SEECCM III
3rd South-East European Conference on Computational Mechanics-
an ECCOMAS and IACM Special Interest Conference
M. Papadrakakis, M. Kojic, I. Tuncer (eds.)
Kos Island, Greece, 12–14 June 2013

DISCRETE OPTIMIZATION APPROACH FOR STEEL FRAMES AND TRUSSES, BASED ON GENETIC ALGORITHM

Ioana D. Balea [1], Radu Hulea[1], and Georgios E. Stavroulakis [2]

[1] Technical University of Cluj-Napoca
400020 Cluj-Napoca, Romania
e-mail: ioana.balea@mecon.utcluj.ro

[2] Technical University of Crete
Kounoupidiana, Akrotiri, Greece
gestavr@dpem.tuc.gr

Keywords: optimization, steel, eurocode, genetic algorithms

Abstract. *This paper presents an implementation of Eurocode load cases for discrete global optimization algorithm for planar and space structures based on the principles of finite element methods and genetic algorithms. The final optimal design is obtained using IPE sections chosen as feasible by the algorithm, from the available steel sections from industry, used for ease of comparison with benchmarks. The algorithm is tested on several planar steel frames and a truss from the literature, with good results.*

1 INTRODUCTION

In all engineering fields, designers attempt to find solutions that combine performance and satisfaction of critical requirements. But these techniques and requirements are subject to constant change. The oldest surviving books on engineering are two thousand year old - Ten Books on Architecture by Vitruvius. These demonstrate that the way engineers thought about design in ancient times appears to have been fundamentally the same as how we think today, and the problems faced at the time and in that technological context, just as complex as those we face today. So, for the past two thousand years, engineers have been tackling design problems defined by competing goals and bounded by technical, aesthetic and economic constraints. Architects and engineers have always had to be masters of a wide variety of knowledge and skill, to be broadly interdisciplinary people. The design of structures in ancient times was based on geometry, and, to whichever form was used for buildings, ancient engineers found limits of use defined by failure. And, just as the ancient Greeks and Romans learned by the method of trial and error and deduced rules of thumb, accumulating knowledge and engineering advances today guides us in extending the state of the art towards new limits. Though engineers may not be building now ancient monuments, our contemporary masterpieces of engineering design and technology have their origins in the same type of conceptualization processes. A design with loose constraints might float off into irrelevance. Constraint offers an opportunity for growth and innovation. Designers can obtain the optimum within the imposed conditions by using optimization techniques. In the field of structural engineering, structures designed in this way are safer, more reliable and less expensive than the traditional designed ones, where the success of the design is based solely on the experience of the engineer. Optimization techniques require some expertise, but with the implementation of these algorithms in computer aided design software it can become a very powerful tool in the hands of engineers. In general, the optimization techniques used in structural design can be categorized into classical and heuristic search methods.

Classical optimization methods such as linear programming, nonlinear programming and optimality criteria often require substantial gradient information. In these methods the final results depend on the initially selected points and the number of computational operations increases as the size of the structure increases. The solution in these methods does not necessarily correspond to the global optimum. Many engineering design problems are too complex to be handled with mathematical programming methods. In comparison, heuristic search methods do not require the data as in the conventional mathematical programming and have better global search abilities than the classical optimization algorithms [1]. For the past 60 years a new branch of optimization techniques was continuously developed, which mimics the design methods existing in nature. Genetic algorithms, simulated annealing and evolutionary strategies are among such algorithms that are used in the design optimization of structures. Among these, genetic algorithms, are a search method that is based on the principal of the survival of the fittest and adaptation. They operate on a population of design variables sets. Each population consists of individuals that are potential solutions to the design problem. A fitness value is calculated for each individual using the objective function and constraints as a measure of performance of the design variables. If the individual is fit, it is selected as candidate to take part in the construction of the next population, if not then it is discarded. The solution to a general optimization process can be associated with this system behavior. The genetic algorithm (GA) is one of the best-known heuristic methods; it has been used to solve structural optimization problems by some researches such as Rajeev and Krishnamoorthy [2], Saka and Kameshki [3], Camp et al. [4], Pezeshk et al. [5], Erbatur et al. [6], Shook et al. [7], among many others.

2 PLANAR FRAME AND TRUSS OPTIMIZATION PROBLEMS

The structural optimization problem could be formulated as follows, for a structure that consists of M nodes and N bars with cross-sectional areas: Ai, i=1,2,…,N. All cross-sectional areas Ai compose the vector x of the optimization parameters:

$$\mathbf{x} = [A_1 \quad A_2 \quad \dots \quad A_N]^T \tag{1}$$

The problem to be solved refers to the computation of the vector x achieving the minimum weight W of the frame or truss

$$W = \sum_{i=1}^{N} \rho A_i L_i \tag{2}$$

for given stress- and deflection-constraints (dis-placement-constrained trusses):

$$\begin{aligned} \sigma_i &\le \sigma_{0i} \qquad 1 \le i \le N \\ u_i &\le u_{0i} \qquad 1 \le i \le M \end{aligned} \tag{3}$$

In equation (3), s_{0i} and u_{0i} denote the allowable stress and displacement upper limits, respectively.

3 EUROCODE PROVISIONS

Constraints regarding material strength and stability are taken from "EN 1993: Design of steel structures" [8] and implemented in the algorithm. Resistance must be checked by verification by the partial factor method. The design resistance for cross-sections is:

$$N_{c,Rd} = \frac{Af_y}{\gamma_{M0}}$$

The design should satisfy the requirements under the most unfavorable loading cases considered. Elastic analysis is routinely used to obtain member forces for subsequent use in the member checks based on the ultimate strength considerations. This is well accepted, can be shown to lead to safe solutions, and has the great advantage that superposition of results may be used when considering different load cases. In the Eurocodes, partial factors g_{Mi} are applied to different components in various situations to reduce their resistances from characteristic values to design values (or, in practice, to ensure that the required level of safety is achieved). The uncertainties (material, geometry, modeling, etc.) associated with the prediction of resistance for a given case, as well as the chosen resistance model, dictate the value of g_M that is to be applied. The choice for the different loading scenarios was made considering all the predictable conditions and situations that can appear during the construction phase and the utilization phase of the building.

$$E_d = E \cdot (\gamma_{G,j} \cdot G_{k.j}; \gamma_{Q,1} \cdot Q_{k.1}; \gamma_{Q,i} \cdot \psi_{0,i} \cdot Q_{k.i}) \quad (2)$$

Where:
E_d – design load
γ – Partial coefficient
G – Permanent action
Q – Variable action
ψ – Coefficient variable action

4 THE OPTIMIZATION ALGORITHM

Evolutionary Computation (EC) is an enormous field of research concerned with the application of Evolutionary Algorithms (EA) to complex real-world optimization problems (for a literature review see Kicinger [9]). These methods are stochastic search algorithms modeling the natural phenomenon of evolution, a combination of the Darwinian concept of survival of the fittest and the inheritance of genetic material within a species. Because of their heuristic nature they generally cannot guarantee to find a global optimum solution, however, they can be applied to highly complex models for which standard optimization methods (e.g. gradient based algorithms) are not applicable, and in practice they often yield promising results. The terminology of EC is inspired by the resemblance to biological processes and lots of terms are borrowed from genetics and cellular biology. A candidate design solution is called an individual and a set of such solutions is called population. The representation (parameterization, encoding) of an individual is called a genome consisting of a series of genes. For some algorithms the search space (genotype space) is explicitly separated from the solution space (phenotype space) and a link between the two is made by a mapping procedure that codes (phenotype to genotype space) and decodes (genotype to phenotype space) the candidate solutions. Producing new solutions by modifying individual solutions is referred to as mating or breeding and the resulting solutions are called offspring. A fitness value is assigned to each individual indicating its quality in the context of the given problem in order to compare different candidate solutions. Selection is then used in order to determine which of the parents and the offspring survive and are transferred to the new population, which is typically called a new generation. Although many different implementations of EAs exist, the basic concept of the so-called canonical EA forms the basis of all of them. The canonical EA can be understood as a search process in which a population experiences gradual changes and consists of the following steps:

```
Canonical EA
1: t=0
2: Initialize the population
3: Evaluate the entire population
4: while (not termination condition) do
5: t=t+1
6: Mating selection
7: Apply variation operators
8:Evaluate offspring
9: Select individuals for survival
10: end while
```

The mating selection process favors individuals with higher fitness values over candidate solutions with below-average fitness values, or for minimization problems a lower fitness value indicates better quality of the candidate solution. New individuals are created by copying them and applying genetic variation operators. The most used operators are mutation and recombination (crossover). The steps 5 to 9 of the algorithm are repeated until a given termination condition, e.g. a maximum number of generations or evaluations or a target fitness value is met.

Population size	*300*
Crossover parameter	0.7
Mutation parameter	0.1
Maximum number of iterations	100

Table 1: The specifications of the used GA method

The optimization problems addressed within this paper have a single objective, but they are also subjected to an arbitrary number of constraints. The number of optimization parameters defines the size of the genotype and accordingly the size of the search space. A fitness function mapping the evaluated stiffness, strength and mass measures from the FE analysis to a unique fitness value was defined.

5 NUMERICAL IMPLEMENTATION

The first example refers to nine variables (nine-bar truss), the second to eight design variables (eight-element frame), and finally the third to 105 variables (105-bar truss).

In total, 20 program runs were performed, of which 5 are presented.

The profiles used in the optimization procedures are presented bellow

5.1 The first optimization problem : frame 8 bars 8 nodes

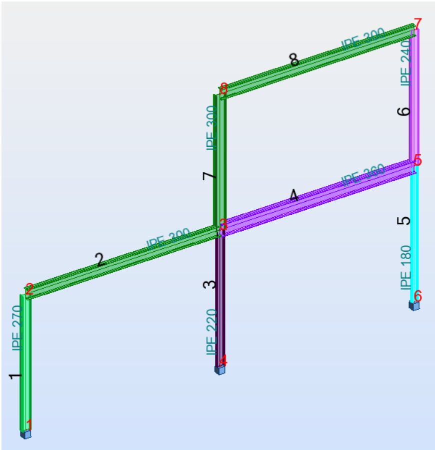

Figure 1: Structure no. 1 – 8 bars

The problem has 8 variables represented by cross sections of the elements.

Bar	Uniformly distributed load							
	Permanent loads		Live Loads		S now Loads		Wind Loads	
	X	Z	X	Z	X	Z	X	Z
1	0	0	0	0	0	0	-2.5	0
2	0	-10	0	-15	0	-6	0	0
3	0	0	0	0	0	0	0	0
4	0	-10	0	-15	0	0	0	0
5	0	0	0	0	0	0	-5	0
6	0	0	0	0	0	0	-5	0
7	0	0	0	0	0	0	-2.5	0
8	0	-10	0	-15	0	-6	0	0

Table 2: Loads used for problem 1.

Bar	Profiles				
	Run 1	Run 2	Run 3	Run 4	Run 5
1	IPE 120	IPE 140	IPE 120	**IPE 140**	IPE 120
2	IPE 180	IPE 200	IPE 180	**IPE 200**	IPE 160
3	IPE 330	IPE 270	IPE 330	**IPE 300**	IPE 360
4	IPE 140	IPE 180	IPE 180	**IPE 140**	IPE 140
5	IPE 330	IPE 330	IPE 330	**IPE 330**	IPE 330
6	IPE 140	IPE 140	IPE 140	**IPE 120**	IPE 120
7	IPE 200	IPE 200	IPE 200	**IPE 180**	IPE 160
8	IPE 330	IPE 330	IPE 330	**IPE 330**	IPE 390
Total weight	1193.9	1200.4	1217.5	**1152.5**	1240.7

Table 3: Profiles chosen by algorithm

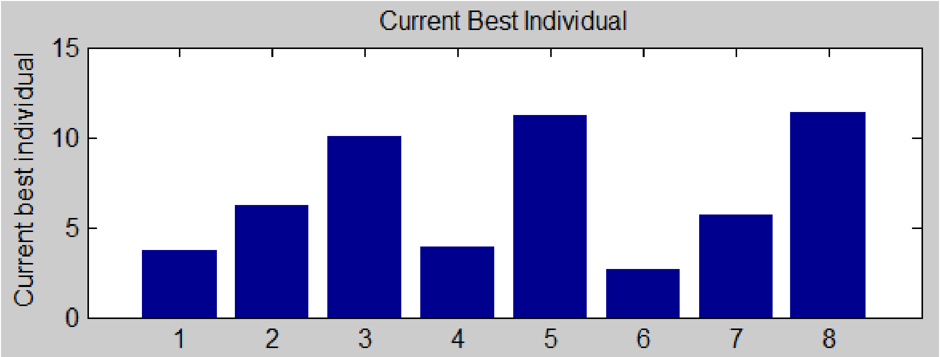

Figure 2: Best individual

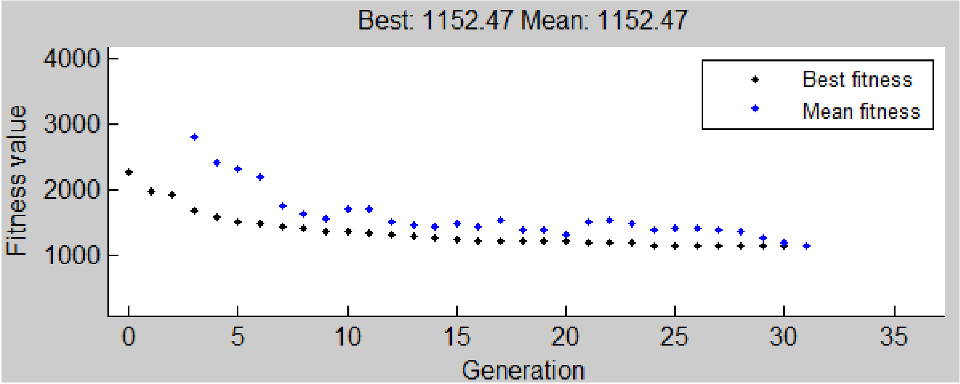

Figure 3: Convergence of algorithm

The structure was also optimized using Autodesk Robot Structural Analysis. By employing this software a total weight of 1466 Kg by comparison with the 1152Kg solution obtained when using the Matlab code.

5.2 The second optimization problem: truss 9 bars 6 nodes

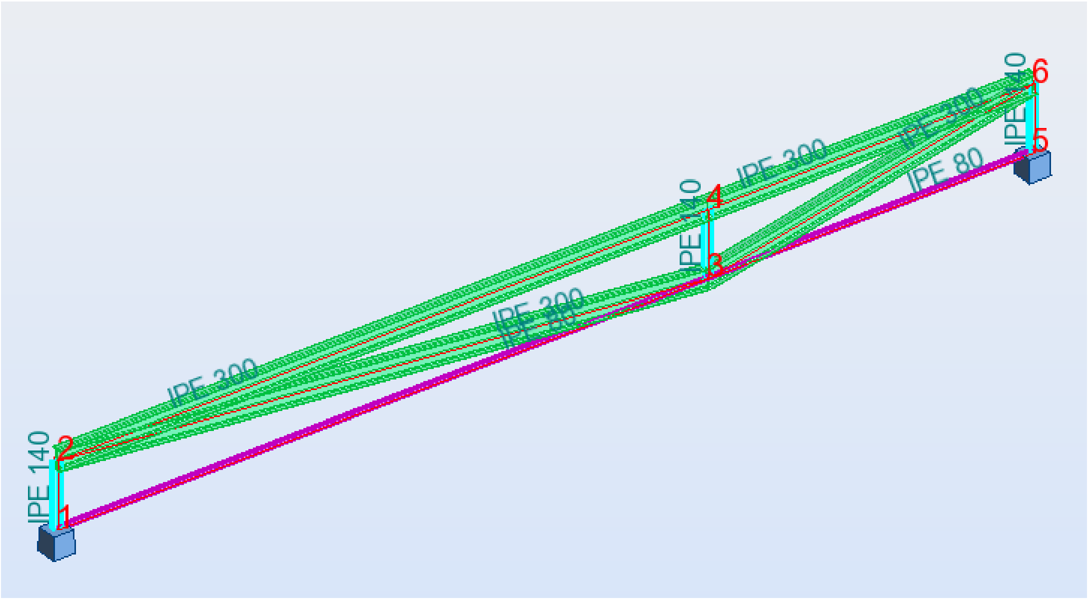

Figure 4: Structure no. 2 – 9 bars

The problem has 9 variables represented by cross sections of the elements.

Nodes	Permanent Loads	Live loads	Snow loads
	Z	Z	Z
1	0	0	0
2	-50	-75	-30
3	0	0	0
4	-75	-112.50	-45
5	0	0	0
6	-25	-37.5	-15

Table 4: Loads on structure

Bar	Profiles				
	Run 1	Run 2	Run 3	Run 4	Run 5
1	**IPE 80**	IPE 80	IPE 80	IPE 80	IPE 80
2	**IPE 80**	IPE 80	IPE 80	IPE 80	IPE 80
3	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300
4	**IPE 300**	IPE 330	IPE 300	IPE 300	IPE 330
5	**IPE 180**	IPE 180	IPE 140	IPE 140	IPE 140
6	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300
7	**IPE 180**	IPE 140	IPE 140	IPE 200	IPE 140
8	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300
9	**IPE 180**	IPE 220	IPE 140	IPE 160	IPE 140
Total weight	**1402.25**	1456.0	1402.25	1414.6	1436.8

Table 5: Profiles chosen by algorithm

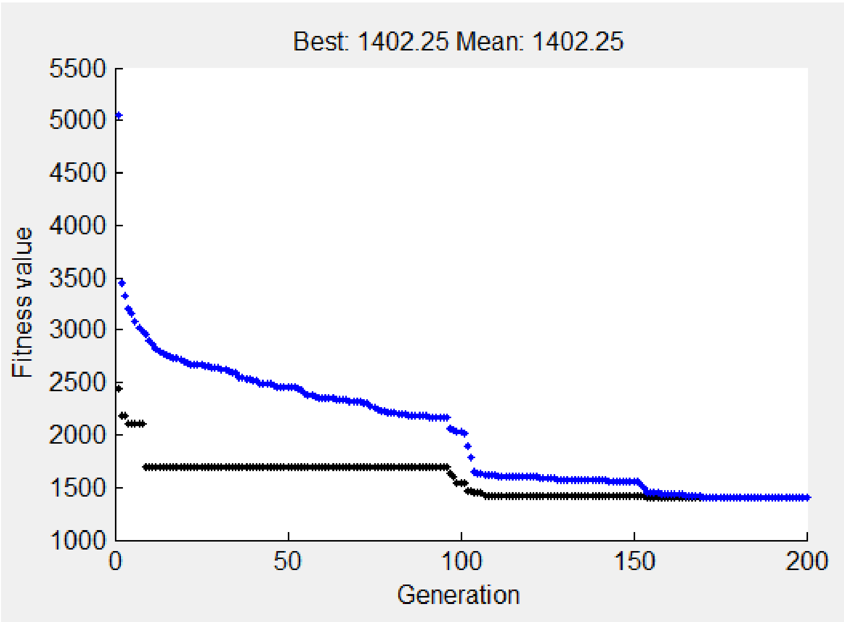

Figure 5: Convergence of algorithm

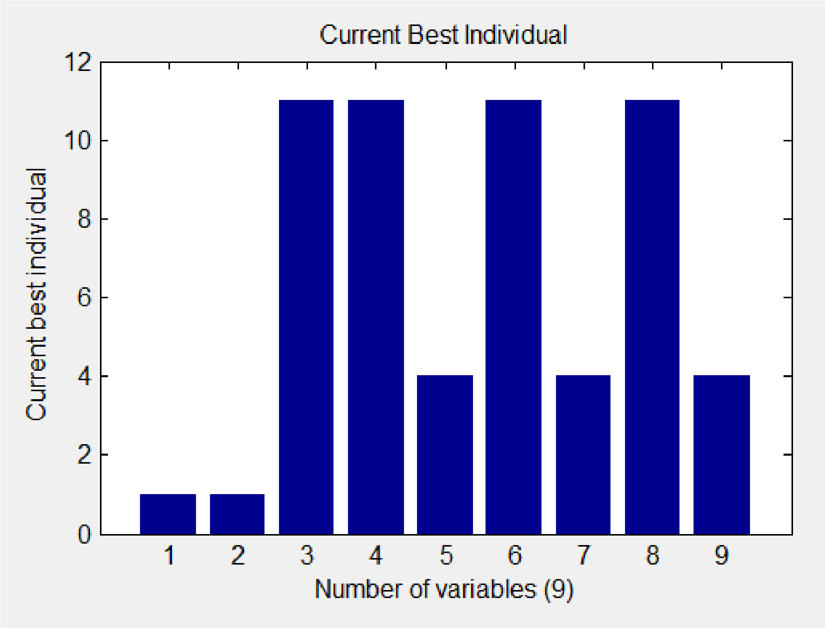

Figure 6: Best individual

The structure was also optimized using Autodesk Robot Structural Analysis. By employing this software a total weight of 1403 kg was obtained, close to the 1402 Kg solution obtained when using the algorithm developed in Matlab.

5.3 The third optimization problem: truss 120 bars 49 nodes

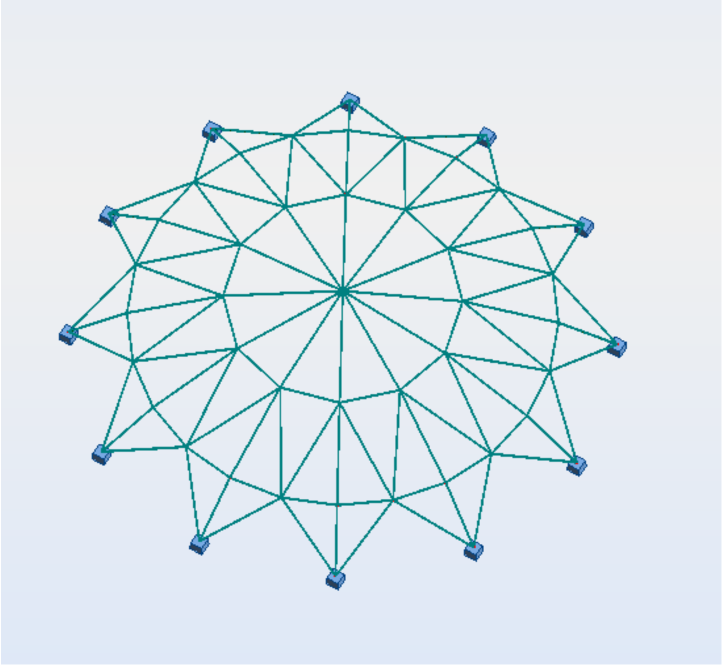

Figure 7: Structure no. 3 – 120 bars

The problem has 120 variables represented by cross sections of the elements.

Bar	Profiles				
	Run 1	Run 2	Run 3	Run 4	Run 5
Total weight	14008.1	15080.4	14941.6	14466.9	14953.9

Table 7: Profiles chosen by algorithm

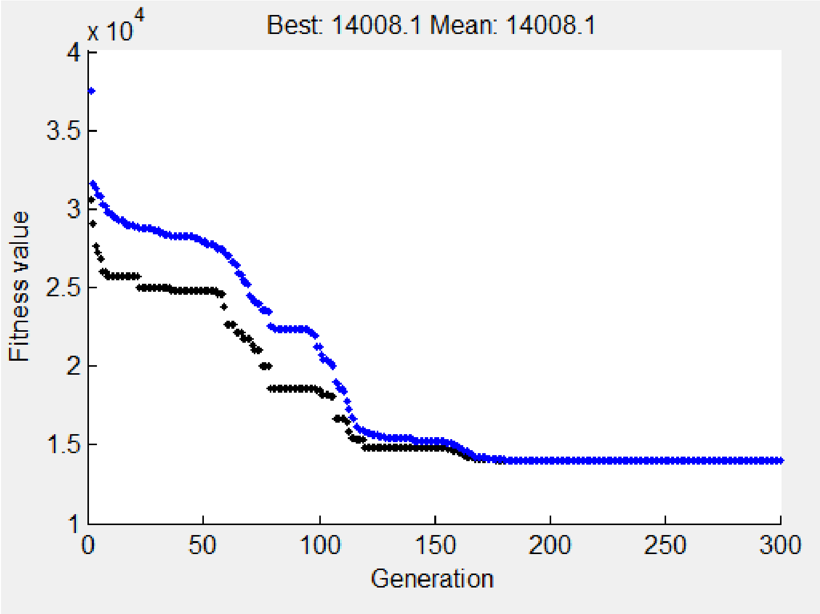

Figure 8: Convergence of the algorithm

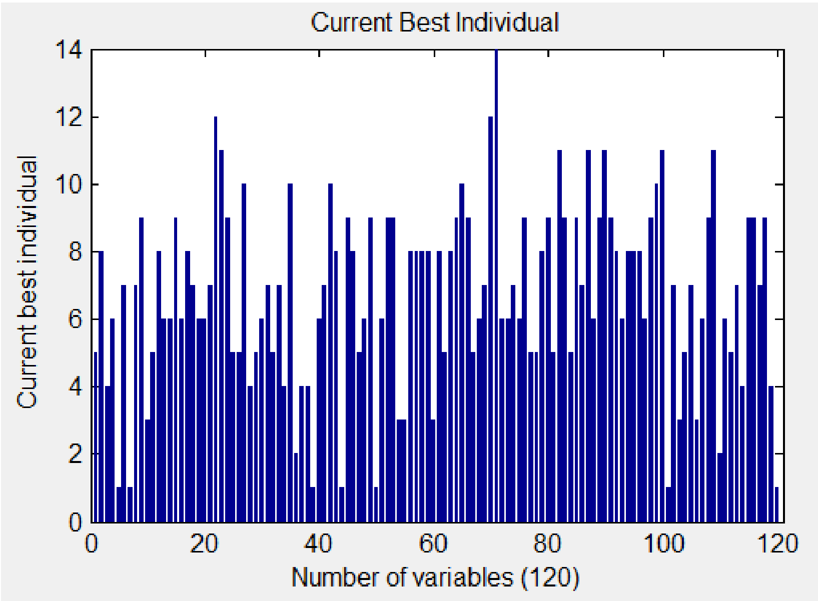

Figure 9: Best individual

6 CONCLUSIONS

Structural Optimization, while very interesting, is a tricky class of optimization problems, usually being characterized by a large number of variables that represent the shape of the design and the stiffness of the available materials. In any practical problem, the researcher deals with a unique combination of mentioned factors and has to decide what numerical tools to use or modify to reach the final goal. Therefore, the optimization of a structure always relies on both creativity and intuition, it is said that optimization combines both science and art.

Acknowledgements
This paper was supported by the project "Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Program Human Resources 2007-2013. A special thanks goes to the Technical University of Crete for hospitality and support.

REFERENCES

[1] A. Belegundu, "A Study of Mathematical Programming Methods for Structural Optimization. PhD thesis," University of Iowa, Dept. of Civil and Environmental Engineering, 1982.

[2] S. Rajeev and C. Krishnamoorthy, "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering,* vol. 118, no. 5, pp. 1233-1250, 1992.

[3] S. M.P. and K. E., "Optimum Design of Multi-Storey Sway Steel Frames to BS 5950 using a Genetic Algorithm," in *Advances in Engineering Computational Technology*, Edinburgh, UK, Civil-Comp Press, 1998, pp. 135-141.

[4] C. V. Camp and S. :. G. Pezeshk, "Optimized Design of Two-Dimensional Structures Using A Genetic Algorithm," *Journal of Structural Engineering, ASCE,* vol. 124, no. 5, pp. 551-559, 1998.

[5] S. Pezeshk and C. C. V., "State of the Art on the Use of Genetic Algorithms in Design of Steel Structures," in *Recent Advances in Optimal Structural Design*, S. Burns (Editor), American Society of Civil Engineers, 2002.

[6] F. Erbatur and M. M. Al-Hussainy, "Optimum design of frames," *Comp. and Struct.,* vol. 5/6, no. 45, pp. 887-891, 1992.

[7] D. A. Shook and e. al., "GA-optimized fuzzy logic control of a large scale building for seismic loads," *Engineering Structures,* vol. 30, pp. 436-449, 2008.

[8] *Eurocode 3: EN 1993-1-1 Design of steel structures – Part 1-1: general rules and rules for buildings.*

[9] K. R., A. T. and D. J. K., "Evolutionary computation and structural design: A survey of the state-of-the-art," *Comp. and Struct.,* vol. 83, no. 23-24, pp. 1943-1978, 2005.