XLIII CILAMCE
21–25, November, 2022
FOZ DO IGUAÇU – PR – BRAZIL

Optimization of metallic truss using genetic algorithms via *CS-ASA*/MATLAB® softwares coupling

Laís B. Lecchi[1], Francisco A. Neves[1], Ricardo A. M. Silveira[1], Walnório G. Ferreira[2], José Eduardo S. Cursi[3]

[1]*Dept. of Civil Engineering, Federal University of Ouro Preto*
122 St. Diogo de Vasconcelos - Ouro Preto, 35400-000, Minas Gerais, Brazil
lais.lecchi@aluno.ufop.edu.br, fassis@ufop.edu.br, ricardo@ufop.edu.br
[2]*Dept. of Civil Engineering, Federal University of Espírito Santo*
514 Av. Fernando Ferrari - Vitória, 29075-910, Espírito Santo, Brazil
walnorio@gmail.com
[3]*LMN Lab, National Institute of Applied Sciences of Rouen*
685 Av. de l'Université - Saint-Étienne-du-Rouvray, 76800, Normandy, France
eduardo.souza@insa-rouen.fr

Abstract. Optimization consists of finding the best solution for a given objective, under given constraints. In Engineering, the application of optimization algorithms has greatly developed in the last decades, but it faces fundamental difficulties connected to the complexity of the mathematical models to be solved. In this last context, optimizing structures tries to achieve a reduction of the structural cost, under the restriction of not compromising efficiency and safety. This work aims to apply a heuristic optimization method — genetic algorithms (GA) — for the determination of optimal aluminium truss structures, considering design constraints associated with minimum areas and the maximum allowable stress. In this way, a computational routine is implemented in the MATLAB® program, using the GA with the advanced structural analysis program, based on the Finite Element Method, named *CS-ASA* (*Computational System for Advanced Structural Analysis*). MATLAB® manages all stages of the process, from the internal call of the *CS-ASA* to carry out the structural analysis, to the application of the optimization function, with the evaluation of the objective function and design constraints. Besides the analyses and comparison with literature, it is shown how numerical strategies to make the process less computationally expensive influence the total processing time.

Keywords: Structural optimization, Genetic algorithms, Metallic trusses, *CS-ASA*/MATLAB® coupling.

1 Introduction

Optimization techniques are powerful mathematical tools and potentially useful in many areas of knowledge. In structural design engineering, especially, given the demands for increasingly economic structures and, of course, with safety and durability properly assured, these techniques can be of great help to professionals in the area, in their search for the best design.

Dede et al. [1] emphasize that there is a growing interest in using optimization algorithms for the design of structures, especially in the last two decades. Modern methodologies, such as genetic algorithms, prove to be efficient in the search for the global optimum, since, by generating a population of individuals, they sweep the search space more broadly. It is used in several areas and problems, as in the work of Deligia et al. [2], who used GA's to optimize composite structures of concrete and steel in lattice format. The objective was to find the minimum weight, considering the beam geometry elements as design variables.

Lage [3] performed the weight optimization of steel trusses, by genetic algorithms, from the coupled operation of the programs ANSYS®-MATLAB ®, using first-order elastic analysis. The author emphasizes the computational cost involved, a factor that can limit the efficiency of the method.

In this sense, the present work aims to perform the optimization via genetic algorithms of the mass of a truss in aluminum material, through the coupled use of the programs *CS-ASA* (*Computational System for Advanced Structural Analysis*) and MATLAB®. It is shown how the strategies to reduce the analysis time, such as parallel computing and vectorization of functions, influence the process.

2 Structural optimization

2.1 Initial notions

In general, mathematical optimization comprises maximizing or minimizing one or more objective functions, within specific design conditions previously established [4]. In mathematical language, the optimization problem can be formulated as follows:

$$Find\ \ \mathbf{X} = \left\{ \begin{array}{cccc} x_1 & x_2 & ... & x_n \end{array} \right\},\ \ that\ \ minimizes\ \ f(\mathbf{X}),$$

subject to:

$$c_i(\mathbf{X}) \leq 0, \ \ i = 1, 2, ..., m, \tag{1}$$

$$d_j(\mathbf{X}) = 0, \ \ j = 1, 2, ..., p, \tag{2}$$

$$x_k^{low} \leq x_k \leq x_k^{up}, k = 1, 2, ..., n, \tag{3}$$

where:

- $\mathbf{X}$ is the *n*-dimensional vector containing the design variables to be optimized;
- $f(\mathbf{X})$ is the objective function of the problem, which in structural optimization, can represent the weight, volume or manufacturing cost, for example;
- $c_i(\mathbf{X})$ and $d_j(\mathbf{X})$ are inequality and equality constraints, respectively;
- x_k^{low} and x_k^{up} are the lower and upper bounds that design variables can assume;
- *i*, *j*, *k*, *m*, *n* and *p* are arbitrary values.

Regarding the design constraints of eqs. (1), (2) and (3), they can be divided into two distinct classes, namely: *i) behavior constraints* (or functionality), which is related to the performance and limit states of the structural system under study, represented in eqs. (1) and (2); *ii) lateral constraint* (or geometric), in which feasible physical limits are considered, such as availability, manufacture, transport, etc., as shown in eq. (3) [5].

2.2 The Genetic Algorithms

Genetic algorithms are part of a set of so-called modern optimization methodologies [5], originally proposed by Holland [6]. They are based on principles of nature, such as genetics and natural selection in the reproduction of species. As they are stochastic and gradient-free methods, they have good applicability in problems like multi-objective optimization; problems with continuous and discrete variables together; when the functions are discontinuous, or non-differentiable, as well as for non-convex design spaces. The basic terminology relevant to genetic algorithms is presented:

- **Objective function:** is the function to be optimized;
- **Penalty function:** mathematical expression applied to the fitness value of an individual, calculated based on the violation of the problem's constraints;
- **Fitness function:** mathematical expression given by the sum of the objective and penalty functions, which works as an indicator of the quality (fitness) of an individual to be the best solution to the problem. The fitness function can assume the same value as the objective function;
- **Individual:** is a point (vector), containing the value of each one of the variables, to which the fitness function is to be applied, getting a *score* as a result. It can also be called *chromosome* and, its entries, *genes*. Below is a representation of this structure with the vector $\mathbf{X}$:

$$\mathbf{X} = \left[\begin{array}{ccccc} x_1 & x_2 & x_3 & ... & x_n \end{array} \right];$$

- **Population:** is the matrix of individuals. The user must specify a value *p*, for the population size. Therefore, the population matrix will have dimension $p \times n$, where *n* is the number of variables in the problem;
- **Generation:** each generation represents an iteration, in which a new population matrix will be created, by applying the genetic operators, known as: **selection**, **elitism**, **crossover** and **mutation**;
- **Diversity:** is measured by the distance between individuals in a population. It is a property of great importance in the performance of GA's, since a greater diversity of the population means a greater scan of the design space;
- **Parents and Children:** The GA's, through the selection process, use the individuals with the best fitness value of the current generation, called parents, to create those of the next iteration (children).

The flowchart in Fig. 1 outlines the running of genetic algorithms.

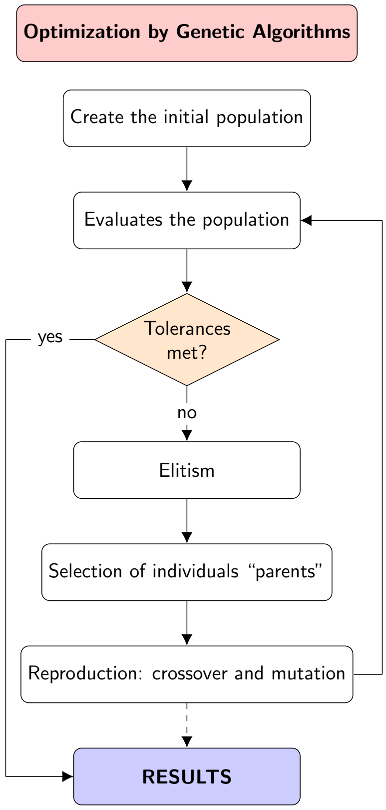

Figure 1. Genetic algorithms flowchart.

2.3 The *CS-ASA* and MATLAB® softwares coupling

MATLAB® manages all the steps of the optimization process, which are: opening *CS-ASA*, through the command *"system('CS-ASA.exe');"*; the call of the chosen optimization algorithm, with its proper operating settings; reading the output file (*"FileOut.s"*) with the results of the structural analysis from *CS-ASA*, to evaluate the behavior constraints; and writing the new input file to *CS-ASA* (*neutral1.d*), with the structural model containing the new values of the optimized design variables. The steps described can be checked in Table 1.

Table 1. Command sequence for structural optimization.

Command sequence for structural optimization via Matlab®
(genetic algorithms)

```
Define n; (number of executions)

Start of time counting;

For i from 1 to n, do:

    system('CS-ASA.exe'); (calls CS-ASA)

    opts = optimoptions(@ga, ...); (set genetic algorithms options)

    A = [...]; b = [...]; Aeq = [...]; beq = [...]; lb = [...]; ub = [...];
    intcon = [...];  (set behavior and lateral constraints)

    [x,fval] = ga(@Fobj,nvars,A,b,Aeq,beq,lb,ub,@nonlcon,intcon, opts);
     (calls GA, performs the optimization and returns the variables
     X optimized and also the value of the objective function fval)
    (obs: the @ is used to call functions in files external to the
     main file (function handle))

    WriteFile1(X); (calls function that will rewrite the file for CS-ASA,
    with the new X design variables)

End-For

Stop of time counting;
```

3 Numerical example

3.1 General information

The numerical example of the 10 bars truss in Fig. 2 was implemented on a notebook with a *Intel(R) Core(TM) i7-7500U CPU 2.70* GHz processor and 16 GB of RAM. The optimization is carried out in such a way to find the minimum mass of the structure, having the areas of the bars as design variables. This structure has been studied by several researchers over the last few years, such as Olsen and Vanderplaats [7], Haftka and Gürdal [8], Lombardi [9], Toğan and Daloğlu [10], Kaveh [11], Noii et al. [12], Lage [3], Singh and Kapania [13], constituting a *benchmark*, for validation of optimization algorithms.

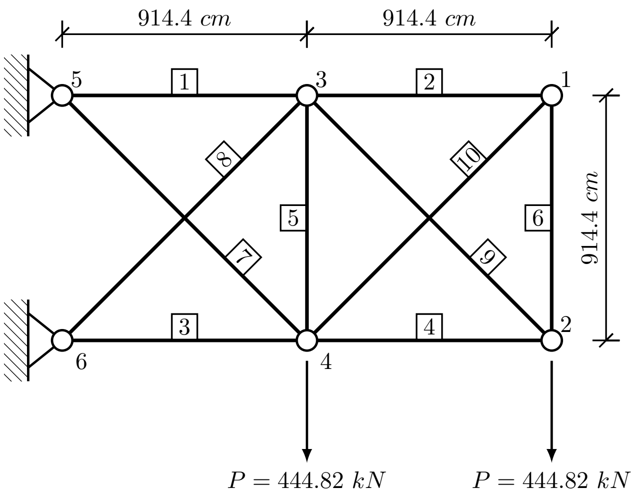

Figure 2. 10-bars aluminium truss.

- Material: Aluminium;
- Young's Modulus: $E = 68947.573\ MPa$;
- Minimum area of bars: $A_{min} = 0.645\ cm^2$;
- Specific mass: $\rho_{alu} = 2767.99\ kg/m^3$;
- Maximum allowable stress in the bars: $\sigma_{max} = \pm 172.369\ MPa$;
- First Order Elastic Analysis.

3.2 Optimization algorithm setting

Genetic algorithms were used to optimize the structure, contained in the *Global Optimization Toolbox*, of MATLAB®, with the following specific settings:

- Population size for each generation ('PopulationSize'): 200 individuals (default);
- Creation function ('CreationFcn'): 'gacreationuniform' (default);
- *Crossover* function ('CrossoverFcn'): 'crossoverscattered' (default);
- Mutation function ('MutationFcn'): 'mutationgaussian' (default);
- Elite individuals ('EliteCount'): 5% of population size;
- Maximum number of generations ('MaxGenerations'): 50;
- Algorithm for handling nonlinear constraints ('NonlinearConstraintAlgorithm'): 'auglag';
- Tolerance for objective function ('FunctionTolerance'): 10^{-6};
- Tolerance for constraints ('ConstraintTolerance'): 10^{-3};
- Use of vectorized functions ('UseVectorized'): 'true';

3.3 Design variables

The variables are continuous, with the side bounds starting from the minimum required of $0.645\ cm^2$ to the value of $58.064\ cm^2$. It stipulated the maximum value in order to simplify the optimization problem. Thus, the problem is delimited between the following lateral constraints:

$$0.645\ cm^2 \leq x_1, x_2, x_3, ..., x_{10} \leq 58.064\ cm^2.$$

3.4 Design constraints

Besides the lateral constraints of the previous item, there are also constraints of maximum tensile and compressive stress in the bars:

$$T(\mathbf{X}) = \frac{|\sigma_i|}{|\sigma_{max}|} - 1 \leq 0,\ \ with\ i = 1...10, \tag{4}$$

where σ_i is the stress in bar i, given by the normal stress N_i divided by the area of bar x_i:

$$\sigma_i = \frac{N_i}{x_i}. \tag{5}$$

3.5 Objective function

The calculation of the objective function $M(\mathbf{X})$, which represents the minimum mass of the structure, is given by eq. (6):

$$M(\mathbf{X}) = \sum_{i=1}^{n=10} \rho_{alu} l_i x_i, \tag{6}$$

where l_i is the length of the bar i.

3.6 Results

100 executions were performed, with a total analysis time of 7.5 minutes. The initial value of the variables of $32.258\ cm^2$ was adopted. 50 generations were stipulated for each objective function evaluation, but the genetic

algorithm converged with less than 10 generations, every time.

Table 2 shows how the application of parallel computing or vectorization of functions, resources available in the program MATLAB®, can rationalize the analysis time, representing a large gain the use of vectorization of functions.

Table 2. Comparison of the time of an optimization run according to each strategy applied.

Strategy	**Time of analysis**
None	$\cong 5\ min$
Parallel computing	$\cong 2\ min$
Vectorization	$\cong 5\ sec$

Table 3 shows the best, worst, mean and standard deviation for the 100 executions:

Table 3. Results.

Execution	Mass (kg)	x_1 (cm^2)	x_2 (cm^2)	x_3 (cm^2)	x_4 (cm^2)	x_5 (cm^2)	x_6 (cm^2)	x_7 (cm^2)	x_8 (cm^2)	x_9 (cm^2)	x_{10} (cm^2)
Best	722.074	51.032	0.645	52.193	25.225	0.645	0.645	37.354	35.612	35.677	0.838
Worst	769.701	50,516	10.387	52.903	15.612	9.225	10.387	38.322	34.838	21.870	14.709
Mean	723.979	50,967	1.032	52.322	24.967	0.774	1.032	37.483	35.548	35.354	1.225
Standard deviation	5.711	0.903	1.419	0.903	1.483	0.903	1.419	1.225	1.225	2.129	2.064

In Table 4, the best result found in this work is compared to that of Haftka and Gürdal [8], who used two methodologies to solve the proposed problem: the FSD (*Fully Stressed Design*) and another based on the Optimality Criteria, by Berke and Khot [14].

Table 4. Comparison with the literature.

Bar	**Haftka and Gürdal (1991)**		**Present work**	
	Area $(\boldsymbol{cm^2})$	**Stress** $(\boldsymbol{MPa})$	**Area** $(\boldsymbol{cm^2})$	**Stress** $(\boldsymbol{MPa})$
1	51.225	172.369	51.032	172.369
2	0.645	107.420	0.645	124.106
3	51.999	−172.369	52.193	−172.369
4	25.419	−172.369	25.225	−172.369
5	0.645	−0.069	0.645	−4.136
6	0.645	106.179	0.645	123.623
7	37.032	172.369	37.354	172.369
8	35.935	−172.369	35.612	−172.369
9	35.935	172.369	35.677	172.369
10	0.645	−151.822	0.838	−170.852
Mass (kg)	**722.663**		**722.074**	
$\Delta(\%) - f_{obj}$	**0.08%**			

4 Conclusions

As it can be seen, the genetic algorithms proved to be efficient in the search for the minimum mass of the 10-bar truss in Fig. 2, and its results are practically equivalent to those found in the literature. Also noteworthy is the considerable improvement in the execution time of the analysis, from the use of the vectorization strategy, available in the program MATLAB®.

Furthermore, given the stochastic nature of the method addressed, the number of runs performed directly influences the statistical results, such as the mean and standard deviation shown in Table 3. Hence the importance of introducing strategies that reduce the computational cost of the process. Finally, it is noteworthy that the definition of lateral constraints (minimum and maximum) also helps in the proper functioning and achievement of results.

Acknowledgements. The authors thank CAPES, CNPq, Fapemig and UFOP institutions, for the finantial support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] T. Dede, M. Kripka, V. Togan, V. Yepes, and R. V. Rao. Usage of optimization techniques in civil engineering during the last two decades. *Current Trends in Civil & Structural Engineering*, vol. 1, n. 1, pp. 1–17, 2019.

[2] M. Deligia, E. Congiu, G. Marano, B. Briseghella, and L. Fenu. Structural optimization of composite steel trussed-concrete beams. *Procedia Structural Integrity*, vol. 33, pp. 613–622, 2021.

[3] C. M. Lage. *Otimização de Projeto de Estruturas Treliçadas Formadas por Perfis Tubulares Soldados, Usando Algoritmos Genéticos e Método dos Elementos Finitos*. PhD thesis, Programa de Pós-Graduação em Engenharia Civil, Ouro Preto - MG, 2019.

[4] A. Messac. *Optimization in practice with MATLAB®: for engineering students and professionals*. Cambridge University Press, New York, 2015.

[5] S. S. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, Florida, 2019.

[6] J. Holland. Adaptation in natural and artificial systems. *Ann Arbor: University of Michigan Press*, 1975.

[7] G. R. Olsen and G. N. Vanderplaats. Method for nonlinear optimization with discrete design variables. *AIAA journal*, vol. 27, n. 11, pp. 1584–1589, 1989.

[8] R. T. Haftka and Z. Gürdal. *Elements of Structural Optimization*, volume 11. Springer Science & Business Media, Dordrecht, 1991.

[9] M. Lombardi. Optimization of uncertain structures using non-probabilistic models. *Computers & structures*, vol. 67, n. 1-3, pp. 99–103, 1998.

[10] V. Toğan and A. T. Daloğlu. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers & Structures*, vol. 86, n. 11-12, pp. 1204–1218, 2008.

[11] A. Kaveh. *Advances in metaheuristic algorithms for optimal design of structures*. Springer, Cham, 2014.

[12] N. Noii, I. Aghayan, I. Hajirasouliha, and M. M. Kunt. A new hybrid method for size and topology optimization of truss structures using modified alga and qpga. *Journal of Civil Engineering and Management*, vol. 23, n. 2, pp. 252–262, 2017.

[13] K. Singh and R. K. Kapania. Alga: Active learning-based genetic algorithm for accelerating structural optimization. *AIAA Journal*, vol. 59, n. 1, pp. 330–344, 2021.

[14] L. Berke and N. Khot. Use of optimality criteria for large scaled systems. *AGARD. Lecture Series No. 170 on Structural Optimization, AGARD-LA-70*, 1974.