

PAPER • OPEN ACCESS

Weight optimization of plane truss using genetic algorithm

To cite this article: D Neeraja *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 032015

View the [article online](#) for updates and enhancements.

Related content

- [Comparison of the Reliability Methods for Steel Trusses Subjected to Fire](#)
Katarzyna Kubicka, Urszula Rado and Waldemar Szaniec
- [Brief research on arch hinge of the steel truss arch bridge by contact problem under local stress](#)
Wei He, Yinqiao Zhu and Renlong Chen
- [Optimization of Structural Design for Sustainable Construction of Transmission Tower Based on Topographical Algorithm](#)
Zakaria Che Muda, Kamal Nasharuddin Mustapha, Rohayu Che Omar et al.

Weight optimization of plane truss using genetic algorithm

Neeraja D, Thejesh Kamireddy, Potnuru Santosh Kumar, Vijay Simha Reddy
School of Civil and Chemical Engineering, VIT University, Vellore, Tamilnadu-
632014, India.

Email: neeraja.d@vit.ac.in

Abstract. Optimization of structure on basis of weight has many practical benefits in every engineering field. The efficiency is proportionally related to its weight and hence weight optimization gains prime importance. Considering the field of civil engineering, weight optimized structural elements are economical and easier to transport to the site. In this study, genetic optimization algorithm for weight optimization of steel truss considering its shape, size and topology aspects has been developed in MATLAB. Material strength and Buckling stability have been adopted from IS 800-2007 code of construction steel. The constraints considered in the present study are fabrication, basic nodes, displacements, and compatibility. Genetic programming is a natural selection search technique intended to combine good solutions to a problem from many generations to improve the results. All solutions are generated randomly and represented individually by a binary string with similarities of natural chromosomes, and hence it is termed as genetic programming. The outcome of the study is a MATLAB program, which can optimise a steel truss and display the optimised topology along with element shapes, deflections, and stress results.

1. Introduction

Indubitably the material cost plays a vital role in design and construction of a structure; the reduction in weight of the structure is by minimizing the weight or volume of the structure's arrangement. The minimization of weight of the material used can be used as tool for optimum and economical design. The method of reducing the weight to attain an optimum design has a set of variables under specific limitation. For a proper optimization method for structural design one should have a good idea on the characteristics of the problem. For a structural design optimization, the solution should be near the global optimal solution and the design variables should be distinct and be chosen from the pre-established set of elements used in structural design problems. An optimal structure is a structure that can be optimal in different features. The different features also called objectives, and for a case, it may be weight, cost or stiffness of a structure. Optimisation will be done with some limitations. Firstly, the design limitations can be limited to geometrical extension or structural components. Secondly, their structural behaviour like displacements, stresses, forces and dynamic response be sorted. Finally, there is an behavioural constraint valid for all structures, and it is kinematical stability or mechanisms. The structure satisfying all the restrictions are called feasible solutions to the optimization problem.

2. Genetic Algorithm (GA)

Genetic Algorithms has three operators namely selection, crossover, and mutation. In each iteration or generation, these operators are used on a population of all possible solutions, in order to develop their fitness function. Every solution is described by a string, and these strings are very much of the original



chromosomes, hence it is named genetic algorithms. Initially, the population is randomly generated, and it continues until a terminating criterion is attained, e.g. the exceeding of a given limit of generations.

2.1 System of algorithm

Among the easily accessible toolboxes for MATLAB, GPLAB stands out as one of the most adaptable and robust. But, the implementation of the algorithms in this paper does the integrated Global Optimization toolbox and customizes the functions where required. A general principle system of the algorithm practiced is shown in Figure. 1. The algorithm starts by randomly creating a population of candidate solutions within the design space. From this point the population is repeated in a loop (each loop giving birth to a new generation of individuals), developing towards what is feasibly an optimal solution. Each individual in the population has a fitness value defined by the how well confirmed and economic its correspondent solution (truss) is. The idea of the fitness value is produced by calculating the total weight of the suggested solution and by determining with FEA(Finite Element Analysis) how well it fits the maximum stress and maximum permissible displacement. The individuals in each new generation are obtained from the old generation by using the GA operators of selection, crossover and mutation.

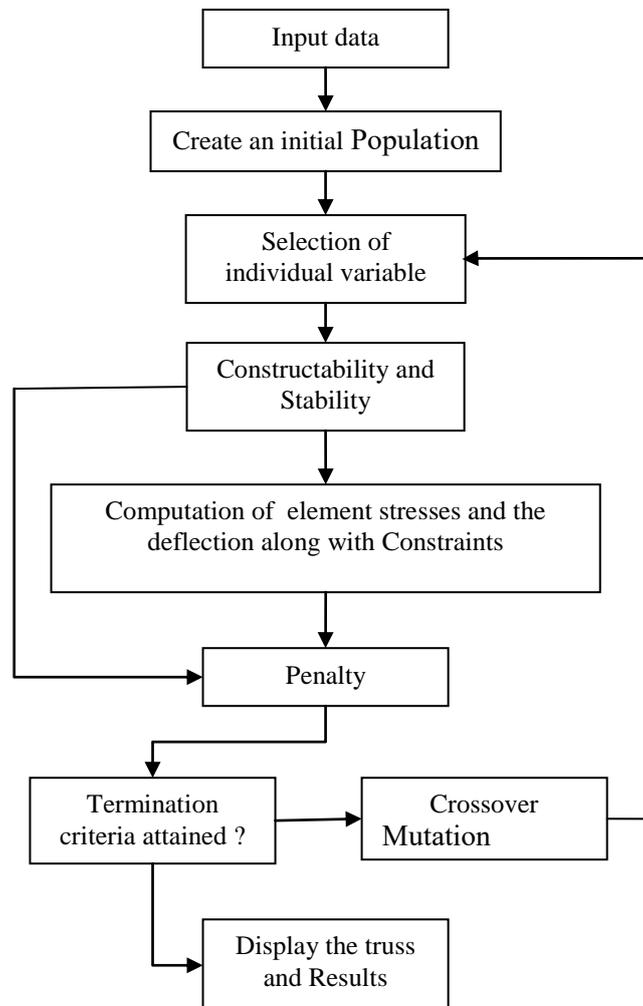


Figure 1. Working methodology of Genetic algorithm

2.2. Representation

Just like a chromosome, the string has various sections, or genes, that match to different characteristics of the solution. In biological terms, the total data stored in a string of the solution is named a genotype of an individual or the population. The outer appearance of the population is called a phenotype, and between the two, there exists a transformation named a genotype-phenotype mapping. In the conventional GA, there exists a fixed length binary string. The number of genes is dependent on the number of variables that need description. For example, the following string consists of five genes, for five problem variables (Figure.2).

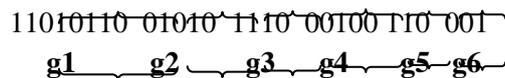


Figure 2: String of five genes

The bits required for a particular gene is calculated as Eq. 1

$$l_i = \log_2 \left(\frac{x_i^{\min} - x_i^{\max}}{\varepsilon_i} \right) \quad \text{--- Eq. 1}$$

x_i^{\min} is lower bound of variable i , x_i^{\max} is upper bound of variable i , and ε_i is desired precision in variable i . The higher number of possible combinations indicates that more data has to be cached. The string is split in to genes which can be read and interpreted individually.

2.3. Fitness evaluation

The purpose of the optimization is to reduce the total weight of the structure while holding it below the maximum permissible displacement and stress. The stresses and displacements evaluation of candidate solutions is created utilizing the finite element method for the case of plane truss structures, with code implemented by the authors. The common equations of the problem are defined in any FEM book, the evidence used in this study being [11]. For dealing with nonlinear limitations, MATLAB's GA toolbox offers the enlarged lagrangian barrier algorithm [12]. Still, early tests of this method have been disappointing so we chose a different path in the form of a penalty function applied to the nonconforming individuals. Thus, the primary fitness of an individual is its weight, but this value is then penalized with a certain amount, depending on how large the constraint violation is. The penalty function we designed for this study is presented in Eq. 2.

$$PF(\sigma) = OF \cdot \left[IP + P2 \cdot \left(\frac{\sigma - \sigma_a}{\sigma_a} \right)^2 \right] \quad \text{--- Eq. 2}$$

Where PF – penalized fitness, OF – original fitness, IP – initial penalty at the point of stress limit, $P2$ –penalty scale factor at double the stress limit, σ - Actual maximum stress in the structure, σ_a - maximum allowable stress for the chosen material. The above function is used for the "stress constraint". For the "displacement constraint" a similar function was used and the two penalties add up in the case of both stress and displacement constraint violation

2.4. Proposed Algorithm

The optimization algorithm in the present study is a bit-string encoded genetic algorithm, which is feasible to optimize planar steel trusses by its weight in accordance to IS 800. The shape, size, and topology characteristics are handled simultaneously. The elements are chosen from the table in appendix B, and the nodal positions are with precision of one-tenth of a meter. The dead weight and dimensional stability are taken in to consideration.

The flow chart of the algorithm is shown in Figure 1.

2.5 Constraints

The five constraints in the suggested Genetic algorithm are as follows:

2.5.1 Fabrication. The available profiles are taken from IS 4923(1997): Hollow steel sections for structural use Specification. The list of dimensions available is given in the input of the programme. While creating a truss in the truss, it will pick elements from the table, satisfying the constraints automatically. The list of profiles is detailed and ranges from 25x25 to 150x150 mm. this helps the algorithm to give precise solution.

2.5.2 Basic node. Firstly, we define the coordinates of the nodes necessary i.e. node at support or under a load. The truss thus formed should have all primary nodes, and thus satisfied this constraint automatically.

2.5.3 Stability. The truss should be kinematically stable, to verify the stability of truss, the determinant of its stiffness matrix is calculated. If its zero, the structure is considered not stable, but all other values of determinants tells us its stable.

2.5.4 Nodal displacements. The limit of nodal displacements is related to span to width ratio of the structure. The maximum allowed deflection is restricted to $(L_x / 250)$ in y-direction and $(L_y / 250)$ in x-direction as per code which are used as normal limits in this case.

2.5.5 Constructability. Along with the fabrication and deflection constraints, additional constructability constraints are to be given defined. This is considered by not providing two or more elements to have their nodes in common and they cannot exist in same place. The elements are also not allowed to start and end up in same node. Breaking of the above constraints would lead to a penalty.

2.6 Elemental stresses

2.6.1 Tension

The basic criterion for tensile stresses in a steel bar at each of its cross section should satisfy $T < T_d$ [design strength due to yielding of gross section] (IS code SP 38-clause 6.2).

2.6.2 Compression

For compressive stresses, the effects of buckling have to be considered, mainly dependent on geometrical properties of the element. Buckling resistance of bar: the design compressive strength taken from (IS code SP 38-clause 6.2clause 7.1.1). All the values have been taken as per Indian standard codes.

3 Results and Discussion

In this study, benchmark problem was considered, which is a ten bar truss. The positioning of nodes and simultaneous optimization of shape, size and topology is considered which was not done in the previous research.

The truss shown in Figure 3 has two vertical supports, distance between them is 9.144 m and two loads are at a distance of 9.144 m and 18.288 m from left lower support, the load is 445.375 kN.

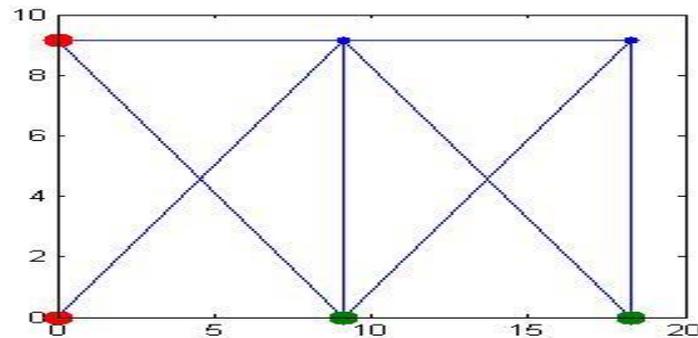


Figure 3: The Ten-bar truss

For comparison, some of the results obtained by various papers using the above parameters were given. In the above research papers, all the constraints were not considered at a time. In this research, an attempt was made to apply all the constraints size, shape and topography simultaneously. Further in the study, steel is used as material for the truss to optimize for shape, size and topology according to Indian standard codal provisions. The material properties of our steel were $E = 2 \times 10^5 \text{ N/mm}^2$ and density is 7.85 kg/m^3 , Quality of steel section used is $f_y 310$. 2323.32 kg as in Size and shape optimization by using genetic algorithm by Deb and Gulati (2000). 2421.79 kg as in Size and topology optimization by a genetic algorithm by Hajela and Lee (1995). 2259.95 kg as in Size optimization by a particle swarm optimizer by Li, Huang and Liu (2006). 2310.90 kg as in Size optimization by a hybrid search method by Kripakaran, Gupta and Baugh Jr. (2007). 2232.80 kg as in Size and shape optimization by a genetic algorithm by Galante (1996).

The Truss was optimized for two runs. In the first run population size was given to be 600 and maximum generations is kept as 500 and the output given is a truss of 2669 kg with one free element and six nodes as the result. Using the value obtained from first run, second run was given again with population size 900 and maximum generations as 250. Also, the number of free element is kept one from the first run and the results shown in command window are tabulated in Table 1. An output plot resembling the optimized truss is shown in Figure. 4.

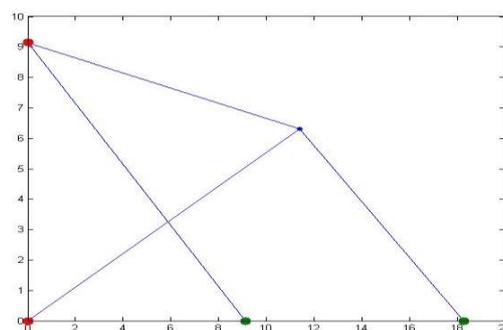


Figure: 4 Optimized topology of truss

Here it can be observed that elements were reduced because of the proposed topology optimization. Total Weight of the truss is 1918.25 kg where the original weight of the truss is 2669 kg. The results obtained were shown in Table 1. Table 1 shows the numerical results of the bench mark problem.

Table 1. Numerical results of the bench mark problem

Bar	Dimension	Area (cm ²)	Weight per meter (kN)	Start coordinate(X,Y)(m)	End coordinate(X,Y)(m)	Length (m)	Weight (kg)	Element Stresses(N/mm ²)
1	TUB1131136	24.8	18.228	0,9.144	9.144,0	9.144	226.7712	278.213
2	TUB1501506	33.6	24.696	0,0	9.144,0	9.144	305.76	181.421
3	TUB1131136	33.6	24.696	0,0	11.4,6.3	9.302	312.5472	-231.454
4	TUB1251256	27.6	20.286	9.144,0	18.288,0	11.749	324.2724	-325.79
5	TUB1321326	28.4	20.874	11.4,6.3	18.288,0	12.932	367.2688	-259.553
6	TUB1501505	29.3	21.5355	11.4,6.3	0,9.144	13.025	381.6325	191.129
						Total Weight=	1918.2521	

4 Conclusions

From the present study, the following conclusions were drawn:

1. The weight optimization of the truss was done basis on size, shape, and topology. The proposed algorithm gives near optimal solutions to all weight optimisation of most of planar trusses, but would take time for complex geometry.
2. Optimization on the truss on size, shape, topology parameters gives an extreme number of possibilities of population for simple structures also. To ensure earlier convergence, we consider a large population and therefore, the calculations are time consuming on a regular PC.
3. The effective ways to decrease the complexity of the problem is to start with simple problems with fewer nodes and thus the numbers of possibilities come down and computer effort needed would be less.
4. Optimization should be done initially with fewer nodes and elements as possible which would give an idea on how many nodes and elements to begin with, to include fewer possibilities.
5. It is appropriate to go for two runs on optimization. First one provides a global minimum and gives an idea on the number of nodes and elements to start with. The second run helps to finds the optimal solution by reducing the complexity.

References

- [1] Indian Standard "IS 4923 (1997): Hollow steel sections for structural use Specification [MTD 19: Steel Tubes, Pipes and Fittings]", *Bureau of Indian Standards*.

- [2] Indian Standard “IS 806 (1968): Code of Practice for Use of Steel Tubes In General Building Construction [CED 7: Structural Engineering and structural sections]”, *Bureau of Indian Standards*.
- [3] Indian Standard “SP 38 (1987): Handbook of Typified Designs for Structures with Steel Roof Trusses (With and Without Cranes) (Based on IS Codes) [CED 12: Functional Requirements in Buildings]”, *Bureau of Indian Standards*.
- [4] Deb, Kalyanmoy. Gulati, Surendra. (2000). *Design of Truss-Structures for Minimum Weight using Genetic Algorithms. Finite Elements in Analysis and Design*. Volume **37**, issue 5. Pages 447-465.
- [5] Galante, Miguel. (1996). Genetic Algorithms As An Approach To Optimize Real-World Trusses. *International Journal for Numerical Methods in Engineering*. Volume **39**, issue 3. Pages 361-382.
- [6] Hajela, P. Lee, E. (1995). Genetic Algorithms In Truss Topological Optimization. *International Journal of Solids and Structures*. Volume **32**, issue 22. Pages 3341-3357.
- [7] Kripakaran, Prakash. Gupta, Abhinav. Baugh Jr. John W. (2007). A novel optimization approach for minimum cost design of trusses. *Computers & Structures*. Volume **85**, issue 2324. Pages 1782-1794.
- [8] Li, Lijuan. Huang, Zhibin. Liu, Feng. (2006). An Improved Particle Swarm Optimizer for Truss Structure Optimization. 2006 *International Conference on Computational Intelligence and Security*. Volume 1. Pages 924-928.
- [9] Chipperfield, A. J., & Fleming, P. J. (1995). *The MATLAB genetic algorithm toolbox*.