CAMES, **27**(2–3): 177–184, 2020, doi: 10.24423/cames.282

Size and Shape Design Optimization of Truss Structures Using the Jaya Algorithm

Maksym GRZYWIŃSKI

Czestochowa University of Technology
Faculty of Civil Engineering
Akademicka 3, Częstochowa, Poland
e-mail: maksym.grzywinski@pcz.pl

The metaheuristic algorithm is proposed to solve the weight minimization problem of truss structures, considering the shape and sizing design variables. Design variables are discrete and/or continuous. The design of truss structures is optimized by an efficient optimization algorithm called Jaya. The main feature of Jaya is that it does not require setting algorithm-specific parameters. The algorithm has a very simple formulation in which the basic idea is to approach the best solution and escape from the worst solution [6]. Analyses of structures are performed by a finite element code in MATLAB. The effectiveness of the Jaya algorithm is demonstrated using two benchmark examples: planar truss 18-bar and spatial truss 39-bar, and compared with results in references.

Keywords: planar and spatial truss, size and shape optimization, discrete and continuous variables, Jaya algorithm.

1. Introduction

The structural optimal design has always been of interest for engineers in practice. The attention is not only paid to the cost of construction but also to the geometry of structures. Engineers are responsible for designing structures with high reliability and low cost [1]. For these purposes, many optimal algorithms were investigated to accomplish the tasks including the classical methods and innovative algorithms.

Metaheuristic techniques have been developed to solve structural optimization problems. Genetic algorithms (GA), particle swarm optimization (PSO), harmony search (HS), teaching-learning-based optimization (TLBO), and firefly algorithm (FA) are all well-established methods for the optimal design of structures. Depending on the optimization purpose, cross-sectional areas of the members and/or nodal coordinates separately or simultaneously can be included as the design variables of the problem.

An interesting metaheuristic algorithm that has a very simple formulation and does not require internal parameters is the Jaya algorithm (JA) developed in [8]. JA was also used for the optimum design of steel grillage by Dede [3]. Degertekin *et al.* [4] presented a study on the sizing, layout and topology design optimization of truss structures using the Jaya algorithm. Grzywiński *et al.* [5] optimized the braced dome structures with natural frequency constraints.

In [6], the shape and size optimization of trusses with dynamic constraints was presented using the Jaya algorithm.

The word "Jaya" means "victory" in Sanskrit. This population-based algorithm is based on the concept that the search process should always move toward the best design and avoid the worst design. The search engine continuously tries to get closer to success (i.e., to reach the best design) trying at the same time to avoid failure (i.e., by moving away from the worst design). JA does not include any algorithm-specific parameter when compared to other metaheuristic optimization algorithms.

In fact, JA only requires two standard control parameters such as population size (i.e., the number of truss designs in the population) and a maximum number of iterations. In the optimization process, ndv is the number of design variables (i.e., the number of member groups ng in sizing optimization problems, the summation of the number of member groups and the number of shape variables in sizing-shape optimization problems), and np is the population size (i.e., the number of truss designs).

The design corresponding to the lowest penalized objective function F_p^{best} is stored as the best design while the design corresponding to the highest penalized objective function F_p^{worst} is the worst design stored in the population.

Let $X_{k,l,i}$ denote the value of the k-th design variable (cross-sectional areas A and nodal coordinates X) for the l-th design of the population at the beginning of the i-th iteration. Jaya algorithm perturbs this design variable using the following equation [4, 6]:

$$X_{k,l,i}^{\text{new}} = X_{k,l,i} + r_{1,k,i}(X_{k,\text{best},i} - |X_{k,l,i}|) - r_{2,k,i}(X_{k,\text{worst},i} - |X_{k,l,i}|), \qquad (1)$$

where $X_{k,l,i}^{\text{new}}$ is the new value assigned to the design variable $X_{k,l,i}$, $r_{1,k,i}$ and $r_{2,k,i}$ are two randomly generated real numbers in the [0,1] range for the k-th design variable in the i-th iteration. $X_{k,\text{best},i}$ is the value of the k-th design variable for the best design of the population at the i-th iteration while $X_{k,\text{worst},i}$ is the value of the k-th design variable for the worst design stored in the population.

The term $r_{1,k,i}(X_{k,\text{best},i} - |X_{k,l,i}|)$ indicates the tendency of the solution to move closer to the best solution. The term $r_{2,k,i}(X_{k,\text{worst},i} - |X_{k,l,i}|)$ indicates the tendency of the solution to avoid the worst solution.

2. Optimization of truss structures

One of the most important factors in structural design is the total structural weight. There are two types of design variables in this case: cross-sectional areas of members and nodal coordinates. For this aim, the objective function for the truss structures is defined as:

$$\text{minimize} \quad W(X) = \sum_{k=1}^{ng} A_k \sum_{i=1}^{mk} \rho_i \cdot L_i(x_i), \tag{2}$$

where $W(X)$ is the weight of the truss, A_k (bar cross-sectional areas), and x_i (nodal coordinates) are the design variables, respectively; ρ_i and L_i is the material density and the length of the i-th element, respectively; ng is the total number of groups, mk is the total number of groups in group k, and nn is the total number of nodes.

The constraints imposed on the structure are:

• member stress:

$$\sigma_k^c \leq \sigma_k \leq \sigma_k^t, \qquad k = 1, 2, ..., ng, \tag{3}$$

where σ_k represents the stress for the k-th group elements, σ_k^t is the allowable tensile stress, and σ_k^c is the allowable compressive stress, respectively;

• Euler buckling stress:

$$\sigma_k \leq \sigma_k^b, \qquad k = 1, 2, ..., ng, \tag{4}$$

where σ_k^b is the Euler buckling compressive stress limit for the k-th group elements; it is usually taken as:

$$\sigma_k^b = \frac{K \cdot E \cdot A_k}{L_k^2}, \qquad k = 1, 2, ..., ng, \tag{5}$$

where K is a constant determined from the cross-sectional geometry (in this case $K = 4$), and E is Young's modulus of the material;

• nodal displacement:

$$|d_i| \leq d_{\max}, \qquad i = 1, 2, ..., nn. \tag{6}$$

The prescribed limit for the nodal point of the structure does not violate the displacement constraints;

• limit of design variables:

$$A_{\min} \leq A_k \leq A_{\max}, \qquad k = 1, 2, ..., ng, \tag{7}$$

$$x_{\min} \leq x_i \leq x_{\max}, \qquad i = 1, 2, ..., nn. \tag{8}$$

Inequalities (7) and (8) indicate that the design variables including either shape and/or sizing variable must take a value between the minimum and maximum bounds,

$$c_i = \frac{|d_i|}{d_{\max}}, \qquad c_k = \frac{|\sigma_k|}{\sigma_{\max}}, \tag{9}$$

where c_i and c_k are the values of each constraint.

The objective function must be changed as to include constraints. For this aim, a penalty function calculating the value of violation of constraints is determined. By means of this function, the objective function is changed to a function including constraints.

The penalty function is given as:

$$C = \sum_{i=1}^{nn} c_i + \sum_{k=1}^{ng} c_k. \tag{10}$$

The objective function is changed to the penalized objective function by adding a penalty function to it. The penalized objective function $F(X)$ can be given as:

$$F(X) = W(X)[1 + P \cdot C], \tag{11}$$

where P is a positive constant, which is a variable for each problem. This constant can be determined by the user to take into account the constraints. At the end of the optimization process, the total constraints must be zero. Then, the penalized objective function can be equal to the total weight of the structure. That is, the algorithm tries to find the best solution without violating the constraints.

3. Numerical examples

3.1. Planar truss 18-bar

The first example is the 18-bar planar truss shown in Fig. 1a. Properties of applied material are shown in Table 1. Moving nodes coordinates and grouping of the elements are presented in Table 2. This truss structure was previously presented in [7] and [1].

The lower bound and the upper bound for the cross-sectional areas are 2.00 in^2 and 21.75 in^2 and the interval is 0.25 in^2.

For the design optimization, the cross-sectional areas were categorized into four groups for size optimization and eight nodal coordinates were selected for the shape optimization. The design variables are discrete for the cross-sectional areas while they are continuous for the nodal coordinates. Optimal shape after the optimization with JA is shown in Fig. 1b. The comparison of results with

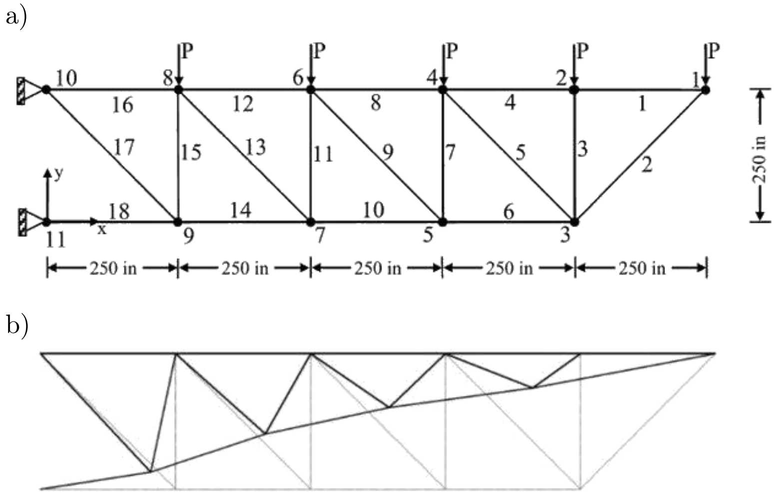

FIG. 1. The geometry of the 18-bar truss: a) initial shape, b) optimized shape.

TABLE 1. Structural constraints and material properties.

Properties/constraints	Unit	Value/notes
Modulus of elasticity	E [ksi]	10 000
Material density	ρ [lb/in^3]	0.1
Nodal forces	P [kips]	20
Displacement constraints	δ [in]	± 10 for each direction
Stress constraints	σ [ksi]	20 for tension −20 for compression

TABLE 2. Initial shape and member grouping for the 18-bar truss problem.

Shape variables		Size variables	
node x [in]	node y [in]	cross-area	element
$775 \leq x_3 \leq 1225$	$-225 \leq y_3 \leq 245$	A1	1, 4, 8, 12, 16
$525 \leq x_5 \leq 975$	$-225 \leq y_5 \leq 245$	A2	2, 6, 10, 14, 18
$275 \leq x_7 \leq 725$	$-225 \leq y_7 \leq 245$	A3	3, 7, 11, 15
$25 \leq x_9 \leq 475$	$-225 \leq y_9 \leq 245$	A4	5, 9, 13, 17

those of the other references is given in Table 3. The number of populations in [1, 7], and this study is 50. The maximum generation numbers are 100, 500, and 4500 for [7, 1], and this study, respectively. The best results are obtained in [1]. Due to the fact that the process is a random result, subsequent solutions may be different. The number of generations can be increased to find a better solution, but this also does not guarantee obtaining the best solution. The cost of the calculations was very high. The CPU calculation time for one run was 1021.4688 sec.

Table 3. Optimal size and shape for the 18-bar truss.

Design variables	Kaveh & Kalatjari [7]	Cheng *et al.* [1]	This study
A1	12.25	12.50	12.50
A2	18.00	18.00	18.00
A3	5.25	5.25	5.25
A4	4.25	3.75	3.75
X3	913.0	914.524	915.1937
Y3	186.8	188.793	188.4463
X5	650.0	647.351	647.6893
Y5	150.5	149.683	148.9354
X7	418.8	416.831	416.6843
Y7	97.4	101.332	100.6179
X9	204.8	204.165	203.8285
Y9	26.7	31.662	31.3023
Weight [lb]	4547.9	4526.708	4527.850

3.2. Spatial truss 39-bar

The second test problem is the combined sizing and shape optimization of the 39-bar spatial truss tower shown in Fig. 2a. Problem specifications are listed in Table 4. Fixed nodes coordinates and elements connectivity are presented in Table 5. The top and bottom nodes have fixed positions, while the middle nodes' coordinates are taken as design variables.

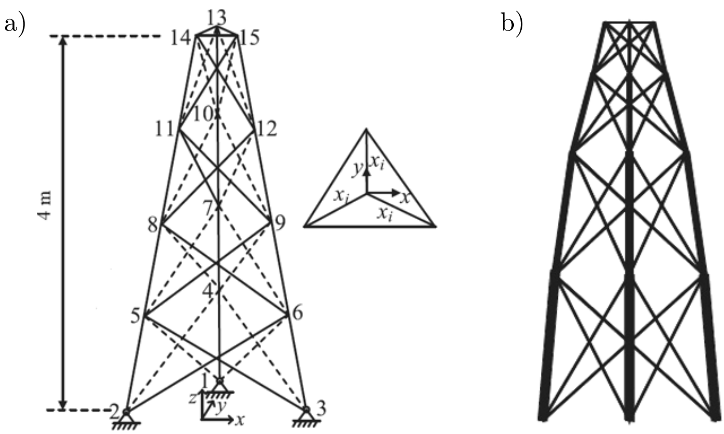

Fig. 2. The geometry of the 39-bar truss: a) initial shape, b) optimized shape.

The symmetry of the structure is maintained during the optimization process. The population size and the allowable number of iterations are set to 40 and 100, respectively.

TABLE 4. Input data for the spatial 39-bar truss problem.

Properties/constraints	Unit	Value/notes
Modulus of elasticity	E [GPa]	210
Material density	ρ [kg/m^3]	7800
Stress constraints	σ [MPa]	240 for tension −240 for compression
Displacement constraints	δ [cm]	0.4 for Y directions (nodes 13–15)
Nodal forces	F [kN]	± 10 for Y directions (nodes 13–15)
Euler buckling	σ_e [MPa]	$\sigma_e \leq \dfrac{-K_e E A_e}{L_e^2}$

TABLE 5. Initial shape and member grouping for the spatial 39-bar truss problem.

Shape variables				Size variables	
joint	x [m]	y [m]	z [m]	cross-area	node-node
1	0.000	1.000	0.000	A1	(1–4), (2–5), (3–6)
2	−0.866	−0.500	0.000	A2	(4–7), (5–8), (6–9)
3	0.866	−0.500	0.000	A3	(7–10), (8–11), (9–12)
13	0.000	0.280	4.000	A4	(10–13), (11–14), (12–15)
14	−0.242	−0.140	4.000	A5	the remaining elements
15	0.242	−0.140	4.000		

The optimum design found by JA is demonstrated in Fig. 2b, while in Table 6 it is compared with those obtained from other methods. The solution obtained by JA is better than the cited references [2] and [9]. The fully stressed design (FSD) algorithm and teaching-learning-based optimization (TLBO) is used by

TABLE 6. Optimal size and shape for the 39-bar truss.

Design variables	Wang *et al.* [9] (FSD)	Dede & Ayvaz [2] (TLBO)	This study (Jaya)
A1	11.01	11.9650	12.0000
A2	8.63	11.1457	10.1794
A3	6.69	7.8762	6.5537
A4	4.11	2.7013	2.1396
A5	4.37	2.4058	1.7422
Y4	0.805	0.8996	0.8926
Z4	1.186	1.3507	1.1098
Y7	0.654	0.6917	0.6514
Z7	2.204	2.3122	2.5000
Y10	0.466	0.4825	0.4115
Z10	3.092	3.3031	3.4962
Weight [kg]	203.18	154.13	134.62

the references [2] and [9], respectively. This optimum design is obtained at the 100th generation (i.e., after 4000 structural analyses).

4. Conclusions

The Jaya algorithm showed good performance when searching the minimum weight of the truss system. It did not require control parameters as in other optimization techniques. The design results were compared with the results given in the literature. This comparison clearly shows that the proposed Jaya algorithm can be effectively used in the design of truss structures. To optimize the truss structures, a new and efficient Jaya algorithm was coded in the Matlab.

References

1. W. Cheng, F. Liu, L.J. Li, Size and geometry optimization of trusses using teaching-learning-based optimization, *International Journal of Optimization in Civil Engineering*, **3**(3): 431–444, 2013.
2. T. Dede, Y. Ayvaz, Combined size and shape optimization of structures with a new meta-heuristic algorithm, *Applied Soft Computing*, **28**: 250–258, 2015, doi: 10.1016/j.asoc.2014.12.007.
3. T. Dede, Jaya algorithm to solve single objective size optimization problem for steel grillage structures, *Steel and Composite Structures*, **26**(2): 163–170, 2018, doi: 10.12989/scs.2018.26.2.163.
4. S.O. Degertekin, L. Lamberti, I.B. Ugur, Sizing, layout and topology design optimization of truss structures using the Jaya algorithm, *Applied Soft Computing*, **70**: 903–928, 2018, doi: 10.1016/j.asoc.2017.10.001.
5. M. Grzywiński, T. Dede, Y.I. Özdemir, Optimization of the braced dome structures by using Jaya algorithm with frequency constraints, *Steel and Composite Structures*, **30**(1): 47–55, 2019, doi: 10.12989/scs.2019.30.1.047.
6. M. Grzywiński, J. Selejdak, T. Dede, Shape and size optimization of trusses with dynamic constraints using a metaheuristic algorithm, *Steel and Composite Structures*, **33**(5): 747–753, 2019, doi: 10.12989/scs.2019.33.5.747.
7. A. Kaveh, V. Kalatjari, Size/geometry optimization of trusses by the force method and genetic algorithm. *Journal of Applied Mathematics and Mechanics*, **84**(5): 347–357, 2004, doi: 10.1002/zamm.200310106.
8. R.V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *International Journal of Industrial Engineering Computations*, **7**(1): 19–34, 2016, doi: 10.5267/j.ijiec.2015.8.004.
9. D. Wang, W.H. Zhang, J.S. Jiang, Combined shape and sizing optimization of truss structures, *Computational Mechanics*, **29**(4–5): 307–312, 2002, doi: 10.1007/s00466-002-0343-x.

Received January 31, 2020; revised version April 15, 2020.