Genetic Algorithms with Local Improvement for Composite Laminate Design

Nozomu Kogiso, Layne T. Watson,
Zafer Gürdal, and Raphael T. Haftka

TR 93-17

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

May 28, 1993

Genetic Algorithms with Local Improvement for Composite Laminate Design

Nozomu Kogiso
Graduate School, University of Osaka Prefecture, 1-1 Gakuen-cho Sakai, 593 Japan
Layne T. Watson, Zafer Gürdal, Raphael T. Haftka
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA

ABSTRACT.

This paper describes the application of a genetic algorithm to the stacking sequence optimization of a composite laminate plate for buckling load maximization. Two approaches for reducing the number of analyses required by the genetic algorithm are described. First, a binary tree is used to store designs, affording an efficient way to retrieve them and thereby avoid repeated analyses of designs that appeared in previous generations. Second, a local improvement scheme based on approximations in terms of lamination parameters is introduced. Two lamination parameters are sufficient to define the flexural stiffness and hence the buckling load of a balanced, symmetrically laminated plate. Results were obtained for rectangular graphite-epoxy plates under biaxial in-plane loading. The proposed improvements are shown to reduce significantly the number of analyses required for the genetic optimization.

INTRODUCTION.

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°. Designing the laminate then becomes a stacking sequence optimization problem which can be formulated as an integer programming problem.

The laminate stacking sequence design problem with frequency constraints has been formulated by Mesquita and Kamat [2] with the numbers of plies as the design variables, leading to a nonlinear integer programming problem. More recently, Haftka and Walsh [3] showed that the use of ply identity design variables linearizes the integer programming formulation of the stacking sequence buckling maximization design problem. However, when strength constraints are also considered, the problem becomes nonlinear again and has been solved by Nagendra et al. [4] with a sequence of linearized integer programming problems. The branch and bound algorithm was used to solve the integer programming problems in Refs. [2 - 4]. More recently, Le Riche and Haftka [5] solved this problem by genetic algorithms (GA).

An early implementation of genetic search methods is credited to Rechenberg [6], although Holland's work [7] has provided the theoretical basis of most contemporary developments. Genetic algorithms are stochastic optimization methods [8-12] that work on a population of designs by recombining the most desirable features of existing designs. Following the evolutionary concept of survival of the fittest, selection of mates favors the fittest members (designs) of the population, and offspring (newly created designs) are created by splicing together features (genes) of the parent designs. Additionally, genetic mutation is used to create new design features. Genetic algorithms do not use any gradient information, and thus are particularly suited for problems (such as discrete optimization) where derivatives are not available. In the last decade, genetic algorithms have proven their ability to deal with a large class of combinatorial problems. In structural optimization applications, GAs have appeared only recently [5, 13-15].

Despite their numerous advantages, a serious drawback of GAs is their high computational cost. Genetic algorithms usually require a large number of analyses, sometimes in the range of thousands or even millions. Therefore, improvements in both the efficiency of the analysis and the execution of the GA are needed in order to make the genetic optimization affordable. The objective of the present work is to reduce the cost of the genetic search for optimization of composite panels. We propose the use of a binary tree data structure to store the results of all new analyses performed during optimization, and retrieve the information for designs that appeared in previous iterations. We also propose an approximation of the buckling load based on two lamination parameters. After evaluating exactly the buckling load for each design created by the genetic operators, new design strings are created by trying all possible exchanges of the locations of pairs of plies in the laminate. Then the buckling loads for these new designs are estimated using the approximation, and the best of these designs replaces the nominal design. By searching for a local optimum in a small neighborhood of the nominal design, we try to improve the performance on combinatorial optimization problems. This local improvement (searching for a local optimum in a small neghorhood) was used on combinatorial problems by Refs. [16,17]. We apply this idea to improve the performance of genetic optimization for composite panel design.

The efficiency of this local improvement for the genetic search, as well as the use of the binary tree to retrieve previously analyzed designs, are investigated for buckling load maximization of a rectangular 48-ply unstiffened laminated composite plate subjected to biaxial in-plain loads. The analysis cost associated with this problem is low enough so that thousands of genetic optimizations can be carried out for the purpose of averaging out the randomness in the performance of a single genetic optimization.

Analysis and Lamination Parameters.

The simply supported plate, shown in Figure 1, has longitudinal and lateral dimensions a and b, respectively, and is loaded in the x and y directions by λN_x, λN_y, respectively, where λ is a load amplitude parameter. The laminate is composed of N plies and assumed to be a symmetric, balanced laminate, made up of 0°, 90°, and ±45° plies of thickness t each. To reduce the number of design variables and enforce the balanced condition, the laminate is constrained to be made up of stacks of two 0° plies, two 90° plies, or a +45° and -45° pair of plies. These stacks are denoted by $0°_2$, $90°_2$, or ±45°. Taking into account the symmetry, only N/4 ply orientations are required to define the entire laminate.

For a simply supported plate under biaxial compression loading, the plate buckles when the load amplitude parameter λ reaches a critical value λ_{cb} given as

$$\frac{\lambda_{cb}(m,n)}{\pi^2} = \frac{D_{11}\left(\frac{m}{a}\right)^4 + 2(D_{12}+2D_{66})\left(\frac{m}{a}\right)^2\left(\frac{n}{b}\right)^2 + D_{22}\left(\frac{n}{b}\right)^4}{\left(\frac{m}{a}\right)^2 N_x + \left(\frac{n}{b}\right)^2 N_y} \quad , \qquad (1)$$

where m and n are the number of half waves in the x and y directions, respectively, that minimize λ_{cb}. The D_{ij}s are the flexural stiffnesses, which depend on the lamination sequence. When a single material is used, the flexural stiffnesses can be expressed in terms of only two lamination sequence parameters and material constants (e.g., Miki and Sugiyama [18]).

$$D_{11} = \frac{h^3}{12}\left(U_1 + U_2 W_1^* + U_3 W_2^*\right) ,$$

$$D_{22} = \frac{h^3}{12}\left(U_1 - U_2 W_1^* + U_3 W_2^*\right) ,$$

(2)

$$D_{12} = \frac{h^3}{12}\left(U_4 - U_3 W_2^*\right) ,$$

$$D_{66} = \frac{h^3}{12}\left(U_5 - U_3 W_2^*\right) ,$$

where U_i (i=1,...,5) are the material constants, h is the total plate thickness, and W_1* and W_2* are the bending lamination parameters defined as

$$W_1^* = \frac{12}{h^3}\left(\int_{-h/2}^{h/2} z^2 \cos 2\theta dz\right),$$

(3)

$$W_2^* = \frac{12}{h^3}\left(\int_{-h/2}^{h/2} z^2 \cos 4\theta dz\right),$$

where θ is the ply orientation angle. The flexural stiffnesses D_{16} and D_{26} are assumed to be negligible.

The optimization problem is to maximize the critical buckling load by changing the laminate stacking sequence. Additionally, strain constraints are applied, and the number of contiguous plies of the same orientation is limited to four plies to alleviate matrix cracking problems.

The strain failure constraint requires all strains to remain below their allowable limits. In our case γ_{xy} is zero, and the laminate strains are related to the loads on the plate by the relations

$$\lambda N_x = A_{11}\varepsilon_x + A_{12}\varepsilon_y \ ,$$

(4)

$$\lambda N_y = A_{12}\varepsilon_x + A_{22}\varepsilon_y \ .$$

The strains in the ith layer are obtained from the laminate strains by

$$\varepsilon_{1i} = \cos^2\theta_i\varepsilon_x + \sin^2\theta_i\varepsilon_y \ ,$$

$$\varepsilon_{2i} = \sin^2\theta_i\varepsilon_x + \cos^2\theta_i\varepsilon_y \ , \tag{5}$$

$$\gamma_{12i} = \sin 2\theta_i\left(\varepsilon_y - \varepsilon_x\right) \ ,$$

where the A_{ij}s are the in-plane stiffnesses, and θ_i is the ply orientation angle of the ith ply. These stiffnesses can also be expressed in terms of in-plane lamination parameters and material constants as

$$A_{11} = h\left(U_1 + U_2V_1^* + U_3V_2^*\right) \ ,$$

$$A_{22} = h\left(U_1 - U_2V_1^* + U_3V_2^*\right) \ , \tag{6}$$

$$A_{12} = h\left(U_4 - U_3V_2^*\right) \ ,$$

where the in-plane lamination parameters are defined as

$$V_1^* = \frac{1}{h}\left(\int_{-h/2}^{h/2}\cos 2\theta dz\right),$$

(7)

$$V_2^* = \frac{1}{h}\left(\int_{-h/2}^{h/2}\cos 4\theta dz\right).$$

The strain failure load λ_{cs} is the smallest load factor λ such that one of the principal strains in one of the layers is at its allowable value.

The ply contiguity constraint is implemented by a penalty parameter p (=0.9 here) which reduces the objective function when the contiguity constraint is violated. Based on the above discussion, the objective function λ^* is given as

$$\lambda^*_c = p^n \min(\lambda_{cs}, \lambda_{cb}) \,, \tag{8}$$

where n is the number of contiguous plies in excess of the constraint value of four.

GENETIC ALGORITHM.

For a genetic algorithm, each design must be coded as a finite string of digits. In the present work, a 0°_2 stack is assigned the digit 1, a ±45° stack the digit 2, and a 90°_2 stack the digit 3. For example, the laminate $[90^\circ_2/\pm45^\circ_2/90^\circ_2/0^\circ_2/\pm45^\circ_2/0^\circ_2]_S$ is encoded as 1 2 2 1 3 2 2 3. The leftmost 1 corresponds to the layer closest to the laminate plane of symmetry. The rightmost 3 describes the outermost layer.

Figure 2 shows the pseudocode for the algorithm. The genetic search begins with the random generation of a population of design alternatives. Each individual has a fitness value based on its objective function that determines its probability for selection as a parent. Parents exchange parts of their genes (strings) in a process called crossover to create offspring (new design strings). Additional genetic operators, namely mutation and permutation, are applied to the child designs which then replace the parent generation.

In this study, a two point interchange operator is introduced in order to produce neighboring designs in which two digits of the design string are interchanged. The local improvement procedure replaces each design generated by the genetic operators by the estimated best of the neighbor designs.

Here, the best design is always carried to the next generation, which is an "elitist plan" version of the genetic algorithm. The optimization process is repeated until some specified number of generations provide no improvement in the best design.

Efficient Retrieval of Designs by a Binary Tree.

During the evolution process, populations often contain designs that have also appeared in previous generations. If the calculation of the objective function is expensive, it is worthwhile to keep track of designs to avoid duplicate calculation. The binary tree data structure [19, pp. 139-143] provides a way to store previous designs which permits efficient search for duplicate designs. The tree is used to store all data pertinent to the design: the design string, in-plane lamination parameters, bending lamination parameters, strain failure load, buckling load, and objective function.

Figure 3 shows the calculation of the objective function with the aid of the binary tree. After a new generation of design strings is created by the genetic operations, the binary tree is searched for each new design. If the design is found, the objective function value is obtained from the tree without analysis. Otherwise, the tree is searched for designs with identical in-plane lamination parameters, and hence identical in-plain strains. If a design with identical in-plain lamination parameters is found, then the strain failure load is obtained from the tree. Otherwise, the strain failure value is obtained by exact analysis. Then the buckling load is calculated, and finally, the objective function value is adjusted for the ply contiguity constraint. This new design and its concomitant data are then inserted in the tree.

Local Improvement.

Local improvement is used to improve the performance of combinatorial optimization algorithms, by searching for a local optimum in a small neighborhood of the nominal design. The present work considers as neighbors all the designs obtained by interchanging two stacks in the laminate. An example of a stack interchange is

$$\text{nominal design} \quad : \quad 1\,2\,3\,\underline{1}\,3\,\underline{2}\,2\,1\,2\,2\,3\,1$$

(9)

$$\text{perturbed design} \quad : \quad 1\,2\,3\,\underline{2}\,3\,\underline{1}\,2\,1\,2\,2\,3\,1$$

The number of all possible different laminates obtained by interchanges is less than or equal to $\binom{n}{2} = n(n-1)/2$, where n is the string length.

Figure 4 shows an example of the distribution of the perturbed designs in the bending lamination parameter space. The central point corresponds to the nominal design, and the three branches correspond to the three possible exchanges ($1 \leftrightarrow 2$, $1 \leftrightarrow 3$, $2 \leftrightarrow 3$). The distance from the nominal point depends on the locations of the exchanged stacks.

The interchange operation does not change the in-plane stiffnesses and the strain failure load. Only the buckling load is influenced by this operation. To reduce the cost of evaluating all the possible interchanges, the buckling load at neighboring designs is estimated by a linear least squares approximation based on the bending lamination parameters.

$$\lambda = \lambda_0 + A\Delta W_1^* + B\Delta W_2^* \ , \tag{10}$$

where λ_0 is the buckling load of the nominal design, and ΔW_1^* and ΔW_2^* are the changes in the bending lamination parameters (W_1^*, W_2^*) from the nominal design. The coefficients A and B are determined as follows: first, the binary tree is searched for the five nearest neighbors of the nominal design in the Euclidean (W_1^*, W_2^*) plane. Then the coefficients are determined by the least squares fit of the form (10) to the five nearest neighbors.

The approximate buckling load is then used to evaluate the objective function for all the perturbed designs, and the best one is used to replace the nominal one. Figure 5 shows an example of the effectiveness of this local improvement. The first column of each pair is the objective function value of the nominal design, while the second column represents the best design which replaces that nominal design (which was calculated exactly for the purpose of this figure). The objective function is improved before going on to the next generation, and the design obtained by local improvement is used to produce the offspring of the next generation. The accuracy of the approximation depends on the distribution of the nearest neighbors. The nominal design of Figure 4 together with the five nearest neighbors are shown in Figure 6. The accuracy of the approximations for all the perturbed designs of Figure 4 is shown in Figure 7. The horizontal axis in Figure 7 is the distance from the nominal design in the bending lamination plane. The vertical axis is the buckling load normalized by the buckling load of the nominal design

$[90°_2/(\pm45°/0°_2)_4/\pm45°_2/90°_2]_S$. When the circles (approximate analysis) overlap the diamonds (exact analysis), the approximation works well. The approximation fails completely when it predicts an increase in buckling load while the true buckling load actually decreases or vice versa. Figure 7 shows that such failures tend to occur only when the change in the buckling load from the nominal design is small. Thus, it is unlikely that the design selected as the best among the perturbed designs on the basis of the approximation is worse than the nominal design.

Figure 8 shows the distribution of accuracy of all points from one genetic optimization run. The horizontal axis is the normalized improved value by approximation $(\lambda_{ap} - \lambda_{nom}) / \lambda_{nom}$, and the vertical axis is that by the exact analysis $(\lambda_{ex} - \lambda_{nom}) / \lambda_{nom}$. When the approximation works well, points lie close to the 45° line where λ_{ap} is equal to λ_{ex}. When the point lies above the line, the approximation underestimates the buckling load, otherwise, it overestimates the load. If the point is in the first or the third quadrants, the approximation predicts correctly whether the interchange increases or decreases the buckling load. When the point is in the second or the fourth quadrants, the approximation does not even capture the sense of the effect of the interchange. As can be seen from the figure, very few points lie in the bad quadrants, and these have small normalized values. So, these points are not likely to be selected as the best among the perturbed designs.

Variation of Failure Load and Buckling Load in the Lamination Parameter Planes.

The objective function for the genetic algorithm combines the buckling load and the strength failure load which are determined uniquely by the bending and in-plane lamination parameters, respectively. The variation of the failure load, as calculated from equations (4) - (7), as a function of V_1^* is shown in Figure 9 for a fixed value of V_2^*, $V_2^* = -0.5$. It can be seen that there is a singularity at the boundary where $V_2^* = -2V_1^*-1$. This boundary represents laminates which do not have any 0° plies. Points near this boundary have few 0° plies, and the strain in these plies is critical, reducing the failure load. When the last 0° ply is eliminated, the failure load increases suddenly because failure of 0° plies need not be considered. Such singularities are known to cause difficulties for continuous optimization algorithms because, unless the ply that causes the singularity is absent, the algorithm would tend to increase that ply's thickness rather than eliminate that ply. Genetic algorithms can circumvent singularities because they permit temporary degradation in performance. Thus a design with two zero stacks can change into one with only a single stack by mutation or crossover, even though this reduces the failure load. A subsequent mutation or crossover can then eliminate the remaining stack.

The distribution of the buckling load with respect to the bending lamination parameters is shown in Figure 10. The contour plot for each buckling mode is obtained from equations (1) - (3), and the contours are given by

$$W_2^* = \left[\pi^2 h^3 U_2\left(a^4n^4 - b^4m^4\right)W_1^* + 12\lambda a^2b^2\left(b^2m^2N_x + a^2n^2N_y\right) - \pi^2h^3\left\{U_1\left(a^4n^4 + b^4m^4\right) + 2a^2b^2m^2n^2\left(U_4 + 2U_5\right)\right\}\right] / \pi^2h^3\left(a^4n^4 - 6a^2b^2m^2n^2 + b^4m^4\right) \quad (11)$$

The contours in Figure 10 are piecewise linear with each segment corresponding to combinations of the wave numbers m and n that minimize the buckling load.

RESULTS.

Results were obtained for a 48-ply graphite-epoxy plate with the following material properties: $E_1 = 18.50E6$ psi (127.59 Gpa); $E_2 = 1.89E6$ psi (13.03 GPa); $G_{12} = 0.93E6$ psi (6.41 GPa); $\nu_{12} = 0.3$; $t = 0.005$ in. (0.127 mm). The ultimate allowable strains are $\varepsilon_1^{ua} = 0.008$, $\varepsilon_2^{ua} = 0.029$, $\gamma_{12}^{ua} = 0.015$. These allowable strains were reduced by a safety factor of 1.5. The plate has longitudinal and lateral dimensions of a = 20 in. (0.508 m) and b = 5 in. (0.127 m), respectively. Because of the symmetry and the use of 2-ply stacks, the 48-ply laminate is described by a 12-gene string. The genetic algorithm was applied to three load cases with N_y/N_x = 0.125, 0.25, and 0.5, called load cases 1, 2, and 3, respectively, where N_x is set to 1.0 lb/in (175 N/m).

In this problem there are many near optimal designs. For this reason, designs that are within a tenth of a percent of the global optimum are accepted as optimal and are called *practical optima* here. To evaluate the efficiency of the algorithm we define a *normalized price*, which is the average number of evaluations (also called *price*) of the objective function divided by the probability of reaching a practical optimum (called *practical reliability*).

Average prices and practical reliabilities were calculated by performing one hundred genetic optimizations for each of the three load cases. The algorithm was considered satisfactory only if the practical reliability was at least 0.8. This means that a single genetic optimization run has at least an 80 percent chance of finding a design within 0.1 percent of the global optimum. The requirement of 0.8 practical reliability was used to determine the stopping criterion. We start 100 optimization runs with the stopping criterion set to 10 generations without improvement. We then increase the stopping criterion until a practical reliability of 0.8 is achieved.

Performance of Genetic Algorithm without Local Improvement.

In [5], Le Riche and Haftka investigated the performance of a genetic algorithm without local improvement for the same problem with the same three load cases. They found that good performance was obtained with a population size of 8, probability of mutation of 0.01, probability of crossover of 1.0, probability of permutation of 1.0, and a penalty constant in the objective function of 0.9. The same values are used here. The performance of the algorithm depends strongly on the load case, as shown in Table 1. The table shows the stopping criterion (number of generations without improvement), the normalized price, and the practical reliability for the three load cases. For load case 2, the practical reliability did not reach 0.80 even with a stopping criterion of more than 60 generations, but in the other load cases, it reached 0.80 with a much lower stopping criterion. This behavior was investigated and found to depend on the nature of the optimum for each load case.

For the optimum design for load case 1, the failure load is critical while the buckling load is not. The failure load depends only on the ratio of total thicknesses associated with the ply orientation angles, and does not depend on the through-the-thickness location of the plies as long as the contiguous ply constraint is satisfied. Consequently, there are many optimum designs, some of which are shown in Table

2 (a). Therefore, it is easy to reach a practical optimum design and the price of the optimization is low.

For load case 2, there are only three practical optimum designs as shown in Table 2 (b). Two of these designs have critical failure load and one has critical buckling load. The failure load and the buckling load are very close at the optimum. Changes in the stacking sequence easily degrade either the buckling load or failure load, so there are few practical optimum designs, and it is more difficult to find one.

For load case 3, only the buckling load is critical for the practical optimum designs as shown in Table 2 (c). The optimum designs do not have any zero plies, and there is some freedom to change the ratio of the ±45° plies and 90° plies without degrading the buckling load significantly.

Effect of Binary Tree.

The efficiency of the binary tree for the objective function evaluation without local improvement is shown in Table 3. The second column in the table, the price, is the mean (over 100 runs) of the number of designs considered by the algorithm. Since an elitist strategy is adopted in our genetic algorithm implementation, the best design is passed on from one generation to the next, and the reanalysis of this design is not necessary. The elitist price is the mean of the number of analyses required by taking advantage of this property. Finally, the average number of nodes in the tree shows the mean of the number of different designs per optimization. The standard deviations of these means are also given in this table. It is clear from Table 3 that

20 to 50 percent of the analyses can be avoided by keeping track of previous designs.

Table 3 also shows that the number of nodes in the binary tree increases with permutation rate. This indicates that the permutation operator contributes to the creation of new design strings. Table 3 may indicate that if a binary tree is used to avoid reanalyses of designs the optimum permutation probability may be lowered.

Effect of Local Improvement.

Table 4 shows the performance of the local improvement with the same genetic parameters as in Table 1. For load cases 1 and 2, local improvement worked very well, especially for the load case 2, where the performance improved by about a factor of three.

For load case 3, however, local improvement made the performance worse (see tables). We investigated the reasons for the poor performance of the algorithm for load case 3. Table 5 shows frequently obtained designs by local improvement for this load case. In comparison with the practical optima found without the local improvement (see Table 2 (c)), these designs all have 0° plies near the midplane. The rest of the stacking sequence is the same as those of the practical optima which do not have any 0° plies.

Upon further investigation the problem was found to be due to the singularity of the optimum for load case 3. As discussed earlier, the failure load exhibits a singularity at the boundary of the lamination diagram (see Figure 9) where the laminate does not have any 0° plies. This situation is illustrated in Figure 11 which shows the effect of pushing the 0° plies toward the midplane and replacing them with ±45° stacks when they reach the midplane. As can be seen in Figure 11, the buckling load increases monotonically during this operation. However, the failure load decreases as we reduce the number of 0° plies until they are all eliminated, when it jumps up. Recall that this happens because we no longer need to enforce strain constraints in the 0° plies.

Crossover and mutation are two genetic operators that can potentially get rid of the undesirable layers from the laminate. Consider, for example, the following crossover scenario where both parents have 0° plies occupying different positions in the strings.

parent 1	2 2 / 1 1
parent 2	1 1 / 2 2
child	2 2 2 2

Local improvement interferes with this mechanism because it will move 0° plies towards the midplane, where they have the least detrimental effect on the buckling load, in all members of the population. For example, parent 1 in the above design is likely to be changed to

2 2 1 1 ⟶ 1 2 1 2

local improvement

Now the crossover cannot eliminate both 0° plies, because some of the 0° plies occupy the same location in both laminates and, therefore, appear in both children.

To ameriolate this situation, we seeded the initial population with "no-zero-ply" (NZP) designs. With the population sized fixed at 8, we tried two, four, or six initial NZP designs. This corresponds to 25%, 50% or 75% of the population.

The results obtained by this NZP seeding are shown in Table 6 for load case 3 for both with and without the local improvement and for two permutation probabilities. The performance is seen to improve both with and without local improvement as the number of NZP designs increase in the initial population. The effect on the average of all three load cases is shown in Table 7. The practical reliability reached 80% at a very small normalized price.

The comparison in Table 7 shows that while the NZP seeding procedure helps also without local improvement, its effect on the local improvement procedure is dramatic. With half of the initial population seeded NZP, the local improvement procedure reduced the price of the GA by better than a factor of two.

We also checked the need for permutation when seeding and local improvement are used. The results for 50% NZP seeding and three permutation probabilities are given in Table 8. It is seen that without local improvement permutation is critical for achieving the desired reliability. However, with local improvement it may not be necessary. Furthermore, if the performance of the algorithm is based on the number of different designs (nodes in the binary tree), no permutation could be the most efficient choice.

For the present problem, the calculations of the buckling load and the strength failure load involve the use of simple algebraic formulae, so that it takes longer to search for the nearest neighbors and construct the approximation than to evaluate the objective function. Also, the binary tree requires a large amount of memory to keep track of all the designs. Therefore the two approaches proposed, the binary tree and local improvement, are not cost-effective for this problem. However, these procedures have high potential and wide applicability to problems with a very expensive objective function.

CONCLUDING REMARKS.

This research introduced two approaches for reducing the number of analyses required by a genetic algorithm for the stacking sequence optimization of composite plates. The binary tree data structure is effective for avoiding the reanalysis of designs which appeared in previous generations. A local improvement scheme considered all designs that can be obtained by interchanging two stacks of plies in the nominal designs and estimated the buckling load of these designs using a linear approximation based on lamination parameters. This local improvement was found to substantially reduce the cost of the genetic optimization. We also found that to avoid difficulities with singular optima, the initial population needs to be seeded with designs containing no zero degree plies.

ACKNOWLEDGMENTS.

This work was supported in part by NASA grants NAG-1-168 and NAG-1-643, Department of Energy grant DE-FG05-88ER25068, Air Force Office of Scientific Research Grant F49620-92-J-0236, and a scholarship from the Osaka Foundation of International Exchange.

References.

1. Schmit, L. A., and Farshi, B., "Optimum Design of Laminated Fiber Composite Plates", International Journal for Numerical Methods In Engineering, Vol. 11, 1977, pp. 623 - 640.
2. Mesquita, L., and Kamat, M. P., "Optimization of Stiffened Laminated Composite Plates with Frequency Constraints", Engineering Optimization, Vol. 11, 1987, pp. 77 - 88.
3. Haftka, R. T., and Walsh, J. L., "Stacking-Sequence Optimization for Buckling of Laminated Plates by Integer Programming", AIAA Journal, Vol. 30, No. 3, 1992, pp. 814 - 819.
4. Nagendra, S., Haftka, R. T., and Gürdal, Z., "Stacking Sequence Optimization of SImply Supported Laminates with Stability and Strain Constraints", AIAA Journal, Vol. 30, No. 8, 1992, pp. 2132 - 2137.
5. Le Riche, R., and Haftka, R. T., "Optimization of Laminate Stacking Sequence For Buckling Load Maximization By Genetic Algorithm", Proceedings of the AIAA/ASME/ASCE/AHS/ASC 33rd Structures, Structural Dynamics and Material Conference, Dallas TX, April, 1992, Part. 5, pp. 2564 - 2575.
6. Rechenberg, J., "Cybernetic Solution Path of an Experimental Problem", Royal Aircraft Establishment, Liberty translation, 1122, Farnborough, England, 1965.
7. Holland, J. H., Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, 1975.
8. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of State Calculations by Fast Computing Machines,", J. Chem. Physics, Vol. 21, No. 6, 1953, pp. 1087-1092.
9. Rinnoy Kan, A. H. G., and Timmer, G. T., "Stochastic Methods for Global Optimization", American Journal of Mathematical and

Management Sciences, Vol. 4, 1984, pp. 7 - 40.
10. Byrd, R.H., Dert, C., Rinnoy Kan, A. H. G., and Schnabel, R. B., "Concurrent stochastic Methods for Global Optimization", Tech. Rep., CU-CS-338-86, Dept. Computer Sci., Univ. Colorado, 1986.
11. Eskow, E., and Schnabel, R. B., "Mathematical Modeling of a Parallel Global Optimization Algorithm", Tech. Rep., CU-CS-395-88, Dept. Computer Sci., Univ. Colorado, 1988.
12. Rinnoy Kan, A. H. G., and Timmer, G. T, "A Stochastic Approach to Global Optimization", Numerical Optimization 1984, Boggs, P., Byrd, R., and Schnabel, R. B., eds., SIAM, Philadelphia, 1985, pp. 245 - 262.
13. Hajela, P., "Genetic Search - An Approach To The Nonconvex Optimization Problem", AIAA Journal, Vol. 28, No. 7, 1990, pp. 1205 - 1210.
14. Rao, S. S., Pan, T. S., and Venkayya, V. B., "Optimal Placement Of Actuators In Actively Controlled Structures Using Genetic Algorithms", AIAA Journal, Vol. 28, No. 6, 1990, pp. 942 - 943.
15. Hajela, P. and Lin, C. Y., "Genetic Search Strategies In Multicriteria Optimal Design", Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Material Conference, Baltimore MD, April, 1991, Part. 2, pp. 354 - 363.
16. Tovey, C. A., "Low Order Polynominal Bounds on the Expected Performance of Local Improvement Algorithms", Mathematical Programming, Vol. 35, 1986, pp. 193 -224.
17. Tovey, C. A., "Hill Climbing with Multiple Local Optima", SIAM J. Alg. Disc. Meth., Vol. 6, No.3, July, 1985, pp. 384 - 393.
18. Miki, M., and Sugiyama, Y.,"Optimum Design of Laminated Composite Plates Using Lamination Parameters", Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Material Conference, Baltimore MD, April,1991, Part. 1, pp. 275 - 283.
19. Kernighan, B. W. and Ritchie, D. M., "The C Programming Language", Second Edition, Prentice Hall, Englewood Cliffs, N.J., 1988.

Table 1. Normalized prices and practical reliabilities without local improvement (probability of permutation = 1.0, population size = 8).

Load case	Stopping criterion	Normalized price	Practical reliability
1	19	350	0.84
	44	530	0.99
2	44	1126	0.71
	63	1250	0.78
3	38	832	0.81
	44	963	0.77
Average	44	836	0.823

Table 2. (a) Optimal designs for load case 1*.

Design variable		Load factor λ	
Stacking sequence	Genetic code	Buckling	Failure
$[\pm45_5/0_4/\pm45/0_4/90_2/0_2]_S$	131121122222	14659.583	13518.661
$[\pm45_5/0_4/90_2/0_4/\pm45/0_2]_S$	121131122222	14610.845	13518.661
$[\pm45_2/90_2/\pm45/(\pm45/0_4)_2/\pm45/0_2]_S$	121121122322	14421.311	13518.661
$[\pm45_4/0_2/\pm45/0_4/\pm45/0_4/90_2]_S$	311211212222	14284.145	13518.661
$[\pm45_4/0_2/\pm45/0_4/90_2/0_4/\pm45]_S$	211311212222	14251.656	13518.661
$[\pm45_3/0_2/\pm45_2/0_4/90_2/0_2/\pm45/0_2]_S$	121311221222	14029.490	13518.661
$[90_2/\pm45_2/(\pm45/0_2)_3/0_2/\pm45/0_2]_S$	121121212223	14013.722	13518.661
$[90_2/(\pm45_2/0_2)_2/\pm45/0_4/\pm45/0_2]_S$	121121221223	13831.525	13518.661
$[\pm45_3/(0_2/\pm45)_2/0_4/\pm45/0_2/90_2]_S$	312112121222	13744.604	13518.661

* There are many other practical optimum designs because the strain failure is critical.

Table 2. (b) Optimal and near-optimal* designs for load case 2.

Design variable		Load factor λ	
Stacking sequence	Genetic code	Buckling	Failure
$[\pm45_2/90_2/\pm45_3/0_2/\pm45/0_4/\pm45/0_2]_S$	121121222322	12743.451	12678.777
$[\pm45/90_2/\pm45_4/(0_2/\pm45/0_2)_2]_S$	121121222232	12725.257	12678.777
$[90_2/\pm45_5/(0_2/\pm45/0_2)_2]_S$	121121222223	12674.853	12678.777
$[\pm45_2/90_2/\pm45_3/0_4/(\pm45/0_2)_2]_S$	121211222322	12622.464	12678.777
$[\pm45/90_2/\pm45_4/0_4/(\pm45/0_2)_2]_S$	121211222232	12617.837	12678.777
$[\pm45_2/90_2/\pm45_3/(0_4/\pm45)_2]_S$	211211222322	12592.217	12678.777
$[\pm45/90_2/\pm45_4/(0_4/\pm45)_2]_S$	211211222232	12590.982	12678.777
$[\pm45_3/90_2/\pm45_2/(0_2/\pm45/0_2)_2]_S$	121121223222	12568.942	12678.777

* Only the top three designs are the practical optima.

Table 2. (c) Practical optimal designs for load case 3.

Design variable		Load factor λ	
Stacking sequence	Genetic code	Buckling	Failure
$[90_2/\pm45_2/(90_2/\pm45)_2/\pm45_5]_S$	222222323223	9998.198	10398.136
$[90_2/\pm45_2/(90_2/\pm45)_2/\pm45_4/90_2]_S$	322222323223	9997.614	10187.937
$[(90_2/\pm45_2)_2/(90_2/\pm45)_2/\pm45_2]_S$	222323223223	9997.614	10187.937
$[(90_2/\pm45)_2/\pm45_2/(\pm45/90_2/\pm45)_2]_S$	232232222323	9994.836	10187.937
$[\pm45/90_4/\pm45_2/90_2/\pm45_4/90_2/\pm45]_S$	232222322332	9994.836	10187.937
$[(\pm45/90_2)_2/90_2/\pm45_4/90_2/\pm45_2]_S$	223222233232	9994.836	10187.937
$[90_4/\pm45_7/90_2/\pm45_2]_S$	223222222233	9994.694	10398.136
$[90_4/\pm45_6/(\pm45/90_2)_2]_S$	323222222233	9994.110	10187.937
$[(90_2/\pm45)_2/\pm45_3/(90_2/\pm45)_2/\pm45]_S$	223232222323	9994.110	10187.937
$[\pm45/90_4/(\pm45_2/90_2/\pm45)_2/\pm45]_S$	223222322332	9994.110	10187.937
$[90_4/\pm45_7/90_4/\pm45]_S$	233222222233	9990.606	10187.937
$[\pm45/90_4/\pm45_3/90_4/\pm45_4]_S$	222233222332	9990.606	10187.937
$[90_2/\pm45_3/90_4/\pm45/90_2/\pm45_4]_S$	222232332223	9990.606	10187.937

Table 3. The efficiency of the binary tree; population size=8, stopping criterion=56 generations without improvement.

(a) probability of permutation=0.5.

Load case	Price	"Elitist" price*	Avg. no. of nodes in tree	% saving	Practical reliability
1	656.0 ± 11	575.0 ± 10	329.5 ± 6	42.7	0.99
2	937.0 ± 25	820.8 ± 22	467.8 ± 12	43.0	0.63
3	937.0 ± 27	820.8 ± 23	394.9 ± 12	51.9	0.74
Average	843.3 ± 15	738.9 ± 13	397.4 ± 7	46.2	0.786

(b) probability of permutation=1.0.

Load case	Price	"Elitist" price*	Avg. no. of nodes in tree	% savings	Practical reliability
1	646.2 ± 12	566.4 ± 11	453.0 ± 9	20.0	0.99
2	907.0 ± 27	794.7 ± 23	631.9 ± 18	20.5	0.71
3	918.5 ± 27	804.7 ± 23	504.4 ± 16	37.3	0.86
Average	823.9 ± 15	721.9 ± 13	529.7 ± 10	26.6	0.853

* Price excluding repeated analyses of best design (passed on to next generation with elitist strategy)

Table 4. Normalized price and practical reliability with local improvement (probability of permutation = 1.0, population size = 8).

Load case	Stopping criterion	Normalized price	Practical reliability
1	10	193	0.80
	50	520	1.00
2	22	435	0.81
	50	720	0.96
3	50	1646	0.46
	63	2209	0.45
Average	50	813	0.807

Table 5. Frequently obtained designs for load case 3 (with local improvement).

Design variable		Load factor λ	
Stacking sequence	Genetic code	Buckling	Failure
$[90_4/\pm45_6/(\pm45/0_2)_2]_s$	121222222233	9910.385	10251.197
$[(90_2/\pm45)_2/\pm45_3/90_2/0_4/90_2/0_2]_s$	131132222323	9803.273	10787.530
$[\pm45/90_2/(90_2/\pm45_2)_2/0_4/90_2/0_2]_s$	131122322332	9803.273	10787.530
$[(\pm45/90_2)_4/0_4/\pm45/0_2]_s$	121132323232	9803.273	10787.530
$[90_2/\pm45_2/(90_2/\pm45)_2/\pm45/0_4/\pm45/0_2]_s$	121122323223	9803.105	11404.473
$[(\pm45/90_2)_4/0_4/90_2/0_2]_s$	131132323232	9801.937	10193.772
$[90_4/\pm45_6/0_4/90_2/0_2]_s$	131122222233	9801.119	11404.473
$[90_2/\pm45_2/(90_2/\pm45)_2/\pm45/0_4/90_2/0_2]_s$	131122323223	9799.017	10787.530
$[\pm45/90_4/\pm45_3/90_4/0_4/\pm45/0_2]_s$	121133222332	9795.513	10787.530
$[90_2/\pm45_3/90_4/\pm45/90_2/0_4/\pm45/0_2]_s$	121132332223	9795.513	10787.530
$[(90_2/\pm45)_2/\pm45_2/90_2/\pm45/0_4/\pm45/0_2]_s$	121123222323	9792.593	11404.473
$[\pm45/90_4/\pm45_3/90_4/0_4/90_2/0_2]_s$	131133222332	9791.425	10193.772
$[90_2/\pm45_3/90_4/\pm45/90_2/0_4/90_2/0_2]_s$	131132332223	9791.425	10193.772

Table 6. Normalized price near 0.80 practical reliability for load case 3 starting with seeded nonzero degree plies in initial population.

No local improvement (permutation probability = 0.50)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	56	0.80	1181
6	2	32	0.83	566
4	4	22	0.85	378 (147)*
2	6	16	0.80	280 (104)*

Local improvement (permutation probability = 0.50)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	63	0.30	2997
6	2	63	0.73	1161
4	4	16	0.81	284 (131)*
2	6	16	0.85	218 (121)*

No local improvement (permutation probability = 1.00)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	38	0.81	832
6	2	25	0.83	496
4	4	13	0.83	249 (134)*
2	6	10	0.80	173 (97)*

Local improvement (permutation probability = 1.00)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	63	0.45	2209
6	2	56	0.83	943
4	4	13	0.80	256 (144)*
2	6	10	0.81	190 (109)*

* Average number of nodes in binary tree.

Table 7. Normalized price near 0.80 practical reliability for average of the three load cases starting with seeded nonzero degree plies in initial population.

No local improvement (permutation probability = 0.50)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	56	0.827	1034
6	2	44	0.810	821
4	4	32	0.820	602 (250)*
2	6	38	0.827	734 (292)*

Local improvement (permutation probability = 0.50)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	63	0.767	968
6	2	19	0.810	362
4	4	13	0.820	263 (135)*
2	6	16	0.860	287 (148)*

No local improvement (permutation probability = 1.00)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	44	0.823	836
6	2	50	0.857	853
4	4	38	0.833	665 (365)*
2	6	38	0.827	737 (362)*

Local improvement (permutation probability= 1.00)

Initial population		Stopping criterion	Practical reliability	Normalized price
Normal	Nonzero			
8	0	50	0.807	813
6	2	19	0.827	353
4	4	16	0.817	307 (179)*
2	6	16	0.840	305 (188)*

* Average number of nodes in binary tree.

Table 8. Normalized price near 0.80 practical reliability for the average of the three load cases starting with 4 NZP and 4 normal designs in the initial population.

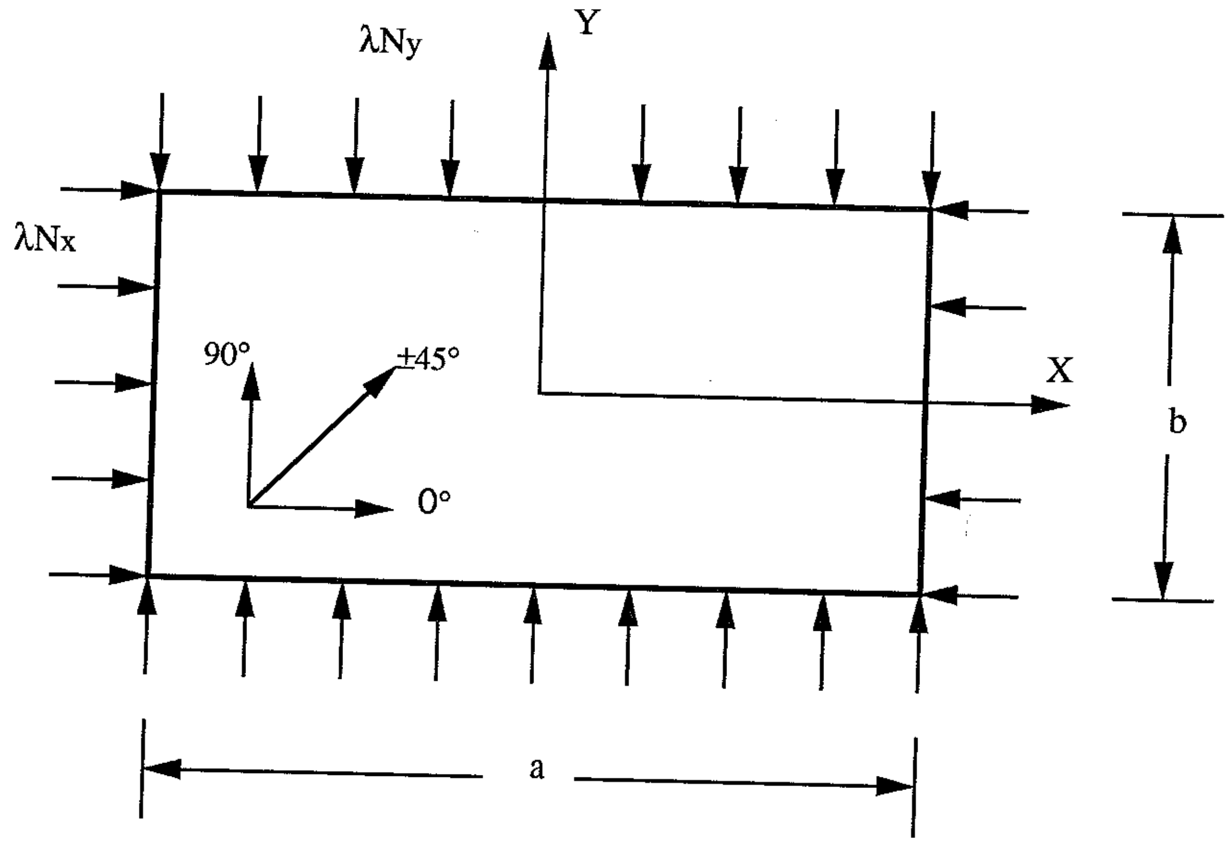
No local improvement

Permutation probability	Stopping criterion	Practical reliability	Normalized price
0.0	63	0.283	3278 (115)*
0.5	32	0.820	602 (250)*
1.0	38	0.833	665 (365)*

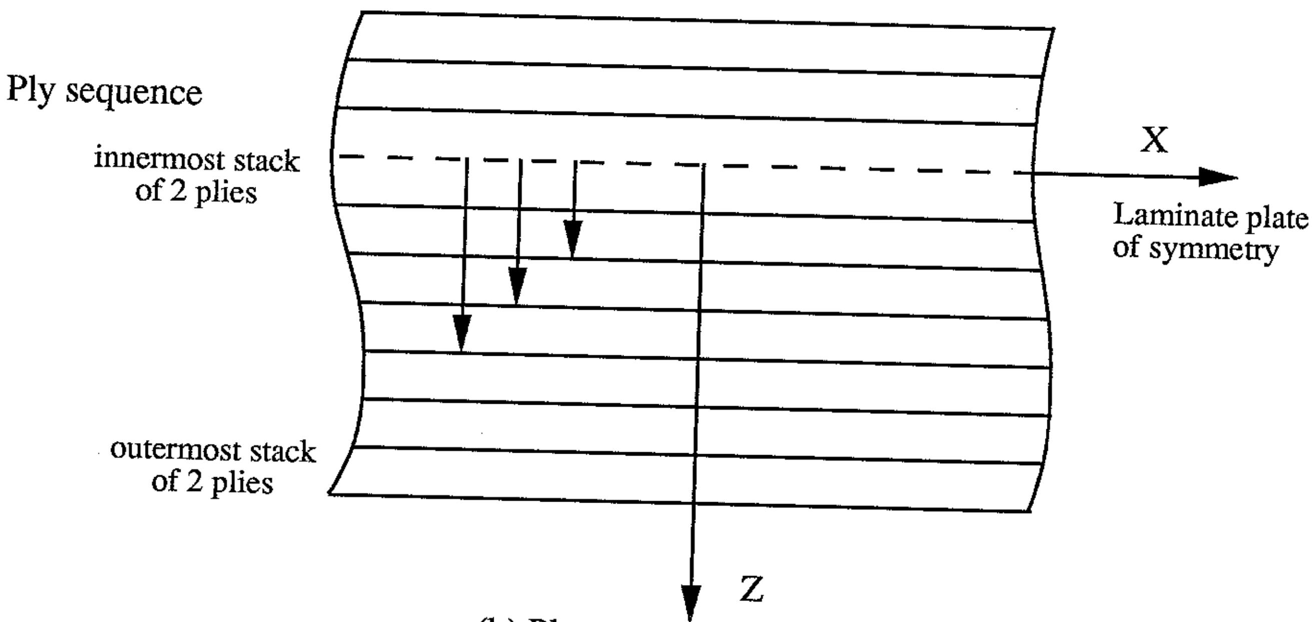
Local improvement

Permutation probability	Stopping criterion	Practical reliability	Normalized price
0.0	25	0.807	439 (106)*
0.5	13	0.820	263 (135)*
1.0	16	0.817	307 (179)*

* Average number of nodes in binary tree.



(a) Laminate plate geometry and applied loading.



(b) Ply sequence location.

Figure 1. Laminated plate geometry and loading.

```
Procedure Genetic algorithm
begin
    initialize population;
    do I=1, population size
        evaluate objective function;
    enddo
    rank designs;
    while number of consecutive generations without improvement in
        the best design less than a specified number do
    begin
        do I=1, population size
            select parents;
            create children by crossover;
            perform mutations;
            perform permutations;
        enddo
        one of new designs replace by the best design of
            the previous generation;
        do I=1, population size
            evaluate objective function;
            local improvement;
        enddo
        rank designs;
    end
end


Procedure Local improvement
begin
    search for 5 nearest neighbors in the binary tree;
    construct a least squares approximation to buckling load;
    while two point interchange not finished do
    begin
        perform two point interchange of stacks;
        compute buckling load approximation;
        adjust objective function for strain failure load
            and contiguous ply constraint;
    end
    replace nominal design by the best interchanged design;
end
```

Figure 2. Pseudocode for genetic algorithm with local improvement.

```
Procedure Evaluation of objective function using binary tree
begin
    search for the given design in the binary tree;
    if found;
        get objective function value from the binary tree;
    else
        search for design having identical in-plane lamination
          parameters;
        if found;
            get strain failure load from the binary tree;
        else
            perform in-plane strain analysis;
        endif
        perform buckling analysis;
        adjust objective function for contiguous ply constraint;
        add design to the binary tree;
    endif
end
```

Figure 3. Calculation of objective function with aid of binary tree.

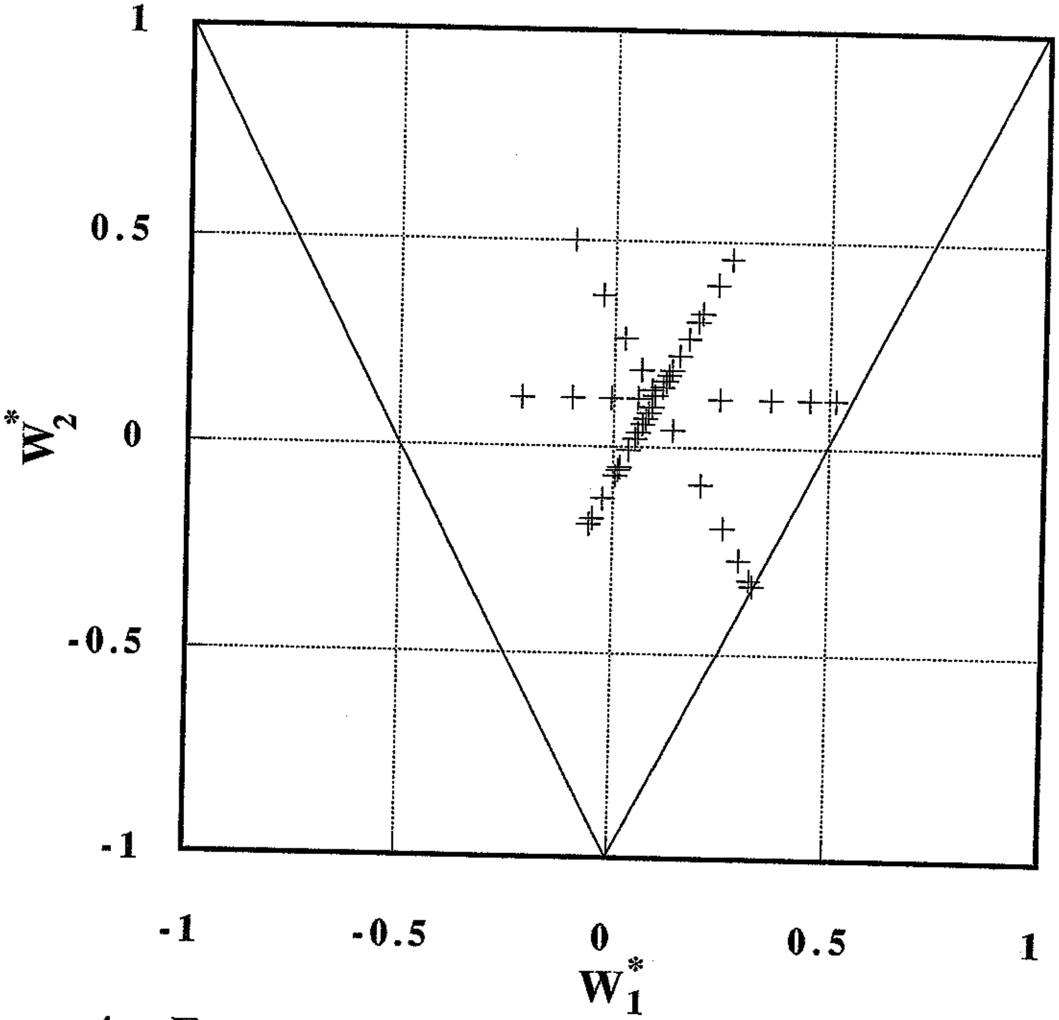


Figure 4. Example of distribution of perturbed designs, nominal design $[\,90^\circ_2/(\pm45^\circ/0^\circ_2)_4/\pm45^\circ_2/90^\circ_2\,]_s$, $W_1^* = 0.09838$, $W_2^* = 0.11806$.

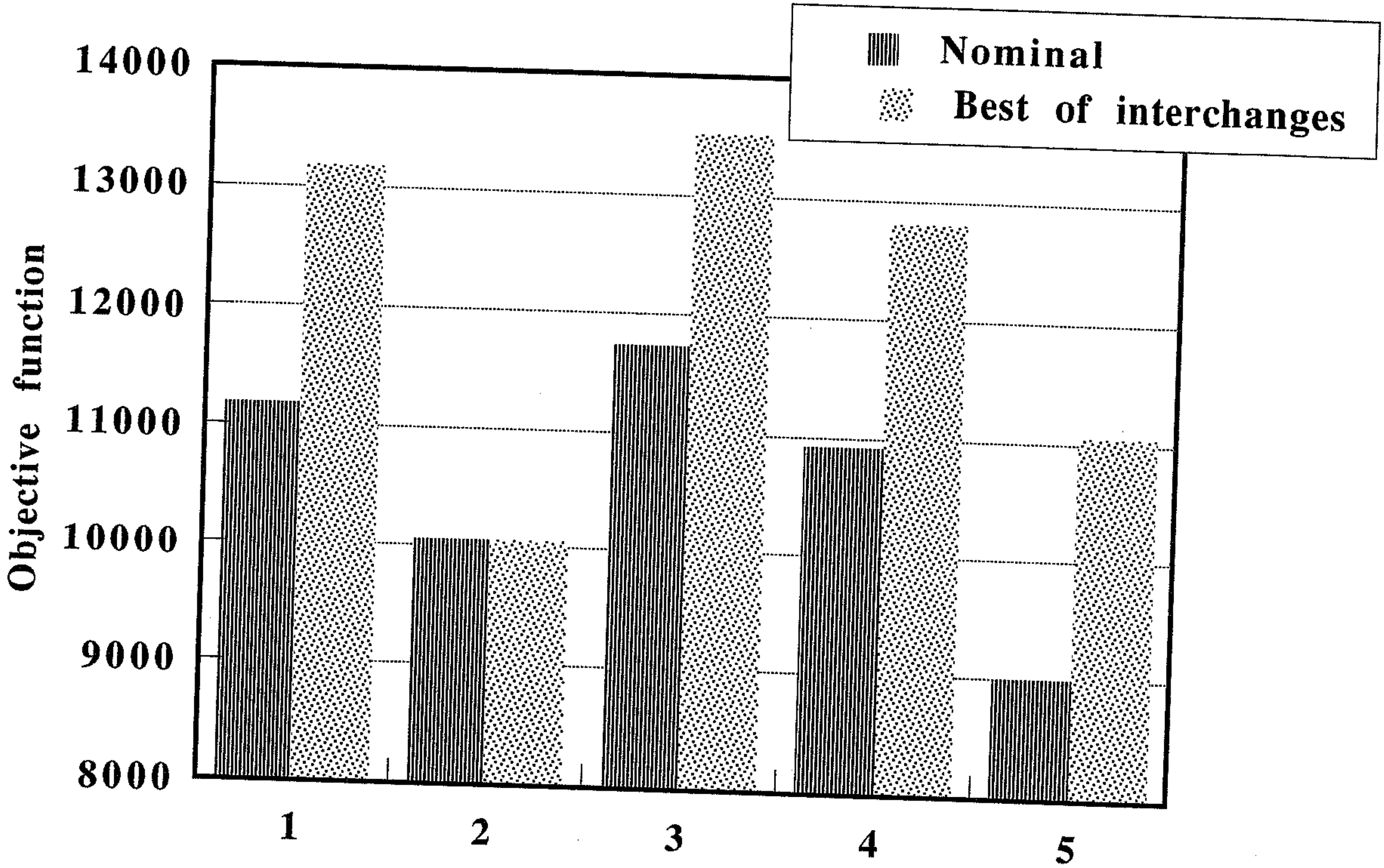


Figure 5. Example of improvements due to local improvement during 10th generartion.

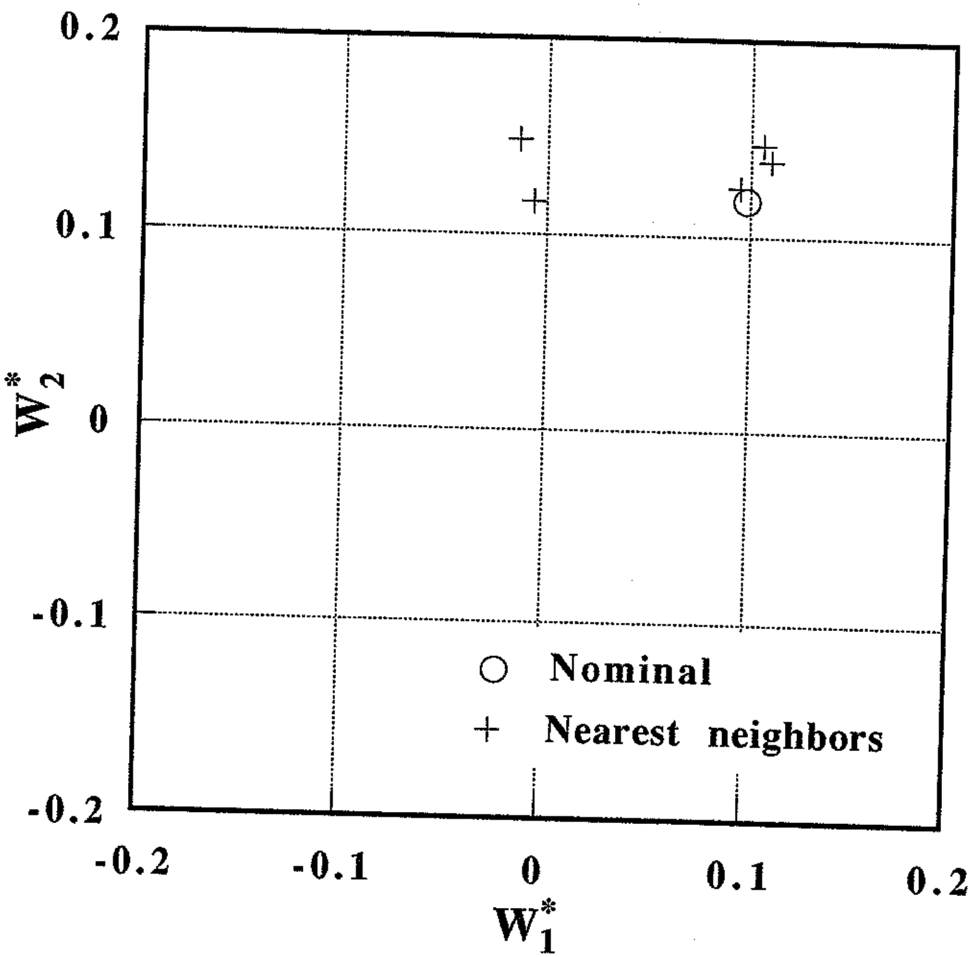


Figure 6. Example of distributions of nearest neighbors.

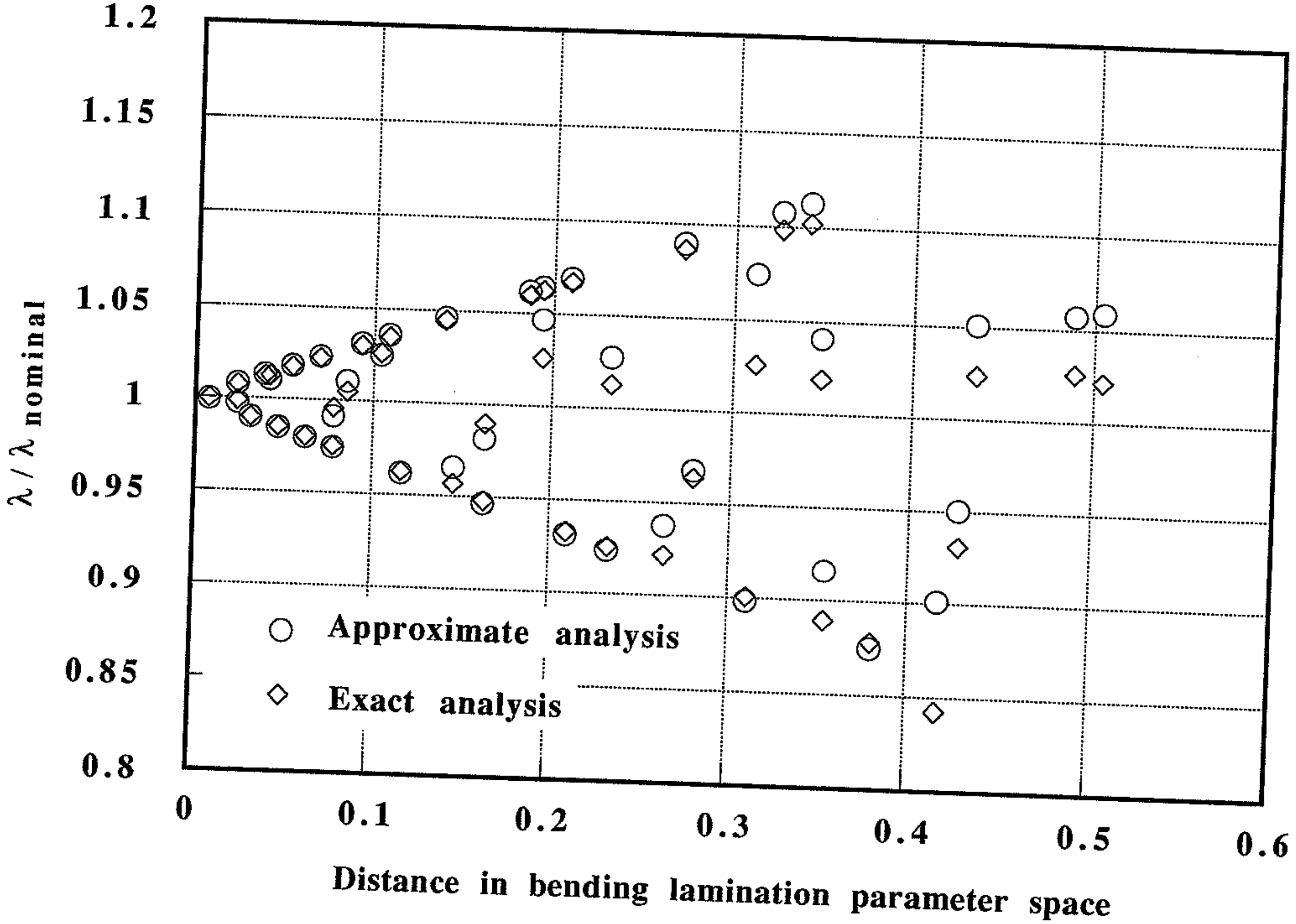


Figure 7. Exact versus approximate normalized buckling loads. Nominal design and perturbed designs shown in Figure 4.

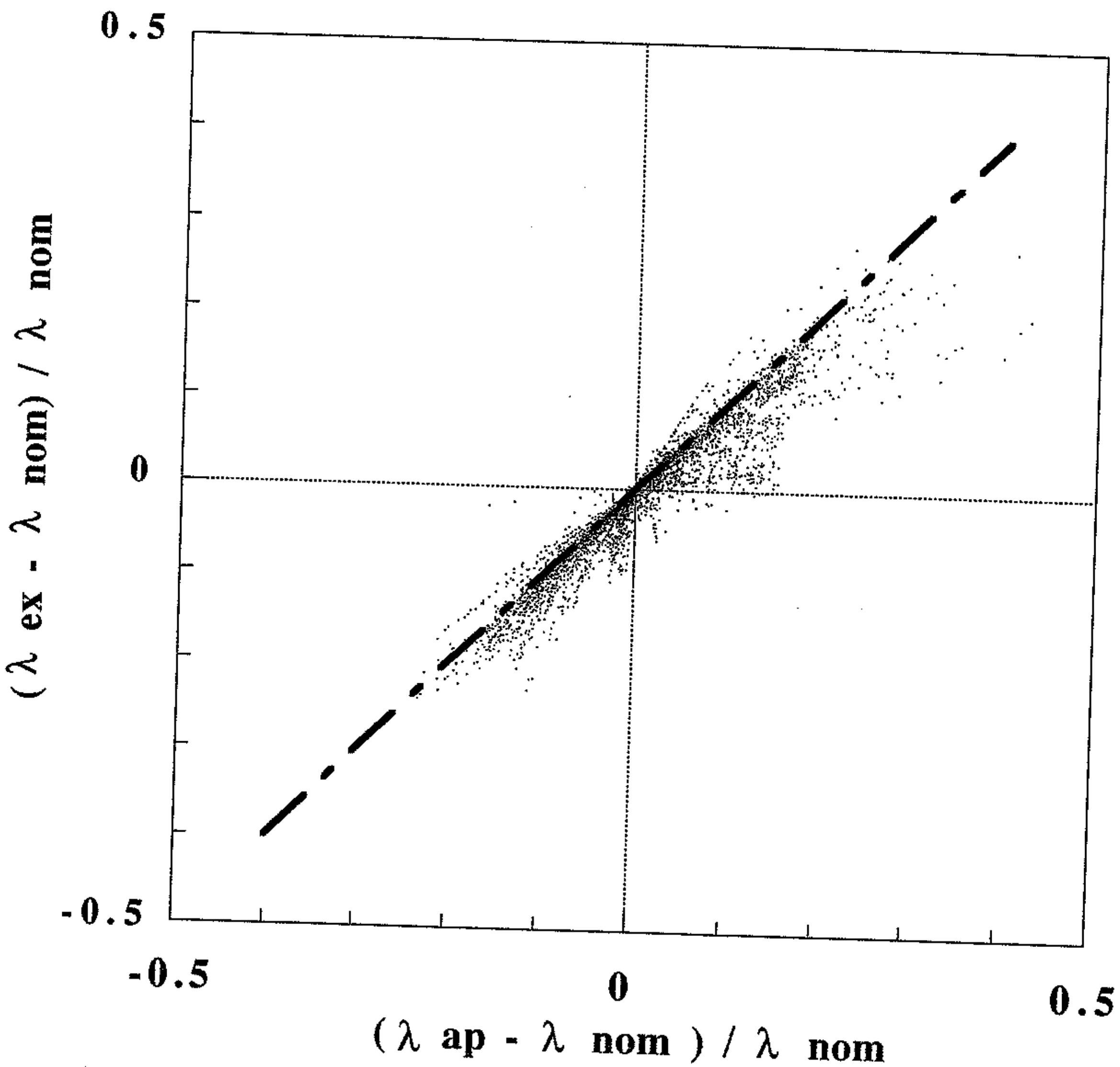


Figure 8. Distribution of accuracy of all designs from a single run.

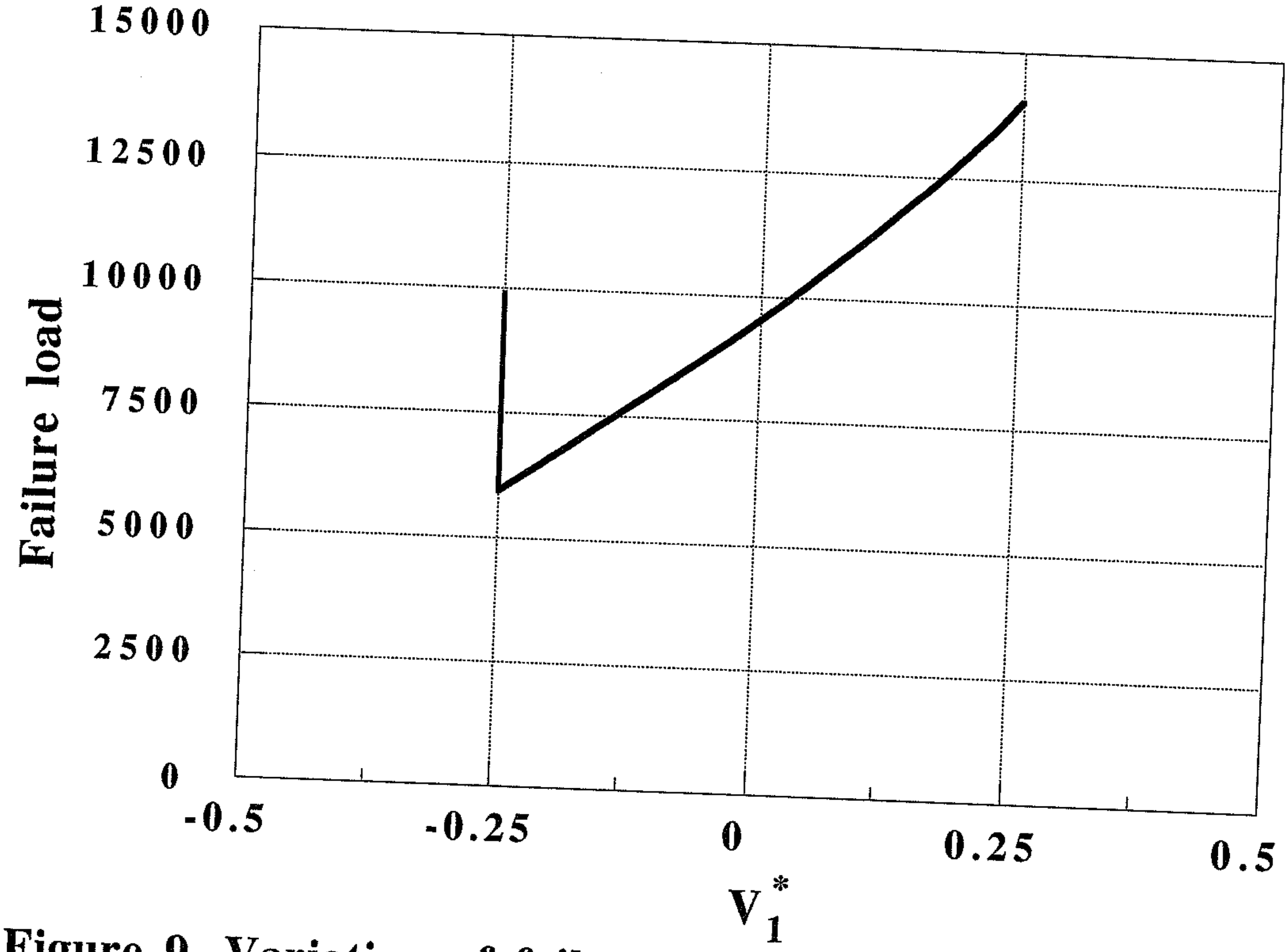


Figure 9. Variation of failure load in the lamination parameter space at $V_2^* = -0.5$ for load case 3.

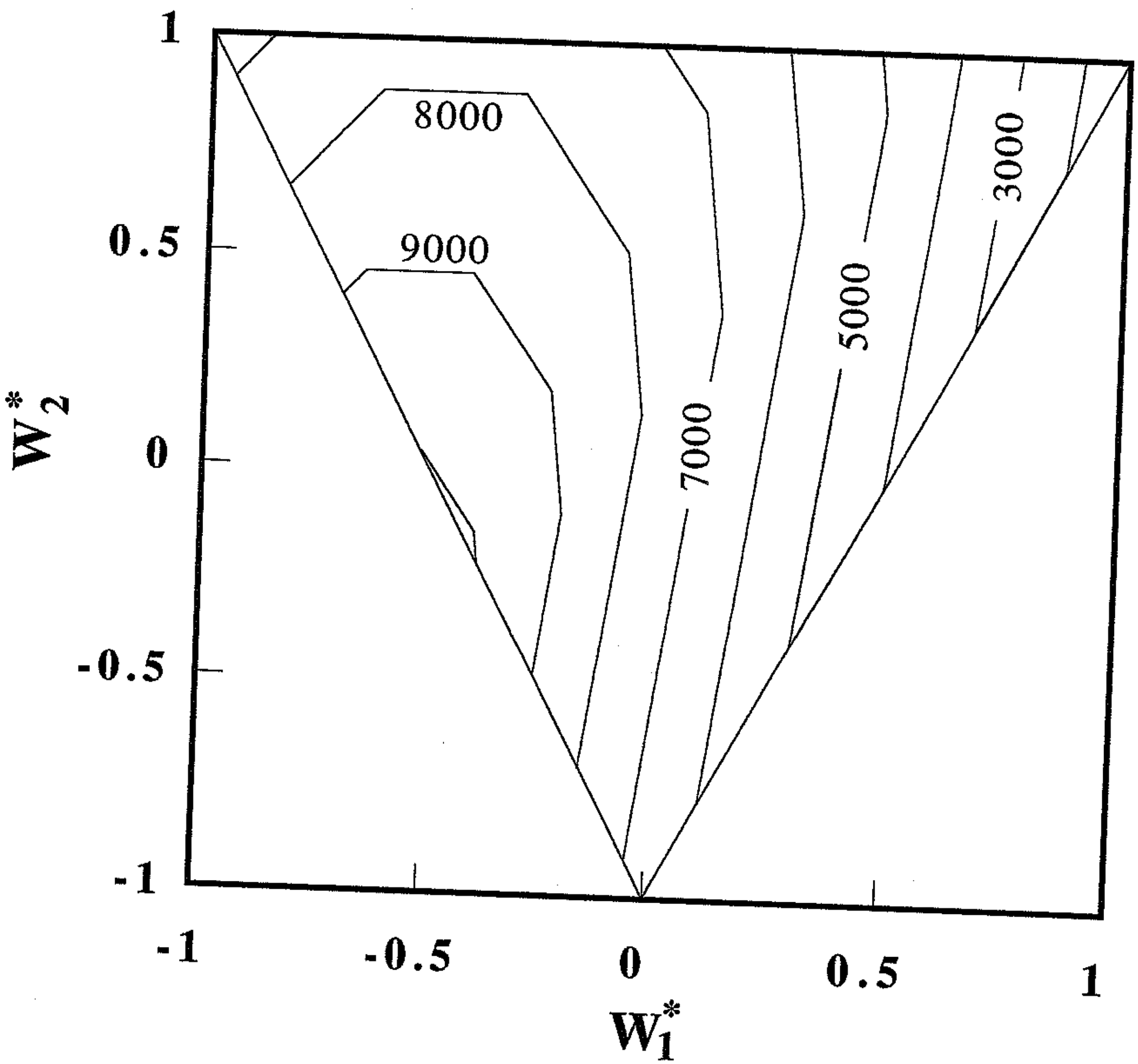


Figure 10. Variation of buckling load for load case 3.

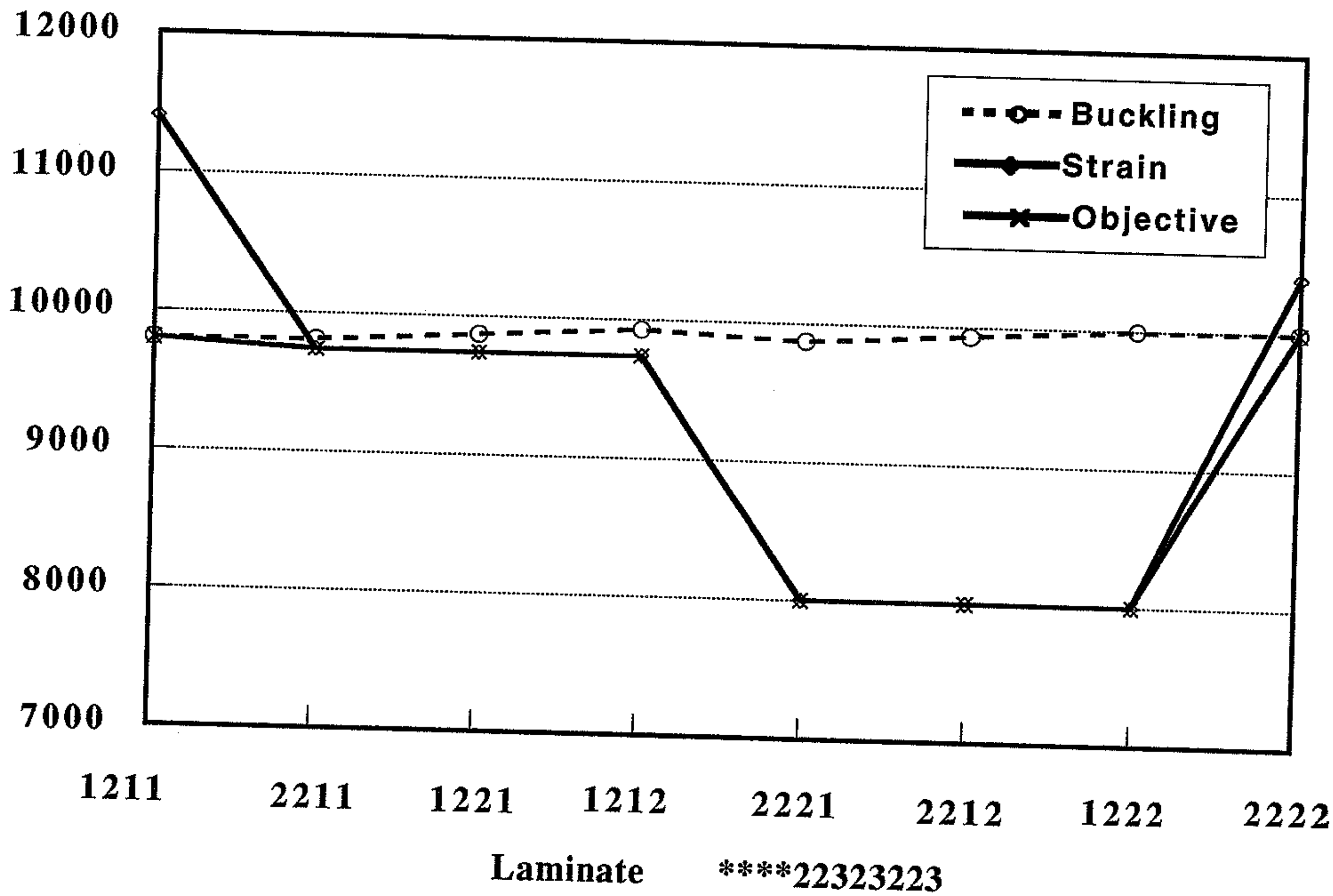


Figure 11. The behavior of the laminate plate near the optimum.