Indian Institute of Technology, Kanpur

Project Report

ME752A: Optimmization Methods

Application of Genetic Algorithm for Combinatorial Optimization of Weight of a Composite Plate

Submitted By:
Gaurav Gupta(11272)

Instructor:
Dr. Bhaskar Dasgupta

December 1, 2015

Abstract

A multi-constraint combinotorial optimization methodology for the design of laminated composite materials is presented. Maximum allowable displacement for given loading condition was taken as the design constraint. The stacking sequence of layers and the thickness of the structure were optimized for a preassigned geometry of the structure, constituent materials and loading condition. Genetic algorithm was used for the optimization process and the results for a sample problem show the validity of the proposed methodology.

Introduction

Composite structural element are widely used in a variety of components for automative, aviation, defense industry as well as household applications. Major advantage of composite materials over conventional materials is their high specific load carring capacity. This ability is imparted to these materials by load carrying fibers held together by a resin matrix.

For analysis, composite laminate with unidirectional fiber mats were considered and the thickness of each layers was uniform. The elastic moduli of the composites were derived from the moduli of the fiber and of the matrix by the Halpin sai equations -

$$E_1 = E_f V_f + E_m V_m \tag{1}$$

$$\nu_{12} = \nu_f V_f + \nu_m V_m \tag{2}$$

$$\frac{M}{M_m} = \frac{1 + \zeta\eta V_f}{1 - \eta V_f} \tag{3}$$

$$\eta = \frac{(M_f/M_m) - 1}{(M_f/M_m) + \zeta} \tag{4}$$

where

E elastic modulus

G shear modulus

ν Poisson ration M composite modulus E_2, G_{12}, or ν_{23}

M_f corresponding fiber modulus E_f, G_f, or ν_{23}

M_m correponding matrix modulus E_m, G_m, or ν_m

V_f fiber volume fraction

ζ 2 for calculation for E_2
ζ 1 for calculation for G_{12}.

The aforementioned formulation provides the equiavalent mechanical properties of an individual lamina. However, for strength and loading analysis, it is generally more convenient to express the combined properties of all the lamina i.e. effective laminate engineering constants. The following equations establish the relationship between loading and deformation of the laminate.

$$\begin{Bmatrix} N_x \\ N_Y \\ N_{XY} \\ M_X \\ M_y \\ M_{XY} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{5}$$

N_x Force along x direction per unit width (Nm^{-1})
N_y Force along y direction per unit width (Nm^{-1})
N_{xy} Shear force per unit width (Nm^{-1})
M_x Moment along x direction per unit width (N)
M_y Moment along y direction per unit width (N)
M_{xy} Shear moment per unit width (N)

In the above formulation, the sub-matrices A, B and D are defined as follows -

$$[A] = \sum_{k=1}^{n} [\bar{Q}_{ij}]_k (h_k - h_{k-1}) \tag{6}$$

$$[B] = \frac{1}{2} \sum_{k=1}^{n} [\bar{Q}_{ij}]_k (h_k^2 - h_{k-1}^2) \tag{7}$$

$$[D] = \frac{1}{3} \sum_{k=1}^{n} [\bar{Q}_{ij}]_k (h_k^3 - h_{k-1}^3) \tag{8}$$

Here,

$$\bar{Q} = T^{-}1QT \tag{9}$$

$$T = \begin{bmatrix} cos^2\theta & sin^2\theta & 2sin\theta cos\theta \\ cos^2\theta & sin^2\theta & 2sin\theta cos\theta \\ -sin\theta cos\theta & sin\theta cos\theta & cos^2\theta - sin^2\theta \end{bmatrix} \tag{10}$$

θ Fiber inclination angle from the x-axis

Q Stiffness Matrix of a single lamina aligned with the principal axes

Problem Statement

Minimize H(θ_j)$_i$ $\quad$ $x_i = [\theta_j]_i$, H_i
Subject to

$$\begin{Bmatrix} \epsilon_{x^k} \\ \epsilon_{y^k} \\ \epsilon_{xy^k} \end{Bmatrix}_{max} \leq \begin{Bmatrix} \epsilon^*_x \\ \epsilon^*_y \\ \epsilon^*_{xy} \end{Bmatrix} \tag{11}$$

Here

$[\ \epsilon_{x^k}\ \epsilon_{y^k}\ \epsilon_{xy^k}\ ]^T$	Maximum stress in the laminate
$[\ \epsilon_{x^*}\ \epsilon_{y^*}\ \epsilon_{xy^*}\ ]^T$	User defined maximum allowable strain values.
W	weight
x_i	design vector ($[\theta_j]_i$, H_i)

Genetic Algorithm

In the field of artificial intelligence, genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms generate solutions to optimization problems using techniques inspired by natural evolution such as inheritance, mutation, selection and crossover. Genetic algorithms were first used to solve optimization problems in 1970's.

The method operates with a set of potential solutions. This is referred to as a population and members of the population are sometimes called individuals. The population changes over time but always has the same size N, say. Each individual is represented by a single string of characters. At every iteration of the algorithm a fitness value $f(i)(i = 1, 2, 3, 4...., N)$, is calculated for each of the current individuals. Based on the fitness function a number of individuals are selected as potential parents. These form what is called a mating pool.

Two new individuals can be obtained from two parents in the mating pool by choosing a random point along the string, splitting both strings at that point and then joining the front part of one parent to the back part of the other parent and vice versa. Thus parents A-B-C-A-B-C-A-B-C and A-A-B-B-C-C-C-B-A might produce offspring A-B-C-B-C-C-C-B-A and A-A-B-A-B-C-A-B-C when mated. This process is called crossover. It is not necessary for every member of the mating pool to be involved in crossover; some will be left unchanged. Individuals in the mating pool may also change through random mutation, when characters within a string are changed directly. Normally each character is given such a small probability of being changed, that most of the time individuals are left unaltered. The processes of crossover and mutation are collectively referred to as recombination. The end result is a new population (the next "generation") and the whole process repeats. Over time this leads to convergence within a population with fewer and fewer differences between individuals. When a genetic algorithm works well the population converges to a good solution of the underlying optimization problem and the best individual in the population after many generations is likely to be close to the global optimum.

Strategy

Composite consists of $2n$ layers of laminates. Each layer can have an orientation of either 0^o or 90^o Orientation sequence is symmetric about the mid plane so we only consider the top n layers. Total thickness of the composite is say H. Every composite can be represented by a string containing information regarding the thickness of the composite and orientation of each layer. As there are only two possible orientations for each layer, it can be represented by a binary digit (say 0 for 0^o and 1 for 90^o). The thickness of the composite can be discretized and it can be stored in a binary string of 8 bits, i.e. we have 256 discrete values for thickness. Every possible composite (individual) can be represented by a binary string whose first 8 bits give the thickness and the succeeding n bits represent the orientation of each layer. In total the every string is n+8 bits long.
A general string: $(x_1x_2x_3.....x_{n+7}x_{n+8})$

We initially select a population of N random individuals. All the individuals in the population are rated by a fitness criteria given by a function f(x). Top 0.2 individuals are carried to the next generation directly and the remaining 0.8 form the mating pool. Individuals in the mating pool are paired (parents) and undergo crossover operations to give two new individuals (child 1 and child 2). Fitness of the new individuals is calculated and compared with those of the parents. Two best out of the four are selected and sent to the next generation. This step is performed for every pair. The individuals in the next generation that come from the mating pool undergo mutation with a probability,p_m.

Algorithm

Step 1. Select a population of N random individuals
Repeat
Step 2.For each individual assign fitness using f(x)
Step 3.Top 0.2 individuals of the population are carried forward to next generation
Step 4. Remaining 0.8 individuals are paired and undergo crossover
Step 5. Fitness is assigned to all the new individuals (child strings)
Step 6. From every group of parent and child strings two best are sent to next generation
Step 7. Mutation is performed on the individuals coming from the mating pool with a probability p_m
until population variance in fitness of the individuals is small
Step 8. The best individual in the population is the solution

Results

The program was run for two values of n (n=8 and n=16). Population size, N was set equal to 10 or 100. The effect of these variables on the convergence was observed. The constraints were set by the strain constrains. On changing the value of maximum allowable strain, the allowed height changes as well.
For strain $\epsilon > 0.1$ H belongs to $(1.8, 2.2)$. This solution was attained after 100 iterations. See figure 1.
For strain $\epsilon > 0.01$ H belongs to $(8, 13)$. This code was run for 30 iterations. See figure 2.
For strain $\epsilon > 0.01$ H belongs to $(8.61, 9.61)$. This solution was attained after 100 iterations. See figure 3.

Addendum in results

Apart from being a function of the tolerable strains and number of layers, the optimum height and the orientation obtained were also a function of the loading applied on the specimen. Both the magnitude and the direction of loading changed the solution. For a given magnitude of loading, the height varied negligibly with the direction of loading. Let us look at some results that were obtained in the original algorithm and were later verified using the updated algorithm.

1. For just axial loading, without any moments, the height is purely a function of the tolerances and magnitude of loading. As can be seen in the construction of the A matrix, the position and orientation of layers do not matter. And hence for all the orientations we get the same results for optimum height.

2. Once the moments are applied, for the same height, one orientation is preferred over others. This happens because the stress varies with height in case of moments. Since laminates have differing directional strengths, it is important to place appropriate ply at the places of maximum stress. Thus if only the M_x component of moment is applied, the orientation of the form [0 0 0 0 . . .] are preferred. If similarly only the M_y component is applied the orientation [90 90 90 90 . . .] is preferred. For a general state of stress in between, a combination is obtained as the optimum solution.

3. A strange result was observed in the case of loading of format N = [1 1 1] and M = [1 1 1]. Here all the components of normal forces and moments are non zero and are equal to 1. In such a case as well, the orientation becomes obsolete. This happens as due to symmetry in X, Y and XY axis, the orientation of the layers becomes obsolete.

Verification

The objective function where height is minimized was observed for several values of strain constrains. It was observed that as the constraints were tightened, the optimal height of the laminate increased. If the strain constraints were set to 0.5, i.e. each component was expected to be less than 0.5, the optimal height was found in the range (0.81,0.91). If the constraint was tightened, say to 0.1, the optimal height increased. The updated heights were in the range (1.8, 2.2). If the constraints were further tightened, say to 0.01. The updated heights were in the range (8.61, 9.61). This behavior can be expected and can be seen as logically consistent with the theories in place. As the thickness of the laminate decreases, for given stresses, the strains developed are higher since the moments of inertia reduce as laminate becomes thin. Hence, it can also be expected that for tight enough constraints no solutions can be found. The code was checked for strain constrains valued at 0.0001. Given the accuracy of the inherent methods, no solutions were found to be satisfying these tight constrains.

Checking on Simpler Problems

Since we were treading in uncharted territories, we thought it was advisable to run the code on simple problems whose solutions we actually know. If the code works properly

for these problems and could generate accurate solutions, it could be expected of the code that it could generate accurate solutions for the laminate problems as well. The simple problem chose was $x + y + z$. we expected to minimize $x + y + z$ using the genetic algorithm techniques. While solving the simple problem, valuable insights in the working of algorithm were developed that helped us shape our expectations of the solutions to the real problem. These are discussed in what follows.

1. It was seen that the given algorithm and the coding done for it were successful in finding out the solution to both the unconstrained and the constrained versions of the simple problem. For unconstrained problem, the optimum zero was reached. When the constraints $x >= 3$, $y >= 3$, $z >= 3$ were applied, the optimum solution of 12 was achieved.

2. It was observed that for small initial population sizes larger number of iterations were required to arrive at the solution. This led us to expect that if the solution in the laminate problem does not converge for the given iteration count, but seems to be converging slowly, the exact solution can be achieved by running the code for a longer duration of time.

Further Verification

For the problem described above, the solution was found keeping both the orientation and the height as genetic variables, this leads to desretisation of height. Hence height cannot take all the possible values and we get a nearby solution depending upon the accuracy chosen. We tried to work our way around this by making the height a continuous variable. Thus in the new formulation, for a given orientation that is a genetic variable, the optimum height is calculated using a univariate line search for H. It was observed that the values obtained as solution for this algorithm were very close, within the accuracy limits, of the solutions obtained using purely genetic algorithm that was employed previously. The new algorithm was highly efficient though as the solutions were arrived at in only one or two iteration, we take this as the benchmark for verification.

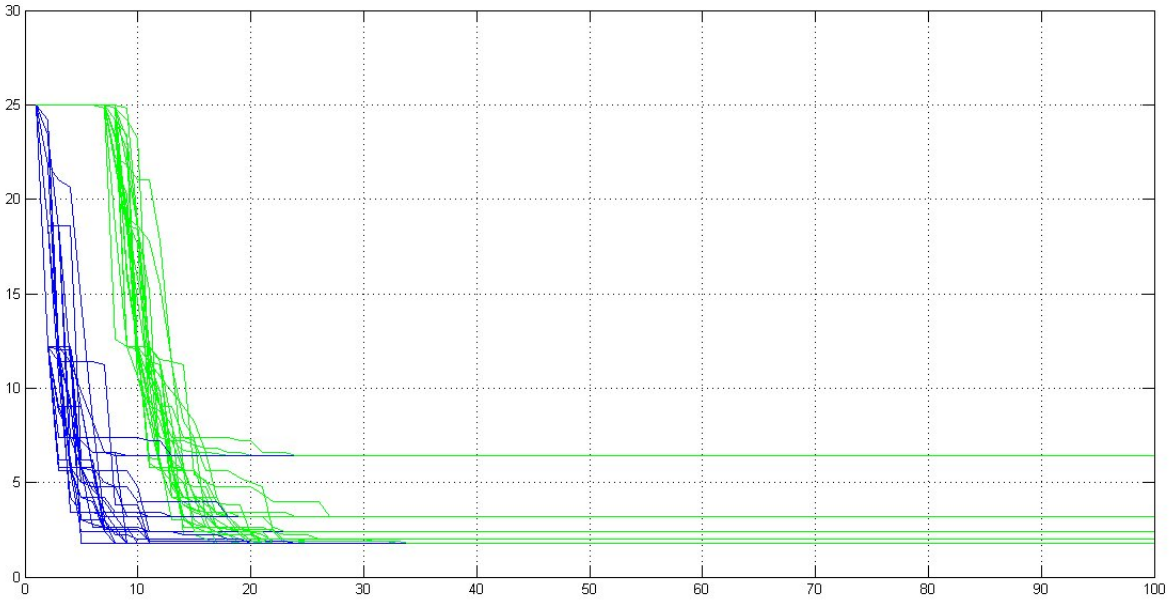

Figure 1: n = 8, N = 10 and 100 iterations

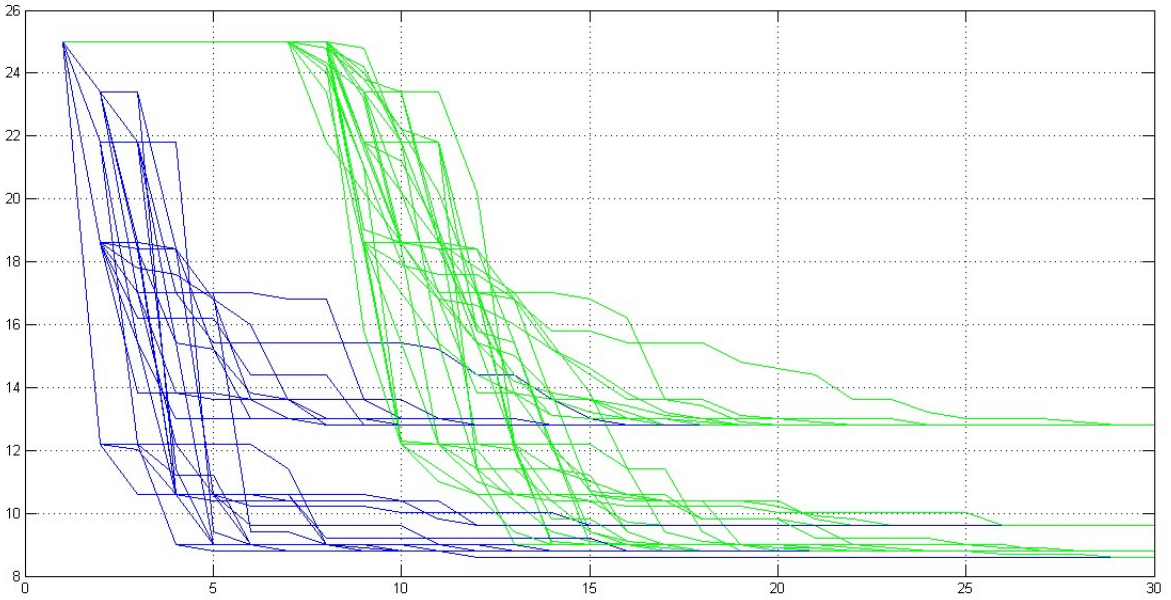

Figure 2: n = 8, N = 10 and 30 iterations

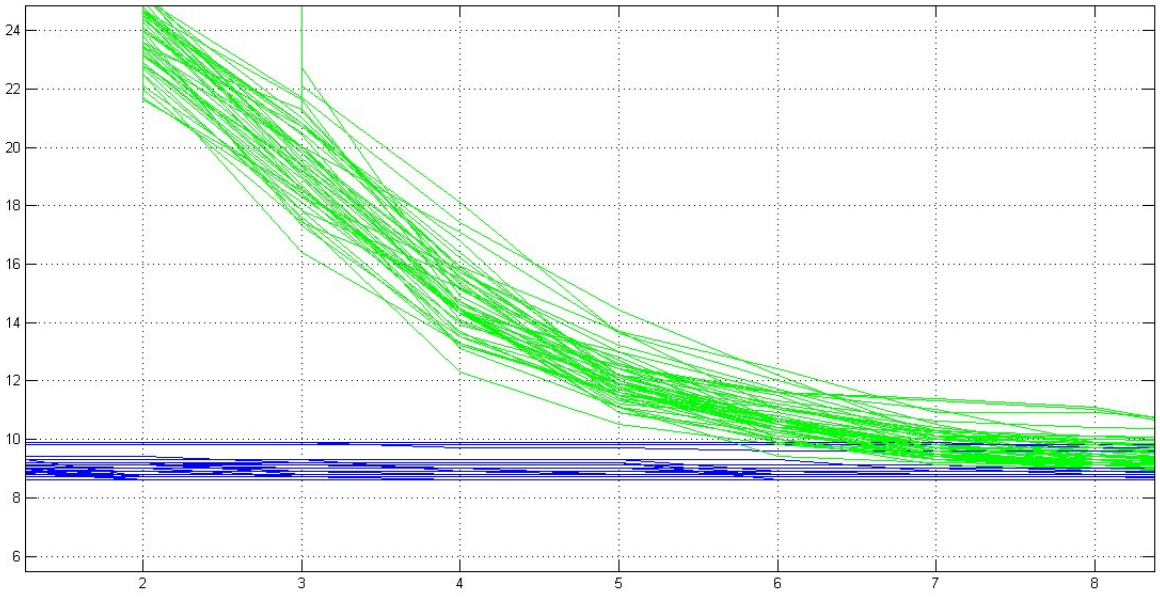

Figure 3: n = 16, N = 100 and 100 iterations