JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
31st December 2017, Volume 2, Issue 3, JSEIS, CAOMEI Copyright © 2016-2017
www.jseis.org

Size optimization of steel trusses using a genetic algorithm in MATLAB

Ersilio Tushaj
Polytechnic University of Tirana, Albania
Email: ersilio.tushaj@gmail.com

ABSTRACT

An important aspect of any engineering design problem is to achieve efficiency and efficacy. This can be in terms of energy consumption, performance, time, total weight and costs. In many cases, there are multiple solutions to a problem and you should select the one which satisfies better the criteria. This engineering design process is known as optimization. Optimization plays an important role in various engineering applications. Engineers are in continuity, challenged to design structures that use the least amount of resources and satisfy the structural requirements. The optimal design of structures can be decomposed into three major categories: topology, shape and size optimization. These methods have evolved with time and they may be divided in two maxi-groups: deterministic and non-deterministic algorithms. Size optimization of non-deterministic methods with genetic algorithms (GA) are investigated in this article and applied to some steel trusses in MATLAB soft R2017a. This is done by building an algorithm consisting in scripts and sub-functions, which are applied to the trusses for different constraints on stresses, displacements and buckling, depending on the case analyzed. Different values for the GA parameters are analyzed in such way to achieve the best design. The results are put in comparison with previous studies.

Keywords: genetic algorithm; steel trusses; structural optimization; engineering; optimization; performance;

1. INTRODUCTION

Reducing costs while meeting performance standards is a common challenge in structural design. Engineers typically rely on experience and standardized design procedures to make their structures more efficient [1]. A lot of systematic methods based on mathematical algorithms and grouped under the generic name of *Structural Optimization* are available to help designing efficient structures. Optimization is a vast field of mathematics whose theory is still actively being developed. But when applied to structural engineering, it is essentially regarded as a helpful to the engineer willing to design more efficient structures.

Optimization of steel trusses has been largely investigated by authors from the beginning of structural optimization in civil engineering. The first who gave a mathematical formulation of nonlinear optimization of steel trusses was Schmit in 1960 [2]. Others will follow introducing better performant algorithms which can offer more reliable solutions at a minor time [3]. Optimization of steel trusses, with the developments of programming and computers, can be considered as an integration of knowledges in structural matrix analysis, optimization algorithms, and computer programming. Kirsch [4] in his book Structural optimization: Fundamentals and applications, reported the necessary step to follow a total layout optimization using matrix analysis of monodimensional structures and deterministic optimization techniques.

2. PREVIOUS STUDIES AND GENETIC ALGORITHMS IN STRUCTURAL OPTIMIZATION

Different algorithms have been applied successfully in the steel structural design. A survey was prepared by the same author of this paper in [5]. Previous state of the art and reviews in structural size optimization have been prepared by different authors [6-9].

Optimization of steel structures have been largely studied in the international literature. Stasa [10] is an albanian case, from the Polytechnic University of Tirana. In her PhD Dissertation in 1994, she analyzed the optimal design of steel trusses using two deterministic methods: the Fully Stressed Design (FSD) and the Sequential Linear Programming with move limits (SLP). For each algorithm applied and constraints imposed, were given the results of the optimal weight and the final design of the steel elements. The Objective Function imposed was the minimal weight of the trusses. Stresses, displacement and slenderness criteria were applied to the problem. Comparisons were reported between the two methods.

Hasancebi, 2009 [11], has studied the performance of some non-deterministic algorithms applied to the optimum design of steel trusses. Chain, 2015 [12] has done a survey on deterministic approaches applied to steel structures design. A state of the art in the use of genetic algorithms in structural optimization was prepared by Pezeshk as a chapter in the Report in Recent Advances in Optimal Structural Design, in 2002 [13].

Genetic algorithms are part of the evolutionary optimization techniques. The first to introduce these algorithms was Holland [14] in 1975. The (GA)s uses concepts from evolutionary biology. (GA)s are efficient and applicable in search procedures based on a stochastic approach which relies on the "survival of the fittest". (GA)s can be a powerful design tool in optimization. They don't require gradient information and can handle random data. (GA)s are search algorithms and are based on concepts of natural selection and natural genetics. (GA)s have a different approach from traditional optimization methods as they work with a coding of variables and not with the variables themselves; can operate on a population of potential solutions; search the optimum from an objective function, called fitness function, without having information on gradient; and they use a probabilistic search scheme.

GAs have been largely used in optimization by many researchers [15-22]. Dhingra and Lee (1994) [23], applied a GA in obtaining a single - and multiple-objective design problems and presented several cases dealing with optimum design of truss structures with discrete variables. Adeli and Cheng (1993) [24] in their study used a GA in the design of space truss structures. They presented a (GA)s procedure for the optimization of three-dimensional truss structures. Others have continuously developed and improved the performance of genetic algorithms. Combination of (GA)s with other algorithms have been studied too.

3. GENERAL FORMULATION OF THE PROBLEM

The optimization problem finds a minimal optimal solution for an objective function, which can be total weight or cost of the structure. In this paper the study is done on the optimization of the total weight. The analysis is for steel trusses with constraints on stresses, displacements and buckling. The objective function for total weight is given in equation 1.

$$minimal\ total\ weight\ Z = f(x) = f(x_1, x_2, \dots, x_n) = \sum_{i-1}^{n} l_i x_i \tag{1}$$

$l_i - length\ of\ every\ element\ of\ the\ truss$
$x_i - section\ of\ the\ element$
$n - number\ of\ the\ elements$

Stress constraints consider that member forces should satisfy some maximum values as shown in equation 2.

$$\sigma_{min,i}\, x_i \leq F_i \leq \sigma_{max,i}\, x_i \tag{2}$$

Sometimes in the comparison between different algorithms, loads and reduction factors are not applied, otherwise the constraints should include formulas from Structural Design Codes (Euro Code, AISC-LRFD, NTC 2008 etc.)

Displacement limitations impose maximum values of deformation: $|[u]| \leq [\Delta]_{max}$

The variables x_i of every section should satisfy geometrical criteria. They can be continuous or discrete. Discrete sizing is more realistic since sections are taken from a commercial list. There are lower and upper limits to the sections equation 3.

$$[x]_{lower} \leq [x] \leq [x]_{upper} \tag{3}$$

In this study, buckling when considered, is applied using NTC 2008 formula for compressed members. The resistance force member is given following by the formula in equation 4.

$$N_{cr} = \frac{\pi^2 \cdot E \cdot J}{{l_0}^2};\ \lambda = \sqrt{\frac{A \cdot f_{yd}}{N_{cr}}};\ \Phi = 0{,}5 \cdot [1 + 0{,}34 \cdot (\lambda - 0{,}2) + \lambda^2]\ ; \tag{4}$$
$$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - \lambda^2}};\quad N_{b,Rd} = \frac{\chi \cdot A \cdot f_{yk}}{\gamma_{M1}};$$

$E - elastic\ modulus;\ \ l_0 - effective\ length\ column;\ \ J - moment\ of\ inertia; A - section\ area;\ f_{yk}, f_{yd} - characteristics\ and\ design\ strength\ value;\ \Phi, \lambda, \chi - coefficients.$

The structural design problem is a constrained optimization problem. The solution is found applying algorithms which satisfy the given criteria, and seeks the best optimal solution. The objective function in genetic algorithms is given by the fitness function.

4. TRUSS MATRIX STRUCTURAL ANALYSIS

In general, the truss bars should satisfy the equilibrium condition equation 5 and the stiffness matrix equations

$$\{K\}[u] = [P] \qquad (direct\ stifness\ method)$$

$$\{C\}[F] = [P] \quad equilibrium\ conditions\ (methods\ of\ joints) \tag{5}$$

$$\{K\}[u] = [P] \qquad (direct\ stifness\ method) \tag{6}$$

$F - member\ forces,\ \ C - truss\ geometry\ matrix,\ \ K - stiffnex\ matrix,$
$P - joint\ external\ loads$

The direct stiffness method consist in evaluating the stress-deformation of the truss using the stiffness matrix. The global matrix of the structure $\{K\}$ is found by joining and reducing the single member matrixes of the truss, respective to the global system. Since some joint loads and displacements are known and some are not, the direct stiffness equation can be partitioned as given in equation 7.

$$\begin{Bmatrix} \{K_{11}\} & \{K_{12}\} \\ \{K_{21}\} & \{K_{22}\} \end{Bmatrix} \begin{bmatrix} [u_u] \\ [u_k] \end{bmatrix} = \begin{bmatrix} [P_u] \\ [P_k] \end{bmatrix} \tag{7}$$

where $[P_k]$ denote the vector of known forces, $[u_k]$ of known displacements, and $[P_u]$, $[u_u]$ unknown forces and displacements. Applying some matrix manipulation is possible to obtain the force members and final displacements of control joints.

5. MATLAB SOFT R2017A

MATLAB SOFT R2017 is the last version of the software. It is a multi-paradigm numerical computing environment and a fourth-generation programming language, developed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces with other language programming such as C, C++, C#, Java, Fortran and Python.

More information can be found on www.mathworks.com . Both commercial and educational versions are available. For example a simple script of mapping values in MATLAB applied in this study is:

```
function x=discretevariable(x)
global AV
% Mapping of discrete variable
% The possible values for all x are from AV
x=AV(x);
```

MATLAB can easily work on matrix analysis. Several authors have used MATLAB to operate on structural optimization problems. Cazacu, 2014 [22] has used MATLAB environment to optimize steel trusses using genetic algorithms and FEA. Hultman, 2010 [25] prepared a Report on weight minimization of steel trusses applying genetic algorithms.

MATLAB can also be applied in collaboration with other open source structural design softs such as SAP2000. Wu, 2012 [26], proposed a procedure for wind-resistant optimization design of long-span portal-rigid frame by adopting the Optimality Criteria (OC) method and SAP2000 Application Programming Interface (API) for MATLAB developing environment.

MATLAB has some toolboxes of optimization, the “Optimization Toolbox” and the “Global Optimization Toolbox”, which allow to perform some optimization problem. Some functions can be incorporated in the programming language of scripts.

6. GENETIC ALGORITHM APPROACH

Genetic algorithm (GA) are part of the evolutionary algorithms group. (GA)s generate a series of new designs based on Darwin's survival of the fittest theory, seeking to mimic genetics. At each iteration, parents are selected among the best designs and the values of their optimization variables are mixed to generate children designs, and random changes are also applied to prevent early convergence of the population, using genetic crossovers and mutations. The best children are added to the population, while old and less fit designs are removed from it. Before starting the (GA) procedure, it is necessary to:

(1) Define the objective function and Constraints of the problem .
(2) Code design variables into strings. Every variable is coded in strings of bits 10101. The variables then are concatenated in a single string to define a potential solution of the problem.
(3) The fitness function f(x) can be the total weight of the structure. It determine how the algorithm choses the best individuals in the population.

Most (GA)s are variations of the simple genetic algorithm (SGA) proposed by Goldberg in 1989 [15]. Its general procedure consist in the following steps:

a. [Start] Generate random population of n chromosome (strings of 101010100110101, each of one corespond to a potential solution of the problem)
b. [Fitness] Evaluate the fitness function f(x) for each chromosome in the population
c. [New population] Create the new population picking parents among the best individuals applying the (GA) operators:
    i. [Selection] Selection of two parents from a population according to their fitness (best fitness, more chance to be selected etc.)
    ii. [Crossover] Generate children by mixing the parents properties with a crossover probability. If no crossover is applied the string is an exact copy of the parents.
    iii. [Mutation] Apply with a mutation probability changes to the children properties at each locus.
    iv. [Accepting] Place the new strings in the population.
d. [Replace] Use new generated population for a further run for the algorithm.
e. *[Test] If the end condition is satissfied, stop and give the best solution in the current population.*

Goldberg (GA)s consists in three genetic operators: reproduction, crossover and mutation.

A. The selection scheme

The selection or reproduction operator is the basic engine of the algorithm. The objective of the reproduction process is to allow the information stored in strings with good fitness values to survive into the next generations. Typically, each string is assigned a probability of being selected as a parent string based on string's fitness. Rank-based selection scheme is largely used. Given the population, the selection scheme starts by sorting the population according to the values of the fitness function, constructing a ranking, where better solution have a higher rank. Individuals in the population are then selected in such a way, than higher the ranking, higher the probability of being chosen for reproduction.

B. The Crossover Operators

The Crossover operators specify how the genetic algorithm combines, to form a crossover child for the next generation. In this study is used a "scattered" approach. This operator, creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child.

C. The Mutation Operator.

Mutation is a guarantee that some important regions of the search space may never be explored. Non-existing features from both parent strings may be created and passed to their children. The mutation operation can be beneficial in reintroducing diversity in a population.

After the recombination step and again inspired in Nature, a mutation operator is introduced to simulate the errors that may arise during the copy process. The mutation operator is applied to each bit in the offspring chromosomes with a probability p_m. The effect of this operator is a simple change of 1 into a 0 and vice-versa.

To apply the (GA) it is further necessary to set up the population size. A population size of 150-1000 individuals (potential solutions) can applied with a maximum number of generations of the algorithm in the range 200-1000, a crossover probability of 0.5-0.8 and mutation probability of 0.001-0.05.

7. METHODOLOGY USED IN THIS STUDY

The optimization of the steel schemes will be done applying a genetic algorithm to the matrix analysis equations of the deformation method, with stresses, deformation and buckling constraints. The analysis is done in MatLab soft R2017a.

Three algorithms are built: the first one with stresses and displacement criteria is used in the analysis of seeking the minimal total weight of a benchmark truss schemes and put in comparison with other previous studies. It consists in 4 scripts with function of Data.m, fitness function.m, constraint function.m, (GA) general parameters.m.

The other two includes also buckling constraints for the members in compression. The algorithm is applied with discrete variables and results are put in comparison with previous studies.

A. Coding and decoding

All substrings are decoded and mapped to some integer values representing the sequence numbers of commercial steel sections in a given profile list. A multi parameter mapping consists in the equation 8.

$$I_i = I_{min} + \left(\frac{I_{max} - I_{min}}{2^l - 1}\right) \cdot \Omega \tag{8}$$

$l - is\ the\ length\ of\ the\ substring, \quad I_{min}, I_{max} - sequential\ number\ of\ the\ first\ and\ last\ section$
$\Omega - binar\ coding\ convertion\ (example) \quad 10100 = 2^4 + 2^2 = 20$

B. Fitness and penalty function

Fitness function is the total weight and it is applied to determine the selection process as shown in equation 9:

$$1/\,f_i = W = \sum_e^N \rho_e\, L_e A(\eta_e) \tag{9}$$

Probability of selection is found based on the fitness function equation 10.

$$p_C = \frac{f_i}{\sum f_i} \tag{10}$$

Penalty function is a linear function applied to limit the generation of individuals that do not satisfy constraint requirements. The penalty function is defined from equation 11.

$$\emptyset_i = \begin{cases} 1 & if\ \left|\frac{p_i}{p_{max}}\right| \le 1 \\ \frac{k|p_i|}{p_{max}} & if\ \left|\frac{p_i}{p_{max}}\right| > 1 \end{cases} \tag{11}$$

$\emptyset_i -\ is\ the\ penalty\ function\ for\ constraints\ i,\ \ p_i -\ factored\ stresses\ or\ displacements,$
$p_{max} -\ maximum\ resistance\ value, and\ k_i - penalty\ factor.$

C. Crossover and mutation probability

Various values of crossover and mutation are applied. They vary 0.6-0.8 for crossover and 0.01-0.03 for mutation. Higher values can retard the time processing of the algorithm.

D. Population size, max generations and tolerance

In this study, the (GA) parameters of population size, max generations, elite count and tolerances are considered with values as follow:

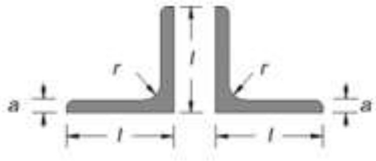
options = optimoptions(@ga,'PopulationSize', 100-200, 'MaxGenerations', 100-400, 'EliteCount', 10,'FunctionTolerance', 1e-6);

8. CASE STUDIES

Case studies are taken from previous studies of optimization. A comparison between the different results obtained is given. The optimization problem is discrete with variables that can have values only from standard commercial sections.

```
% Available sections
AV=[616 758 896 700 860 1018 1172 778 960 1138 1312 1262 1646 1164 1382 1806 2220 1740 2200 1660 1880 2380 2860 2300 2820 3340 2460 3020 3580 2780 3100 3740 3420 3900 3840 4540 4120 5920 4240 5020 5800 4640 5080 5940 6780 8020 6000 6940 7860 7000 8000 9000 8060 9140 10200 11300]; %mm2 sections double L www.oppo.it
% Minimal moment of inertia
I=[89400 110600 126200 128600 156800 183200 208000 179400 220000 256000 292000 346000 440000 390000 456000 584000 698000 668000 828000 742000 846000 1050000 1236000 1178000 1424000 1652000 1444000 1750000 2040000 2080000 2320000 2760000
```

3160000 3540000 3540000 4140000 4700000 5240000 4780000 5600000 6380000 6260000 6820000 7880000 8900000 10340000 9440000 10800000 12100000 12780000 14460000 16100000 16900000 19000000 21000000 284840000];



A. Case study Nr.1. 10 bar truss with stress and displacement criteria

Case Nr.1 is a typical case study, analyzed by most authors in structural optimization. A ten bar cantilever truss is given with loads at lower points. Due to its simple configuration the ten bar truss has been used as a benchmark to verify the efficiency of diverse optimization methods, with stresses and displacements criteria.

Case 1a.

The case study is given in Figure 1. This scheme was firstly analyzed by Venkayya and other authors in 1971 and 1979 [27, 28]. The scheme was analyzed for aluminium alloy with mass density $\rho = 27.14\ kN/m^3$, elastic modulus $E = 10{,}000\ ksi\ (68.95\ GPa)$, stress limits of $25\ ksi$ (172.37 MPa), and displacements of $2\ inch\ (50.8mm)$. After 25 cycles the study reported an optimal result of 2306 kg.

Rayeev and Krishnamoorthy [29] in 1992, used a modifiable simple genetic algorithm to optimize the scheme. The result reported a best total weight of 2 497 kg.

Pezeshk [18] in 1998, built a FEAPGEN algorithm using genetic algorithms. A six digit binary number is built to represent the cross sectional areas of each member. The parameters of the genetic algorithm were $\alpha = 1.005$, the crossover probability 0.85; the mutation probability 0.05. Small population sizes (20, 30 and 40) were applied and solutions were developed for 50 generations. The result reported a best design of 2 472 kg.

Flager and others in 2014 [30] proposed the fully constrained algorithm based on the virtual work principle, for discrete sizing optimization of steel structures that balances computation efficiency with solution quality for application to large-scale problems. The method was based on optimality criteria and didn't require gradient information. It was reported a minimal total weight of 2317 kg for the scheme.

Kazemzadeh and Hacansebi in 2014 [31] proposed the (GSS) Guided Stochastic Search, used for discrete sizing optimization of steel trusses. The method worked on the basis of guiding the optimization based on the virtual work principle and on the information collected during the structural analysis. The optimal best result reported was 2490 kg. The performance of the proposed techniques was investigated according to AISC – LRFD specifications.

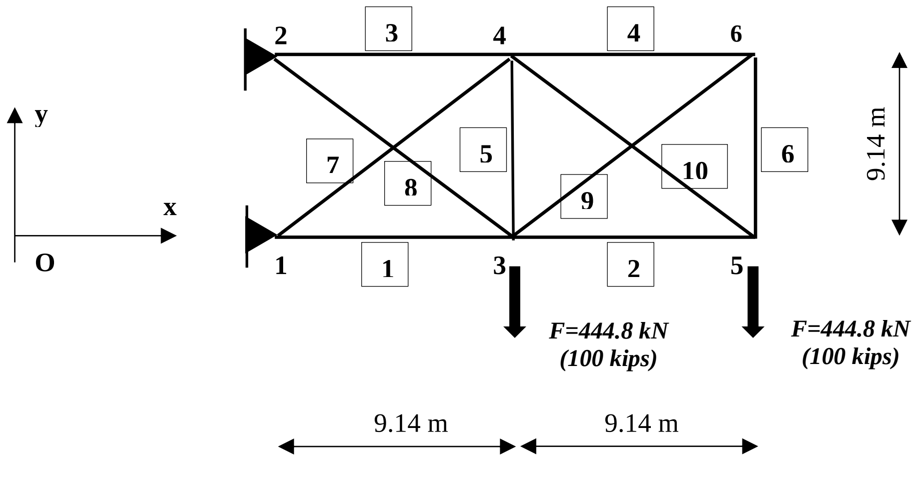


Figure. 1 First case study - configuration of a ten-bar truss

In this study the 10 bar truss geometry, was analyzed using the Genetic algorithm built in MATLAB soft R 2017a, with the same conditions. An optimal result of 2298 kg was reported in (Table.)

Table. 1 Case study nr.1a of the benchmark truss

Stresses and displacements criteria	Venkayya [26]	S. Rajeev and Krishnamoorthy [29]	Pezeshk study [18]	Flager and others in 2014 [30]	Kazemzadeh and Hacansebi in 2014 [31]	This study
Algorithm applied	Optimality Criteria	Modifiable simple genetic algorithm	FEAPGEN using genetic algorithm	Improved optimality criteria (Fully Constrained Design)	Guided Stochastic Search	Genetic algorithm Matlab soft R2017a
Minimum weight	2306 kg	2497 kg	2472 kg	2317 kg	2490 kg	**2298 kg**

Case 1b.

This Case is similar to the previous one, but with some modification on geometry and material characteristics taken from Stasa [10]. Constraints are imposed on stresses and displacements. Load is approximated to 500 kN. Stasa [10] applied two different algorithm methods: the FSD (Fully stressed design) and the SLP with move limits (Sequential Linear Programming method). Stresses, displacements, geometry and materials characteristics are considered the same. The lowest value of minimal total weight is considered for comparison.

In this study buckling constraints are applied from from NTC 2008 (7.METHODOLOGY USED IN THIS STUDY). Instead, Stasa [10] applied some empirical combination between the area of the section A and the minimum radius of gyration ρ_{min} of standard commercial sections, using a β coefficient equation 12.

$$\rho_{min} = \beta\sqrt{A} \tag{12}$$

The optimization problem is discrete with variables taking values from the standard commercial section list *(8.*CASE STUDIES*).* A function is built to map variables. Results are given in Table..

Table. 2 Results and comparison of case study 1b

	Stasa [10]	**This study**	**This study**	**Stasa [10]**	**This study**
Constraints	f_{yk}: 210 N/mm2 Δ: 3 cm Buckling approx.	f_{yk}: 210 N/mm2 Δ: 3 cm	f_{yk}: 210 N/mm2 Δ: 3 cm Buckling NTC 2008	f_{yk}: 210 N/mm2 Δ: 1 cm	f_{yk}: 210 N/mm2 Δ: 1 cm Buckling NTC 2008
Minimum weight	**888.08 kg**	**698 kg**	**1099 kg**	**1504.52 kg**	**1453 kg**

B. Case study Nr.2. English Truss with stresses and buckling constraints

This case consist in the analysis of an English steel roof truss (Figure 2) with dimensions of 16.5m x 2.2m. Truss nodes are numbered from 1 to 16 and elements from 1 to 29. Lower elements 1-8 connect nodes : 1-2,2-4,4-6,6-8,8-10,10-12,12-14,14-16; upper elements 9-16 connect nodes: 1-3,3-5,5-7,7-9,9-11,11-13,13-15,15-16; vertical elements 17-23 connect nodes: 2-3,4-5,6-7,8-9,10-11,12,-13,14-15 and diagonals 24-29 connect 2-5,4-7,6-9,9-10,11-12,13-14.

The truss has been analyzed applying the direct stiffness method. The design is based on the Italian Code NTC 2008, for steel S275. For the design are considered double L angular sections. Firstly, the truss is designed applying conventional methods, then is redesigned in MATLAB applying the genetic algorithm, built with constraints on stresses and buckling based on the NTC 2008. The fitness function is the total weight of the structure. The problem is discrete and variables are taken from the standard commercial sections list

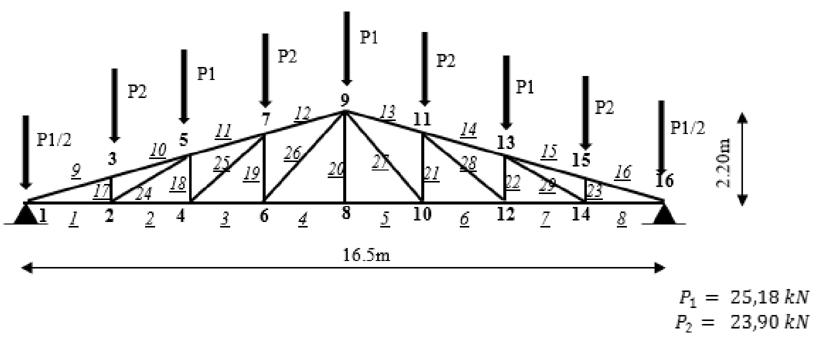


Figure. 2 Case Nr.2 English roof truss

(Case 1b). Results and a comparison with previous studies is given in Table 3.

Table. 3 Standard design of the roof truss under the NTC 2008 Code

Standard design under NTC 2008 Code			**(GA)s optimized in this study (mm^2)**
Elements /Data	Member forces (max values in kN)	Section design (mm)	**Different sections for every element:** *Total weight 749 kg* **Sections grouped:** *Total weight: 832 kg*
Lower elements	334.64 Kn	1312 mm^2	
Upper elements	-346.45kN (compression)	3840 mm^2	
Diagonals	72.42 Kn	778 mm^2	
Vertical elements	-52.95 kN (compression)	778 mm^2	
Standard design truss weight		$832\ kg$	Optimized weight $749 - 832\ kg$

Using genetic algorithms it is possible to optimize by 10% the total weight. Grouping elements may create some loops in the MatLab Code, so it should be written correctly developing the 4 variables. Difficulties may appear when setting up the design process.

C. Case study nr.3 - 39 bars, with stresses, displacements and buckling constraints

This case study is a 39 bar truss (Figure 3) with dimensions of 24.0 m x 1.0m (width x height). The truss is statically indeterminate, with steel material and a stress resistance of $f_{yd} = 210N/mm^2$. Buckling constraints are defined in (7. METHODOLOGY USED IN THIS STUDY). There are considered two cases for displacements, the first one with a limit of $\Delta_1 = 96mm$ and the second one with $\Delta_2 = 40mm$. Two load combinations are applied to the truss:

$$(1): P_1 = P_2 = 75.4\ kN \quad (2): P_1 = 75.4\ kN\ P_2 = 65.95kN$$

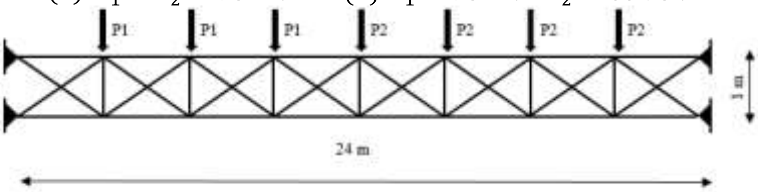


Figure. 3 Case study – 39 bars truss.

Due to the high number of variables and to the high range of the set of values, in this operations was observed a need to consider a higher value for the 'EliteCount' function in the (GA) operator: 'EliteCount', 50. Elite count specifies the number of individuals that are guaranteed to survive to the next generation. It should be a positive integer less than or equal to the population size. The default value is 5% of the population size. Higher is this value, more are the individuals that survive in the next generations. Population size and max generations are defined to 200: 'PopulationSize', 200, 'MaxGenerations', 200. The first loading combination gives greater values of the sectional areas. The best results that satisfy all the constraints are presented in Table 4.

Table. 4 Results and comparison for case study Nr.3 – 39 bars

	Stasa [10] Variant 1 $\Delta_1 = 96mm$ (best result mm^2)	**This study** Variant 1 (mm^2) $\Delta_1 = 96mm$	**Stasa [10]** Variant 2 $\Delta_2 = 40mm$ (best result mm^2)	**This study** Variant 2 (mm^2) $\Delta_2 = 40mm$
Total Weight	2657 kg	2356 kg	4032 kg	3514 kg
Number of iterations	2 cycles	200 cycles 16 000 function evaluations 90 sec.	8 cycles	200 cycles 14 000 function evaluations 104 sec.

The analysis conducted for this case resulted in an increase in efficiency in the design of the truss. The results are compared for the same conditions, with the example analyzed by Stasa [10]. It is reported an improvement in the value of the objective function; the total weight of the truss is 10% less. In contrast, a large number of iterations and a longer time is needed to achieve the result, compared to other examples given.

9. DISCUSSION AND CONCLUSIONS

Genetic algorithms have been largely applied in the engineering design problems. In this study a (GA) was built in MatLab soft R2017a, with fitness function, the total weight of the structure. The truss schemes were solved applying the direct stifness method. Different values of (GA) parameters were analyzed in a such way to achieve better results in terms of efficiency and efficacy. Three algorithms were applied with constraints on stresses and displacements, stresses and buckling and all three of them simultaneously.

(GA)s as meta-heuristic show advantages in comparison to deterministic methods since they explore a larger space of values. In this study were reported better total weight designs of more than 10% in comparison to other studies, for some truss schemes.

The bar elements of three different truss schemes were calculated and designed. The first one is a benchmark optimization problem, analyzed by most authors. In this case, since the number of variables is low, the algorithm process was fast, and it carried out also better results than the other optimization cases taken in consideration. Another similiar scheme was analyzed for steel sections and put in comparison with the design done by another author. In the second and third case, the optimization problem was applied to a roof english truss and a 39 bars truss, with constraints on stresses, displacements and buckling. There were reported good results too.

(GA) offers an instrument with a high potential in increasing the design efficiency. There are some difficulties in writing down the optimization problem, since every case may have some particularities. But, once the algorithm has been written, it can be applied successfully in the analysis and the design of various trusses. A longer time may be necessary for further processing of the algorithm for obtaining a better result. To get better designs, it is necessary to analyze the broad range of possible values that algorithm parameters can take.

Various authors have analyzed and continue to analyze these algorithms. Continuous improvements are reported in the way the Fitness function and the Penalty function is applied in finding a better solution in a shorter time. Use of these algorithms in the engineering practice would be higher, if there are built more readible interfaces with a variety of options. Further analyzes are needed to build fuller algorithms that allows in an automated way to solve problems facilitating with graphical interfaces. Topological and shape optimization can also be applied with size optimization, by starting a simultaneous process in finding the most favorable configuration, filling all the boundary conditions, constraints and loads.

ACKNOWLEDGEMENT

The author is thankful to the Polytechnic University of Tirana, Albania in order to provide research support towards completion of this research.

REFERENCES

1. Christensen, P.W. and A. Klarbring, *An introduction to structural optimization*. Vol. 153. 2008: Springer Science & Business Media.
2. Schmit, L.A. *Structural design by systematic synthesis*. in *Proceedings of the 2nd conference on electronic computation, ASCE, New York*. 1960.
3. F. Bontempi, *Phd course in structural optimization*. 2016: Rome: Dottorato in Ing. Strutt. e Geotec. Universita' degli Studi di Roma "La Sapienza".
4. Kirsch, U., *Fundamentals and applications of structural optimization*. 1993, Springer-Verlag, Heidelberg.
5. Lako, E.T.a.N. *A survery of size optimization algorithms applied in the structural engineering design*. in *1st International Conference on New Research and Advances on Computer Science and Information Technology*. 2017. Vlora, Albania.
6. Arora, J.S., *Methods for discrete variable structural optimization*, in *Advanced Technology in Structural Engineering*. 2000. p. 1-8.
7. Kicinger, R., T. Arciszewski, and K. De Jong, *Evolutionary computation and structural design: A survey of the state-of-the-art.* Computers & Structures, 2005. **83**(23): p. 1943-1978.
8. Azad, S.K. and O. Hasançebi, *Optimum Design of Skeletal Structures Using Metaheuristics: A Survey of the State-of-the-Art.* 2014.
9. Rolvink, A., J. Coenders, and C. Mueller. *State on the art of computational tools for conceptual structural design*. in *Proceedings of the IASS-SLTE 2014 Symposium" Shells, Membranes and Spatial Structures: Footprints", Brasilia, Brazil, 15-19 September 2014*. 2014: International Association for Shell and Spatial Structures (IASS).

10. Stasa, K., *Projektimi Optimal i Kapriatave Metalike*. 1994, Tirana, Albania: Universiteti Politeknik i Tiranes (UPT).
11. Hasançebi, O., et al., *Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures.* Computers & Structures, 2009. **87**(5): p. 284-302.
12. Chen, Y., *Application of deterministic operations research for structural optimization*. 2015, Massachusetts Institute of Technology.
13. Pezeshk, S. and C.V. Camp, *State of the art on the use of genetic algorithms in design of steel structures.* Recent advances in optimal structural design, 2002: p. 55-80.
14. Holland, J.H., *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence.* Ann Arbor, MI: University of Michigan Press, 1975.
15. Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning, 1989.* Reading: Addison-Wesley, 1989.
16. Lin, C.-Y. and P. Hajela, *Design optimization with advanced genetic search strategies.* Advances in Engineering Software, 1994. **21**(3): p. 179-189.
17. Chan, C.-M. and P. Liu, *Design optimization of practical tall concrete buildings using Hybrid Optimality Criteria and Genetic Algorithms*, in *Computing in Civil and Building Engineering (2000)*. 2000. p. 263-270.
18. Camp, C., S. Pezeshk, and G. Cao, *Optimized design of two-dimensional structures using a genetic algorithm.* Journal of structural engineering, 1998. **124**(5): p. 551-559.
19. Kameshki, E. and M. Saka, *Genetic algorithm based optimum bracing design of non-swaying tall plane frames.* Journal of Constructional Steel Research, 2001. **57**(10): p. 1081-1097.
20. Petrucci, M., *Structural optimization: an approach based on genetic algorithms and parallel computing*. 2009, University of Trento.
21. Toğan, V. and A.T. Daloğlu, *An improved genetic algorithm with initial population strategy and self-adaptive member grouping.* Computers & Structures, 2008. **86**(11): p. 1204-1218.
22. Cazacu, R. and L. Grama, *Steel truss optimization using genetic algorithms and FEA.* Procedia Technology, 2014. **12**: p. 339-346.
23. Dhingra, A.K. and B. Lee, *A genetic algorithm approach to single and multiobjective structural optimization with discrete–continuous variables.* International Journal for Numerical Methods in Engineering, 1994. **37**(23): p. 4059-4080.
24. Adeli, H. and N.-T. Cheng, *Integrated genetic algorithm for optimization of space structures.* Journal of Aerospace Engineering, 1993. **6**(4): p. 315-328.
25. Hultman, M., *Weight optimization of steel trusses by a genetic algorithm.* Lund University, Department of Structural Engineering, 2010.
26. Wu, J., et al. *Structural optimization of long span portal-rigid frames under wind action*. in *The Seventh International Colloquium on Bluff Body Aerodynamics and Applications (BBAA7) Shanghai, China*. 2012.
27. Venkayya, V., *Design of optimum structures.* Computers & Structures, 1971. **1**(1-2): p. 265-309.
28. NS Khot, L.B. and V.B. Venkayya, *Comparison of optimality criteria algorithms for minimum weight design of structures.* AIAA Journal, 1979. **17**(2): p. 182-190.
29. Rajeev, S. and C. Krishnamoorthy, *Discrete optimization of structures using genetic algorithms.* Journal of structural engineering, 1992. **118**(5): p. 1233-1250.
30. Flager, F., et al., *Fully Constrained Design: A general and scalable method for discrete member sizing optimization of steel truss structures.* Computers & Structures, 2014. **140**: p. 55-65.
31. Azad, S.K., O. Hasançebi, and M. Saka, *Guided stochastic search technique for discrete sizing optimization of steel trusses: A design-driven heuristic approach.* Computers & Structures, 2014. **134**: p. 62-74.

AUTHOR PROFILE

Ersilio Tushaj is a PhD Candidate at the Polytechnic University of Tirana in Civil Engineering. He currently works as lecturer at POLIS University, International School of Architecture and Urban Planning. He has a master degree from University of Bologna in Building Engineering and Architecture. He has some years of academic experience and is author of some publications. His main interest area structural optimization and application of algorithms