sections 16,20	0.69	2.00	0.64	1.32
Sections 17,19	1.12	1.72	2.30	1.11
Section 18	4.53	3.36	2.63	2.75
Total Weight (lb)	10171.33	8020.62	7510.12	6873.05

obtained are compatible with results obtained by the commercial code SAP in each case. For the plane frame, Problem 1, the minimum weight is obtained with the restriction on ω_3. For the truss, Problem 2, the minimum weight is obtained with the restriction on ω_2. For the industrial building, Problem 3, the minimum weight is obtained with restrictions on several frequencies.

It is considered that the results presented show the versatility and robustness of the algorithm, and that this can be used by designers in order to obtain initial values for cross-sectional

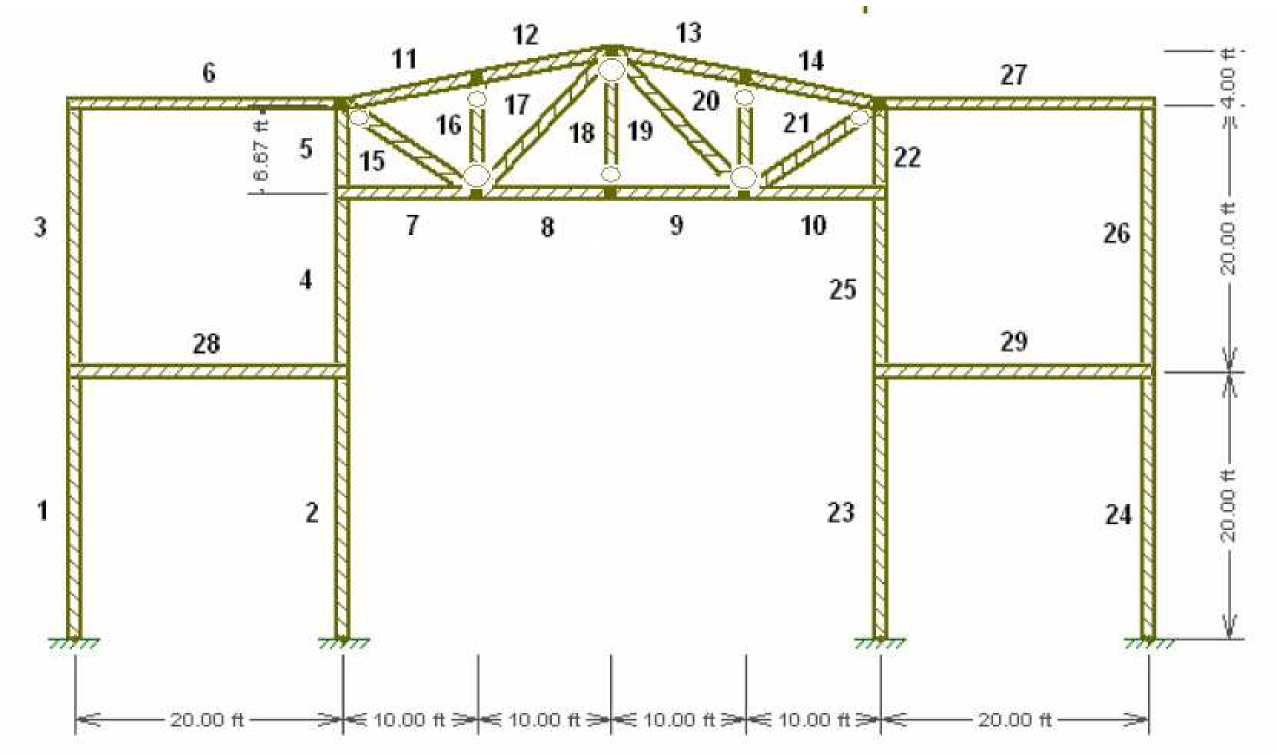

Figure 8: Problem 3: Industrial Building

Table 9: Results for natural frequencies for Problem 3, (Hz)

	Case 1		Case 2		Case 3		Case 4	
frequency	GA	SAP	GA	SAP	GA	SAP	GA	SAP
ω_1	4.92	4.42	12.13	12.01	19.31	19.20	6.12	5.89
ω_2	11.20	10.80	14.87	14.53	25.20	24.92	19.65	19.31
ω_3	19.89	19.69	19.53	19.20	29.27	29.03	33.44	32.99

areas for steel structures.

Acknowledgments: The authors would like to acknowledge the financial support of the brazilian agency CNPq.

References

[1] N. Ali, K. Behdinan, and Z. Fawaz. Applicability and viability of GA based finite element analysis architecture for structural design optimization. *Computers & Structures*, 81:2259–2271, 2003.

[2] J. Baker. Reducing bias and inefficiency in selection algorithm. *Proceedings of The Second International Conference on Genetic Algorithm and Their Application*, pages 14–21, 1987.

[3] L. L. B. Castro. Genetic algorithm for the optimization of bar structures. Master's thesis, University of Brasilia, 2005. In Portuguese.

[4] A. K. Chopra. *Dynamics of Structures - Theory and Applications to Earthquake Engineering.* Pretice Hall, New Jersey, USA, 1995.

[5] J-H Chou and J Ghaboussi. Genetic algorithm in structural damage detection. *Computers and Structures*, 79:1335–1353, 2001.

[6] R. W. Clough. *Dynamics of Structures.* McGraw-Hill, New York, USA, second edition, 1993.

[7] R. R. S. Craig Jr. *Structural Dynamics - An Introduction to Computer Methods.* John Wiley & Son, New York, USA, 1981.

[8] F. Erbatur, I. T. Hasançebi, and H. Kiliç. Optimal design of planar and space structures with genetic algorithms. *Computer & Structures*, 75(2):209–224, 2000.

[9] R. L. Fox. *Optimization Methods for Engineering Design.* Addison-Wesley, London, UK, 2nd edition, 1973.

[10] M. Galante. Structures optimization by a simple genetic algorithm. *Numerical Methods In Engineering And Applied Sciences*, pages 863–870, 1992.

[11] M. Galante. Genetic algorithms as an approach to optimize real-world trusses. *International Journal for Numerical Methods In Engineering*, 39:361–382, 1996.

[12] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[13] D. E. Goldberg, K. Zakrzewshi, B. Sutton, R. Gadient, C. Chang, P. Gallego, B. Miller Miller, and E. Cantú-Paz. Genetic algorithms: A bibliography. Technical Report No. 97011, 1999.

[14] R. T. Hafka and Z. Gurdal. *Elements of Structural Optimization.* Kluwer Academic Publishers, London, U.K, 3th edition, 1992.

[15] J. Hesser and P. Manner. *Pratical Guide to Genetic Algorithms. Naval Research Laboratory.* Chemistry Division, 2000.

[16] R. J. Jiang, F. T. K. Au, and Y.K. Cheung. Identification of masses moving on multi-span beams based on a genetic algorithms. *Computers and Structures*, 81:2137–2148, 2003.

[17] N. Khan. *Population Sizing in Genetic and Evolutionary Algorithms - Lecture Notes.* University of Illinois at Urbana-Champaign, 2002.

[18] N. D. Lagaros, M. Papadrakakis, and G. Kokossalakis. Structural optimization using evolutionary algorithms. *Computer and Structures*, 80:571–589, 2002.

[19] A. C. C. Lemonge and H. J. C. Barbosa. An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical methods in engineering*, 59:703–736, 2004.

[20] Q. S. Li, D. K. Liu, J. Q. Fang, and C. M. Tam. Multi-level optimal design of buildings with active control under winds using genetic algorithms. *Journal of Wind Engineering and Industrial Aerodynamics*, 86:65–86, 2000.

[21] Q.S. Li, D. K. Liu, J. Tang, N. Zhang, and C. M. Tam. Combinatorial optimal design of number and positions of actuators in acively controlled structures using genetic algorithms. *Journal of Sound and Vibration*, 270:611–624, 2004.

[22] D. K. Liu, Y. L. Yang, and Q. S. Li. Optimum positioning of actuators in tall buildings using genetic algorithms. *Computers and Structures*, 81:2823–2827, 2003.

[23] C. Mares and C. Surace. An application of genetic algorithms to identify damage elastic structures. *Journal of Sound an Vibration*, 192(2):195–215, 1996.

[24] O. G. McGee and K. F. Phan. A robust optimality criteria procedure for cross-sectional optimization for frame structures with multiple frequency limits. *Computers And Structures*, 38(5):485–500, 1991.

[25] Z. Michalewics. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-verlag, Berlin, Germany, third edition edition, 1996.

[26] Z. Michalewicz. Genetic algorithms, numerical optimization, and constraints. *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 151–158, 1995.

[27] L. S. Ochi and M. L. Rocha. A new hybrid evolutionary algorithm for the vehicle routing and scheduling problems. *Proceedings of the Ninth International Conference on Intelligence Systems: Artificial Intelligence Applications for the New Millennium*, pages 135–140, 2000.

[28] AISC American Institute of Steel Construction. *Manual of steel construction: Allowable stress design*. Chigaco, USA, 9th edition.

[29] M. Paz. *Structural Dynamics: Theory and Computation*. Van Nostrand Reinhold, New York, 1980.

[30] M.A. Rao, J. Srinivas, and B. S. N. Murthy. Damage detection in vibrating bodies using genetic algorithms. *Computers and Structures*, 82:963–968, 2004.

[31] M. Shoenauer and S. Xanthakis. Constrained GA optimization. *Proceedings of The 4th International Conference on Evolutionary Programming*, pages 573–580, 1993.

[32] W. Weaver Jr. and P. R. Johnston. *Structural Dynamics by Finite Elements*. Pretice Hall, Englewood Cliffs, New Jersey, 1987.