APPLICATION OP A REDUCED QUADRATIC PROGRAMMING TECHNIQUE TO OPTIMAL STRUCTURE DESIGN

by

St«T«n J. F«nv«s, Ni«n-H»u Cbao»
& Arthur W. Westerberg

DRC-12-08-82
April, 1982

APPLICATION OF A REDUCED QUADRATIC PROGRAMMING TECHNIQUE TO OPTIMAL STRUCTURAL DESIGN

Nien-Hua Chao[1], Graduate student, Mechanical Engr. Dept.
S. J. Fenves, University Professor, Civil Engr. Dept.
A. W. Westerberg, Professor, Chemical Engr. Dept.
Carnegie-Mellon University, Pittsburgh, Pa. 15213

1. INTRODUCTION

The general optimal structural design problem is to minimize a measure of the cost of the structure subject to applicable performance requirements. In most of the literature, the cost function is taken as the weight (or volume) of the structure, the design variables are separated into element size parameters (i.e.,cross-sectional area) and structure configuration parameters (i.e., joint coordinates), and the performance requirements include behavior constraints on the stresses in each element and on the displacements of certain joints, as well as the usually implicit equality constraints arising from the joint equilibrium equations.

One popular approach for solving optimal structural design problems is to formulate them as a sequence of mathematical programming subproblems. Numerous papers ([1]-[11]) use a variety of such mathematical programming techniques.

The purpose of this paper is to introduce and illustrate a recently developed optimization technique of practical potential. The technique is based on two developments. First, it utilizes a fast successive quadratic programming algorithm originally stated by Han [12] and implemented by Powell ([13]-[20]) for solving nonlinear constrained optimization problems. The algorithm uses a Quasi-Newton method to approximate the Hessian matrix, resulting in near-quadratic convergence to at least a local optimum. Second, the technique uses the work of Berna et al. [21], who developed a decomposition procedure for Powell's algorithm which partitions the original design variables into independent and dependent variables, eliminates the dependent variables, and thus yields a much reduced quadratic programming subproblem at each step.

---

1 Current address: Bell Telephone Laboratories, Whippany, New Jersey

The technique is first derived and then illustrated using problems previously presented in structural optimization literature. The examples all deal with the simplest structural application of the technique, namely element size parameter optimization for trusses. Extensions of the technique, including additional structural types (plane frames and systems of finite elements) combined with configuration optimization, are completed [34] and will be presented in later papers.

2. DERIVATION OF OPTIMIZATION PROBLEM

2.1 GENERAL FORMULATION

The optimal structural design problem is:

(PI) Find the vector of variables j which minimizes an objective function *f(j)* subject to the constraints

$$g_i(y) = 0 \qquad i = 1,\ldots,m' \tag{1a}$$

$$h_i(y) \le 0 \qquad i = m'+1,\ldots,m \tag{1b}$$

Where $f, g_1, \ldots, g_m, h_{m+1}, \ldots, h_m$ denote real-valued functions of the vector y in N-dimensional Euclidean space R^N.

In the specific case of element size parameter optimization using an elastic finite element model, the vector j contains both the element size parameters and the joint displacements. In this case, the equality constraints (la) are given by the system equilibrium equations:

$$g = [\underline{K}]\{\underline{u}\} - \{p\} \tag{2}$$

where $\underline{K}$ = structure stiffness matrix

$\underline{u}$ = vector of joint displacements

2 = vector of applied joint loads.

The inequality constraints (lb) are of the form

$$h_i = s_{il} - s_i \text{ or } s_{iu} - s_i \tag{3}$$

where s_{il} and s_{iu} are upper and lower limits on the behavior variable s_i. The constraints pertain to element stresses and joint displacements. In the former case, the controlling stresses in element i are

$$s_i = [\, H_i \,]\ \underline{T}_i\ \underline{A}_i\ \{\ \underline{u}\ \} \tag{3a}$$

where n_i = modulus matrix

$\underline{T}_i$ = coordinate transformation matrix of element i

$\underline{A}_i$ = branch-node incidence matrix of element i

The equality constraint equations (2) are bi-linear in the variables $\underline{jr}$, as the element stiffnesses $\underline{K}_i$ entering into $[\ \underline{K}\ ]$ are linear functions of the element size parameters. The modulus matrices II_i for certain element types, eg. trusses, are independent of the element size parameters, which can be demonstrated as follows.

a) In a basic local coordinate system, the element force, r_i, is

$$r_i = [\ \underline{K}_i\ 3\ \underline{T}_i \underline{A}_i\ \{\ \underline{u}\ I = [\ Ea_i VL_i 3\ \underline{T}_i\ \underline{A}_i\ \{\ \underline{u}\ 1 \tag{4}$$

where $£$ = Young's modulus

a_i = cross-sectional area of element i

L_i = length of element i

b) The controlling bar stress, s_i, is

$$s_i = r_i/a_i = (I/a_i)\ CEa_i/L_i 3\ \underline{T}_i\ \underline{A}_i\ \{\ \underline{u}\ \} = E/L_i\ \underline{T}_i\ \underline{A}_i\{\underline{u}\} = n_i\ \underline{T}_i\ \underline{A}_i\ \{\ \underline{u}\ \} \tag{5}$$

where $II_i = E/L_i$

The inequality constraints are thus linear functions of the joint displacements $\underline{u}$, and therefore of j.

The non-linearity of equations g in the variables $\underline{jr}$ and the very large number of constraint equations makes the direct solution of (PI) infeasible or economically impractical. One step to reformulate (PI) so as to reduce the problem dimensions is to take advantage of symmetry and repetition of identical components to link together some design variables x_i as a function of distinct design variables x_j, i.e., to set

$$\{ x_i \} = [L] \{ x_j \} \tag{6}$$

$$i = 1,2,\ldots\ldots,n$$
$$j = 1,2,\ldots\ldots,n'$$

$$n' \ll n$$

It is to be noted that only the element size parameters a_i may be linked together by Equation (6); all the joint displacements must be retained as distinct design variables in $\underline{x}$. Further steps in reducing the problem dimensions will be introduced later.

The design problem now becomes:

(P2) Minimize $f(\underline{x})$

$$\text{Subject to } \underline{g}(\underline{x}) = 0 \tag{7}$$

$$\underline{h}(\underline{x}) \leq 0$$

The general form of the vector $\{ \underline{x}^T \}$ is $\{ \underline{a}^T \mid \underline{u}^T \}$

where $\{ \underline{a} \}$ = vector of element size parameters (cross-sectional areas for trusses)

$\{ \underline{u} \}$ = vector of joint displacements

The Lagrangian function of problem (P2), which will be used later, is

$$L(\underline{x}, \underline{\mu}, \underline{\xi}) = f(\underline{x}) + \underline{\mu}^T \cdot \underline{g}(\underline{x}) + \underline{\xi}^T \cdot \underline{h}(\underline{x}) \tag{8}$$

where $\underline{\mu}$, $\underline{\xi}$ are vectors of Lagrangian and Kuhn-Tucker multipliers, respectively.

2.2 QUADRATIC PROBLEM FORMULATION

Han [12] has suggested that the nonlinear optimization problem (P2) can be solved by generating a sequence of points $\{ \underline{x}^k \}$ which are the solutions to the following quadratic approximation programming subproblem:

(P3) Minimize

$$\nabla f(\underline{x}^k)^T \cdot \Delta\underline{x} + 1/2 \cdot \Delta\underline{x}^T \cdot \underline{H}_k \cdot \Delta\underline{x}$$

subject to :

$$\underline{g}(\underline{x}^k) + \nabla\underline{g}(\underline{x}^k)^T \cdot \Delta\underline{x} = 0.0 \qquad (9)$$

$$\underline{h}(\underline{x}^k) + \nabla\underline{h}(\underline{x}^k)^T \cdot \Delta\underline{x} \geq 0.0$$

where $\Delta\underline{x} = \underline{x}^{k+1} - \underline{x}^k$

$$\nabla f = \partial f / \partial \underline{x}$$

$$\nabla \underline{g} = \partial \underline{g} / \partial \underline{x}$$

$$\nabla \underline{h} = \partial \underline{h} / \partial \underline{x}$$

The n x n matrix $\underline{H}_k$ is intended to be an approximation of the Hessian matrix of the Lagrangian function of problem (P2). In Han's original work, the Hessian matrix is updated by the Davidon-Fletcher-Powell [22] method. Powell has used Han's method to solve optimization problems with nonlinear constraints [13]-[20] with the Hessian matrix approximated by a Quasi-Newton method. Powell suggests an empirical rule so that the updated Hessian matrix remains positive definite or positive semi-definite. A Quasi-Newton method which was simultaneously presented by Broyden, Fletcher, Goldfarb and Shanno (BFGS) [22] has been used in his study. Numerical results have proven that the efficiency of Han's method can be improved by Powell's modifications [13].

2.3 REDUCED QUADRATIC PROBLEM FORMULATION

Berna et al. have suggested a decomposition procedure whereby the optimization problem (P3) can be solved more efficiently [21]. In Berna's work, the design variables { $\Delta\underline{x}$ } are partitioned into two subvectors, the vector { $\Delta\underline{a}$ } of independent variables and the vector {$\Delta\underline{u}$} of dependent variables.

The necessary conditions for solving the quadratic approximation problem (P3) are:

(1) Stationary condition of the Lagrangian function of the quadratic approximation problem (P3):

$$[H]\begin{Bmatrix}\Delta a\\ \Delta u\end{Bmatrix} + \begin{Bmatrix}\partial g^T/\partial a\\ \partial g^T/\partial u\end{Bmatrix}\{\mu\} + \begin{Bmatrix}\partial h^T/\partial a\\ \partial h^T/\partial u\end{Bmatrix}\{\eta\} = -\begin{Bmatrix}\partial f/\partial a\\ \partial f/\partial u\end{Bmatrix} \quad \text{aw}$$

(2) Satisfaction of the linearized original constraint equations:

$$[\partial g/\partial a^T]\{\Delta a\} + [\partial g/\partial u^T]\{\Delta u\} = -\{g\}$$

$$[\partial h/\partial a^T]\{\Delta a\} + [\partial h/\partial u^T]\{\Delta u\} \leq -\{h\} \quad (11)$$

(3) Complementary slackness and nonnegativity of the Kuhn-Tucker multipliers:

$$\{\eta\}^T\{\partial h/\partial a^T \cdot \Delta a + \partial h/\partial u^T \cdot \Delta u + h\} = 0.0$$

$$\{\eta\} \geq \{0.0\} \quad (12)$$

It is easier to present the remainder of the derivation if the following notation is adopted :

$$
\begin{array}{ll}
H_{AA} = \partial^2 L/\partial a \partial a^T & H_{Au} = \partial^2 L/\partial a \partial u^T \\
H_{UA} = \partial^2 L/\partial u \partial a^T & H_{uu} = \partial^2 L/\partial u \partial u^T \\
G_A = \partial g/\partial a^T & G_u = \partial g/\partial u^T \\
h_A = \partial h/\partial a^T & h_u = \partial h/\partial u^T \\
f_A = \partial f/\partial a & f_u = \partial f/\partial u
\end{array}
$$

Using the above notation, the necessary conditions given by Equations (10) through (12) are (rows and columns of the coefficient matrix are numbered for later reference):

$$
\begin{array}{c}
\begin{array}{cc}
 & \begin{array}{cccc} (1) & (2) & (3) & (4) \end{array} \\
\begin{array}{c} (1) \\ (2) \\ (3) \\ (4) \end{array} &
\begin{bmatrix}
H_{AA} & H_{Au} & G_A^T & h_A^T \\
H_{uA} & H_{uu} & G_u^T & h_u^T \\
G_A & G_u & 0 & 0 \\
h_A & h_u & 0 & 0
\end{bmatrix}
\end{array}
\begin{Bmatrix} \Delta a \\ \Delta u \\ \mu \\ \eta \end{Bmatrix}
\begin{array}{c} = \\ = \\ = \\ \leq \end{array}
\begin{Bmatrix} -f_A \\ -f_u \\ -g \\ -h \end{Bmatrix} \\[2ex]
\eta^T \begin{bmatrix} h_A & h_u & I \end{bmatrix} \begin{Bmatrix} \Delta a \\ \Delta u \\ h \end{Bmatrix} = 0 \\[2ex]
\eta \geq 0
\end{array}
\qquad (13)
$$

The size of the coefficient matrix in Equation (13) is extremely large. The major contribution by Berna et al. is the procedure for eliminating the dependent variables {u} efficiently, resulting in a much reduced quadratic programming problem. The reduction is accomplished in two steps.

The rows and columns in Equation (13) are first rearranged as shown:

$$
\begin{array}{cc}
 & \begin{array}{cccc} (2) & (3) & (1) & (4) \end{array} \\
\begin{array}{c} (3) \\ (2) \\ (1) \\ (4) \end{array} &
\begin{bmatrix}
G_u & 0 & G_A & 0 \\
H_{uu} & G_u^T & H_{uA} & h_u^T \\
H_{Au} & G_A^T & H_{AA} & h_A^T \\
h_u & 0 & h_A & 0
\end{bmatrix}
\end{array}
\begin{Bmatrix} \Delta u \\ \mu \\ \Delta a \\ \eta \end{Bmatrix}
\begin{matrix} = \\ = \\ = \\ \leq \end{matrix}
\begin{Bmatrix} -g \\ -f_u \\ -f_A \\ -h \end{Bmatrix}
\qquad (14)
$$

$$
\eta^T \begin{bmatrix} h_A & h_u & I \end{bmatrix} \begin{Bmatrix} \Delta a \\ \Delta u \\ h \end{Bmatrix} = 0
$$

$$
\eta \geq 0
$$

Next a reduction or condensation is performed on the first two matrix columns of Equation (14), producing:

$$
\begin{array}{cc}
 & \begin{array}{cccc} (2) & (3) & (1) & (4) \end{array} \\
\begin{array}{c} (3) \\ (2) \\ (1) \\ (4) \end{array} &
\begin{bmatrix}
I & 0 & M_1 & 0 \\
0 & I & M_2 & N_1 \\
0 & 0 & \hat{H} & Q \\
0 & 0 & Q^T & 0
\end{bmatrix}
\end{array}
\begin{Bmatrix} \Delta u \\ \mu \\ \Delta a \\ \eta \end{Bmatrix}
\begin{matrix} = \\ = \\ = \\ \leq \end{matrix}
\begin{Bmatrix} m_1 \\ m_2 \\ q \\ \hat{h} \end{Bmatrix}
\qquad (15)
$$

The terms appearing in Equation (IS) are defined in Appendix A.

The conditions that the quadratic programming problem must satisfy are then given by the last two rows of Equation (15):

$$\begin{bmatrix} \hat{H} & Q \\ Q^T & 0 \end{bmatrix} \begin{Bmatrix} \Delta a \\ \gamma \end{Bmatrix} \begin{matrix} = \\ \le \end{matrix} \begin{Bmatrix} q \\ \hat{h} \end{Bmatrix} \tag{16}$$

The original complementary slackness conditions. Equation (12), become

$$*^T \cdot (Q^T \cdot Aa - \hat{h}) = 0$$

Again, nonnegativity of the Kuhn-Tucker multipliers in Equation (12) gives 9 *t* 0

The reduced QPP is, therefore:

(P4) Minimize

$$f(Aa) = q^T * Aa + 1/2 \cdot Aa^T * \hat{H} \cdot Aa \tag{17}$$

subject to

$$Q^T \cdot Aa \le \hat{h}$$

The corresponding Lagrangian function is

$$V (Aa , rj) = q^T * Aa + 1/2 \cdot Aa^T \cdot \hat{H} \cdot Aa +*^T * (Q^T \cdot Aa - \hat{h}) \tag{18}$$

The optimization design problem (P4) can now be solved in terms of the independent design variables { Aa }. The results,{ Aa } and { 7 }, of problem (P4) can then be used to calculate the vectors { Au } and { /* } in the first two rows of the Equation (15), thus completing the solution of the quadratic approximation subproblem.

3. MODIFICATIONS OF THE TECHNIQUE

3.1 ACTIVE CONSTRAINTS

In applying the reduced quadratic programming technique presented above to optimal structural design problems, the number of constraint equations $Q^T * Aa \le \hat{h}$ may be 10 to 100 times larger than the number of independent design variables { Aa }. To further reduce the problem dimensions, in each iteration only the critical and potentially critical constraint equations are included, so that only about 5% to 30% of the original number of constraint equations is used in each iteration.

3.2 CONTROLLING STEP SIZE

In optimal structural design problems which include both stress and displacement constraints, the optimal solution is usually found at an interior point of the design space such that the total number of active constraint equations is less than the total number of independent design variables. In such situations, Powell's algorithm [13] may not converge.

To control the step size of the independent design variables and to stabilize the algorithm, a constraint of the form

$$1/2 \ \{ \Delta a^T \} * \{ \Delta a \} \leq \epsilon \tag{19}$$

may be added to the problem (P4).

Adding constraint (19) to the problem results in adding a diagonal matrix $\pi * [I]$ to the Hessian of the reduced quadratic problem, where π (≥ 0.0) is the Kuhn–Tucker multiplier for the above constraint. Rather than chosing ϵ, we can treat π as an adjustable parameter, to be increased if we wish to reduce the step size and decreased if we wish to allow for larger steps. A minimum value of zero for π releases the step size constraint completely.

No automatic adjustment algorithm for π has been developed to date, but one similar to that used by Reid [23] and Westerberg [24] could be devised. Such an algorithm would increase π if the actual change in the Lagrangian function for the step taken is significantly different from the value predicted by the linearized Lagrangian function, hold π fixed if the linearization is moderately acceptable, and decrease π (to zero perhaps) if the linearization is excellent. The addition of a diagonal matrix to the approximated Hessian matrix H_k also stabilizes the algorithm. The updating procedure assures that the Hessian matrix H_k remains symmetric and positive definite (see Appendix C), but in the limit it may make H_k nearly singular (positive semi-definite). We have on occasion found it effective to have a small diagonal term in the approximated Hessian matrix to control the conditioning.

3.3 INCONSISTENT CONSTRAINTS

In solving the reduced quadratic programming problem (P4), inconsistent constraints may exist among the set of linearized constraint equations ($Q^T \cdot \Delta a \le h^{**}$). This inconsistency arises from using the linearized constraints to substitute for the original constraints, when the initial guess or an intermediate solution is too far from the optimum solution. Powell [13] introduced a dummy variable ξ ($0.0 \le \xi \le 1.0$) into the quadratic programming problem to solve this problem.

By adding the variable ξ to the reduced QPP , Equation (16) becomes:

$$\begin{bmatrix} \hat{H} & 0 & Q_i & Q_j \\ 0 & 1 & \hat{h}_i^T & 0 \\ Q_i^T & \hat{h}_i & 0 & 0 \\ Q_j^T & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta a \\ \xi \\ \eta_i \\ \eta_j \end{Bmatrix} \begin{matrix} = \\ = \\ \le \\ \le \end{matrix} \begin{Bmatrix} q \\ C \\ \hat{h}_i \\ \hat{h}_j \end{Bmatrix} \tag{20}$$

where C = large negative constant

$\hat{h}_i < 0.0$

% * CO

and a feasible solution can always be found for Equation (20).

4. IMPLEMENTATION

The complete algorithm for applying the reduced quadratic programming technique to optimal structural design is described in Appendix E A modular software system is being developed to implement the technique. The main feature of the system is that only the subroutines for generating the matrices k_i, T_i and Π_i and their derivatives need to be compiled with the system for each different structural type.

5. EXAMPLES

Three truss examples previously reported in the literature have been chosen to test the accuracy and efficiency of the program.

Example 1 is a ten bar planar cantilever truss shown in Figure 1, previously studied by Schmit [3], Khan [25] and others. Design and loading data are given in Table 1.

Example 2 is a twenty-five bar transmission tower truss, shown in Figure 2. previously studied by Schmit [3], Arora [4] and others. The design data and the two loading conditions applied are given in Table *Z* The elements are linked into eight groups as in Reference [3].

Example 3 is a seventy-two bar space truss, shown in Figure 3, previously studied by Schmit L33. Arora [4] and others. Design and loading data are given in Table 3. The elements are linked into sixteen groups as in Reference [3].

For each of the three examples, two cases were investigated: Case l_f using element stress constraints only and Case 2, with displacement constraints included.

6. RESULTS

The results obtained for the three examples are shown in Tables 4 through 6, showing the final areas, the total weight, and the number of cycles to convergence. The optimal results for Case 2 of the three examples are compared to published results in Tables 7 through 9.

It can be seen from the tables that the technique presented converges in all cases to the same optimum point as the previous studies. It can also be seen that with one exception (Venkayya [1] on Example 2), the present technique requires fewer cycles to converge to an approximate optimal point[2] than the fastest of the previous methods.

7. CONCLUSION

A fast optimization technique for optimal structural design has been presented. The speed of the technique derives from two key factors: first, the dependent design variables are eliminated or condensed out of the quadratic approximation subproblems, and second, near-quadratic convergence for the independent design variables is obtained. The technique appears to be particularly attractive for large-scale optimal structural design problems, since all joint displacements (a subvector of length equal to the number of degrees of freedom times the number of loading conditions) are eliminated, resulting in a reduced quadratic programming subproblem involving only the distinct element sizing parameters as design variables.

---

[2] Defined as $\max |q_i| / \max |P_i| \leq 10^{-5}$

Results obtained with the technique for a number of standard test problems are in agreement with previous results and show a general reduction in the number of cycles to convergence.

An interesting consequence of the optimization technique described is that the solution of each quadratic approximation subproblem may not be a feasible one unless the optimum is reached. Specifically, a trial solution given in terms of the current variables { a } and { u } is not in equilibrium with the applied joint loads{p}«

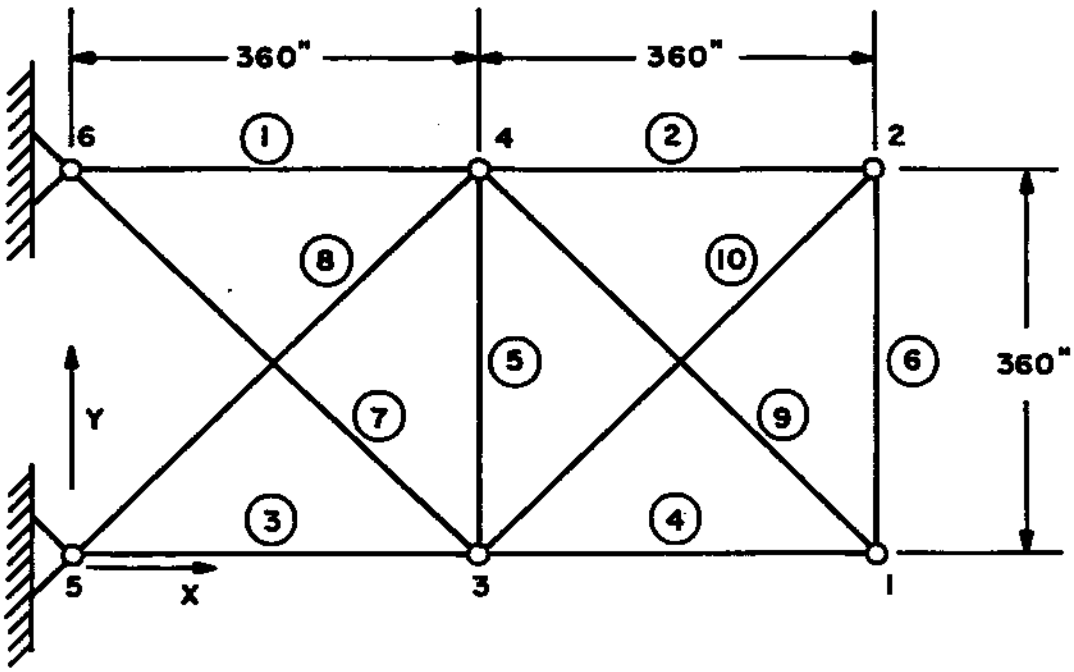

Figure 1. Ten bar cantilever truss

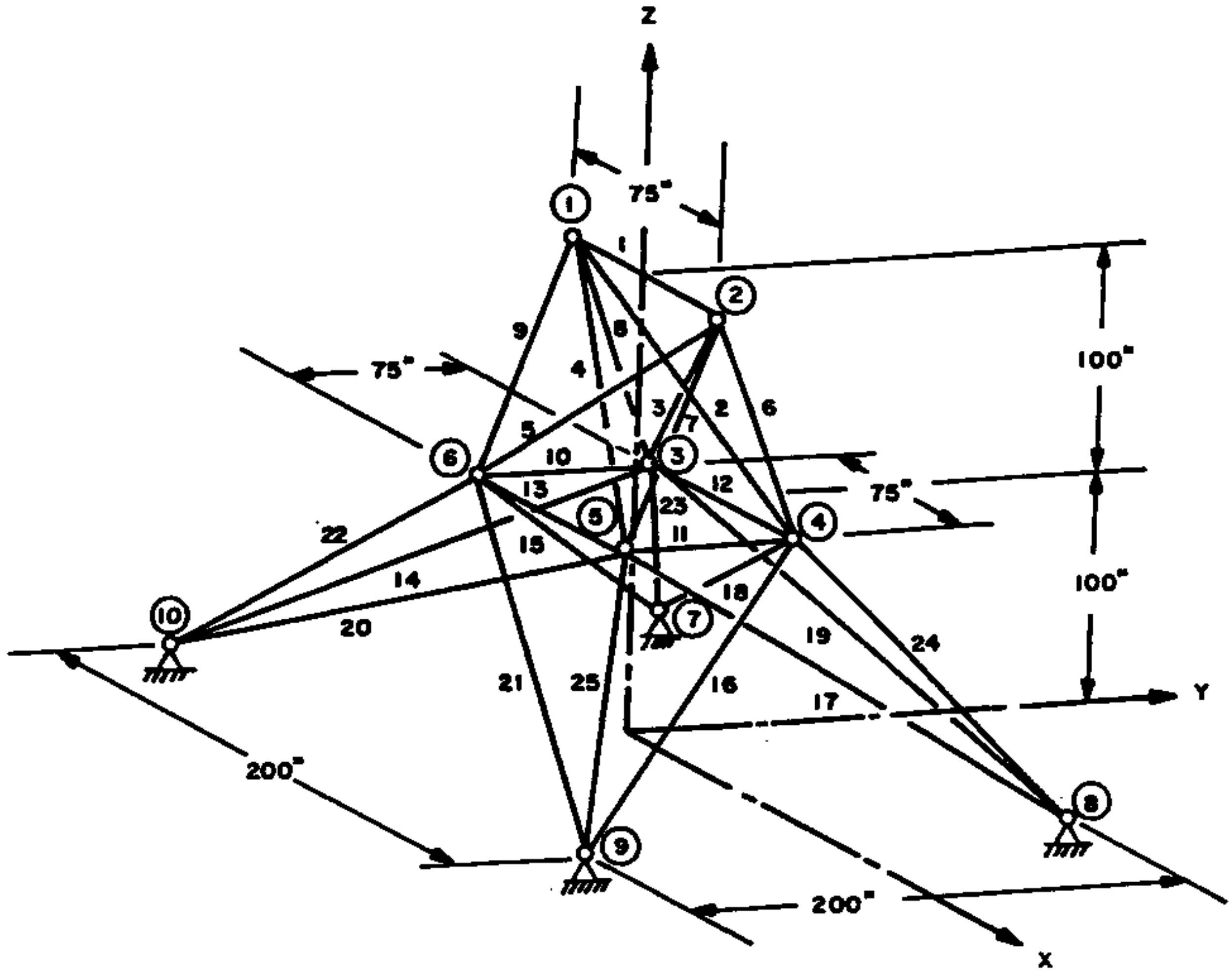

Figure 2. Twenty-five bar transmission tower truss

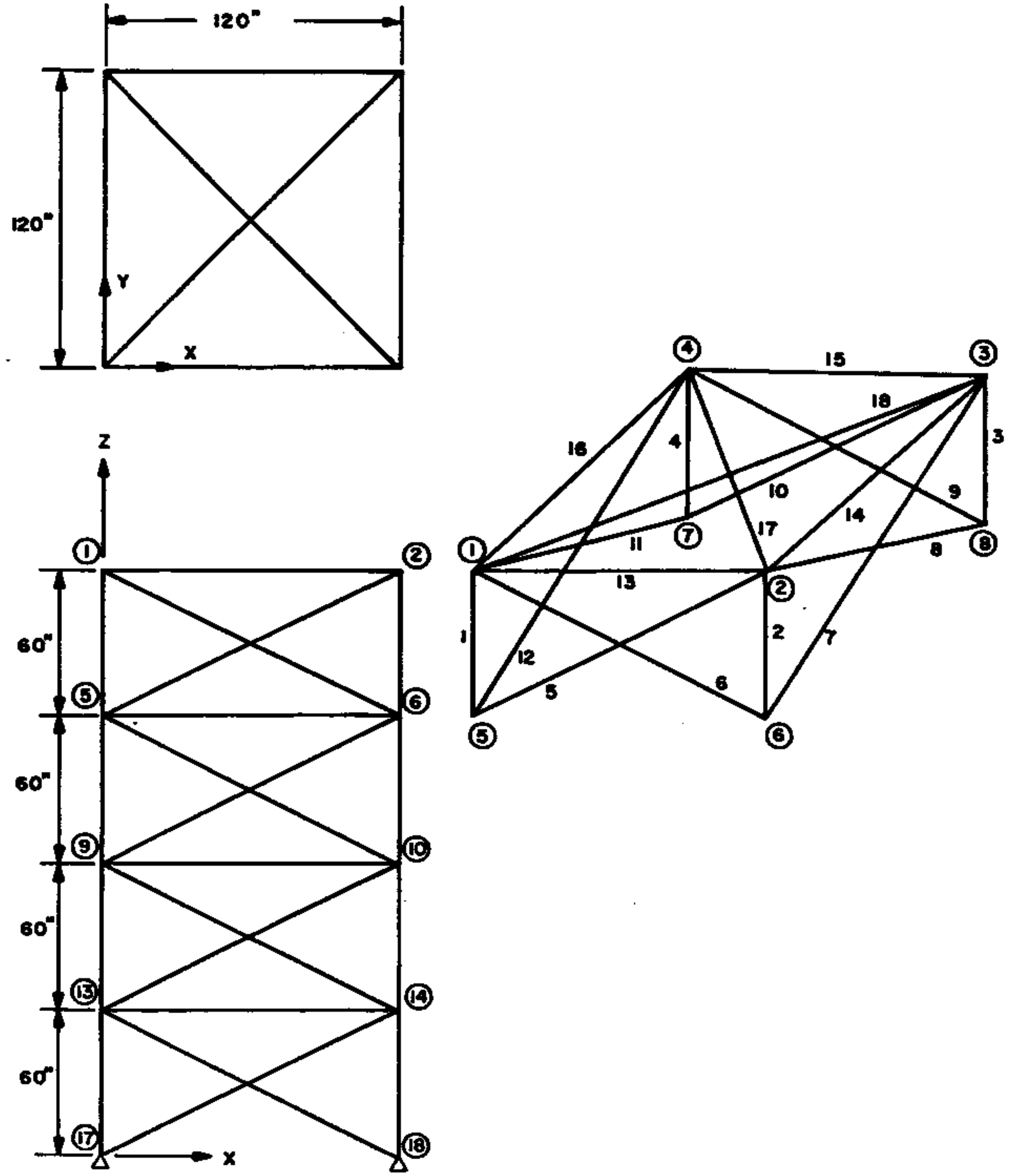

Figure 3. Seventy-two bar space truss

Table 1. Design datai for 10 bar truss

Modulus of elasticity	=	10^4 ksi
Material density	=	0.10 lb/in^3
Stress limits	=	± 25 fcsi
Lower limit on cross-sectional area	=	0.10 in^2
Upper limits on displacements	=	2.0 in
Number of loading conditions	=	1

Node	Magnitude of load (Kips) X	y
1	0.0	-10.0
3	0.0	-10.0

Table 2. Design data for 25 bar transmission tower

Modulus of elasticity	=	10^4 fcsi
Material density	=	0.10 lb/in^3
Stress limits	=	± 40 ksi
Lower limit on cross-sectional area	=	0.10 in^2 *
		0.01 in^2 **
Upper limits on displacements	=	0.35 in
Number of loading conditions	=	2

		Magnitude of load (Kips)		
Loading condition	Node	X	y	z
	1	-1.0	-10.0	-5.0
1	2	0.0	-10.0	-5.0
	3	-0.5	0.0	0.0
	4	-0.5	0.0	0.0
	1	0.0	-20.0	-5.0
2	2	0.0	20.0	-5.0

* Stress constraints only

** Stress and displacement constraints

Table 3. Design data for 72 bar truss

Modulus of elasticity	= 10^4 Ksi
Material density	= 0.10 lb/in^3
Stress limits	= ± 25 ksi
Lower limit on cross-sectional area	= 0.10 in^2
Upper limits on displacements	= 0.25 in
Number of loading conditions	= 2

Loading condition	Node	Magnitude of load (Kips) X	Y	Z
1	1	5.0	5.0	-5.0
2	1	0.0	0-0	-5.0
	2	0.0	0.0	-5.0
	3	0.0	0.0	-5.0
	4	0.0	0.0	-5.0

Table 4. Optimum 10 bar truss

Member number	Final area (in^2) case 1	case 2
1	7.9379	30.7928
2	0.1000	0.1000
3	8.0621	23.9655
4	3.9379	14.7038
5	0.1000	0.1000
6	0.1000	0.1000
7	5.7447	8.5321
8	5.5690	20.9519
9	5.5690	20.8014
10	0.1000	0.1000
Final weight (lb)	1593.18	5076.64
Number of iterations	3	10

Table 5. Optimum 25 bar transmission tower

Group No.	Member numbers	Final area (in^2) case 1	Final area (in^2) case 2
1	1	0.1000	0.0100
2	2 3 4 5	0.3761	2.0415
3	6 7 8 9	0.4709	3.0011
4	10 11	0.1000	0.0100
5	12 13	0.1000	0.0100
6	14 15 16 17	0.1000	0.6836
7	18 19 20 21	0.2773	1.6248
8	22 23 24 25	0.3801	2.6716
Final weight (lb)		91.13	545.03
Number of iterations		3	8

Table 6. Optimum 72 bar truss

Group No.	Member numbers	Final area (in^2) case 1	Final area (in^2) case 2
1	1 2 3 4	0.1888	0.1565
2	5 6 7 8 9 10 11 12	0.1000	0.5493
3	13 14 15 16	0.1000	0.4061
4	17 18	0.1000	0.5550
5	19 20 21 22	0.1904	0.5127
6	23 24 25 26 27 28 29 30	0.1000	0.5289
7	31 32 33 34	0.1000	0.1000
8	35 36	0.1000	0.1000
9	37 38 39 40	0.1987	1.2521
10	41 42 43 44 45 46 47 48	0.1000	0.5214
11	49 50 51 52	0.1000	0.1000
12	53 54	0.1000	0.1000
13	55 56 57 58	0.2941	1.8321
14	59 60 61 62 63 64 65 66	0.1000	0.5119
15	67 68 69 70	0.1000	0.1000
16	71 72	0.1000	0.1000
Final weight (lb)		96.637	379.62
Number of iterations		3	8

Table 7 Optimum designs for ten-bar truss

Group No.	Schmit & NEWSUMT	Kiura(3) CONMIN	Schmit & Farshi(2)	Venfcayya(1)	Arora & Haug(4)	Khan & Willmert(25)	This paper
	Final area (in^2)						
1	30.6700	30.5700	33.4320	30.4160	***	30.9800	30.7928
2	0.1000	0.3690	0.1000	0.1280	***	0.1000	0.1000
3	23.7600	23.9700	24.2600	23.4080	***	24.1690	23.9655
4	14.5900	14.7300	14.2600	14.9040	***	14.8050	14.7038
5	0.1000	0.1000	0.1000	0.1010	***	0.1000	0.1000
6	0.1000	0.3640	0.1000	0.1010	***	0.4060	0.1000
7	8.5780	8.5470	8.3880	8.6960	***	7.5470	8.5321
8	21.0700	21.1100	20.7400	21.0840	***	21.0460	20.9519
9	20.9600	20.7700	19.6900	21.0770	***	20.9370	20.8014
10	0.1000	0.3200	0.1000	0.1860	***	0.1000	0.1000
Final weight(ID)	5076.85	5107.30	5089.00	5084.90	***	5066.98	5076.64
Number of iterations	13	14	23	25	***	18	10

*** data not available

Table 8 Optimum designs for 25-bar transmission tower

Group No.	Schmit & NEWSUMT	Miura(3) CONMIN	Schmit & Farshi(2)	Venkayya(l)	Arora & Haug(4)	Khan & Willmert(25)	This paper
	Final area (in^2)						
1	0.0100	0.1660	0.0100	0.0280	0.0100	0.0100	0.0100
2	1.9850	2.0170	1.9640	1.9420	2.0476	1.7550	2.0415
3	2.9960	3.0260	3.0330	3.0810	2.9965	2.8690	3.0011
4	0.0100*	0.0870	0.0100	0.0100	0.0100	0.0100	0.0100
5	0.0100	0.0970	0.0100	0.0100	0.0100	0.0100	0.0100
6	0.6840	0.6750	0.6700	0.6930	0.6853	0.8450	0.6836
7	1.6770	1.6360	1.6800	1.6780	1.6217	2.0110	1.6248
8	2.6620	2.6690	2.6700	2.6270	2.6712	2.4780	2.6716
Final weight(lb)	545.17	548.47	545.22	545.49	545.04	553.94	545.03
Number of iterations	10	9	17	7	12	9	8

Table 9 Optimum designs for 72-bar truss

Group No.	Final area (in^2)						
	Schmit & NEWSUHT	Miura(3) CONMIN	Schmit & Farshi(2)	Venfcayya(l)	Arora & Haug(4)	Khan & Willmert (25)	This paper
1	0.1565	0.1558	0.1580	0.1610	0.1564	0.1519	0.1565
2	0.5458	0.5484	0.5940	0.5570	0.5464	0.5614	0.5493
3	0.4105	0.4105	0.3410	0.3770	0.4110	0.4378	0.4061
4	0.5699	0.5614	0.6080	0.5060	0.5712	0.5317	0.5550
5	0.5233"	0.5228	0.2640	0.6110	0.5263	0.5814	0.5127
6	0.5173	0.5161	0.5480	0.5320	0.5178	0.5273	0.5289
7	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
8	0.1000	0.1133	0.1510	0.1000	0.1000	0.1583	0.1000
9	1.2670	1.2680	1.1070	1.2460	1.2702	1.2526	1.2521
10	0.5118	0.5111	0.5790	0.5240	0.5124	0.5244	0.5241
11	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
12	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
13	1.8850	1.8850	2.0780	1.8180	1.8656	1.8589	1.8321
14	0.5125	0.5118	0.5030	0.5240	0.5131	0.5259	0.5119
15	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
16	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
Final weight(lb)	379.64	379.79	388.63	381.20	379.62	387.67	379.62
Number of iterations	9	8	22	12	12	10	8

APPENDIX A. TERMS APPEARING IN EQUATION (15)

The terms appearing in the condensed Equation (15) are:

Define:

$G_1 \bullet « ^\circ u)^{-1}$

$G_2 = (G_u^T)^{-1} = G_1^T$

$^C uuu \; " \; ^G 2 \; ' \; ^H «u \; ' \; ^C I$

Then:

$^M 1 \bullet ^\circ 1 \cdot ^G A$

$^M 2 - " ^G 2 * ^H UA" \; ^C uuu * ^G A$

$^N 1 = ^G 2 * ^h u$

$\hat{}$

$» " \; ^H AA " \; ^H Au * ^G l * ^G A " \; ^G A^T * ^G 2 * ^H UA \; ^+ \wedge * \; ^C uuu * ^G A$

$Q^T = ^h A " \; ^h u * ^G 1 * ^G A$

$* \; ^= " \; ^f A \; ^+ \; ^H Au * ^G l *8 - G_A^T . G_2 . C_{uuu} * g * G_A^T * G_2 * f_u$

$h = -h + h_u * G_1 - * g$

$^m 1 \; ^= " \; ^G 1 * \; ^8$

$"2 \; ^= " \; ^G 2 * \; ^F u " \; ^C uuu * \; ^g$

APPENDIX B. ALGORITHM

The procedure for performing optimal design is described by the following algorithm.

Step 0 Initialization

i) Set k (iteration index) = 0 and the Hessian matrix $H_k = [I]$ (results in the first step in the steepest descent direction),

ii) Initialize vector { a } and solve Equation (2) for vector { u }.

Step 1 Compute right hand side and derivatives for approximation problem (Equations (10) - (12)).

i) **k** = **k** • 1

ii) Compute matrices G_A and G_u and vector g

iii) Compute vectors f_A and f_u

iv) Compute matrices h_A and h_u and vector h

v) If k = 1 go to Step 2

vi) Compute

$$\left\{\frac{\partial L}{\partial X}\right\} = \left\{\begin{matrix} f_A \\ f_u \end{matrix}\right\} + \begin{bmatrix} G_A^T \\ G_u^T \end{bmatrix} \mu + \begin{bmatrix} h_A^T \\ h_u^T \end{bmatrix} \eta$$

vii) Compute

$$\left\{\begin{matrix} W_A \\ W_u \end{matrix}\right\}_{11*1} = b + \sum_{j}^{n} (-1)^{j} 2 \left\{\begin{matrix} W_A \\ W_u \end{matrix}\right\}_j \quad a_1 / \sqrt{\sigma}$$

$$\left\{\begin{matrix} \iota \\ W_u \end{matrix}\right\}_{n+2} = \frac{\zeta}{(\delta^T \zeta)^{1/2}}$$

Where n = (k - 1) • 2

$$\gamma = \{\partial L/\partial X\}_2 - \{\partial L/\partial X\}_1$$

$$\sigma_j = [\, W_A^T \, W_u^T \,]_j \, \delta \qquad j = 1 \ldots n$$

$$\sigma = \sum_{j-J} <-l)^J * «r/_J + {}^T \delta$$

$$e = \begin{cases} L & \text{if } 5^T{}_r \geq 0.2\ a \\ a - 8^T y & \text{otherwise} \end{cases}$$

$$C = By + (L - 0)\ \& + (1. - 8)\sum_{j=1}^{n}(-1)^j \begin{Bmatrix} W_{A_j} \\ W_{u_j} \end{Bmatrix} \sigma_j$$

Step 2 Reduction (Condensation) **and setup** of Quadratic Programming **problem**

i) Compute matrices H and Q_A (see Appendix A)

ii) Compute vectors q and h (see Appendix A)

iii) Select critical and potentially critical constraints for problem (P4)

Step 3 Optimization

i) Solve problem (P4) for vectors { Aa } , { *j)* }

ii) Backsubstitute in the first two rows of Equation (IS) for vectors

{ Au } and { *n* }

iii) Compute

$$\left\{\frac{\partial L}{\partial x}\right\} = \begin{Bmatrix} f_A \\ f_u \end{Bmatrix} + \begin{bmatrix} G_A{}^T \\ G_u{}^T \end{bmatrix} \mu + \begin{bmatrix} h_A{}^T \\ h_u{}^T \end{bmatrix} \eta$$

Step 4 Determine step size parameter

i) For k = 1

$v_i = 0.0$ $\qquad$ i = 1,......m

ii) For i = 1,__,m'

$v_i = \mathrm{Max}\ C\,|/ij\,1, 0.5*(v_i + |\mu_i|\ )\}$

iii) For i = nf+l...,m

$v_i = \max\{\ ?_i\ ,\ 0.5 * (\ v_i + y\ )\ \}$

iv) Select the largest value of *a*. O.OSaS 1.0,

a) If ¥(A.ul») > $*_k$ go to b)

al) If *V(ZSL»)* < *WA.U.V)* or

uX.uC/1.9) < UA.u./i.f) go to vi)

b) If *Wi\£*)* < WA,u,») go to vi)

c) Go to v) of Step 5

where V(A,u,») = F(A,u) + Z» |g-(A,u)j + Z» Max{ 0., $h_{\{}$ }

L(X, /». $_n$) = F(X) • /i^T • g(X) +i7^T • h(X)

V_k = Min{ ¥(A ,ı *Jh ; j = 0,1,.......,k−1}

$\tilde{A} = A + \alpha\,\Delta A\ ,\ \tilde{u} = u + \alpha\,\Delta u$

vi) Set A = $\tilde{A}$, u = $\tilde{u}$ **and** $\delta = \alpha \begin{Bmatrix} \Delta a \\ \Delta u \end{Bmatrix}$

Step 5 Check **for** convergence

i) Let f = S^T $\begin{Bmatrix} \\ f_u \end{Bmatrix}$ A (+ . J ». gj(A,u)

ii) If f £ € print result and go to Step 6

iii) Adjust the step controlling parameter *n*

iv) If k < maximum number of allowed iterations, go to Step 1

v) Print error message and go to Step 6

Step 6 Stop.

APPENDIX C UPDATING OF THE HESSIAN MATRIX

In Powell's work ([13]-[20]), the Hessian matrix was updated by the BFGS rank 2 method In that updating method, the Hessian matrix H is initially set equal to an identity matrix, and in each iteration on the quadratic approximation subproblem the Hessian matrix is updated by the following formula:

$$H_{k+1} = H_k + \frac{y\,y^T}{< y , s >} - \frac{H_k\, s\, s^T H_k}{< s , H_k s >}$$

where $y = L (x^{k+1}) - L (x^k)$

$s = x^{k+1} - x^k$

$< y , s > = y^T s$

Instead of keeping the full matrix H_k, Beraa suggested the following expression to update the Hessian matrix:

$$\begin{bmatrix} H_{AA} & H_{Au} \\ H_{uA} & H_{uu} \end{bmatrix} = [I] + \sum_{j=1}^{n} (-1)^j \begin{Bmatrix} W_{A_j} \\ W_{u_j} \end{Bmatrix} [\, W_{A_j}^T \;\; W_{u_j}^T \,]$$

where W_{A_j} and W_{u_j} are defined in Appendix B and n = (k - l) * 2

The following formula, used by Berna, is implemented in the present method to update he reduced Hessian matrix in each iteration . By using this formula, the number of arithmetic operations performed in updating the reduced Hessian matrix is reduced dramatically.

$$\hat{H} - H_{AA} - H_{AU} * G_3 " G_3^T * H_{UA} + G_3^T * H_{uu} * G_3$$

$$= (I_f + Z W_{A_j} \cdot W_{A_j}^T) - (Z W_{A_j} \cdot W_{u_j}^T) * G_3 - G_3^T * (Z W_{u_j} \cdot W_{A_j}^T)$$

$$* G_3^T \cdot (I_z * Z W_{u_j} * W_{u_j}^T) * G_3$$

$$= I_r * G_3^T * G_3 \cdot Z (W_{A_j} - G_3^T \cdot W_{u_j}) * (W_{A_j} - G_3^T \cdot W_{u_j})^T$$

where $G_1 = (G_U)^{-1}$ and $G_3 = G_X \cdot G_A$

r = number of independent variables

z = number of dependent variables.

REFERENCES

I. V. B. Venkayya, "Design of Optimum Structures,** *Computer & Structures,* VoL 1, 1971, pp. 265-309.

2 L. A. Schmit and B. Farshi, "Some Approximation Concepts for Structural Synthesis,[9]* *AIAA Journal,* VoL 12, No. 5, 1974, pp. 692-*99.

3. L. A. Schmit and H. Miura. **Approximation Concepts for Efficient Structural Synthesis,** Tech. report CR-2552, NASA, 1976.

4. Arora, J. S. and Haug, E. J. Jr., "Efficient optimal design of structures by generalized steepest descent programming,** *Int. J. Num. Meth. Engng,* VoL 10, 1976, pp. 747-766.

5. Arora, J. S. and Govil, A. K., **An efficient method for optimal structural design by substructuring," *Computers & Structures,* VoL 7, 1977, pp. 507-515.

6. Govil, A. K. Arora, J. S. and Haug, E. J., "Optimal design of wing structures with substructures,** *Computers & Structures,* VoL 10, 1979, pp. 899-910.

7. Arora. J. S. and Haug, E. J., "Methods of design sensitivity analysis in structural optimization," *AIAA Journal,* Vol. 17, 1979, pp. 970.

8. R. T. Haftka & J. H. Starnes Jr., "Application of a Quadratic Extended Interior Penalty Function for Structural Optimization,** *AIAA Journal,* VoL 14, No. 6, 1976, pp. 718-724.

9. R. T. Haftka & R Prasad. "Programs for Analysis and Resizing of Complex Structures," *Computers & Structures,* VoL 10, 1979, pp. 323-330.

10. B. Prasad & R. T. Haftka, "A Cubic Extended Interior Penalty Function for Structural Optimization,** *Int. J. for Numerical Methods in Engineering,* VoL 14, 1979, pp. 1107-1126.

II. G. N. Vanderplaats & F. Moses, "Structural Optimization by Methods of Feasible Directions,*' *Computers & Structures,* VoL 3, 1973, pp. 739-755.

12. S. P. Han, "A Globally Convergent Method for Nonlinear Programming." *Journal of Optimization Theory and Applications,* VoL 22, No. 3. July 1977, pp. 297-309.

13. M. J. D. PoweU, "A fast algorithm for nonlinear constrainted optimization calculations", Presented at the 1977 Dundee Conference on numerical Analysis

14. M. J. D. Powell, "The convergence of variable metric methods for nonlinear constrained optimization calculations*[9]. Presented at the Nonlinear Programming 3 symposium held at Madison, Wisconsin 1977

15. M. J. D. Powell, "Variable metric methods for constrained optimization**. Presented at the Third International Symposium on Computing Methods in Applied Sciences and Engineering (Paris) 1977

16. M. J. D. Powell "Constraint optimization by a variable metric method,** Tech. report, DAMTP University of Cambridge , 1978.

17. M. J. D. Powell, "Gradient conditions and Lagrange multipliers in nonlinear programming,** Tech. report, DAMTP University of Cambridge, 1979.

18. M. J. D. Powell, "Quasi-Newton formulae for sparse second derivative matrices,** Tech. report, DAMTP University of Cambridge, 1979.

19. M. J. D. Powell, "Optimization algorithm in 1979**, Presented at the Ninth IFIP Conference on Optimization Techniques (Warsaw) 1979

20. R.M. Chamberlain, C Lemarechal H.C Pedersen and MJ.D. Powell, **The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization", Tenth International Symposium on Mathematical Programming August, 1979

21. T. J. Berna, M. H. Locke and A. W. Westerberg, "A New Approach to Optimization of Chemical Processes," *AlChE. Journal,* Vol. 26, January 1980, pp. 37-43.

22. J.E.Dennis and Jorge J. More, "Quasi-Newton Methods^Motivation and Theory," *SIAM Review,* VoL 19, No. 1, January 1977, pp. 46-89.

23. J. K. Reid, "Least squares solution of sparse systems of nonlinear equations by a modified Marquardt algorithm.," *Decomposition of Large-Scale Problems,* 1973, pp. 437-445.

24. A. W. Westerberg and S. W. Director, "A Modified Least Squares Algorithm for Solving Sparse N X N Sets of Nonlinear Equations," *Computers and Chemical Engineering,* Vol. 2, 1978, pp. 77-81.

25. Khan, M. R. , Willmert, K. D. and Thornton, W. A., "An optimality criterion method for large-scale structures/* *A/AA Journal,* Vol. 17, 1979, pp. 753.

26. C Fleury and M Gerandin, "Optimality Criteria and Mathematical Programming in Structural Weight Optimization," *Computer & Structures,* VoL 8. 1978, pp. 7-17.

27. G. Sander and C Fleury, "A Mixed Method in Structural Optimization,9* *Int. J. Numer. Meth. Eng.,* VoL 13. 1978, pp. 385-404.

28. C Fleury, "A Unified Approach to Structural Weight Minimization,** *Comp. Meth. Appl. Mech. and Eng.,* VoL 20, 1979, pp. 17-38.

29. C Fleury, "Structural Weight Optimization by Dual Methods of Convex Programming,** *Inter. J. Numer. Meth. Eng.,* VoL 14, 1979, pp. 1761-1783.

30. C Fleury and L. A. Schmit, "Primal and Dual Methods in Structural Optimization,** *ASCE J. Structural Div.,* May 1980, pp. 1117-1135.

31. R. Fletcher, 'The calculation of feasible point for linearly constrained optimization problems," Tech. report AERE R.6354, U.K.A.E.A. Research Group, April 1970.

31 R. Fletcher, "A Fortran Subroutine for General Quadratic Programming/* Tech. report AERE R.6370, U. K. A. E. A. Research Group, June, 1970.

33. R. Fletcher, "A General Quadratic Programming Algorithm,** •/. *Inst. Maths Applies,* VoL 7. 1971, pp. 76-91.

34. Nien-Hua Chao, *Application of a Reduced Quadratic Programming Technique to Optimal Structural Design,* PhD dissertation, Carnegie-Mellon University, November 1981.

FIGURES