Efficient Single-Level Solution of Hierarchical Problems in Structural Optimization

Rajiv R. Thareja*
Lockheed Engineering and Sciences Company, Hampton, Virginia
and
Raphael T. Haftka†
Virginia Polytechnic Institute and State University, Blacksburg, Virginia

Engineering design is hierarchical in nature, and the hierarchy can be used to improve the efficiency of design optimization. Multilevel optimization techniques incorporate the hierarchy at the formulation stage and result in the coordinated optimization of a number of subsystems. However, the use of these techniques is associated with numerical difficulties such as discontinuous derivatives. Single-level hierarchical solution techniques take advantage of the hierarchical nature at the solution stage. The present paper demonstrates the use of the latter approach in structural optimization for a penalty-function optimization method based on Newton's method. A single-level decomposition technique is developed that reduces the computational costs and memory requirements without incurring the disadvantages of multilevel optimization. The technique is demonstrated for a portal frame example. Substantial savings of about 75% are obtained for a case with 500 design variables and 2408 constraints with indication of higher potential savings for larger problems. For truly large systems, this decomposition technique provides the necessary reduction of computational effort to make the optimization process viable.

Introduction

THE design of complex structures typically proceeds at several levels of detail. As an example, consider the current practice for designing an aircraft wing. The overall design of the wing proceeds on the basis of a beam analysis or a gross finite-element model. From this analysis, the loads acting on the major components of the wing such as individual panels are obtained. Each panel then is designed based on the assumption that these loads are fixed. The panel design then is followed by detail design to prevent stress concentration around cutouts and discontinuities. Again, it is assumed that loads obtained from panel design do not change when the details of cutout are changed. This approach neglects the effects of the redesign of one part of the structure on other parts and does not exploit the interaction between local structures. As a result, a truly optimized structure cannot be achieved since there is no mechanism for redistribution of forces or material between local structures.

One way to exploit the interaction between local structures is to carry out a local-global design optimization simultaneously. Such an optimization is often prohibitive with respect to computational effort and memory requirements. It is more economical to take advantage of the fact that the local structures are only weakly coupled and to formulate the design problem as a hierarchical problem in which the lower-level design problems are coupled only through interaction with higher levels. Such a hierarchical formulation typically requires the introduction of global variables and the addition of equality constraints to assure consistency between lower-level (local) and upper-level (global) variables.

Two types of techniques are used to capitalize on the hierarchical structure. Multilevel design techniques break up the problem into several smaller problems at the formulation stage and use multilevel techniques to solve the resulting hierarchical optimization problem.[1-7] One shortcoming of this technique is that lower-level subsystems need to be optimized several times. Also, the derivatives of the lower-level optima may be discontinuous functions of the higher-level design variables,[8] and the equality constraints used in the hierarchical formulation may make the multilevel optimization problem more ill conditioned than the single-level problem.[9]

For hierarchical systems, the alternative to multilevel techniques is single-level formulation accompanied by a decomposition technique for the solution of the single-level problem. For linear problems, linear programming (LP) decomposition methods are used to convert a large problem into a sequence of smaller LP problems. The Dantzig-Wolfe decomposition method[10] was the first to take advantage in the solution process of a block-diagonal structure that exists in most large linear problems. Rosen[11] suggested a modified approach that had better convergence properties. His approach alternates between two levels of solution until the optimal solution is obtained. These algorithms are guaranteed to converge to the correct optimum only for linear or convex problems.

For nonlinear problems, Haftka[12] developed a penalty-function-based decomposition technique using Newton's method that results in substantial computational savings. However, this technique still requires equality constraints that make the problem ill conditioned. The purpose of the present paper is to propose a novel treatment of the global variables that are used to create the hierarchical structure. These variables are not used as optimization variables, but instead they are used as intermediate variables. This scheme helps to reduce computational cost but does not affect the optimization at all. It therefore allows us to reap the computational benefits of hierarchical design without incurring any of the disadvantages. An example of a portal frame is used to demonstrate the method and its computational advantages. Because the paper is intended to demonstrate the use of intervening variables rather than develop a general theory, only two-level systems are discussed.

Presented as Paper 87-0716 at the AIAA/ASME/AHS/ASEE 28th Structures Structural Dynamics and Materials Conference, Monterey, CA, April 6-8, 1987; received June 17, 1987; revision received Feb. 25, 1988.

*Principal Engineer.

†Professor, Department of Aerospace and Ocean Engineering. Member AIAA.

Solution Procedure

Single-Level Penalty-Function Approach

Consider the general optimization problem of the form

$$\text{minimize} f(X) \tag{1}$$

such that

$$g_i(X) \geq 0, \quad i=1,\ldots,n_i$$

and

$$h_i(X) = 0, \quad i=1,\ldots,n_e$$

where X is a design variable vector with components $x_1,\ldots,x_{\text{NDV}}$.

Among the numerous numerical techniques available to obtain a solution to Eq. (1), one popular technique is the sequence of unconstrained minimization technique (SUMT).[13] In this technique, the constrained optimization problem is transformed into a series of unconstrained minimization problems by replacing the constraints by penalty terms. Here, the inequality constraints are replaced by a quadratic extended interior penalty function[14] as implemented in the NEWSUMT-A optimization package.[16,17] NEWSUMT-A was modified to handle equality constraints using exterior penalty terms. The form of the augmented objective function Φ is

$$\Phi(X,r) = f(X) + r\sum_{i=1}^{n_i} p(g_i) + \frac{\beta}{\sqrt{r}}\sum_{i=1}^{n_e} q(h_i) \tag{2}$$

where r is a penalty multiplier, β is used to adjust the equality constraint penalty, and $p(g_i)$ is a quadratic extended interior penalty function for inequality constraints.[14]

The penalty $q(h_i)$ associated with the ith equality constraint is defined as an exterior penalty function of the form

$$q(h_i) = h_i^2 \tag{3}$$

The solution of the optimization problem is obtained by minimizing the function Φ for a decreasing sequence of r values using Newton's method with approximate second derivatives of the penalty terms.[18] The optimization is started with the penalty multiplier r at a value R_{init}, and the total function Φ given by Eq. (2) is minimized. The design space for a fixed value of r is called a response surface. Then, r is reduced by multiplying it by a factor R_{mult}, and another response surface is optimized. This process is continued until convergence to a desired accuracy is achieved.

Newton's Method with Approximate Second Derivatives

The direction vector S that minimizes $\Phi(X, r)$ is found by using Newton's method with a one-dimensional search.

$$\Delta X = X - X_0 = \alpha S = -\alpha H^{-1} G \tag{4}$$

where α is the step size obtained by a one-dimensional search, S is the direction vector, G is the gradient vector of Φ, X_0 is the initial design variable vector, and H is the Hessian matrix (second derivative matrix of Φ):

$$H_{ij} = \frac{\partial^2\Phi}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_i \partial x_j} + r\sum_{k=1}^{n_i}\frac{\partial^2 p(g_k)}{\partial x_i \partial x_j} + \frac{\beta}{\sqrt{r}}\sum_{k=1}^{n_e}\frac{\partial^2 q(h_k)}{\partial x_i \partial x_j} \tag{5}$$

For equality constraints,

$$\frac{\partial^2 q(h_k)}{\partial x_i \partial x_j} = 2\frac{\partial h_k}{\partial x_i}\frac{\partial h_k}{\partial x_j} + 2h_k\frac{\partial^2 h_k}{\partial x_i \partial x_j} \tag{6}$$

Approximate second derivatives are evaluated as suggested by Haftka.[18] The second term in Eq. (6) goes to zero as the optimization progresses and the equality is satisfied. Thus, Eq. (6) becomes

$$\frac{\partial^2 q(h_k)}{\partial x_i \partial x_j} = 2\frac{\partial h_k}{\partial x_i}\frac{\partial h_k}{\partial x_j} \tag{7}$$

A similar approximation is used for the second derivatives of p.[14]

These approximations for the second derivatives of Φ require only the first derivatives of the constraints; therefore, the computational effort required in finding search directions is reduced. During a one-dimensional minimization, instead of exact evaluations of constraints, various approximations using Taylor-series expansions can be used.[17,19] NEWSUMT-A implements linear, quadratic, reciprocal, and conservative convex approximations.

We denote the total number of constraints as NCON where $\text{NCON} = n_i + n_e$. Then from the foregoing relations, it is evident that for the gradient vector we need to evaluate $\text{NDV} \times \text{NCON}$ derivatives, whereas we need approximately $\text{NCON} \times \text{NDV} \times \text{NDV}/2$ terms for the symmetric Hessian matrix. The evaluation of the Hessian matrix can constitute a major computational effort when the numbers of design variables and constraints are large.

Based on past experience[21,22] for better computational performance, Eq. (4) is modified into

$$\Delta X = X - X_0 - = \alpha[H + H_c]^{-1} G \tag{8}$$

where H_c is a diagonal matrix with terms of the form ϵH_{ii} along the diagonal. These correction terms are used to make the original Hessian matrix H more diagonally dominant to improve the conditioning of the system.[15] Typical values of ϵ are in the range 0.01–0.05.

Hierarchical Formulation

It is now assumed that the system can be divided into s substructures, each with its own independent design variable vector X_i defined for $i = 1,\ldots,s$.

Consider a structural optimization problem of the form

$$\text{minimize} f(X_1,\ldots,X_s)$$

such that

$$g_i(X_1,\ldots,X_s) \geq 0, \qquad i=1,\ldots,m \tag{9}$$

where X_i is a vector of local design valuables associated with the ith part of the structure. In the form of Eq. (9), the system is fully coupled.

Often, it is possible to find a vector of decoupling (or global) variables Y such that the system of Eq. (9) may be written as

$$\text{minimize} f = f_0(Y) + \sum_{i=1}^{s} f_i(Y, X_i) \tag{10}$$

such that

$$g_{0j}(Y) \geq 0, \qquad j=1,\ldots,n_0$$

$$g_{ij}(Y,X_i) \geq 0, \quad i=1,\ldots,s, \qquad j=1,\ldots,n_i$$

$$h_{ij}(Y,X_i) = 0, \quad i=1,\ldots,s, \qquad j=1,\ldots,n_e$$

The equality constraints h_{ij} usually define the relationship of the decoupling variables Y and the original X_i.

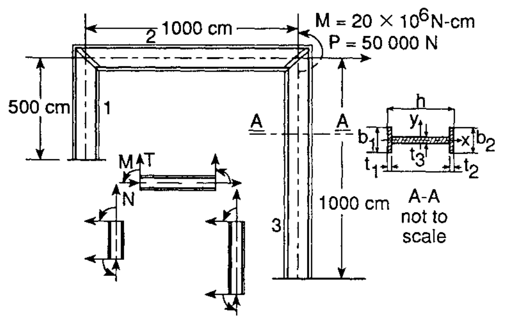

Fig. 1 Geometry of frame with *I*-beam cross section.

For example, consider the portal frame problem (Fig. 1) considered by Sobieski and co-workers.[7,20] The cross section of each beam element is an *I*-section and the dimensions defining the section are the detail design variables. With these detail design variables, the problem is fully coupled in that a change in a detail of one beam influences the stresses in the other two beams. However, it is possible to decouple the problem by using the cross-sectional area A and the moment of inertia I of the beam. With these two variables defined for each beam, it is possible to determine the global response of the complete structure, i.e., displacements and forces that act between the beams. These two quantities can be considered the decoupling variables; the detail variables of one beam do not influence the stresses in the other two beams. Equality constraints are used to relate A and I to the detail variables describing the beam cross section.

It is possible to take advantage of the decoupled form of Eq. (10) by using a two-level optimization solution. However, even if a single-level solution is used, the decoupling process produces several computational advantages. One advantage of the decoupled system is that it is much cheaper to calculate derivatives of the constraints with respect to the design variables. The decoupling implies that changes in the local variables X_i of one subsystem do not cause any changes in all other subsystems as long as the decoupling variables Y are fixed. Therefore, derivatives with respect to the X_i variables usually become much cheaper to calculate. For example, in the frame problem if details of one beam cross section are changed without changing A and I, only stresses in that beam are affected, and they can be calculated without an overall analysis of the frame.

A second advantage of the decoupled system is that the single-level solution process can become cheaper by taking advantage of the decoupling. This is demonstrated here for the penalty-function optimization procedure discussed earlier.

Penalty-Function Equivalent of Decoupled System

For the decoupled system of Eq. (10), the total function Φ to be minimized is

$$\Phi_i(Y, X_i) = f_0(Y) + \sum_{i=1}^{s} f_i(Y, X_i) + r\sum_{j=1}^{n_0} p[g_{0j}(Y)] + r\sum_{i=1}^{s}\sum_{j=1}^{n_i} p[g_{ij}(Y, X_i)] + \frac{\beta}{\sqrt{r}}\sum_{i=1}^{s}\sum_{j=1}^{ne} q[h_{ij}(Y, X_i)] \tag{11}$$

We rewrite Eq. (11) as

$$\Phi(Y, X_i) = \Phi_0(Y) + \sum_{i=1}^{s}\Phi_i(Y, X_i) \tag{12}$$

where

$$\Phi_0(Y) = f_0(Y) + r\sum_{j=1}^{n0} p[g_{0j}(Y)] \tag{13}$$

and

$$\Phi_i(Y, X_i) = f_i(Y, X_i) + r\sum_{j=1}^{n_i} p[g_{ij}(Y, X_i)] + \frac{\beta}{\sqrt{r}}\sum_{j=1}^{n_e} q[h_{ij}(Y, X_i)] \tag{14}$$

Applying Newton's method directly to Eq. (11) for the minimization of Φ, we have the following form of the equation for ΔY and ΔX:

$$\begin{bmatrix} H_{00} & H_{01} & \cdots & H_{0s} \\ H_{01}^T & H_{11} & & \bigcirc \\ \vdots & & \vdots & \\ H_{0s}^T & \bigcirc & & H_{ss} \end{bmatrix} \begin{Bmatrix} \Delta Y \\ \Delta X_1 \\ \vdots \\ \Delta X_s \end{Bmatrix} = - \begin{Bmatrix} \partial\Phi/\partial Y \\ \partial\Phi/\partial X_1 \\ \vdots \\ \partial\Phi/\partial X_s \end{Bmatrix} \tag{15}$$

where

$$H_{00} = \frac{\partial^2\Phi}{\partial Y^2} = \frac{\partial^2\Phi_0}{\partial Y^2} + \sum_{i=1}^{s}\frac{\partial^2\Phi_i}{\partial Y^2} \tag{16}$$

$$H_{ii} = \frac{\partial^2\Phi}{\partial X_i^2} = \frac{\partial^2\Phi_i}{\partial X_i^2} \tag{17}$$

$$H_{0i}^T = \frac{\partial^2\Phi}{\partial X_i\partial Y} = \frac{\partial^2\Phi_i}{\partial X_i\partial Y} \tag{18}$$

The block-sparse structure of the Hessian matrix in Eq. (15) is beneficial in contrast to a fully populated Hessian obtained for the coupled problem.

Benefits of Decoupling in the Solution Process

Equation (15) that needs to be solved for the direction vector in the Y and X_i space is block-sparse due to the lack of coupling between the subsystems, i.e., $\partial^2\Phi/\partial X_i\partial X_j = 0$ for $i \neq j$.

Savings in storage and computation of the Hessian matrix can be achieved by taking advantage of the sparsity. The contributions to the Hessian matrix need not be summed over all the constraints but only over those affected by changes in the design variable being perturbed.

The magnitude of the computational savings is a function of the number of subsystems and the number of local variables in each subsystem. The larger these numbers are, the greater the benefits of using the decoupled system. For typical large engineering systems, in which multilevel techniques are being considered, these numbers are high, and such systems are, therefore, ideally suited for the proposed scheme of solution. There are three main sources of savings that can be identified: 1) savings in computing the terms of the Hessian matrix, 2) savings in computing the constraint gradients since changes in local variables do not affect global quantities, and 3) savings in the solution of Eq. (15) if the sparsity of this equation is used in the solution process.[12]

Decoupling Intermediate Variables

The hierarchical structure described in the previous section is often generated by using equality constraints that can cause severe numerical difficulties.[9] These are apparently associated with the violation of the constraints during straight-line moves in the design space. In addition, the use of equality constraints increases the number of design variables because both local and global variables are retained. It is possible to reduce the number of variables by using a local linearization of the equality constraints to eliminate some of the local variables. However, such a technique would still be expected to have numerical difficulties associated with constraint violations at the end of each one-dimensional move.

In this work, an alternate approach is proposed in which the global variables Y are used only as intermediate variables for the purpose of reducing the cost of the optimization process. The original design variables are retained for the purpose of the optimization, which is carried out at a single level. Therefore, problems associated with multilevel optimization and problems attributed to the use of equality constraints are avoided. However, all the computational advantages of the multilevel formulation are still retained.

The proposed technique has two stages. In the first stage, the equality constraints are linearized and used to eliminate some of the local variables. The system consisting of the global variables and the remaining local variables is decoupled in the sense that local variables affect only local constraints. The gradient and second-derivative calculations are inexpensive in this decoupled system. The second stage consists of a transformation of the gradient and second derivatives from the decoupled system to the original system. This transformation (which is shown to be trivial) permits us to carry out the optimization in the original design space while still enjoying the computational benefits of a hierarchical formulation.

The first stage of the proposed technique begins with the equality constraints, Eq. (10). We consider these constraints as implicit equations for n_e of the components of the X_i in terms of the vector Y and the remaining components of X_i. For a general complex case, it is not possible to do this explicitly, but an explicit solution is not required for our purpose. For the case considered here, we make a simplifying assumption that is not essential in general; we assume that the global variables can be expressed in terms of the local variables. For example, for the frame problem, the cross section area A is expressed in terms of the local variables as (see Fig. 1)

$$A - [t_3(h-t_1-t_2) + b_1 t_1 + b_2 t_2] = 0 \tag{19}$$

Thus, the equality constraints for the ith subsystem define a subset $\boldsymbol{D}_i$ of order n_e of the vector Y that can be expressed in terms of the local variables X_i. Since we plan to use the equality constraints to eliminate some of the local variables, we denote those as $\boldsymbol{E}_i$ (the eliminated set) and the remaining variables of X_i as $\boldsymbol{R}_i$ (the retained set). The set of variables $\boldsymbol{D}_i$ (the decoupling set) are of the same order as $\boldsymbol{E}_i$ (both are of order n_e). The equality constraint is, therefore, a relation of the form

$$\boldsymbol{D}_i = \boldsymbol{D}_i(\boldsymbol{R}_i, \boldsymbol{E}_i) \tag{20}$$

In general, Eq. (20) cannot be explicitly solved for $\boldsymbol{E}_i$, so that it is considered to be only an implicit relation for the $\boldsymbol{E}_i$. However, it is assumed that the relation is invertible in the sense that the mapping from the $\boldsymbol{D}_i$ to the $\boldsymbol{E}_i$ variables is one-to-one, onto and continuous. We also assume that the $\boldsymbol{D}_i$ variables do not intersect, i.e., each subsystem has its own global or decoupling variables. The original set of design variables X may be rearranged and written as

$$X^T = [\boldsymbol{E}_1^T, \boldsymbol{R}_1^T, \ldots, \boldsymbol{E}_s^T, \boldsymbol{R}_s^T] \tag{21}$$

and the new set of variables X^* as

$$X^{*T} = [\boldsymbol{D}_1^T, \boldsymbol{R}_1^T, \ldots, \boldsymbol{D}_s^T, \boldsymbol{R}_s^T] \tag{22}$$

The objective function may be written as

$$f = f_0(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_s) + \sum_{i=1}^{s} f_i(\boldsymbol{D}_i, \boldsymbol{R}_i) \tag{23}$$

and the constraints as

$$g_{0j}(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_s) \geq 0, \quad j = 1, \ldots, n_0 \tag{24a}$$

$$g_{ij}(\boldsymbol{D}_i, \boldsymbol{R}_i) \geq 0, \quad i = 1, \ldots, s, \quad j = 1, \ldots, n_i \tag{24b}$$

For the frame problem of Fig. 1, the area A and moment of inertia I are a natural decoupling set. Equation (19) and a similar equation for I are the particular form of Eq. (20). With A and I of each beam fixed, the detail sectional variables of one beam do not affect constraints in the other two beams. Because of the decoupled form of the objective function and constraints in Eqs. (23) and (24), their derivatives are inexpensive to calculate. We next consider the transformations that may be used to calculate the derivatives of the original system from the derivatives of the decoupled system.

Derivative Transformation

Consider a general function

$$f = f(X_1, \ldots, X_s) \tag{25}$$

We need to compute the derivatives of f with respect to all its variables. Introducing the decoupling variables $\boldsymbol{D}$, we can write the function in Eq. (25) as

$$f = f_0(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_s) + \sum_{i=1}^{s} f_i(\boldsymbol{D}_i, \boldsymbol{R}_i) \tag{26}$$

For simplicity, let us start by considering the case of a single subsystem. Let f be the function in the coupled $\boldsymbol{E}$-$\boldsymbol{R}$ space and f^* the function in the decoupled $\boldsymbol{D}$-$\boldsymbol{R}$ space. Then, we have

$$f_i(X_i) = f_i(\boldsymbol{E}_i, \boldsymbol{R}_i) = f_i^*(\boldsymbol{D}_i, \boldsymbol{R}_i) \tag{27}$$

For convenience, we drop the index i to get

$$f(X) = f(\boldsymbol{E}, \boldsymbol{R}) = f^*(\boldsymbol{D}, \boldsymbol{R}) \tag{28}$$

and

$$\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{R}, \boldsymbol{E}) \tag{29}$$

We denote the number of variables of $\boldsymbol{X}$, $\boldsymbol{D}$, $\boldsymbol{R}$, and $\boldsymbol{E}$ as n_x, n_d, n_r, and n_e, respectively. Obviously, $n_x = n_r + n_e$, and $n_d = n_e$. We assume that f and f^* are continuous functions and so are their first and second derivatives with respect to each element of the vectors $\boldsymbol{R}$, $\boldsymbol{E}$, and $\boldsymbol{D}$.

We define the design variable vector X in the $\boldsymbol{E}$-$\boldsymbol{R}$ space as

$$X^T = [\boldsymbol{E}, \boldsymbol{R}]^T \tag{30}$$

and the design variable vector X^* in the transformed $\boldsymbol{D}$-$\boldsymbol{R}$ space as

$$X^{*T} = [\boldsymbol{D}, \boldsymbol{R}]^T \tag{31}$$

The derivatives of f^* are easily obtained, and we seek transformation rules to obtain the derivatives of f. Using chain rule differentiation yields

$$f_R = f_R^* + f_D^* \boldsymbol{D}_R \tag{32}$$

and

$$f_E = f_D^* \boldsymbol{D}_E \tag{33}$$

The derivatives with respect to R and E are used in one-dimensional searches to approximate the constraints.

System Transformation Matrix

For calculating a search direction, we need the relation between increments of the design variables. By differentiating Eq. (29) and inverting, we get

$$\Delta X = t \Delta X^* \tag{34}$$

where the transformation matrix t is

$$[t] = \begin{bmatrix} I & 0 \\ -D_E^{-1}D_R & D_E \end{bmatrix} \tag{35}$$

When applied to the entire system, the foregoing transformation becomes

$$\Delta X = T\Delta X^* \tag{36}$$

where

$$T = \begin{bmatrix} t_1 & & & 0 \\ & t_2 & & \\ & & \ddots & \\ 0 & & & t_s \end{bmatrix} \tag{37}$$

Calculating a Search Direction

If the optimization were conducted in the decoupled design space X^*, then a search direction S^* would be obtained by solving

$$H^*S^* = -G^* \tag{38}$$

where H^* and G^* are the Hessian matrix and gradient vector in X^*. The corresponding direction in the original space is TS^* and, in general, it will be different from the direction S obtained by solving the direction equation in the original space

$$HS = -G \tag{39}$$

It is tempting to conduct the optimization in X^* because Eq. (38) is inexpensive to set and solve compared to the fully coupled Eq. (39). However, there are several problems associated with working in X^*. First, simple side constraints on the E_i can become complex nonlinear constraints in X^*. Second, after each one-dimensional search, the eliminated variables (the E_i) need to be calculated by solving Eq. (29) for each subsystem. Because Eq. (29) is, in general, nonlinear, there is no guarantee that a solution exists. Therefore, instead of working in X^*, we seek a cheap way of calculating S.

Denoting by $\hat{S}^*$ the direction in X^* corresponding to S, we have

$$S = T\hat{S}^* \tag{40}$$

Also, it is easy to check from Eqs. (32–35) that

$$G = (T^T)^{-1}G^* \tag{41}$$

Substituting from Eqs. (40) and (41) into Eq. (39) and premultiplying by T^T we obtain

$$\hat{H}\hat{S}^* = -G^* \tag{42}$$

where

$$\hat{H} = T^THT \tag{43}$$

It can be shown[21] that $\hat{H}$ has the same sparsity pattern as H^* so that it is much cheaper to calculate than H, and Eq. (42) is much cheaper to solve than Eq. (39). Furthermore, when the matrix H is calculated using the first-derivative approximations [such as Eq. (7)], it can be shown[21] that $\hat{H} = H^*$. Thus, for this case, we can eat our cake and have it too! Therefore, the two computational steps of search direction calculations are as follows:

1) Use Eq. (8) to solve for the search direction in the decoupled space, i.e., use the inexpensive Eq. (38) to calculate S^* that is now equal to $\hat{S}^*$.

2) Obtain a search direction in the original coupled space S from Eq. (40).

It should be noted that Eq. (38) is replaced by

$$[H^* + H_c^*]S^* = -G^* \tag{44}$$

where $H_c^* = T^TH_cT$.

Here, H^* is obtained directly from finite-difference calculations in the decoupled design space. Then it is corrected by adding H_c^* to it. The computation of H_c^* is not expensive since H_c is a diagonal matrix and the transformation matrix T is also block diagonal.

Numerical Results

Consider the minimum volume design of the portal frame shown in Fig. 2. This structure was first formulated and solved by Sobieski et al.[20] The frame is made of three beams, and the cross sections of the beams were I-sections (see Fig. 1) as considered by Sobieski. Six detailed design variables governed the size of the beam cross sections. The problem is fully coupled in that any of the details of each beam influences the global displacements and the stresses in the other beams by changing

Table 1 Portal frame example: design information

Material		Aluminum		
Young's modulus				7.06E6 N/cm^2
Normal stress limit				20,000 N/cm^2
Shear stress limit				11,600 N/cm^2
Number of load conditions		2		
Load case		Degree of freedom		Load
1		4		5.0E4 N
2		6		−2.0E7 N-cm
Degree of freedom				Maximum allowable displacement
4				± 4.0 cm
6				± 0.015 rad
Dimension variable	Design variable	Initial value, cm	Lower bound, cm	Upper bound, cm
d_1	x_1	1.0	0.1	5.0
d_2	x_2	50.0	10.0	–
d_3	x_3	0.25	0.01	–
d_4	x_4	0.025	0.001	–
d_6	x_5	0.5	0.01	–

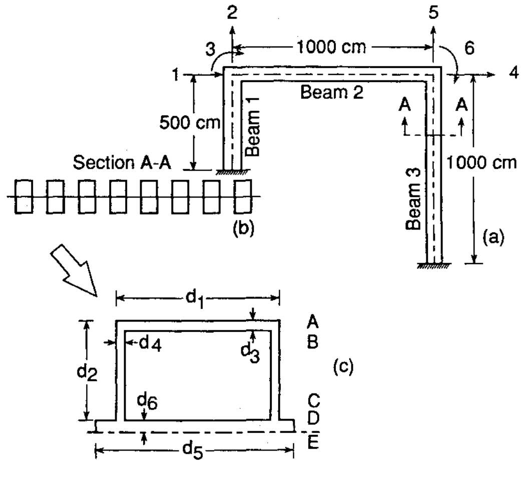

Fig. 2 Geometry of frame with hat-stiffener cross section.

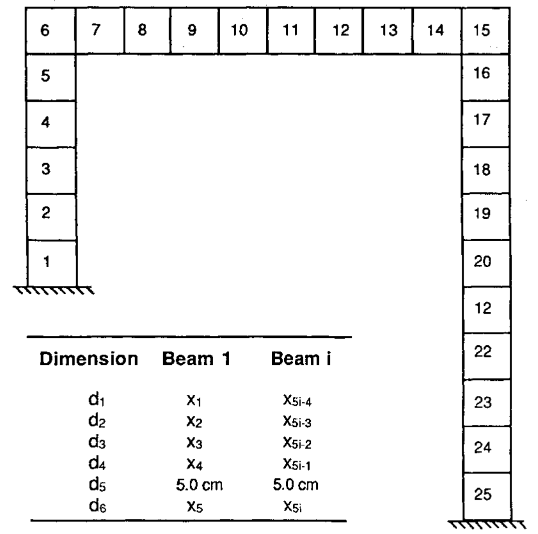

Fig. 3 Finite-element model for portal frame FRAME125.

Table 2 Optimization results for portal frame FRAME050 using linear approximations for constraints

	Coupled system	Decoupled system	Savings, %
Total function, cm^3	18,057.5	18,057.5	
Objective function, cm^3	18,045.2	18,045.2	
Number of ODM	36	36	
Global analyses	1910	830	56.54
	Number of evaluations		
Objective function	4119	4123	
Constraints	73	73	
Constraint gradients	36	36	
Approximate constraints	410	414	
	CPU times[a]		
Total	40.95	28.19	31.16
ODM	1.42	1.41	
Direction	39.36	26.59	
Hessian setup	18.92	8.97	52.59
Solution	1.56	1.55	
Objective function	1.12	1.16	
Constraints	0.70	0.72	
Constraint gradients	16.86	9.31	44.78
Approximate constraints	0.09	0.09	

[a]CPU seconds on IBM 3084.

the internal load distribution. For an *I*-beam with six cross-sectional parameters and two global parameters A and I, it is possible to replace two local parameters by the global parameters. In order to consider a more complex cross section, hat-stiffened beams are assumed instead of the *I*-section beam. Each beam is 40 cm wide with eight identical symmetric hat-stiffeners as shown in Fig. 2b. Two different load conditions act on the portal frame, which is required to satisfy displacement and stress constraints for each load condition separately as shown in Table 1. The first eight constraints are global displacement constraints in degrees of freedom 4 and 6 (see Fig. 2) due to load conditions 1 and 2. The remaining constraints are local stress constraints, 18 for each beam. There are four normal stresses and five shear stresses monitored at points A, B, C, D, and E in Fig. 2c of the hat-stiffener for each of the two load conditions. There are also six side constraints, five lower bounds on each of the design variables, and one upper bound on d_1. Variable d_5 is fixed at 5 cm (Fig. 2) to preserve the overall width of the beams. Each design variable affects each displacement and stress constraint. As a result, evaluation of finite-difference gradients of constraints requires a complete solution of the entire structure after perturbing each design variable in turn. The displacement constraints are global constraints as they depend only on the global design variables A and I and not on local details. On the other hand, the stress constraints in a beam are local constraints as they depend on the local details of the cross section of that beam. They also depend on the global design variables that determine the internal forces in each beam but not on the detail design variables of other beams. Equality constraints such as Eq. (19) ensure consistency of global and local design variables. No attempt was made to develop explicit expressions relating the detail design variables to the global design variables.

A variable cross section version of the portal frame example is considered, with each of the three beams modeled by several finite elements. The size of the problem is varied by changing the number of elements in each beam. Three models of the frame that are studied are designated as FRAME050, FRAME125, and FRAME500 with 50, 125, and 500 design variables, respectively. The finite-element model for FRAME125 is shown in Fig. 3.

Decoupling was achieved by eliminating the design variables associated with d_1 and d_3 of Fig. 2c and replacing them by the global variables, the cross-sectional area A, and the moment of inertia I.

The results presented focus on the computational efficiency of the decoupling technique. The total CPU time consists of two main components: 1) the CPU time required to compute search directions displayed in the tables as direction time, and 2) the CPU time for performing one-dimensional minimizations displayed as ODM time. The direction time further consists of three main components: 1) the CPU time required to compute the derivatives of the constraints displayed as constraint gradients time, 2) the CPU time required to evaluate the Hessian matrix displayed as Hessian setup time, and 3) the CPU time required to solve the direction equation displayed as solution time. Additionally, the results also show the reduction in the number of global analyses. These are very inexpensive for the frame problem, but in more complex problems they are a major computational task. The details of the FRAME125 model designs are given in Ref. 22. The other two models are presented next.

Portal Frame FRAME050

The portal frame was modeled with 10 250-cm long elements: two elements for beam 1, and four elements each for beams 2 and 3. Each element had five design variables controlling its cross-sectional shape. Thus, in this model (denoted FRAME050), there were a total of 50 design variables, including 20 global variables. The total number of constraints was 248, eight global displacement constraints and 24 constraints on stresses and upper and lower bounds in each element.

Using the decoupling method, similar weights for the final design were obtained for exact constraints[21] and for the linear (Table 2), approximations of the constraints. The same design is obtained with either the coupled or the decoupled system with a 20–30% reduction in computational time. A summary of the CPU times for this case is shown in Fig. 4a. The computational time for evaluating the Hessian matrix was reduced by 50% and the computational time for the constraint derivatives by 45–60%. These savings were much larger than the increased effort required for the implementation of the decoupling algorithm, resulting in a 20–30% net reduction in the total CPU time. The number of global analyses was reduced by 45–60% (see Fig. 4b). Each constraint derivative evaluation required 51 global analyses for constraint evaluations for the coupled system, whereas only 21 global analyses were necessary for constraint evaluations for the decoupled system.

Portal Frame FRAME500

The finest mesh used 100 25-cm long elements for a total of 500 design variables. This model was named FRAME500. The number of global variables was 200, and the total number of constraints was 2408, eight global displacement constraints and 24 constraints on stresses and upper and lower bounds in each element.

The computational times and memory requirements of this case were very high. Consequently, this case was run on a larger and faster computer, a CDC CYBER 205. Furthermore, instead of a complete optimization, only a single one-dimensional minimization with a linear approximation for the constraints was carried out. Consequently, the computational times for this case should not be directly compared with those from FRAME050 and FRAME125.

The same design is obtained from using either the coupled or the decoupled system with almost a 75% reduction in computational time (Table 3). Figure 5a summarizes the CPU times. The computational time for evaluating the Hessian matrix was reduced by 97%, and the computational time for constraint derivatives by 58%. Each constraint derivative evaluation required 501 global analyses for constraint evaluations for the coupled system, whereas only 201 global analyses were necessary for constraint derivatives for the decoupled system (Fig. 5b). This resulted in almost a 75% reduction in the total CPU time. This figure would be 80% if the solution strategy outlined in the previous section had been implemented. In that case, the solution time for the decoupled system would have been about 2 s instead of 30.

A summary of the CPU time savings for the three models of the portal frame is given in Table 4. As the size of the model and the degree of sparsity of the Hessian matrix increase, greater savings are obtained with the decoupled system. The savings in constraint gradient calculations is limited to 60% for this example since there are five local variables and two global variables for each subsystem. For structures where a larger number of detailed design variables are used, this component of savings will increase proportionately. For very large systems, the evaluation of the Hessian matrix can be a domi-

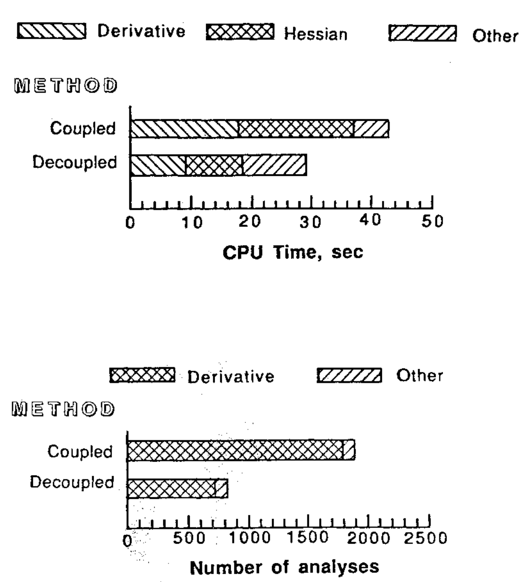

Fig. 4 CPU times and number of global analyses for coupled and decoupled systems for portal frame FRAME050, linear approximation.

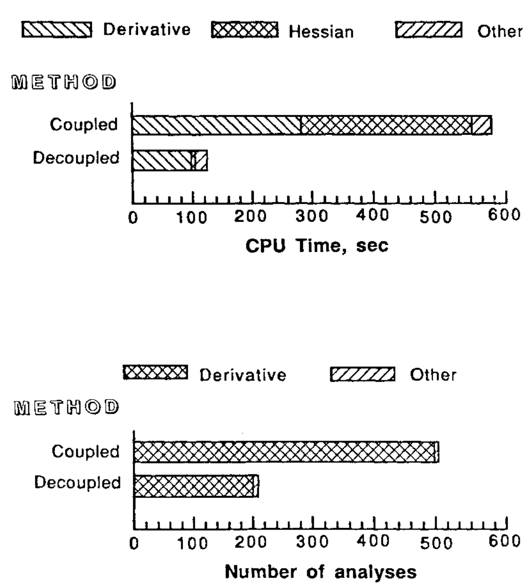

Fig. 5 CPU times and number of global analyses for coupled and decoupled systems for portal frame FRAME500, linear approximation.

Table 3 Optimization results for portal frame FRAME500 using linear approximation for constraints

	Coupled system	Decoupled system	Savings, %
Total function, cm^3	6.676*E*5	6.676*E*5	
Objective function, cm^3	1.472*E*5	1.471*E*5	
Number of ODM	1	1	
Global analyses	504	204	59.52
	Number of evaluations		
Objective function	1031	1028	
Constraints	2	2	
Constraint gradients	1	1	
Approximate constraints	28	25	
	CPU times[a]		
Total	576.72	156.10	72.93
ODM	0.92	0.89	
Direction	574.42	136.61	
Hessian setup	279.43	5.89	97.89
Solution	29.98	29.80	
Objective function	1.07	1.07	
Constraints	1.37	1.37	
Constraint gradients	262.62	109.01	58.49
Approximate constraints	0.04	0.04	

[a]CPU seconds on CDC CYBER 205.

Table 4 Comparison of results for all models of the portal frame

	FRAME050	FRAME125	FRAME500
No. of design variables	50	125	500
No. of constraints	248	608	2408
Optimized volume, cm^3	18,045	16,761	–
Average cost of a global analysis	9.19*E*−3[a]	6.96*E*−2[a]	4.87*E*−1[b]
	Savings, %		
Total CPU time	31.2	49.9	72.9
Global analyses	56.5	58.5	59.5
Storage			
Hessian	31.8	34.3	35.6
Constraint gradients	52.3	56.8	59.2

[a]CPU seconds on IBM 3084. [b]CPU seconds on CDC CYBER 205.

nant portion of the total computational time, and the use of the decoupled system can reduce the computational costs by as much as 90%.

For the frame example just discussed, the finite-element modeling was relatively inexpensive. A global analysis of FRAME125 required only about 0.07 CPU s on a IBM 3084, whereas FRAME500 took about 0.5 s on a CDC CYBER 205. For complex structures, this time can be significantly larger, especially if a nonlinear analysis is required. Using the decoupled system saves 55–66% in the number of global analyses required (see Table 5).

This decoupling technique also has been applied to the optimization of a wing box structure. See Ref. 21 for details.

Concluding Remarks

Engineering design is hierarchical in nature, and if no attempt is made to benefit from this hierarchical nature, design optimization can be very expensive. There are two alternatives to take advantage of the hierarchical nature of typical structural systems. Multilevel optimization techniques incorporate the hierarchy at the formulation stage and result in the coordinated optimization of a hierarchy of subsystems. The use of multilevel optimization techniques often necessitates the use of equality constraints. These constraints can cause numerical difficulties during optimization. Single-level decomposition techniques take advantage of the hierarchical nature to reduce the optimization cost.

This paper demonstrated a single-level decoupling technique for a penalty-function-based optimization method employing Newton's method. This technique retains the advantages of a partitioned system of smaller independent subsystems without the disadvantages of multilevel formulations.

The proposed decoupling technique was demonstrated for a portal frame design problem. The size of the problem was varied by refining the mesh used to model the frame. Computational savings of up to 75% were obtained for large problems. The savings increase as the size of the problem and the amount of decoupling are increased. For truly large systems, this decoupling technique provides the necessary reduction of computational effort to make the optimization process viable.

Acknowledgment

This research was supported by NASA Grant NAG-1-168.

References

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

[2]Kirsch, U., *Optimum Structural Design*, McGraw-Hill, New York, 1981, pp. 198–215.

[3]Mesarovic, M. D., Macko, D., and Takahara, Y., *Theory of*

Hierarchical Multi-Level Systems, Academic, New York, 1970.

[4]Kirsch, U., "Multilevel Approach to Optimum Structural Design," *Journal of the Structural Division,* 1975, pp. 957–974.

[5]Schmit, L. A. and Ramanathan, R. K., "Multilevel Approach to Minimum Weight Design Including Buckling Constraints," *AIAA Journal,* Vol. 16, Feb. 1978, pp. 97–104.

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

[7]Sobieszczanski-Sobieski, J., James, B., and Riley, M. F., "Structural Optimization by Generalized Multilevel Optimization," AIAA Paper 85-0697, April 1985.

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

[9]Thareja, R. R. and Haftka, R. T., "Numerical Difficulties Associated with Using Equality Constraints to Achieve Multi-Level Decomposition in Structural Optimization," AIAA Paper 86-0854, May 1986.

[10]Dantzig, G. B., *Linear Programming and Extensions,* Princeton Univ. Press, Princeton, NJ, 1963.

[11]Rosen, J. B., "Primal Partition Programming for Block Diagonal Matrices," *Numerische Mathematik,* Vol. 6, 1964, pp. 250–260.

[12]Haftka, R. R., "An Improved Computational Approach for Multilevel Optimum Design," *Journal of Structural Mechanics,* Vol. 12, No. 2, 1984, pp. 245–261.

[13]Fiacco, A. V. and McCormick, G. P., *Nonlinear Programming, Sequential Unconstrained Minimization Techniques,* Wiley, New York, 1968.

[14]Haftka, R. T. and Starnes, J. H., "Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization," *AIAA Journal,* Vol. 14, June 1976, pp. 718–724.

[15]Miura, H. and Schmit, L. A., "NEWSUMT—A Fortran Program for Inequality Constrained Function Minimization—User's Guide," NASA CR-159070, June 1979.

[16]Thareja, R. R. and Haftka, T. T., "NEWSUMT-A—A Modified Version of NEWSUMT for Inequality and Equality Constraints—Version 1," VPI Rept. 148, Aerospace and Ocean Engineering Dept. Virginia Tech., Blacksburg, VA, June 1985.

[17]Grandhi, R. V., Thareja, R. R., and Haftka, R. T., "NEWSUMT-A: A General Purpose Program for Constrained Optimization Using Constraint Approximations," *Journal of Mechanisms, Transmissions, and Automation in Design,* Vol. 107, No. 1, March 1985, pp. 94–99.

[18]Haftka, R. T., "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," NASA TN D-7264, July 1973.

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

[20]Sobieszczanski-Sobieski, J., James, B., and Dovi, A., "Structural Optimization by Multilevel Decomposition," AIAA Paper 83-0832, May 1983.

[21]Thareja, R. R., "Efficient Single-Level Solution of Hierarchical Problems in Structural Optimization," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, April 1986.

[22]Thareja, R. R. and Haftka, T. T., "Efficient Single-Level Solution of Hierarchical Problems in Structural Optimization," AIAA Paper 87-0716, April 1987.