Application of Genetic Algorithm in Design of Laminated Composites

[1]Piyush Sondankar, [2]R. R. Arakerimath
Department of Mechanical Engineering, G. H. Raisoni College of Engineering and Management, India.

Abstract:
Composite laminate configuration requires discrete programming to locate the right number of handles with thicknesses, angular orientation, and material sorts, which are typically confined to a discrete arrangement of genes. Genetic algorithms (GA's) are one of only a handful couple of streamlining instruments accessible that are appropriate to such discrete critical thinking situations. The fundamental objective of this work is to consider the GA's capacity to be effectively adjusted to various kinds of composite laminate structure streamlining issues. Two distinct variants of a genetic algorithm, GA-I and GA-II, were grown explicitly to achieve these errands. To exhibit the adaptability of the GA structure, the GA-II algorithm was conceived to deal with increasingly complex composite laminate designs developed from numerous materials. The changed GA used two chromosome strings to speak to the composite laminate. The main string characterized the introduction point of every lamina, and the second string characterized a lamina's material kind. By utilizing two diverse chromosome strings, just little adjustments to the different genetic administrators were required. The streamlining definition was done by deciding separate expense and weight target capacities. A raised mix of these two goals was utilized for laminate wellness, and in this manner required no extra adjustments to the GA. The target of this paper is to devise a genetic algorithm for stacking succession structure of symmetrically laminated composite plates. Stacking succession configuration infers the assurance of the quantity of employs in the laminate just as their introduction. With this component, the GA might be utilized to control laminate weight by changing the quantity of handles in the laminate stacking succession. The genetic algorithm will not be permitted to change the dimensional components of the plate all through the improvement procedure. Two distinct forms of a genetic algorithm are investigated.

Keywords: *Design of laminated composites, genetic algorithms.*

I. INTRODUCTION

For the GA-I algorithm, one string of genes is utilized to speak to one portion of a symmetrically laminated composite plate. The length of the gene string is kept fixed all through the optimization run. Every gene in the string is spoken to by an integer esteem somewhere in the range of 0 and 10 and determines whether the lamina stack location is vacant or occupied with a lamina which might be oriented at any angle somewhere in the range of 0^o and 90^o, in increments of 10^o, see Figure1. In spite of the fact that the gene string length is fixed, having void plies makes it possible to change the laminate value of thickness during optimization run.

The execution of irregular choices with given probabilities is easier to describe and program for integer intervals than for arbitrary sets of items as well [1]. All plies in the stacking succession have the equivalent prescribed thickness esteem. A case of a decoded stack is given in Figure 2, where E speaks to an unfilled lamina. Note that unfilled laminae are pushed to the external edge (left end) of the laminate stacking succession to avoid having voids in the laminate.

0 = emprty lamina (E)
1 = lamina oriented at 0^o
2 = lamina oriented at 10^o
3 = lamina oriented at 20^o
4 = lamina oriented at 30^o
5 = lamina oriented at 40^o
6 = lamina oriented at 50^o
7 = lamina oriented at 60^o
8 = lamina oriented at 70^o
9 = lamina oriented at 80^o
10 = lamina oriented at 90^o

Figure 1: GA-I code for laminate stack.

II. PROCEDURE FOR GENETIC ALGORITHM

An initial population, of genetic strings with haphazardly picked genes, is made first. The size of the population utilized in the present work remains steady all through the genetic optimization. Various genetic administrators are applied at given probabilities to create new laminates. In request to frame successive generations, guardians are browsed the present population dependent on their fitness. The fitness calculation as a rule involves function esteems that are determined from independent analysis subroutines or bundles. Next, the crossover, mutation, and swapping administrators are applied to make child designs, who are ideally more qualified to their environment than their folks. The child population is then broke down and positioned. To finish the generation cycle, a selection plot is implemented which determines which laminates from the child and parent population will be set in the generations to come. One generation after another is made until some stopping criterion is met. A flowchart of genetic algorithm technique is given in Figure 3.

Coded orientation = [0/8/6/4/3]
Decoded orientation = [E/70^o/50^o/30^o/20^o]

E
70^o
50^o
30^o
20^o

Figure 2: Sample stack sequence structure for GA-I

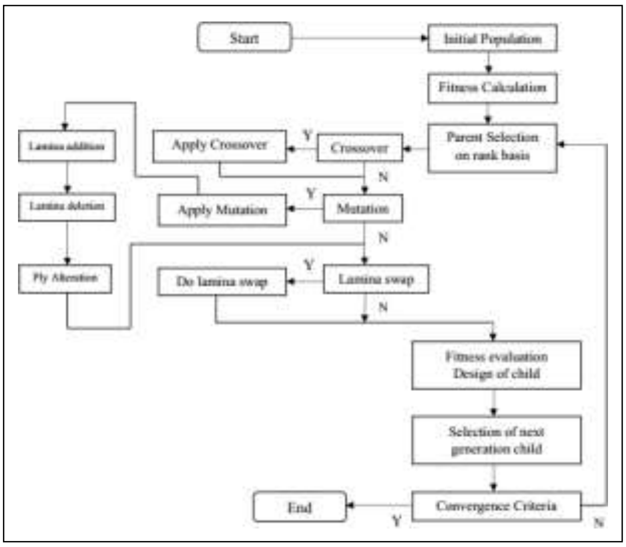
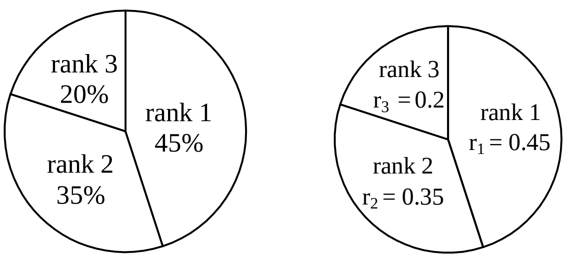


Figure 3: Genetic algorithm flowchart

Before beginning of parent selection, all laminates must be positioned from best to worst according to the estimation of each laminate's objective function [2]. A roulette wheel selection is implemented where the i[th] positioned laminate in the population is given an interval [i-1, i), whose size relies upon the population size, P, and its position, 'i' in the population.

$$(\square) = \square(\square - 1) + \frac{2(\square - \square + 1)}{(\square + 1)} \quad \dots (1)$$

For instance, if there are three laminates in a population, the roulette wheel is divided into three pieces with the best laminate taking 45% of the wheel, the second best taking 35%, and the most unfortunate taking 20%, see Figure 4(a). A uniformly distributed irregular number is created somewhere in the range of 0 and 1. Laminate i is chosen as a parent if the number lies in the interval [i-1, i), Continuing with the above precedent, if arbitrary numbers $r_1 = 0.45$ and $r_2 = 0.35$ are drawn with random distribution concept. At that point laminate 1 and laminate 2 will move toward becoming guardians of the first child, see the Figure 4(b). Guardians of a child are required to be distinct laminates from the population [3].



a) Roulette wheel distribution for 3 laminates

b) Parent selection using random numbers

Figure 4

2.1 Crossover

Children are made by combining a portion of each parent's genetic string in an operation called crossover. To determine the point of crossover, a uniformly distributed arbitrary number is picked and after that multiplied by one less exactly the maximum number of non-void genes in the two guardians. The integer ceiling estimation of this item determines the crossover point, see Figure 5. The gene string is then split at a similar point in the two guardians. The left piece from parent 1 and the right piece from parent 2 are combined to frame a child laminate. To guarantee that unfilled plies are not swapped, every single void employ are pushed to one side of the coded string. Child laminates are likewise compelled to be distinct from one another and from laminates in the parent population. If a distinct child cannot be found after a prescribed number of iterations, at that point one of the guardians is cloned into the child population too. The procedure is rehashed the same number of times as important to make another population of laminates [4].

GA Code

Parent 1	[0/6	/8/7/5]
Parent 2	[0/2	/9/4/3]
Child	[0/6 /9/4/3]	

Decoded sequence

Parent 1	[E/50º	/70º/60º/40º]
Parent 2	[E/10º	/80º/30º/20º]
Child	[E/50º /80º/30º/20º]	

Figure 5: Crossover

2.2. Mutation

After a child is made, the operations of adding, deleting, or mutating genes happen with little probabilities. These administrators make up genetic mutation, and are illustrated in Figure 6. While adding a lamina stack, a uniform irregular number is picked to determine the orientation. For the design issues considered in this work, external plies in the laminate will get set up quicker because they affect the objective function. In this manner, included lamina stacks are constantly introduced at the mid-plane of the laminate. To erase a lamina stack, an irregular number is picked and the corresponding stack is expelled from the stacking grouping by replacing it with a 0 gene. The laminate is then re-stacked with the goal that every single void handle are pushed to the external edge of the laminate, Every gene in the string switches with a little probability to some other permissible integer aside from 0's and the estimation of the genes before lamina alteration happens. Lamina alteration does not work on void genes either.

2.3 Lamina Swap

A permutation administrator was regularly used to aid the genetic search. However if it brings about shuffling of digits a lot in the gene string [12], a less disruptive administrator, swapping of lamina, would be inserted for permutation for this work. The swapping of lamina administrator is implemented by arbitrarily selecting two genes in the string and switching their positions, see Figure 6(d). Swapping of lamina can be effective for issues where certain pieces of the laminate stacking succession get set up quicker than others [5]. For instance, if the optimal stacking succession for the external section of the laminate has been determined first the swapping of lamina administrator may enable the GA to determine the optimal orientations for the inner piece of the laminate by swapping plies from each section.

GA code

Before lamina addition	[0/6/8/7/5]
After lamina addition	[6/8/7/5/10]

Decoded sequence

Before lamina addition	[E/50º/70º/60º/40º]
After lamina addition	[50º/70º/60º/40º/90º]

(a) Lamina addition (at least 1 empty stack)

GA code

Before lamina deletion	[4/6/8/7/5]
After lamina deletion	[4/6/0/7/5]
Restack	[0/4/6/7/5]

Decoded sequence

Before lamina deletion	[30 º/50 º/70 º/60 º/40 º]
After lamina deletion	[30 º/50 º/E/60 º/40 º]
Restack	[E/30 º/50 º/60 º/40 º]

(b) Lamina deletion

GA code

Before lamina alteration	[3/2/9/7/4]
After lamina alteration	[3/2/9/2/4]

Decoded sequence

Before lamina deletion	[20 º/10 º/80 º/60 º/30 º]
After lamina deletion	[20 º/10 º/80 º/10 º/30 º]

(c) Single lamina alteration (filled plies only)

GA code

Before lamina alteration	[6/1/7/3/5]
After lamina alteration	[6/1/5/3/7]

Decoded sequence

Before lamina alteration	[50 º/0 º/60 º/20 º/40 º]
After lamina alteration	[50 º/0 º/40 º/20 º/60 º]

(d) Swapping of lamina

Figure 6: Mutation

III. COMPOSITE LAMINATES WITH MULTIPLE MATERIALS

In this section, modifications to the GA-I algorithm to take into account stacking groupings with multiple materials will be discussed. The second version of the genetic algorithm will be called GA-II. In the previous section, the entire laminate was comprised of one material. Along these lines, one chromosome consisting of one gene was sufficient to speak to the laminate stacking arrangement. Nevertheless, to oblige at least two materials, every chromosome is extended to include two gene strings, one for lamina orientation and another for material definition. The representation of genes by integers in each string is maintained. Genes in the first string will indeed determine whether the lamina location is vacant or filled with a lamina of prescribed orientation. Corresponding genes in the second string determine the lamina material if the lamina is available. By employing two gene strings, the quantity of materials that might be utilized in the stacking succession might be changed easily by adjusting the size of the material gene letters in order [6]. In the application of the two material design issue, with lamina orientation choices of 0º, through 90º with 15º increments, see Figure 7. Lamina thickness may take one of two prescribed genes depending on the material that a lamina is comprised of, as appeared in the example stacking succession of Figure 8.

3.1 Decoding the Gene Strings

To fuse the two material idea into the GA, a complicated decoding technique is required. Orientation genes that are coded as number genes somewhere in the range of 2 and 6 speak to either the positive or the negative estimation of the relating lamina orientation angle characterized in Figure 7. For instance, a 4 speaks to either a +45º or a –45º lamina in the orientation gene. For the issue, laminates will be compelled to have a decent stacking grouping to disentangle examination techniques. To keep up a reasonable laminate or get a laminate as near adjusted as could be expected under the circumstances, the θ utilizes are decoded then again. For instance, the initial 4 (beginning from the external edge of the laminate) experienced for a specific material is decoded as a +45º lamina and the following 4 for a similar material is decoded as a –45º, etc. This interpreting technique applies for all utilizes acknowledge those arranged at 0º and 90º, which are decoded in the ordinary style. In this manner, a laminate stacking succession of θ handles is adjusted if each +θ lamina is coordinated with a -θ , lamina of a similar material. On the off chance, the θ handles are adjusted for one material however not the other, the laminate is unbalanced.

Orientation gene
0 = empty lamina (E)
1 = lamina oriented at 0º
2 = lamina oriented at +/-15º
3 = lamina oriented at +/-30º
4 = lamina oriented at +/-45º
5 = lamina oriented at +/-60º
6 = lamina oriented at +/-75º
7 = lamina oriented at 90º

Material gene
0 = 1 empty lamina (e)
1 = Material #1
2 = Material #2
3 = Material #3

Figure 7: GA-II code key for sample stack.

Coded orientation	[0/2/6/4/7/4/6/1/3/5]
Coded Material	[0/1/2/3/1/3/1/1/3/2]
Decoded orientation	[E/15º/75º/45º/90º/-45º/-75º/0º/30º/60º]
Decoded material	[e/m1/m2/m3/m1/m3/m1/m1/m3/m2]

E	e
15º	m1
-75º	m2
45º	m3
90º	m1
-45º	m3
75º	m1
0º	m1
30º	m3
60º	m2

Figure 8: Sample stacking structure for GA-II.

3.2 Modifications in Genetic Operators

The strategy for the GA-II calculation remains for the most part unaltered from the one utilized in the GA-I form, aside from little alterations made to the genetic administrators [7]. At the point when a parent is chosen for propagation, both the lamina orientation gene string and material gene string are utilized while making a kid. In the crossover strategy, the orientation and material gene strings are part at a similar point in the two guardians. The left bits of both the orientation and material gene strings from parent one and the comparing right pieces from parent 2 are then joined to shape a laminate. At the point when the youngster is made, the lamina coded as a 3 that is passed from parent 2 currently gets decoded as a -30º lamina. This is on the grounds that parent 1 likewise passed a lamina to the kid laminate that was coded as a 3 made of a similar material. Since the 30º lamina from parent 1 will be nearer to the external edge of the laminate, it will be decoded as +30º where as the second 3, which came structure parent 2 gets decoded as -30º. The strategy for the change administrator is adjusted marginally too. Lamina expansion and cancellation are done all the while on both the orientation and material gene strings. Included utilizes are by and by presented at the mid-plane of the laminate. At the point when a lamina is included, the relating material gene is additionally included. At the point when a lamina is erased, it is picked aimlessly with the comparing material gene additionally being erased, see Figure 10(b). Gene change is actualized independently on every gene string, with the equivalent or diverse probabilities. In the event that a gene is modified in the orientation gene, the comparing material genes may not really be adjusted. Besides, when genes are exchanged in the swapping of lamina administrator, both the orientation and material genes are swapped at the same time, see Figure 11. As in crossover, the other genetic administrators may switch the sign on the orientation point of a lamina when they are connected, see for instance in the lamina modification strategy portrayed in Figure 6(c).

GA code

Parent 1		
Orientation	[0/2/6/4	/7/4/6/1/3/5]
Material	[0/1/2/3	/2/1/3/1/3/2]

Parent 2		
Orientation	[0/3/5/4	/3/5/7/1/2/4]
Material	[0/2/3/1	/3/2/1/3/1/2]

Child	
Orientation	[0/2/6/4/3/5/7/1/2/4]
Material	[0/1/2/3/3/2/1/3/1/2]

Decoded format of stack

Parent 1		
Orientation	[E/-15º/75º/45º	/90º/-45º/-75º/0º/30º/60º]
Material	[e/m1/m2/m3	/m2/m1/m3/m1/m3/m2]

Parent 2		
Orientation	[E/30º/60º/45º	/-30º/-60º/90º/0º/15º/-45º]
Material	[e/m2/m3/m1	/m3/m2/m1/m3/m1/m2]

Child	
Orientation	[E/-15º/75º/45º/-30º/-60º/90º/0º/15º/-45º]
Material	[e/m1/m2/m3/m3/m2/m1/m3/m1/m2]

Figure 9: Modified crossover operator

GA code

Before lamina addition	
Orientation	[**0**/2/6/4/3/6/7/1/5/4]
Material	[**0**/1/2/3/2/1/3/1/3/2]

After lamina addition	
Orientation	[2/6/4/3/6/7/1/5/4/**5**]
Material	[1/2/3/2/1/3/1/3/2/**1**]

Decoded format of stack

Before lamina addition	
Orientation	[**E**/15º/75º/45º/30º/-75º/90º/0º/60º/-45º]
Material	[**e**/m1/m2/m3/m2/m1/m3/m1/m3/m2]

After lamina addition	
Orientation	[15º/75º/45º/30º/-75º/90º/0º/60º/-45º/**-60º**]
Material	[m1/m2/m3/m2/m1/m3/m1/m3/m2/**m1**]

(a) Lamina addition

GA code

Before lamina deletion	
Orientation	[0/2/6/4/3/**6**/7/1/5/4]
Material	[0/1/3/2/2/**3**/3/1/3/2]

After lamina deletion	
Orientation	[0/2/6/4/3/**0**/7/1/5/4]
Material	[0/1/3/2/2/**0**/3/1/3/2]

Restack	
Orientation	[0/**0**/2/6/4/3/7/1/5/4]
Material	[0/**0**/1/3/2/2/3/1/3/2]

Decoded format of stack

Before lamina deletion	
Orientation	[E/15º/-75º/45º/30º/**75º**/90º/0º/60º/-45º]
Material	**[**e/m1/m3/m2/m2/**m3**/m3/m1/m3/m2]

After lamina deletion	
Orientation	[E/15º/-75º/45º/30º/**E**/90º/0º/60º/-45º]
Material	[e/m1/m3/m2/m2/**e**/m3/m1/m3/m2]

Restack	
Orientation	[E/**E**/15º/-75º/45º/30º/90º/0º/60º/-45º]
Material	[e/**e**/m1/m3/m2/m2/m3/m1/m3/m2]

(b) Lamina deletion

Figure 10: Modified mutation operator

GA code

Before lamina swap	
Orientation	[0/**2**/6/4/3/6/7/1/**5**/4]
Material	[0/1/**2**/3/2/1/3/1/**3**/2]
After lamina swap	
Orientation	[0/**5**/6/4/3/6/7/1/**2**/4]
Material	[0/1/**3**/3/2/1/3/1/**2**/2]

Decoded format

Before lamina swap	
Orientation	[E/**15º**/75º/45º/30º/-75º/90º/0º/**60º**/-45º]
Material	[e/m1/**m2**/m3/m2/m1/m3/m1/**m3**/m2]
After lamina swap	
Orientation	[E/**60º**/75º/45º/30º/-75º/90º/0º/**15º**/-45º]
Material	[e/m1/**m3**/m3/m2/m1/m3/m1/**m2**/m2]

Figure 11: Modified lamina swap operator

3.3 Selection of Stopping Criterion

The main issue is the stopping criterion for the genetic algorithm. The genetic pursuit might be halted after a recommended number of cycles with no improvement of the top structure in the populace. This stopping criterion is appropriate for estimating the hunt if a push to improve the productivity of the GA is being made. A less difficult stopping criterion is to utilize an upper bound on the absolute number of capacity assessments directed by the GA. The second stopping criterion might be favoured when leading a number of free searches, and improves the statistics of the measurements of the GA toward the finish of the seek since every optimization run will have a similar number of ages.

3.3 Fitness Evaluation

The weight of the laminate can be verifiably or unequivocally characterized in the target work for a laminate. In spite of the fact that the most slender laminates will yield the best execution, they are vigorously punished if the material flops under the given stacking condition [8]. Accordingly, the laminates that yield the best execution without coming up short the material gene requirement will consequently be the lightest (i.e., have the least number of employs). For the multi-target advancement issue exhibited in Paper 6, two target capacities will be used. The main target capacity will unequivocally contain the weight of the laminate by checking the all out number of utilizes in the stacking grouping. A second capacity will contain data about the assembling and material expense of the laminate. The physical weight and cost the of the laminate are then balanced utilizing data relating to the clasping and gene imperative fulfilment of the laminate [9]. The target capacities are then scaled by the relating target elements of an ostensible plan to guarantee that the expense and weight of the laminate are spoken to in like manner. The general wellness of the laminate is gotten as a curved blend of the two target capacities. The curved mix would then be able to be changed in accordance with enable expense and weight to add to the wellness estimation in any ideal way [10].

IV. CONCLUSION

The genetic algorithm technique do not calculate the numerical value for a given parameter selected. Instead, it selects the best solution from the given range for a parameter to be optimized. The range of values for that parameter are defined based on some assumptions or experimental test, and other engineering constraints based on their feasibility. For laminate optimization, generally weight or strength or material costs are considered as the objective functions which are to be improved. The optimization can be single objective or multi-objective. The fitness value for those parameters determines the best feasible solution for the objective function. In other words, this also means where the convergence criterial is fulfilled, the point of convergence gives the optimal solution for that objective function.

REFERENCES

[1] Alliot, J., Gruber, H., Joly, G., “Genetic Algorithms for Solving Air Traffic Control Conflicts." Proceeding of the 9th Conference on AI Applications, 1993.

[2] Davis, L., “Job Shop Scheduling with Genetic Algorithms." Proceedings of the First International Conference on Genetic Algorithms, 1985, pp. 136-140.

[3] Giffer, B. and Thompson, G. L., “Algorithms for Solving Production Scheduling Problems." Operations Research, Vol. 8, 1969, pp. 90-99.

[4] Jones, R.M., “Mechanics of Composite Materials." Hemisphere, New York, 1975.

[5] Holland, J. H., “Adaptation in Natural and Artificial Systems." The University of Michigan Press, Ann Arbor, MI, 1975.

[6]. Goldberg, D. E., “Genetic Algorithms in Search, Optimization, and Machine Learning." Addison-Wesley Publishing Company, Inc., Reading, MA, 1989

[7] Le Riche, R., “Optimization of Composite Structures by Genetic Algorithm." Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1994.

[8] Le Riche, R., Haftka R. T., Optimization of Stacking Sequence Design for Buckling Load maximization by Genetic Algorithm." Proceedings of the 33rd AIAA/ASME/ASCE/AHS/ASC SDM Conference, Dallas, TX, 1992.

[9] Orvosh D. and Davis L., “Shall We Repair? Genetic Algorithms, Combinatorial Optimization, and Feasibility Constraints." University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers, 1993, pp. 650.

[10] Nagendra, S., Haftka, R. T., “Stacking Sequence Optimization of Simlamina Supported Laminates with Stability and Strain Constraints. " AIAA Journal, Vol. 30, No. 8, 1992, pp. 2132 (2137).