Int. J. Solids Structures, 1975, Vol. 11, pp. 793–802. Pergamon Press. Printed in Great Britain.

ITERATIVE DESIGN WITH DEFLECTION CONSTRAINTS

W. R. Spillers† and J. Funaro‡
Department of Civil Engineering, University of Columbia, New York, NY 10027, U.S.A.

(*Received* 22 *April* 1974; *revised* 4 *November* 1974)

Abstract—A procedure is developed for extending simple iterative design techniques for trusses to the case in which deflection constraints are present. For statically determinate trusses, this procedure involves modifying both the stiffness matrix and the joint load matrix in the analysis/redesign cycle; for statically indeterminate trusses, the allowable stress must also be modified.

INTRODUCTION

The formulation of structural design as a nonlinear programming problem was an extremely important step but it is now clear that it was only a first step. In pursuing the nonlinear programming problems which arise in design, considerable algorithmic difficulties have developed[1] to the extent that a simple design situation can require a massive computational effort.

Somewhat anomalous to these algorithmic difficulties is the fact that engineers do "quite well" today in complex design situations which would be impossible as nonlinear programming problems. The engineer usually performs a kind of iterative design in which initial assumptions are subsequently modified on the basis of an analysis/redesign cycle. In [2–3] this iterative procedure was examined for a simple truss design problem and it was shown that in this case, allowable stress design—which would appear to deal with questions of safety—can produce designs of minimum weight. In more complex situations such as frame and sandwich plate design, allowable stress design does not lead to minimum weight but can be modified[4] to do so.

In these studies of iterative design, simplification was achieved through the relaxation of the constitutive equations and the optimization problems formulated in terms of forces and moments. When deflection constraints are included, it is, of course, not possible to relax the constitutive equations and the optimization problems which result are considerably more complicated than those which do not involve deflection constraints.

In this paper, the simple case of a truss design problem with constant allowable stress and deflection constraints is examined. It will be shown that in the case of statically determinate trusses, iterative design using a modified load matrix and a modified stiffness matrix will lead to minimum weight design. The general case is treated as a two phase process involving the statically determinate case and a second phase in which the allowable stress is modified. it may be noted that Schmit and Fox[9] have used an "integrated approach" which also simultaneously considers the effects of deflection and stress constraints, but their algorithm is unrelated to the algorithm used in this paper.

The paper begins in the next section with an outline of the analysis problem, this is followed by a formal statement of design with deflection constraints as an optimization problem, iterative design is then developed, and finally examples are presented.

THE ANALYSIS PROBLEM

The displacement (node) method of structural analysis can be represented as [5]

$$\begin{aligned} \tilde{N}F &= P \quad \text{node equilibrium} \\ F &= K\Delta \quad \text{constitutive equation} \\ \Delta &= N\delta \quad \text{member–joint displacement relationship} \end{aligned} \tag{1}$$

†Professor of Civil Engineering and Engineering Mechanics.
‡Graduate Student.

For the case of a truss with B bars and J joint degrees of freedom, N is a $B \times J$ matrix of direction cosines, K is a $B \times B$ diagonal matrix in which $K_{ii} = A_i E / L_i$, F and Δ are $B \times 1$ member force and displacement matrices, and P and δ are $J \times 1$ joint force and displacement matrices. Here

A_i–area of member i
L_i–length of member i
E–Young's modulus.

Equation (1) is usually first solved for δ as

$$\delta = (\tilde{N}KN)^{-1}P \tag{2}$$

from which Δ and F follow directly.

THE DESIGN PROBLEM

In the analysis problem, it is required to find F, Δ, and δ given N, K, and P; that is, given a structure and the applied loads, find the response. In the design problem considered here, in addition to determining the response, the member areas are to be determined so that the weight is minimized while certain constraints described below are satisfied.

It is convenient at this point to introduce the concept of the allowable length change Δ_i^a, of member i,

$$\Delta_i^a = \sigma^a L_i / E \geq 0 \tag{3}$$

where $\sigma^a \geq 0$ is the given allowable stress. it may be noted that when the member length change is less than the allowable length change Δ_i^a, the member stress is less than the allowable member stress:

$$|\Delta_i| \leq \Delta_i^a \Rightarrow \left|\frac{F_i L_i}{A_i E}\right| \leq \frac{\sigma^a L_i}{E} \tag{4}$$

or

$$|F_i / A_i| \leq \sigma^a. \tag{5}$$

In the context of this paper a deflection constraint is simply a statement that the absolute value of a given displacement component at a node is limited in magnitude. Since stress constraints such as equation (5) are equivalent to constraints on member length changes, it is possible to specify deflection constraints by adding fictitious bars of appropriate length to the structure while at the same time requiring the forces in these bars to be zero. This is indicated schematically in Fig. 1.

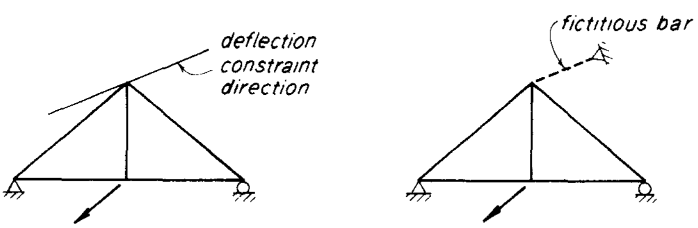

Fig. 1.

It is now possible to state the design problem formally as

$$\text{minimize} \quad \sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i^a)^2$$

subject to

$$\begin{aligned} \tilde{N}F &= P \\ \Delta &= N\delta \\ |\Delta| &\leq \Delta^a \quad \text{or} \quad |\Delta_i| \leq \Delta_i^a \\ F_i &= 0 \quad \text{in all fictitious bars} \\ \operatorname{sgn} F_i &= \operatorname{sgn} \Delta_i \end{aligned} \tag{6}$$

The objective function in equation (6) is proportional to the truss volume which is proportional to the truss weight since

$$\sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i^a)^2 = \sum_i K_i(\Delta_i^a)^2 = \frac{(\sigma^a)^2}{E}\sum_i A_i L_i \tag{7}$$

This form of the design problem has several advantages. The unknowns in this case are F, δ and Δ, i.e. it is required to find member forces and displacements which minimize the objective function (weight) while satisfying the constraints. With the exception of the last constraint, the constraints are linear and the forces and displacements are coupled only in the objective function. Equation (6) is furthermore similar in form to the dual linear programming formulation:

Primal problem

$$\text{minimize} \quad \sum_i |F_i|\Delta_i^a \quad \text{subject to } \tilde{N}F = P. \tag{8}$$

Dual problem

$$\text{maximize} \quad \tilde{P}\delta \quad \text{subject to } |N\delta| \leq \Delta^a$$

which was developed[2] for the case in which there are only stress constraints. Since this similarity will actually be used to develop an algorithm for the solution of equation (6) in the sections which follow, it is of interest to note here that the solution of equation (8) may be obtained through an "allowable stress design" type iterative procedure at each step solving

$$\tilde{N}F^{(n)} = P \qquad F^{(n)} = K^{(n)}\Delta^{(n)} \qquad \Delta^{(n)} = N\delta^{(n)} \tag{9}$$

and iterating

$$K_{ii}^{(n+1)} = |F_i^{(n)}|/\Delta_i^a \tag{10}$$

where the only requirement on $K^{(0)}$ is that it be positive definite. This is, of course, a repeated application of the node method, equation (1). It has been shown this procedure converges monotonically[3] and that it is a Newtonian procedure[4].

It is also of some interest to note here that the formulation with deflection constraints, equation (6) differs from the formulation which does not include deflection constraints, equation (8), by the fact that in the former it is not possible to assume a fully-stressed design. For fully-stressed designs

$$\begin{aligned} |F_i|/A_i = \sigma^a \Rightarrow |F_i|/|\Delta_i| = K_{ii} &= |F_i|E/(\sigma^a L_i) \\ &= |F_i|/\Delta_i^a \end{aligned} \tag{11}$$

and equation (6) reduces to equation (8).

THE STATICALLY DETERMINATE CASE

Because of the manner in which the variables separate in equation (6), it is natural to think of solution procedures in which either F or Δ is varied while the other is held fixed. The case in which F is known will be referred to here as the statically determinate case and discussed in this section. This is a slight variation upon common usage of the designation of statical determinacy which usually implies that the bar forces can be computed from equilibrium as $F = \tilde{N}^{-1}P$.

When F is known, equation (6) becomes

$$\text{minimize} \quad \sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i^a)^2$$

subject to (12)

$$\Delta = N\delta$$
$$|\Delta| \leq \Delta^a$$
$$\operatorname{sgn}\Delta_i = \operatorname{sgn} F_i$$

In the work which follows the constraint $\operatorname{sgn}\Delta_i = \operatorname{sgn} F_i$ will be satisfied by the computational procedure developed and no longer stated explicitly. The Lagrangean of equation (12) is then

$$\mathcal{L} = \sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i^a)^2 + \tilde{\alpha}(\Delta - N\delta) + \tilde{\beta}(\Delta^a - |\Delta|) \tag{13}$$

in which α and β are (matrix) Lagrange multipliers. The Kuhn Tucker conditions for this system are

$$\text{(a)} \quad \frac{\partial\mathcal{L}}{\partial\Delta_i} = 0 \Rightarrow -F_i\left(\frac{\Delta_i^a}{\Delta_i}\right)^2 + \alpha_i - \beta_i \operatorname{sgn}\Delta_i = 0$$

$$\text{(b)} \quad \frac{\partial\mathcal{L}}{\partial\delta_i} = 0 \Rightarrow (\tilde{N}\alpha)_i = 0 \tag{14}$$

$$\text{(c)} \quad \frac{\partial\mathcal{L}}{\partial\alpha_i} = 0 \Rightarrow \Delta_i = (N\delta)_i$$

$$\text{(d)} \quad \beta_i(\Delta_i^a - |\Delta_i|) = 0$$

The first two of these conditions can be combined by multiplying condition (a) by the matrix $\tilde{N}$ and using condition (b) to give

$$-\tilde{N}\left(F\left(\frac{\Delta^a}{\Delta}\right)^2\right) - \tilde{N}(\beta \operatorname{sgn}\Delta) = 0 \tag{15}$$

in somewhat symbolic notation. Let

$$P^* = -\tilde{N}\left(F\left(\frac{\Delta^a}{\Delta}\right)^2\right), \tag{16}$$

the Kuhn Tucker conditions for equation (12) then become

$$\tilde{N}(\beta \operatorname{sgn}\Delta) = P^*$$
$$\Delta = N\delta \tag{17}$$
$$\beta_i(\Delta_i^a - |\Delta_i|) = 0$$

The dual problem of equation (8) has the Lagrangean

$$\mathcal{L} = \tilde{P}\delta + \tilde{\gamma}(\Delta^a - |N\delta|) \tag{18}$$

in which γ is a (matrix) Lagrange multiplier and has the Kuhn Tucker conditions

$$\text{(a)} \quad \frac{\partial\mathcal{L}}{\partial\delta_i} = 0 \Rightarrow P_i - (\tilde{N}\gamma \operatorname{sgn}(N\delta))_i = 0$$

$$\text{(b)} \quad \gamma_i(\Delta_i^a - |(N\delta)_i|) = 0 \tag{19}$$

At this point the similarity between equation (17) and equation (19) can be used to develop a solution procedure for equation (12). While equation (17) makes explicit use of the Δ and equation (19) does not, this is only the question of a simple transformation. The real difference between these equations lies in the fact that in the former P^* is a function of the variable Δ while in the latter, P is a given matrix. Put another way, in the linear programming problem the equilibrium equations are linear while their analog in equation (17) is nonlinear.

The standard Newtonian algorithm requires the linearization of a system at each step. When the nonlinear equilibrium equations have been linearized, equation (17) becomes identical in form with equation (19) and the iterative procedure described in equations (9–10) becomes, with appropriate modification, a scheme for the solution of equation (12). This is done most simply by expanding the term $(\Delta_i^a/\Delta_i)^2$ in a Taylor series about the point $\Delta^{(n)}$,

$$\left(\frac{\Delta_i^a}{\Delta_i}\right)^2 \sim \left(\frac{\Delta_i^a}{\Delta_i^{(n)}}\right)^2 - 2\frac{(\Delta_i^a)^2}{(\Delta_i^{(n)})^3}(\Delta_i - \Delta_i^{(n)}) = 3\left(\frac{\Delta_i^a}{\Delta_i^{(n)}}\right)^2 - 2\frac{(\Delta_i^a)^2}{(\Delta_i^{(n)})^3}\Delta_i \tag{20}$$

When multiplied by the matrix $\tilde{N}$, the constant term in equation (20) gives rise to the analog of the joint load in the node method; the linear term, when written in terms of node displacements, corresponds to a simple modification of the stiffness matrix—something like the geometric stiffness term (6) used in nonlinear structural analysis.

Symbolically, this solution procedure involved iterating

$$\tilde{N}(K_E^{(n)} + K_G^{(n)})N\delta^{(n+1)} = \bar{P}^{(n)} \tag{21}$$

and

$$(K_E^{(n+1)})_{ii} = (K_E^{(n)})_{ii}|\Delta_i^{(n)}|/\Delta_i^a \tag{22}$$

$$(K_G^{(n+1)})_{ii} = 2F_i\frac{(\Delta_i^a)^2}{(\Delta_i^{(n)})^3} \tag{23}$$

where

$$\bar{P}^{(n)} = 3\tilde{N}\left(F\left(\frac{\Delta_i^a}{\Delta_i^{(n)}}\right)^2\right). \tag{24}$$

In summary the two modifications which must be made to the linear programming problem (equation 8) in order to include the effects of deflection constraints may now be seen in equation (20). When multiplied by the matrix $\tilde{N}$ in order to write joint equilibrium, the constant term in this equation corresponds to the usual given (constant) joint load term. The linear term in equation (18), when written in terms of joint displacements for the node method, requires changing the usual stiffness matrix. The solution follows by modifying any given computer program for truss analysis to (a) iterate, (b) adjust the joint load matrix as the displacements change from iteration to iteration, and (c) include a corrected stiffness matrix. Given a basic truss analysis program, these modifications are simple to carry out.

Examples are included in the final section of this paper.

THE STATICALLY INDETERMINATE CASE

The formulation in the statically indeterminate case (the general case) has been given in equation (6). In this case the Lagrangean is

$$\mathcal{L} = \sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i^a)^2 + \tilde{\alpha}(\Delta - N\delta) + \tilde{\beta}(\Delta - |\Delta|) + \tilde{\lambda}(P - \tilde{N}F) \tag{25}$$

which represents an addition of one term to equation (13). The Kuhn Tucker conditions in the general case then contain two terms,

$$\begin{aligned}&\text{(a)}\ \frac{\partial \mathscr{L}}{\partial F_i}=0 \Rightarrow (\operatorname{sgn} F_i)\frac{(\Delta_i{}^a)^2}{|\Delta_i|}-(N\lambda)_i=0\\&\text{(b)}\ \tilde{N}F=P\end{aligned} \tag{26}$$

in addition to those in equation (14).

As suggested above, the general case will be solved by iterating varying F while holding Δ fixed and then varying Δ, holding F fixed. The case in which F is held fixed (known) is given in the preceding section. When Δ is held fixed, equation (6) reduces to

$$\text{minimize}\quad \sum_i \frac{|F_i|}{|\Delta_i|}(\Delta_i{}^a)^2 \quad \text{subject to } \tilde{N}F=P. \tag{27}$$

This problem is identical in form with the primal problem of equation (8) and therefore any algorithm which can be used to solve equation (8) can be used to solve this problem. In terms of detail, what was $\Delta_i{}^a$ in equation (8) has become $(\Delta_i{}^a)^2/|\Delta_i|$ in equation (27). For that reason, the problem of deflection constraints can be regarded as requiring the use of a fictitious allowable length change or allowable stress.

Since equation (21) is a linear programming problem, it follows that there exists an optimal statically determinate structure (2). Were this not true the fundamental theorem of linear programming would be violated when applied to equation (27) using the optimal value of Δ.

SOME EXAMPLES

The method proposed above for weight minimization in the presence of deflection constraints is essentially a generalization of the well known allowable stress design procedure and involves simply an analysis/redesign cycle. While the general efficacy of the method will only be understood after it has been used for some time, it has the immediate advantage of being easy to implement. In fact, the examples below have been carried out using a 15–20 card modification of the plane truss program given in [5].

Figure 2 shows a rather simple but instructive example. In this case, equation (12) becomes

$$\text{minimize}\quad \frac{1}{\Delta_1}+\frac{1}{2\Delta_a}$$

subject to

$$\begin{aligned}&|\Delta_1|\le 1\\&|\Delta_2|\le 1/2\\&|\Delta_1+\Delta_2|\le 1\\&\Delta_1,\Delta_2\ge 0.\end{aligned} \tag{28}$$

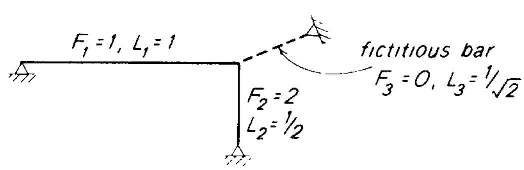

Fig. 2.

Several aspects of Fig. 3 are typical of problems with deflection constraints:

1. The solution lies on the boundary of a convex region defined by the linear constraints but may not be an extreme point. (It is not in this case.)

2. The force vector has been plotted in Fig. 3. Since in the absence of deflection constraints the design problem may be written as maximize $\tilde{P}\delta=\tilde{F}\Delta=F.\Delta$, point A is an optimal solution when the deflection constraint is absent. Point B is easy to get from the linear programming problem, is feasible, and thus a tempting approximate solution to the problem with deflection constraints.

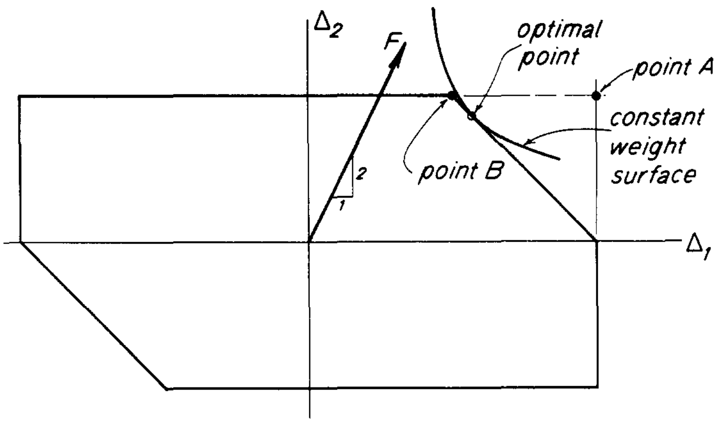

Fig. 3.

3. The surfaces of constant weight are convex surfaces as indicated by Prager[7].

The convergence of the iterative scheme is indicated in Fig. 4.

Figure 5 shows a less interesting example in which a unit load is applied vertically at the center of the truss on the lower chord. This turns out to be a case in which in order to meet the deflection constraints, all the bar areas are increased *proportionally*. Here $E = 30 \times 10^6$ psi and $\sigma^a = 20{,}000$ psi. Convergence is shown in Fig. 6.

The final two examples deal with the statically indeterminate case; since the optimal solution is statically determinate the algorithm must select an optimal substructure from the given bars and joints. The first of these examples which is shown in Fig. 7 is rather predictable. Convergence in this case is indicated in Fig. 8. Here $E = \sigma^a = 1$.

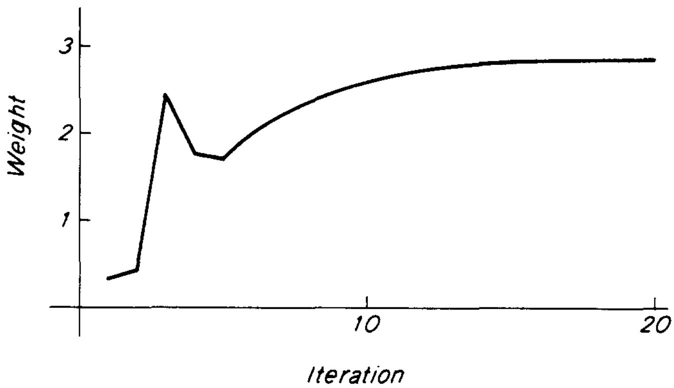

Fig. 4.

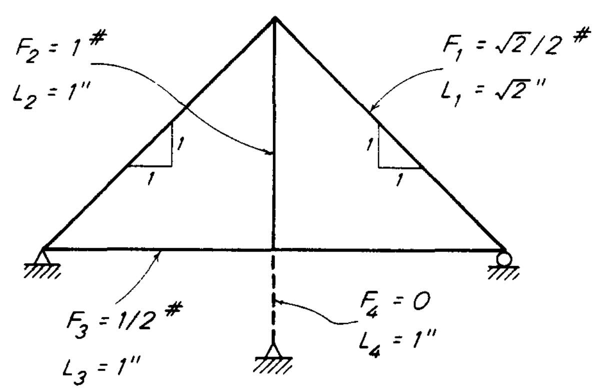

Fig. 5.

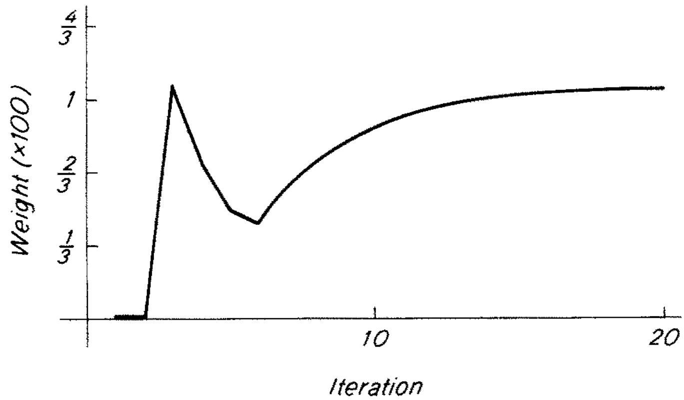

Fig. 6.

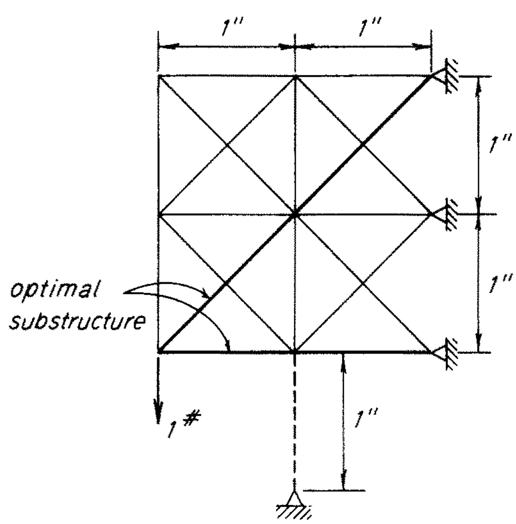

Fig. 7.

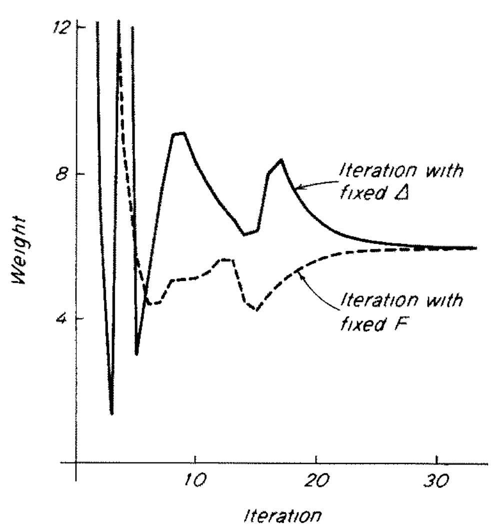

Fig. 8.

In the final example shown in Fig. 9, the basic structure from the previous example is used, but this time an additional load has been added and the deflection constraint changed. Were it not for the deflection constraint an optimal structure would contain an element like the optimal truss shown in Fig. 7 for each load and the weight would be $1\frac{1}{2}\times 6 = 9$. Because of the deflection constraint, the optimal structure has a weight of 11 and has the configuration indicated in Fig. 10.

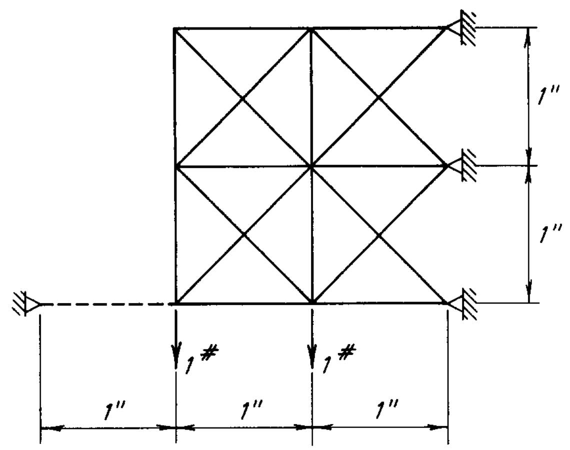

Fig. 9.

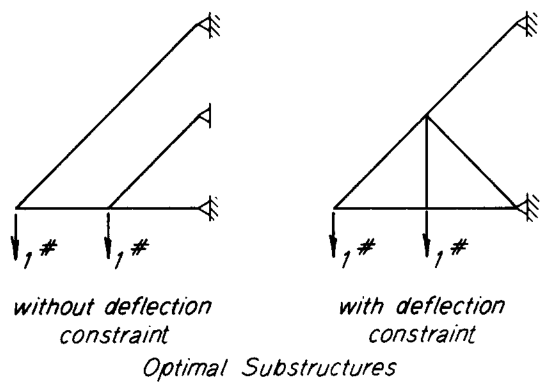

Fig. 10.

Figure 11 shows the convergence for the case of this final example and, in fact, as shown, the basic algorithm did not converge. Two things of interest did, however, occur:

(a) While it appears in Fig. 11 that nothing is happening after about iteration 18, a more detailed examination of the computer printout shows that the forces in the members are in fact converging to the optimal force system while the objective function remains almost constant.

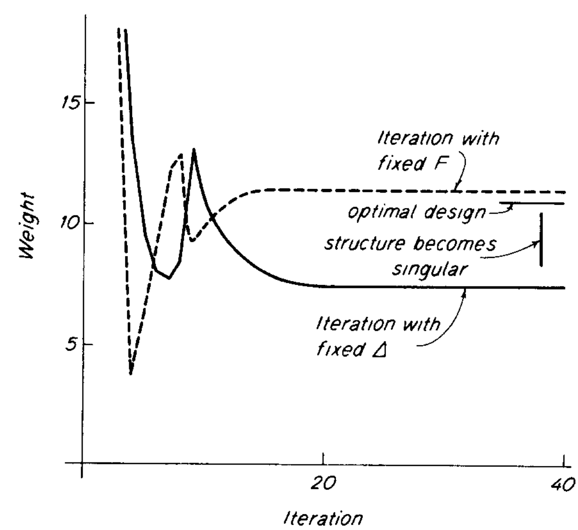

Fig. 11.

(b) Before this process terminated, the structure became somewhat singular due to the many bar areas going to zero making it difficult to continue the calculations. However, by the time the structure had become singular, the optimal substructure had become evident and it was possible to complete the calculations using the optimal statically determinate substructure.

The difficulty with the structure becoming singular when many bar areas go to zero appeared in an earlier work [8] in which an algorithm was also asked to select an optimal substructure. The earlier conclusion that this is not an effective way in which to generate configurations automatically seems again verified by the last example here.

CONCLUDING REMARKS

This paper extends earlier work on iterative design to include questions of deflection constraints. The thrust of the paper is computational and no new properties of optimal designs have been identified. The principal advantage of the method proposed is simplicity: It only requires minor modifications of an analysis program for its implementation.

With regard to more general classes of structures, it would appear that the extension of this proposed algorithm to frames and sandwich plates is straightforward using [4].

Acknowledgement—This work has been supported by the National Science Foundation.

REFERENCES

1. L. A. Schmit, Jr. and B. Farshi, Some approximation concepts for structural synthesis. *AIAA paper* #73-341, *AIAA/ASME/SAE 14th Structures, Structural Dynamics, and Materials Conference, Williamsburg, Va.* (*March* 1973).
2. **W. R. Spillers and John Farrell, On the analysis of structural design. *J. Math. Anal. Appl.* 25, 2 (1969).**
3. W. R. Spillers and John Farrell, An absolute-value linear programming problem. *J. Math. Anal. Appl.* **28**, 1, 153 (1969).
4. W. R. Spillers and S. Al-Banna, Optimization using iterative design techniques. *Computers and Structures* **3**, 5 (1973).
5. W. R. Spillers, *Automated Structural Analysis: An Introduction*. Pergamon Press, Oxford (1972).
6. J. S. Przemienieck, *Theory of matrix Structural Analysis*. McGraw-Hill, New York (1968).
7. J. M. Chern and W. Prager, Minimum-weight design of statically determinate trusses subject to multiple constraints. *Int. J. Solids and Structures* **7**, 931 (1971).
8. W. R. Spillers and S. Al-Banna, Iterative design techniques. *Preprint* #1373. ASCE Structural Engineering Meeting, Baltimore (April 1971).
9. L. A. Schmit and R. L. Fox, An integrated approach to structural synthesis and analysis. *AIAA J.* **3**, 6, 1104 (1965).