AIAA JOURNAL
Vol. 31, No. 1, January 1993

Approximation-Based Global Optimization Strategy for Structural Synthesis

A. E. Sepulveda* and L. A. Schmit†
University of California, Los Angeles, Los Angeles, California 90024

A global optimization strategy for structural synthesis based on approximation concepts is presented. The methodology involves the solution of a sequence of high-quality approximate problems using a global optimization algorithm. The global optimization algorithm implemented consists of a branch and bound strategy based on the interval evaluation of the objective function and constraint functions, combined with a local feasible directions algorithm. The approximate design optimization problems are constructed using first-order approximations of selected intermediate response quantities in terms of intermediate design variables. Numerical results for some example problems known to exhibit relative minima are presented to illustrate the efficacy of the design procedure set forth.

Introduction

THE rather general class of structural optimization problems that can be posed as NLP problems may exhibit relative minima, and it is usually very difficult or impossible to demonstrate that the actual design space is convex. The structural optimization literature contains numerous reports of encounters with the relative minima problem. For example, in the early 1960s it was found that the minimum weight design of integrally stiffened waffle plates[1,2] exhibited relative minima associated with alternative design concepts such as 1) pure thick sheet design, 2) thin sheet with thick stiffeners design, and 3) thin sheet with relatively thin stiffeners design. Furthermore, it was found that the concept corresponding to the global optimum design depended on the total depth available, shifting from concept 1) to 2) and then to 3) as the available depth for the waffle plate increased. In the mid 1960s (see Ref. 3), it was shown that even very simple structural optimization problems such as the cable tied cantilever beam or configuration optimization of a planar three-bar truss could be shown (by mapping two-dimensional design spaces) to exhibit relative minima. Also, the structural synthesis results for integrally stiffened cylindrical shells reported in Ref. 4 indicated that alternative stiffening arrangements could be viewed as distinct design concepts, each associated with its own local optimum design. For example, all inside stiffening, all outside stiffening, or longitudinal outside and circumferential inside stiffening each represent a distinct design concept contained within the general problem formulation of Ref. 4.

In the late 1960s, Moses and Onoda[5] demonstrated (by mapping two-dimensional design spaces and using different starting designs) that the minimum weight elastic design of grillages generally leads to nonconvex programming problems. Subsequently, Kavlie and Moe[6] introduced the extended interior penalty function method and suggested that (with proper choice of the initial response factor r) the SUMT technique described in Ref. 6 could be helpful in guiding the design iteration path toward the global optimum point, at least for the class of elastic grillage problems they examined. In 1974 Reinschmidt and Norabhoompipat[7] and Reinschmidt and Russell[8] presented an innovative method of coping with the relative minima that are found when seeking the minimum weight design of elastic grillages. The essence of the method presented in Refs. 7 and 8 is to initially neglect the conditions of elastic compatibility and first find the minimum weight design of the grillage subject only to equilibrium requirements and stress constraints. The solution of the foregoing alternative problem was found to provide a good starting point (i.e., a design close to the global optimum solution with the compatibility conditions enforced) such that the global solution of the original design problem could be found using a local optimization algorithm.

Design problems that exhibit disjoint feasible regions usually lead to even more severe relative minima problems. For example, the minimum weight design of redundant elastic trusses subject to multiple distinct static load conditions exhibits disjoint feasible regions when members are allowed to vanish. In 1972 a bounding strategy aimed at finding the global minimum weight solution for this class of truss problems was presented in Ref. 9. The essence of the method set forth in Ref. 9 was to use a conventional NLP method (feasible directions) to solve for upper bound estimates of the global minimum and to use LP methods (dual simplex algorithm) to efficiently solve a large number of subproblems (involving only stress constraints, equilibrium requirements, maximum member size limitations, and constraints that declare the omission of certain members) to guide the rational deletion of appropriate members, thus pointing the way to more promising (lower weight) candidate configurations.

Structural optimization problems involving various kinds of dynamic response constraints are also known to exhibit disjoint or severely nonconvex feasible regions. For example, Cassis[10] showed that the feasible domain for simple planar frame structure, subject to a half-cycle sine pulse horizontal ground motion, clearly exhibited disjointness. Also Johnson[11] showed that constraints on the steady-state dynamic response of harmonically loaded structures leads to disjoint feasible regions (in the absence of damping). Furthermore, the two-dimensional design space maps in Figs. 1 and 2 of Ref. 11 clearly show that disjoint feasible regions are, in this case, associated with resonance between the natural and forcing frequencies. In Ref. 12 a systematic method for minimum weight design of lightly damped linear elastic structural systems subject to periodic loading was reported. In this work it was recognized that (in the absence of damping) a different disjoint region, or frequency subspace, is encountered any time a natural fre-

Presented as Paper 91-1035 at the AIAA/ASME/ASCE/AHS 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 8–10, 1991; received Aug. 12, 1991; revision received May 13, 1992; accepted for publication May 15, 1992.

*Assistant Professor, Department of Mechanical Aerospace and Nuclear Engineering. Member AIAA.

†Rockwell Professor of Aerospace Engineering, Emeritus, Department of Mechanical Aerospace and Nuclear Engineering. Fellow AIAA.

quency crosses over a loading frequency value as a result of changing the design during optimization. The solution procedure presented in Ref. 12 was based on 1) using explicit upper bound approximation to replace the time parametric dynamic response constraints, and 2) implementing a multistage mathematical programming scheme that seeks an optimum frequency subspace using a bound-type sorting strategy similar in spirit to that employed in Ref. 9.

For efficient structural synthesis, the approximation concepts method is widely used. In this method an explicit approximate optimization problem is formulated and solved at each design stage. In the mid 1970s approximate representations for constraints and objective functions were generated using first-order Taylor series expansions in terms of direct or reciprocal design variables (e.g., see Refs. 13 and 14). More accurate approximations can be constructed using intermediate design variables and intermediate response quantities. The intermediate design variables and response quantity ideas have been successfully applied to stress constraints,[15] frequency constraints,[16] complex eigenvalues constraints,[17] steady-state response constraints,[18] and transient dynamic response constraints.[19] In this approach, simple approximations of intermediate response quantities in terms of a set of intermediate design variables are used, while retaining the explicit nonlinear dependence of the constraints on the intermediate response quantities and the explicit dependence of the intermediate design variables on the actual design variables. The intelligent choice of intermediate design variables and intermediate response quantities can significantly improve the quality of behavior constraint and objective function approximations. Although these high-quality approximations are often complicated nonlinear functions of the actual design variables, they provide accurate explicit approximate problem statements that capture the inherent nonconvexity of the actual problem.

Although a variety of methods for attacking the global optimization problem can be found in the literature, all of the known algorithms require a very large number of function evaluations (i.e., a large number of analyses). Therefore, in the research reported here, attention has been focused on extending the approximation concepts approach to structural synthesis problems that exhibit nonconvex and/or disjoint feasible regions. The key ingredients of the design optimization method described here are 1) replacement of the actual design optimization problem by a sequence of explicit approximate problems, 2) the use of high-quality explicit approximations (based on intermediate variable and intermediate response quantity concepts) capable of capturing nonconvexity and disjointness, and 3) application of interval arithmetic to effectively implement the exhaustion principle inherent to the branch and bound strategy.

The procedure uses approximation concepts to construct a sequence of highly accurate explicit approximate problems. The global optimum of each approximate problem is obtained using a global optimizer based on the exhaustion principle[20] combined with a feasible directions algorithm. It is important to appreciate that in the work reported here the global optimization algorithm is only applied to the explicit approximate design optimization problems. Clearly, it is reasonable to expect that the use of approximation concepts will reduce the number of actual analyses needed to converge the optimization process via any available global optimization algorithm. However, the fact that each approximate design optimization problem is explicit makes the interval arithmetic implementation of the exhaustion principle attractive because explicitness facilitates generation of inclusion functions that provide an efficient way to test subboxes in the design space (discarding those that are infeasible or cannot contain the global optimum). However, it must be recognized that, even when the global optimum for each approximate problem is obtained, convergence to the global optimum of the actual design problem is strongly dependent on the ability of the approximations used to capture the nonconvexity and/or disjointness inherent to the actual design optimization problems.

Global Optimization

A wide variety of alternative methods for addressing global optimization problems can be found in the literature (e.g., see Refs. 20–27). Most of these approaches fall into one of the following three categories: 1) Kuhn-Tucker and related methods, 2) exhaustion methods, and 3) statistical methods. The Kuhn-Tucker methods aim to solve the Kuhn-Tucker conditions and obtain the set of local minima that includes the global minimum. Exhaustion methods check the whole domain under consideration for a certain property, for example, feasibility, and reduce the region to a smaller one by eliminating the regions that do not exhibit the desired property. Statistical methods are very effective, but the result has only a certain percentage of reliability, and usually for a highly reliable result the number of function evaluations required is very large.

In this work the exhaustion principle inherent to a branch and bound strategy[20] is used to solve the optimization problem. The reason for using the exhaustion principle is that it checks the design space systematically, dealing with the constraints directly, as opposed to penalty-type methods.[21,26] Furthermore, the exhaustion principle lends itself well to combination with local minimization algorithms.

Consider a general optimization problem of the form

$$\text{Min } f(x)$$

such that

$$g_j(x) \le 0 \qquad j = 1, \ldots m$$
$$x \in X \tag{1}$$

where the set X contains the side constraints for the vector of design variables x, and is defined as

$$X = \{x \mid x_i^L \le x_i \le x_i^U, \qquad i = 1, \ldots n\} \tag{2}$$

In what follows the term "box" will be used to designate any n-dimensional interval set of the form of X [see Eq. (2)].

The exhaustion principle is used to discard those areas within the box X that cannot contain the global minimum. This process leads to the determination of a subbox, having prespecified size, that contains the global optimum. The main tool that makes it possible to use the exhaustion principle in an effective manner is interval arithmetic.

Principles of Interval Arithmetic and Inclusion Functions

In this section the basic concepts of interval analysis as applied to the solution of the global optimization problem are briefly reviewed. The work in interval arithmetic began in the late 1950s[28–30] and the basic concept underlying the theory is that, instead of operating with the real values of x, all operations are done using an interval that contains x. One of the most popular applications of interval arithmetic is as a tool for estimating and controlling round-off errors in floating point computations.[32] The basic operations in interval arithmetic are defined (see Refs. 20 and 30) as follows:

$$[a,b] + [c,d] = [a + c, b + d]$$
$$[a,b] - [c,d] = [a - d, b - c] \tag{3}$$
$$[a,b] \cdot [c,d] = [\min\,(ac,ad,bc,bd),\ \max\,(ac,ad,bc,bd)]$$
$$[a,b]/[c,d] = [a,b] \cdot [1/d, 1/c] \quad \text{if} \quad 0 \notin [c,d]$$

(For an extension to the case $0 \notin [c,d]$ see Ref. 30.)

Clearly these definitions are a natural extension of the four basic operations in real analysis. In the treatment of optimization problems using interval arithmetic, the main concept is the application of inclusion functions (Refs. 20 and 30). Let D be a bounded subset of real numbers, f a real function defined on D, $I(D)$ the smallest box that contains D, and I the space of all real intervals. Furthermore, let $\#f(D)=\{f(x)|x\in D\}$ be the range of f over D. An interval function F defined from $I(D)$ to I is called an inclusion function of f if $\#f(D)\subseteq F[I(D)]$, that is, $F[I(D)]$ is a real interval that contains the range of f over D (see Fig. 1). This definition basically states that the inclusion function reflects the bounded behavior of f over the entire domain D. Inclusion functions for vector-valued or matrix-valued functions are defined analogously component wise or element wise.

Interval analysis provides the tools for constructing inclusion functions recursively for a large class of functions. It is assumed that some fundamental functions are available for which inclusion functions are known (e.g., sin, cos, exp, etc., see Ref. 31). The natural interval extension of f to the box X is defined as the interval that is obtained from the expression $f(x)$ by replacing the variable x by the interval X, the arithmetic operations in $\mathbb{R}$ by the corresponding interval arithmetic operations, and the occurrence of a fundamental function by its corresponding inclusion function. The natural interval extension of $f(x)$ to X is denoted by $F(X)$. It can be proved that $F(X)$ is an inclusion function for f.[20] Consider, for example, $f(x_1,x_2)=x_1^2\sin(x_2)$, then the natural interval extension of f is given by $F(X)=X_1\cdot X_1\cdot \mathrm{SIN}(X_2)$, where SIN is a predeclared interval function for sin,[31] and $X=X_1\times X_2\in I^2$ is a box containing x_1 and x_2. It is understood that all interval operations necessary to evaluate $F(X)$ are performed according to the four interval arithmetic operations given by Eqs. (3).

Figure 1 illustrates these concepts graphically. For a generic real function f evaluated on the interval $[a,b]$, the range is shown by the dashed interval $[f_L,f_U]$. The inclusion function F constructed using a natural interval extension as described previously gives the interval $[F_L,F_U]$ when evaluated on the interval $[a,b]$. Since this is an inclusion function, then $[F_L,F_U]$ contains the range of f, i.e., $[f_L,f_U]\subseteq[F_L,F_U]$. Moreover, the error between the estimate of the range given by F and the actual range is linear (or superlinear in some cases) with the length of the interval $[a,b]$.[20] It is important to note that different expressions for a function f lead to interval expressions that are also different functions. The following example drawn from Ref. 20 shows how $F(X)$ is strongly dependent on how $f(x)$ is defined, since the distributive law for algebraic operations does not hold for interval arithmetic. Specifically, $f(x)=x-x^2$, which can also be written as $f(x)=x(1-x)$, gives different interval extensions, for the first case $F([0,1])=[-1,1]$ whereas $F[(0,1)]=[0,1]$ for the second case, and $\#f([0,1])=[0,1/4]$. It is therefore very important to find expressions, for a given function, that lead to the best possible natural interval extensions, that is, $F[(I(D)]$ should approximate $\#f(D)$ as well as possible.

In the context of global optimization, the use of inclusion functions provides an efficient way to systematically test subboxes within the design space and discard those subboxes that are either infeasible or cannot contain the global optimum.

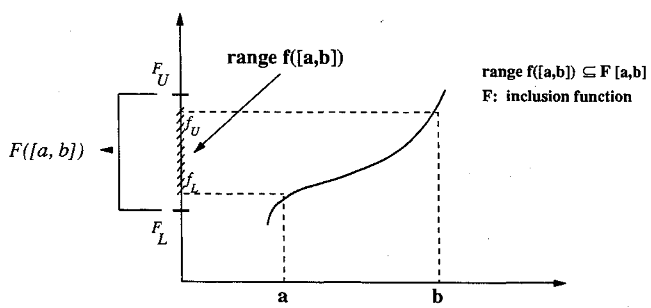

Fig. 1 Inclusion functions.

Basic Global Optimization Algorithm[20]

In this section the basic interval algorithm for solving the constrained problem stated in Eq. 1 is described. The algorithm is based on a branch and bound strategy that uses the exhaustion principle with the aid of inclusion functions constructed via interval arithmetic. The main ideas underlying the method are the following:

1) It systematically subdivides the initial box into smaller subboxes (exhaustion principle).

2) It discards all of those subboxes that are either infeasible by interval evaluation of the constraints or that cannot contain the global optimum by comparing the interval evaluation of the objective function with the current best upper bound for the objective function.

The aim of the algorithm is then to determine a box X^*, of a prescribed size, such that the global optimum $x^*\in X^*$.

The basic global algorithm starts with an initial box that contains the feasible region, usually X from Eq. (2) that represents the side constraints for the design variables. This box is bisected into smaller subboxes, creating a tree of problems (in a branch and bound context). The vertices of the tree correspond to a problem in which a subset of the variables has been bisected to create subboxes from the original box. These vertices are arranged into levels in the following way: there is a single vertex at level 0 (the root of the tree) that corresponds to the original box X. Subsequent levels (branching) are generated by choosing a variable x_i and bisecting X to generate two vertices, each one corresponding to half the original box. Therefore, for higher number levels within the tree, the size of the subbox to be analyzed is reduced. Figure 2 shows an example where three levels have been generated. The variable chosen for bisection is the one corresponding to the direction in which the current box has an edge of maximum length. Alternately, first-order information could also be used at this stage to select the direction for bisection (see Ref. 32).

As will be shown later, the algorithm performs more efficiently if an upper bound, for the objective function, is provided. Usually this is accomplished by using a local technique to find a feasible point (x_0) for the problem and setting the initial upper bound to $\tilde{f}=f(x_0)$. If no initial upper bound is

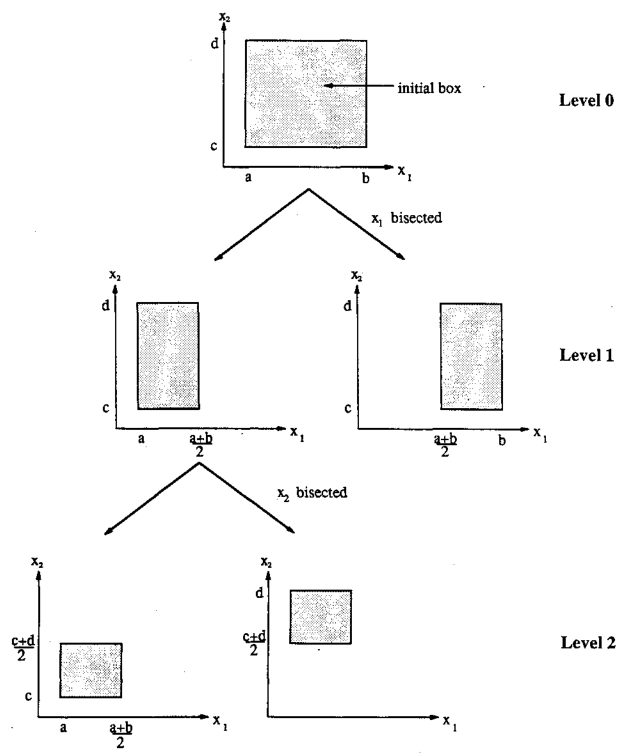

Fig. 2 Branch and bound tree with three levels.

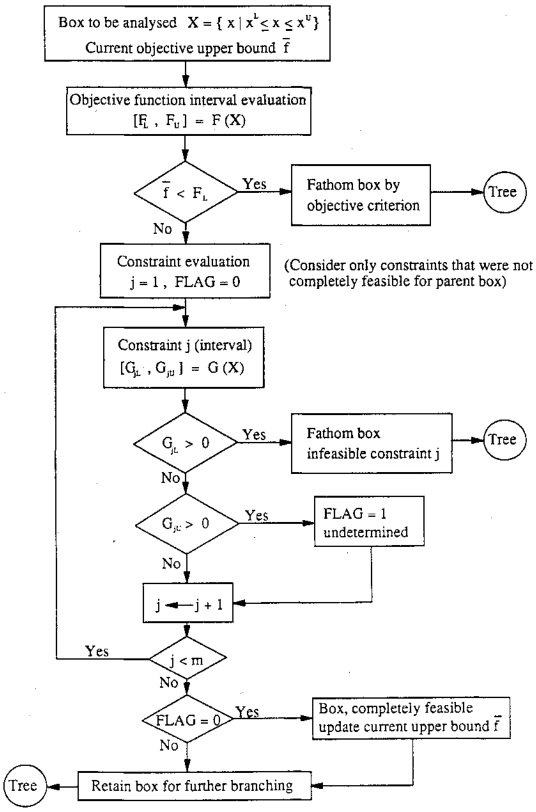

Fig. 3 Fathoming and branching tests for a box.

provided, then $\bar{f} = +\infty$. In both cases, the algorithm converges, but the computational effort increases dramatically if the upper bound provided is too high. Therefore, it is important to provide the lowest possible upper bound $\bar{f}$. In our implementation, the initial feasible point x_0, is, in fact, a local minimum for the problem.

For every vertex of the tree, the objective function and the constraints are evaluated in an interval sense, giving $[F_L, F_U]$ and $[G_{jL}, G_{jU}]$, $j = 1, \ldots, m$. The following tests determine if the vertex has to be subsequently branched (see Fig. 3):

1) If $\bar{f} < F_L$, the vertex and its descendent vertices do not contain a better solution than the current one and they are eliminated from further analysis or fathomed.

2) If $G_{jL} > 0$ for some $j = 1, \ldots, m$, then the vertex is infeasible and therefore fathomed.

3) If $G_{jU} \leq 0$ for all $j = 1, \ldots, m$, then the vertex is completely feasible and the objective function upper bound can be updated. In Ref. 20, it is suggested that the current upper bound be updated using $\bar{f} = \text{Min}\ [(\bar{f}, ub\ F(c_j)]$, where c_j is the center of the vertex box and $ub\ F(c_j)$ is an upper bound for F at c_j. An alternative, which is used in this work, is to find the minimum of $f(x)$ over the box using a local optimizer. This is a very simple optimization problem since the only constraints are side constraints for the variables. Lower and upper bounds for the objective function at the vertex are readily available from F_L and F_U, and the box is retained for further branching.

4) If for some $j = 1, \ldots, m$ $G_{iL} \leq 0$ and $G_{jU} > 0$, then the vertex is undetermined, i.e., it is not possible to decide if the box is feasible or not. If this is the case, the vertex will be fathomed if $\bar{f} < F_L$ (test 1) or it will be retained for further branching.

It is important to note that test 1 is applied before the feasibility of the box is determined, thus, the interval evaluation of the constraints is only necessary when the vertex is not fathomed. Also, since for each vertex the domain is contained in the domain of the parent vertex, only those constraints that were undetermined in the previous level have to be evaluated. The procedure for the analysis of a given box in the tree is summarized in Fig. 3. When the level has been completely analyzed, it is efficient, if possible, to further reduce the number of active vertices in the tree by applying a criterion called the midpoint test.[20] The midpoint test consists of comparing the available lower bound for each vertex (F_L) with the current upper bound for the tree ($\bar{f}$). Clearly if $\bar{f} < F_L$, the associated box cannot contain the global optimum; therefore, it is fathomed. Since the upper bound has been updated (test 3), this test increases the possibility of fathoming previously retained vertices.

The procedure is applied until 1) all vertices are fathomed (the problem is infeasible or x_0 is the global optimum), 2) $|1 - F_L/\bar{f}| \leq \epsilon_f$, for ϵ_f a given tolerance, and the corresponding vertex (box) contains the global optimum, or 3) the size of the boxes contained in the last level generated is less than a prescribed value. Usually when the algorithm stops because of criteria 2) or 3), there are several retained boxes for the last level of the branch and bound tree. Most of these boxes are adjacent and their union defines the region where the global optimum is contained. For computational purposes, the final region is determined as the smallest box that contains all of the retained boxes. In the case that the union of the final boxes contains two or more separate regions, the algorithm gives the box corresponding to the region containing the lowest lower bound.

The basic algorithm described previously may be further modified by incorporating accelerating devices that make the overall optimization more efficient. The first device implemented is to search for an initial feasible point (preprocessing device). If such a feasible point is known, the application of the basic algorithm will be much more efficient. The reason for this is that the initial minimum function value is set to the value of f at the feasible point. This value is then used in test 1 and the midpoint test to eliminate boxes that may not contain any global optima, thus preventing superfluous processing. If such a value is not known, then test 1 or the midpoint test cannot be applied. Further improvement can be achieved, as mentioned earlier, if a local optimum solution of the optimization problem is provided.

The second kind of accelerating devices are called internal devices, and they include procedures used to improve feasible

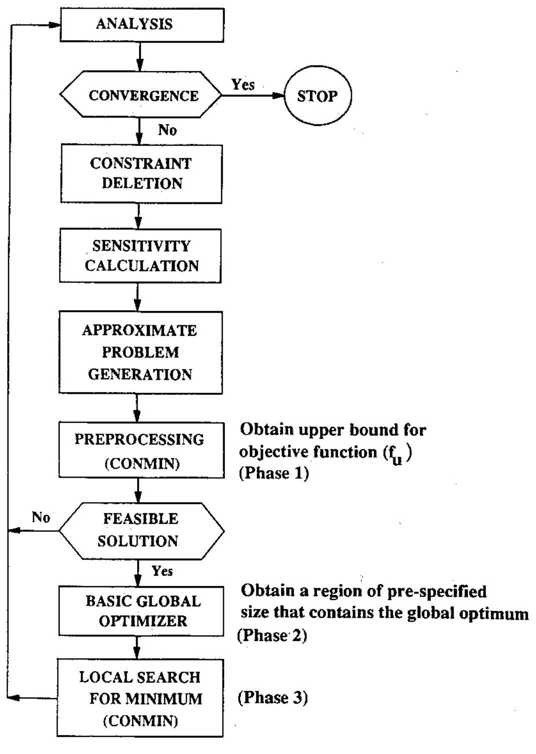

Fig. 4 Optimization procedure.

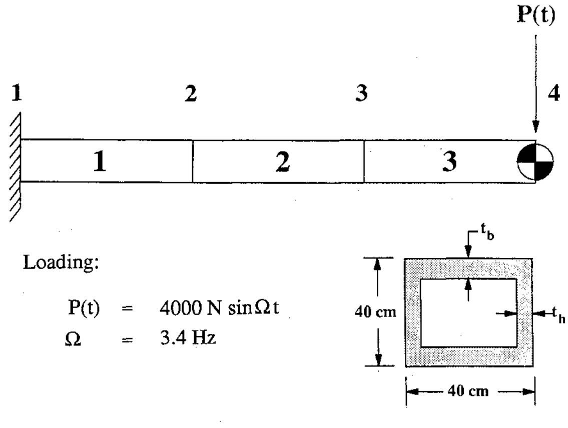

Fig. 5 Example problem 1, disjoint design space.

points to make the midpoint test more efficient. There are two types of internal devices: 1) local devices and 2) global devices. The local devices use point information, and they are used to improve the feasible points locally. In this work, a local search is used to determine the feasiblity of undetermined subboxes when the number of undetermined boxes exceeds a prespecified number. The global devices use all of the information available on a current subbox, and therefore they are interval procedures. These kinds of procedures are only used if the current subbox is feasible. In this work, the monotonicity[20] test is applied to feasible boxes to reduce their dimension and eliminate unnecessary computations. The monotonicity test is applied to any feasible box and requires the interval evaluation of the gradient of the objective function

$$\nabla F(Y) = \prod_{i=1}^{n} [lb(\partial f/\partial X_i)(Y),\ ub\ (\partial f/\partial X_i)(Y)]$$

Clearly if $0 \notin [lb(\partial f/\partial X_i)(Y),\ ub\ (\partial f/\partial X_i)(Y)]$, then f is strictly monotone with respect to the ith coordinate (since the gradient does not change sign), and the optimum resides on one of the edges of the subbox. Therefore, according to the sign of $lb(\partial f/\partial x_i)$ or $ub(\partial f/\partial x_i)$, the box Y is reduced by setting x_i to upper or lower bound, (i.e., the dimension of the box is reduced).

Global Optimization Algorithm

The basic global algorithm should be combined with a local optimizer to accelerate the overall process and make it more efficient. The global optimizer used in this work has three basic phases (see Fig. 4): 1) preprocessing, 2) basic global algorithm, and 3) local search for minimum.

The purpose of the preprocessing phase is to supply a good starting point to the basic algorithm. In this phase it is determined whether or not the feasible region is empty, and if it is not, the preprocessor finds a local minimum for the problem. The objective function evaluated at this local minimum is used as an upper bound of the objective function for the basic global algorithm. This upper bound value is then used in test 1 and in the midpoint test to eliminate boxes that cannot contain the global minimum, thus reducing the number of constraint function evaluations. If the feasible domain is not empty, the application of the basic global algorithm (phase 2) gives a box of a pre-established size that contains the global optimum. Finally, in phase 3, the local minimizer is used to search for an optimum design within the smaller region established at the end of the previous phase 2 pass. In this work the computational implementation of phases 1 and 3 is carried out using CONMIN.[33]

Global Optimization and Approximation Concepts

An efficient approach to global design synthesis is to generate and solve a sequence of approximate optimization problems. At the beginning of each design stage, the system is analyzed to determine the value of the behavior constraints. To lower the cost of the sensitivity analysis, constraints that are not active or potentially active are then deleted from the design problem. The first-order analytical sensitivities of appropriate intermediate response quantities with respect to the intermediate design variables are calculated and used to construct approximate optimization problems. These problems are nonlinear but explicit, and therefore they are inexpensive to solve compared with the solution of the actual problem. Each approximate optimization problem is solved using the global optimization algorithm, described in the previous section. It is important to note that the efficient application of interval arithmetic to find inclusion functions is made possible by the availability of algebraically explicit functions for the objective function and constraint functions of the approximate problem. The new design point, produced at the end of the previous global optimization stage, is then analyzed at the beginning of the next stage. This process is continued until the objective function changes less than some prescribed amount for three consecutive stages. The entire synthesis process is summarized in Fig. 4.

Numerical Results

The previously described approximation-based global optimization strategy has been implemented in a computer program that can be applied to a wide range of structural design optimization problems, namely, those for which high-quality explicit approximations capable of capturing nonconvexity and disjointness can be generated. The example problems presented are used to assess the applicability of the proposed method when applied to optimum structural design problems known to exhibit nonconvexities in the design space.

Problem 1: Disjoint Design Space

This first example problem is used to graphically illustrate the optimization procedure. The problem involves minimum mass design of a cantilever beam with a fixed tip mass (200 kg) subject to the loading shown in Fig. 5. The beam is modeled with three prismatic frame elements, each having a square (40×40 cm) cross section with independent flange (t_b) and web (t_h) wall thicknesses. The total length of the beam is 1000 cm and is made of a material having the following properties: $\rho = 2.768 \times 10^{-3}$ kg/cm³, $E = 7.1 \times 10^{6}$ N/m². The structure is subjected to a harmonic load given by $P(t) = 4000N \sin \Omega t$, where the forcing frequency $\Omega = 3.4$ Hz.

The design variables for this problem are the web and flange thicknesses (t_h and t_b) of the beam elements, linked along the length of the beam. The side constraints for these variables are $0.5\ \text{cm} \leq t_b,\ t_h \leq 10.0$ cm. A behavior constraint is imposed on the tip dynamic steady-state displacement ($|w_4| \leq 10.0$ cm).

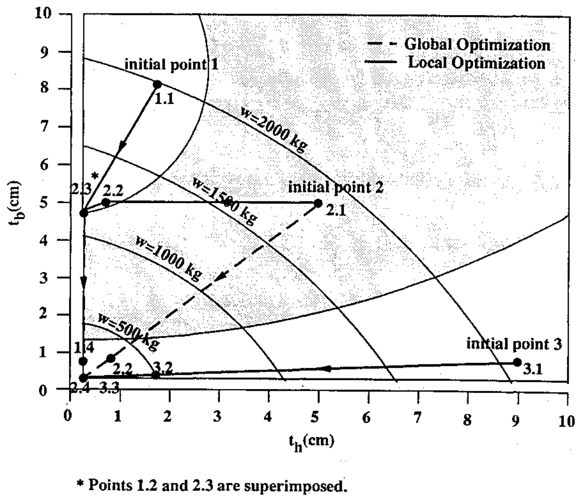

Fig. 6 Optimal trajectories for problem 1.

This problem is similar in spirit to the axial and torsional example problems presented in Refs. 11 and 12, since the forcing frequency is near the first natural frequency the design space is disjointed because of resonance (see Fig. 6). The global optimum is obtained at $t_b = t_h = 0.5$ cm. High-quality approximations for the steady-state dynamic displacement are generated using a modal approach as described in Ref. 18. For these high-quality approximations, modal energies ($T_n = \{\phi_n\}^T[M]\{\phi_n\}$ and $U_n = \{\phi_n\}^T[K]\{\phi_n\}$) are used as intermediate response quantities, and cross-sectional properties (A,I) are used as intermediate design variables. Move limits of 80% were used in this problem.

Three different initial points were used to test the algorithm: point 1—$t_b = 8$ cm, $t_h = 2$ cm; point 2—$t_b = 5$ cm, $t_h = 5$ cm; and point 3—$t_b = 1$ cm, $t_h = 9$ cm. Figure 6 shows the sequence of design points generated from these three starting points using the global algorithm and a local optimizer to solve only the approximate problems. It is to be understood that the design points plotted in Fig. 6 correspond to the optimal solution for each approximate problem. The trajectory corresponding to the use of a local optimizer to solve the approximate problems is shown using a continuous line, whereas a discontinuous line is used for the case of the global optimizer. It is observed that, using the local optimizer, the optimal solution obtained from initial points 1 and 2 is $t_b = 4.57$ cm and $t_h = 0.5$ cm, which is a local minimum for the problem, and the trajectories are 1.1–1.2 and 2.1–2.3, respectively. On the other hand, for these two cases the global algorithm is able to find the global optimum and the trajectories are 1.1–1.4 for initial point 1 and 2.1–2.4 for initial point 2. Finally, for initial point 3, both algorithms give the same trajectory to the global optimum (3.1–3.3). This is because point 3 is contained in the feasible region that includes the global optimum. To further illustrate how each approximate problem is solved using the global optimizer, Fig. 7 shows the basic steps for the solution of the approximate problem constructed at initial point 2. Figure 7 shows the approximate constraints and the move limits box for this problem. In the first phase (preprocessing), CONMIN is used to find a feasible point, which in fact corresponds to a local minimum for the approximate problem. The value of the objective function at this point is used as an upper bound for the basic global algorithm. The basic global algorithm generates a subbox (see Fig. 7) that contains the global optimum for the approximate problem. Finally, starting from the centroid of this subbox, CONMIN is used to search locally for the optimum, which in this case is the lower left corner of the domain. This point is then the global optimum for the approximate problem, and it is used to construct the next approximate problem in the sequence.

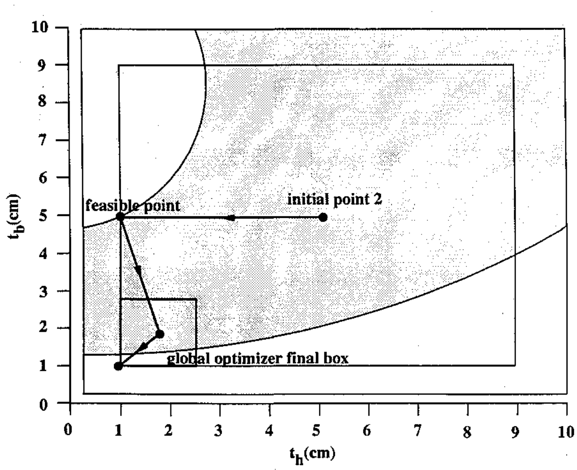

Fig. 7 Solution scheme for the approximate problem constructed about initial point 2 (problem 1).

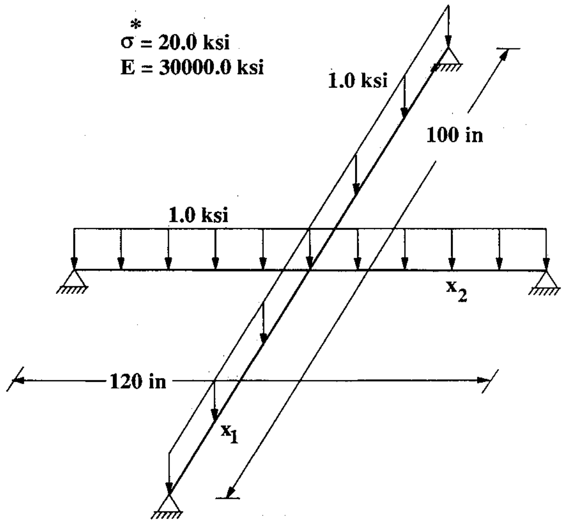

Fig. 8 Example problem 2, grillage structure.

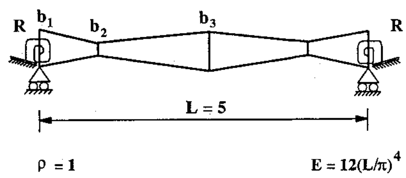

Fig. 9 Optimal trajectories for problem 2.

Problem 2: Grillage Structure, Nonconvex Design Space

The second problem is the two-beam grillage structure shown in Fig. 8 subjected to distributed static loads. This example problem has been treated in Refs. 5, 6, and 23. The design variables are the cross-sectional areas of each beam (x_1,x_2). These are related to the moments of inertia (I_1,I_2) and the section moduli (Z_1,Z_2) by empirical relations given by[5,6]:

$$Z_i = (x_i/1.48)^{1.82} \tag{4a}$$

$$I_i = 1.007\ (x_i/1.48)^{2.68} \qquad i = 1,2 \tag{4b}$$

The structure is designed for minimum weight (volume) subjected to constraints on allowable stresses ($\sigma^* = 20$ ksi) at the centerpoint and at the location of maximum absolute bending moment along each of the spans. The design space for this problem is shown in Fig. 9. The nonconvexity in the design space can be clearly seen.

From the analysis results, the moment can be computed, and the stress found from the following equation:

$$\sigma_j = M_j/Z_j \tag{5}$$

For high-quality approximations, the moments M_j are linearly approximated in terms of the moments of inertia I_i (Ref. 15). Thus, I_i and Z_i are used as intermediate design variables. In this problem, 80% move limits on the actual design variables x_i were used.

Four different initial points were used to test the algorithm: point 1—$x_1 = 6$ cm^2, $x_2 = 26$ cm^2; point 2—$x_1 = 13$ cm^2, $x_2 = 20$ cm^2; point 3—$x_1 = 15$ cm^2, $x_2 = 30$ cm^2; and point 4—$x_1 = 5$ cm^2, $x_2 = 20$ cm^2.

Figure 9 shows the sequence of design points generated from these four starting points using the global algorithm, and the iteration histories are given in Tables 1–4. It is observed that for all cases the global algorithm finds the global optimum for the problem within a difference of 0.09% in objective function value. The local optimization using CONMIN

Table 1 Iteration history for problem 2, grillage structure, initial point 1

Analysis number	Design variables, in.^{2} x_1	x_2	Volume, in.^{3}	Constraint violation, %
0	6.00	26.00	3720	0.0
1	6.42	24.69	2605	3.0
2	11.91	24.34	4111	0.0
3	20.77	7.54	2481	22.0
4	23.46	7.33	3226	0.0
5	23.40	7.17	3200	0.0
6	23.40	7.17	3200	0.0
7	23.40	7.17	3200	0.0

Table 2 Iteration history for problem 2, grillage structure, initial point 2

Analysis number	Design variables, in.^{2} x_1	x_2	Volume, in.^{3}	Constraint violation, %
0	13.00	20.00	3700	0.0
1	12.99	18.86	3562	0.3
2	18.46	16.19	3789	0.7
3	22.58	13.52	3881	0.0
4	24.37	9.72	3603	0.0
5	23.62	7.84	3303	0.0
6	23.43	7.20	3207	0.0
7	23.41	7.14	3197	0.0
8	23.41	7.14	3197	0.0

Table 3 Iteration history for problem 2, grillage structure, initial point 3

Analysis number	Design variables, in.^{2} x_1	x_2	Volume, in.^{3}	Constraint violation, %
0	5.00	20.00	2900	53.1
1	6.25	24.51	3566	5.5
2	11.88	17.13	3243	18.5
3	20.31	17.21	3497	13.8
4	24.81	8.78	3334	0.0
5	23.55	7.62	3269	0.0
6	23.42	7.21	3207	0.0
7	23.40	7.16	3199	0.0
8	23.40	7.16	3199	0.0

Table 4 Iteration history for problem 2, grillage structure, initial point 4

Analysis number	Design variables, in.^{2} x_1	x_2	Volume, in.^{3}	Constraint violation, %
0	15.00	30.00	5100	0.0
1	11.16	20.33	3069	10.4
2	20.22	8.72	3069	25.3
3	23.66	7.93	3317	0.0
4	23.45	7.32	3223	0.0
5	23.42	7.16	3201	0.0
6	23.42	7.16	3201	0.0
7	23.42	7.16	3201	0.0

converged to the first local minimum (point A) from points 1 and 4 and to the second local minimum (point B) from points 2 and 3.

Problem 3: Beam Structure, Nonconvex Objective Function

The third example involves the symmetric beam shown in Fig. 10, and it has been previously studied in Ref. 34. The rotation at both ends is resisted by rotational springs of stiffness $\boldsymbol{R}$. The finite element model consists of four tapered beam elements having rectangular cross section with unit depth. The heights b_1, b_2, and b_3 at the ends of the tapered elements are chosen as the design variables. The optimization problem involves maximization of the smallest natural frequency subjected to an upper limit constraint on the beam volume of 10, and lower limit side constraints of 0.1 on the design variables.

For $\boldsymbol{R}=100$ (shown as $\boldsymbol{R}\approx\infty$ in Fig. 11), a large value corresponding to clamped ends, the objective function has

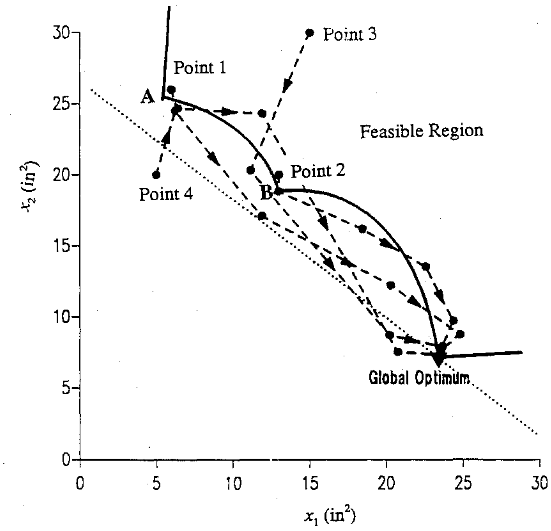

Fig. 10 Example problem 3, nonconvex objective function.

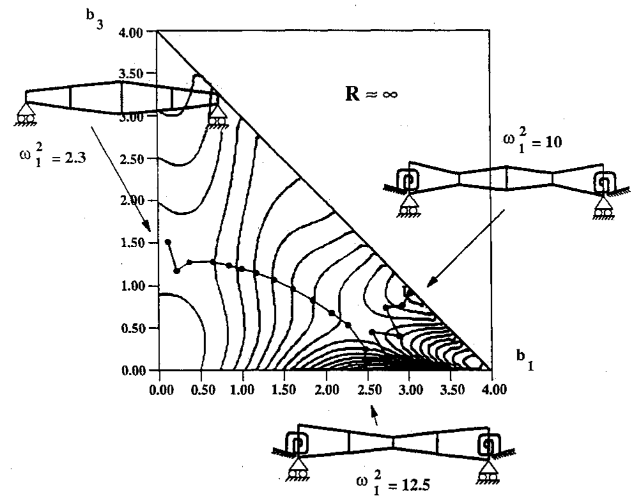

Fig. 11 Optimal trajectories for problem 3.

three local minima as shown in Fig. 11. In this figure, the volume equality constraint was used to restate the problem in terms of b_1 and b_3 only and therefore to facilitate the graphical representation of the isocurves for the objective function. This is permissible in view of the fact that the upper limit volume constraint is always active at the optimum design.

Accurate approximations are constructed for the natural frequencies using a Rayleigh quotient approximation (Ref. 16), and the section properties (A, I) at the end of each element are used as intermediate design variables. In this problem, 80% move limits on the intermediate design variables were used.

Two different starting points were used to test the algorithm, namely, the two local minima with smaller objective function values (i.e., $b_1=0.1$, $b_2=1.2$, and $b_3=1.5$ where $\omega_1^2=2.3$ and $b_1=3.0$, $b_2=0.1$, and $b_3=0.9$ where $\omega_1^2=10$). The optimal trajectories in the (b_1, b_3) space are shown in Fig. 11. For both cases the global optimization algorithm is able to find the global optimum for the problem ($b_1=2.5$, $b_2=0.7$, and $b_3=0.1$ where $\omega_1^2=12.5$).

Problem 4: 23 Design Variable Problem

This problem involves mass minimization of the structure shown in Fig. 12. The material properties for both beams are $\rho=2.768\times10^{-3}$ kg/cm^3 and $E=7.1\times10^6$ N/cm^2. The structure is subjected to a harmonic load given by $P(t)=40000\ N$ sin Ωt, where the forcing frequency $\Omega=10$ Hz. The finite element model and the cross sections for both beams are shown in Fig. 12. The design variables for this case are flange (t_b) and web (t_h) wall thicknesses for the beam elements and the nonstructural point masses located at nodes 3, 7, and 9. Linking is imposed such that t_b and t_h are the same in the following groups of elements: 9 and 12, 10 and 11. These linking conditions reduce the total number of independent design variables from 27 to 23 (i.e., 10 independent t_b, 10 independent t_h, and 3 independent masses). The side constraints are $0.1 \text{ cm} \leq t_b, t_h \leq 10.0$ cm for the structural sizing variables, and $0.0 \text{ kg} \leq m \leq 30.0$ kg for the nonstructural mass variables. A behavior constraint is also imposed on the dynamic steady-state vertical displacement at node 9 ($|w_9| \leq 10$ cm). High-quality approximations for the steady-state dynamic displacements were generated using the same approach as in problem 1. Move limits of 60% on the actual design variables were used in this problem.

Table 5 gives the final designs for solutions obtained from two different starting points using 1) local optimization only and 2) the global optimization strategy presented here. It is observed that the local optimization yields different optimal solutions corresponding to two different local minima. On the other hand, the global optimization strategy leads to final designs that are essentially the same having an objective function difference of only 0.2%. Figure 13 shows the iteration history generated by the global optimization strategy from both starting points. In all four cases the displacement constraint is active at the optimum. It is interesting to observe that the two local minima solutions (see Table 5) can be given physical interpretations. The local minimum found from point 1 exhibits the typical characteristics of a vibration absorber where most of the mass is concentrated at the tip of the cantilever beam. The second local minimum is a structural design in which the three balancing masses have small values.

To measure the computational effort involved in the global optimization procedure, Table 5 gives the total CPU times (IBM 3090) for all four runs. Even though the CPU time is strongly dependent on the computer architecture and quality of the code, it gives an estimate of the relative cost between the global and local procedure. For runs starting at point 1, the CPU time ratio between the global and local optimization is 1.8, and the ratio for runs starting at point 2 is 4.1. The larger ratio for the case starting at point 2 compared with point 1 is expected, since the local optimizer does not give a good upper bound, and thus fewer boxes can be fathomed by using the objective function criterion.

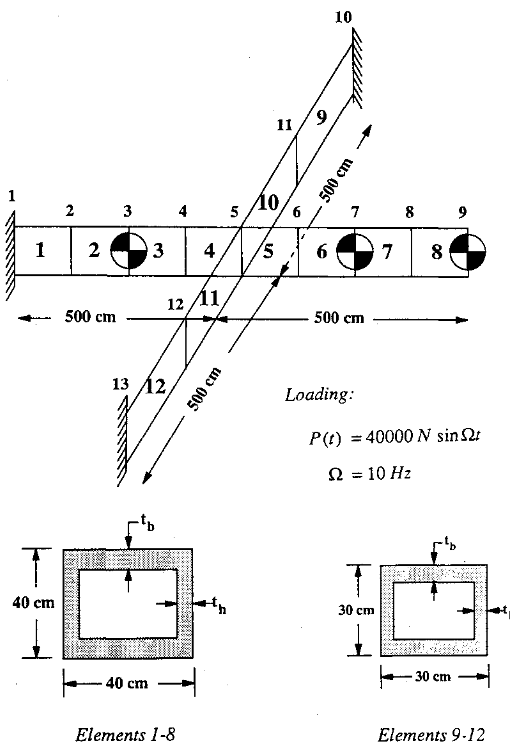

Fig. 12 Twenty-three design variable problem.

Table 5 Optimization results, problem 4

Beam element	Variable	Initial point 1	Optimal solution Local	Optimal solution Global	Initial point 2	Optimal solution Local	Optimal solution Global
		Initial and optimal design, cm, kg					
1	t_b	0.5	0.1	0.1	0.5	0.650	0.1
	t_h	0.5	0.1	0.1	0.5	0.395	0.1
2	t_b	0.5	0.1	0.1	0.5	0.614	0.1
	t_h	0.5	0.1	0.1	0.5	0.395	0.1
3	t_b	0.5	0.1	0.1	0.5	1.516	0.1
	t_h	0.5	0.1	0.1	0.5	0.395	0.1
4	t_b	0.5	0.1	0.1	0.5	2.301	0.1
	t_h	0.5	0.1	0.1	0.5	0.394	0.1
5	t_b	0.5	0.1	0.1	0.5	2.260	0.1
	t_h	0.5	0.1	0.1	0.5	1.532	0.1
6	t_b	0.5	0.1	0.1	0.5	1.316	0.1
	t_h	0.5	0.1	0.1	0.5	0.1	0.1
7	t_b	0.5	0.1	0.1	0.5	0.458	0.1
	t_h	0.5	0.1	0.1	0.5	0.1	0.1
8	t_b	0.5	1.993	0.102	0.5	0.1	1.8472
	t_h	0.5	2.052	3.778	0.5	0.1	2.2074
9,12	t_b	0.5	0.1	0.1	0.5	1.294	0.1
	t_h	0.5	0.1	0.1	0.5	0.743	0.1
10,11	t_b	0.5	0.1	0.1	0.5	1.229	0.1
	t_h	0.5	0.1	0.1	0.5	0.726	0.1
Mass 3	m	30	0.722	0.001	30	1.522	0.001
Mass 7	m	30	1.152	0.118	30	1.522	0.001
Mass 9	m	30	30.000	30.000	10	0.494	30.000
Optimal mass, kg			190.07	188.87		473.65	188.49
Number of analyses			10	12		12	10
CPU time, s			6.1	11.2		9.1	37.1

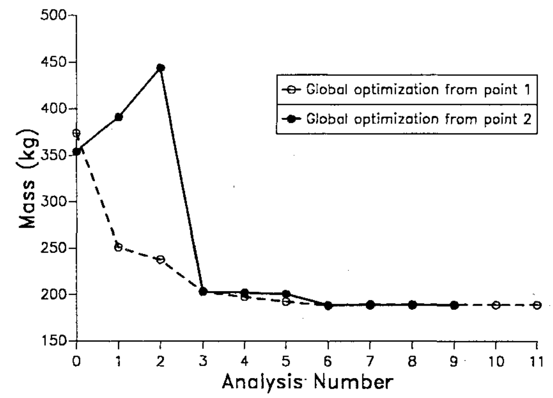

Fig. 13 Iteration history for problem 4, 23 design variables.

Conclusions

In this paper, a methodology aimed at finding the global optimum for structural synthesis problems is set forth. The methodology is based on the solution of a sequence of explicit approximate problems constructed using intermediate design variables and intermediate response quantities. Each of the approximate problems is solved for the global optimum using a combination of local techniques and a basic global optimization algorithm based on interval arithmetic. It is important to recognize that the basic global algorithm does not require approximate problem sensitivity information if the monotonicity test is not implemented.

In all of the example problems, the number of real analyses required to achieve convergence is less than 16. The number of function evaluations using the global algorithm are, for the problems solved, 6 to 10 times the number of function evaluations required by CONMIN. But even if this number is relatively high, the computational requirement is not severe since only the evaluation of the explicit approximate functions is necessary for the solution of the approximate design optimization problems.

Numerical results indicate the following: 1) the approximate problems capture the nonconvexity of the actual problem, 2) the local-global algorithm is able to find the global optimum for each approximate problem, and 3) the local-global strategy gives significantly better optimal designs than a strictly local technique.

Acknowledgments

The sponsorship provided by the NASA Langley Research Center under NASA Grant NSG-1490 is gratefully acknowledged. The authors wish to thank Alejandro Diaz of the Department of Mechanical Engineering, Michigan State University, for his generous cooperation in providing the computer code for analysis and the design space map for problem 3.

References

[1]Schmit, L. A., Kicher, T. P., and Morrow, W. M., "Structural Synthesis Capability for Integrally Stiffened Waffle Plates," *AIAA Journal,* Vol. 1, No. 12, 1963, pp. 2820–2836.

[2]Schmit, L. A., and Kicher, T. P., "Structural Synthesis of Symmetric Waffle Plates," NASA TN D-1691, Dec. 1962.

[3]Reinschmidt, K. F., Cornell, A. C., and Brotchie, J. F., "Iterative Design and Structural Optimization," *Journal of the Structural Division,* American Society of Civil Engineers, Vol. 92, No. ST6, 1966, pp. 281–318.

[4]Morrow, W. M., and Schmit, L. A., "Structural Synthesis of a Stiffened Cylinder," NASA CR-1217, Dec. 1968.

[5]Moses, F., and Onoda, S., "Minimum Weight Design of Structures with Application to Elastic Grillages," *International Journal for Numerical Methods in Engineering,* Vol. 1, No. 4, 1969, pp. 311–331.

[6]Kavlie, D., and Moe, J., "Automated Design of Frame Structures," *Journal of the Structural Division,* American Society of Civil Engineers, Vol. 97, No. 511, 1971, pp. 33–62.

[7]Reinschmidt, K. F., and Norabhoompipat, T., "Structural Optimization by Equilibrium Linear-Programming," *Proceedings of the 6th National Conference on Electronic Computation,* ASCE, Georgia Inst. of Technology, Atlanta, GA, Aug. 7–9, 1974, pp. 581–604.

[8]Reinschmidt, K. F., and Russell, A. D., "Applications of Linear Programming in Structural Layout and Optimization," *Computers and Structures,* Vol. 4, No. 4, 1974, pp. 855–869.

[9]Sheu, C. Y., and Schmit, L. A., "Minimum Weight Design of Elastic Redundant Trusses Under Multiple Static Loading Conditions," *AIAA Journal,* Vol. 10, No. 1, 1972, pp. 155–162.

[10]Cassis, J. H., "Optimum Design of Structures Subjected to Dynamic Loads," Univ. of California at Los Angeles, School of Engineering and Applied Science, UCLA-ENG-7451, Los Angeles, CA, June 1974.

[11]Johnson, E. H., "Disjoint Design Spaces in the Optimization of Harmonically Excited Structures," *AIAA Journal,* Vol. 14, No. 2, 1976, pp. 259-261.

[12]Mills-Curran, W. C., and Schmit, L. A., "Structural Optimization with Dynamic Behavior Constraints,"*AIAA Journal,* Vol. 23, No. 1, 1985, pp. 136–138.

[13]Schmit, L. A., and Farshi, B., "Some Approximation Concepts for Efficient Structural Synthesis," *AIAA Journal,* Vol. 12, No. 5, 1974, pp. 692–699.

[14]Schmit, L. A., and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA CR 2552, March 1976.

[15]Vanderplaats, G. N., and Salajegheh, E., "A New Approximation Method for Stress Constraints in Structural Synthesis," *AIAA Journal,* Vol. 27, No. 3, 1989, pp. 352–358.

[16]Canfield, R. A., "An Approximation Function for Frequency Constrained Structural Optimization," *AIAA Journal,* Vol. 28, No. 6, 1990, pp. 1116–1122.

[17]Thomas, H. L., Sepulveda, A. E., and Schmit, L. A., "Improved Approximations for Control Augmented Structural Synthesis, *AIAA Journal,* Vol. 30, No. 1, 1992, pp. 171–179.

[18]Thomas, H. L., Sepulveda, A. E., and Schmit, L. A., "Improved Approximation for Dynamic Displacements Using Intermediate Response Quantities," *Third NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization,* San Francisco, CA, Sept. 24–26, 1990, pp. 95–104.

[19]Sepulveda, A. E., Thomas, H. L., and Schmit, L. A., "Improved Transient Response Approximations for Control Augmented Structural Optimization," *Proceedings of the Second Pan American Congress of Applied Mechanics,* Valparaiso, Chile, Jan. 2–5, 1991, pp. 611–614.

[20]Ratschek, H., and Rukne, J., *New Computer Methods for Global Optimization,* Wiley, New York, 1988.

[21]Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing,* Wiley, New York, 1988.

[22]Pardalos, P. M., and Rosen, J. B., "Constrained Global Optimization: Algorithms and Applications," *Lecture Notes in Computer Science,* Vol. 268, Springer-Verlag, Berlin, 1987.

[23]Hajela, P., "Genetic Search—An Approach to the Nonconvex Optimization Problem," *AIAA Journal,* Vol. 28, No. 7, 1990, pp. 1205–1210.

[24]Dunlop, A. E., and Kernigham, B. W., "A Procedure for Placement of Standard-Cell VLSI Circuits," *IEEE Transactions on Computer-Aided Design,* CAD-4, Vol. 1, 1985, pp. 92–98.

[25]Kirkpatrick, S., Gellatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science,* Vol. 220, No. 4596, 1983, pp. 671–680.

[26]Hopfield, J. J., and Tank, D. W., "Neural Computations of Decisions in Optimization Problems," *Biological Cybernetics,* Vol. 52, No. 3, 1985, pp. 141–152.

[27]Levy, A. V., and Montalvo, A., "The Tunnelling Algorithm for the Global Minimization of Functions," *SIAM Journal Scientific and Statistical Computing,* Vol. 6, 1985, pp. 15–29.

[28]Moore, R. E., "Automatic Error Analysis in Digital Computation," Lockheed Missiles and Space Division, TR LMSD-4842, Sunnyvale, CA, 1959.

[29]Sunaga, T., "Theory of an Interval Algebra and its Application to Numerical Analysis," *RAAG Memoirs,* Vol. 2, 1958, pp. 547–564.

[30]Moore, R. E., *Internal Analysis,* Prentice-Hall, Englewood Cliffs, NJ, 1966.

[31]IBM High-Accuracy Arithmetic Subroutine Library, Program Description and User's Guide, IBM Publication, Mechanicsburg, PA, 1986.

[32]Sepulveda, A. E., and Schmit, L. A., "Optimal Placement of Actuators and Sensors in Control Augmented Structural Optimization," *International Journal for Numerical Methods in Engineering,* Vol. 32, No. 6, 1991, pp. 1165–1187.

[33]Vanderplaats, G. N., "CONMIN—A Fortran Program for Constrained Function Minimization," NASA TM X-62, 682, Aug. 1973.

[34]Diaz, A. R., Shaw, S. S., and Pen, J., "On Domains of Convergence in Optimization Problems," *Proceedings of the Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Optimization,* San Francisco, CA, Sept. 24–26, 1990, pp. 198–203.