IMPROVED GENETIC ALGORITHM FOR THE DESIGN OF STIFFENED COMPOSITE PANELS

S. NAGENDRA*, D. JESTIN+, Z. GÜRDAL‡, R. T. HAFTKA†, and L.T. WATSON**

Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061, U.S.A.

ABSTRACT

The design of composite structures against buckling presents two major challenges to the designer. First, the problem of laminate stacking sequence design is discrete in nature, involving a small set of fiber orientations, which complicates the solution process. Therefore, the design of the stacking sequence is a combinatorial optimization problem which is suitable for genetic algorithms. Second, many local optima with comparable performance may be found. Most optimization algorithms find only a single optimum, while often a designer would want to obtain all the local optima with performance close to the global optimum. Genetic algorithms can easily find many near optimal solutions. However, they usually require very large computational costs. Previous work by the authors on the use of genetic algorithms for designing stiffened composite panels revealed both the above strength and weakness of the genetic algorithm. The present paper suggests several changes to the basic genetic algorithm developed previously, and demonstrates reduced computational cost and increased reliability of the algorithm due to these changes. Additionally, for a stiffened composite panel used in this study, designs lighter by about 4 percent compared to previous results were obtained.

INTRODUCTION

The growing use of high performance composite materials has motivated the development of optimization procedures for the design of laminates. Due to their high strength, composite plates are usually thin laminates that are buckling critical. Thus optimization of composite plates to maximize buckling loads has drawn considerable attention in recent years (e.g., [1–4]).

The design of composite laminates is often formulated as a continuous optimization problem with ply thicknesses and ply orientation angles used as design variables (e.g., Schmit and Farshi, [1]). However, for many practical problems, ply thicknesses are fixed and ply orientation angles are limited to a small set of angles such as 0, 90, and 45 degrees. Designing the laminate then becomes a stacking sequence optimization which can be formulated as an integer programming problem. In

---

[*] Research Associate, Engineering Science and Mechanics, Member AIAA.
[+] Student, University of Technology of Compiègne (FRANCE), currently visiting VPI&SU.
[‡] Associate Professor, Engineering Science and Mechanics, Associate Fellow AIAA.
[†] Christopher Kraft Professor of Aerospace and Ocean Engineering, Associate Fellow AIAA.
[**] Professor, Departments of Computer Science and Mathematics.

Paper presented at the 15th Intl. Symp. on Math. Prog., Ann Arbor, Aug. 15–19, 1994.

recent years there has been growing interest in the application of genetic algorithms to this integer programming problem.

Applications of genetic algorithms to the optimum design of nonconvex engineering problems can be found in several recent studies [e.g., 5-9]. In the area of composite structural design, recent applications of genetic algorithms include Callahan and Weeks [10], and Ball and Sargent [11]. Le Riche and Haftka [12] solved the laminate stacking sequence design problem subject to buckling and strength constraints. Nagendra et al. [13, 14] applied the same approach to the design of stiffened composite panels. In both works [12, 14] it was found that the genetic algorithm can find a variety of alternative designs with similar performance, thus giving the designer a choice of alternatives. However, the cost of the genetic search was high, requiring thousands of analyses.

Kogiso et al. [15, 16] considered the use of information about past designs (memory) for the purpose of reducing the number of analyses required by the genetic algorithm. The information was used to construct a linear approximation to the buckling load in the neighborhood of each member of the population of designs. The approximations were then used to seek nearby improved designs. Substantial reductions in the number of required analyses for designing unstiffened laminates were achieved. However, the procedure could reduce the chance of locating isolated singular optima. Le Riche and Haftka [17] have found that substantial reduction in cost of the GA could be achieved by tailoring GA operators to the special features of stacking sequence design. The present paper similarly investigates the effect of various modifications to the basic genetic algorithm for the minimum weight design of stiffened panels subject to stability and strain constraints. The primary objective is to improve reliability and reduce the computational cost of the algorithm.

PROBLEM DESCRIPTION

The panel under consideration is 30 in long and 32 in wide and has four equally spaced blade stiffeners (see Figure 1). The panel is designed for an axial load (N_{x} = 20,000 lb/in) and a shear load (N_{xy} = 5,000 lb/in). The small shear load was added to reduce sensitivity to off-design conditions. The skin, stiffener blade, and stiffener flange laminates are constrained to be balanced and symmetric laminates made up of 0, ±45, and 90 degree plies. The stiffener blade and flange (see Figure 2) were constrained to have identical laminates. Furthermore, the number of adjacent (contiguous) plies of the same orientation is restricted to be four or less in order to prevent matrix cracking. Taking damage tolerance aspects under compressive loading into consideration, the outer plies of the skin and stiffener blade laminates are restricted to be ±45 degree plies. To account for shape imperfections and eccentricities, it is assumed that the panel could have a longitudinal bow of 3 % of the panel length.

In reference [18], strain and buckling critical stiff-skin designs (panel designs with 0^o plies in the skin laminate) and soft-skin designs (panel designs with no zero-degree plies in the skin laminate) were identified. The soft-skin designs are expected to be better designs as the strain margin is higher than for the stiff-skin designs. In this paper stiffened composite panels are designed by genetic algorithms using the PASCO program [19,20] for analysis purposes only (this is equivalent to the use of the analysis module VIPASA [21] of PASCO).

OPTIMIZATION FORMULATION

The goal of the optimization is to find the stacking sequence of minimum weight panels that will not buckle, or fail due to excessive strains for a particular set of design loads and boundary conditions. Additionally, ply contiguity constraints which require that the designs not have more than 4 contiguous plies of the same orientation are imposed so as to reduce the danger of matrix cracking.

The panel weight W is a function of the number of layers in the skin n_s, number of layers in the stiffener blade n_b, and the planform areas A_s, A_b, and A_f of the skin, blade, and flange, respectively. Thus, the panel weight (with 4 equal stiffeners) can be calculated as

$$\mathrm{W} = t[n_s\ A_s + 4\ (n_b\ A_b + 2\ n_b\ A_f)], \tag{1}$$

where t is the ply thickness given in Table 1.

The optimization problem can be formulated as finding the stacking sequences of the panel (i.e., ply orientation θ_i of the i-th ply) of the skin, and of the stiffener-blade laminates, as well as the stiffener height H_b in order to minimize the weight W of the panel :

$$\begin{gathered}\text{Minimize}\quad \mathrm{W}(\mathrm{H}_b, n_s, n_b)\\ \text{Subject to}\\ \lambda_b \geq 1,\\ \lambda_s \geq 1,\\ g(\hat{\theta}) \leq 0,\end{gathered} \tag{2}$$

where λ_b is the critical buckling load factor and λ_s is the strength failure load factor obtained as the smallest (over all strain components) ratio of the strain allowable ϵ_{ij}^{all} to the corresponding strain component ϵ_{ij}. $g(\hat{\theta})$ represent the contiguous ply limit and the balanced condition. The vector $\hat{\theta}$ of ply orientations has $n_s + n_b$ components. In order to reduce the number of design variables and ensure balanced laminates, laminates are assumed to be composed of stacks with each stack made up of two plies.

The constrained optimization problem must be transformed into an unconstrained problem for the genetic algorithm. This is done by using penalty parameters and defining an objective function F as

$$\mathrm{F} = \begin{cases} \left(\dfrac{\mathrm{W}}{\lambda_{\text{fail}}^{q}}\right)\ \mathrm{P}_{\text{cont}}^2, & \lambda_{\text{fail}} < 1\ , \\ \\ (\mathrm{W} - \epsilon\lambda_{\text{fail}})\ \mathrm{P}_{\text{cont}}^2, & \lambda_{\text{fail}} \geq 1\ , \end{cases} \tag{3}$$

where the penalty parameter, P_{cont}, is used to enforce contiguity constraints and λ_{fail} is the failure load factor defined as

$$\lambda_{\text{fail}} = \min\{\lambda_b, \lambda_s\}. \tag{4}$$

In Eq. 3, the term W/λ_{fail} ensures that the objective function is penalized for designs that do not satisfy failure constraint. The penalty parameter q is problem dependent and a value of q = 0.9 was employed here. When the failure constraint is satisfied ($\lambda_{\text{fail}} \geq 1$, see Eq. 3) a small fraction ($\epsilon = 0.001$) of the critical failure load factor λ_{fail} is subtracted from the weight, to discriminate between multiple designs with the same weight all of which satisfy the failure constraint. The best design has the largest possible failure margin of all panels of the same weight.

The ply contiguity constraint is imposed by modifying the objective function with a penalty parameter P_{cont}, which is calculated as a function of the degree of constraint violations. The ply contiguity parameter P_{cont} is calculated as the degree of constraint violation measured by the variables nv_{sk} and nv_{bl}, which are defined to be the number of contiguous 0_2 or 90_2 degree stacks in excess of the constraint value of 2 in the skin and blade laminates, respectively. For example, the number of violations for a $[90_8/0_2]_s$ laminate is 2. Then, P_{cont} is calculated as

$$P_{\text{cont}} = 1 + \frac{(nv_{\text{sk}} + nv_{\text{bl}})}{10} \; . \tag{5}$$

Because of the laminate mid-plane symmetry the contiguity constraint can be violated by having more than one stack of 0_2 or 90_2 degree plies near the midplane. Thus for the laminate $[0_6, 90_4]_s$ the number of violations is 2. One violation is counted for the 90 degree stacks, since there are two stacks at the midplane, and an additional violation is counted for the three 0_2 stacks.

GENETIC CODE

The design variables are the blade height, number of plies, and ply orientations restricted to a discrete set of angles (0, ± 45, and 90 degrees). Three possible stacks (see Figure 3) are 0_2, ± 45, and 90_2, and are represented by the codes 1, 2, and 3 respectively. Since the total laminate thickness is not fixed, but the length of the string used to represent laminates is fixed, a string length corresponding to an upper bound on the thickness is used here. The length of the string needs to be such that at least one feasible design is guaranteed for the specified loads. The upper bound for the skin laminate was chosen as 15 stacks for the present study, while for the blade laminate it was chosen as 25 stacks. Therefore, to represent laminates with thicknesses smaller than the upper bound, a code of '0' is introduced to identify empty stacks. For example, using a string length of 10 (total number of possible plies = 40), the 32-ply symmetric laminate $[\,*/*/90_2/\pm 45_2/90_2/0_2/\pm 45_2/0_2]_s$ is encoded as $\lfloor_m$ 1 2 2 1 3 2 2 3 0 0, where '*' denotes an empty stack. The leftmost 1 corresponds to the 0_2 stack closest to the laminate plane of symmetry (denoted by $\lfloor_m$). The rightmost 3 describes the outermost 90_2 stack. The zeros on the right correspond to empty stacks. During the optimization process, zeros representing empty stacks can be generated in the interior of a laminate design. This problem is resolved by packing the string to move all zeros toward the outside of the laminate without altering the order of occurrence of the filled stacks.

The blade height is a continuous variable approximated by a ternary rational number. The blade height variable is modelled using alleles 0, 1, and 2. During genetic operations, if a 0 is present in the interior of the blade height variable, no packing is done on that portion of the string and hence the order of the genes in the blade height string is not altered. The string length L_h of the blade height array is based on the desired precision of the approximation. Given a ternary string $[Y_{\mathrm{L_h}}, ..., Y_2, Y_1]$, the corresponding blade height $\mathrm{H_b}$ is given as

$$\mathrm{H_b} = \left[\frac{(X_{\mathrm{u}} - X_{\mathrm{l}})}{2\sum_{i=1}^{\mathrm{L_h}} 3^{i-1}} \right] \left[\sum_{i=1}^{\mathrm{L_h}} (Y_{\mathrm{i}}) 3^{i-1} \right] + X_{\mathrm{l}}, \tag{6}$$

where X_{u} and X_{l} are the upper and lower bounds of the range of the continuous variable $\mathrm{H_b}$. Based on the continuous optimum design obtained in [18], the stiffener blade height was assumed to be in the range $1.5 \leq \mathrm{H_b} \leq 3.3$ inch. The blade height H_b was discretized as a 9 digit ternary number representing 3^9 or about 20,000 values in the range of 1.5 – 3.3 inch.

A design (a member in the population) is formed by concatenating the blade height substring with the skin and blade laminate substrings. This is identified as a complete design string. With string lengths of 15 and 25 chosen for the skin and blade laminates respectively and the string length of 9 for the blade height the total string length is 49. Since the skin laminate does not include any zero plies and the outermost stack has to be a ±45 stack in the skin and blade laminates, the possible number of designs is 3^{15-1} in the skin and 4^{25-1} in the blade. Hence the complete set of possible designs is $3^9 * 3^{14} * 4^{24} \approx 2.65 \times 10^{25}$.

IMPLEMENTATION OF GENETIC ALGORITHM

The implementation of the genetic algorithm is shown schematically in Figure 4. The process starts with the generation of a random initial population of n_{d} designs. Next, the PASCO preprocessor translates the genetic string of each design into input for PASCO. PASCO evaluates the panel mass, buckling loads, strains and loads in different sections of the panel. Postprocessing of the PASCO output assigns the information regarding the panel mass, buckling loads and strains to individual parent strings in the population base. This procedure is repeated for each string in the population. The fitness processor evaluates the objective function for each design and ranks the designs. The evaluated population is processed then, by means of genetic operators, to create a new population which combines the most desirable characteristics of the old population. The old population is replaced by the new one except for the best design which is always kept unchanged – an "elitist" version of the genetic algorithm [5, 22]. This scheme guarantees that the best design will occur in the final population. The process is repeated until convergence, defined as a maximum number of generations without improvement in the best design, or a fixed number of generations chosen a priori.

The genetic algorithm begins with the random generation of a population of design alternatives. To ensure that a large number of initial designs are feasible in the initial population, the probability of choosing an empty stack denoted by '0' is forced to be small in the initial population. The selection process is biased so that high performance designs have a higher probability of transmitting their features to the next generation. This is implemented by allocating each individual a portion of a roulette wheel in proportion to their fitness. The fitness was chosen to be the rank in terms of a design's objective function in the population. The fitness (fraction of a simulated roulette wheel assigned to the design) assigned to the i-th best individual of n_d designs is then equal to $2(n_\mathrm{d} + 1 - i)/({n_\mathrm{d}}^2 + n_\mathrm{d})$.

Once parents are selected various genetic operators are applied to create child designs. The genetic algorithm of [13] called here the "basic GA" includes the operators of crossover, mutation and permutation, described in the next section. To improve the performance of the algorithm, these operators were modified, as described in the next section. Additionally, the selection of two identical parents was not allowed. The algorithm based on these modified operators is called herein as the "improved GA".

GENETIC OPERATORS

CROSSOVER: Crossover allows selected individuals to trade characteristics of their designs by exchanging parts of strings. A two-point crossover, where two break points in the string are chosen at random, is applied to the complete design string. Two offspring are created by swapping the parent substrings. The crossover operator is applied with a given probability P_c, chosen here to be 0.95. If crossover is not applied, the first parent is copied into the next generation. The following is an example where a $[*_3, \pm 45_4, 90_4, \pm 45_2, 90_2, \pm 45_2, 90_2]_s$ laminate is mated with a $[*_6, \pm 45_3, 90_4, \pm 45_2, 90_4]_s$ laminate to produce $[*_3, \pm 45_4, 90_2, \pm 45, 90_4, \pm 45_3, 90_2]_s$ and $[*_6, \pm 45_2, 90_2, \pm 45_2, 90_2, \pm 45, 90_4,]_s$ laminates.

$$\begin{array}{rl}
\text{Parent 1} & \lfloor_m 3\ 2\ /\ 2\ 3\ 2\ 2\ 3\ /\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0. \\
\text{Parent 2} & \lfloor_m 3\ 3\ /\ 2\ 2\ 3\ 3\ 2\ /\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0. \\
\text{Child 1} & \lfloor_m 3\ 2\ /\ 2\ 2\ 3\ 3\ 2\ /\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0. \\
\text{Child 2} & \lfloor_m 3\ 3\ /\ 2\ 3\ 2\ 2\ 3\ /\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0.
\end{array} \tag{7}$$

SUBSTRING CROSSOVER: To improve the performance of the genetic algorithm, we tested a crossover applied to each substring representing the skin, the blade laminate, and the stiffener height separately. Thus in comparison with the traditional crossover, the substring crossover is a eight point crossover instead of a two-point crossover.

MUTATION: Mutation introduces small changes in children produced by crossover. One of the two children is randomly picked, and endures mutation with a low probability $\mathrm{P_m}$ (here 0.03 per gene in an individual parent).

$$\begin{array}{rl}
\text{Before mutation} & \lfloor_m 2\ 2\ 2\ 3\ 2\ \underline{2}\ 3\ 3\ 0\ 0\ 0\ 0\ 0\ 0\ 0. \\
\text{After mutation} & \lfloor_m 2\ 2\ 2\ 3\ 2\ \underline{0}\ 3\ 3\ 0\ 0\ 0\ 0\ 0\ 0\ 0.
\end{array} \tag{8}$$

NEW MUTATION: The mutation procedure can change ply orientation as well as delete or add a stack. The probabilities of the three events are linked and depend on the number 0's (empty stacks) in the string. To improve the performance we also considered three independent mutations of *stack deletion*, *stack addition*, and *orientation mutation.* The primary advantage of the new procedure was to separate and independently control the individual substring rates of mutation. The blade height mutation is not changed.

STACK DELETION: First the skin or blade laminate is chosen at random for stack deletion. Then the stack closest to the midplane is deleted with probability $\mathrm{P_{del}}$. The choice of the stack closest to the midplane to be deleted is based on the fact that these stacks have little influence on the buckling load and so are more likely to be non-optimal than an outer stack.

$$\begin{array}{rl} \text{before stack deletion} & \lfloor_m 2\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{after stack deletion} & \lfloor_m 0\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{after string packing} & \lfloor_m 3\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0. \end{array} \tag{9}$$

STACK ADDITION: The skin or blade laminate is chosen at random for stack addition, then a stack is added at the midplane with probability $\mathrm{P_{add}}$ and all the other stacks shifted by a stack away from the midplane without altering the order of the stacks.

$$\begin{array}{rl} \text{before stack addition} & \lfloor_m 2\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{after stack addition} & \lfloor_m 3\ 2\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0. \end{array} \tag{10}$$

ORIENTATION MUTATION: The skin or blade laminate is chosen at random for orientation change, then a random stack is chosen and the stack orientation is changed to a new orientation with probability $\mathrm{P_{om}}$. A change in the stack orientation is allowed anywhere in the filled part of the laminate substring.

$$\begin{array}{rl} \text{before orientation mutation} & \lfloor_m 2\ 3\ 2\ 2\ \underline{3}\ 3\ 2\ 2\ 2\ 2. \\ \text{after orientation mutation} & \lfloor_m 2\ 3\ 2\ 2\ \underline{2}\ 3\ 2\ 2\ 2\ 2. \end{array} \tag{11}$$

PERMUTATION: Permutation, a new stochastic operator devised for laminate design [15], inverts the order of the genes between two randomly assigned points.

$$\begin{array}{rl} \textit{Before permutation} & \lfloor_m 1\ 1\ /\ 2\ 2\ 2\ 1\ 3\ /\ 2\ 3\ 2\ 1\ 3\ 0\ 0\ 0. \\ \textit{After permutation} & \lfloor_m 1\ 1\ /\ 3\ 1\ 2\ 2\ 2\ /\ 2\ 3\ 2\ 1\ 3\ 0\ 0\ 0. \end{array} \tag{12}$$

In the above example, the laminate described by the pair of strings before permutation is $[*_3, 90_2, 0_2, \pm45, 90_2, \pm45, 90_2, 0_2, \pm45_3, 0_4]_s$ and becomes $[*_3, 90_2, 0_2, \pm45, 90_2, \pm45_4, 0_2, 90_2, 0_4]_s$ after permutation. The permutation operator introduces drastic changes in the laminate stacking sequence and the new design can be feasible or infeasible by a wide margin. In order to introduce a change in the laminate stacking sequence that is less dramatic, Kogiso et al. [15] introduced an interchange operator which swaps two stacks. Here we differentiate between intra- and interlaminate swap.

INTRALAMINATE SWAP: The intralaminate swap operator is applied with a probability of $\mathrm{P_{ils}}$ either to the skin laminate or to the blade laminate with the laminate chosen at random (never to both). Two stacks are selected at random within a substring and swapped by interchanging the stacks with respect to their location in the string. The intralaminate swap operator is applied to the complete laminate substring and admits the swapping of a filled stack with an empty stack, which allows the stack to be moved towards the outside of the laminate. The swapping of two empty stacks is not allowed. For example in the skin laminate :

$$\begin{array}{rl} \text{before intralaminate swap} & \lfloor_m 2\ \underline{3}\ 2\ 2\ 3\ 3\ 2\ \underline{2}\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{after intralaminate swap} & \lfloor_m 2\ \underline{2}\ 2\ 2\ 3\ 3\ 2\ \underline{3}\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \end{array} \tag{13}$$

INTERLAMINATE SWAP: The interlaminate swap operator exchanges a stack between the skin and the blade laminates. This particular operator is the only operator which exchanges genetic information between the skin and blade laminates. The interlaminate swap operator is applied with a probability of $\mathrm{P}_{\mathrm{ilsw}}$. A stack in the skin laminate and a stack in the blade laminate are chosen at random. If the chosen stack in the blade is not a 0_2 stack then the two stacks are swapped, otherwise no swapping occurs. This check ensures that the skin does not get altered from a soft-skin to stiff-skin design during the course of the search.

$$\begin{array}{rl} \text{before interlaminate swap [SKIN]} & \lfloor_m 2\ \underline{3}\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{before interlaminate Swap [BLADE]} & \lfloor_m 2\ 2\ 1\ 1\ 3\ 3\ 2\ 3\ 3\ 2\ \underline{2}\ 2\ 1\ 1\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0. \\ \text{after interlaminate swap [SKIN]} & \lfloor_m 2\ \underline{2}\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0. \\ \text{after interlaminate swap [BLADE]} & \lfloor_m 2\ 2\ 1\ 1\ 3\ 3\ 2\ 3\ 3\ 2\ \underline{3}\ 2\ 1\ 1\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0. \end{array} \tag{14}$$

POPULATION DIVERSITY

Maintaining diversity of individuals within a population is necessary for the long term success of any evolutionary process. Genetic diversity helps a population adapt quickly to changes in the environment and allows the population to continue searching for productive niches, avoiding premature convergence at local optima [22–25]. In GAs it is difficult to maintain diversity because the algorithm assigns an exponentially increasing numbers of trials (for an infinite population) to the observed best parts of the search space (Schema Theorem, Holland [26]). As a result the standard GA has strong convergence properties which may restrict the design space and lead to premature convergence at a local optimum [25]. Hence in order to ascertain whether the population is diverse during each generation of the search, a *heterogeneity measure* based on the individual members of the population and their relation to each other in terms of gene composition is devised.

The heterogeneity measure H_{m} is calculated by comparing individual members of the population of size n_{d} on a 'gene by gene' basis and counting the number of nonidentical genes at the same location in the strings. That is

$$H_{\mathrm{m}} = \frac{2 \sum\limits_{i=1}^{n_{\mathrm{d}}-1} \sum\limits_{j=i+1}^{n_{\mathrm{d}}} S_{\mathrm{ij}}}{n_{\mathrm{d}} * (n_{\mathrm{d}} - 1) * \mathrm{L}} , \tag{15}$$

where the number of genes that are different between designs i and j is denoted by S_{ij}. For a random population the ratio of S_{ij} to the string length L averages $(m-1)/m$, where m is the number of alleles (in this case 3 or 4).

At the beginning of the genetic search, the population is diverse and the heterogeneity measure H_m is about 0.5, while towards the end of the search it becomes low as the population indicates an increased tendency to become homogeneous.

ALGORITHM PERFORMANCE

The GA is a stochastic process and hence the algorithm performance has to be evaluated based on performance criteria that take into account the algorithm's reliability in reaching the *best* solution. Here we assume that feasible designs with weights within one or two percent of the best known designs are acceptable, and we denote them as practical optima.

To account for the variability inherent in a stochastic algorithm we test the algorithm by repeating its execution a number of times with different seeds for the random number generator. The reliability R is then defined as the fraction of runs which produced a practical optimum. In comparing different GAs, premature convergence can lower both the reliability and the cost of the genetic algorithm. We can normalize these variations by estimating the number of analyses, N_{T}, required to reach a target reliability T as shown from [27]

$$N_{\mathrm{T}} = N_{\mathrm{R}} \left[\frac{\ln(1-T)}{\ln(1-R)} \right]. \tag{16}$$

RESULTS

To reduce computational cost, the probabilities associated with the various genetic operators were tuned on a simplified problem. The simplified problem had buckling and ply contiguity constraints only (no strain constraints), the blade height fixed at 3.2121 in., the blade laminate set to $[\pm 45/(\pm 45_2/0_4)_2/(\pm 45/0_4)_2/\pm 45/0_2]_S$, and did not include bow. With the stacking sequence of the skin laminate being the only design variable, the design space was reduced from 2.65×10^{25} to approximately 4.78×10^{6}. Results were obtained for graphite-epoxy stiffened panels with AS4/3502 manufacturer specified properties shown in Table 1.

For this simplified problem, the optimum skin design was obtained as $[\pm 45/90_4/\pm 45/90_2/\pm 45_3]_s$ with a mass of 22.68 lbs. The basic GA obtained this design with a reliability of 80–85 percent after 80 generations (800 analyses), depending on the type of random number generator used (see the Appendix for a discussion of random number generators). Note that for this simplified problem the reliability is based only on the designs that achieved the optimum weight of 22.68 lbs, since the next heaviest design of 23.82 lbs lies outside the one or two percent tolerance used for defining practical optima. Increasing the design space by freeing the blade laminate yielded the best known design of 22.68 lbs although reliability was poor. The poor reliability of the basic GA motivated the proposed modified genetic operators to improve the reliability of the algorithm.

TUNING THE IMPROVED GA: The probabilities of the new genetic operators were tuned to obtain optimal performance of the algorithm. Tuning was carried out using a population size n_d of 10, and 10 individual optimization runs of the genetic algorithm was performed for each combination of the various probabilities. The blade height was fixed at 3.2121 in., and the stopping criterion was 40 generations without improvement. The algorithm was assumed to be tuned when it produced good reliability in obtaining the 22.68 lbs with 8 stacks (32 plies) in the skin and 17 stacks (68 plies) in the blade. The results of some of these trials are summarized in Table 2. Because of the stopping criterion, both the average number of analyses, N_{R}, and the reliability changed. To normalize the results, the number of analyses needed to obtain 50 percent reliability, N_{50}, was calculated using Eq. 16. However, because of the small number of optimizations, the results are just a rough indication of the effect of the parameters.

Keeping the mutation rates due to addition, deletion, and orientation change small (1%) and having high rates for crossover (95%), intralaminate swap (75%), and interlaminate swap (55%), reliabilities as high as 70% with an average number of analyses of 1341 could be attained. It was observed that if the deletion probability was increased (keeping addition and stack swaps constant) from 1% to 30% the reliability decreased from 70% to 10%. This is attributed to the fact that if a large number of plies are deleted then the algorithm exhibits premature convergence. The best set of rates was tested further with 70 independent runs giving an average reliability of about 60% with an average number of analyses of 1233. This corresponds to a predicted N_{50} of 958 versus 772 based on only 10 optimizations. The substantial difference shows that the parameter tuning is not very precise.

The reliability of finding the best design in the population as a function of the generation number is plotted in Figure 5a, where 40 genetic optimizations were used to calculate each point on the curve. As the number of generations increases, the reliability of the algorithm increased reaching about 65% reliability in 160 generations. The corresponding average mass of the best design is plotted as a function of the generation number in Figure 5b. It is observed that the average mass of the best design quickly converges to about 22.80 in about 60 generations.

COMPLETE STRING DESIGNS: Next the design space was increased further to allow the blade height (a continuous variable) to change, so that the entire design space was considered. The panel was still assumed to have no bow and no strain constraints were applied. Designs obtained in 10 genetic optimizations are shown in Table 3. The introduction of the blade height allows a smooth variation of weight. The best design had a mass of 22.77 lbs; the blade height increased by 1.18% and the buckling load factor increased by 0.14% compared to the best soft-skin design with a fixed blade height. Using a 1% weight tolerance, 3 of the 10 designs are practical optima, and with a 2% tolerance 6 designs were found to be practical optima. Clearly, increasing the design space made the problem more difficult as the optimal design of 22.68 lbs with fixed blade height could not be found.

FULL PROBLEM TESTS: The complete design problem (Eqs. 2, 3, and 4) was considered next. As in [16], a population size of 20 was selected and the best probabilities (see Table 2) for the tuned algorithm were used. The improved algorithm was run with a fixed number of 150 generations for 40 optimizations. Improvements to the genetic operators were studied by replacing one of the new operators with its old version while keeping the others unchanged, and comparing the results with the basic genetic algorithm. The test runs with such variations were made by using 10 optimizations. First the traditional crossover (applied on the complete string) was used while keeping all the other genetic operators identical to those in the improved GA. Next, the effect of replacing the mutation operator with the old (single) mutation operator, replacing the intralaminate swap by the basic permutation operator, and eliminating the interlaminate swap were tested. The results are summarized in Table 4 and show that in all cases the improvements help the GA to increase reliability.

The effect of perturbations in the rates of individual mutations are shown in Table 5. It is observed that increasing deletion and orientation mutation probabilities up to 2% improves the reliability of the improved GA. However, increasing the orientation mutation probability while keeping deletion probability fixed at 2% reduces the practical reliability, as does increasing deletion probability to 3% while keeping orientation mutation probability at 1%. In comparison with the old mutation procedure the deletion, addition, and orientation mutations perform consistently better.

THE PAYOFF: The payoff of the improved GA can be seen by comparing the designs obtained with it to those obtained in [16] with the basic GA. Table 6 shows the designs (obtained with the basic GA) along with the continuous optimum obtained from [23]. Table 7 shows practical optimal designs obtained by the improved GA. In [16] it was observed that the GA improved upon the continuous rounded optimum by reducing the weight from 26.08 lbs. to 25.19 lbs. The improved GA pushed the weight further to 24.14 lbs. Tables 6 and 7 also illustrate the wealth of design alternatives that can be generated by GAs. The skin laminate varied from 32 to 40 plies, the blade laminate from 64 to 80 plies, and the blade height from 2.86 in. to 3.29 in.

The heterogeneity measure H_{m} is plotted as a function of generation number in Figure 6. It can be observed that the population has a large amount of diversity (H_{m} =0.5) initially and slowly becomes homogeneous. The reliability of the algorithm for 40 optimization runs is shown in Figure 7. It can be observed that practical optima that satisfy the 2% practical reliability requirement are easily found compared to the tighter 1% practical reliability condition.

CONCLUDING REMARKS

Several improvements in a genetic algorithm (GA) procedure for designing stiffened composite panels were developed. The improvements were systematically tested by performing hundreds of genetic optimizations. The improved genetic algorithm reduced the weight of a previously optimized panel by about 4%. Additionally the GA procedure produced a wealth of near optimal designs.

ACKNOWLEDGEMENT

This research was supported in part by NASA grants NAG-1-168 and NAG-1-643.

REFERENCES

(1.) Schmit, L.A., and Farshi, B., "Optimum Design of Laminated Fibre Composite Plates," *International Journal for Numerical Methods in Engineering,* 11, No. 4, 1979, pp. 623 - 640.

(2.) Hirano, Y., "Optimum Design of Laminated Plates Under Axial Compression," *AIAA Journal,* 17, No. 9, November 1979, pp.1017 - 1019.

(3.) Adali, S., and Duffy, K.J., "Design of Antisymmetric Hybrid Laminates for Maximum Buckling Load: I. Optimal Fibre Orientation. II. Optimal Layer Thickness," *Composite Structures,* 14, 1990, No.1, pp. 49-60, and No.2, pp. 113-124.

(4.) Hajela, P., and Shih, C.-J., "Optimal Design of Laminated Composites Using a Modified Mixed Integer and Discrete Programming Algorithm," *Computers and Structures,* Vol. 32, No. 1, 1989, pp. 213-221.

(5.) Goldberg, D.E., **Genetic Algorithms in Search, Optimization, and Machine Learning**, Addison-Wesley Publishing Company, Inc., Reading, MA. 1989.

(6.) Hajela, P., "Genetic Search-An Approach To The Nonconvex Optimization Problem", *AIAA Journal*, Vol. 26, No. 7, 1990, pp. 1205-1210.

(7.) Rao, S.S., Pan, T.S., and Venkayya, V.B., "Optimal Placement of Actuators In Actively Controlled Structures Using Genetic Algorithms", *AIAA Journal*, Vol. 29, No. 6, June 1991, pp. 942-943.

(8.) Underbrink, Jr., A.J., and Williams, Jr., G.P.W., "Genetic Algorithms Applied to Assembly Operations", Proc. of the 1994 SPIE Conference KB AI Systems in Aerospace and Industry, SPIE Code Number 2244-04.

(9.) Forrest, S., Javornik, B., Smith, R.E., and Perelson, A.S., "Using Genetic Algorithms to Explore Pattern Recognition in the Immune System," *Evolutionary Computation*, Vol. 1, No. 3, 1993, pp 191-211.

(10.) Callahan, K.J., and Weeks, G.E., "Optimum Design of Composite Laminates Using Genetic Algorithms ", *Composites Engineering*, Vol. 2, No. 3, 1992, pp. 149-160.

(11.) Ball, N. R., Sargent, P.M., and Ige, D.O., "Genetic Algorithm Representations for Laminate Layups ", *Artificial Intelligence in Engineering*, Vol. 8, No. 2, 1993, pp. 99-108.

(12.) Le Riche, R., and Haftka, R.T., "Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm", *AIAA Journal*, Vol. 31 No. 5, May 1993, pp. 951-956.

(13.) Nagendra, S., Haftka, R.T., and Gürdal, Z., "Design of Blade Stiffened Composite Panels by a Genetic Algorithm Approach," Published in *Proc. of the 34'th AIAA/ASME/AHS SDM Conf.,* La Jolla CA, April 19-22, 1993. pp. 2418-2436.

(14.) Nagendra, S., Haftka, R.T., and Gürdal, Z., "PASCO-GA : A Genetic Algorithm based Design Procedure For Stiffened Composite Panels under Stability and Strain Constraints," *Tenth*

DOD/NASA/FAA Conf. on Fibrous Composites in Structural Design, Nov 1-4, 1993, Hilton Head, SC 29928.

(15.) Kogiso, N., Watson, L. T., Gürdal, Z., and Haftka, R. T., "Genetic Algorithms with Local Improvement for Composite Laminate Design", Structural Optimization, Vol. 7, No.4, 1994, pp. 207-218.

(16.) Kogiso, N., Watson, L. T., Gürdal, Z., Haftka, R. T., and Nagendra, S., "Designing Composite Laminates with Genetic Algorithms with Memory", Mechanics of Composite Materials and Structures, (accepted for Publication)

(17.) Le Riche, R., and Haftka, R.T., "Improved Genetic Algorithm for Minimum Thickness Composite Laminate Design", Proc. of International Conf. on Composite Engineering, Aug 28-31, 1994, New Orleans, La.

(18.) Nagendra, S., Haftka, R.T., Gürdal. Z., and Starnes, J.H., Jr."Design of a Blade Stiffened Composite Panels with a Hole", *Composite Structures,* Vol. 18, No. 4, 1991, pp. 195-219.

(19.) Anderson, M.S., Stroud, W.J., Durling, B.J., and Hennessy, K.W., "PASCO: Structural Panel Analysis and Sizing Code. User's Manual", *NASA Technical Memorandum 80182*, November, 1981.

(20.) Stroud, W.J., and Anderson, M.S., "PASCO: Structural Panel Analysis and Sizing Code. Capability and Analytical Foundations", *NASA Technical Memorandum 80801*, November, 1981.

(21.) Wittrick, W.H., and Williams, F.W., "Buckling and Vibration of Anisotropic or Isotropic Plate Assemblies under Combined Loadings," *International Journal of Mechanical Science,* Vol. 16, No. 3, 1973, pp 209-239.

(22.) Davis, L., **Handbook of Genetic Algorithms**, *Van Nostrand Reinhold*, New York, 1991.

(23.) Goldberg, D.E., "Genetic Algorithms and Evolutionary Algorithms Come of Age", *Communications of the ACM,* Vol. 37, No. 3, March 1994, pp. 113-119.

(24.) Forrest, S., "Genetic Algorithms: Principles of Natural Selection Applied to Computation," *Science*, Vol. 261, No. 8, August 1993, pp. 872-878.

(25.) Forrest, S. and Mitchell, M., "What Makes a Problem Hard for a Genetic Algorithm ? - Some Anomalous Results and Their Explanation," *Machine Learning*, Vol. 1, No. 13, 1993, pp 285-319.

(26.) Holland, J.H., **Adaptation In Natural and Artificial Systems**, Ann Arbor, The University of Michigan Press, 1975.

(27.) Le Riche, R., and Haftka, R.T., "Optimal Number of Searches and Search Length for Reliable Genetic Optimization", Submitted for publication *Evolutionary Computation* .

(28.) Lehmer, D.H., **Mathematical Methods in Large Scale Computing Units**, Prentice Hall, Englewoods Cliff, 1989.

(29.) Kahaner, D., Moler, C., and Nash, S., **Numerical Methods and Software**, Prentice Hall, Englewoods Cliff, 1989.

APPENDIX - Random Number Generators and the GA.

The GA is a stochastic procedure which depends on the sequence of pseudo random numbers based on an initial input seed. Two random number generators were tested for generating random numbers. The first one was a random number generator (RAN-1) based on the linear congruential method [28], which has the advantage of being very fast and requiring only a few operations per call. However, RAN-1 suffers from a disadvantage that it is not free of sequential correlation on successive calls. If k random numbers at a time are used to plot points in a k-dimensional space (with each co-ordinate between 0 and 1), the points will not tend to "fill up" the k-dimensional space, but rather will lie on $(k-1)$-dimensional hyperplanes. There will be at most $m^{1/k}$ such planes [29], where m is the modulus (typically 2^{32}). Thus if a linear congruential random number generator is used in the GA, attention of the algorithm would be focused in a small fraction of the complete design space and the entire design space will not be explored during the course of the search.

The second algorithm (RAN-2) was a Fibonacci based recursive random number generator [36] which is free from the correlation problem. To compare the performance of the two random generators, 20 runs were carried out using the basic GA for each of the random number generators. Based on some experimentation, probabilities of crossover $P_{\mathrm{c}} = 0.95$, mutation $P_{\mathrm{m}} = 0.03$, permutation $P_{\mathrm{p}} = 0.95$, and a population size n_{d} of 10 were used. It was observed that the best design is a buckling critical 8 stack (32 plies) laminate with a weight of 22.68 lbs. The reliability of obtaining the practical optima for an increasing number of generations is shown in Figure A1. As expected RAN-2 performed better than RAN-1 in terms of reliability over most of the range.

Table 1. Hercules AS4/3502 graphite-epoxy lamina material properties.

Material Properties		
Young's Modulus (longitudinal)	E_{11}	$= 18.50\text{x}10^6$ lb/in^2
Young's Modulus (transverse)	E_{22}	$= 1.64\text{x}10^6$ lb/in^2
Shear Modulus	G_{12}	$= 0.87\text{x}10^6$ lb/in^2
Poisson's Ratio	ν_{12}	$= 0.3$
Density	ρ	$= 0.057$ lb/in^3
Ply Thickness	t_{ply}	$= 0.0052$ in

Strain Allowables (away from the cutout)		
ϵ_{1max} (tension)	=	0.008
ϵ_{1min} (compression)	=	0.008
ϵ_{2max} (tension)	=	0.02
ϵ_{2min} (compression)	=	0.0036
γ_{12}	=	0.012

Table 2. Tuning the GA—simplified problem, blade height of 3.1212 in.

P_c	P_{ils}	P_{ilsw}	P_{add}	P_{del}	P_{om}	q	*Reliability*	N_{R}	N^*_{50}
INTRA- AND INTERLAMINATE SWAP PROBABILITY									
0.95	0.95	0.95	0.01	0.01	0.01	0.9	30	1037	2015
0.95	0.75	0.95	0.01	0.01	0.01	0.9	50	1099	1099
0.95	0.55	0.95	0.01	0.01	0.01	0.9	50	1106	1106
0.95	0.75	0.90	0.01	0.01	0.01	0.9	50	1133	1133
0.95	0.75	0.75	0.01	0.01	0.01	0.9	60	1160	877
0.95	0.75	0.55	0.01	0.01	0.01	0.9	70	1341	772
DELETION OPERATOR PROBABILITY TUNING									
0.95	0.75	0.55	0.01	0.01	0.01	0.9	70	1341	772
0.95	0.75	0.55	0.01	0.05	0.01	0.9	40	1071	1453
0.95	0.75	0.55	0.01	0.10	0.01	0.9	40	1043	1415
0.95	0.75	0.55	0.01	0.25	0.01	0.9	40	1127	1529
0.95	0.75	0.55	0.01	0.30	0.01	0.9	10	989	6506
BEST SET (70 Runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.9	59	1233	958

---

* The target reliability T was chosen as 50%, and N^*_{50} calculated from Eq. 16.

Table 3. Soft-skin designs for panels without eccentricity.

P_c=0.95 P_{ils}=0.55 P_{ilsw}=0.75 P_{add}=0.01 P_{del}=0.01 P_{om}=0.01 P_m=0.03 q=0.9

PANEL	BLADE	BUCKLING	PLIES	STACKING SEQUENCE
MASS	HEIGHT	EIGENVALUE		S : SKIN B : BLADE
(lbs)	(in.)	λ_{cr}		
22.77	3.25	1.0090	S-32	$[\pm45/90_4/\pm45_5]_S$
			B-68	$[\pm45_4/0_2/\pm45_2/(0_4/\pm45)_3/0_2]_S$
22.79	3.11	1.0154	S-36	$[\pm45/90_2/\pm45\ /90_4/\pm45_4]_S$
			B-64	$[(\pm45_2/0_2)_2/0_2/(\pm45/0_2)_2/\pm45/0_4/\pm45/0_2]_S$
22.95	2.72	0.9776	S-32	$[\pm45/90_4/\pm45_5]_S$
			B-76	$[\pm45/0_2/\pm45_2/0_4/(\pm45/0_4)_2/\pm45/0_2/\pm45/0_4/\pm45/0_2]_S$
22.99	3.02	0.9947	S-32	$[\pm45/90_4/\pm45_5]_S$
			B-72	$[\pm45_2/0_2/\pm45_4/0_2/90_2/(0_4/\pm45)_3]_S$
23.09	2.91	1.0006	S-36	$[\pm45_2/90_4/\pm45_5]_S$
			B-68	$[\pm45_3/(0_4/\pm45)_2/0_2/\pm45/(0_4/\pm45)_3]_S$
23.11	2.78	0.9850	S-32	$[\pm45_2/90_4/\pm45_6]_S$
			B-76	$[\pm45_5/0_2/(\pm45/0_4)_2/90_2/0_4/\pm45/0_4/\pm45]_S$
23.41	2.63	0.9833	S-32	$[(\pm45/90_2)_2/\pm45_4]_S$
			B-80	$[\pm45_3/0_2/\pm45/0_4/\pm45/0_2/90_2/(0_4/\pm45)_3/0_2]_S$
23.45	2.64	0.9986	S-32	$[(\pm45/90_2)_2/\pm45_4]_S$
			B-80	$[\pm45_4/(0_4/\pm45)_5/0_2]_S$
23.53	2.79	1.0223	S-36	$[\pm45_2/90_4/\pm45_5]_S$
			B-72	$[\pm45_3/0_4/\pm45/0_2/\pm45/(0_4/\pm45)_2/0_4/90_2/0_2]_S$
23.80	2.52	1.0047	S-32	$[\pm45/90_4/\pm45_5]_S$
			B-88	$[\pm45_3/(0_4/\pm45)_3/0_2/\pm45/(0_4/\pm45)_2/\pm45/0_2]_S$

Table 4. Effect of improvements on GA performance for complete problem

(based on 40 optimizations of 150 generations and population n_{d}=20).

P_c	P_{ils}	P_{ilsw}	P_{add}	P_{del}	P_{om}	P_m	q	R	
								1%	2%
Improved GA (40 Runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.03	0.9	35	63
Basic CROSSOVER - COMPLETE STRING (10 Runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.03	0.9	10	30
Basic MUTATION - COMPLETE STRING (10 Runs)									
0.95	0.75	0.55	0.03				0.9	10	40
PERMUTATION P_p=0.75 (10 Runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.03	0.9	10	50
ABSENCE OF INTERLAMINATE SWAP (10 Runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.03	0.9	20	30

Table 5. Effect of mutation probabilities for complete problem.

(150 generations, and population size n_{d}=20).

P_c	P_{ils}	P_{ilsw}	P_{add}	P_{del}	P_{om}	P_m	q	R	
								1%	2%
BEST SET (40 runs)									
0.95	0.75	0.55	0.01	0.01	0.01	0.03	0.9	35	63
MUTATION PERTURBATIONS (10 runs)									
0.95	0.75	0.55	0.01	0.02	0.01	0.03	0.9	40	70
0.95	0.75	0.55	0.01	0.02	0.02	0.03	0.9	40	70
0.95	0.75	0.55	0.01	0.02	0.03	0.03	0.9	20	70
0.95	0.75	0.55	0.01	0.03	0.01	0.03	0.9	20	60
MUTATION - COMPLETE STRING (10 runs)									
0.95	0.75	0.55	0.03				0.9	10	40

Table 6. Panel designs with buckling and strain constraints with BASIC GA.

MASS (lbs)	λ_b	λ_s	BLADE Ht.(in.)	PLIES	LAMINATE S : SKIN B : BLADE
CONTINUOUS VARIABLE OPTIMUM DESIGN					
26.08**	0.9969	0.9910	3.0400	(S)-[40]	$[(45_3/-45_3)_3/90_2]_s$
				(B)-[76]	$[(45/-45/0_4)_5/45/-45/90/0_4/90]_s$
GA BEST DESIGNS					
25.19	1.0453	1.0331	3.2555	(S)-[32]	$[\pm45/90_2/\pm45_4/90_2/\pm45]_s$
				(B)-[80]	$[\pm45_4/90_2/0_2/(\pm45/0_4)_2/90_2/0_4/\pm45/0_4/\pm45_2]_s$
25.24	1.0020	1.0331	3.2716	(S)-[32]	$[(\pm45/90_2)_2/\pm45_4]_s$
				(B)-[80]	$[\pm45_5/0_4/90_2/\pm45/0_2/90_2/0_2/\pm45/(\pm45/0_4)_2/\pm45]_s$
25.26	1.0691	1.0190	3.2780	(S)-[32]	$[\pm45/90_2/\pm45/90_4/\pm45_3]_s$
				(B)-[80]	$[\pm45_2/0_4/\pm45_3/(0_2/\pm45)_2/\pm45/0_2/90_2/0_2/\pm45/0_4/90_2/\pm45]_s$
25.30	1.0182	0.9759	3.2946	(S)-[32]	$[\pm45/90_4/\pm45_5]_s$
				(B)-[80]	$[(\pm45_2/0_2)_2/\ 0_2/90_2/0_2/\pm45/(\pm45/0_2)_3/0_2/\pm45_2/0_2]_s$
25.30	0.9903	1.0321	3.2946	(S)-[32]	$[\pm45/90_4/\pm45/90_2/\pm45_3]_s$
				(B)-[80]	$[\pm45_2/90_2/\pm45_2/0_4/90_2/0_2/\pm45_2/(0_2/\pm45)_4/0_2]_s$
25.53	1.0813	1.1552	3.2546	(S)-[36]	$[\pm45/(90_2/\pm45_2)_2/\pm45_2]_s$
				(B)-[76]	$[\pm45/0_2/\pm45_2/90_2/\pm45/(0_4/\pm45)_2/0_2/\pm45/0_4/90_2/0_2/\pm45]_s$
25.55	1.0485	1.0030	3.2641	(S)-[36]	$[(\pm45_3/90_2)_2/\pm45]_s$
				(B)-[76]	$[\pm45_3/0_4/\pm45/90_2/(\pm45/0_4)_2/\pm45_2/0_2/\pm45/90_2/0_2]_s$
25.56	1.0709	1.04361	3.2626	(S)-[36]	$[\pm45/90_2/\pm45_7]_s$
				(B)-[76]	$[\pm45_3/0_4/90_2\ /\pm45_3/0_4/90_2/(0_4/\pm45)_2/90_2]_s$
25.58	1.0322	1.10911	2.8620	(S)-[40]	$[\pm45/90_4/\pm45_2/90_2/\pm45_2]_s$
				(B)-[88]	$[\pm45_3/(0_2/\pm45)_2/90_2\ /\pm45_2/0_4/(\pm45/0_4)_2/90_2/0_2/\pm45/0_2]_s$

** Converged soft-skin design from reference [23], panel eccentricity = ±0.03.

Table 7. Panel designs with buckling and strain constraints with IMPROVED GA.
(Panel Eccentricity = ±0.03)

MASS (lbs)	λ_b	λ_s	BLADE Ht.(in.)	PLIES	LAMINATE S : SKIN B : BLADE
24.14	1.0021	1.0033	3.1632	(S)-[32]	$[\pm45/90_4/\pm45/90_2/\pm45_3]_s$
				(B)-[76]	$[\pm45_3/0_4/(\pm45/0_2)_3/\pm45/0_4/90_2/0_4/90_2]_s$
24.17	1.0215	1.0484	3.0387	(S)-[36]	$[\pm45/90_4/\pm45/90_2/\pm45_4]_s$
				(B)-[72]	$[\pm45_3/0_2/(\pm45/0_4)_3/90_2/0_4/90_2/0_2]_s$
24.20	1.0148	1.0210	3.0497	(S)-[36]	$[\pm45/90_4/\pm45/90_2/\pm45_4]_s$
				(B)-[72]	$[\pm45_2/0_2/\pm45/0_4/\pm45/0_2/(\pm45/0_4)_2/90_2/0_2/\pm45/0_2]_s$
24.20	1.0145	1.0411	3.1860	(S)-[32]	$[\pm45/90_4/\pm45_5]_s$
				(B)-[76]	$[\pm45_4/0_4/(\pm45/0_2)_2/0_2/90_2/0_4/\pm45/0_4/90_2/0_2]_s$
24.34	1.0403	1.0216	3.1058	(S)-[36]	$[\pm45/90_2/\pm45/90_4/\pm45_4]_s$
				(B)-[72]	$[\pm45_5/0_4/\pm45/0_4/90_2/0_4/\pm45/0_4/90_2/0_2]_s$
24.51	1.0514	1.0810	3.1742	(S)-[36]	$[\pm45/90_4/\pm45_6]_s$
				(B)-[72]	$[\pm45_4/(\pm45/0_4)_2/90_2/0_4/\pm45/0_4/90_2/0_2]_s$
24.58	1.0249	1.0390	3.0384	(S)-[32]	$[(\pm45/90_2)_2/\pm45_4]_s$
				(B)-[80]	$[\pm45_4/(0_4/\pm45)_2/0_2/\pm45/0_4/(90_2/0_2)_2/\pm45/0_2]_s$
24.60	1.0465	1.0117	3.0477	(S)-[32]	$[\pm45/90_2/\pm45_6]_s$
				(B)-[80]	$[\pm45_3/0_4/(\pm45/0_2)_3/0_2/(\pm45/0_4)_2/90_2/0_2]_s$
24.61	1.0124	1.0075	3.0705	(S)-[40]	$[\pm45/90_4/\pm45_3/90_2/\pm45_3]_s$
				(B)-[68]	$[\pm45_2/(\pm45/0_4)_2/90_2/0_4/(\pm45/0_2)_2/0_2/\pm45]_s$
24.62	1.0650	1.0888	2.9174	(S)-[36]	$[\pm45/90_4/\pm45_6]_s$
				(B)-[76]	$[\pm45/(\pm45/0_4)_2/(\pm45/0_2)_2/0_2/\pm45/0_4/90_2/0_4/\pm45]_s$

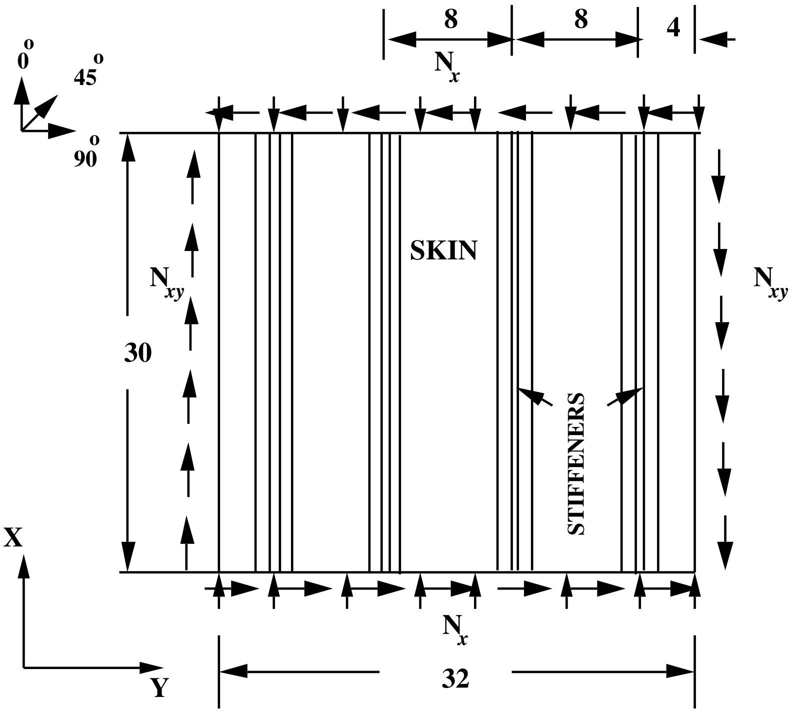

Figure 1. Blade stiffened panel under compression and shear (all dimensions in inches).

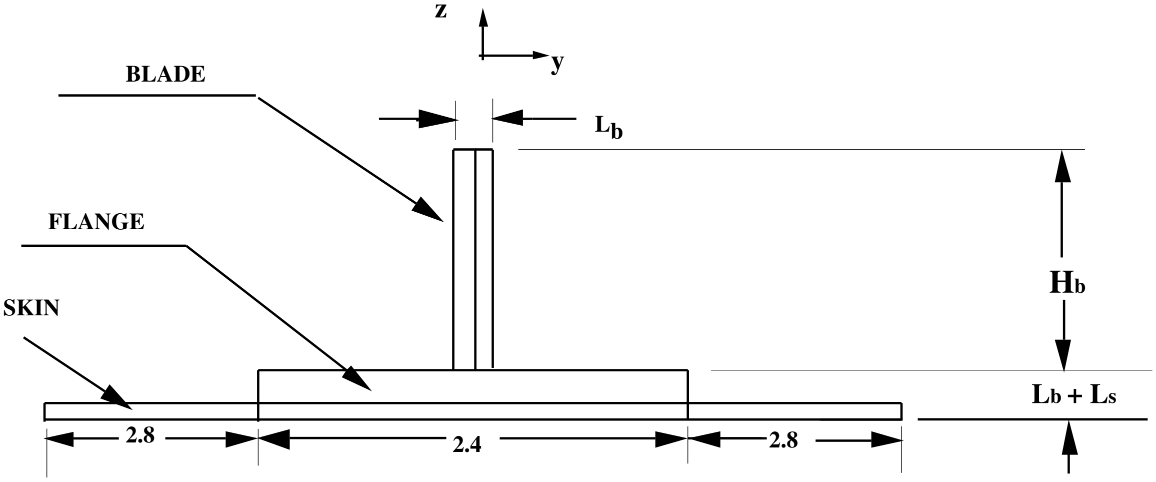

Figure 2. PASCO repeating element with variable laminate thicknesses and stiffener blade height.

STIFFENED PANEL DESIGN VARIABLES

BLADE HEIGHT: [0, 1, 2]

LAMINATE STACK VARIABLES

SOFT-SKIN STACKS [$\pm 45, 90_2, *$]: [2, 3, 0]

BLADE STACKS [$0_2, \pm 45, 90_2, *$]: [1, 2, 3, 0]

$*$ IDENTIFIES AN EMPTY STACK IN THE LAMINATE STRING DENOTED BY 0.

FIXED STRING REPRESENTATION FOR INDIVIDUAL DESIGNS

BLADE HEIGHT SUBSTRING

ARRAY LENGTH L_h=9 TOTAL NUMBER OF POSSIBLE DESIGNS = 3^9

BLADE HEIGHT H_b=2.6590in=101101221

SKIN LAMINATE SUBSTRING

ARRAY LENGTH L_s=15 TOTAL NUMBER OF POSSIBLE DESIGNS = 3^{15-1}

$[*_5/\pm 45_5/90_4/\pm 45_3]_s = \lfloor_m 222332222200000$

BLADE LAMINATE SUBSTRING

ARRAY LENGTH L_b=25 TOTAL NUMBER OF POSSIBLE DESIGNS = 4^{25-1}

$[*_5/\pm 45/0_2/\pm 45_2/0_2/90_2/0_4/\pm 45/0_2/\pm 45_2/0_4/\pm 45_4/0_4/\pm 45]_s = \lfloor_m 21122221122121131221200000$

REPRESENTATIVE GENETIC DESIGN

BLADE HEIGHT — SKIN LAMINATE — BLADE LAMINATE: 101101221 — $\lfloor_m$222332222200000 — $\lfloor_m$21122221122121131221200000

TOTAL NUMBER OF COMBINATORIAL DESIGNS $= 3^9 * 3^{14} * 4^{24} \approx 2.65 \times 10^{25}$

Figure 3. Genetic representative design of the stiffened composite panel and design variables ($\lfloor_m$ denotes the laminate midplane).

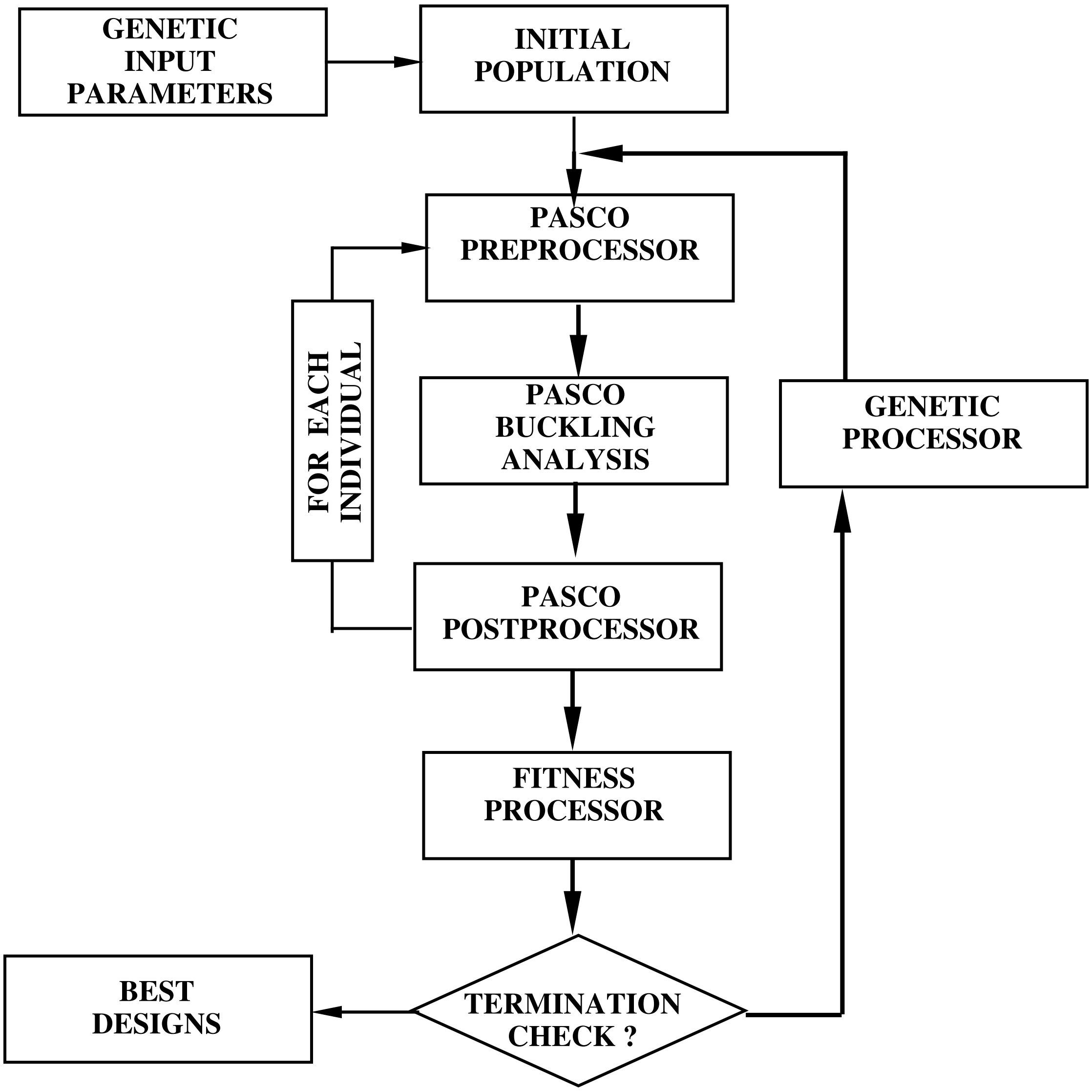

Figure 4. The basic genetic algorithm.

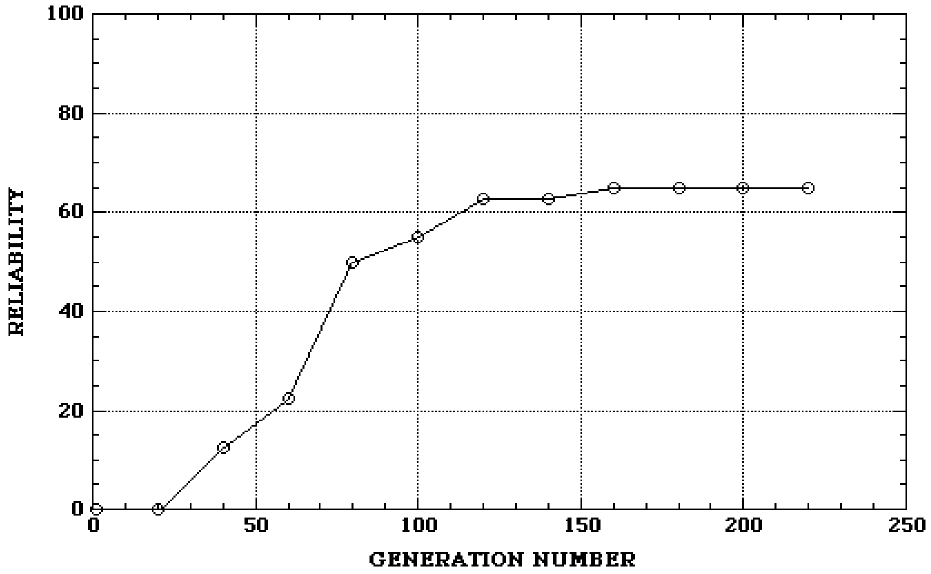

Figure 5a. Modified GA reliability for varying skin and blade laminate design; blade height fixed at 3.2121in; $P_c = .95$, $P_{ils} = .75$, $P_{ilsw} = .55$, $P_{add} = .01$, $P_{del} = .01$, $P_{om} = .01$, $q = .9$.

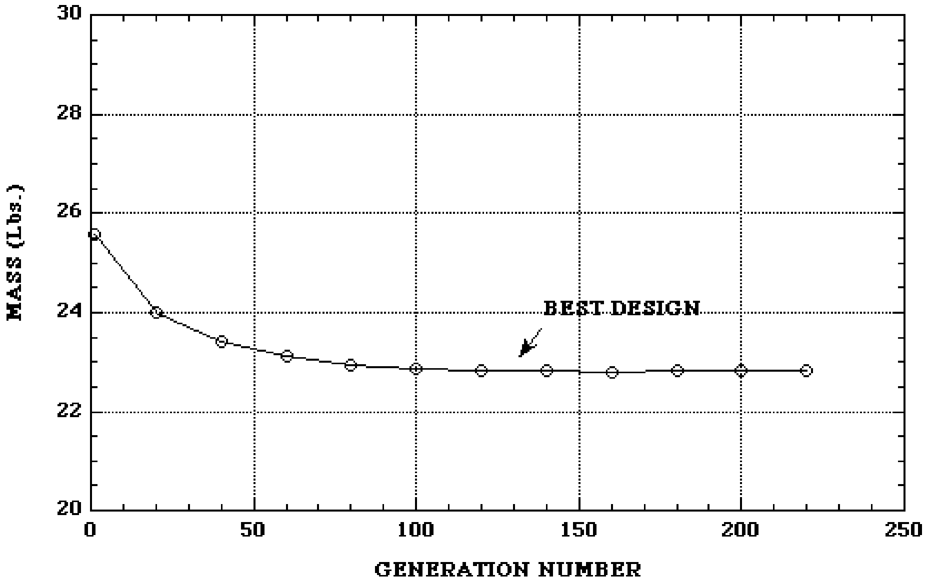

Figure 5b. Average mass of the best design for varying skin and blade laminate design; blade height fixed at 3.2121in; $P_c = .95$, $P_{ils} = .75$, $P_{ilsw} = .55$, $P_{add} = .01$, $P_{del} = .01$, $P_{om} = .01$, $q = .9$.

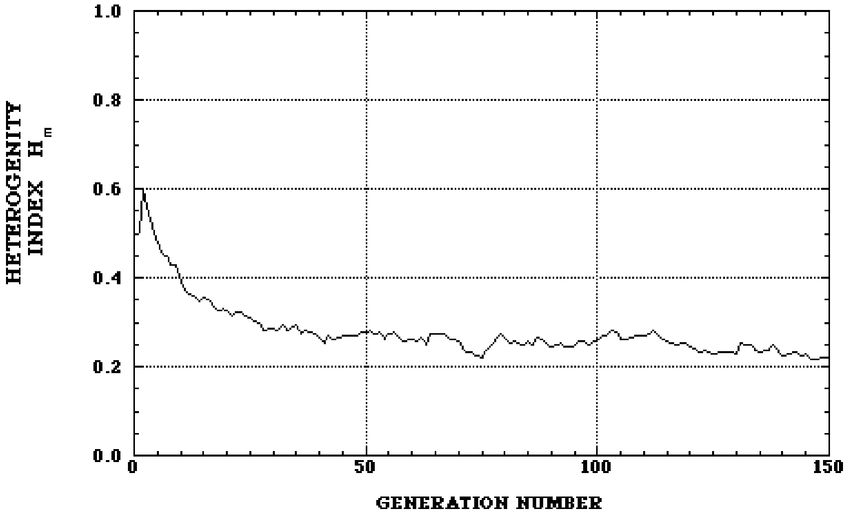

Figure 6. Average heterogeneity index for buckling and strain case. $P_c = .95$, $P_{ils} = .75$, $P_{ilsw} = .15$, $P_{add} = .05$, $P_{del} = .2$, $P_{om} = .05$, $q = .9$.

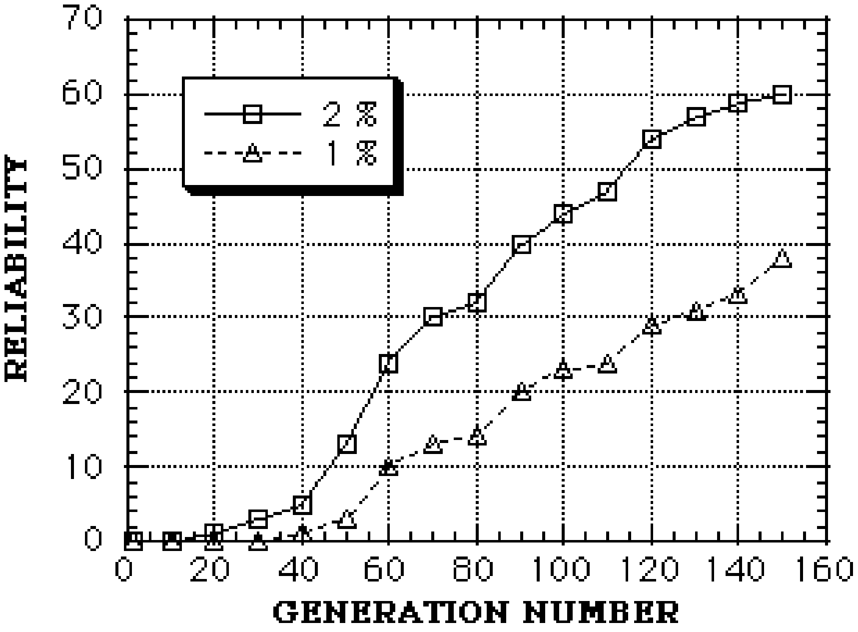

Figure 7. Algorithm reliability for stiffened composite panels (40 runs). $P_c = .95$, $P_{ils} = .75$, $P_{ilsw} = .55$, $P_{add} = .01$, $P_{del} = .01$, $P_{om} = .01$, $q = .9$.