Computers and Structures 198 (2018) 51–60

Contents lists available at ScienceDirect

Computers and Structures

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

A real-coded genetic algorithm for optimizing the damping response of composite laminates

M.V. Pathan [a], S. Patsias [b], V.L. Tagarielli [a,*]

[a] Department of Aeronautics, Imperial College London, SW7 2AZ London, UK
[b] Rolls-Royce plc, PO Box 31, DE24 8BJ Derby, UK

ARTICLE INFO

Article history:
Received 11 June 2017
Accepted 12 January 2018

Keywords:
Real-coded genetic algorithm
Composite laminates
Optimization
Damping

ABSTRACT

We develop a real-coded constrained genetic algorithm (GA) and assess its performance for the case of selected classical optimisation problems. The proposed GA uses a roulette selection method, BLX-α crossover operation, non-uniform mutation along with single elitist selection at every generation. The GA is then applied, in conjunction with the finite element (FE) method, to optimise the damping response of a laminate comprising unidirectional composite laminae and viscoelastic damping layers. Modal loss factors are maximised against the constraints of given structural stiffness and mass.

1. Introduction

Fibre reinforced polymers (FRPs) are being increasingly used in automotive and aerospace sector due to their superior specific stiffness, strength and damping. These materials can be tailored by adjusting volume fractions of the constituents, layer thickness and stacking sequence. Recently, the issue of accurate numerical prediction of the stiffness and damping properties of the FRPs has been addressed at both ply [1–4] and laminate [5] level. In certain applications however, the inherent damping of composites plies is not sufficient and viscoelastic damping inserts are used to increase structural damping; this is typically done placing a compliant damping layer between relatively stiffer composite plies, to induce shear deformation in the soft material, thereby dissipating energy. The poor stiffness of damping layers, with density comparable to that of composite plies, may degrade the specific structural stiffness of the laminate.

The effectiveness of damping layers also depends on their thickness and location in a given laminate. Several authors have studied the optimal location of viscoelastic layers for maximum damping in composite laminates, using analytical or numerical techniques [6–15]. The task can be formulated as a constrained optimization problem, where the objective is to maximize the damping capacity of a laminate, having certain constraints on mass, stiffness and load-carrying capacity [7,15]; single- and multi-objective algorithms have been published (e.g. [11,16]). The effectiveness of optimisation algorithms to maximise the damping of structures made from laminates also depends on the type of objective function; different objective functions have been considered in the literature: maximum modal loss factors or their sums [6,8], minimum deflection at resonant frequencies [12], minimum vibrational energy [13], among others. Several studies have also explored the use of discontinuous damping surface patches to increase the modal damping capacity of the laminate [8,17].

The design space of laminated composite structures has a high number of dimensions of both discrete and continuous nature. Classical non-linear programming techniques are unsuited for such non-convex search spaces given the fact they are local search methods that have a tendency to get stuck in the local extrema. Typically, such problems are better handled by techniques belonging to the class of evolutionary algorithms, most popular of which are genetic algorithms (GAs). GAs are inspired by the natural evolutionary principles of selection, crossover, mutation and evolution; for a comprehensive discussion of GAs the readers are referred to [18,19]. The idea is to select the best performing candidate solutions in a certain population and then combine their genomic information to possibly create children with better performance; GAs are stochastic search-based approaches which makes them efficient over other methods. Several authors have applied GAs to optimize damping in FRP laminates. Zheng et al. [13,20] optimized thickness and location of a viscous patch for the case of a simply-supported beam; Xie et al. [12] minimized structural displacements at resonant frequencies by tailoring the thicknesses of the constituent plies. Montemurro et al. [7] performed damping optimization of a hybrid laminate consisting of

* Corresponding author.
E-mail address: v.tagarielli@imperial.ac.uk (V.L. Tagarielli).

https://doi.org/10.1016/j.compstruc.2018.01.005
0045-7949/

transversely isotropic FRP plies and isotropic viscoelastic damping plies. The design variables considered in this study were ply number, laminate sequence and ply thickness. More recently, Xu et al. [8] performed multi-objective design optimization of damping in a laminate using FE analysis with a modified NSGA-II algorithm [21]. Most of the works dealing with optimization of FRP laminates are based on binary-coded GAs, which are not efficient in dealing with real-valued design variables [22]. Moreover, the inherent material damping due to the fibre composites has been largely ignored in all studies.

In this paper we develop a real-coded GA (i.e. a GA using real number representation for the candidate solutions) and test its effectiveness in dealing with non-convex benchmark optimization problems, comparing to selected state-of-the-art evolutionary algorithms. The algorithm is then employed to maximise the damping of a cantilever beam, with constraints on the structural mass and stiffness; the design variables are ply thickness and stacking sequence. This is done in conjunction with FE simulations, in which the response of a cantilever beam is simulated in detail, including all non-linearities as well as the anisotropic, viscoelastic response of all constituent materials.

The outline of the paper is as follows: in Section 2 we define the optimization problem and give details of the proposed real-coded GA. In Section 3 we test the algorithm in selected benchmark problems and in Section 4 we apply the proposed GA to the case of layered composites with damping layers.

2. Optimisation algorithm

2.1. Constrained optimization problem

A single-objective constrained optimization problem can be formulated as the minimization problem

$$\min f(\vec{x}),\ \text{subject to}$$
$$\begin{cases} g_j(\vec{x}) \leqslant 0, & j = 1, \dots, l \\ h_j(\vec{x}) = 0, & j = l+1, \dots, m \\ p_j \leqslant x_j \leqslant q_j, & j = 1, \dots, n \end{cases} \tag{1}$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the n-dimensional solution vector; the function is subjected to l inequality constraints (g_j) and $m - l$ equality constraints, represented by h_j. In practice the equality constraints are difficult to handle and are usually converted to inequality constraints by allowing a small tolerance ε, i.e.

$$|h_j(\vec{x})| - \varepsilon \leqslant 0. \tag{2}$$

Penalty based methods [18] are often used in optimization algorithms to handle constraints, which can be linear or non-linear in nature. This approach transforms a constrained optimisation problem into an unconstrained one, by suitable modification of the original objective function $f(\vec{x})$. The 'penalised' objective function $f_p(\vec{x})$ is constructed as

$$f_p(\vec{x}) = \begin{cases} f(\vec{x}) & \text{if } g_i(\vec{x}) \leqslant 0 \text{ and } h_j(\vec{x}) = 0 \\ f(\vec{x}) + \sum_{i=1}^{l} c_i G_i(\vec{x}) & \text{if } g_i(\vec{x}) > 0 \text{ and } h_j(\vec{x}) = 0 \\ f(\vec{x}) + \sum_{j=l+1}^{m} d_j H_j(\vec{x}) & \text{if } g_i(\vec{x}) \leqslant 0 \text{ and } h_j(\vec{x}) \neq 0 \\ f(\vec{x}) + \sum_{i=1}^{l} c_i G_i(\vec{x}) + \sum_{j=l+1}^{m} d_j H_j(\vec{x}) & \text{if } g_i(\vec{x}) > 0 \text{ and } h_j(\vec{x}) \neq 0 \end{cases} \tag{3}$$

$$\begin{aligned} G_i(\vec{x}) &= \max\{0, g_i(\vec{x})\}; \quad i = 1, \dots, l \\ H_j(\vec{x}) &= \max\{0, h_j(\vec{x})\}; \quad j = l+1, \dots, m \end{aligned} \tag{4}$$

The selection of the penalization coefficients c_i and d_j is difficult, as high penalty coefficients limits the accuracy in proximity of the constraints while low coefficients results in a large number of iterations.

2.2. Genetic algorithm

Encoding of the candidate solutions is essential for an efficient GA search process. Traditionally, the candidate solutions (chromosomes) are coded using binary representations due to simplicity of implementation. However, the binary representation has limitations when dealing with continuous search spaces, where the size of binary strings can grow in length, resulting in storage and manipulation problems [23]. Binary coding also suffers from the "Hamming Cliffs" problem [22]. The use of real-coding is more natural for real-valued design variables, i.e. continuous search spaces, as it substantially simplifies the algorithm, resulting in higher efficiency.

Genetic algorithms are unconstrained optimization techniques, and constraints are imposed using penalty-based methods [7,24] or via multi-objective optimization approach [11,16]. In this study we use a recently proposed penalty-based method, referred to as automatic dynamic penalization strategy [25].

2.3. Description of proposed GA

In this study we develop a real-coded GA with a 'roulette' selection method, BLX-0.5 crossover operator [26] and non-uniform mutation (NUM) operator proposed by Michalewicz [19]. The choice of GA operators are based on findings of Herrera et al. [22], where several types of real-coded GA operators were compared, concluding that the BLX-0.5 and NUM gave the best performance. We briefly describe these operators below.

2.3.1. BLX-α crossover operator

For two parent candidate solutions with n design variables, $\vec{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ and $\vec{x}_j = [x_{j,1}, x_{j,2}, \dots, x_{j,n}]$ selected from a population of size P_N at generation t, $X^t = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_{P_N}]$, the BLX-$\alpha$ operator generates the k-th component of a new offspring $\vec{x}_z$ belonging to the next generation, i.e. to the population at time $t+1$, X^{t+1}. The k-th component of $\vec{x}_z$ is a uniform random scalar in the range $[\min(x_{i,k}, x_{j,k}) - \alpha I, \max(x_{i,k}, x_{j,k}) + \alpha I]$, where I defines the distance between parent candidates given by $I = \max(x_{i,k}, x_{j,k}) - \min(x_{i,k}, x_{j,k})$ and α is a user defined parameter.

The effectiveness of the BLX-α is in its ability to search in a space domain not necessarily bounded by that of the parents; in addition, the GA is self-adaptive, since the search space depends on the distance between the parents. The choice of the parameter α is crucial as it quantitatively defines the search domain. In this study we use $\alpha = 0.5$, based on the findings in Herrera et al. [22].

For the case of child solutions violating the design variable bounds, the values of the design variables are forced to the value of the nearest bound, to ensure that the search stays within the desired space.

2.3.2. Non-uniform mutation operator

The NUM operator, as the BLX-α operator, possesses self-adaptive capabilities; this algorithm reduces the range of the allowable mutations with increasing generation number t, allowing for larger mutations at small t and fine-tuning towards the end of the optimisation problem. This allows for efficient search throughout the allowable search space but it ensures that good solutions are not lost at later generations. The operator mutates the k-th component of a certain parent $\vec{x}_i$ as