PERGAMON

Computers and Structures 79 (2001) 131–143

Computers & Structures

www.elsevier.com/locate/compstruc

Composite laminate design optimization by genetic algorithm with generalized elitist selection

G. Soremekun [a,1], Z. Gürdal [b], R.T. Haftka [c], L.T. Watson [d,*]

[a] *Department of Engineering Science and Mechanics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA*

[b] *Departments of Aerospace and Ocean Engineering Science and Mechanics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA*

[c] *Department of Engineering Mechanics and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA*

[d] *Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106, USA*

Received 30 November 1998; accepted 10 April 2000

Abstract

Genetic algorithms with elitist selection based on cloning a best single individual (SI) from one generation to the next are popular, but generalized elitist selection (GES) procedures have been proposed and tried in the past. The present paper explores several generalized elitist procedures for the design of composite laminates. It is shown that GES procedures are superior to an SI procedure for two types of problems. The first type involves many global optima, and the GES procedure can find several global optima more efficiently than the SI procedure. This may give a designer more design freedom. The second type of problem involves an isolated optimum surrounded by many designs with performance that is very close to optimal. It is shown that GES procedures can find the optimum and near optimal designs much more easily and reliably than the SI procedure. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Composite laminate; Elitist selection; Genetic algorithm; Optimal design

1. Introduction

In the past, laminate stacking sequence design optimization was formulated as a continuous optimization problem and solved using gradient based methods by Jacoby et al. [1] and Gürdal and Haftka [2]. These efforts met with limited success for two reasons. First, stacking sequence design involves discrete design variables, such as ply thickness and orientation angle, which must be converted to continuous variables before the problem is solved. Once a continuous-valued solution is found, it must be rounded to the nearest manufacturable ply angle or ply thickness, which may result in a design which is either nonoptimal or violates certain imposed constraints. Second, composite laminate design problems typically involve multimodal search spaces. Gradient based algorithms for such problems may often converge to locally optimal regions of the design space, especially if the starting point is chosen far from the global optimum. These drawbacks can be avoided by using a genetic algorithm (GA) optimization procedure.

GAs, first conceived by Holland [3] in the 1970s, have received substantial attention during the last decade in an effort to better understand their search characteristics and capabilities, and address problems associated with computational inefficiency [4–6]. Based on a set of probabilistic rules, GAs utilize the processes of natural selection by mimicking the concept of survival of the

* Corresponding author. Tel.: +1-540-231-7540; fax: +1-540-231-6075.

E-mail address: ltw@vtopus.cs.vt.edu (L.T. Watson).

[1] Now at: ADOPTECH, Inc., Virginia Tech Corporate Research Center, Suite 1204, Blacksburg, VA 24060, USA.

PII: S0045-7949(00)00125-5

fittest. GAs are excellent all-purpose optimization algorithms because they can accommodate both discrete and continuous valued design variables and search through nonlinear or noisy search spaces by using payoff (objective function) information only. By working with a population of designs instead of a single point, GAs are less likely to get trapped in locally optimal areas of the design space. For these reasons, GAs have been utilized by many researchers [7–10] in the field of composite laminate design optimization during the past few years.

A standard genetic algorithm (SGA) is applied to two composite laminate design problems in the present paper. In this context, the SGA is comprised of a linear ranking scheme for parent selection, a set of basic genetic operators (crossover, mutation, and gene swap), and an elitist selection procedure. The first design problem, buckling load maximization of a simply supported composite plate, is known to contain multiple global optimum points. The second problem, maximizing the twisting displacement of a cantilevered composite plate, has one global optimum point that is surrounded by a large number of points with fitnesses that are very close to the optimal solution. The purpose of this study is to show that the SGA does not provide the most robust means of optimization for either composite laminate design problem

In the first problem, the SGA is likely to find one of the optimal solutions, but not the others. A more desirable result would be finding the entire set of global optimum designs. The designer could then choose one of them, based on manufacturing or other design considerations, which would provide the most efficient, high performance structure. Many researchers have used GAs for multimodal GA optimization [5,11]. In particular, Goldberg and Richardson [12] have addressed the problem of multimodal search spaces through fitness sharing, where a distance metric is defined over the search space to force neighboring individuals in a population to share their fitness values. This reduces competition between different global optimum design points. To use fitness sharing, one must assign appropriate values to the distance metric as well as a level of exponential scaling to the fitness function. Sharing functions have been successfully used for composite laminate design optimization with an L_1 or L_∞ norm for the separation metric.

In the second problem, there is a large number of plies in the stacking sequence and each ply can have any value from a large set of prescribed orientation angles. Moreover, a small perturbation (caused by changing the orientation angle of a single ply) in a design located in the neighborhood of the optimum does not produce a large change in fitness. For this reason, the SGA has trouble sifting through the number of designs in this relatively "flat" portion of the design space around the optimum, and is likely to converge prematurely. The goal in this case is to have the GA search through the condensed area of near-optimum designs and find the global optimum.

The present paper suggests a generalized elitist genetic algorithm (GEGA) as an alternative to the SGA. The GEGA replaces elitist selection with a more general selection scheme, namely, multiple elitist or variable elitist selection, to produce the desired results discussed above. In what follows, the SGA procedure is discussed. Next, the multiple elitist and variable elitist selection schemes are presented. The feasibility of the GEGA utilizing the new selection schemes is then tested on a simple trigonometric function with multiple global optimum points. Finally, the SGA and the GEGA are applied to each composite laminate optimization problem, and numerical results generated by both algorithms are presented and compared.

2. Standard genetic algorithm procedure

An initial population of genetic strings with randomly chosen genes is created first. The size of the population used in the present work remains constant throughout the genetic optimization. Various genetic operators are applied at given probabilities to generate new laminates. In order to form successive generations, parents are chosen from the current population based on their fitnesses. Next, crossover, mutation, and permutation operators are applied to create children, who replace their parents in the next generation. One generation after another is created until some convergence criterion is met (running the GA for a fixed number of generations or for a certain number of generations without improvement are common convergence strategies). Detailed descriptions of these operators are provided in the following subsections. A schematic of the GA procedure is given in Fig. 1. More details on the genetic operators can be found in the literature [8,10,13,15,16].

2.1. Parent selection

Parent selection is accomplished using a linear ranking procedure. According to Baker and Grefenstette [14], the primary advantage of linear ranking over other popular parent selection schemes (such as proportional selection) is that the resulting algorithm is less prone to premature convergence caused by individuals that have fitnesses far above the population's average fitness value. Before parent selection can begin, all laminates must be ranked from best to worst according to the value of each laminate's objective function. A roulette wheel is implemented where the ith ranked laminate in the population is given an interval $[\phi_{i-1}, \phi_i)$, whose size depends on the population size, P, and its rank, i, in the population

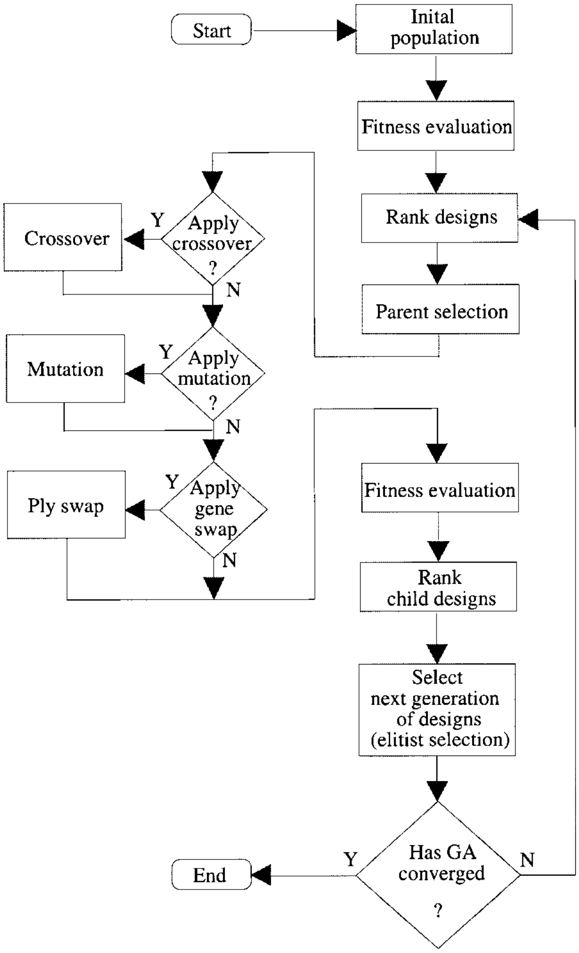

Fig. 1. Standard GA procedure.

$$\phi_i = \phi_{i-1} + \frac{2(P-i+1)}{P(P+1)}, \tag{1}$$

where $\phi_0 = 0$, and $i = 1, \ldots, P$. For example, if there are three laminates in a population, the roulette wheel is divided into three pieces with the best laminate taking 50% of the wheel, the second best taking 33%, and the poorest taking 17%.

A uniform random number is generated between 0 and 1; laminate i is selected as a parent if the number lies in the interval $[\phi_{i-1}, \phi_i)$. Continuing with the above example, if random numbers $0.3 \in [0, 0.5)$ and $0.7 \in [0.5, 0.83)$ are drawn, then laminate 1 and laminate 2 will become parents of the first child. Parents of a child are required to be distinct laminates from the population.

2.2. Crossover

Children are created by combining a portion of each parent's genetic string in an operation called one-point crossover. The operation is applied by first drawing a random number to determine the crossover point. The gene string is then split at the same point in both parents. The left piece from parent 1 and the right piece from parent 2 are combined to form a child laminate. If, during the creation of the child population, crossover is not applied, then one of the parent laminates is cloned into the child string. Child laminates are also forced to be distinct from each other and from other designs in the parent population. If a distinct child cannot be found after a prescribed number of iterations, then one of the parents is cloned into the child population. The crossover process is repeated as many times as necessary to create a new population of laminates.

2.3. Mutation

After a child has been created, it is exposed to a mutation operator with a small fixed probability (typically around 0.01 per gene). The purpose of mutation is to provide new, random bits of information during the genetic search. Although crossover provides the primary means of moving through the search space, it loses its effect when the population becomes uniform. Thus, by adding diversity to the population through random mutations, the crossover operator can remain effective throughout the search. Mutation also provides a random search capability which may discover new areas in the design space with attractive features. In this implementation, each gene is given a small probability to switch to any other permissible integer value except the value of the gene before ply alteration occurs.

2.4. Gene swap

In previous works, GAs applied to composite laminate design often used a permutation operator to aid in the genetic search, but it was found to shuffle the digits in the gene string too much. Thus, a less disruptive operator, gene swap, was implemented by Le Riche and Haftka [13] and is used in favor of permutation for the design problems considered in this work. The gene swap operator is implemented by randomly selecting two unique genes in the string and switching their positions. Gene swap can be effective for problems where certain parts of the laminate stacking sequence get set up faster than others. For example, if the optimal stacking sequence of the outer section of the laminate is determined first (as is the case for the laminate design problems studied in this work which involve bending), the gene swap operator may help the GA determine the optimal orientations for the inner part of the laminate by swapping genes from each section.

2.5. Creating the next generation

The fundamental difference between the GEGA proposed here and the typical SGA is in the determination of the next generation. The purpose of the various genetic operators is to produce successively improved populations of laminates. There is, however, no guarantee that there will be a design in the child population that is better than the best design of the parent population. In order to preserve the largest value of the objective function, an elitist method is typically used.

An elitist method (EL), first implemented by De Jong [5] in 1975, ranks the child population and parent population separately. The best laminate from the parent population and the worst laminate from the child population are identified. To create the new population, the best design from the parent population replaces the worst laminate from the child population. The EL method provides an explorative genetic search since each successive population is provided with a large number of new designs. The potential problem with EL selection is that important genetic information that may exist in other desirable designs of the parent population is lost. This result may not be significant if the search space is unimodal since the best designs migrate toward the single optimum. However, it will be shown that by not identifying and preserving such information, EL selection is incapable of finding multiple global optima. Furthermore, a highly explorative search technique does not provide the most robust means of sifting through a large number of designs in relatively "flat" portions of a design space.

3. Multiple elitist and variable elitist selection

The GEGA replaces elitist selection with multiple elitist (ME) or variable elitist (VE) selection. Both the ME and VE schemes, discussed in greater detail in the following subsections, are based on the (μ, λ) selection strategies developed by Bäck [17,18], where λ is the number of members in the population of which μ best are deterministically selected as parents in the next generation. According to Bäck, the selection mechanism is one of the primary means of controlling the GA's convergence rate and its likelihood of finding global optima. In this paper, we explore some of the implementation details and some of the effects of varying μ with respect to λ for two types of composite laminate design problems which present difficulties for the SGA.

3.1. Multiple elitist selection

While an elitist procedure is sufficient for many GA applications, there are some cases often found in composite laminate design which can benefit from selection schemes that provide different types of search capabilities. Different stacking sequences for some composite laminate design problems can provide similar performance, yielding multimodal search spaces. Other problems, such as those which involve bending, yield design spaces with a single global optimum surrounded by many near optimum designs. This occurs because the outermost plies of the stacking sequence have the greatest influence on the stacking sequence. Changing the ply orientation angles of the innermost plies has little effect on laminate performance, resulting in large groups of designs with identical properties and high fitness values. These two types of problems, which are studied in this work, can benefit from a selection scheme which can (a) track multiple global optima, and/or (b) interrogate local areas of the design spaces quickly. Multiple elitist and variable elitist selection, described in the following paragraphs, were designed to accomplish these specific tasks.

Multiple elitist selection preserves more information about good designs from the parent population than the elitist method does. Two different methods of Multiple elitist selection are explored. The first Multiple elitist selection scheme, ME_1, is an extension of the EL method. The top N_k designs from the parent population are selected and placed into the new population. The number of child designs required to fill the remainder of the new population are created from the parent population and placed into the new population. This selection scheme is computationally less intensive because fewer child designs require analysis. For ME_1 selection, the value of N_k cannot be set equal to the population size since this would code all the parent laminates into the new generation and prevent the GA from exploring the design space.

In the second Multiple elitist selection method, ME_2, the parent and child populations of size P are each separately ranked from best to worst. Both populations are then combined and ranked together (forming $2P$ laminates). The number (N_k) of top designs from the combined population are then carried over to the next generation. If N_k is equal to P (the maximum possible value of N_k), then the next generation is filled with the first half of the combined population and the procedure is complete. If N_k is less than P, then N_k laminates from the combined population comprise the first part of the new generation. To fill the remainder of the new generation, the ranked child population is searched, starting from the best laminate, for laminates that have not already been passed on to the new generation. The first child found, which is not located in the new generation, fills the first empty location in the new generation, and successive children fill out the rest of the new generation. Note that setting $N_k = 1$ in ME_2 selection is identical to the EL method if the top member from the combined populations is a parent design.

The selection pressure of the GEGA is controlled by the value of N_k. For $N_k = 1$, the GEGA preserves only the best design which allows many new child designs to enter into the population. A large supply of new designs allows the GA to search for other attractive areas in the search space. In this case, ME selection is similar to the EL strategy. However, as the value of N_k increases, the number of new designs entering the population decreases, causing the genetic search to become more localized. Thus, using large values of N_k at the beginning of the optimization procedure will slow the genetic search procedure and may prevent the GA from finding any optimum design point in a reasonable amount of time. However, if N_k is kept relatively small compared to the population size, a number of good designs can be tracked while maintaining a sufficient influx of new designs to continue searching other areas of the design space effectively. This feature is important for two reasons. First, for multimodal search space, some or all of the global optima can be tracked simultaneously. Second, for unimodal search spaces, the area of the design space which contains the global optimum can be exploited quickly once it has been located. The exploitative nature of GA occurs because many or all of the top designs that ME selection preserves from the parent population will have similar properties and fitnesses. Since these designs are likely to be chosen as parents, the crossover operator will produce child designs very similar to their parents and thus, allow the GA to interrogate the area. This second feature is especially beneficial if a global optimum point is surrounded by a large group of near-optimum designs.

3.2. Variable elitist selection

The value of N_k is held constant in ME selection. From the foregoing discussion, it is clear that one would like to set N_k to a low value in order to maintain the highest search capability of the GA, but at the same time would like to keep it larger during the final stages of the search to ensure that attractive areas of the design space are searched rigorously. This is best accomplished by varying N_k during the search, which is referred to as variable elitist selection. Although there are an infinite number of ways to vary N_k, a simple methodology based on the ME_2 selection scheme is explored here: a relatively small value of N_k is applied in the beginning of the search and then set to a maximum value (which is equivalent to the population size) towards the end of the search. This implementation provides an explorative search in the beginning to locate attractive areas of the design space which, in turn, can be searched rigorously by increasing N_k towards the end of the optimization procedure. Details and examples for these new selection strategies are given by Soremekun [16].

4. Test problem: a simple trigonometric function

The purpose of this section is to provide a simple comparative study between the capabilities of EL, ME and VE selection. The problem is to maximize

$$z(x,y) = (1 - \sin[x])(2 - \cos[2y]) + (2 + \sin[x])(1 + \cos[2y]) \tag{2}$$

over $-2 \leqslant \{x,y\} \leqslant 2$. This problem provides a challenge because it has three global optimum points: {1.571, 0, 6}, {−1.571, −1.571, 6}, {−1.571, 1.571, 6}.

A binary coding is used to provide a simplified means of discretizing the continuous problem. Each individual chromosome contains fourteen genes. The first seven genes determine the x value of an individual and the remaining seven determine the y value, yielding 65,536 possible combinations for x and y. The gene string for the ith member in the population is converted to real values for both x and y by

$$x_i = \frac{1 + \Sigma_{j=1}^{7} 2^{j-1} g_i^j}{32} - 2, \quad y_i = \frac{1 + \Sigma_{j=8}^{14} 2^{j-8} g_i^j}{32} - 2, \tag{3}$$

where g_i^j is the value of the jth gene in the chromosome. This genetic encoding amounts to a simplified discretization of the continuous problem.

Numerical experiments for each selection scheme consisted of 25 runs for 100 generations (fixed) each. Population sizes ranging from 10 to 60 were used to monitor the search capability required to simultaneously find multiple global optima. Other experiments included adjusting the value of N_k from 1 to $P/2$ when evaluating ME and VE selection.

Four evaluation criteria are used: The first criterion is the normalized cost per genetic search, C_n, determined by

$$C_n = \begin{cases} N_g P/R & (\text{EL, ME}_2 \text{ and VE selection}), \\ N_g N_c/R & (\text{ME}_1 \text{ selection}), \end{cases} \tag{4}$$

where N_g is the number of generations per run, P is the size of the population, R is the apparent reliability, and N_c is the number of child designs created in each generation. Apparent reliability is determined by taking the number of runs for which the GA finds at least one global optimal, divided by the total number of runs conducted, N_r.

A second criterion is the average number of optima found per genetic search

$$A_{N_o} = \frac{\sum_{j=1}^{N_r} N_o^j}{N_r}, \tag{5}$$

where N_o^j is the number of optima found in the jth optimization run. Eq. (5) is also used to determine criterion three, which is defined as the cost per optimum found:

$$C_o = \begin{cases} N_g P / A_{N_o} & \text{(EL, ME}_2 \text{ and VE selection)}, \\ N_g N_c / A_{N_o} & \text{(ME}_1 \text{ selection)}. \end{cases} \tag{6}$$

The final criterion is final population richness, which helps to monitor how the GA exploits global optimum regions of the design space. Population richness P_r is

$$P_r = \frac{N_{\Delta f}}{P N_r}, \tag{7}$$

where $N_{\Delta f}$ is the number of members in the final population of each run with fitness values within a certain small amount (Δf) of the optimum.

Results were gathered using the following set of genetic operator probabilities: crossover = 1.0, mutation = 0.02, and gene swap = 0.7. Relevant findings are summarized in Table 1. All selection schemes were capable of finding at least one global optimum using a population size of 10. However, given a typical initial population of randomly generated points (Fig. 2a), the EL scheme was incapable of converging to more than one global optimum during an optimization run (Fig. 2b), regardless of the population size used. In contrast, the ME and VE schemes consistently found multiple optima (Fig. 2c). ME_1 selection was the most effective scheme, requiring the fewest number of function evaluations to find global optima. The ME_2 strategy produced similar results in terms of its ability to find multiple global optima, but was more expensive than ME_1 selection. The VE selection scheme provided the most powerful search, locating 2.8 global optima on the

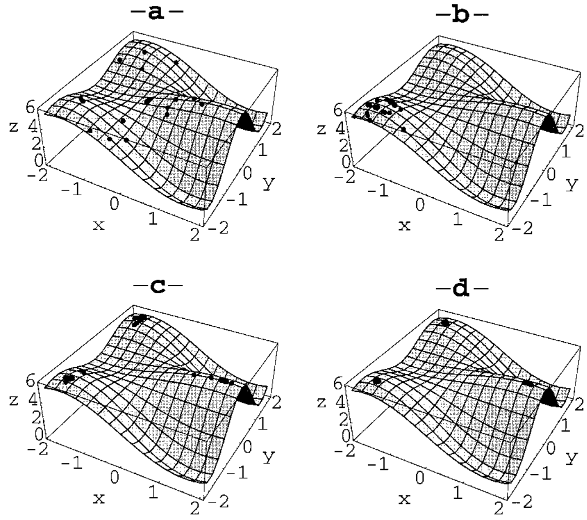

Fig. 2. (a) Typical initial population ($P = 30$); (b) final population using EL selection; (c) final population using ME_1 (or ME_2) selection and, (d) final population using VE selection.

average. The VE scheme rigorously searched the design space around all three global optima as well, yielding final populations with high population richness (Fig. 2d). The increased number of optima found per run using the ME and VE schemes resulted in a considerable increase in the cost of the search. This is because a large population size is required to maintain the explorative characteristics of the GA (which helps to find global optima) for larger values of N_k (which helps to preserve multiple optima). Note also that C_o is lower for VE than for ME_2.

Table 1
Global optima for trigonometric optimization problem

Selection scheme (N_k)	P	R	P_r[a]	C_n	C_o	A_{N_o}
EL	10	1.00	0.136	1000	1000	1.0
EL	30	0.96	0.053	3125	3125	0.96
EL	60	0.96	0.027	6000	5769.23	1.04
ME_1 (3)	10	0.96	0.408	732.3	676.0	1.04
ME_1 (12)	30	1.00	0.431	1812	943.8	1.92
ME_1 (26)	60	1.00	0.552	3124	1381.5	2.48
ME_2 (5)	10	1.00	0.552	1000	892.9	1.12
ME_2 (13)	30	1.00	0.449	3000	1595.7	1.88
ME_2 (29)	60	1.00	0.465	6000	2307.7	2.60
VE[b] (2)	10	0.96	0.96	1041.7	806.5	1.24
VE[b] (3)	30	1.00	0.872	3000	1363.6	2.20
VE[b] (11)	55	1.00	0.550	5500	1964.3	2.80

[a] $\Delta f = 0.001$.
[b] N_k set to P after 75 generations.

5. Composite laminate design: buckling load maximization

Results from the previous section provide the necessary justification to apply the GEGA towards a similar type of problem found in composite laminate design: buckling load maximization of a simply supported laminated composite plate. This problem was studied extensively by Le Riche and Haftka [8], using a version of the SGA discussed previously (Le Riche utilized a two-point crossover operator and a modified gene swap operator where two sets of genes were swapped instead of one). Over the course of several optimization runs, Le Riche discovered that the design space for this problem contained multiple global optima. The present work shows that it is possible to find many or all of these global in a single optimization run using the GEGA.

5.1. Plate analysis

The composite panel under consideration is 20 in. long, 10 in. wide, and simply supported on all four sides, (Fig. 3). The panel may be loaded under any combination of axial loads (i.e., N_x and N_y) and comprises two ply stacks which may be oriented 0°_2, $\pm 45^\circ$, or 90°_2, as shown in Fig. 3. The stacking sequence is constrained to be balanced and symmetric about the laminate mid-plane.

These assumptions eliminate coupling between extension and bending (i.e., $[B_{ij}] = 0$, where $[B_{ij}]$ are the coupling stiffnesses from classical lamination theory), and normal-shear extension coupling (i.e., $[A_{16}] = [A_{26}] = 0$, where $[A_{ij}]$ are the extensional stiffnesses) [19].

The laminate buckles into m and n half waves in the x and y directions, respectively, when the loads reach the values $\lambda_b N_x$ and $\lambda_b N_y$. The buckling load factor λ_b is defined in terms of the bending stiffness $[D_{ij}]$

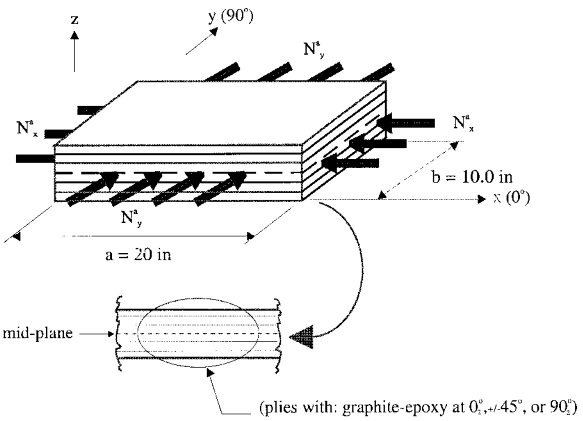

Fig. 3. Panel configuration and loading conditions.

$$\lambda_b = \sum_{m=1}^{M}\sum_{n=1}^{N}\pi^2\left[\frac{m^4 D_{11} + 2(D_{12} + 2D_{66})(rmn)^2 + (rn)^4 D_{22}}{(am)^2 N_x^a + (ran)^2 N_x^a}\right], \quad (8)$$

where a is the length of the plate, b is the width of the plate, and r is the plate aspect ratio (a/b). In Eq. 8, we assume the laminate to be specially orthotropic, allowing the effects of D_{16} and D_{26} to be neglected. The smallest value of λ_b over all possible combinations of m and n is the critical buckling load factor λ_{cb}, which determines the critical buckling loads for a specified combination of N_x^a and N_y^a

$$\begin{bmatrix} N_{x_{cb}} \\ N_{y_{cb}} \end{bmatrix} = \lambda_{cb}\begin{bmatrix} N_x^a \\ N_y^a \end{bmatrix}. \quad (9)$$

If λ_{cb} is larger than 1, the laminate can sustain the applied loads N_x^a and N_y^a without buckling.

5.2. Genetic code and optimization formulation

The length of the gene string is kept fixed throughout the optimization process. Each gene in the string is an integer between 1 and 3 corresponding to a two ply stack oriented at 0°, $a \pm 45^\circ$ ply stack, and a 90° two ply stack, respectively. To satisfy the mid-plane symmetry constraint, the GA is configured to design the left half of the stacking sequence only. The balance constraint, which ensures that each ply oriented at +45° is complemented with another ply oriented at −45°, is automatically satisfied by using two ply stacks. As a result, only $N/4$ ply orientations are required to describe the entire laminate. The optimization formulation is straightforward: maximize the buckling load of the laminate, which is defined as $\Phi = \lambda_{cb}$.

5.3. Results

Results were obtained for a 64-ply laminate made of graphite–epoxy ($E_1 = 18.5E6$ psi, $E_2 = 1.89E6$ psi, $G_{12} = 0.93E6$ psi, $\nu_{12} = 0.3$, and $t = 0.005$ in.), and ($N_x^a/N_y^a = 1.0$) loading. This panel configuration yields 16 design variables 43.05×10^6 possible stacking sequence combinations. Convergence studies showed that using $\{M, N\} = 2$ in Eq. (8) was sufficient to achieve a reasonable estimate of the buckling load. To generate sufficient levels of reliability with the fewest number of function evaluations, the GA was run for 150 generations (fixed) using the following set of genetic operator implementation probabilities: crossover = 1.0, mutation = 0.01, gene swap = 0.9. Numerical experiments involved varying the population size from 15 to 75 in increments of 10 to determine the maximum number of global optima that could be found by both the SGA and GEGA. Once again, the value of N_k was varied from 1 to $P/2$ for the ME and VE selection schemes. Average

Table 2
Global optima for buckling load maximization problem ($a/b = 2$)

Selection scheme (N_k)	P	R	P_r[a]	# of optima found over 50 runs 1	2	3	4	5	6	A_{N_o}
EL	15	0.78	0.072	38	1	0	0	0	0	0.8
EL	45	1.00	0.115	46	4	0	0	0	0	1.08
EL	75	1.00	0.069	43	6	1	0	0	0	1.16
ME_1 (2)	15	0.80	0.128	16	24	0	0	0	0	1.28
ME_1 (4)	45	1.00	0.090	0	6	15	29	0	0	3.46
ME_1 (5)	45	1.00	0.112	0	11	11	27	1	0	3.36
ME_1 (10)	75	1.00	0.134	0	2	4	40	4	0	3.92
ME_2 (2)	15	0.84	0.181	14	16	12	0	0	0	1.64
ME_2 (5)	45	1.00	0.115	2	6	14	28	0	0	3.36
ME_2 (5)	75	1.00	0.069	0	0	2	46	2	0	4.00
VE[b] (2)	15	0.36	0.073	13	5	0	0	0	0	0.46
VE[b] (5)	45	0.88	0.125	11	13	16	4	0	0	2.02
VE[b] (5)	75	1.00	0.117	7	15	13	13	1	1	2.78

[a] $\Delta f = 4.53 \times 10^{-6}$.
[b] N_k set to P after 140 generations.

values were calculated over 50 optimization runs for all results presented.

Table 2 lists the number of optima found over fifty runs, and the average number of optima found per run. The EL scheme rarely found more than one global optimum per run, while the ME and VE schemes were successful in finding multiple optima. Although the GA requires a greater number of function evaluations to find multiple optima (Fig. 4), the cost per optimum found increases only slightly (Fig. 5). The ME_2 selection scheme found a maximum of four optima per run on average using a population size of 75. Although this required over 11,000 function evaluations, the ME_2 had a lower cost per optimum found when compared to the EL using a population size of 15. The ME_1 scheme was the most efficient and found a maximum of 3.92 optima per run using a population size of 75. Note that runs which use a larger population size can accommodate larger values of N_k because a sufficient number of child designs still comprise each new generation, preventing the genetic search from becoming too localized. Typical sets of global optima found by both the ME_1 and ME_2 schemes are given in Table 3. From an engineering standpoint, designs with ±45° plies on the outer (left) edge of the laminate may be preferred for their improved damage tolerance.

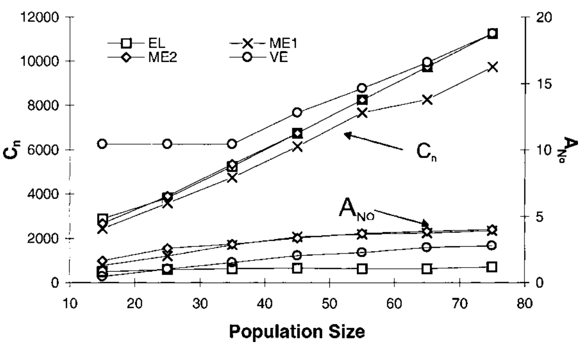

Fig. 4. Normalized cost and average optimum found/run vs. population size.

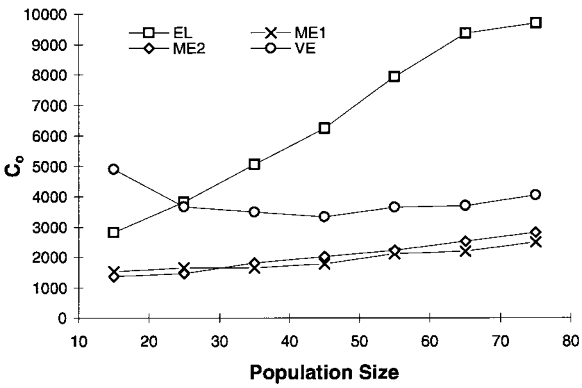

Fig. 5. Cost per optimum vs. population size.

Early test results revealed the existence of a small group of designs very close to optimal. Thus, richness values were calculated for $\Delta f = 4.53 \times 10^{-6}$. Richness values achieved using the ME are directly related to the value of N_k. For example, the ME_1 achieved a population richness of 0.128 which is approximately 2 (the value of N_k in this case) designs found per run within Δf of optimal. It was expected that the VE schemes would

Table 3
Optimal designs found during an optimization run ($P = 75$, $a/b = 2$, 64 plies)

Selection scheme (N_k)	Design	Buckling load factor
ME_1 (10)	$[\pm 45/90_{10}/\pm 45/90_8/\pm 45/90_8]_s$	3973.01
	$[90_2/\pm 45/90_6/\pm 45/90_8/\pm 45/90_{10}]_s$	3973.01
	$[\pm 45/90_8/\pm 45/90_{18}/\pm 45]_s$	3973.01
	$[90_{10}/\pm 45_2/90_2/\pm 45_3/90_2/\pm 45_4]_s$	3973.01
ME_2 (5)	$[90_4/\pm 45_2/90_{16}/\pm 45/90_6]_s$	3973.01
	$[\pm 45/90_{10}/\pm 45/90_8/\pm 45/90_8]_s$	3973.01
	$[90_2/\pm 45/90_6/\pm 45/90_8/\pm 45/90_{10}]_s$	3973.01
	$[90_5/\pm 45/90/\pm 45_7/90_2/\pm 45]_s$	3973.01

find a large number of optima per run with high population richness. However, a highly exploitative GA (resulting from a large value for N_k) appears to reduce the performance of the GA considerably for this problem, even though the value of N_k was set to the population size late (at generation number 140) in the search.

Plotting A_{N_o} versus N_k shows that there is an "optimal" choice for N_k ($N_k = 4$ for this problem). For small N_k, the GA can preserve a number of designs with good fitnesses from different areas of the design space, while still retaining a large portion of the population to search other areas. However, as N_k continues to increase, too many designs are preserved and the GA loses its ability to effectively search the design space for global optima.

6. Composite laminate design: twist angle maximization

In this section, a simple study is conducted to show how the ME and VE selection schemes can be useful in solving other types of composite laminate design optimization problems. Maximizing the twisting displacement of a cantilevered composite plate is one such problem which provides difficulties for the SGA. The design space contains a single global optimum that is surrounded by a large number of designs that are very close to optimal. While the SGA is prone to prematurely converge to one of the nearly optimum points, the exploitative characteristics of the GEGA help to find the global optimum design.

6.1. Plate analysis

The cantilevered plate is 14.0 in. long by 5.5 in. wide and loaded under a pure bending moment ($M_x^a = 1200.0$ lb) applied along the right end of the panel (Fig. 6). The stacking sequence is assumed to be symmetric about the laminate mid-plane and comprised of glass–epoxy plies ($E_1 = 5.6E6$ psi; $E_2 = 2.0E6$ psi; $G_{12} = 0.7E6$ psi; $\nu_{12} = 0.2$; $t = 0.011$ in.; $X_t = 3.0E4$ psi; $Y_t = 4.0E3$ psi; $X_c = 3.5E4$ psi; $Y_c = 9.0E3$ psi; $S = 1.5E4$ psi). Each ply in the stacking sequence may be oriented at any angle between 0° and 90° in increments of 10° as shown in Fig. 6.

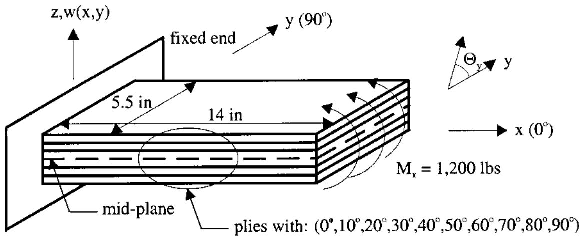

Fig. 6. Plate configuration and loading conditions.

This type of composite plate problem is often used as a simplified model of wing or hydrofoil structures. Aerodynamic loading caused by a wind gust or certain aircraft maneuvers is simulated by the applied moment at the tip of the plate, causing the wing to bend and/or twist. An unbalanced laminate stacking sequence allows the plate to twist as it bends under the influence of the moment applied at the plate tip. The GA is used to design the stacking sequence of the plate for maximum opposite twist at the plate tip, reducing or eliminating the effect of the aerodynamic loading or maneuver (similar research on this topic was done by Malott and Averill [20]).

Panel analysis is based on CLT. For the case of pure bending (i.e., $M_y = M_{xy} = 0$), the twist angle Θ_y (Fig. 6) of the plate can be determined by

$$\Theta_y = \frac{\partial w}{\partial y} = -\frac{x}{2} M_x d_{16}, \tag{10}$$

where $[d]$ is the inverse of the bending stiffness $[D]$ matrix. The twist angle is a function of x only and has a maximum value at the tip of the plate. The Tsai–Hill failure criterion is used to ensure that the material does not fail, see Jones [19]. For the kth ply in the laminate, the Tsai–Hill criterion, P_{th}^k, is determined by

$$P_{th}^k = \frac{(\sigma_1^k)^2}{X^2} - \frac{\sigma_1^k \sigma_2^k}{X^2} + \frac{(\sigma_2^k)^2}{Y^2} + \frac{(\tau_{12}^k)^2}{S^2}, \tag{11}$$

where X, Y, and S are the longitudinal, transverse, and shear strengths for glass–epoxy, respectively. Values for the material strength constants used in Eq. (11) depend on whether the ply is in tension or compression. Material failure is assumed to occur if Eq. (11) yields values greater than one. Note that for a more detailed analysis, the Tsai–Hill criterion would be enforced at the plate root where stresses would be critical.

6.2. Genetic code

One string of genes is used to represent one half of the symmetrically laminated composite plate. Each gene in the string is an integer between 0 and 10, where 0 represents one empty ply, 1 represents one ply oriented at 0°, 2 represents a ply oriented at 10°, and so on. Empty plies are used because the number of genes is fixed while the number of plies in a laminate is not.

To accommodate variable thickness laminates, some modifications to the existing genetic operators were required and two additional genetic operators were incorporated into the GA procedure. First, all empty plies are pushed to the left end (outer edge) of the laminate to eliminate voids in the stacking sequence. The crossover operator was modified to restrict the random crossover point to fall in the nonempty region of both parent laminates. Mutation and gene swap remain unchanged but are restricted from operating on empty genes. Ply addition and deletion operators (developed by Le Riche [13]) were added to manipulate ply thickness. To add a ply, a uniform random number is chosen to determine the orientation. Since the outer plies get set up faster than the inner plies due to their greater influence on the objective function, added plies are always introduced at the mid-plane of the laminate. The ply addition operator can only be implemented if there is at least one empty ply in the laminate stacking sequence. The second operator is ply deletion. To delete a ply, a random number is chosen and the corresponding ply is removed from the stacking sequence by replacing it with a 0 gene. Ply deletion can only be applied if there are at least two filled plies in the stacking sequence. After conducting several numerical experiments to tune the algorithm, a set of genetic operator probabilities were determined: crossover = 1.0, mutation = 0.02, ply addition = 0.05, ply deletion = 0.10, gene swap = 0.8.

6.3. Optimization formulation

The goal of the optimization is to find the thinnest stacking sequence of the plate that will yield the greatest twist angle at the end of the plate, due to the applied bending moment, without laminate failure due to excessive stress. The stacking sequence is also constrained to have no more than three contiguous plies with the same orientation to avoid problems with matrix cracking. Once again, the GA works with a string that corresponds to one-half of the laminate stacking sequence only, thereby automatically satisfying the symmetry constraint.

The optimization problem can be formulated as

$$\text{maximize } \Theta_y \quad \text{such that } \max_k \, P_{th}^k \leqslant 1. \tag{12}$$

Due to the nature of the problem (maximization of the twisting displacement), the weight of the laminate does not have to be incorporated into the optimization formulation as laminates with fewer plies always produce the maximum deflections. The minimum weight of a laminate will be governed by the Tsai–Hill strength constraint. If a laminate is too thin, then the material strength constraint will be violated making the laminate design less desirable. The thinnest laminates that do not violate any constraint will yield the largest twist angles at the end of the plate and the best performance.

To apply the genetic algorithm, the degree of constraint violation must be transformed into penalty parameters which augment the unconstrained objective function [21,22]. For this problem, the numerical values of the penalty parameters are used to scale the value of the twist angle at the plate tip. The augmented function is defined as

$$\Phi = \begin{cases} \frac{\Theta_y}{P_c}, & \max_k \, P_{th}^k \leqslant 1, \\ \frac{\Theta_y}{P_c \, (P_{th}^k)^\lambda}, & \max_k \, P_{th}^k > 1, \end{cases} \tag{13}$$

where P_c is the ply contiguity constraint and is defined as

$$P_c = 1 + n_c. \tag{14}$$

The variable n_c used in Eq. (14) is the number of contiguous plies with the same orientation in half the laminate. If the strength constraint is not violated, then the first expression in Eq. (13) is used to determine the value of the objective function; otherwise, the second expression is used. While testing the algorithm, it was found that very thin laminates appeared desirable even when the strength constraint was being violated. This problem was handled by adding the exponent λ to the scale factor. The value of λ may vary depending on the thickness of a ply. For the given value of t, λ is set to a value of 2.

6.4. Results

The maximum number of plies in the laminate stacking sequence is 60, giving $11^{30} \cong 1.74 \times 10^{31}$ possible laminate designs. The stacking sequence, strength constraint, and twist angle for the best known design are

Table 4
Properties of best known laminate design

Optimum design	P_{th}^{k}	Θ_y
$[E_{10}/(10/20)_2/10_3/20/10_2/20_2/10_3/20/10_2/20/10]_s$	1.0000	2.15°

given in Table 4. Due to the large number of possible laminate designs, the size of the population was set to 200 and the GA was run using a convergence criterion (instead of a fixed number of generations) of 700 generations without improvement in an effort to obtain acceptable levels of reliability. Fifty optimization runs were conducted to monitor reliability, final population richness, and computational effort.

Fig. 7 shows a comparison of reliability and richness for the EL and ME selection methods, with N_k taking on odd values from 1 to 99 for ME selection. The EL scheme found the optimal stacking sequence only once in 50 runs. Reliability values for ME selection are lowest for the smallest values of N_k since the EL and ME schemes are most similar in these cases. However, as N_k increases to values between 15 and 30, reliability values increase substantially with ME_1 and ME_2 peaking at 0.72 ($N_k = 19$) and 0.82 ($N_k = 19$), respectively. Richness values were substantially higher for the ME schemes also. The exploitative characteristics inherent in the ME schemes for $15 \leqslant N_k \leqslant 30$ are responsible for improved performance, permitting the GA to rigorously explore the design space around the optimum.

The EL scheme can only find approximately 100 laminates that are within 0.05% of the optimal design. In contrast, the ME_2 selection scheme ($N_k = 19$) finds 150 laminates that match the fitness of the optimal laminate and approximately 750 laminates with fitnesses that are within 0.01% of optimum. In runs where the GA finds the optimal laminate, other designs are good enough to be considered optimal due to the accuracy of the fitness calculation. This is why reliability is 0.82 and not 1.00 for ME_2 selection ($N_k = 19$), even though 150 optimal designs were found.

To implement the VE scheme for this problem, the value of N_k was increased to the maximum value after 600 iterations without improvement of the best design. Reliability reached as high as 0.98 when using VE_1 selection ($N_k = 1$, see Fig. 8), an improvement of over 0.90 when compared to ME_1 selection for $N_k = 1$. Thus, for problems where many nearly optimum designs are located around the optimum, utilizing both a highly explorative search at the beginning ($N_k = 1$), and a more focused search towards the end of the optimization run ($N_k = P = 200$), is highly beneficial. Richness values for VE selection are also shown in Fig. 8. The robust nature of the VE scheme is illustrated here, achieving a richness value of 1.00 for $N_k = (1, 2, 3)$, showing that there are numerous nearly optimum designs for this problem. For a value of $N_k = 1$, VE selection found optimal designs over 250 times and found laminates with fitnesses no more than 0.015% from optimal almost 10,000 times. Thus, almost every design found by the GA in the final population of each run was extremely close to optimal when using VE selection.

The average computational cost over fifty runs is depicted in Fig. 9 for each selection scheme. These results show that both ME selection schemes (for $20 \leqslant N_k \leqslant 30$) provide higher reliability and richness without requiring additional computational effort when compared to EL selection. The highest levels of reliability and richness achieved using VE selection ($1 \leqslant N_k \leqslant 10$) required a substantial increase in computational effort. However, for values in the range $10 < N_k \leqslant 30$,

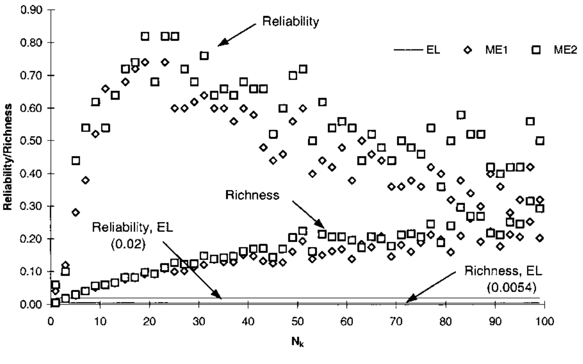

Fig. 7. Reliability/richness comparison – EL vs. ME_1 and ME_2 ($P = 200$).

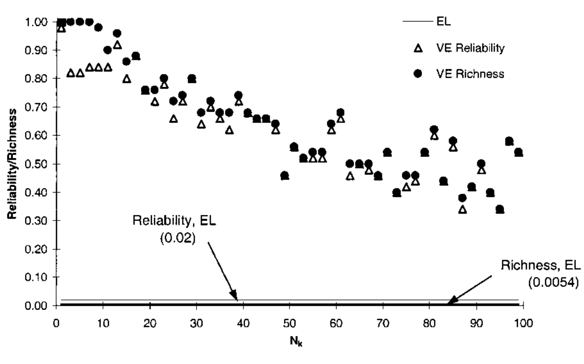

Fig. 8. Reliability/richness comparison – EL vs. VE selection ($P = 200$).

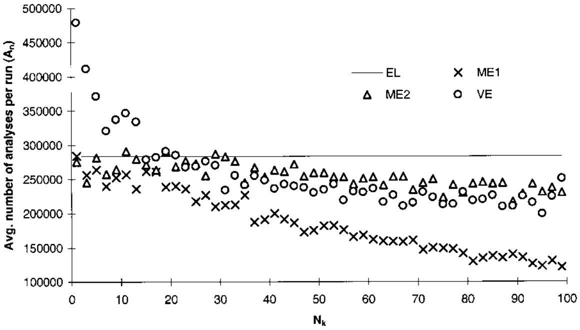

Fig. 9. Cost comparison – EL vs. ME_1, ME_2, and VE selection ($P = 200$).

VE selection can provide reasonable levels of reliability and richness at costs comparable to those found using EL selection.

7. Conclusions

Simple modifications were made to improve the performance of the SGA in two composite laminate design optimization problems. The first problem has multiple global optima while the second has a single optimum surrounded by a large number of near optima. The SGA was unable to find more than one optimum per optimization run in the first problem, and converged prematurely to nearly optimum designs in the second problem. The GEGA utilizes ME and VE selection schemes which preserve a certain number (N_k) of good designs from previous generations. For the first design problem, buckling load maximization of a laminated plate, best results are produced when the value of N_k is relatively small. This allows the GEGA to find a larger number of designs with good fitnesses from different areas of the design space, while still retaining a large portion of the population to search other areas. Although the number of function evaluations required to find multiple optima was substantially higher, the cost per optimum found using the ME and VE schemes was lower than the EL schemes. In the second problem, larger values of N_k were used to increase the exploitative characteristics of the genetic search. This enabled the GEGA with VE selection to search through a condensed area of nearly optimum designs and find the global optimum without requiring additional computational effort.

Acknowledgements

This research was supported in part by the Air Force Office of Scientific Research Grant F49620-96-1-0104.

References

[1] Jacoby SLS, Kowalik JS, Pizzo JT. Iterative methods for nonlinear optimization problems. Englewood Cliffs, NJ: Prentice Hall; 1972.
[2] Gürdal Z, Haftka RT. Optimization of composite laminates. Presented at the NATO Advanced Study Institute on Optimization of Large Structural Systems, Berchtesgaden, Germany, September 23–October 4, 1991.
[3] Holland JH. Adaptation in natural and artificial systems. Ann Arbor, MI: The University of Michigan Press; 1975.
[4] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison–Wesley; 1989.
[5] De Jong KA. Analysis of the behavior of a class of genetic adaptive systems. Doctoral dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.
[6] Baker JE. Reducing bias and inefficiency in the selection algorithm. Proceedings of the Second International Conference on Genetic Algorithms and Their Applications. Cambridge, MA, 1987. p. 14–21.
[7] Callahan KJ, Weeks GE. Optimum design of composite laminates using genetic algorithms. Compos Engng 1992; 2(3):149–60.
[8] Le Riche R, Haftka RT. Optimization of stacking sequence design for buckling load maximization by genetic algorithm. AIAA J 1993;31(5):951–6.
[9] Ball NR, Sargent PM, Ige DO. Genetic algorithm representations for laminate layups. Artif Intell Engng 1993; 8(2):99–108.
[10] Nagendra S, Haftka RT, Gürdal Z, Watson LT. Design of Blade Stiffened Composite Panels by Genetic Algorithm. Proceeding of the 34th Structures, Structural Dynamics, and Materials Conference. La Jolla, CA, April 19–21, 1993. p. 2418–36.
[11] Harik GR. Finding multimodal solutions using restricted tournament selection. Proceedings of the Sixth International Conference on Genetic Algorithms. San Francisco, CA: Morgan Kaufmann; 1995. p. 24–31.
[12] Goldberg DE, Richardson J. Genetic algorithms with sharing for multi-modal function optimization. Proceedings of the Second International Conference on Genetic Algorithms. Cambridge, MA: Morgan Kaufmann, 1987. p. 41–9.
[13] Le Riche R, Haftka RT. Improved Genetic Algorithm for Minimum Thickness Composite Laminate Design. Proceedings of the International Conference on Composite Engineering, August 28–31, New Orleans, LA, 1994.
[14] Baker JE, Grefenstette JJ. How Genetic Algorithms Work: a critical look at implicit parallelism. Proceedings of the Third International Conference on Genetic Algorithms. Fairfax, VA: Morgan Kaufmann; June 1989. p. 20–7.
[15] Nagendra S, Jestin D, Gürdal Z, Haftka RT, Watson LT. Improved genetic algorithm for the design of stiffened composite panels. Comput Struct 1996;58:543–55.
[16] Soremekun G. Genetic algorithms for composite laminate design and optimization. MS Thesis, Department of Engineering Science Mechanics, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1997.

[17] Bäck T, Hoffmesiter F. Extended selection mechanisms in genetic algorithms. Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufmann, 1991. p. 92–9.
[18] Bäck T. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. Proceedings of the First International IEEE Conference on Evolutionary Computation, IEEE, 1994. p. 57–62.
[19] Jones RM. Mechanics of Composite Materials. New York: Hemisphere; 1975.
[20] Malott B, Averill RC. Use of genetic algorithms for optimal design of laminated sandwich panels with bend twisting coupling. AIAA Paper 96-1538-CP, 1996.
[21] Gürdal Z, Haftka RT, Nagendra S. Genetic algorithm for the design of laminated composite panels. SAMPE J 1994;30(3):29–35.
[22] Gürdal Z, Haftka RT, Hajela P. Deisgn and Optimization of Laminated Composite Materials. New York: Wiley; 1991.