Advanced Engineering Letters **Vol.1, No.4, 142-147 (2022)** **e-ISSN: 2812-9709**

COMPARING TRUSS SIZING AND SHAPE OPTIMIZATION EFFECTS FOR 17 BAR TRUSS PROBLEM

UDC:624.041
Original scientific paper https://doi.org/10.46793/adeletters.2022.1.4.4

Nenad Petrović[1*], Nenad Kostic[1], Nenad Marjanovic[1], Anja Velemir[1], Ljubica Spasojević[1]

[1]University of Kragujevac, Faculty of Engineering, Kragujevac, Serbia

Abstract:
This article aims to demonstrate the difference in results for minimal weight optimization for a 17 bar truss sizing and shape optimization problem. In order to attain results which can be produced in practice Euler bucking, minimal element length, maximal stress and maximal displacement constraints were used. Using the same initial setup, optimization was conducted using particle swarm optimization algorithm and compared to genetic algorithm. Optimal results for both algorithms are compared to initial values which are analytically calculated. The individual element lengths are observed, along with the overall weight, surface area and included number of different cross-sections.

ARTICLE HISTORY
Received: 04.07.2022.
Accepted: 12.10.2022.
Available: 31.12.2022.

KEYWORDS
Truss optimization, sizing and shape, minimal weight, element length

1. INTRODUCTION

Truss structural optimization in general considers the optimization of sizing, shape and topology of a structure. The implementation of all of these aspects is not always possible, or even in some cases favorable, so combinations of two are most frequently employed. Previous research [1] shows the difference in results comparing individual aspect optimization and combinations of optimization aspects. Aside from a complete simultaneous sizing, shape, and topology optimization, the next most favorable type in terms of mass decrease is sizing and shape combined. The decrease in complexity of this approach is also an important factor, since the search-space becomes drastically more discretized when including topological optimization, making a global optimum even harder to achieve. A balance of input effort and achieved effects is grater using this combination.

The bulk of research in the truss optimization field is generally based on applying novel optimization algorithms to solve standardized problems [2-7]. This type of research provides valuable data on parameter settings for truss problems and shows improvements in convergence and resulting weights.

In order for the optimization process to result in trusses which can be implemented in practice, realistic variables and constraints. For sizing optimization discrete cross-section sets must be used to give results representative of available stock in a given material. This is best described in papers [1,8,9] where authors showed how continuous sizing variables lead to unusable solutions. Realistic and definable constraints are just as important, and they allow for the design of trusses which can hold up in practice. These include, other than the typical stress and displacement constraints, Euler bucking constraints, minimal element length and cardinality constraints [10-13]. The use of these constraints in research are more and more frequent and are becoming a norm in truss optimization.

2. MATERIAL AND METHOD

This research is aimed at presenting the improvements gained using sizing and shape optimization simultaneously on a 17 bar truss

*CONTACT: Nenad Petrović, e-mail: npetrovic@kg.ac.rs

example. The proposed optimization algorithms used are genetic algorithm and particle swarm optimization. These optimization algorithms are not novel, however they provide comparable results to newer methods, and their greatest advantage is their availability and ease of use. The results will be compared not just in terms of goal function, but also the other important aspects of total length, outer area and used number of different cross-sections.

For the purposes of this research an original software was developed in Rhino 6 using Grasshopper's Karamba 3D and Silvereye plugins which use genetic algorithm and particle swarm optimization operators.

3. OPTIMIZATION

Structural optimization problems mostly have the goal function of finding the minimal weight, which is also the case in this research. In order to validate only practically applicable results a variety of constraints must be introduced. The static constraints used are minimal and maximal allowed stress and minimal length of individual elements and point displacement while buckling constraints is a dynamic constraint. The goal function for minimal weight is given as follows (Eg. 1):

$$\begin{cases} minW\left(l,d\right)=\sum_{i=1}^{i=n}\rho\frac{d_i^{\,2}\pi}{2}l_i \\ \text{subjected to}\begin{cases}\sigma_{min}\le\sigma_i\le\sigma_{max} & \text{for } i=1,\dots,n \\ u_{min}\le u_j\le u_{max} & \text{for } j=1,\dots,k \\ \left|F_{Ai}^{comp}\right|\le F_{Ki} & \text{for } i=1,\dots,n \\ l_i\ge 1m & \text{for } i=1,\dots,n\end{cases}\end{cases} \quad (1)$$

where the number of truss elements is n, the length of the i^{th} element is l_i, the number of nodes is k, the cross-section diameter is d, the i^{th} element stress is σ_i and j^{th} node displacement is u_j. Axial compression force is F_{Aicomp} and Euler's critical load of the i^{th} element is F_{Ki}, Eg. 2:

$$F_{Ki}=\frac{\pi^2\cdot E\cdot I_i}{l_i^2} \quad (2)$$

E is the modulus of elasticity for the defined material, and the minimum area moment of inertia of the i^{th} element cross-section which changes in each iteration with the change of element diameter is I_i. The change in element length with the change in node position also changes in each iteration. The complexity of finding a global maximum is therefore very complex as the search-space has a varying, or dynamic, constraint.

In addition to these constraints, the minimal element length constraint is implemented due to the possibility of a global extreme value having a small length which could not be produced. The value assigned to this constraint is taken from engineer experience or design guidelines given in literature or corresponding standards [1]. This constraint is given as (Eg. 3):

$$\begin{aligned} l_i &\ge l_{\min} \quad for \quad i=1,\ \dots,\ n \\ l_i &=\sqrt{\left(x_b^i-x_a^i\right)^2+\left(y_b^i-y_a^i\right)^2}\end{aligned}\ . \quad (3)$$

The i^{th} element length l_i is between nodes a (x_{ia}, y_{ia}) and b (x_{ib}, y_{ib}), in that order.

3.1 PSO Algorithm

Particle swarm optimization (PSO) is a swarm intelligent-based algorithm which searches the entirety of the acceptable domain. This gives it an advantage in that it uses only one phase, which effects the algorithm's performance and controllability. Due to its outstanding characteristics PSO has been used in many fields to solve complex problems.

The key operating principle is centered on particle acceleration, the distance from a particle position to the best value of a particle and its position from the globally best particle. Potential solutions are the positions of particles in a given moment. Only the best position is accepted and passed through an iterative process of optimization. Every new result is defined by two components, velocity, v_i, and position, x_i. The number of positions and accelerations is n depending on the total defined number of particles. Every new value is derived as follows (Eg. 4):

$$x_{new,i}=x_{old,i}+\upsilon_{new,i} \quad (4)$$

where $i=1,2,\dots,N$, is the total size of the population (Eg. 5):

$$\upsilon_{new,i}=\omega\cdot\upsilon_{old,i}+c_p\cdot r_p\left(x_{p,i}-x_{x,i}\right)+c_g\cdot r_g\left(x_{g,i}-x_{x,i}\right) \quad (5)$$

Constants c_p and c_g are both suggested in literature as 1.5. Random values r_p and r_g are from the interval between 0 and 1. Current particle position is defined as $x_{x,i}$. Particle intensity is ω (inertia weight) and is defined as (Eg. 6):

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iteration_{max}} \cdot Iteration \tag{6}$$

Values ω_{max}=0.9 and ω_{min}=0.4 are defined in literature.

3.2 Genetic algorithm

Genetic algorithm, or GA, is a heuristic method for optimizing whose operation is based on mimicking natural processes [14]. The algorithm contains three basic operators: selection, crossover, and mutation which are illustrated in Fig. 1.

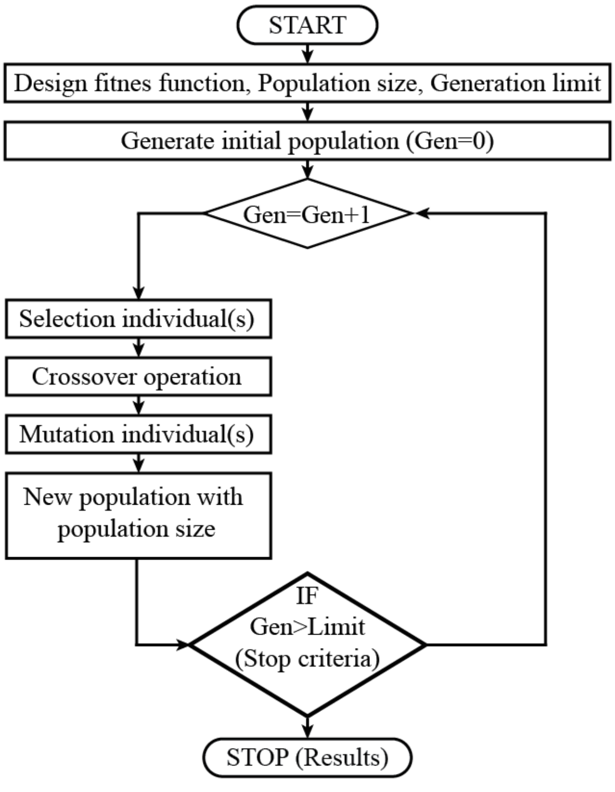

Fig. 1. Genetic algorithm

Selection is the process of transferring genetic information through generations, while crossover represents the operations between two parents, where an exchange of genetic information and new generations are made. A mutation operator creates a random change in the genetic composition for some individuals for overcoming early convergence.

Algorithm operation is founded on survival of the fittest through evolution by exchanging genetic material. Selection ranks individuals using values from the fitness function that defines the quality of an individual.

Both these algorithms have been used in research over the years. Their application spans many different fields and they have been used successfully to find optimal solutions for complex problems with few known inputs [15].

4. TEST EXAMPLE

The initial truss model bar and node layout for the 17 bar truss example is given in Fig. 2. This is a commonly used example. The material characteristics for all bars are: Young modulus 206842.719 MPa and density of 7.4 g/cm^3. There is only one point load applied in node 9 which is 444.82 kN, in the - y direction.

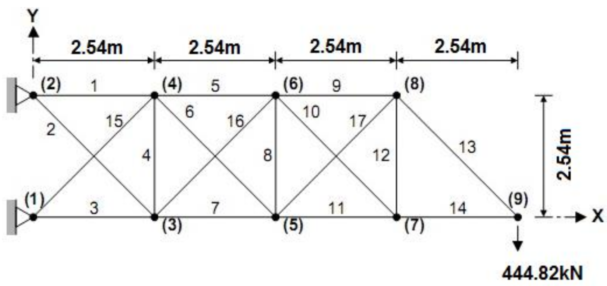

Fig. 2. The 17 bar truss problem [12]

Each bar cross-section is an independent variable. There is only one fixed constraint for displacement on all nodes of ±0.0508 m in any direction. Cross-section variables are taken from the same discrete set as in [12].

5. RESULTS AND DISCUSSIONS

The initial model was first analytically calculated and it was determined that the cross-section for all elements should be a diameter of 105 mm (86.590 cm^2), resulting in a structure weighing 3181.777 kg. This result matches the result in [12] and will be used as a benchmark to measure the improvement made by using sizing and shape optimization.

The optimal solution achieved using GA used a population size of 100 with a 2x initial boost, 45% inbreeding and maintaining 5% from previous generations. The optimal solution using PSO used 100 iterations with a max velocity of 0.200 and a swarm size of 20. Both algorithms used the same initial setup with the layout from Fig. 2 and diameters for all bars from the analytical calculation. All results were achieved using an original software solution developed for Rhino 6 using Grasshopper's Karamba 3D and Silvereye plugins.

Table 1 gives the optimal cross-sections for each bar including the total weight, length and outer area for both GA and PSO.

Table 1. Optimal results

Element	Cross-section areas (cm^2)		
	Analytical	GA	PSO
1	86.590	70.882	38.485
2	86.590	23.758	50.265
3	86.590	86.590	122.718
4	86.590	7.069	3.801
5	86.590	33.183	63.617
6	86.590	28.274	4.524
7	86.590	70.882	78.540
8	86.590	9.621	44.179
9	86.590	63.617	4.524
10	86.590	2.545	63.617
11	86.590	50.265	56.745
12	86.590	11.341	9.621
13	86.590	33.183	44.179
14	86.590	63.617	23.758
15	86.590	56.745	0.283
16	86.590	23.758	38.485
17	86.590	63.617	44.179
Weight (kg)	3181.777	1456.573	1455.92
Total length (m)	50.545	48.145	49.401
Total area (m^2)	4.377	0.200	0.200

Table 2 gives the optimal coordinates of nodes for each solution.

Table 2. Optimal node positions

Node	Coordinates [m]			
	GA		PSO	
	x	y	x	y
1	0	0	0	0
2	0	2.54	0	2.54
3	2.694	-0.122	3.304	-0.275
4	2.749	2.41	2.451	2.343
5	5.124	0.136	5.497	0.425
6	5.228	2.389	6.275	2.426
7	7.52	0.458	7.883	2.12
8	7.992	2.21	7.883	-0.051
9	10.16	1.118	10.16	0.513

Fig. 3 shows the differences in element length between the analytical, GA and PSO solutions.

For both optimization algorithms the process was repeated ten times each, and the best result was used from each algorithm. It should also be noted that the resulting optimal solutions did not vary greatly between each of the 10 solutions. Fig. 4 and 5 show the resulting shapes for GA and PSO respectively.

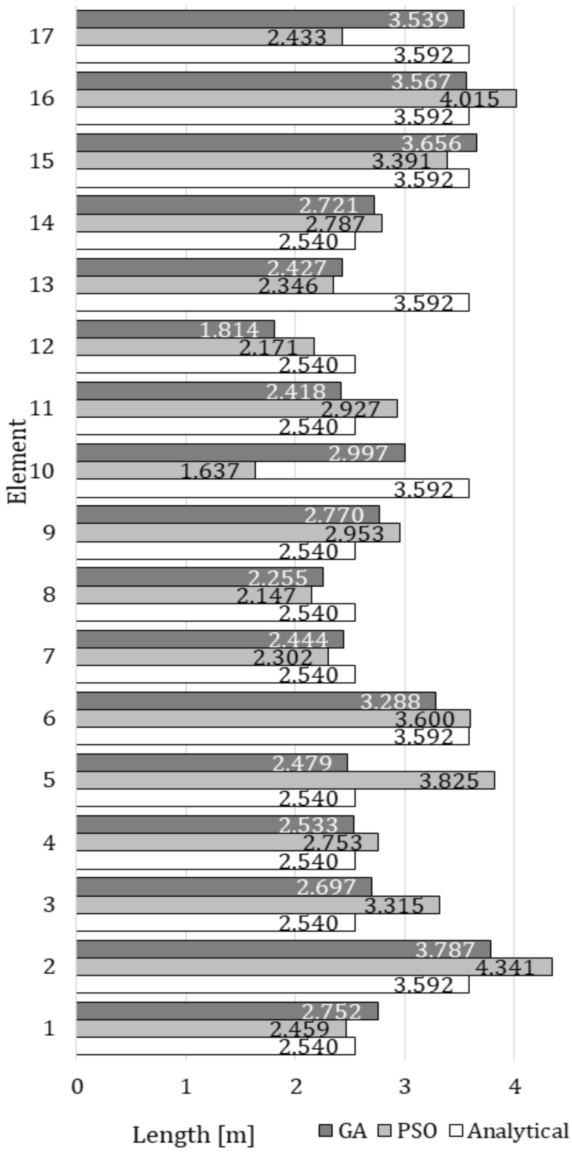

Fig. 3. Individual element lengths for all solutions

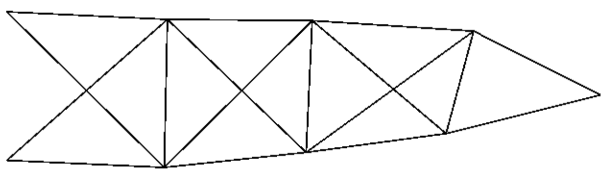

Fig. 4. GA optimized shape

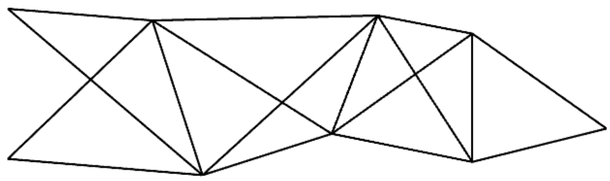

Fig. 5. PSO optimized shape

Both the GA and PSO optimal structures use 12 different cross-sections on 17 bars. This is an unrealistically high number. The use of a cardinality constraint, such as the one used in [12,13] is warranted.

6. CONCLUSION

With the increase in computer memory and processing power, the increase of complexity of possible optimization problems has increased as well. Truss structural optimization has followed this trend with the increase in the number of variables, constraints and possible iterations aiming to finding the best possible solutions. Implementing realistic constraints which mimic analytical calculations is important in order to have this technology used successfully and widely in practice.

The research presented in this paper shows the implementation of such constraints on a typical sizing and shape optimization problem. The minimization of weight is in ~54% for both optimization algorithms showing that it is possible to half the weight of such a structure through the use of this technology. Overall element length is decreased by 2.3% and 4.7% using GA and PSO, respectively. The greatest savings aside from weight can be seen in outer surface area where there is a ~95%. The small decrease in length is a negligible but necessary step as the new shape allows for a better stress distribution and the use of much thinner profiles in most of the structure. The GA optimized model only has one element the same thickness as the analytically calculated model, which is subjected to compression and sized according to limiting bucking load. The PSO model has the same element with a slightly larger profile for that specific bar, allowing for other bars to be sized accordingly and achieving similar results. Further research in this field should see a greater implementation of cardinality constraints in order to limit the impractical use of a large number of different cross-sections.

The different shapes, but similar weights and areas achieved using two different optimization methods shows that in a large search-space there are many results which are applicable and have a drastically decreased weight than the analytically calculated model.

REFERENCES

[1] N. Petrović, N. Kostić, N. Marjanović, V. Marjanović, Influence of Using Discrete Cross-Section Variables for All Types of Truss Structural Optimization with Dynamic Constraints for Buckling. *Applied Engineering Letters*, 3(2), 2018: 78-83. https://doi.org/10.18485/aeletters.2018.3.2.5

[2] H. Cao, W. Sun, Y. Chen, F. Kong, L. Feng, Sizing and Shape Optimization of Truss Employing a Hybrid Constraint-Handling Technique and Manta Ray Foraging Optimization. *Expert Systems with Applications*, 213(Part B), 2023: 118999. https://doi.org/10.1016/j.eswa.2022.118999

[3] S. Anosri, N. Panagant, S. Bureerat, N. Pholdee, Success History Based Adaptive Multi-Objective Differential Evolution Variants with an Interval Scheme for Solving Simultaneous Topology, Shape and Sizing Truss Reliability Optimization. *Knowledge-Based Systems*, 253, 2022: 109533. https://doi.org/10.1016/j.knosys.2022.109533

[4] S. Zheng, L. Qiu, F. Lan, TSO-GCN: A Graph Convolutional Network Approach for Real-Time and Generalizable Truss Structural Optimization. *Applied Soft Computing*, 134, 2023: 110015. https://doi.org/10.1016/j.asoc.2023.110015

[5] M.S. Gonçalves, R.H. Lopez, L.F.F. Miguel, Search Group Algorithm: A New Metaheuristic Method for the Optimization of Truss Structures. *Computers & Structures*, 153, 2015: 165-184. https://doi.org/10.1016/j.compstruc.2015.03.003

[6] H.-A. Pham, T.-D. Tran, Optimal Truss Sizing by Modified Rao Algorithm Combined with Feasible Boundary Search Method. *Expert Systems with Applications*, 191, 2022: 116337. https://doi.org/10.1016/j.eswa.2021.116337

[7] M. Shahabsafa, A. Mohammad-Nezhad, T. Terlaky, L. Zuluaga, S. He, J.T. Hwang, J.R.R.A. Martins, A Novel Approach to Discrete Truss Design Problems Using Mixed Integer Neighborhood Search. *Structural and Multidisciplinary Optimization*, 58, 2018: 2411-2429. https://doi.org/10.1007/s00158-018-2099-8

[8] G. Bekdaş, S.M. Nigdeli, X.-S. Yang, Sizing optimization of truss structures using flower pollination algorithm. *Applied Soft Computing*, 37, 2015: 322-331.

https://doi.org/10.1016/j.asoc.2015.08.037

[9] V. Ho-Huu, D. Duong-Gia, T. Vo-Duy, T. Le-Duc, T. Nguyen-Thoi, An Efficient Combination of Multi-Objective Evolutionary Optimization and Reliability Analysis for Reliability-Based Design Optimization of Truss Structures. *Expert Systems with Applications*, 102, 2018: 262-272. https://doi.org/10.1016/j.eswa.2018.02.040

[10] S. Gholizadeh, Layout Optimization of Truss Structures by Hybridizing Cellular Automata and Particle Swarm Optimization. *Computers & Structures*, 125, 2013: 86-99. https://doi.org/10.1016/j.compstruc.2013.04.024

[11] H. Madah, O. Amir, Truss Optimization with Buckling Considerations Using Geometrically Nonlinear Beam Modeling. *Computers & Structures*, 192, 2017: 233-247. https://doi.org/10.1016/j.compstruc.2017.07.023

[12] N. Petrović, V. Marjanović, N. Kostić, N. Marjanović, Means and Effects of Constraining the Number of Used Cross-Sections in Truss Sizing Optimization. *Transactions of FAMENA*, 44(3), 2020: 35-46. https://doi.org/10.21278/TOF.44303

[13] J.P.G. Carvalho, A.C.C. Lemonge, É.C.R. Carvalho, P.H. Hallak, H.S. Bernardino, Truss Optimization with Multiple Frequency Constraints and Automatic Member Grouping. *Structural and Multidisciplinary Optimization*, 57, 2017: 547-577. https://doi.org/10.1007/s00158-017-1761-x

[14] E.G. Shopova, N.G. Vaklieva-Bancheva, BASIC-A Genetic Algorithm for Engineering Problems Solution. *Computers & Chemical Engineering*, 30(8), 2006: 1293-1309. https://doi.org/10.1016/j.compchemeng.2006.03.003

[15] B. Stojanović, S. Gajević, N. Kostić, S. Miladinović, A. Vencl, Optimization of Parameters that Affect Wear of A356/Al2O3 Nanocomposites Using RSM, ANN, GA and PSO Methods. *Industrial Lubrication and Tribology*, 74(3), 2022: 350-359. https://doi.org/10.1108/ILT-07-2021-0262