Sequential Linearization Method for Multilevel Optimization

G. N. Vanderplaats*
VMA Engineering, Goleta, California
and
Y. J. Yang† and D. S. Kim†
University of California, Santa Barbara, Santa Barbara, California

A sequential linearization method for multilevel design optimization is presented. The proposed method is based on the concept of providing linearized information between the system level and subsystem level optimization tasks. The advantages of the method are that it does not require optimum sensitivities, nonlinear equality constraints are not needed, and the method is relatively easy to understand and use. The disadvantage is that the coupling between subsystems is not dealt with in a precise mathematical manner. The principal motivation for the proposed method is that is closely represents the traditional multidiscipline design environment. Examples are presented to demonstrate the features of the proposed method.

Introduction

IN recent years, considerable research has been devoted to the development of multilevel optimization methods for design optimization. Two principal motives are offered for this effort. The first is that engineering organizations operate in multilevel and multidiscipline organizational structure. Thus, in order to encompass all engineering disciplines in the optimization process, it is desirable to provide techniques that encompass all disciplines and that allow each group to work with reasonable independence. The second motivation is to effectively break the overall optimization task into smaller parts that are more amenable to a solution using numerical optimization methods.

It should be understood at the outset that multilevel methods will probably not reduce overall computational costs relative to more traditional single-level methods. Multilevel methods are primarily intended to make a problem tractable by formally dealing with the interactions that exist among disciplines that will normally not be considered simultaneously either because of their complexity or because organizational structure prevents it. Thus, there is an analogy with substructuring in finite-element analysis where the process is performed for convenience rather than for reduced computational costs.

Much of the work in this area has followed the concepts presented by Sobieszczanski-Sobieski and co-workers.[1,2] The basic approach defined in these works is to decompose the overall design task into two or more levels while maintaining the essential interdependence among them. While it is not representative of the scale of design tasks that are to be addressed, the simple portal frame shown in Fig. 1 has served as a common example to test and demonstrate the concepts developed. Here, a system-level design task is created where the structural weight is minimized, subject to system-level constraints, which are displacement and rotation limits at the joints of the structure. The subsystem design tasks are to determine the dimensions of the individual elements such that the local constraints on stress and local buckling are satisfied by the widest possible margin. The suboptimization problem requires that several nonlinear, equality constraints must be satisfied relating the local variables (member dimensions) to the system variables (cross-sectional areas and moments of inertia). Upon completion of the subsystem optimization, a "cumulative" constraint is created as a linear function of the system-level variables. This is returned to the system level and is used as a set of additional conditions that must be satisfied during system-level optimization. This general method has the attraction that it properly accounts for the interactions among the system and the various subsystems. However, it has the disadvantages that nonlinear equality constraints must be used at the subsystem level, as potentially highly nonlinear cumulative constraints must be sent to the system level, and this cumulative constraint requires the calculation of "optimum sensitivities" which can be discontinuous.[3,4]

Recent work[5,6] has shown that an alternative method, but still within the overall concept of multilevel optimization, may be efficient in dealing with the difficulties of the works cited above. Here, the system-level variables and constraints are the same as before. However, an additional set of constraints are also considered. These are the linearized approximation to the subsystem constraints. At the subsystems, the engineer is free to choose his optimization problem as he traditionally would. For example, the structural designer may still minimize weight, subject to local stress and buckling constraints, and the aerodynamicist may still maximize lift-to-drag ratio, subject to pitching-moment and pressure-gradient constraints. Now, during the sublevel optimization problem, a linearized form of the system-level constraints is included. At the end of the sublevel optimization task, all active and near-active subsystem constraints are returned to the system-level optimization in linearized form. This method has the advantage that the design process more nearly corresponds to the actual industrial environment. Also at the subsystem level, it is only necessary to calculate the sensitivity of the subsystem constraints with respect to the system and subsystem level variables and return these to the system. It was not necessary to deal with equality constraints at the sublevel or to calculate sensitivities of the sublevel optimum with respect to the system-level variables, although sensitivity of the sublevel constraints with respect to

Presented as Paper 88-2226 at the AIAA SDM Issues of the International Space Station, Williamsburg, VA, April 21-22, 1988; received Sept. 9, 1988; revision received May 15, 1989.

*President. Associate Fellow AIAA.

†Graduate Research Assistant, Department of Mechanical and Environmental Engineering.

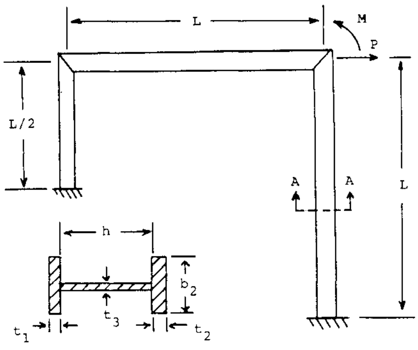

Fig. 1 Portal frame.

system-level variables must still be calculated. However, two notable disadvantages were reported. The first is that the formal mathematical interactions among the sublevels must be relaxed, and the second (far more critical, and related to the first) is that very small move limits had to be imposed at the sublevel optimization. Also, at the subsystem level, the system-level constraints were allowed to be changed by only a fractional amount. This was an attempt to deal with the interactions among the subsystem.

Here the limitation of small move limits is removed. Also, at the subsystem level, the system-level constraints are imposed (in linearized form) with their original limits. In the following sections, the general problem is outlined. In this development, it is understood that no effort is made to create system or subsystem problems that are especially efficient using modern approximation techniques. This is because the proposed method is intended for general application to disciplines where such approximations are not available. Clearly, in the case of structural optimization, where approximation methods are available, efficiency and reliability can only be enhanced. Examples are provided to demonstrate the features of the proposed method.

Basic Concepts

The basic concept of multilevel design may be understood by considering the simple cantilevered beam shown in Fig. 2. The objective is to minimize the material volume subject to limits on the deflection at the beam junction, at the tip, and on the maximum bending stresses in members. The design variables of interest are the width B_i and height H_i of each beam, and the length L_1 ($L_2 = L - L_1$). Clearly, for such a simple problem, this would be solved directly. However, for demonstration purposes, it is possible to formulate it as a multilevel problem with a system level and two subsystems.

The system-level problem may be stated as find the beam length L_1 and dimensions B_1, H_1, B_2, and H_2 to

Minimize

$$V = B_1 H_1 L_1 + B_2 H_2 (L - L_1) \tag{1}$$

Subject to

$$\delta_1 \le \bar{\delta}_1 \qquad \delta_2 \le \bar{\delta}_2 \tag{2}$$

Here, the system-level design includes all variables that control the system objective and all constraint functions. Thus,

$$Y = \begin{Bmatrix} B_1 \\ H_1 \\ B_2 \\ H_2 \\ L_1 \end{Bmatrix} = \begin{Bmatrix} x_1 \\ x_2 \\ X \end{Bmatrix} \tag{3}$$

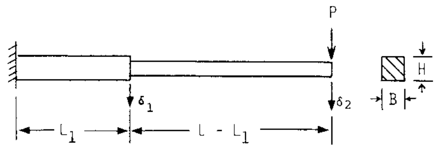

Fig. 2 Cantilevered beam.

The subsystem design problem for subsystem i is to find the actual member dimensions B_i and H_i that will

Minimize

$$V_i = B_i H_i L_i \tag{4}$$

Subject to

$$\sigma_i \le \bar{\sigma} \tag{5}$$

$$H_i \le 10 B_i \tag{6}$$

where

$$\chi_i = \begin{Bmatrix} B_i \\ H_i \end{Bmatrix} \tag{7}$$

Equation 5 is a stress constraint and Eq. (6) limits the depth to 10 times the width.

Note that, at the system level, all design variables are included. At the subsystem, the design variables that are important to the subsystem are considered, but the strictly system level variable L_1 is held fixed.

Because of the interdependence between the system and subsystem variables, each level will effect the other. The key issue is how to account for these interactions and, in the general case, how to account for competition between subsystems.

In previous methods, for this example, the system-level variables would be taken as the cross-sectional areas and moments of inertia since these directly effect the system objective and constraints. Then, at each subsystem optimization, equality constraints would be imposed to insure that the system optimization problem is maintained, and so the interaction among subsystems is directly controlled. In the present approach a much heavier burden will be imposed at the system level, in terms of the number of design variables and constraints, in order to reduce the overall nonlinearity of the problem.

Two-Level Optimization Formulations

For brevity, only two levels of the optimization problem are presented here. It is clear that multiple levels can be considered simply by treating each sublevel as a system with further subsystems. The general program organization is shown in Fig. 3 with a single system and multiple subsystems.

The complete set of design variables are contained in vector Y as

$$Y^T = \{(X_1, X_2, X_3, X_4), (x_5, x_6), (\ldots), (x_{N-2}, x_{N-1}, x_N)\} \tag{8}$$

where the number of components in parentheses relating to the system and subsystems is arbitrary and is given here as an example. Capital letters refer to the system and lower case letters refer to the subsystems. The vector Y contains the complete set of design variables, including those changed only by the system as well as all subsystems.

Now, the system-level design task can be stated in mathematical terms as find the set of system variables X and subsystem variables x_i that will

$$\text{Minimize } F(Y) \tag{9}$$

Subject to

$$G_j(Y) \le 0 \qquad j = 1,MS \tag{10}$$

$$g_{ji}(X,x_i) \le 0 \qquad j = 1,mss \qquad i = 1,nss \tag{11}$$

$$X_i^L \le X_i \le X_i^U \qquad i = 1,\text{NS} \tag{12}$$

$$x_{ij}^L \le x_{ij} \le x_{ij}^U \qquad i = 1,ns \qquad j = 1,nss \tag{13}$$

where *nss* and *mss* refer to the number of subsystems and their constraints, respectively, and *NS* and *MS* refer to the system design variables and system constraints, respectively. Here, it is understood that the subsystem constraints, $g_{ji}(x_i)$ are linearized to give the form

$$g_{ji}(X,x_i) = g_{ji}^0 + \nabla x g_{ji}(X,x_i) \cdot (x - x^0) + \nabla x_i g_{ji}(x,x_i) \cdot (x_i - x_i^0) \tag{14}$$

At each subsystem optimization, the problem solved is to find the set of subsystem variables x_i that will

$$\text{Minimize } F(x_i) \tag{15}$$

Subject to

$$G_j(x_i) \le 0 \qquad j = 1,MS \tag{16}$$

$$g_j(X,x_i) \le 0 \qquad j = 1,mss \tag{17}$$

$$x_i^L \le x_i \le x_i^U \qquad i = 1,ns \tag{18}$$

Now, it is understood that the system constraints $G_j(x_i)$ are the linearized form of the actual constraints in terms of the subsystem variables

$$G_j(x_i) = G_j + \nabla x G_j(X,x_i) \cdot (x_i - x_1^0) \tag{19}$$

Furthermore, there is an additional set of inputs to the subsystem for which there must be an account. This may be considered to be the boundary conditions to the suboptimization problem. For example, in the case of the simple beam example, the end forces on the elements are functions of the design variables. This is a departure from the method of Ref. 1, where the section properties were held constant at the subsystem and the end forces were constant as well. Here, these forces must be approximated in terms of the design variables. Now, when the subsystem variables are changed, the boundary conditions must first be updated as

$$BC = BC^0 + \nabla x BC \cdot (x - x^0) \tag{20}$$

where *BC* is taken to be any input to the subsystem that is dependent on the values of the subsystem variable themselves. Now when the subsystem variables are changed, the boundary conditions are first updated by Eq. (20), and then the necessary functions are calculated. Note also that, if the system constraints are functions of the boundary conditions, the appropriate additional terms must be added to Eq. (19) as well.

Finally, there must be allowance for the case where the system constraints are not functions of the subsystem variables. This is easily identified by the fact that the gradient term of Eq. (19) will be zero. In this case, the corresponding system constraint should be deleted from the subsystem level optimization.

The key idea here is that system-level functions are sent to the subsystems in linearized form, and the subsystem functions are sent to the system in linearized form. The net effect is that there is two-way communication relative to the functions involved. Furthermore, if the subsystems wish to send constraints relative to their own objective functions, this is possible. For example, a subsystem may send an upper bound on its own objective function (such as weight) to the system, and this must be respected in the system-level optimization.

In this approach, more system-level design variables and constraints are created. However, these are linearized relative to the subsystem variables and therefore more easily managed. At each subsystem, there are additional linearized constraints, but these are inequality constraints and thus are also more easily managed. Also, additional constraints are usually not a problem at this level since optimization is not too sensitive to the actual number of imposed constraints. Finally, the sensitivity information that must be provided is not optimum sensitivities in the sense of Ref. 1 but instead is sensitivity information in the usual sense.

The theoretical limitation of the proposed method is that the subsystems have some control over the system-level constraints. Thus, each subsystem may drive the system constraints to zero, yet on return to the system, these constraints will be violated. The system-level optimization must then overcome this constraint violation. Numerical experience has shown that this is a more theoretical than practical consideration. Occasionally, early in the design process, the system constraints will become violated for this reason. However, as the design progresses, this problem corrects itself, in part because of sequentially reduced move limits. Mainly, as the system constraints become critical, the subsystems maintain this criticality or drive the system constraints more feasible.

The overall algorithmic flow is shown in Fig. 4. The first step is to evaluate the system-level constraints and their gradients. Next, each subsystem optimization task is solved with the linearized system-level constraints imposed. At the end of the subsystem optimization, the constraints are linearized with

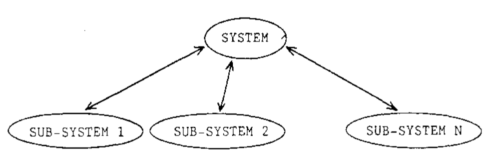

Fig. 3 General program structure.

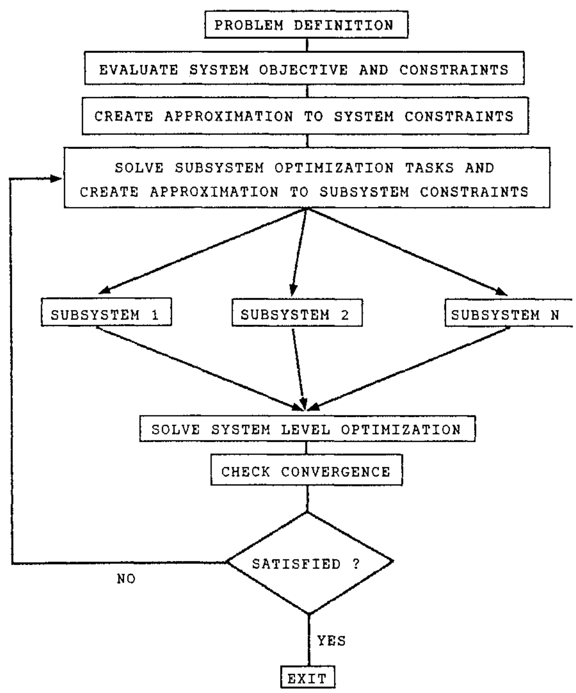

Fig. 4 Multilevel optimization flow.

Table 1 Results for simple beam

Parameter	Initial	Iteration 1 SS	Iteration 1 S	Iteration 2 SS	Iteration 2 S	Single level
L_1	80.0	80.00	50.40	50.40	50.40	50.40
B_1	1.0	0.68	0.71	0.71	0.71	0.71
H_1	5.0	6.82	7.10	7.10	7.10	7.10
B_2	1.0	1.60	0.64	0.53	0.53	0.53
H_2	5.0	8.00	4.98	5.30	5.30	5.30
Volume	500.0	628.72	412.10	393.19	393.43	393.42

respect to both the system and subsystem variables, and this information is returned to be included in the system-level opimization. Now, at the system level, all design variables are considered together. The subsystem constraints are included in their linearized form, and the system constraints are treated in their original form, except that the effects of the subsystem variables on the system objective and constraints are included in linearized form. Thus each level uses linearized information from the other to account for the necessary interactions.

This leads to an optimization task at each level in which the number of constraints is increased. Also, at the system-level, there is a dramatic increase in the number of design variables, even though the functions are explicit with respect to many of the variables. Finally, it should be noted that this is only the simplest description of the algorithm. For example, only critical and near-critical constraints need be linearized at each level, instead of all constraints. Also, the actual form of the approximation may not be linear since sufficient information is available to create reciprocal or mixed approximations if that is desirable.

Examples

Three design examples are presented here to indicate the features of the method.

As with any linearization technique, move limits are required at each stage. Here, initial move limits of 30% were used for the design variables at each level. These move limits are adjusted as the optimization progresses, based on a simple system-level criteria. The first is that if the system-level optimization appears to have converged (based on diminishing improvement in the system objective), all move limits are reduced by 50%. The second is that if on return from the subsystem optimizations, the system-level constraints are violated by an amount greater than they were on the previous return, all move limits are reduced by 20%. The system-level move limits are never reduced to less than 50%, and the subsystem move limits are never reduced to less than 3%.

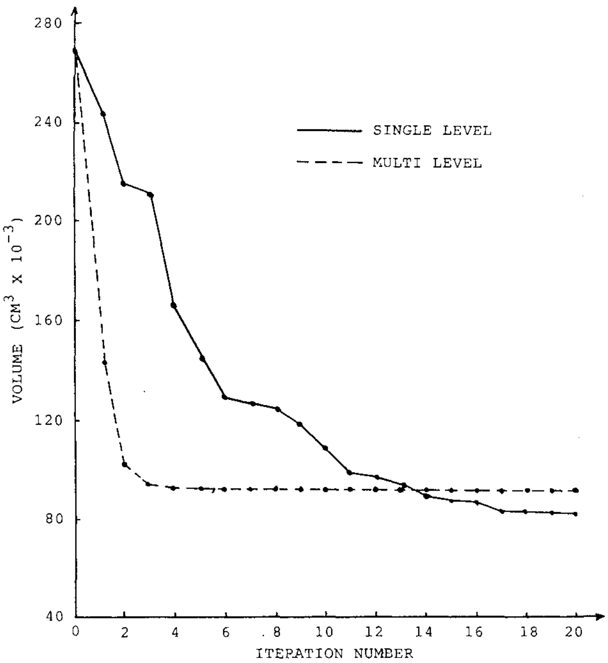

Fig. 5 Portal frame iteration history, initially feasible.

In order to test the algorithm as a general procedure, no effort was made to create special linearizations at either level, something which can be expected only to improve the results but which is dependent on problem type (e.g., structural design). No effort was made to "tune" the optimization process to solve a particular problem efficiently either.

In cases 2 and 3, the iteration histories are shown for the single and multilevel method. The purpose for this is only to show the convergence characteristics, and not as an indication of relative efficiencies. Clearly, because the multilevel method requires numerous single-level suboptimization tasks, it will be computationally more costly. However, for the examples considered here, the computational costs are not high enough to provide meaningful comparisons by themselves.

Table 2 Results for portal frame

Set	Parameter	Initial values	Single level	Multi-level	Initial values	Single level	Multi-level
1	b_1	30.0	10.5	7.67	11.00	10.4	9.92
	t_1	1.00	0.47	0.60	0.275	0.45	0.47
	h	50.0	78.3	79.3	22.00	79.4	79.0
	t_3	1.00	0.52	0.52	0.275	0.52	0.52
	b_2	30.0	10.6	10.2	5.50	10.2	10.0
	t_2	1.00	0.47	0.45	0.275	0.42	0.47
2	b_1	30.0	11.1	6.62	11.00	10.5	7.48
	t_1	1.00	0.47	0.67	0.275	0.49	0.68
	h	50.0	100.0	99.7	22.00	100.0	100.0
	t_3	1.00	0.43	0.43	0.275	0.43	0.43
	b_2	30.0	11.0	10.3	5.50	10.3	10.4
	t_2	1.00	0.46	0.44	0.285	0.49	0.48
3	b_1	30.0	5.00	5.00	11.0	5.00	5.00
	t_1	1.00	0.17	0.25	0.275	0.17	0.31
	h	50.0	25.0	39.2	22.0	25.0	25.0
	t_3	1.00	0.10	0.16	0.275	0.10	0.10
	b_2	30.0	10.0	10.0	5.50	10.0	10.0
	t_2	1.00	0.25	0.30	0.275	0.25	0.25
	Volume	270,000	83,989	86,466	26,090	83,951	84,465

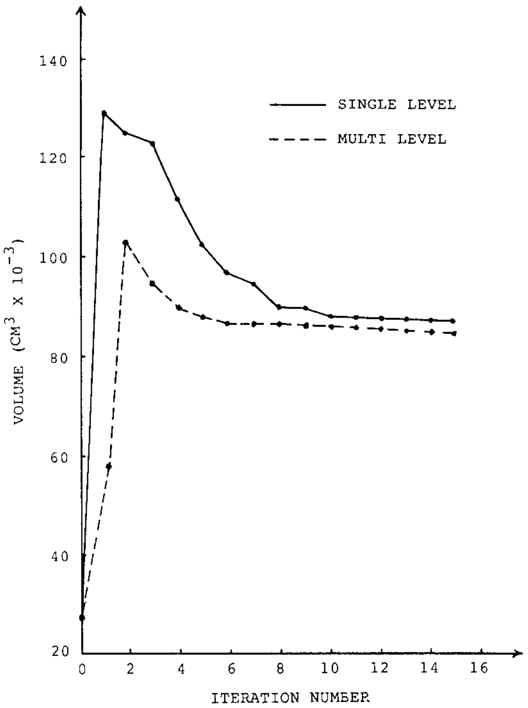

Fig. 6 Portal frame iteration history, initially infeasible.

Case 1: Two-part Beam

The beam shown in Fig. 2 was designed to provide a simple example of the method. The load $P = 1000$ lb, Young's modulus $E = 10^7$ psi, and the allowable bending stress is 20,000 psi. The allowable deflection is 0.5 in. at the beam joint and 2 in. at the tip. Because of the simplicity of this problem, move limits of 60% were used at both the system and subsystem levels.

Table 1 gives the numerical results for this example, compared to a single-level solution. In the table, the step shown as "SS" is the result after solving the subsystem optimizations and the step shown as "S" is the result after the system-level optimization. Three design cycles were performed, but the last did not change the design and so is omitted from the table. As seen from the table, the optimization makes progress on the real objective at both the system and subsystem levels. The initial increase in the objective function was needed to overcome the initial constraint violations. At the end of the first system-level optimization, all system and subsystem constraints were satisfied and remained satisfied.

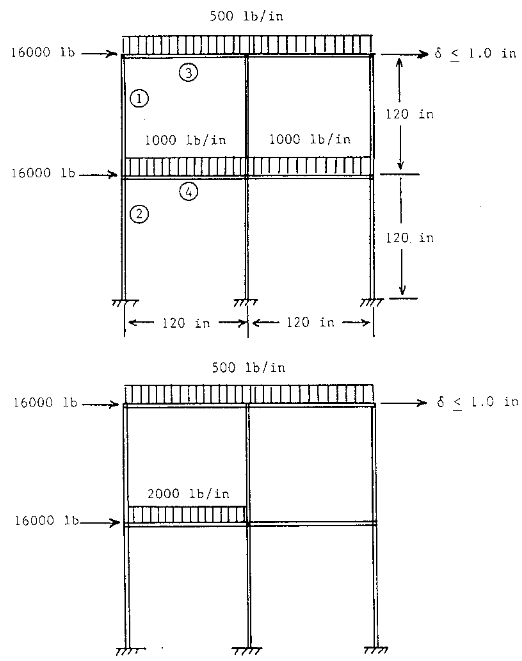

Fig. 7 Two-bay frame.

Case 2: Portal Frame

The portal frame shown in Fig. 1 was designed using the proposed method. This is considered to be a standard test case, and the details of the loads, materials, system, and subsystem

Table 3 Results for two-bay frame

Set	Parameter	Initial values	Single level	Multi-level	Initial values	Single level	Multi-level
1	b_1	5.00	5.66	2.34	2.00	3.08	3.55
	t_1	0.80	0.18	0.29	0.10	0.20	0.17
	h	15.0	22.9	23.2	10.0	19.2	20.9
	t_3	0.80	0.11	0.11	0.10	0.10	0.10
	b_2	5.00	4.46	2.45	2.00	4.95	5.40
	t_2	0.80	0.13	0.41	0.10	0.35	0.26
2	b_1	5.00	9.22	9.22	2.00	4.46	12.5
	t_1	0.80	0.39	0.45	0.10	0.52	0.33
	h	15.0	35.9	33.8	10.0	27.0	32.5
	t_3	0.80	0.16	0.15	0.10	0.14	0.15
	h_2	5.00	3.65	2.14	2.00	2.85	3.55
	t_2	0.80	0.17	0.28	0.10	0.42	0.16
3	b_1	5.00	5.39	2.09	2.00	3.88	3.66
	t_1	0.80	0.14	0.35	0.10	0.14	0.13
	h	15.0	9.44	12.1	10.0	15.8	16.4
	t_3	0.80	0.06	0.07	0.10	0.09	0.09
	b_2	5.00	5.59	2.97	2.00	4.22	2.95
	t_2	0.80	0.15	0.27	0.10	0.17	0.21
4	b_1	5.00	1.60	1.00	2.00	1.26	2.59
	t_1	0.80	0.05	0.42	0.10	0.09	0.13
	h	15.0	13.0	4.79	10.0	9.60	8.74
	t_3	0.80	0.05	0.15	0.10	0.06	0.05
	b_2	5.00	1.50	1.00	2.00	1.08	2.58
	t_2	0.80	0.05	0.43	0.10	0.09	0.13
	Volume	22,464	5,690	5,896	1,656	5,843	5,791

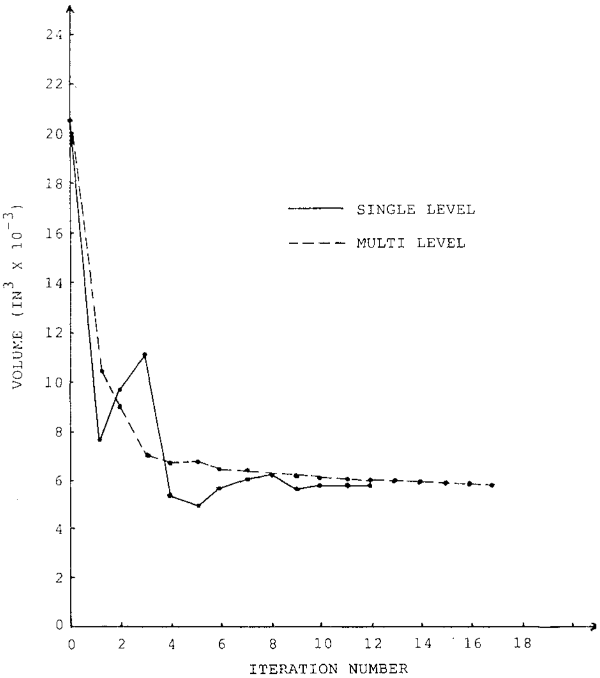

Fig. 8 Two-bay frame iteration history, initially feasible.

constraint calculations are presented in Ref. 1.

The system-level constraints are displacement and rotation limits at the joint where the loads are applied. The subsystem constraints included stress, local buckling, and sizing limits.

There are three subsystems, being the design of the individual beam elements. The subsystem design variables are the six individual dimensions of the cross section of each element.

The objective function at both the system and subsystem levels is to minimize the volume of material.

Table 2 presents results, compared to solution as a single optimization problem. Two cases were considered. In the first, the initial design was well within the feasible region, whereas in the second, the initial design was quite infeasible. Significant differences in the optimum are noted, particularly the height of the third member. This is not unexpected for this structure, which is known to possess relative minima. The iteration histories for the two cases are shown in Figs. 5 and 6. The multilevel approach did not produce as good an optimum. There is no clear reason for the differences although this structure is known to have relative minima.

Case 3: Two-bay Frame

The two-bay frame shown in Fig. 7 was designed for minimum material using the proposed method. The design variables for each beam and the subsystem constraints are the same as for case. 1. The system-level constraints as well as the loading conditions are shown in the figure. The material properties are the same as for case 1. Symmetry was used; therefore the system includes four subsystems that are the vertical members of each bay and the floor members of each bay. Each subsystem consists of six design variables for a total of 24 independent design variables.

Table 3 presents the results starting from a feasible design and from an infeasible design. The results for single-level optimization are also included. The iteration histories for the two cases are shown in Figs. 8 and 9.

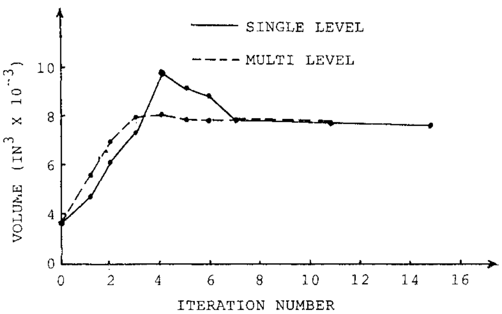

Fig. 9 Two-bay frame iteration history, initially infeasible.

Summary

An efficient multilevel optimization method is presented that is consistent with the need to develop a multilevel and multidiscipline design capability. The proposed method reduces the need for very tight move limits at the subsystem level reported earlier. However, as with any general linearization technique, reasonable move limits are still required.

The key features of the proposed method are that 1) it reduces the degree on nonlinearity to be dealt with by the optimizer, 2) it allows for the use of relatively large move limits early in the design process where major progress is made, 3) it does not require "optimum sensitivities," 4) it eliminates the need for nonlinear equality constraints at the subsystem level, and 5) it is relatively easy to understand and use in an industrial environment.

The last advantage is considered the most important one for acceptance of such methods. Here, the designer at the subsystem level must only include a set of linearized information that may be contained in a data base. Also, he must provide sensitivity information at the end of the sublevel optimization. Otherwise, he is free to formulate his problem as he chooses. The net effect is that the necessary information transfer is kept to a minimum and is well ordered.

Current work is directed toward creation of the general program environment and testing on large and more complex problems.

Acknowledgment

The research reported herein was supported by NASA Grant NAG 1567.

References

[1]Sobieszczanski-Sobieski, J., "A Linear Decomposition Method for Large Optimization Problem – Blueprint for Development," NASA TM 83,248, Feb. 1982.

[2]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,* AIAA, New York, May 1983.

[3]Vanderplaats, G. N. and Yoshida, N., "Efficient Calculation of Optimum Design Sensitivity," *AIAA Journal,* Vol. 23, Nov. 1985, pp. 1798–1803.

[4]Vanderplaats, G. N. and Cai, H.D., "Alternative Methods for Calculating Sensitivity of Optimized Designs to Problems Parameters," NASA CP 2457, 1987, pp. 19–32.

[5]Beers, M., "A Practical Multilevel Optimization Method," Master's Thesis, Department of Mechanical and Environmental Engineering, The University of California, Santa Barbara, CA, June 1986.

[6]Beers, M. and Vanderplaats, G. N., "A Linerization Method for Multilevel Optimization," *Proceedings NUMETA 87, 2nd International Conference on Advances in Numerical Methods in Engineering, Theory and Applications,* University College, Swansea, United Kingdom, July 1987.