AIAA JOURNAL
Vol. 31, No. 7, July 1993

Multiobjective Optimization of Large-Scale Structures

Ramana V. Grandhi* and Geetha Bharatram†
Wright State University, Dayton, Ohio 45435
and
Vipperla B. Venkayya‡
Flight Dynamics Directorate, Wright-Patterson Air Force Base, Ohio 45433

This paper presents a multiobjective optimization algorithm based on generalized compound scaling techniques. The algorithm handles any number of objective functions, similar to handling behavior constraints. This technique generates a partial Pareto set while solving the optimization problem. A reliability-based decision criterion is used for selecting the best compromise design. The example cases considered in this work include various disciplines in airframe structures, such as stress, displacement, and frequency with hundreds of design variables and constraints. This paper also discusses the concept of Pareto-optimal solutions in the context of a multiobjective structural optimization problem and the commonly used methods of generating Pareto-optimal solutions.

Introduction

MULTIOBJECTIVE optimization has recently been acknowledged as an advanced design technique in structural optimization.[1] Most of the real-world problems are multidisciplinary and complex, as there is always more than one important objective function in each problem. To accommodate many conflicting design goals, one needs to formulate the optimization problem with multiple objectives. One important reason for the success of the multiobjective optimization approach is its natural property of allowing the designer to participate in the design selection process even after the formulation of the mathematical optimization model. The main task in structural optimization is determining the choice of the design variables, objectives, and constraints. Sometimes only one dominating criterion may be a sufficient objective for minimization, especially if the other requirements can be presented by equality and inequality constraints. But generally the choice of the constraint limits may be a difficult task in a practical design problem. These allowable values can be rather *fuzzy,* even for common quantities such as displacements, stresses, and natural frequencies. If the limit cannot be determined, it seems reasonable to treat that quantity as an objective. In addition, usually several competing objectives appear in a real-life application, and thus the designer is faced with a decision-making problem in which the task is to find the best compromise solution between the conflicting requirements.

A variety of techniques and applications of multiobjective optimization have been developed over the past few years. The progress in the field of multicriteria optimization was summarized by Hwang and Masud[2] and later by Stadler.[3,4] Stadler inferred from his survey that if one has decided that an optimal design is to be based on the consideration of several criteria, then the multicriteria theory (Pareto theory) provides the necessary framework. In addition, if the minimization or maximization is the objective for each criterion, then an optimal solution should be a member of the corresponding Pareto set. Only then does any further improvement in one criterion require a clear tradeoff with at least one other criterion. Radford et al.[5] in their study have explored the role of Pareto optimization in computer-aided design. They used the weighting method, noninferior set estimation (NISE) method, and constraint method for generating the Pareto optimal. The authors discussed applying the knowledge-based engineering to formalize the control and derivation of meaning from the Pareto sets.

Rao[6,7] and Rao et al.[8] treated several different problems mainly by either applying the methods in which the objectives are a priori fixed or using the goal-programming and the game theory approach. In their work, they also considered the design of actively controlled structures as a multiobjective optimization problem. Koski[9,10] formulated a multicriterion problem using the material volume and several chosen nodal displacements as the objectives. Most of the applications were on trusses. Usually, the Pareto-optimal set was determined by the linear weighting, the min-max, or the constraint methods. The isostatic truss problem in Ref. 11 was solved analytically in general terms, using the necessary and sufficient conditions of Pareto optimality, in which the weight and displacement are minimized but subject to stress and Euler buckling constraints.

Using goal-programming techniques, El-Sayed et al.[12] demonstrated an algorithm for solving nonlinear structural optimization problems. The algorithm uses the linear goal-programming techniques with successive linearization for the nonlinear equations to obtain the solution for the nonlinear goal optimization problems. A goal-programming formulation was generated for a three-bar truss problem wherein uncertainty in both load magnitude and direction was considered.[13] Hajela and Shih[14] proposed a minimum variant of the global criterion approach to obtain solutions to multiobjective optimum design problems involving a mix of continuous, discrete, and integer design variables. To obtain solutions to multiobjective optimum design problems involving a mix of continuous, discrete, and integer design variables, Saravanos and Chamis[15] developed a design for lightweight, low-cost composite structures of an improved dynamic minimum variant of the global criterion approach. Saravanos and Chamis's performance was based on multiple objectives. Tseng and Lu[16] proposed a minimax multiobjective optimization model for structural optimization. They used the three typical multiobjective optimization techniques—goal programming, compromise programming, and the surrogate worth tradeoff method—to solve truss problems. The main purpose of their work was to apply the multiobjective optimization techniques to the selection of system parameters and to solve large-scale, structural design optimization problems.

Received April 2, 1992; revision received Oct. 16, 1992; accepted for publication Dec. 3, 1992.

*Associate Professor, Department of Mechanical and Materials Engineering.

†Research Associate, Department of Mechanical and Materials Engineering.

‡Principal Scientist, Analysis and Optimization Branch.

This paper presents an algorithm well suited for multicriteria optimization. The algorithm is based on the generalized compound scaling method in which the objectives are treated similar to constraints. Pseudotargets are defined for the objective functions at each iteration, to convert them as constraints. The algorithm generates a partial Pareto-optimal set for a class of problems in which the optima lie on the constraint surface or the intersection of constraints. The main advantage of this algorithm is that in one optimization run some of the Pareto optima are generated. Details of some of the important multiobjective methods, the optimization algorithm, reliability-based decision criterion, and numerical results are presented. Three structural problems presented demonstrate the robustness of the algorithm in solving large-scale optimization problems with linear and nonlinear objective functions. The computational effort involved in the optimization part of these problems was minimal.

Pareto Concept

Pareto optimality serves as the basic multicriteria optimization concept in virtually all of the previous literature. A general multiobjective optimization problem is to find the vector of design variables $X = (x_1, x_2, \ldots, x_N)^T$ that minimize a vector objective function $F(X)$ over the feasible design space X. It is the determination of a set of nondominated solutions (Pareto optimum solutions or noninferior solutions) that achieves a compromise among several different, usually conflicting, objective functions. The Pareto optimal is stated in simple words as follows: "A vector X^* is Pareto optimal if there exists no feasible vector X which would increase some objective function without causing a simultaneous decrease in at least one objective function." This definition can be explained graphically. An arbitrary collection of feasible solutions for a two-objective maximization problem is shown in Fig. 1. The area inside of the shape and its boundaries are feasible. The axes of this graph are the objectives F_1 and F_2. It can be seen from the graph that the noninferior solutions are found in the portion of the boundary between points A and B. Thus, here arises the decision-making problem from which a partial or complete ordering of the set of nondominated objectives is accomplished by considering the preferences of the decision maker. Most of the multiobjective optimization techniques are based on how to elicit the preferences and determine the best compromise solution.

Types of Multiobjective Techniques

Nearly all of the solution schemes used in multiobjective optimization involve some sort of scalarization of the vector optimization problem. The vector problem is replaced by some equivalent scalar minimization problem. Because the Pareto set is generally infinite, an additional use of scalarization is the selection of a unique member of the Pareto set as the optimum for the vector optimization problem. Usually, a problem is scalarized either by defining an additional supercriterion function or by considering the criteria sequentially.

The three basic techniques used for generating noninferior solutions are the weighting method, the noninferior set estimation (NISE) method, and the constraint method. Balachandran and Gero[17] have discussed the relative merits and demerits of these three methods. These methods also come under the category of nonpreference technique. However, it should be noted that some nonpreference techniques use preference techniques (e.g., weighting and NISE methods) concepts repeated over a number of different parametric values to generate the entire Pareto set. Still confronted with these Pareto solutions, the designer must choose among them by some other means. In the following, brief details of various methods are given because some of these methods are used in the results' comparison.

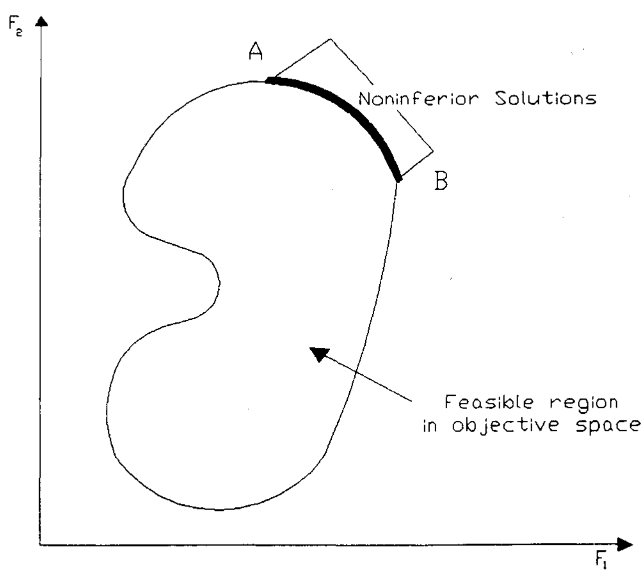

Fig. 1 Graphical interpretation of Pareto optimal.

Weighting Method

This technique is based on the preference techniques of the weights' prior assessment for each objective function. It transforms the multicriteria function to a single criterion function through a parameterization of the relative weighting of the criteria. With the variation of the weights, the entire Pareto set can be generated. Because the results of solving an optimization problem can vary significantly as the weighting coefficients change, and very little is usually known about how to choose these coefficients, a necessary approach is to solve the same problem for many different values of weighting factors. However, because the shape and distribution characteristics of the Pareto set are unknown, it is difficult to determine beforehand the nature of the variations required in the weights so as to produce a new solution at each pass. The second important disadvantage of the method is that it will not identify the Pareto solutions in a nonconvex part of the set.

The NISE is an extension of the weighting method with a mechanism for quickly converging onto the Pareto set. In addition, the accuracy of the approximation can be controlled in the NISE method by using an error criterion. The NISE method operates by finding a number of noninferior extreme points and evaluating the properties of the line segments between them. One of the disadvantages of the NISE method is that, because it is used in conjunction with the weighting method, Pareto solutions in nonconvex parts of the set will not be generated.

Constraint Method

The constraint method is a technique that transforms a multicriteria objective function into a single criterion by retaining one selected objective as the primary criterion to be optimized and treating the remaining criteria as constraints. Stating the remaining objectives as constraints

$$F_i(X) \geq b_i \qquad i = 2, 3, \ldots, q \tag{1}$$

where b_i are parametrically varied target levels of the $q - 1$ objective functions. Each constraint set b_i will produce one Pareto solution. As in the weighting method, many different combinations of values for each b_i must be examined to generate the entire Pareto set. The constraint method provides direct control of generating members of the Pareto set and generally provides an efficient method for defining the shape of the Pareto set.

Multiobjective methods that incorporate preferences are described later. The commonly used methods under this are the goal-programming, game theory, and global criterion methods. In this work, the results of the global criterion method are compared with the solutions of the present approach.

Goal Programming

In goal programming the decision maker is required to specify goals for each objective. Goals are the quantitative values, considered as additional constraints in which new variables are added to represent deviations from the predetermined targets. The objective function specifies the deviations from these goals and priorities for the achievement of each goal, in quantitative terms. The disadvantages of this method are the selection of "correct" or "valid" levels of the under- and overachievement of the goals (deviations from the predetermined targets) that requires knowledge of the individual minima of the objective function, which is not easy to achieve with nonconvex problems.

Game Theory

In the game theory, the multiobjective optimization problem is viewed as a game problem involving several players, each corresponding to one of the objectives. The system is considered to be under the control of these intelligent adversaries, each seeking to optimize their own gain at the expense of their opponents, using all of the available information. In this approach, starting with X_0 as a starting point, q single criterion optimization problems are solved to obtain optimum solutions X^*. The values of all of the objective functions at the vectors $X_1^*, X_2^*, \ldots, X_q^*$ are determined and the elements F_{iu} are defined as

$$F_{iu} = \max \quad F_i(X_j^*); \quad j = 1, 2, \ldots, q \quad i = 1, 2, \ldots, q \tag{2a}$$

Then a supercriterion or bargaining model S is constructed as

$$S = \Pi_{i=1}^{q}[F_{iu} - F_i(X_c^*)] \tag{2b}$$

where X_c^* represents the solution (Pareto-optimal solution) of a minimization problem. The supercriterion S defined in Eq. (2b) is maximized and the optimal convex combination of the objective functions is found; that is, c^* and the corresponding optimum solution of the problem $X^* = X_c^*$ are determined.

Global Criterion Method (Min-Max Formulation)

In this method an optimal vector is found by minimizing some global criterion such as the sum of the squares of the relative deviation of the criteria from the feasible ideal points. Thus X^* is the solution of the problem

$$\tilde{F}(X) = \sum_{i=1}^{q} \left[\frac{F_i(X_i^*) - F_i(X)}{F_i(X_i^*)}\right]^p \tag{3a}$$

subject to

$$g_i(X) \le 0, \qquad i = 1, 2, \ldots, m \tag{3b}$$

where p is generally taken as 2 (Euclidean metric), m is the total number of constraints, q is the total number of objectives, and X^* is the feasible ideal solution of the ith objective, obtained by minimizing individual criterion.

As can be seen from the description of the multiobjective methods, there are various drawbacks in using these approaches for solving large-scale, multidisciplinary optimization problems. For example, many single objective optimization problems have to be solved for generating the complete Pareto set in the constraint or weighting methods. Different weights have to be associated with the objectives to obtain the Pareto set in the weighting or NISE methods. In goal-programming, game theory, and global criterion methods, individual minimas of the single objective function are required, which are difficult to obtain for a nonconvex problem. Hence, it can be seen that multiobjective optimization can become computer intensive while generating the Pareto-optimal set.

Multiobjective Optimization Algorithm

The multiobjective optimization is solved by further advancing the generalized compound scaling algorithm concepts presented by Venkayya.[18] The methodology generates a partial Pareto-optimal set. As stated by Rao[7] and observed by many other researchers, the optimum solution generated by the various multicriteria optimization methods can be different from each other; therefore, a solution concept or procedure may be defined on the basis of other attributes such as the mathematical basis of the method, its generality, and the quality of the final solution. Keeping this in mind, the generalized compound scaling algorithm for multiobjective optimization is developed.

The multiobjective compound scaling (MCS) algorithm is based on compound scaling techniques. The basic idea of the scaling algorithm is to derive a set of scale factors for the design variables such that the design can be brought to the constraint surface in one or more steps from anywhere in the n-dimensional design space by multiplying the current values of the variables and the scale factors. The implication is that the optimum lies on the constraints surface or the intersection of constraints. The scaling techniques are generalized at each iteration to the multiple objective functions by treating them as additional constraints, as both the constraints and the objectives are functions of the same variables.[18] In multiobjective optimization, it is known that the Pareto-optimal set lies on the intersection of objective and constraint function contours (for most structural problems); hence, treating the objective functions in the same way as the constraints seems a reasonable approach. The difficulty in treating the objective functions as constraints is that the constraints have a target to satisfy and the objective functions do not. Hence, to make the objective functions look like constraints, pseudotarget values for the objective functions are generated so that they can be treated as additional constraints that intersect with other constraints. Linear as well as nonlinear objective functions are considered in this work to demonstrate the robustness of the algorithm. This concept is illustrated in Fig. 2 by using the following example:

Minimize

$$F_1 = 4x_1 + x_2 \tag{4a}$$

$$F_2 = \frac{1.0}{x_2 + 0.25x_1} \tag{4b}$$

subject to

$$g_1 = \frac{1.6}{x_2 + 0.25x_1} \le 1.0 \tag{5a}$$

$$g_2 = 0.2\left(\frac{\sqrt{3}}{3x_1} + \frac{2}{x_2 + 0.25x_1}\right) \le 1.0 \tag{5b}$$

The optimization procedure using MCS is started at the lower bounds of the design variables (0.1, 1.0) point A. It can be seen from Fig. 2 that, in one scaling, the design reaches the intersection of the two constraints g_1 and g_2. At point B in Fig.

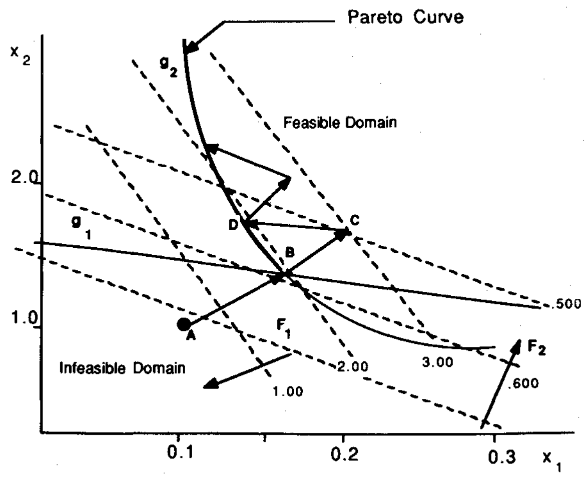

Fig. 2 Demonstration problem.

2, it can be seen that moving in the decreasing direction of objective function F_1 for minimization will violate the constraints. Hence, at point B the designs are scaled by a factor with respect to F_2 depending on the radients of the second objective function; this is point C in Fig. 2. From point C the design moves to the intersection of g_2, F_1, F_2 to point D on the figure. The dark, highlighted curve shows the minimal curve (Pareto optimum). In this procedure, the design moves into the feasible region and then back to the constraint boundary and generates a partial Pareto curve. The details of the algorithm are presented in the following sections.

Multicriteria Problem

The problem of multiobjective optimization subject to multiple constraints can be posed as follows:

Minimize

$$F = [F_1(X), F_2(X), \ldots, F_q(X)] \tag{6a}$$

subject to the constraints

$$g_j(x_1, x_2, \ldots, x_n) = \bar{g}_j \qquad j = 1, 2, \ldots, m_e \tag{6b}$$

$$g_j(x_1, x_2, \ldots, x_n) \le \bar{g}_j \qquad j = m_e + 1, \ldots, m \tag{6c}$$

and the side bounds on the design variables as

$$x_i \ge x_i^{(l)} \qquad i = 1, 2, \ldots, n \tag{6d}$$

where the various F are the conflicting objectives to be minimized, q is the total number of objectives, n is the number of design variables, m is the total number of constraints, and the various $\bar{g}$ are the constraint limits.

MCS Algorithm

This technique makes use of the active constraint set strategy that results in sensitivity calculations for fewer constraints, hence considerably reducing the computational effort. The two most important parameters in this algorithm are the target response ratio β and the characteristic function μ. Figure 3 shows the sequence of operations for the algorithm.

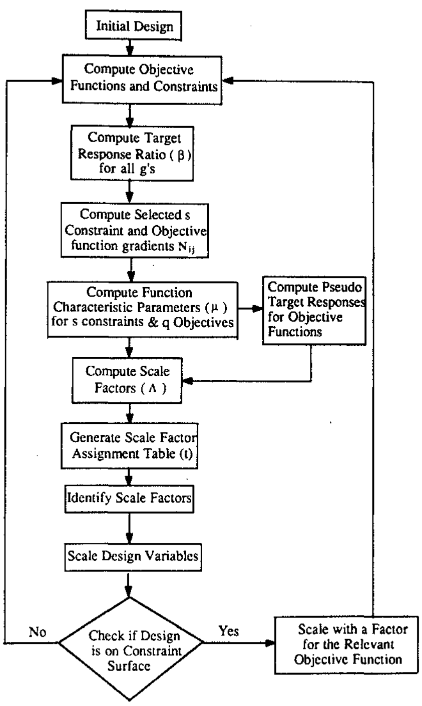

Fig. 3 MCS algorithm flowchart.

First β (target response ratio) parameters are computed for all of the constraints using the following expression:

$$\beta_j = \frac{\bar{g}_j}{g_j} \qquad j = 1, 2, \ldots, m \tag{7}$$

If g_j and overline $\bar{g}_j$ are positive, then the following statements are valid: $\beta_j = 1$ represents the constraint surface, $\beta_j > 1$ represents the feasible region, and $\beta_j < 1$ represents the infeasible region. After arranging the constraints with the various β in the ascending order, the constraints with the lowest β values being the most critical are selected for the active constraint set (about 10–15) and denoted by s.

Next, the characteristic function parameter μ is computed for each of the selected s constraints and the objective functions. The μ parameter consists of two parts based on the sign of the individual terms in the summation.

For all negative μ_{ij} terms

$$\mu_{jN} = -\sum_{i=1}^{n} \mu_{ij} \qquad j = 1, 2, \ldots, s + q \tag{8a}$$

For all positive μ_{ij} terms

$$\mu_{jP} = -\sum_{i=1}^{n} \mu_{ij} \qquad j = 1, 2, \ldots, s + q \tag{8b}$$

where

$$\mu_{ij} = \frac{N_{ij} x_i}{z_j} \qquad j = 1, 2, \ldots, s + q \tag{8c}$$

where z_j is a vector containing the values of s selected constraints and the q objective functions, N_{ij} is the ordered gradient matrix corresponding to the selected active constraint set s in the order of their β values, which are sorted in the ascending order and derivatives of the objective functions with respect to the design variables.

To make the objective functions look like constraints, they are written as

$$F_i(X) \le \bar{F}_i \qquad i = 1, 2, \ldots, q \tag{9}$$

where $\bar{F}_i$ are the pseudotargets for the objective functions. These targets change in each iteration depending on the most active or violated constraint. For example, if the first constraint in the active set is violated, then the pseudotargets for the objective functions are posed higher than their present value. If the constraint is inactive, then the targets are posed lower than their present value to reduce the objective functions. In the following, two distinct cases are identified for computing the β values for these two aspects.

Case 1

When the design is away from the constraint surface, the target response ratio for the objective functions is computed as

$$\beta_k = \delta \qquad \mu_{kN} \ge \mu_{kP} \tag{10a}$$

$$\beta_k = \frac{1}{\delta} \qquad \mu_{kN} < \mu_{kP} \tag{10b}$$

where

$$\delta = \frac{1 + \alpha\beta_1}{1 + \alpha} \tag{10c}$$

and where $k = s + 1, s + 2, \ldots, q$ and β_1 is the target response ratio for the most violated or most active constraint from the active constraint set s. Because the objective functions are conflicting in nature, the pseudotargets for the two objective functions are never equal. The α value is calculated using the same constraint characteristic function μ value:

$$\alpha = \gamma \qquad \mu_{1N} \ge \mu_{1P} \tag{11a}$$

$$\alpha = \frac{1}{\gamma} \qquad \mu_{1N} < \mu_{1P} \tag{11b}$$

where

$$\gamma = \frac{1}{\mu_1} \tag{11c}$$

and where μ_1 is the larger of μ_{1P} or μ_{1N} for the selected constraint. The objective function related to the smallest β_k is selected to participate in the optimization process.

Case 2

In the second case when the design is on the constraint boundary, the β parameters for the objective functions are set to a value less than 1.0. The objective function that had a lower β value in the previous iteration is considered for scaling. The selected objective function is reduced by multiplying or dividing the current values of the design variables by a factor depending on whether the gradient of the objective function is positive or negative with respect to that design variable. The factor signifies a reduction of a selected percentage in the objective function at that iteration step.

The next step is to calculate the scale factors. Scale factors Λ depend on the values of the function characteristic parameters and target response ratios. The μ parameter signifies the measure of nonlinearity in the constraints. If the value of μ_{jN} or μ_{jP} is 1, scaling is exact regardless of the range of β_j. Otherwise, it might take two or three scalings to reach a function boundary. The procedure for calculating the scale factors is as follows:

$$\Lambda_{jN} = \left(\frac{1}{\beta_j}\right)^{1/\mu_{jN}} \qquad \mu_{jN} \geq \mu_{jP} \tag{12a}$$

or

$$\Lambda_{jP} = (\beta_j)^{1/\mu_{jP}} \qquad \mu_{jN} < \mu_{jP} \tag{12b}$$

The following conditions are used in constructing the scale factors matrix:

$$\left.\begin{array}{ll} \Lambda_{ij} = \Lambda_{jN} & \mu_{ij} < 0 \\ \Lambda_{ij} = 1.0 & \mu_{ij} \geq 0 \end{array}\right\} \qquad \mu_{jN} \geq \mu_{jP} \tag{13a}$$

or

$$\left.\begin{array}{ll} \Lambda_{ij} = \Lambda_{jP} & \mu_{ij} > 0 \\ \Lambda_{ij} = 1.0 & \mu_{ij} \leq 0 \end{array}\right\} \qquad \mu_{jN} \geq \mu_{jP} \tag{13b}$$

These conditions result in $s + q$ possible scale factors for each design variable. The strategy in compound scaling is to select a mix of the scale factors from various columns of the scale factor matrix with the object of approaching the intersecting points of the constraints and objective functions. The relevant scale factors are determined with the help of a scale factor assignment procedure. The underlying concept is similar to pivoting for identifying prominent variables. The scale factors assignment matrix is computed as follows:

$$\left.\begin{array}{ll} t_{ij} = \left|\dfrac{N_{ij}x_i}{z_j}\right| \div \mu_{jN} & \mu_{ij} < 0 \\ t_{ij} = 0 & \mu_{ij} \geq 0 \end{array}\right\} \qquad \mu_{jN} \geq \mu_{jP} \tag{14a}$$

or

$$\left.\begin{array}{ll} t_{ij} = \left|\dfrac{N_{ij}x_i}{z_j}\right| \div \mu_{jP} & \mu_{ij} > 0 \\ t_{ij} = 0 & \mu_{ij} \leq 0 \end{array}\right\} \qquad \mu_{jN} < \mu_{jP} \tag{14b}$$

The values of the entries in the scale factor assignment matrix vary from zero to one.

The information in the scale factor and assignment tables provides necessary data for generating new designs that can be evaluated for their merit. The strategy to generate new designs is outlined next. The following new designs are generated by using the compound scaling technique, starting from the initial design:

1) Scale to the most active constraint surface.
2) Scale to the intersection of 1 and 2 constraints.
3) Scale to the intersection of 1, 2, and 3 constraints, and so on.

The last design, however, includes all of the active set constraints plus the objective function (the objective function with a pseudotarget function closest to unity is selected). These designs are generated simply by using scale factors without calling repeated finite element analyses. Now, criteria are developed for selecting the best of these k designs. The percentage reduction in the selected objective function and the constraint violation are the two considerations in choosing the best design. In reality, these two requirements will not necessarily be satisfied by the same design. A decision table was proposed by Venkayya[18] for helping to choose the appropriate design. The decision table is basically a matrix, with each row in the matrix representing a new design obtained by applying the compound scaling to reach the constraint intersections. For example, the row 3 would represent a new design that is obtained by applying compound scaling to the constraints 1, 2, and 3. The new design is selected based on weighting the percentage reduction in the objective function and the maximum violation of constraint. The generation of the decision table for the proposed k designs is done by using the first-order approximation of Taylor's series.

Pareto optima are generated only if the optima lie on the active constraint boundary. At a given iteration step, only one objective function is selected for the previously described scaling operations. The MCS algorithm simply drives the design to the intersection of the constraints and objective functions, if such an intersection exists.

Reliability Criteria

In multiobjective optimization, a number of Pareto-optimal designs can be considered as acceptable and valid designs. Decisions on the choice of the "best" or most "suitable" design are based on preferential ranking in which the decision maker already has set priorities for the objective functions. In this work, the design selection procedure is developed based on the reliability of the design considering the random nature of the cross-sectional dimensions. In the present formulation design, variables are considered as the random variables.

In the optimization problem, the constraint function $G_i(X)$ represents each mode of malfunction or failure of the system, where $G_i(X) = \bar{g}_i - g_i(X)$. For given values of random variables (X), this function separates the space of random variables into safe and failure zones. That is,

$$G_i(X) > 0 \Rightarrow \qquad \text{safe space} \tag{15a}$$

$$G_i(X) < 0 \Rightarrow \qquad \text{failure space} \tag{15b}$$

The boundary between these two regions, $G_i(X) = 0$, is called the failure surface or limit state function. If the joint probability density function of the random variables $f_X(x)$ is known, the reliability measure associated with the ith constraint (or probability of survival) can be obtained as

$$P_{si} = \int_{G_i > 0} f_X(x)\, \mathrm{d}x \tag{16}$$

In practice the joint distribution of the random variables is rarely known, and also the integral evaluation is extremely difficult. Approximations for the preceding integral have been proposed by many researchers.[19] Equation (16) can be approximated as

$$P_{si} = \phi(\eta_i) \tag{17}$$

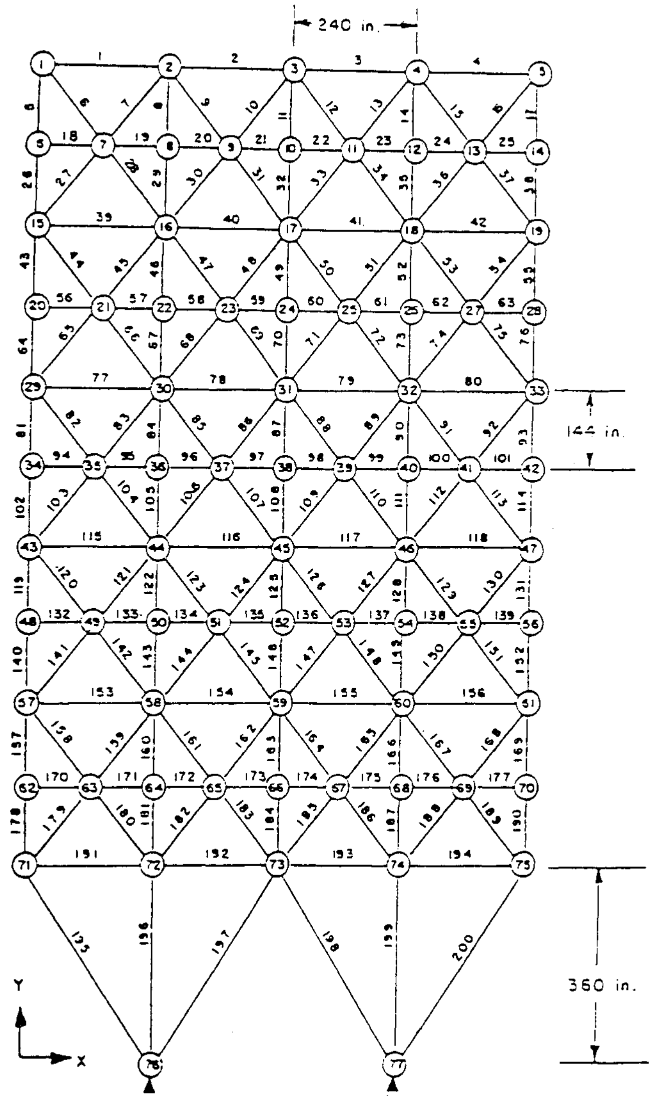

Fig. 4 Two hundred-bar truss structure.

where $\phi(\cdot)$ is the standard normal probability distribution function, and η_i is the reliability index for the ith failure mode, defined as

$$\eta_i = \frac{\bar{G}_i(X)}{\sigma(G_i)} \tag{18}$$

where $\bar{G}_i(X)$ is the mean value of the constraint, and $\sigma(G_i)$ is the standard deviation. The constraint function $G_i(X)$ is in general a nonlinear function of the random variables. Thus, the mean and variance, required in Eq. (18), cannot be computed in exact form by the first and second moments of the random parameters alone. To overcome this difficulty, the constraint function is linearized by means of a Taylor series expansion about an appropriate point $\bar{X}$.

$$\bar{G}_i(X) = G_i(\bar{X}) \tag{19}$$

$$\sigma(G_i) = \left[\sum_{j=1}^{n}\left(\frac{\partial G_i}{\partial x_j}\right)^2_{\bar{X}} \sigma^2_{x_j}\right]^{1/2} \tag{20}$$

The method for evaluating η_i by expanding about the mean value of random variables is defined as a first-order, second moment method.

The probability of failure is approximated as

$$P_{fi} = \phi(-\eta_i) \tag{21}$$

Ditlevsen's first-order upper bound[20] was selected for calculating system reliability. The system failure probability is approximated for an inexpensive computation. Hence, combining the individual failure probabilities of each failure subsystem, the system reliability is evaluated as

$$P_f = \sum_{i=1}^{N} P_{fi} \tag{22}$$

where P_f is the system failure probability, and P_{fi} is the probability of the individual failure mode.

Numerical Results and Discussions

The MCS algorithm was applied on the optimal design of truss and plate structures. Linear (structural weight) as well as nonlinear (displacement, frequency) objective functions were considered. Solutions obtained using the MCS algorithm were compared with the global criterion method for $p = 2$ [Eq. (3a)]. Pareto curves obtained using the MCS technique are also presented in this paper.

Example 1: 200-Bar Truss

The 200-bar truss structure shown in Fig. 4 has 77 nodes, 200 bar elements, and 150 degrees of freedom. At each of these 75 free nodes, 193-lb nonstructural mass was considered. The structure is made up of steel with Young's modulus, 30.0×10^6 psi and a weight density of 0.283 lb/in.^{3} Because of the symmetry of the structure, 105 design variables were considered in the optimization. The structure was subjected to five different loading conditions.

The objectives were to minimize the structural weight and maximize the fundamental natural frequency when subjected to normal stress (σ_x and σ_y) constraints and lower bound limits on the second and third frequencies of the structure. The upper limits on the normal stresses were 30,000.0 psi. The limits on the frequencies were $\omega_2 \geq 10.0$ Hz and $\omega_3 \geq 14.0$ Hz. A total of 1002 constraints were considered in the optimization. At the initial design, all of the cross-sectional areas were 10.0 in.^{2}. The lower limit on the design variables was 0.1 in.^{2}. The initial weight of the structure was 99,634 lb, and the first three frequencies at the initial design were 2.87, 13.02, and 13.72 Hz.

The results obtained by minimizing the individual objective functions are shown in Table 1. It can be observed that the minimization of F_1 (weight) gave a value of $F_1^* = 16{,}686.97$ lb with the corresponding value of F_2 (fundamental frequency) = 3.737 Hz, whereas the maximization of F_2 yielded $F_2^* = 4.211$ Hz with the corresponding $F_1 = 21{,}490$ lb. These values indicate the penalty associated with the other objective function while optimizing a particular objective function. The individual minimizations yielded ω_2 and ω_3 close to each other, which seems to be an inherent property of the structure. The MCS algorithm generated three Pareto-optimal solutions, and these lie at the intersection of constraints. Table 1 also shows

Table 1 Pareto-optimal set for 200-bar truss

No.	System reliability	$F(x) = [F_1(x), F_2(x)]$, weight, frequency
1	0.3765	20,250.9, 4.143
2	0.4023	18,831.0, 4.098
3	0.4567	17,973.0, 3.987
	Global criterion method ($p = 2$)	21,470.228, 4.150

	Initial	Minimize weight subject to constraints	Maximize ω_1 subject to constraints
Weight, lb	99,663.39	16,686.97	21,490.00
ω_1, Hz	2.87	3.737	4.210
ω_2, Hz	13.02	14.390	14.123
ω_3, Hz	13.72	14.904	15.117

the designs and their calculated system reliability. The best possible design selected was $F_1 = 17{,}973.0$ lb and $F_2 = 3.987$ Hz based on their respective system reliability. Figure 5 shows the minimal curve by joining the multiobjective solutions obtained by the MCS algorithm. Solving the same problem using the global criterion method for $p = 2$ (Euclidean metric) yielded $F_1 = 21{,}470.228$ lb and $F_2 = 4.150$ Hz as the optimum. This point lies on the minimal curve shown in Fig. 5.

Example 2: Simply Supported Square Plate

A square plate simply supported at the edges, with dimensions 10×10 in., was optimized for minimum weight and node 1 displacement (Fig. 6). The material properties are Young's modulus $E = 10^7$ psi, Poisson's ratio $\nu = 0.3$, and mass density $\rho = 0.1$ lb/in.^{3}. A uniformly distributed pressure of 100 psi was applied. The plate was modeled using 64 finite elements (a mesh of 8×8). Double symmetry of the plate was

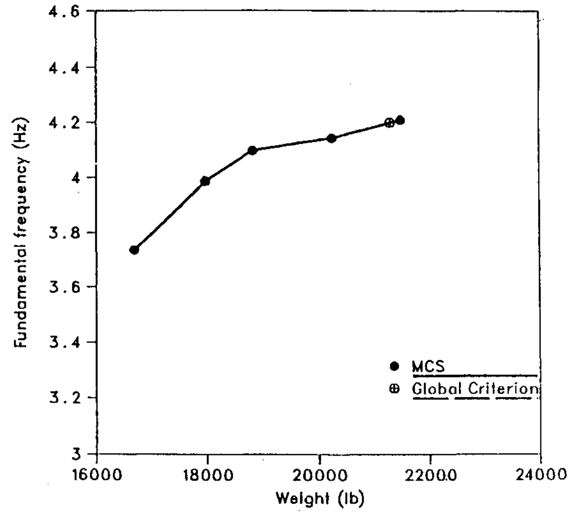

Fig. 5 Minimum curve in criterion space (example 1).

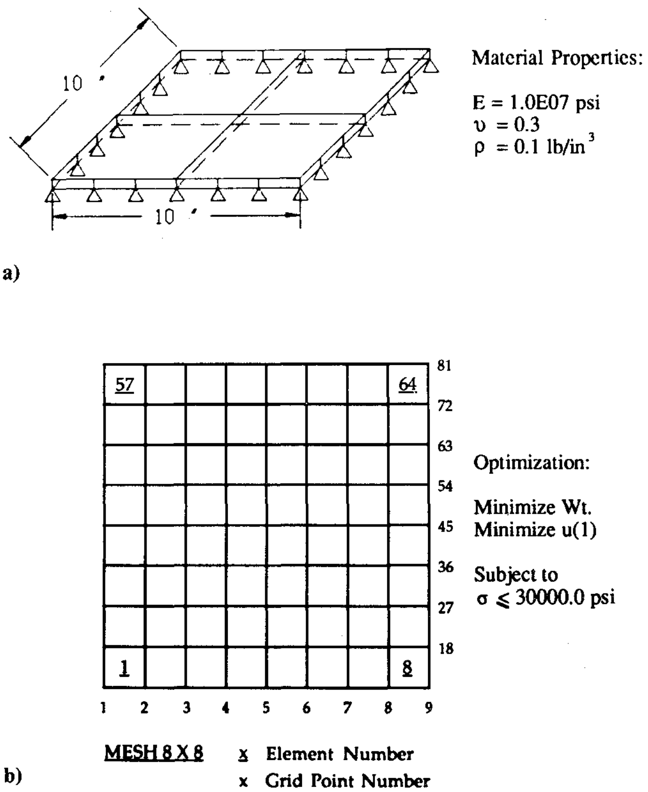

Fig. 6 Simply supported square plate.

Table 2 Pareto-optimal set for simply supported square plate

No.	System reliability	$F(x) = [F_1(x), F_2(x)]$, weight, displacement
1	0.4670	0.9319, 0.0708
2	0.4080	0.6852, 0.1674
3	0.4950	0.6401, 0.1779
4	0.4610	0.6167, 0.1836
	Global criterion method ($p = 2$)	0.6835, 0.1714

	Initial	Minimize weight subject to stress constraints	Maximize displacement subject to stress constraints
Weight, lb	1.2500	0.5915	1.2500
u, in.	0.0372	0.2423	0.0372

Table 3 Pareto-optimal set for plate with internal supports

No.	System reliability	$F(x) = [F_1(x), F_2(x)]$, weight, frequency
1	0.4379	42.88, 36,461.20
2	0.4211	43.15, 37,859.02
3	0.4299	43.82, 38,906.14
4	0.4212	44.36, 39,777.20
	Global criterion method ($p = 2$)	44.56, 39,927.65

	Initial	Minimize weight subject to constraints	Maximize ω_1 subject to constraints
Weight, lb	38.4	42.32	63.82
ω_1, rad/s	29,816.38	33,404.12	43,610.63

observed about the central axes, and hence only a quarter model was considered. Each element was controlled by an independent design variable. The objective was to minimize the structural weight and the maximum nodal displacement (node 1) subjected to normal stress (σ_x and σ_y) constraints. The limits on the normal stresses were 30,000.0 psi. A uniform plate with a thickness equal to 0.5 in. was taken as the initial design, which resulted in an initial weight of 1.25 lb, with a corresponding maximum displacement (node 1) of 0.0372 in., and the design was in the feasible region. The lower bound on the design variables was 0.1 in.

The results obtained by minimizing the individual objective functions are shown in Table 2. It can be observed that the minimization of F_1 (weight) gave a value of $F_1^* = 0.5915$ lb with the corresponding value of F_2 (maximum displacement) = 0.2423 in., whereas the minimization of F_2 yielded $F_2^* = 0.0372$ in. with the corresponding $F_1 = 1.25$ lb. These values provide bounds on the expected Pareto solutions. The MCS algorithm generated four optimal solutions. Table 2 also shows the four designs and their respective system reliability. The best possible design selected was at design 3 (Table 2) where $F_1 = 0.6401$ lb and $F_2 = 0.1779$ in. Figure 7 shows the minimal curve in criterion space generated by MCS. This curve was generated by simply joining the four optimal points and the individual minimization points with line segments. Elements near the supports were thicker than the rest of the structure.

Solving the same problem using the global criterion method for $p = 2$ (Euclidean metric) yielded $F_1 = 0.6835$ lb and $F_2 = 0.1714$ in. as the optimum. This point lies on the minimal curve shown in Fig. 7. The present approach did not select this as the best design, based on the system reliability criterion.

Example 3: Square Plate with Internal and Edge Supports

The third example problem was a square plate with dimensions 12×12 in. (Fig. 8). The plate is supported along its edges and also at internal locations to represent a platform

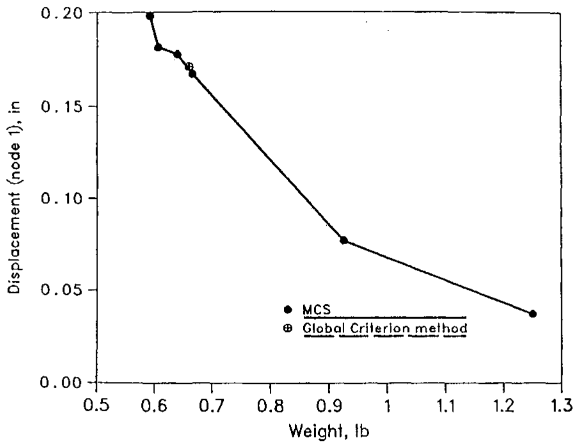

Fig. 7 Minimal curve in criterion space (example 2).

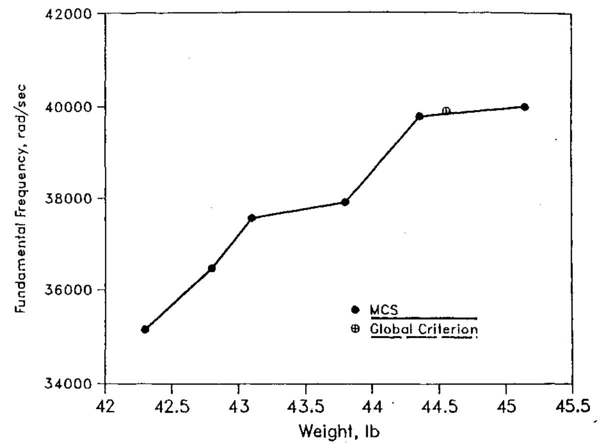

Fig. 9 Minimal curve in criterion space (example 3).

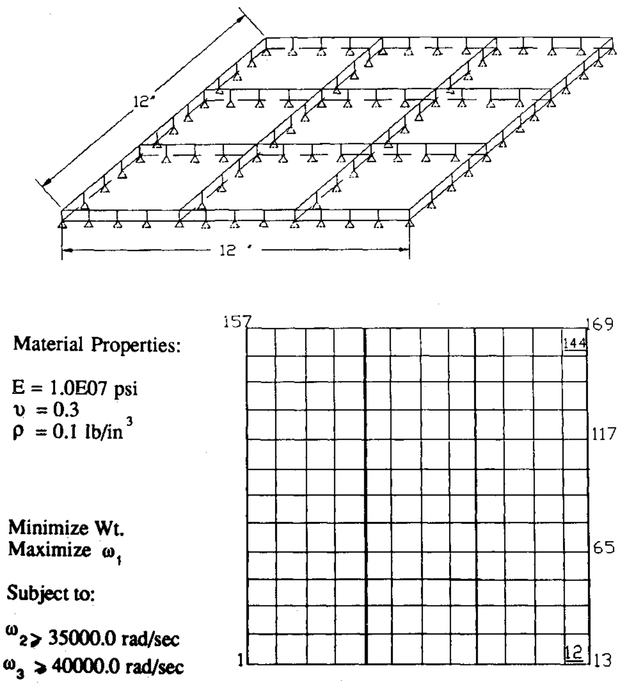

Fig. 8 Square plate with internal and edge supports.

structure with several internal supports (a total of 88 simply supported nodes out of 169 nodes). The plate is made up of nine blocks. The material properties are those of aluminum with Young's modulus $E = 10^7$ psi, Poisson's ratio $\nu = 0.3$, and mass density $\rho = 0.1$ lb/in.^{3}. The plate was modeled with 144 elements as a complete plate because of the inherent properties of the plate. A total of 24 lb of nonstructural mass was added to the structure at all of the free nodes. At the initial design, all of the variables were taken as 1.0 in., which resulted in an initial weight of 38.4 lb, and the resulting fundamental frequency was $\omega_1 = 29{,}816.38$ rad/s. The plate was optimized for both minimum weight and maximum fundamental frequency subject to constraints on the second and third natural frequencies. The two constraints were posed as $\omega_2 \geq 35{,}000.0$ rad/s and $\omega_3 \geq 40{,}000.0$ rad/s. A minimum gauge value of 0.1 in. was imposed on all of the design variables. The initial design was in the infeasible region.

Characteristics of the starting design, along with those of the designs obtained by minimizing the individual objective functions, are shown in Table 3. The minimum of F_1 is $F_1^* = 42.32$ lb with the corresponding value of $F_2 = 33{,}404.12$ rad/s. The maximization of F_2 led to $F_2^* = 43{,}610.63$ rad/s with the associated value of $F_1 = 63.82$ lb. Applying the MCS algorithm resulted in four Pareto designs with their respective system reliability values indicated in Table 3. The best design from Table 3 reduced the value of F_1 to 42.88 lb (from a maximum possible value of 63.82 lb) and increased the value of F_2 to 36,461.20 rad/s (from the worst possible value of 33,404.12 rad/s). The minimal curve in the criterion space generated by MCS is given in Fig. 9. Constraint 2 was active at the optimum. In general, elements at the center of each block were thicker than the others. The solution for this example using the global criterion method ($p = 2$) yielded $F_1 = 44.56$ lb and $F_2 = 39{,}927.65$ rad/s. This point lies on the minimal curve shown in Fig. 9.

Conclusions

In this paper, a new technique for the multiobjective optimization was developed. This technique was demonstrated on large-scale problems with more than 100 design variables. In large-scale problems, where both the finite element and sensitivity analyses are applied several times in calculating just one Pareto optimum, the computational cost becomes very high in generating the complete Pareto optimal set. But the MCS algorithm generates a partial Pareto set while finding the intersections of constraints. The best compromise design among the multiple solutions was selected based on the system reliability of the structure. On the whole, when comparing MCS to the methods described earlier (e.g., constraint, weighting, global criterion, etc.), the algorithm required less computer resources to generate an optimal set. Validation of this technique on truss and plate structures with displacement, stress, and frequency functions (constraints or objectives) where the problems are highly nonlinear demonstrated the algorithm's capabilities.

Acknowledgments

This research work was supported by the U.S. Air Force through Contract F33615-88-C-3204. The authors acknowledge R. A. Brockman, University of Dayton Research Institute, for providing the PROTEC computer program. The authors also wish to thank Mahidhar R. Vallapareddy for his assistance in this work.

References

[1]Eschenauer, H., Koski, J., and Osyczka, A., *Multicriteria Design Optimization—Procedures and Applications,* Springer-Verlag, Berlin, 1990.

[2]Hwang, C. L., and Masud, A. S. M., *Multiple Objective Decision Making—Methods and Applications,* Springer-Verlag, Berlin, 1979, p. 351.

[3]Stadler, W., "Multicriteria Optimization in Mechanics (A Survey)," *Applied Mechanics Reviews,* Vol. 37, No. 3, 1984, pp. 277–286.

[4]Stadler, W., "Update to Multicriteria Optimization in Mechanics (A Survey)," *Applied Mechanics Update,* American Society of Mechanical Engineers, New York, 1987, pp. 417–420.

[5]Radford, A. D., Gero, J. S., Rosenman, M. A., and Balachandran, M., "Pareto Optimization as a Computer-Aided Design Tool," *Optimization in Computer-Aided-Design,* edited by J. S. Gero, Springer-Verlag, Berlin, 1985, pp. 47–69.

[6]Rao, S. S., "Multiobjective Optimization in Structural Design with Uncertain Parameters and Stochastic Processes," *AIAA Journal,* Vol. 22, No. 11, 1984, pp. 1670–1678.

[7]Rao. S. S., "Game Theory Approach for Multiobjective Structural Optimization," *Computers and Structures,* Vol. 25, No. 1, 1987, pp. 119–127.

[8]Rao, S. S., Venkayya, V. B., and Khot, N. S., "Game Theory Approach for the Integrated Design of Structures and Controls," *AIAA Journal,* Vol. 26, No. 4, 1988, pp. 463–469.

[9]Koski, J., "Norm Methods and Partial Weighting in Multicriterion Optimization of Structures," *International Journal for Numerical Methods in Engineering,* Vol. 24, No. 6, 1987, pp. 1101–1121.

[10]Koski, J., "Multicriterion Optimization in Structural Design," *New Directions in Structural Design,* edited by E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, Wiley, New York, 1984, pp. 483–503.

[11]Koski, J., and Silvennoinen, R., "Pareto Optima of Isostatic Trusses," *Computer Methods in Applied Mechanics and Engineering,* Vol. 31, No. 3, 1982, pp. 265–279.

[12]El-Sayed, M. E. M, Ridgely, B. J., and Sandgren, E., "Nonlinear Structural Optimization Using Goal Programming," *Computers and Structures,* Vol. 32, No. 1, 1989, pp. 69–73.

[13]Sandgren, E., "Structural Design Optimization for Latitude by Nonlinear Goal Programming," *Computers and Structures,* Vol. 33, No. 6, 1989, pp. 1395–1402.

[14]Hajela, P., and Shih, C. J., "Multiobjective Optimum Design in Mixed Integer and Discrete Design Variable Problem," *AIAA Journal,* Vol. 28, No. 4, 1990, pp. 670–675.

[15]Saravanos, D. A., and Chamis, C. C, "Multiobjective Shape and Material Optimization of Composite Structures Including Damping," *AIAA Journal,* Vol. 30, No. 3, 1992, pp. 805–813.

[16]Tseng, C. H., and Lu, T. W., "Minimax Multiobjective Optimization in Structural Design," *International Journal for Numerical Methods in Engineering,* Vol. 30, No. 6, 1990, pp. 1213–1288.

[17]Balachandran, M., and Gero, J. S., "Use of Knowledge in Selection and Control of Optimization Algorithm," *Engineering Optimization,* Vol. 12, No. 2, 1987, pp. 163–173.

[18]Venkayya, V. B., "Design Optimization in Dynamic Environment," Fourth Conference on Recent Advances in Structural Dynamics, Univ. of Southhampton, Southhampton, UK, July 15–19, 1991.

[19]Yang, J. S., and Nikolaidis, E., "Design of Aircraft Wings Subjected to Gust Loads: A Safety Index Based Approach," *AIAA Journal,* Vol. 29, No. 5, 1990, pp. 804–812.

[20]Ditlevsen, O., "Narrow Reliability Bounds for Structural Systems," *Journal of Structural Mechanics,* Vol. 7, No. 4, 1979, pp. 453–472.