Advances in Engineering Software 102 (2016) 142–154

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Research paper

Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison

Hoang Anh Pham*

Department of Structural Mechanics, National University of Civil Engineering, 55 Giai Phong Road, Hanoi, Vietnam

ARTICLE INFO

Article history:
Received 7 June 2016
Revised 10 August 2016
Accepted 3 October 2016
Available online 13 October 2016

Keywords:
Truss optimization
Frequency constraint
Differential evolution
Adaptive p-best strategy
Directional mutation
Nearest neighbor comparison

ABSTRACT

Truss optimization with dynamic constraints is well-known as challenging optimization problem and requires appropriate optimization techniques. In this article, a new differential evolution algorithm, named as ANDE, for solving shape and size truss optimization with frequency constraints is presented. Three modifications are introduced to conventional differential evolution (DE), including: 1) the adaptive p-best strategy to balance between global exploration and local exploitation; 2) the directional mutation rule to increase the possibility of creating improved solutions; 3) the nearest neighbor comparison method to prejudge a solution before evaluation and skip unpromising one. These modifications are relatively simple and do not require additional parameter setting for DE. Numerical results of five benchmark examples show that ANDE can provide good and stable results without violation of the frequency constraints. The optimal designs of ANDE in most cases are as good as or better than the results from some state-of-the-art metaheuristics. The benefit of ANDE is that it often uses fewer structural analyses than those required by the other methods.

1. Introduction

Dynamic constraints are essential in structural design to improve the performance of a structure and to prevent the resonance phenomenon [1]. The aim of optimal structural design under frequency constraints is to minimize the weight or an objective function value corresponding to minimal cost of a structure while satisfying frequency constraints. This task in general is complicated and requires appropriate optimization techniques.

Optimal design of truss structure is an important field within structural optimization and has been an extensive research area both in modeling and in development of optimization methods [2]. Truss optimization with frequency constraints is well-known as a challenging optimization problem because of nonlinear constraints and non-convex feasible region. These inherent characteristics of the problem do not favor optimization methods that require gradient information. It is observed that conventional gradient-based methods often fail to obtain global optimum.

Population-based metaheuristic algorithms, such as evolutionary algorithms (EAs), particle swarm optimization (PSO), ant colony optimization (ACO), on the other hand, do not require gradient information and have good global search ability [3]. Therefore, they are useful approaches for tackling this type of problem. Lingyun et al. [4] used genetic algorithm (GA) hybridized with Niche techniques and simplex search for solving shape and sizing optimization of several trusses with multiple frequency constraints. PSO was first presented in [5] for mass minimization of truss structure with frequency constraints with shape and size variables. Since then, numerous population-based metaheuristics have been proposed for solving truss optimization under frequency constraints, such as harmony search (HS) and firefly algorithm (FA) [6], charged system search hybridized with big bang-big crunch (CSS-BBBC) [7], colliding-bodies optimization (CBO) [8], orthogonal multi-gravitational search algorithm (OMGSA) [9], two-dimensional colliding bodies algorithm (2D-CBO) [10], hybridized optimization algorithms (HALC-PSO) [11], teaching-learning-based optimization (TLBO, MC-TLBO) [12], and symbiotic organisms search (SOS, SOS-ABF1, SOS-ABF2) [13]. Although metaheuristics can find good solutions, they do not guarantee that a globally optimal solution can be found. Moreover, metaheuristics often require a high number of function evaluations in order to reach a near optimum. As such, performance enhancement of metaheuristics to obtain sufficiently good result with reasonable computational cost is thus always the issue [14].

While numerous population-based metaheuristic algorithms exist, this article focuses on differential evolution (DE) [15], because

* Fax: +844 35 55 37 86.
E-mail address: anhpham.nuce@gmail.com

http://dx.doi.org/10.1016/j.advengsoft.2016.10.004

it is simple, easy to implement and has shown to be suitable for various optimization problems from diverse domains of science and technology [16]. It was also successful applied for truss optimization problems under static constraints [17–21] and [14]. However, relatively few applications of DE on optimization of truss structures with natural frequency constraints have been reported in the literature. In the work by Pholdee and Bureerat [22], a comparative study of 24 different metaheuristics in solving five benchmark problems of truss optimization was conducted and it was concluded that DE was one of the two best algorithms. In [23], DE was also found competitive to some other metaheuristics, including GA [24] and [25], CSS-BBBC [7], and HRPSO [26]. Both studies [22] and [23] utilized classical DE with the mutation operator 'DE/best/2/bin' to perform the search. Ho-Huu et al. [27] have recently proposed a felicitous approach, which adaptively employs multiple mutation operators in the improved DE algorithm, IDE. The IDE was shown to be able to obtain better results with less function evaluations, comparing with some well-known metaheuristics.

Similar to other metaheuristics, exploration/exploitation balance is a key feature to control the performance of a DE algorithm. Many DE variants have been proposed to deal with this issue and achieved better performance on many problems (see [16,29]). To the best knowledge of the author, none of them (except IDE in [27,28]) has been investigated in solving truss optimization under frequency constraints so far. Moreover, simplicity of DE has been usually decreased in those DE algorithms.

To enhance the performance for solving shape and size truss optimization subjected to multiple frequency constraints while maintaining the simplicity of DE algorithm, three simple modifications are introduced in this article. The modifications include: 1) an adaptive strategy for balancing global exploration and exploitation using a single mutation operator; 2) a directional mutation rule for increasing the possibility of creating an improved solution; and 3) a simple method to prejudge a solution before evaluation so that unpromising solution will be skipped without evaluating it. Combining these modifications, an enhanced DE algorithm, named as ANDE, is developed. Five benchmark examples are used to examine the performance of ANDE. The results obtained by ANDE are compared with those of some most recent metaheuristics reported in the literature.

The rest of this article is organized as follows. In Section 2, the formulation of the truss optimization problem and the constraint handling rules are presented. The basic differential evolution is briefly introduced in Section 3. Then, the ANDE algorithm is described in Section 4. In Section 5, the test problems and numerical results are shown and discussed. Conclusions are given in Section 6.

2. Truss optimization with frequency constraints

2.1. Problem formulation

For the class of truss optimization problems considered in this study, the objective function is the total weight of a truss structure (excluding the added non-structural masses), the design variables are cross-section areas of the truss elements and/or nodal coordinates. The design constraints are limits on natural frequencies. The problem is typically formulated as Eq. (1).

$$\begin{aligned}
\text{Minimize } & W(\mathbf{A},\mathbf{N}) = \sum_{e=1}^{M} L_e A_e \rho_e, \quad e = 1,2,\ldots,M \\
\text{subject to } & f_q \ge f_q^{\min}, \\
& f_r \le f_r^{\max}, \\
& A_i^l \le A_i \le A_i^u, \\
& N_j^l \le N_j \le N_j^u,
\end{aligned} \tag{1}$$

In Eq. (1), $\mathbf{A}=\{A_1, A_2,\ldots, A_m\}$ are m size variables (cross-section areas); $\mathbf{N}=\{N_1, N_2,\ldots, N_n\}$ are n shape variables (nodal coordinates); A_i^l and $A_i^u$are the lower bound and upper bound of the cross-section A_i, respectively; N_j^l and N_j^u are the lower bound and upper bound of the nodal coordinate N_j; $W(\mathbf{A}, \mathbf{N})$ is the weight of the truss; L_e, A_e and ρ_e are the length, the cross-section area and the material density of the eth element, respectively; M is the number of elements; f_q and $f_q^{\min}$ are the qth natural frequency and its minimum limit, respectively; $f_r$and$f_r^{\max}$ are the rth natural frequency and its maximum limit.

2.2. Constraint handling

For convenience, the frequency constraints in Eq. (1) are rewritten in the normalized forms of Eq. (2) and the constraint violation is determined by Eq. (3):

$$\begin{aligned}
c_q &= f_q^{\min}/f_q - 1 \le 0, \\
c_r &= f_r/f_r^{\max} - 1 \le 0
\end{aligned} \tag{2}$$

$$C(\mathbf{A},\mathbf{N}) = \sum_q \max\{0, c_q\} + \sum_r \max\{0, c_r\} \tag{3}$$

where $C(\mathbf{A}, \mathbf{N})$ is the constraint violation; c_q and c_r are the normalized constraints. Thus, a solution is feasible if its constraint violation, $C(\mathbf{A}, \mathbf{N})$, equals to zero, otherwise it is infeasible. In this study, the following constraint handling rules introduced in [30] are employed:

1) A feasible solution is better than any infeasible one.
2) Between two feasible solutions or two solutions with equal constraint violations, the one having smaller objective function value is better.
3) Between two infeasible solutions, the one having smaller constraint violation is better.

To deal with bound constraints, the cutting-off technique [31] is adopted, i.e. the generated violating value is substituted by the bound value, since in many cases the optimum solution is located at one of the bounds of a given design variable.

3. Basic differential evolution

Differential evolution (DE), which was introduced by Storn and Price [15], is a population-based optimizer. DE uses a population of NP candidate vectors $\mathbf{x}_k$ ($k=1, 2,\ldots, NP$) (called individuals) of the design variables. The population is then restructured by survival individuals evolutionally. First, an initial population is randomly sampled from the solution space as Eq. (4),

$$x_{k,i} = x_i^l + rand[0,1] \times (x_i^u - x_i^l), \quad i = 1,2,\ldots,D \tag{4}$$

where x_i^l and x_i^u are the lower and the upper bounds of x_i, respectively; D is the number of design variables of the optimization problem; $rand[0, 1]$ is a uniformly distributed random real value in the range [0, 1]. Then, each individual $\mathbf{x}_k$ (called the target vector) of the current population is compared with a newly generated vector (called the trial vector) and the better one will be selected as member for the population of next generation. The evolution proceeds until a termination criterion is reached.

The crucial idea behind DE is the scheme for producing trial vectors. Two operators, named as 'mutation' and 'crossover', are used for this purpose and they are described as follows.