Applied Soft Computing xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Full Length Article

Sizing, layout and topology design optimization of truss structures using the Jaya algorithm

S.O. Degertekin[a,*], L. Lamberti[b], I.B. Ugur[c]

[a] Civil Engineering Department, Dicle University, 21280, Diyarbakir, Turkey
[b] Dipartimento Meccanica, Matematica e Management, Politecnico di Bari, 70126, Bari, Italy
[c] Civil Engineering Department, Sirnak University, 73000, Sirnak, Turkey

ARTICLE INFO

Article history:
Received 19 April 2017
Received in revised form 6 September 2017
Accepted 5 October 2017
Available online xxx

Keywords:
Metaheuristic optimization methods
Jaya algorithm
Sizing, layout and topology optimization
Truss structures

ABSTRACT

A very recently developed metaheuristic method called Jaya algorithm (JA) is implemented in this study for sizing and layout optimization of truss structures. The main feature of JA is that it does not require setting algorithm-specific parameters. The algorithm has a very simple formulation where the basic idea is to approach the best solution and escape from the worst solution. The original JA formulation is modified in this research in order to improve convergence speed and reduce the number of structural analyses required in the optimization process. The suitability of JA for truss optimization is investigated by solving six classical weight minimization problems of truss structures including sizing, layout and large-scale optimization problems with up to 204 design variables. Discrete sizing/layout variables and simplified topology optimization also are considered. The test problems solved in this study are very common benchmarks in structural optimization and practically describe all scenarios that may be faced by designers. The results demonstrate that JA can obtain better designs than those of the other state-of-the-art metaheuristic and gradient-based optimization methods in terms of optimized weight, standard deviation and number of structural analyses.

1. Introduction

Metaheuristic algorithms try to achieve a dynamic balance between exploration of search space (i.e. “diversification” phase) and exploitation of accumulated search experience (i.e. “intensification” phase). This balance allows to quickly identify regions of design space containing high quality solutions as well as to bypass regions that either were already explored or are far from global optimum [1–3].

Metaheuristic optimization methods such as, for example, genetic algorithms (GA) [4,5], simulated annealing (SA) [6,7], evolution strategies (ES) [8,9], ant colony optimization (ACO) [10,11], particle swarm optimization (PSO) [12,13], harmony search optimization (HS) [14,15], artificial bee colony algorithm (ABC) [16], big bang-big crunch optimization (BB-BC) [17], charged system search (CSS) [18], firefly algorithm (FFA) [2], teaching-learning-based optimization (TLBO) [19], flower pollination algorithm (FPA) [20], swallow swarm optimization algorithm (SSO) [21] and water evaporation optimization (WEO) [22], have been successfully used in every field of science and engineering.

Truss structures are very often selected as benchmark design problems to test the efficiency of metaheuristic algorithms. Just to cite a few examples from the extensive optimization literature, GA using different search operators and re-analysis strategies [23–30], differential evolution with various search schemes [31–35], SA based on single or multi-level search [36–39], particle swarm optimization with different modelling of social/individual behavior, combination of global/local best and center of mass particles [40–44], harmony search optimization with different search and parameter adaptation strategies [45–49], big bang big crunch big bang-big crunch optimization with different definition of trial designs and infrequent explosions [48,50–52], teaching-learning based optimization [53–56], artificial bee colony algorithm with adaptive penalty approach [57], firefly algorithm [58,59], cultural algorithm [60], flower pollination algorithm [61], water evaporation algorithm [62], hybrid methods combining two or more metaheuristic algorithms, as well as metaheuristic algorithms and gradient-based optimization [63–66]. Further information can be found in classical textbooks [2], review papers [67,68] and studies comparing the relative efficiency of several metaheuristic algorithms in static and dynamic truss optimization problems [69,70].

* Corresponding author.
E-mail address: sozgurd@gmail.com (S.O. Degertekin).

https://doi.org/10.1016/j.asoc.2017.10.001
1568-4946/

The continuously increasing computational power has favored the blooming of new metaheuristic algorithms that are often claimed by authors to be very competitive with the most popular state-of-the-art optimizers. However, finding the global optimum at a reasonably low computational cost for all problems with a limited sensitivity to the selection (i.e. size and composition) of initial population and the setting/adaptation of internal parameters that drive the search process remains an unresolved issue in metaheuristic optimization.

An interesting metaheuristic algorithm that has a very simple formulation and does not require internal parameters is the JAYA algorithm (JA) developed by Rao in 2016 [71]. The JA algorithm was successfully tested on several benchmark functions. Rao and Waghmare [72] later utilized the JA for solving constrained mechanical design problems such as robot gripper, multiple disc clutch brake, hydrostatic thrust bearing and rolling element bearing. The efficiency of the JA with respect to other metaheuristic algorithms was demonstrated also for these test problems. The JA was used also for sizing optimization of a micro-channel heat sink [73] by taking thermal resistance and pumping power as objective functions and micro-channel width, depth and fin width as design variables. Once again JA resulted very competitive or even better than those obtained for TLBO and multi-objective evolutionary algorithms.

The main objective of this study is to evaluate the suitability of the JA algorithm for weight minimization of truss structures. This test suite is selected because of the tremendously large amount of data available in literature, which allows comprehensive and detailed comparisons to be carried out. Sizing optimization problems of trusses with 200, 942 and 1938 elements, and combined sizing-layout optimization problems of trusses with 25, 45 and 47 elements are solved for that purpose. Sizing optimization problems include up to 204 design variables while combined sizing-layout optimization problems include up to 81 design variables. Discrete sizing/layout variables and simplified topology optimization also are considered. The original JA formulation is modified in order to improve convergence speed thus reducing the number of structural analyses required in the optimization.

The results obtained by the JA are compared with those of other state-of-the-art metaheuristic optimization methods including variants of genetic algorithms, differential evolution, simulated annealing, particle swarm, harmony search, big bang big crunch, artificial bee colony, teaching-learning based optimization, cultural algorithm, firefly algorithm, flower pollination algorithm, water evaporation optimization, hybrid particle and swallow swarm optimization, hybrid particle swarm, ant colony and harmony search optimization. Comparisons with gradient-based optimizers also are presented in the article. The performance of JA is evaluated in terms of minimum weight, standard deviation on optimized weight and number of structural analyses required in the optimization process. In all test problems, JA is compared with the best solutions available in metaheuristic optimization literature as well as with commercial software. A statistical analysis of the best, average and worst optimized weights and corresponding standard deviations obtained over independent optimization runs is performed. Results prove that the proposed algorithm is very competitive with the other metaheuristic methods and its convergence speed is similar to gradient-based optimizers.

The paper is structured as follows. The formulation of the design optimization problem for truss structures is recalled in Section 2. The JAYA algorithm is described in Section 3 along with its implementation for truss structure problems. Section 4 describes the test problems and discusses optimization results. Section 5 summarizes the main findings of this study. Sensitivity of JA convergence behavior to population size is analyzed in detail in the Appendix.

2. Optimization of truss structures

The objective of truss optimization is to minimize the weight of the structure under design constraints such as element stresses and nodal displacements. The sizing optimization problem of a truss structure can be stated as:

$$\begin{aligned} &\text{Find} \quad A = [A_1, A_2, \ldots.., A_{ng}] \\ &\text{To minimize} \quad \mathrm{W(A)} = \sum_{\mathrm{i=1}}^{\mathrm{nm}} \gamma_i \mathrm{A_i L_i} \\ &\text{Subjected to} \\ &\sigma_i^c \le \sigma_i \le \sigma_i^t, \quad i = 1, 2, \ldots., nm \\ &\delta_{\min} \le \delta_j \le \delta_{\max}, \quad j = 1, 2, \ldots., nn \\ &A_{\min} \le A_k \le A_{\max}, \quad k = 1, 2, \ldots., ng \end{aligned} \tag{1}$$

where the $\boldsymbol{A}$ vector contains the sizing variables (i.e. cross-sectional areas of bars), $W(\boldsymbol{A})$ is the weight of the truss structure. γ_i and L_i, respectively, are the material density and the length of member i. A_i is the cross-sectional area of member i with the corresponding lower/upper bounds $A_{\min}$ and $A_{\max}$. Each truss design must satisfy design constraints on member stresses σ_i for each element i and displacements δ_j for each node j. σ_i^c and σ_i^t are the allowable compression and tension stresses for member i. $\delta_{\min}$ and $\delta_{\max}$ are the allowable displacements for node j. nm is the number of members in the truss structure, nn is the number of nodes, ng is the number of member groups (i.e. number of design variables).

For optimization problems including also layout variables, the cost function can be rewritten as:

$$\text{Minimize} \quad \mathrm{W}(\boldsymbol{A}, \boldsymbol{X}) = \gamma \sum_{\mathrm{i=1}}^{\mathrm{nm}} A_i \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2 + (z_{i1} - z_{i2})^2} \tag{2}$$

where $x_{\mathrm{i1,2}}$, $y_{\mathrm{i1,2}}$, $z_{\mathrm{i1,2}}$ are the coordinates of the nodes limiting the i^{th} element of the structure. The optimization problem hence includes ndv design variables where ndv is the sum of the ng element cross-sectional areas included as sizing variables and $nlay$ nodal coordinates included as layout variables. In topology optimization, elements can be removed from the structure if their cross-sectional areas become very small (for example, 10^{-7} in^2).

Stress and displacement constraints to be satisfied are handled by using a penalty function. The penalized objective function F_p is obtained as the product between the truss weight $W(\boldsymbol{A},\boldsymbol{X})$ and the penalty function ψ_p as follows:

$$F_p = W(A, X) \times \psi_p \tag{3}$$

The penalty function is defined as:

$$\psi_p = (1 + \phi)^{\varepsilon} \tag{4}$$

where ϕ is the summation of the stress and displacement penalties, defined as:

$$\phi = \sum_{i=1}^{nm} \phi_\sigma^i + \sum_{j=1}^{nn} \phi_\delta^j \tag{5}$$

The stress constraint penalty ϕ_σ^i for member i and the displacement constraint penalty ϕ_δ^j for node j are respectively defined as:

$$\begin{cases} \phi_\sigma^i = 0 & \text{if} \quad \sigma_i^c \le \sigma_i \le \sigma_i^t \\ \phi_\sigma^i = \dfrac{|\sigma_i - \sigma^{t,c}|}{|\sigma^{t,c}|} & \text{if} \quad \sigma_i < \sigma_i^c \text{ or } \sigma_i > \sigma_i^t \end{cases} \tag{6}$$

$$\begin{cases} \phi_\delta^j = 0 & \text{if} \quad \delta_{\min} \le \delta_j \le \delta_{\max} \\ \phi_\delta^j = \dfrac{|\delta_j - \delta_{\max,\min}|}{|\delta_{\max,\min}|} & \text{if} \quad \delta_j < \delta_{\min} \text{ or } \delta_j > \delta_{\max} \end{cases} \tag{7}$$

Please cite this article in press as: S.O. Degertekin, et al., Sizing, layout and topology design optimization of truss structures using the Jaya algorithm, Appl. Soft Comput. J. (2017), https://doi.org/10.1016/j.asoc.2017.10.001