OPTIMUM SOLUTIONS FOR TRUSSES USING SEQUENTIAL QUADRATIC PROGRAMMING AND GENETIC ALGORITHMS

Silvana Maria Bastos Afonso
André Andrade Azevedo
Universidade Federal de Pernambuco, Centro de Tecnologia, Departamento de Engenharia Civil,
50740-000, Recife, PE, Brasil. E-mail: smb@npd.ufpe.br

Abstract

In this work computational tools are developed to carry out sizing optimization of truss under static and free vibration conditions. To do that an automatic procedure is implemented on MATLAB environment. Different types of algorithms such as gradient-based SQP methods and GAs are used. Some benchmark examples are analyzed.

Key words: Truss optimization, Genetic Algorithms, SQP

1. INTRODUCTION

The analysis and design of trusses has been a classical problem in structural optimization. Typical designs of trusses require a minimization or maximization of a stated objective function and simultaneous satisfaction of several design constraints. For several decades a great variety of methods have been developed to find the size of the elements that optimize trusses under statics loading with a given geometry and topology.

The nonlinear programming algorithms such as sequential quadratic programming (SQP) investigated here are gradient-based and requires the first derivative of the objective function and constraints with respect to the design variables.

The SQP algorithm is extremely efficient in locating a relative optimum closest to the starting point in the design space. In design applications where the design space is known to be multi-modal, the optimum may be obtained by starting the search from several initial points in the design space (Hajeta,1990). However, even then, there is no guarantee of obtaining the global optimum. To overcome the possibility of local optima in the present study, we investigate here a stochastic method based on genetic algorithm.

Genetic algorithms (GAs) are search procedures based on the mechanics of genetics and natural selection. Although computationally simple, GA-based methods are very powerful in their search for improvement and they are not limited by restrictive assumptions above the search space. They overcome the possibility of local optima in the solution process. In truss optimization GAs are very versatile as they accept both discrete and/or continuous design variables. The genetic algorithm developed by D. Goldberg, described in (Goldberg, 1989) is implemented here. Such algorithm have been applied to a wide range of engineering disciplines and has proven to give very good results in several applications.

In this work both SQP and GA algorithms are employed to carry out the optimization of trusses under static and free vibration conditions. For comparisons purposes with SQP algorithm, only continuos design variables are considered in our applications.

2. GENERAL PROBLEM FORMULATION

The standard mathematical formulation of an optimization problem is:

minimize or (maximize) $F(\mathbf{s})$ *subject to:* $g_i(\mathbf{s}) \leq 0 \qquad i = 1...n_c$ (1)

$\mathbf{s}_k^{\,l} \leq \mathbf{s}_k \leq \mathbf{s}_k^{\,u} \qquad k = 1...\text{ndvab}$

Where **s** is the design variable vector, ndvab is the total number of design variables $F(\mathbf{s})$ is the objective function, g_i is a typical constraint and $\mathbf{s}^l$ and $\mathbf{s}^u$ are side constraints to the design variables. Typical objective functions and constraints considered in this work are shown in figure 1.

TRUSS OPTIMIZATION

Problem type 1: weight minimization

Objective function - $W(\mathbf{A})$

Constraints:	$\lvert\sigma_i\rvert / \sigma_{max} - 1 \leq 0$	i = 1...m
	$\lvert u_j\rvert / u_{max} - 1 \leq 0$	j = 1...n
	$A_{min} \leq A^{(i)} \leq A_{max}$	i = 1...m

Problem type 2: Fundamental frequency Maximization

Objective function - $\omega(\mathbf{A})$

Constraints:	$V = V_0$	
	$A_{min} \leq A^{(i)} \leq A_{max}$	i = 1...m

Figure 1 - Types of problems considered in this work: objectives and constraints.

3. THE TRUSS OPTIMIZATION PROCEDURE

In this work an integrated procedure is developed and implemented on MATLAB environment. SQP and GA algorithms are employed to obtain optimum designs. When dealing with GAs the procedure automatically integrates geometry definition and discretization, FE analysis and method of optimization solution, when SQP is used instead, a part from those, a sensitivity analysis procedure is incorporated in the process. Details of each integrated module can be found elsewhere (Afonso and Horowitz, 1998).

4. THE SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM

The SQP algorithm generates a search direction $\boldsymbol{d}$ at each iteration solving the definite quadratic subproblem, below:

Minimize (or maximize): $\nabla F_k^T \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^T \boldsymbol{B}_k \boldsymbol{d}$ (2)

Subject to: $\nabla \boldsymbol{g}_k^T \boldsymbol{d} + \boldsymbol{g}_k \leq 0$ (3)

$\nabla \boldsymbol{h}_k^T \boldsymbol{d} + \boldsymbol{h}_k = 0$ (4)

where $\boldsymbol{B}_k$ is a positive definite approximation to the Hessian matrix of the Lagrangian function of the original problem (Han, 1976).

$$\leq(\mathbf{s}, \lambda, \mu) = F + \lambda^T \boldsymbol{g} + \mu^T \boldsymbol{h} \quad (5)$$

The approximation to the Hessian is obtained using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) scheme which maintains symmetry and positiveness.

Once direction $\boldsymbol{d}$ is found a line search is performed using a penalty function as the merit function in order to find the next iterand thus balancing the objective function while maintaining feasibility.

There are several implementations of such algorithm. Two fundamental differences exit among them: the procedure used to solve the quadratic subproblem and the merit function used on the line search. In this work the SQP version existing in the MATLAB optimization toolbox is used. Such version uses some form of Gill et al projection method to solve the quadratic subproblem while the exact penalty function ℓ, is used as merit function (Mathworks, 1995).

5. THE GENETIC ALGORITHM

GAs are search procedures based on mechanics of natural selection and natural genetics. They combine the artificial survival of the fittest with genetic operations to form a robust search mechanism. An initial population is randomly generated from the individuals bits in a fixed-length binary string *bs*. Successive generations are produced in which new solutions replaces some of the older ones. For a particular iteration *k* a GA maintains a population of potential solutions of individuals or chromosomes which contain all of the necessary information about the individuals they represent (the structural designs here).

A selection operator identifies the fittest individuals of the current population whereby pair of parents are chosen for the next generations. In the present context, the fitness function might be weight, strain energy or the fundamental frequency.

During reproduction 'crossover' and 'mutation' mechanisms are used to produce new population. The incidence of mutation and crossover is controlled by the user through prescript probabilities *pc* and *mp* respectively.

After each cycle of selection, crossover and possibly, mutation, the fitness of each family is again obtained by converting the binary strings to decimal digits (decoding) and evaluating the objective function. The whole process then continues into the next generation until a stopping criteria is met.

Figure 2 presents a simplified version of the algorithm. Details concern to the procedure implemented here can be found in (Goldberg, 1989).

5.1. Constraints Handling

In general structural designs involve several constraints related to stress, displacements, geometric dimensions and other variables. In GA these are conveniently handled by a penalty function. Using such function, constraint violations are penalized to avoid the future use of the set of parameters. The fitness function of the genetic algorithm is a combination of the objective function and penalty term. There is no unique way to define the penalty term. The fitness function considered here was proposed by Ghasemi and Hinton and al (Ghasemi and Hinton, 1996). We briefly describe their approach in the following.

Considering $\dot{g}_{i,j}$ a normalized constraint for a particular population *j* such is $\dot{g}_{i,j} = c_{i,j} - 1$ and $c_{i,j} = g_{i,j} / g_{i,all}$ in which $g_{i,all}$ is the allowable value of that constraint. The constraint $\dot{g}_{i,j}$ is satisfied if $\dot{g}_{i,j} \leq 0$.

If $\dot{g}_{i,j} \geq 0$ the objective function is penalized. In this case we define the following parameters $p_{v,i} = (c_{i,j})^k$ in which k is related to constraint violation (Ghasemi and Hinton, 1996), and $\hat{g}_{i,j} = p_c\, pv_i\, (g_{i,j})^2$ when p_c is the 'penalty coefficient' After all the constraints be penalized another parameters $\breve{g}_{i,j}$ is calculated such that

$$\breve{g}_{i,j} = \sum_{i=1}^{nc} \hat{g}_{i,j} \quad (6)$$

Finally the penalized objective function F^*_j is obtained such that $F^*_j = F_j(1 + \breve{g}_j)$ where F_j is the original objective function (without penalization).

In GA, F^*_j has to be transformed into fitness values in such that the best design has maximum fitness. For minimization problems, in order to guarantee such aim and also positive values for the fitness, the following expression is considered for fitness.

$$F'_j = (F^*_{max} + F^*_{min}) - F^*_j \quad (7)$$

In which F'_j is the fitness of design j and F^*_{max} and F^*_{min} are the maximum and the minimum values of F^*_j in the population of a particular generation j. It is recommended also to scale fitness function (Goldberg, 1989).

5.2. Convergence Criteria

Several termination criteria can be possibly used. Three convergence criteria for the present truss optimization are used:

1. The number of generations: A fixed number of generations can be provided by user. The GA algorithm will stop when the allocated number is reached.
2. Design changes: If the best design for the last 20 generations has not changed then iterations will stop.
3. Objective Function norm: It the variation of the objective function in a population is smaller than a very small given value *crate* thus if $\left|(\bar{F} - F_{j(best)}) / \bar{F}\right| .100 \leq crate$ were $\bar{F}$ is average value, the algorithm will stop.

GENETIC ALGORITHM PROCEDURE

1. Read data
2. Randomly generates initial population (generation 1)
3. Perform a generation loop
   for each generation do:
   3.1. A design loop
      for each design do:
      - Design variables decoding
      - Calculate original objective function
      - Compute fitness function
   3.2. Check the convergence : if found stop
   otherwise:
   3.3. Store best individual into next generation
   3.4. Proceeds genetic operations: solution, crossover and mutation to create population of new generation and go to step 3.1.

Figure 2 - Basic GA

6. EXAMPLES
6.1. Static Applications

The three and ten bar 2D trusses are considered in this section for optimization using both GA and SQP algorithms described in this article. The objective is to minimize the weigh of the structure by constraining the element stresses at each truss member.

a) Three bar truss

Problem definition: Figure 3 shows the tree bar truss to be optimized. Two cross sectional area design variables are considered: $X_1 = A^1 = A^2$ and $X_2 = A^3$. The material properties are Young's modulus $E = 2.07 \cdot 10^8$ and material density $\rho = 1$. The units are consistent.

Here the allowable tensile stress is 20, and the allowable compressive stress is 15. Table 1 presents the GA solution parameters adopted in this study. The bounds of the design variables are 0.1 and 1.0.

Table 1 – GA parameters for all problems presented here.

		GA parameters					
Condition	Truss	lc	ps	ng	pc	mp	crate
static	3 bar	6	10	100	10	1.0	1e-6
	10 bar	8	200	150	10	1.0	1e-3
free vibration	3 bar	10	300	100	10	1.0	1e-6
	9 bar	10	400	200	10	1.0	1e-6

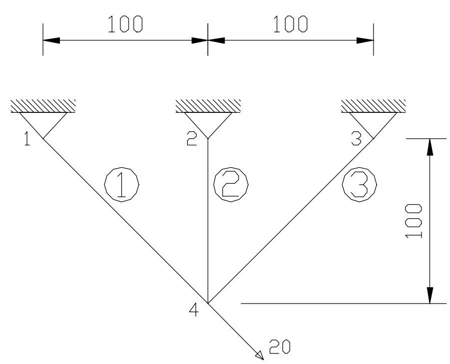

Figure 3 – Three bar truss

Discussions of results: Table 2 shows the optimal solutions for the present problem, obtained here and also the results reported in reference (Ghasemi and Hinton, 1996). As can be observed, all results are within an acceptable range when comparing with the exact solution. The optimum GA solution is achieved at the 2nd generation and remains unchanged afterwards.

Table 2 – Results of 3 bar truss

Design Variable	Present (SQP)	Present (GA)	Hinton (SQP)	Hinton (GA)	Exact
X_1	0.7887	0.7429	0.789	0.814	0.789
X_2	0.4082	0.4143	0.408	0.343	0.408
Optimum Weight	263.9247	251.5403	263.896	264.600	263.9

b) Ten bar truss

Problem definition: The ten bar truss problem shown in figure 4 is the second example used in this section. The design variables are the cross-sectional areas of the ten elements. The upper and lower bounds of the design variables are respectively 10 and 0.1. The material properties considered in this case are: $E = 10^7$, $\rho = 0.1$. For element 9 the allowable stress is $\sigma_{all} = \pm 75$ while the remaining elements the stress allowable is $\sigma_{all} = \pm 25$. Again all units are consistent. The GA solutions parameters used are given in table 1.

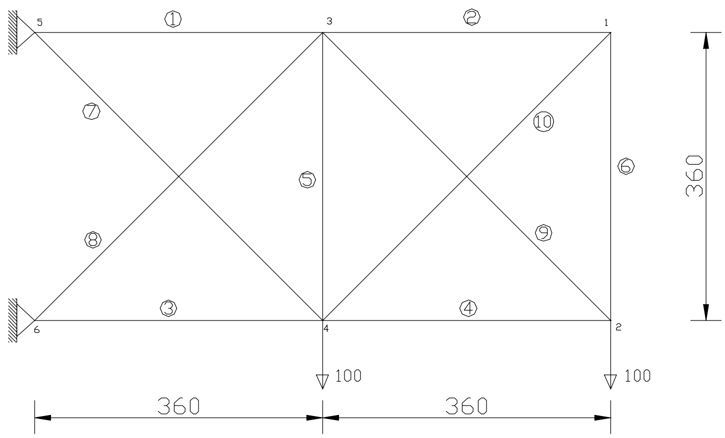

Figure 4 – Ten bar truss

Discussions of results: The optimization results obtained are listed in Table 3 together with the exact solution and those reported in reference (Ghasemi and Hinton, 1996). As can be observed the final objective value when using either GA or SQP are within acceptable range of the exact value. However, some of the optimum design variables differ from each other for the different methods investigated.

Table 3 - Results of 10 bar truss

Design Variable	Present (SQP)	Present (GA)	Hinton (SQP)	Hinton (GA)
X_1	7.9	7.518	7.90	7.518
X_2	0.1	0.458	0.10	0.458
X_3	8.1	8.430	8.10	8.430
X_4	3.9	3.544	3.90	3.544
X_5	0.1	0.100	0.10	0.100
X_6	0.1	0.460	0.10	0.460
X_7	5.7983	6.287	5.80	6.287
X_8	5.5154	4.992	5.51	4.992
X_9	3.677	3.350	3.68	3.350
X_{10}	0.1414	0.645	0.14	0.645
Optimum Weight	1497.6	1516.0	1497.0	1516.0

6.2. Free Vibration Applications

In this section we carry out optimization under free vibration conditions. A three bar and a nine bar benchmark examples are analyzed. In these structure the objective is to maximize the fundamental frequency whilst simultaneously keeping the structure weight constant.

a) Three bar truss

Problem definition: The three bar truss illustrated in Figure 5 is considered first. Two cross-sectional areas are taken as design variable: $X_1 = A^1 = A^2$ and $X_2 = A^3$ with a lower bound of 0.00005 and an upper bound of 0.001. The material properties are: the Young's modulus E = $2.0 \cdot 10^{11}$ and material density ρ = 7860. All units are consistent. The solution parameters used in the GA solution are shown in Table 1.

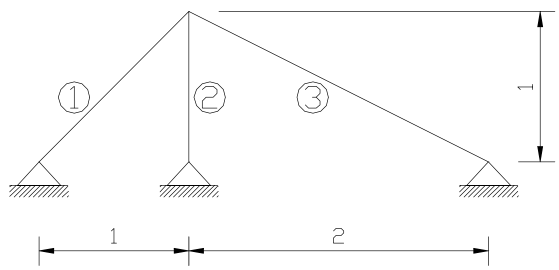

Figure 5 – 3 bar truss under free vibration conditions.

Discussion of results: Table 4 compares the optimal solutions for the present example obtained here and in reference (Alkhamis, 1996). The solution in terms of the optimum objective function matches perfectly while some difference is found in terms of the design variables value (GA solutions). GA solution comes with some constraint violation (0.07%)

Table 4 - Results of 3 bar truss under free vibration

Design Variable	Present (SQP)	Present (GA)	Alkhamis (SQP)	Alkhamis (GA)
X_1	$6.3984.10^{-4}$	$3.230.10^{-4}$	$6.398.10^{-4}$	$3.230.10^{-4}$
X_2	$1.3982.10^{-3}$	$6.824.10^{-3}$	$1.389.10^{-3}$	$6.824.10^{-3}$
Optimum Frequency	547.77 Hz	547.853 Hz	547.77 Hz	547.853 Hz

b) Nine bar truss

Problem definition: The second example studied in this section is the nine bar truss shown in Figure 6. The material properties used are the same of the previous example. The nine cross sectional area of the bars are taken as design variables. The lower and upper bound values are respectively 0.0005 and 0.5. Table 1 shows a list of parameters adopted in GA.

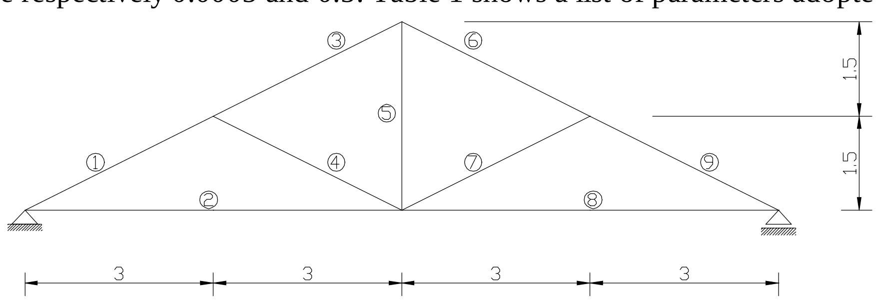

Figure 6 – 9 bar truss under free vibration conditions.

Discussion of results: The results for the different algorithms are provided in Table 5. Again good comparisons are obtained for the objective function, while the optimum design variables values obtained using GA presents some differences.

Table 5 - Results of 9 bar truss under free vibration

Design Variable	Present (SQP)	Present (GA)	Alkhamis (SQP)	Alkhamis (GA)
X_1	0.2721	0.317	0.272	0.317
X_2	0.3253	0.308	0.325	0.308
X_3	0.1429	0.181	0.143	0.181
X_4	0.0341	0.044	0.034	0.044
X_5	0.1008	0.079	0.101	0.079
X_6	0.1522	0.159	0.152	0.159
X_7	0.0327	0.026	0.031	0.026
X_8	0.2711	0.266	0.271	0.266
X_9	0.3063	0.260	0.306	0.260
Optimum Frequency	37.3782 Hz	37.148 Hz	37.38 Hz	37.148 Hz

7. CONCLUSIONS

In this work to carry out truss optimization two methods were considered here: the SQP and the GA. Both optimizers gave an excellent results and compared well. Although not explored in this paper, GA method allows the option of choosing a set of design variables from a certain specified catalogue.

Other main advantage using GA are:

- very simple calculations are involved; complex problems can be solved reasonably reliably; problems that have many local optima can be solved; and it is easy to interface the GA method to existing simulations and models.

One of the major disadvantages of using method is that the CPU time is high, however the use of parallel computations helps to circumvent this problem. This topic is current under our investigation.

8. REFERENCES

•Afonso, S.M.B. and Horowitz, B.,1998, " Utilização do MATLAB no desenvolvimento de ferramentas educacionais para o ensino da otimização estrutural", anais doV CEM – NEM,Fortaleza-CE, pp. 423-430.

•Alkhamis, M. T. A., 1996, "Structural Optimization for Static and Free Vibration Conditions Using Genetic and Gradient-based Algorithms", c/Ph/202/96, Swansea.

•Ghasemi, M.R. and Hinton, E., 1996, "Truss optimisation Using Genetic Algorithms - In Advances in Computational Structures tecnology", Civil-Comp Press Edinburgh, pp 59-75.

•Goldberg, D.E. 1989,"Genetic Algorithms in search, Optimization & Machine Learning", Addison-Wesley, N. York.

•Hajeta, P., 1990, "Genetic search–an approach to the non-convex optimization problem", AIAA, 28(7), pp 1205-1210.

•Han, S. P. 1976, – "Superlinear convergent variable metric algorithm for general nonlinear programming problems", Mathematical Programming, 11, pp- 263 – 282.

•Mathworks,1995, "Optimization Toolbox User's Guide", MATLAB Mathworks.