Structural Optimization by Multilevel Decomposition

Jaroslaw Sobieszczanski-Sobieski*
NASA Langley Research Center, Hampton, Virginia
and
Benjamin B. James† and Augustine R. Dovi‡
Kentron International Inc., Hampton, Virginia

A method for decomposing an optimization problem into a set of subproblems and a coordination problem that preserves coupling between the subproblems is described. The decomposition is achieved by separating the structural element optimization subproblems from the assembled structural optimization problem. Each element optimization and optimum sensitivity analysis yields the cross-sectional dimensions that minimize a cumulative measure of the element constraint violation as a function of the elemental forces and stiffness. The assembled structural optimization produces the overall mass and stiffness distributions optimized for minimum total mass subject to constraints that include the cumulative measures of the element constraint violations extrapolated linearly with respect to the element forces and stiffnesses. The method is introduced as a special case of a multilevel, multidisciplinary system optimization and its algorithm is fully described for two-level optimization for structures assembled of finite elements of arbitrary type. Numerical results are given as an example of a framework to show that the decomposition method converges and yields results comparable to those obtained without decomposition. It is pointed out that optimization by decomposition should reduce the design time by allowing groups of engineers using different computers to work concurrently on the same large problem.

Nomenclature

A = cross-sectional area
C = cumulative constraint
e = equality constraint, element subscript or superscript
f_i = functional relation (labeled by subscript i)
F = objective function, structural mass in structural application
$\{g\}$ = inequality constraint vector of length m ($\{\ \}$ omitted in equations)
$\{g^e\}$ = elemental inequality constraint vector of length $m(e)$
$\{g^s\}$ = system inequality constraint vector of length $m(s)$
$[K]$ = stiffness matrix ($[\ ]$ omitted in equations)
KS = function defined by Eq. (22)
I = cross-sectional moment of inertia about centroidal y axis shown in Fig. 3 (inset)
l,u = lower and upper bounds, respectively
M = mass or moment
n = length of vector
$n(e)$ = number of y variables in element e
NE = number of elements
NLC = number of loading cases
P = load in Eq. (7); concentrated force in numerical example
$\{Q\}$ = elemental force vector
STOC = subject to constraints
$t(e)$ = number of elemental properties in element e
$\{X\}$ = vector of elemental properties, which are design variables at the system level
$\{y\}$ = vector of detailed design variables of length n at the subsystem level

Superscripts

T = transposed vector or matrix
z = loading case, = 1→NLC
$\{^-\}$ = optimum values

Introduction

THE application of formal optimization techniques to the design of large engineering structures such as aircraft is presently hindered because the number of design variables and constraints is so large that the optimization is both intractable and costly and can easily saturate even the most advanced computers available today. A remedy is to break the problem into several smaller subproblems and a coordination problem, the latter being formulated in a manner which preserves the couplings between the subproblems. In addition to making the problem more tractable, this approach would be compatible with the organization of a typical design project in which diverse engineering groups work concurrently on different parts of the problem. Such an approach would also lend itself to parallel or multiple computer processing, thereby shortening the design cycle time and cost.

Several procedures for breaking large structural optimization problems into subproblems have been proposed in the literature. A typical effort is represented by Ref. 1, which describes a procedure consisting of an analysis of the structure followed by optimization of each substructure while holding invariant the forces acting on it from the contiguous substructures. Since the procedure changes the substructure stiffness properties, analysis of the assembled structure has to be repeated to update the forces acting on the substructures for the next round of substructure optimizations, etc., in an iterative manner. A similar approach was formulated in Ref. 2. Although computationally efficient, these approaches do not subject the overall stiffness distribution to the optimization algorithm. Those algorithms, therefore, cannot be guaranteed to find the minimum structural weight because, in general, a controlled tradeoff of the structural material among the substructures is necessary to find such a minimum. Because of this lack of a controlled tradeoff, the

Received June 2, 1984; revision received Nov. 26, 1984. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Head, Multidisciplinary Analysis and Optimization Branch, Loads and Aeroelasticity Division. Associate Fellow AIAA.
†Senior Structures Engineer. Member AIAA.
‡Unit Supervisor, Advanced Computer Methods Unit, Kentron Technical Center. Member AIAA.

methods of Refs. 1 and 2 are conceptually similar to the fully stressed design method. A method designed to incorporate control of the material distribution among the finite elements of an assembled structure has been offered in Ref. 3 for a two-level optimization.

The optimization schemes cited above are all rather specialized and would not be suitable for application to multidisciplinary optimization of large engineering systems. Recently, Ref. 4 proposed a method for decomposing a large multidisciplinary optimization problem into a number of small subproblems and provided a blueprint for development of a computer implementation for the method. The method decomposes a large problem in the manner shown in Fig. 1. In a general application, each subproblem depicted by a box in Fig. 1 represents a physical subsystem of the total system, e.g., airframe or engines in an aircraft, so that the method is entirely general and admits various engineering disciplines for analysis of the system and subsystems. In structural applications, the decomposition for optimization purposes coincides with multilevel substructuring,[5-7] so that the system represents a complete structure, the subsystems become substructures, and the subsystems of the lowest level, $j=j_{\max}$, correspond to the individual finite elements by which the structure is idealized.

In this application, Ref. 4 is similar to Ref. 3 in that it controls the objective function at the highest level; however, it allows decomposition to several hierarchical levels and couples these levels by using optimum sensitivity derivatives defined in Ref. 8. These derivatives are used in a linear extrapolation to estimate the subsystem response to higher level variable changes, thus eliminating the need for reoptimizing the subsystems for each higher level variable change. The method of Ref. 4 has recently been implemented for two-level optimization and applied to a framework structure as a prelude to proceeding with implementation of a general multilevel optimization procedure. The purpose of this paper is to describe the two-level procedure for the general case of a structure modeled by an assembly of arbitrary-type finite elements, and to illustrate its validity in a numerical application to a simple framework structure which includes a comparison with the results of a conventional one-level optimization. Full detail is omitted in this condensed report; see Ref. 9 for further information.

Two-Level Optimization

This section establishes basic definitions and concepts, and develops a general-pupose algorithm for a two-level structural optimization.

Definitions

For the purposes of structural analysis by a finite element method, one defines n cross-sectional dimensions of the finite element model as entries in the vector y

$$y=\{y_i\}; \qquad i=1\rightarrow n \tag{1}$$

that can also be organized into NE partitions

$$y^T=y^1\ldots y^e\ldots y^{\mathrm{NE}} \tag{2}$$

Each partition of length $n(e)$ corresponds to a finite element of the total of NE finite elements. Stiffness and mass properties of each finite element e are defined by quantities X_t^e collected in a vector X^e, which is a partition e of a vector X for all elements. In further discussion, the quantities X_t^e are referred to as elemental properties. They are computable as functions of y^e,

$$X^e=f_1^e(y^e) \tag{3a}$$

$$X^T=X^1\ldots X^e\ldots X^{\mathrm{NE}} \tag{3b}$$

Examples of X_t^e are: cross-sectional area and moment of inertia of a shaft, and bending stiffness coefficient of an orthotropic plate. One can calculate for element e, its mass and stiffness matrices as

$$M^e=f_2^e(X^e) \tag{4}$$

$$K^e=f_3^e(x^e) \tag{5}$$

referred to the global coordinate system.

In the preceding mass and stiffness expressions, the relevant material properties, e.g., density and Young's modulus are implicit in the functional relations f_2^e and f_3^e. It is possible to make the material choice a design variable, in which case the material properties would be included with the cross-sectional dimensions in vector y^e. However, the variable material case is outside the scope of this report.

Although the element mass and stiffness appear in Eqs. (4) and (5) as functions of X^e, ultimately they are functions of y^e through Eq. (3). Consequently, the elemental properties X^e and the finite element cross-sectional dimensions y^e are hierarchically related as shown by a Venn diagram in Fig. 2. The vector y^e carries information that, for given f_2^e and f_3^e, is sufficient to calculate mass and stiffness for element e, while the vector X^e carries the information needed to quantify mass and stiffness of the entire structure.

Proceeding from an element to the assembled structure, its stiffness matrix K is generated from entries K_{pq}^e of matrix K^e as

$$K=S(K_{pq}^e) \tag{6}$$

where S symbolizes a procedure for direct summation of stiffnesses. Formation and solution of the load-deflection equations for displacements u are

$$Ku=P^z, \qquad z=1\rightarrow \mathrm{NLC} \tag{7}$$

where superscript z refers to a loading case, and yields displacements u and elemental forces $Q_r^{e,z}$ for element e. Expressed in an element coordinate system,

$$Q^{e,z}=H^eK^eu^{e,z} \tag{8}$$

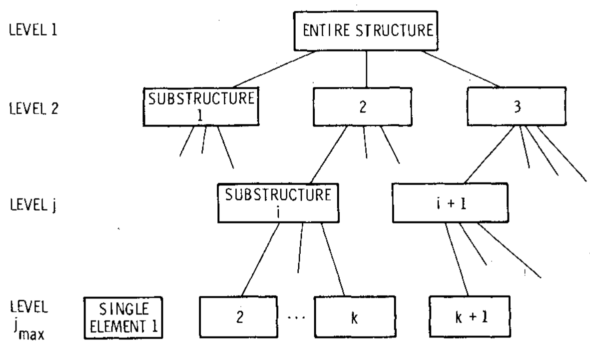

Fig. 1 Multilevel substructuring.

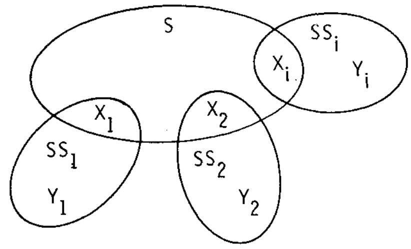

Fig. 2 Venn diagram for a two-level system hierarchy of design variables.

where each vector $Q^{e,z}$ contains $r(e)$ forces $Q_r^{e,z}$. The forces in vector $Q^{e,z}$ are statically independent and related to the full set of elemental forces by the matrix H^e that represents the element equilibrium.

One-Level Optimization

A conventional one-level optimization for minimum mass F can be based on the quantities defined by Eqs. (1-7). Namely, with y as the vector of design variables, one has

$$\min_{\{y\}} F(y) \tag{9a}$$

subject to constraints (STOC)

$$g_j(y) \leq 0; \qquad j = 1 \rightarrow m \tag{9b}$$

$$y_l \leq y \leq y_u \tag{9c}$$

where constraints g are imposed on the static behavior variables, such as stresses and displacements, and y_l, y_u are side constraints.

Two-Level Optimization Procedure

Under a two-level optimization approach, the problem defined by Eqs. (9) is decomposed into a single problem at the assembled structure level and NE subproblems (one for each finite element) at the lower level. The two levels are referred to as system and subsystem levels, respectively. With respect to a general, multilevel substructuring scheme shown in Fig. 1, the two-level case corresponds to the upper two levels of the figure, with the "substructures" of the second level acquiring the physical meaning of individual finite elements.

Conversion to a two-level optimization scheme begins with partitioning the vector of constraints g into

$$g^T = \lfloor g^s, g^1, \ldots g^e, \ldots g^{\mathrm{NE}} \rfloor \tag{10}$$

where g^s contains the constraints on the system behavior, and the remaining constraints are local to each element. Examples are a nodal point displacement limit for the former and an element stress allowable for the latter. Tracing the functional relations for g^s through Eqs. (7) and (3b) one obtains

$$g^s = f_4(x) \tag{11}$$

Similarly, for g^e, the trace through Eqs. (8), (6), (5), and (3a) leads to

$$g^e = f_5^e(y^e, X^e, Q^e) \tag{12}$$

Furthermore, the structural mass in Eq. (9a) becomes

$$F = f_6(X) \tag{13}$$

when the element masses expressed by Eq. (4) are summed.

Subsystem (Element) Level

For conversion to a two-level optimization, it is necessary for each finite element e that the number $n(e)$ of its cross-sectional variables be no less than the number $t(e)$ of its elemental properties, X_t^e.

$$t(e) < n(e), \qquad e = 1 \rightarrow \mathrm{NE} \tag{14}$$

The above equation may be satisfied in both its equality and inequality parts, or in its equality part only, dependent on the type of structural element. If Eq. (14) holds in its inequality part, then it is possible to carry out an isolated local operation of changing values of the entries in g^e by manipulating the design variables in y^e in such a way that X^e and, consequently, $Q^{e,z}$ remain constant. In other words, if the inequality in Eq. (14) is true, there is design freedom to proportion the element in a new way, improved in some sense, without affecting the assembled structure solution. Translated into a formally stated optimization problem; for element e,

$$\min_{\{y^e\}} C^e(g^e) \tag{15a}$$

$$X^e - f_1^e(y^e) = 0 \tag{15b}$$

$$y_l^e \leq y^e \leq y_u^e \tag{15c}$$

Constraints of the problem are the side constraints on y as in Eq. (9c) and equality constraints to enforce invariance of X^e for the duration of the solution process of Eqs. (15). Regardless of the technique used to satisfy the equality constraints in Eq. (15b), their presence has the effect of reducing the number of free variables in y^e by $t(e)$, hence, the condition of Eq. (14).

The problem's objective function, C^e, is a single number that measures the degree of constraint violation for all constraints that make up vector g^e. The quantity C^e is known in the literature[9-11] as a cumulative constraint and can be formulated in a number of ways. For this paper's purpose, $C^e(g^e)$ should be such that

$$\begin{aligned} C^e &= >0, \quad \text{if } g_j^e > 0; \quad j = 1 \ldots m(e); \text{ (at least one violated)} \\ &= \leq 0, \quad \text{if } g_j^e \leq 0; \quad j = 1 \rightarrow m(e); \text{ (all satisfied)} \end{aligned} \tag{16}$$

and must have continuous derivatives. A specific formulation for C^e will be discussed later [Eqs. (22) and (23)].

The choice of C^e for the objective function in the element optimization subproblems, Eqs. (15), is consistent with Eq. (4) which, for constant X^e and for Eq. (14) holding, renders F unaffected by the changes of y^e. This means that, as similarly proposed in Ref. 3, there is no control of the objective function at the lower level of optimization; the only objective of optimization at that level is to achieve the best possible satisfaction of constraints consistent with the element forces $Q^{e,z}$ and the elemental properties X^e.

System (Structure) Level

The objective function and the remaining constraints, g^s, are controlled at the assembled structure level. The single optimization problem to be solved at that level is then

$$\min_{\{X\}} F(X) \tag{17a}$$

$$\text{STOC } g_j^s \leq 0; \qquad j = 1 \rightarrow m(s) \tag{17b}$$

$$C^e \leq 0; \qquad e = 1 \rightarrow \mathrm{NE} \tag{17c}$$

$$y_l^e \leq y^e \leq y_u^e; \qquad e = 1 \rightarrow \mathrm{NE} \tag{17d}$$

$$X_l \leq X \leq X_u \tag{17e}$$

where the objective function

$$F = \Sigma_e M^e \tag{18}$$

depends on X through Eq. (4), and the entries of the vector X are the system-level design variables. Presence of the constraints on C^e in Eq. (17c) and y^e in Eq. (17d) assures that when a solution to the system-level optimization problem is found, satisfaction of all of the local constraints will be a part of that solution.

Coupling between the Levels

The optimization problems in the form given by Eqs. (15) at the subsystem level are coupled to the system-level problem. It is a two-way coupling: As input, each subsystem problem receives the system-level variables X^e and the system-level analysis results $Q^{e,z}$ [Eqs. (12), (15), and (16)] and returns its optimal C^e and y^e [Eqs. (17c) and (17d)]. However, in practical implementation, the constraints on C^e and y^e cannot remain in the form of Eqs. (17c) and (17d) because their evaluation for each new X would require a new solution to the subsystem problem in Eqs. (15)—a reoptimization of the elements affected by new X. Computationally, that would be prohibitively costly in large problems.

There is a way to bypass these costly subsystem reoptimizations in the system-level optimization. It is available in the concept of the sensitivity of the optimum to problem parameters and an associated algorithm for computation of the derivatives to quantify that sensitivity is proposed in Ref. 8. Applying that concept to the optimization represented by Eqs. (15), one recognizes immediately that because of Eqs. (12) and (13) the optimization constant parameters are X^e_t and $Q^{e,z}_r$. Consequently, the optimum solution, $\bar{C}^e$ and $\bar{y}^e$, of Eqs. (15) is a function of these parameters and has derivatives $d\bar{C}^e/dX^e_t$, $d\bar{C}^e/dQ^{e,z}_r$, $d\bar{y}^e/dX^e_t$, and $d\bar{y}^e/dQ^{e,z}_r$, termed optimum sensitivity derivatives.

Table 1 Interlevel flow of information

System level (assembled structure)	Subsystem level (individual elements)
Elemental properties and forces: X^e; $Q^{e,z}$	Elemental optimum solutions and their derivatives: $\bar{C}^3$; $\bar{y}^e$; $\frac{\partial C^e}{\partial \bar{X}}$, $\frac{\partial \bar{C}^e}{\partial Q^{e,z}_r}$; $\frac{\partial \bar{y}^e}{\partial X}$, $\frac{\partial y^e}{\partial Q^{e,z}_r}$

Table 2 Correspondence of the quantities in the framework examples to the generic quantities

Generic	Framework
$y=\{\ldots\{y^e_i\}\ldots\}$	$y=\{\ldots\{b_1,t_1,b_2,t_2,h,t_3\}\ldots\}$
$X^e=f^e(y^e)$	$\begin{Bmatrix} A \\ I_y \end{Bmatrix}^e = f^e_1\{b_1,t_1,b_2,t_2,h,t_3\}$, $e=1,2,3$
$Q^{e,z}$	$\{N,M,T\}^{e,z}$
g^s	Constraints on the loaded node horizontal translation and rotation due to P and M
g^e	Beam stress and local buckling constraints
F	Beam mass, $M=f(A_1,A_2,A_3)$
NLC	2
NE	3
n	18
$n(e)$	6
$t(e)$	2
NE$\times t(e)$	6
$q(e)$	3
$r(e)$	$3\times$NLC$=6$
z	2

The algorithm described in Ref. 8 uses first and second derivatives of behavior with respect to the design variables. The computational cost of the second derivatives is eliminated in a version of the algorithm given in Ref. 12, and it may be reduced by means of the techniques proposed in Refs. 13 and 14.

The optimum sensitivity derivatives can be used in a Taylor series to convert the nonlinear dependence of $\bar{C}^e$ and $\bar{y}^e$ on X in Eqs. (17c) and (17d) into the linear extrapolation approximations

$$C^e=\bar{C}^e_a \simeq \bar{C}^e_0+\sum_t \frac{\partial \bar{C}^e}{\partial X^e_t}(X^e_t-X^e_{t0})$$
$$+\sum_z\sum_e\sum_t\sum_r \frac{\partial \bar{C}^e}{\partial Q^{e,z}_r}\frac{\partial Q^{e,z}_r}{\partial X^e_t}(X^e_t-X^e_{t0}) \tag{19a}$$

$$y^e=\bar{y}^e_a \simeq \bar{y}^e_0+\sum_t \frac{\partial \bar{y}^e}{\partial X^e_t}(X^e_t-X^e_{t0})$$
$$+\sum_z\sum_e\sum_t\sum_r \frac{\partial \bar{C}^e}{\partial Q^{e,z}}\frac{\partial Q^{e,z}_r}{\partial X^e_t}(X^e_t-X^e_t) \tag{19b}$$

where subscripts a and 0 denote, respectively, the approximate and reference values. The chain differentiation reflects the dependence of $Q^{e,z}$ on X that occurs in redundant structures in which, generally speaking, each $Q^{e,z}_r$ depends on each X^e_t, therefore, the summation in the chain differentiation spans the entire vector X. The derivatives $dQ^{e,z}/dX^e_t$ are behavior sensitivity derivatives routinely available through analytical techniques applied to Eqs. (7) and (8).[15-17]

The relations established in Eqs. (19) will be referred to as a linear representation of the subsystem.

System (Structure) Level Problem with Embedded Coupling

Substituting Eqs. (19) into Eqs. (17), the system-level optimization problem becomes

$$\min_{\{X\}} F(X) \tag{20a}$$

$$\text{STOC } g^s_j \le 0, \qquad j=1\rightarrow m(s) \tag{20b}$$

$$\bar{C}^e_a=\bar{C}^e_0+\sum_t \frac{\partial \bar{C}^e}{\partial X^e_t}(X^e_t+X^e_{t0})$$
$$+\sum_z\sum_e\sum_t\sum_r \frac{\partial C^e}{\partial Q^{e,z}_r}\frac{\partial Q^{e,z}_r}{\partial X^e_t}(X^e_t-X^e_{t0})\le 0; \quad e=1\rightarrow \text{NE} \tag{20c}$$

$$y^e_l<\bar{y}^e_a=\bar{y}^e_0+\sum_t \frac{\partial \bar{y}^e}{\partial X^e_t}(X^e_t+X^e_{t0})$$
$$+\sum_z\sum_e\sum_t\sum_r \frac{\partial \bar{y}^e}{\partial Q^{e,z}_r}\frac{\partial Q^{e,z}_r}{\partial X^e_t}(X^e_t-X^e_{t0})<y^e_u; \quad e=1\rightarrow \text{NE} \tag{20d}$$

$$X_l \le X \le X_u \tag{20e}$$

$$f_7(y_l,y_u)\le X \le f_8(y_l,y_u) \tag{20f}$$

$$X^M_l \le X \le X^M_u \tag{20g}$$

Two new groups of constraints appear in Eqs. (20f) and (20g). The constraints of Eq. (20f) are added to keep the optimization algorithm from generating such combinations of X^e_t values that cannot be physically implemented at the finite element level (for example, see Ref. 5, Eqs. C1-C5). The constraints in Eq. (20g) introduce additional bounds on X as move limits X^M_l and X^M_u needed to control the linearization errors.

Two-Level Procedure Algorithm and Salient Features

The linearization of Eqs. (17c) and (17d) resembles the local linearization technique based on the behavior sensitivity derivatives known to be very effective in nonlinear mathematical programming.[17-19] Incidentally, that technique could also be used in Eq. (20b) to linearize g^s; whether to exercise this option is a problem-dependent decision. As is the case with any linearization technique applied to solve an intrinsically nonlinear problem, an iterative procedure has to be constructed to allow recovery from the linearization errors even though the errors are controlled by appropriate move limits [Eq. (20g)]. The iterative procedure algorithm consists of the following steps:

1) Initialize y.
2) Compute X [Eq. (3)].
3) Analyze assembled structure, obtain $Q^{e,z}$, g^s, and their derivatives with respect to X_i^e [Eqs. (7), (8), and appropriate gradient calculation technique].
4) Solve subsystem optimization, Eqs. (15), for each element.
5) Calculate optimum sensitivity derivatives for the optima found in step 4.
6) Solve the system optimization, Eqs. (20).
7) Update X and repeat from step 3 until a converged solution is obtained.

In this procedure, optimizations performed in steps 4 and 6 are iterative within themselves and nested in the overall iteration spanning steps 3 and 7. In further discussion, the latter will be referred to as a cycle, while the term "iteration" will be used in conjunction with steps 4 and 6.

Since the calculations performed in steps 4 and 5 of the procedure are executed separately for each finite element, they can be carried out concurrently using distributed computing technology.

Information flow between the two levels of the procedure is restated in Table 1. Readers familiar with system analysis as formulated in the discipline of operations research[20] will recognize the information returned to the system level as a particular means to solve the so-called system coordination problem.

In the two-level procedure, the system objective function (e.g., structural mass) is entirely controlled at the system level by variables X which can be regarded as generalized design variables that determine the structure mass and stiffness distribution. The system objective function is not directly included in the subsystem optimizations whose only purpose is to achieve the best possible satisfaction of the local constraints consistent with the parameters imposed from the system level. The procedure is entirely open to accommodate the designer's judgment as to the type and number of design variables at each level. The familiar device of variable linking[21] can be used freely at both levels to keep the number of design variables as small as possible and, for the same purpose, one may refrain from including all of the available elemental properties in the set of design variables X.

Equivalence to a One-Level Optimization

According to Eqs. (20), (16), and (15), the two-level procedure, when converged in the Kuhn-Tucker[15] sense, produces a feasible design, just as a single-level optimization [Eqs. (9)] does. Moreover, if the Kuhn-Tucker conditions are satisfied in the subsystem- and system-level problems in the two-level procedure, one can infer that they are also satisfied in the y space in the one-level optimization [Eqs. (9)]. Discussion of the inference is given in Appendix B of Ref. 9. In these respects then, the two procedures are equivalent. However, it does not follow that both will lead to the same design point in nonconvex problems having multiple local minima. In such problems that include many practical applications, the solution depends on the computational path through the design space, and the path taken, in turn, depends on the algorithm. Since the two procedures are algorithmically different, a difference in their results in nonconvex applications should be expected. There is no basis to rate one procedure inherently superior to the other in its effectiveness and efficiency in such applications.

This aspect of the procedure performance, as well as its convergence characteristics and overall computational behavior, can be assessed only by numerical experiments. Such experiments are described in the next two sections.

Framework Structure as a Test Case

A portal framework shown in Fig. 3 is an example of a hierarchical system that can be optimized for minimum mass under static loads subject to strength and displacement constraints using the linear decomposition approach. The decomposition is two-level and results directly from the fact that one can use engineering beam theory to analyze the framework for internal forces (the end forces on each beam) and displacements, assuming that A and I for each beam are given but without knowing the detailed cross-sectional dimensions ($b_1, t_1, \ldots$). These dimensions can be optimized separately for each beam as long as the end forces in each beam are known and assumed fixed which, in turn, requires holding A and I of each beam constant. The correspondence of the basic elements of a two-level decomposition approach to the framework example is given in Table 2.

The Case Definition and Its One-Level Formulation

The framework is composed of three I-beams made of the same material and having cross-sectional dimensions as shown in the inset in Fig. 3. Structural optimization is to be carried out for a minimum mass subject to constraints on static response induced by two loading cases: a concentrated force and a concentrated moment. The constraints are imposed on the framework displacements—horizontal translation and rotation at the loaded point of the framework, and on the stresses in each beam. The extreme normal stresses caused by bending moment and axial force, and the extreme shear stress due to the transverse force, are constrained at both ends of each beam to stay below the material allowable stress and the critical stresses of local buckling. The latter account for buckling of flange and web but ignore the column buckling. The framework is assumed to be supported against displacements out of the plane of Fig. 3 to eliminate the need for constraints on the framework overall instability and the lateral-torsional buckling of its beams. Constraints include the bounds on the design variables. Detailed formulations of all of the constraints are provided in Appendix C of Ref. 9.

Analysis of the framework for displacements and internal forces employs a standard, displacement-based, finite element method representing each beam by a single beam element. Beam stresses are calculated according to the engineering beam-bending theory. The critical buckling stresses are

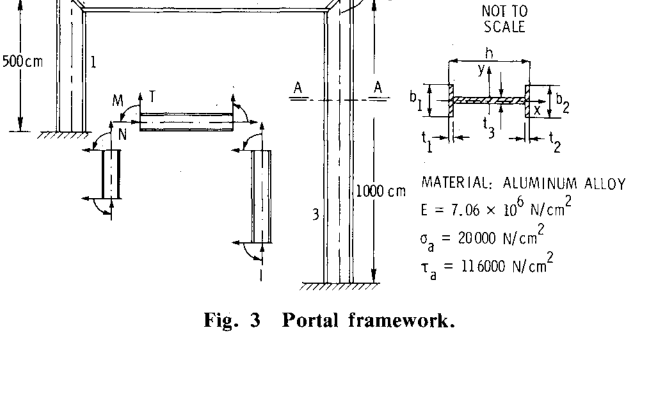

Fig. 3 Portal framework.

computed for each part of the beam, e.g., a flange, as for an isolated plate with appropriate boundary conditions using routine techniques.[22]

Taking the detailed cross-sectional dimensions of each beam as the design variables of the problem, the vector y in Eq. (2) has 3 partitions, each containing 6 variables for one beam, for a total of 18 design variables,

$$y^T = \lfloor \lfloor b_1, t_1, b_2 \ldots \rfloor^1, \quad \lfloor b_1, t_1, b_2 \ldots \rfloor^2, \quad \lfloor b_1, t_1, b_2 \ldots \rfloor^3 \rfloor$$

Optimization for minimum mass can be replaced with optimization for minimum material volume because of the material homogeneity. Denoting the beam length by l_i, the objective function becomes a function of cross-sectional area A,

$$F \equiv M = \sum_i A_i l_i \tag{21}$$

The problem can be solved as a one-level optimization with $n = 18$ design variables in a conventional formulation such as given in Eqs. (9).

Two-Level Formulation

Under the two-level approach, the framework (system) is considered decomposed into three beams (subsystems, finite elements) under the action of the beam-end forces shown in Fig. 3 (inset). If the decomposed framework were superimposed on the general multilevel decomposition scheme shown in Fig. 1, the assembled framework would fall in the "entire structure" box at level 1 and each beam would coincide with a substructure at level 2, although in this case "substructure" simplifies to a single finite element. For the purposes of the framework analysis, each beam's stiffness and mass properties are determined by two elemental properties: cross-sectional area A and moment of inertia I. These quantities become the system-level design variables in vector X, Eq. (3b), as indicated in Table 2. The framework displacement constraints are in the g^s category, while the beam stress constraints are included as g^e in Eq. (10). Since the inequality in Eq. (14) holds for each beam the original problem can be solved decomposed into three subsystem problems of six design variables y^e each and a system problem of six design variables X, according to Eqs. (15) and (20), respectively. The cumulative constraint function defined generally in Eq. (16) was set as a function proposed in Ref. 23 and applied in Ref. 11 to approximate the maximum constraint.

$$\mathrm{KS}(g_j) = \frac{1}{\rho} \ln \left[\sum_{j=1}^{m} \exp(\rho g_j) \right] \tag{22}$$

and has the property of following the maximum constraint:

$$\max(g_j) \leq \mathrm{KS} \leq \max(g_j) + (1/\rho) \ln(m) \tag{23}$$

with a tolerance that depends on the constant ρ supplied by the user. The KS function has continuous derivatives and performs as an extended penalty function because it is defined throughout the infeasible as well as the feasible domains.

Numerical Results

Two-level structural optimization by linear decomposition is demonstrated for the framework example. The seven-step procedure defined in the foregoing is implemented in a Fortran main program that calls a finite element analysis subroutine and an optimization subroutine (program CONMIN[24]) that employs a usable-feasible direction technique. Results obtained on a PRIME 750 computer include benchmark data for a conventional one-level optimization and the two-level optimization data. The detailed description of the implemented procedure and formulation of the constraint functions, including equality constraints on A_i and I_i, used in all of the numerical tests are given in Appendix C of Ref. 9.

Results for a Conventional One-Level Approach

Several variants of one-level optimization and several different starting points in both the feasible and infeasible domains were used to obtain the benchmark results. In variant 1, which was chosen to be the reference technique, all constraints were kept separate, while in variant 2 the constraints, except side constraints, were collected in a cumulative constraint. A piecewise-linear procedure combined with the cumulative constraint [in the form given in Eq. (24)] was carried out in variant 3 using move limits of 15%. The purpose of including variants 2 and 3 in the benchmark testing was to determine to what extent the optimization results were influenced by the use of a cumulative constraint and a piecewise-linear procedure, which are both embedded in the two-level optimization. It turned out that the three variants and different starting points generate designs having masses that fell to within 5% of the variant 1 result. Repetitiveness of the two-level procedure measured by the difference between the minimum objective functions obtained when starting from different initial designs was very good—Table 3 shows the difference well below 0.5%. However, in at least one test case (not included in Table 3) there was as much as a 300% difference in some design variables. These differences are an indication that this particular example problem is nonconvex and has a "shallow" optimum.

The objective function and design variable values are displayed in Table 3 for variant 1 of the one-level optimization for initially infeasible and feasible designs to provide a benchmark for the two-level optimization. (The unusually large beam depth-to-width ratio is a predictable result of excluding the constraint of torsional-bending buckling and has no bearing on the numerical verification of the method.)

Results for the Two-Level Procedure

Numerical studies with the two-level optimization were carried out to assess the method's capability to produce optimum designs similar to the benchmark results, and to evaluate its convergence. The move limits of $\pm 15\%$ on A and $\pm 30\%$ on I in the set of variables X were found satisfactory and were used uniformly in the study.

Similarity to the Benchmark Results

The results show the method's ability to generate designs comparable to the benchmark design when starting from the same point, either feasible or infeasible. As shown in Table 3, the minimum objective function values obtained by means of the two-level optimization started from infeasible and feasible points exceeded in both cases the benchmark values corresponding to the one-level optimization by only 1.6 and 1.9%, respectively. This small but systematic discrepancy may be attributed to the particular cumulative constraint formulation given in Eq. (22) which, as shown in Eq. (23), systematically overconstrains the problem, and in a structural application causes a weight penalty.

More significant differences were recorded among the individual design variables at both local and system levels in the optimal designs corresponding to different initial points. The two-level optimization does not seem to magnify these differences. In fact, a reduction was observed; see Table 3. Specifically, comparison of the optimal design variables obtained by means of the one-level procedure started from the infeasible and feasible points reveals discrepancies up to 50% (beam 1, variable b_2), while the largest such discrepancy for the two-level procedure does not exceed 4% (beam 2, variable b_1).

Convergence History

The convergence was found to be similar in character to that of the conventional one-level method as illustrated by comparison of a sample of history graphs in Fig. 4. The graphs span horizontal intervals of different lengths because the abscissa scales are different: "iterations" and "cycles" for the one- and two-level optimizations, respectively. Each graph, except graph 5, corresponds to an infeasible initial design, and all are normalized by initial values. For the objective function (mass), graph 1 for two-level optimization compares with graph 2 for one-level optimization. For the constraints, graph 3 for a cumulative constraint of beam 1 compares with graph 4 for an individual constraint (one of the flange local buckling constraints) in beam 1. Graph 5, that has no counterpart, shows that, when started from an initially feasible design, the two-level procedure exhibits convergence of the objective function that is smoother and faster than for an infeasible design starting point.

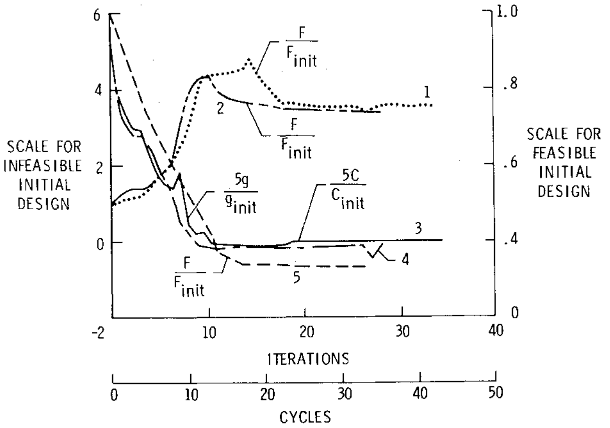

Fig. 4 Examples of history plots for the objective function, an individual constraint, and a cumulative constraint for one- and two-level optimizations starting from feasible and infeasible designs. Factor 5 in graphs 3 and 4 is for scale uniformity.

Accuracy of Linear Extrapolation

The multilevel optimization approach is predicted on the accuracy of the linear extrapolations based on the optimum sensitivity derivatives. Therefore, it is interesting to see how the cumulative constraint predicted by the linear extrapolation at the end of one cycle compares with the result of full analysis carried out at the beginning of the next cycle. Such a comparison is displayed in Fig. 5 and shows that the prediction error eventually vanishes after a number of cycles, thus permitting the procedure to converge. The graphs show that before the convergence is reached the linear extrapolation consistently underpredicts the cumulative constraint value when proceeding from an infeasible design starting point and overpredicts it when the start is made from a feasible design point.

Characteristically, the relative error is larger when the optimization is started from an infeasible design, apparently because the procedure then goes through a number of changes in the membership of the active constraint set that comprises constraints defined in Eq. (15c) for each beam. Consistent with observations reported in Refs. 8 and 25,

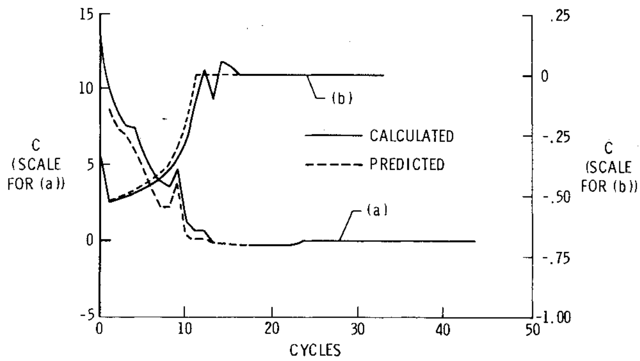

Fig. 5 Comparison of the cumulative constraint of beam 1 as predicted by linear extrapolation at the system level and calculated in full analysis: a) infeasible design start, b) feasible design start.

Table 3 Comparison of optimization results

	Initial values (infeasible design)	Final results		Initial values (feasible design)	Final results	
		One-level optimization	Two-level optimization		One-level optimization	Two-level optimization
Objective function, cm^3	26,469	90,682	92,090	275,000	90,592	92,330
Beam 1						
b_1	11.0	10.0	10.3	30.0	13.0	10.3
t_1	0.275	0.491	0.571	1.0	0.450	0.569
h	22.0	78.1	73.9	50.0	74.9	74.0
t_3	0.275	0.517	0.518	1.0	0.497	0.519
b_2	5.5	8.08	5.08	30.0	12.1	5.13
t_2	0.275	0.511	1.18	1.0	0.487	1.16
Beam 2						
b_1	11.0	10.3	10.3	30.0	11.4	10.7
t_1	0.275	0.439	0.476	1.0	0.404	0.451
h	22.0	95.7	89.4	50.0	89.9	90.1
t_3	0.275	0.421	0.414	1.0	0.397	0.417
b_2	5.5	8.46	5.14	30.0	10.7	5.12
t_2	0.275	0.539	0.984	1.0	0.435	0.960
Beam 3						
b_1	5.5	5.0	5.03	30.0	7.50	4.98
t_1	0.275	0.267	0.253	1.0	0.268	0.253
h	22.0	47.8	59.1	50.0	61.9	59.0
t_3	0.275	0.25	0.251	1.0	0.25	0.25
b_2	11.0	10.0	10.0	30.0	10.0	10.0
t_2	0.275	0.332	0.400	1.0	0.369	0.393

NOTES: See Fig. 3 for dimension definitions. All beam dimensions are in centimeters.

these changes tend to degrade the accuracy of the optimum sensitivity derivatives as the behavior predictors. The larger prediction errors apparently cause the history plots to be somewhat more jagged in all cases for which the optimization is started from an infeasible design rather than from a feasible one (see Fig. 4). They also slow down the convergence so that a larger number of cycles is required when starting from an infeasible design. However, the procedure exhibits a good ability to recover from occasional large prediction errors, e.g., cycle 8 in Fig. 5 (infeasible design start) and cycles 11 and 13 in Fig. 5 (feasible design start), and gets back on track with remarkable robustness.

Computational Cost

Since the purpose of the reported work was a demonstration of a concept, no attempt was made to refine either the reference, one-level procedure, or the two-level procedure for maximum computational efficiency, especially since the framework example is much too small to demonstrate efficiency of any optimization or analysis method. Nevertheless, one may observe that the two-level optimization converges in a number of cycles about equal to the number of iterations in the one-level optimization, with the numerical workload in a cycle being less than in an iteration. While the precise workload difference depends on the algorithmic and implementation details, the major difference stems from having to calculate a number of gradient vectors equal to the number of design variables which is smaller than in the one-level optimization at the system level of the two-level optimization.

Concluding Remarks

A method for decomposing an optimization problem into a set of subproblems and a single coordination problem which preserves the coupling between the subproblems has been described. The resulting procedure is iterative and calls for repetitive analysis of the assembled structure, optimization of the individual components as subproblems, followed by optimization of the assembled structure in which the component optimum solutions are extrapolated linearly using their optimum sensitivity derivatives with respect to the system-level design variables and internal forces. The subproblems are organized hierarchically into two levels. The variables at the lowest level (the subsystem level) are physical cross-sectional dimensions; the variables at the highest level (the system level) are quantities that govern the stiffness and mass distribution among the finite elements of the structure. The overall objective function (such as structural weight) is controlled at the system level. Optimization at that level influences the level below by means of changing the mass and stiffness distribution, and the associated distribution of internal forces.

The method is demonstrated using a portal framework as an example of a two-level structure in which the system-level variables are the cross-sectional areas and moments of inertia of the beams, and the subsystem-level variables are the beam detailed cross-sectional dimensions. Verification of the method by comparison with the results obtained by a conventional one-level optimization shows the validity and effectiveness of the proposed approach.

Satisfactory testing of the two-level approach is a stepping stone for implementation of a multilevel structural optimization procedure. That implementation is seen as a stage in the development of a multilevel optimization for multidisciplinary engineering systems whose goal is to allow groups of engineers using distributed computing technology to work concurrently on various parts of the problem, thereby reducing the real time of the system design.

References

[1]Kirsch, U., Reiss, M., and Shamir, U., "Optimum Design by Partitioning into Substructures," *Journal of the Structures Division, ASCE*, Vol. 98, Jan. 1972, p. 249.

[2]Sobieszczanski-Sobieski, J., "Sizing of Complex Structure by the Integration of Several Different Optimal Design Algorithms," AGARD Lecture Series on Structural Optimization, AGARD LS-70, Sept. 1974, pp. 4-1—4-19.

[3]Schmit, L. A. and Ramanathan, R. K., "Multi-level Approach to Minimum Weight Design Including Buckling Constraints," *AIAA Journal*, Vol. 16, Feb. 1973, pp. 97-104.

[4]Sobieszczanski-Sobieski, J., "A Linear Decomposition Method for Large Optimization Problems—Blueprint for Development," NASA TM 83248, Feb. 1982.

[5]Noor, A. K., Kamel, H. A., and Fulton, R. E., "Substructuring Techniques—Status and Projections," *Computers & Structures*, Vol. 8, May 1978, pp. 621-632.

[6]Przemieniecki, J. S., *Theory of Matrix Structural Analysis*, McGraw-Hill Book Co., New York, 1968, Chap. 9.

[7]Aaraldsen, P. O., "The Application of the Superelement Method in Analysis and Design of Ship Structures and Machinery Components," Paper presented at the National Symposium on Computerized Structural Analysis and Design, George Washington University, Washington, D.C., March 1972.

[8]Sobieszczanski-Sobieski, J., Barthelemy, J.-F., and Riley, K. M., "Sensitivity of Optimum Solutions to Problem Parameters," *AIAA Journal*, Vol. 20, Sept. 1982, p. 1291.

[9]Sobieszczanski-Sobieski, J., James, B., and Dovi, A., "Structural Optimization by Multilevel Decomposition," *Proceedings of the AIAA/ASME/ASCE/AHS 24th Structures, Structural Dynamics and Materials Conference*, May 1983.

[10]Sobieszczanski-Sobieski, J., "From a "Black Box" to a Programming System: Remarks on Implementation and Application of Optimization Methods," *Proceedings of a NATO Advanced Study Institute*, Session on Structural Optimization, University of Liege, Sart-Tilman, Belgium, Aug. 1980, pp. 405-450.

[11]Hajela, P., "Further Developments in the Controlled Growth Approach for Optimal Structural Synthesis," ASME Paper 82-DET-62, Sept. 1982.

[12]Barthelemy, J.-F. and Sobieszczanski-Sobieski, J., "On Optimum Sensitivity Derivatives of Objective Function in Nonlinear Programming," *AIAA Journal*, Vol. 21, June 1983, pp. 913-915.

[13]Haug, E. J., "Second-Order Design Sensitivity Analysis of Structural Systems," *AIAA Journal*, Vol. 19, Aug. 1981, p. 1087.

[14]Haftka, R. T., "Second-Order Sensitivity Derivatives in Structural Analysis," *AIAA Journal*, Vol. 20, Dec. 1983, pp. 1765-1766.

[15]Fox, R. L., *Optimization Methods for Engineering Design*, Addison-Wesley Pub. Co., 1971.

[16]Storaasli, O. O. and Sobieszczanski-Sobieski, J., "On the Accuracy of the Taylor Approximation for Structure Resizing," *AIAA Journal*, Vol. 12, Feb. 1974, pp. 231-233.

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

[18]Anderson, M. S. and Stroud, W. J., "A General Panel Sizing Computer Code and Its Application to Composite Structural Panels," *Proceedings of the AIAA/ASME 19th Structures, Structural Dynamics and Materials Conference*, April 1978.

[19]Schmit, L. A., "Structural Synthesis—Its Genesis and Development," *AIAA Journal*, Vol. 19, Oct. 1981, pp. 1249-1263.

[20]Lasdon, L. S., *Optimization Theory for Large Systems*, The MacMillian Co., New York, 1970.

[21]Schmit, L. A., "Structural Synthesis from Abstract Concept to Practical Tool," *Proceedings of the AIAA/ASME 18th Structures, Structural Dynamics and Materials Conference*, Vol. A, San Diego, Calif., March 1977, paper no. 77-359.

[22]Timoshenko, S. and Gere, J. M., *Theory of Elastic Stability*, McGraw-Hill Book Co., New York, 1961.

[23]Kresselmeier, G. and Steinhauser, G., "Systematic Control Design by Optimizing a Vector Performance Index," *Proceedings of IEAC Symposium on Computer Aided Design of Control Systems*, Zurich, 1971, pp. 113-117.

[24]Vanderplaats, G. N., "CONMIN—A Fortran Program for Constrained Function Minimization: User's Manual," NASA TM X-62282, Aug. 1973.

[25]Barthelemy, J. F. and Sobieszczanski-Sobieski, J., "Extrapolation of Optimization Design Based on Sensitivity Derivatives," *AIAA Journal*, Vol. 21, May 1983, pp. 797-799.