Minimum Weight Design of Elastic Redundant Trusses under Multiple Static Loading Conditions

C. Y. SHEU*
Case Western Reserve University, Cleveland, Ohio
AND
L. A. SCHMIT JR.†
University of California, Los Angeles, Calif.

A method for automated selection of a minimum weight truss, from a subset of configurations obtained by omitting various member combinations from a primary configuration, is presented. A feasible direction algorithm is used to find an upper bound solution for any selected configuration. A dual simplex algorithm is used to rapidly generate lower bound solutions for many subset configurations considering only equilibrium conditions and stress limits. The lower bound solutions guide the selection of subset configurations for which upper bound solutions of reduced weight are sought. Examples of planar and space trusses illustrate the efficacy of the bounding technique presented.

I. Mathematical Formulation

CONSIDER an elastic redundant truss with given geometric configuration and support conditions. The truss is to sustain several given alternative systems of static loads applied at the joints. Only q of these load systems which are supposed to govern the design will be considered. For simplicity, the truss is conventionally idealized; that is, the members of the truss are assumed to be joined together by frictionless connections and only axial forces, tension or compression, occur in the members under the loading. The design problem addressed herein is the selection of a minimum weight optimum truss configuration, as well as its member sizing, from a subset of truss configurations obtained by omitting various members from the given primary configuration. The design is subject to the following constraints: the cross-sectional area A_j of member j, when it remains in the truss, is restricted to a given continuous range, the upper limit A_j^u of which is finite and the lower limit A_j^l may be, and usually is, greater than zero; the stress σ_{jk}, defined as the internal force f_{jk} of the member j under the kth loading condition divided by A_j, has given lower limit $-\sigma_{jk}^c$ and upper limit σ_{jk}^t, and these stress limits are assumed independent of the cross-sectional area of the member; and the displacement component u_{ik} of a joint under loading condition k also has specified lower $-u_{ik}^l$ and upper u_{ik}^u limits (each joint, except for fixed supports, has at least one degree of freedom, the corresponding displacement components are numbered in a fixed order and identified as u_i). The problem can thus be formulated as

$$\min W = \sum_{j=1}^{n} \rho_j l_j A_j \tag{1}$$

subject to

$$\sigma_{jk}^t \geqq \sigma_{jk} \geqq -\sigma_{jk}^c \tag{2}$$

$$A_j^u \geqq A_j \geqq A_j^l \geqq 0, \quad \text{if} \quad A_j \neq 0 \tag{3}$$

and

$$u_{ik}^u \geqq u_{ik} \geqq -u_{ik}^l \tag{4}$$

with

$$f_{jk} = A_j \sigma_{jk} \tag{5}$$

$$\sum_{j=1}^{n} b_{ij} f_{jk} = p_{ik} \tag{6}$$

$$\sigma_{jk} = \frac{E_j}{l_j} \sum_{i=1}^{m} b_{ij} u_{ik} \quad \text{if} \quad A_j \neq 0 \tag{7}$$

for

$$i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, q$$

where l_j, ρ_j are the length and the weight density of the member j; E_j is its Young's modulus of elasticity; p_{ik} is the component of the given external load, in the direction of u_i under the kth loading condition, applied at the joint with the ith degree of freedom; Eqs. (6) are the equilibrium conditions for the joints, and Eqs. (7) are the stress-displacement relations. If $A_j^l > 0$ in Eq. (3), A_j can be zero only in a given set of combinations.

II. A Lower Bound on Design Weight

First, consider the design subject to Eq. (2) only. Of course, $A_j \geqq 0$ and the relations (5–7) still have to be satisfied. Assume r ($r = n - m$) members are chosen as redundants and denote their internal forces f_{jk} as $R_{1k}, R_{2k}, \ldots, R_{rk}$. Solving Eq. (6) for f_{jk} in terms of R_{lk}, we have

$$f_{jk} = f_{jk}^o + \sum_{l=1}^{r} a_{jl} R_{lk} \tag{8}$$

where f_{jk}^o are the statically determinate internal forces when all the redundant members are removed. In view of Eq. (5), the stress constraints (2) thus require that

$$A_j \geqq A_j^m \equiv \max_k (A_{jk}^m) = A_{jk_j}^m \tag{9}$$

where

$$A_{jk}^m \equiv f_{jk} h_{jk} \quad \text{with} \quad h_{jk} \equiv \begin{cases} 1/\sigma_{jk}^t \\ 0 \\ -1/\sigma_{jk}^c \end{cases} \quad \text{if} \quad f_{jk} \begin{cases} > 0 \\ = 0 \\ < 0 \end{cases} \tag{10}$$

and k_j denotes the loading condition under which the last equality in Eq. (9) holds for member j. Set

$$W_m = \sum_{j=1}^{n} \rho_j l_j A_j^m \tag{11}$$

then Eq. (9) implies $W \geqq W_m$ and thus $\min W \geqq \min W_m$. Substitution of Eqs. (8–10) into (11) furnishes

Presented as Paper 71-362 at the AIAA/ASME 12th Structures, Structural Dynamics and Materials Conference, Anaheim, Calif., April 19–21, 1971; submitted April 30, 1971; revision received October 14, 1971. This research was supported by NASA Research Grant NSG 110-61.
Index category: Optimal Structural Design.
* Postdoctoral Research Associate.
† Professor of Engineering and Applied Science. Associate Fellow AIAA.

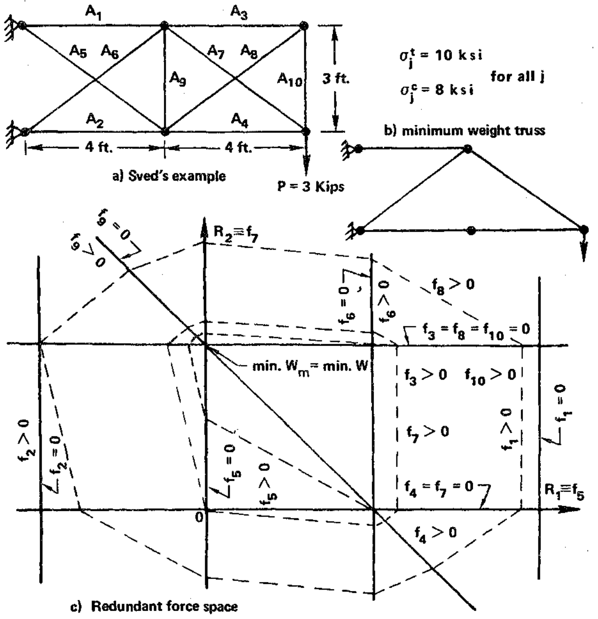


Fig. 1 A minimum weight truss under single loading condition.

$$W_m = \sum_{j=1}^{n} \rho_j l_j f_{jk_j}{}^o h_{jk_j} + \sum_{l=1}^{r} \sum_{j=1}^{n} \rho_j l_j h_{jk_j} a_{jl} R_{lk_j} \tag{12}$$

In a $r \times q$ dimensional space with rectangular coordinates R_{lk}, the hyperplanes $f_{jk} = 0$ divide the whole space into a number of $r \times q$ dimensional polyhedrons (or semi-infinite polyhedrons). Inside each polyhedron, the f_{jk} have definite signs so the h_{jk} are given constants, and in view of Eqs. (8) and (10), the $A_{jk}{}^m$ are given linear functions of R_{lk}. Therefore, any combinations of $A_{jk_t}{}^m = A_{jk_s}{}^m$ constitute hyperplanes which in turn divide the polyhedron into a number of subpolyhedrons. Now inside each of these subpolyhedrons the k_j are fixed. On the boundaries of these subpolyhedrons, the k_j and hence the h_{jk_j} may change. However, W_m and the $A_j{}^m$ are continuous across these boundaries and throughout the whole space (W_m approaches infinity as the R_{lk} do). Because of this continuity W_m, as a linear function of R_{lk}, attains its minimum of a subpolyhedron at a vertex of the subpolyhedron (mathematically it may happen that the minimum is attained on an edge of the subpolyhedron containing that vertex), and therefore, attains its global minimum of the whole space at a common vertex of several subpolyhedrons somewhere in the space. It is not difficult to prove that the $A_{jk}{}^m$ are convex functions of the R_{lk} since each h_{jk} can have only one of three values as defined in Eq. (10) throughout the space. In view of Eq. (9), $A_j{}^m$ is also convex and so is W_m [see Eq. (11)]. Therefore, a local minimum of W_m is also a global minimum.

If the truss is under a single loading condition, i.e., $q = 1$, no subpolyhedrons exist and W_m attains its minimum at a common vertex of several polyhedrons. At the vertex, at least r of the f_j vanish, so their corresponding members can be removed and the truss is no longer statically indeterminate. Hence, the remaining f_j are independent of the A_j which can then be made as small as $A_j{}^m$ without violating the stress constraints, therefore, $W = W_m = \min W_m = \min W$. This conclusion has been known for decades, and similar proofs have been offered by previous authors (see Refs. 1–5) in somewhat different forms. Figures 1a and 1b show an example used by Sved[3] and Fig. 1c shows the polygons, mentioned previously as polyhedrons, in the R_l plane corresponding to the redundants he chose. Some contours of equal W_m are also shown by broken lines. Note that the minimum weight truss thus obtained may become statically unstable as in the example shown in Fig. 1.

For a truss under multiple loading conditions, however, no such simple conclusion can be drawn. First of all, a member j may be removed only if the correspondent $f_{jk} = 0$ for all k. At the vertex of the subpolyhedron, where W_m attains its minimum, we only know, in general, that the number of $f_{jk} = 0$ and $A_{jk_s}{}^m = A_{jk_t}{}^m$ is at least $r \times q$; it may happen that the number of $A_j{}^m = 0$ is less than r so the truss is still statically indeterminate. Figure 2 shows one of these cases for a three-bar truss under two loading conditions (min W_m is shown surrounded by some contours of W_m). If the truss has r' redundants, then there are $r' \times q$ compatibility conditions to be satisfied, therefore, A_j, in general, cannot be made as small as $A_j{}^m$ for all j. Indeed, the compatibility conditions may furnish no all-nonnegative finite A_j under the given R_{lk} at min W_m, which means that the given R_{lk} are completely artificial and do not exist in the real truss (the example in Fig. 2 belongs to this class; it should be understood that plastic behaviors are not considered here).

In short, the truss configuration at min W_m may turn out to be statically indeterminate, so min $W >$ min W_m, in general, and this truss configuration is not necessarily the truss configuration, of min W. Note that compatibility conditions destroy the linearity of the problem and makes it difficult to find min W analytically except for extremely simple problems such as the one shown in Fig. 2. Even assuming min W is a fully stressed design (i.e., the stresses of each member reach their upper or lower limits at least under one loading condition) Cilley[1] showed that an analytical treatment, in which the member forces are expressed in terms of the redundants as in Eq. (8), is not only complicated but also involves too much labor, even in a small truss with one or two redundants, to be used in practice. In spite of all these unfavorable facts, min W_m still serves as a lower bound of min W, and as shown in the sequel, it is easily obtained and useful in guiding the search for an optimum truss configuration.

III. Difficulty by Direct Searching Techniques

In view of the formulation in Sec. I, one might assert that the problem can be solved by directly applying the nonlinear programing (hereafter abbreviated as NLP) techniques developed in recent years in the redundant force space mentioned previously, or in an A_j design variable space. Unfortunately, even if it is assumed that NLP methods can find a global minimum for a fixed truss configuration starting from

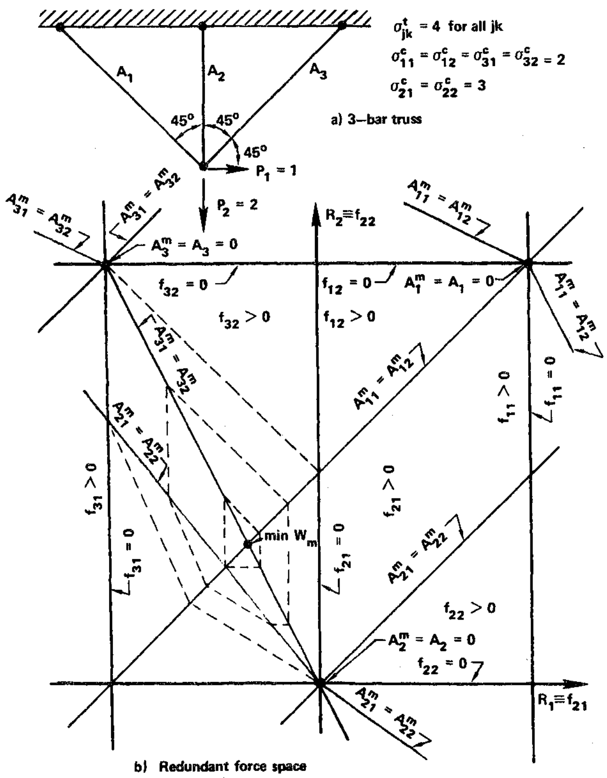


Fig. 2 Redundant truss under multiple loadings.

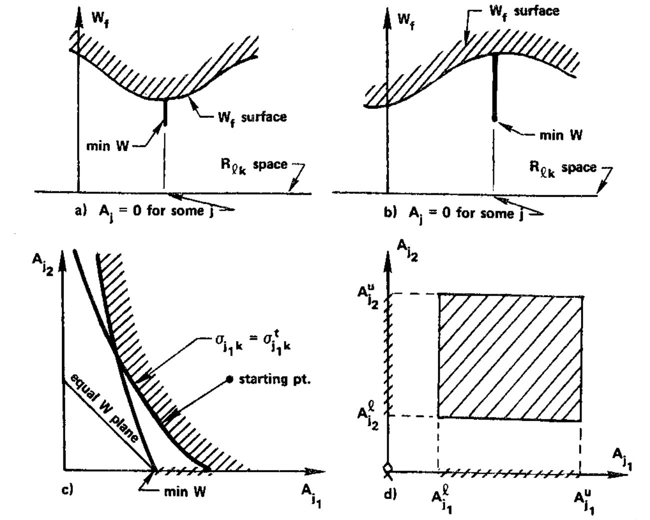


Fig. 3 Feasible or admissible regions (shaded) and the difficulty of direct searching.

arbitrary values of A_j for the configuration, it is found that these techniques often cannot reach, or come close to, achieving min W. This failure to find min W is not, in general, attributable to any defect in the NLP techniques but rather it occurs because of certain special characteristics of the design problem.

Consider the problem in the redundant force space. At each point in the subspace of the redundant force space where feasible designs‡ exist, denote a minimum design weight among them as W_f. The value of W_f is continuous almost everywhere in the subspace except in some subset where some members can be removed; i.e., A_j can be set equal to zero (see Fig. 2b) and the corresponding stress constraints (2) can be released. If any of these constraints is active§ for the neighboring designs with W_f, the abrupt release of these constraints is likely to lead to a "drop" discontinuity of W_f on this subset with respect to its neighborhood. If min W falls in one of these subsets, the existing NLP techniques have little chance of finding it except when this "ditch" is right down in a "valley" of the W_f surface as shown in Fig. 3a. If the ditch is at the top of a "hill" as shown in Fig. 3b, there is no chance at all of finding min W there since existing techniques always search a minimum down to a valley.

When the problem is considered in a design space with A_j coordinates, similar characteristics are exhibited. When an A_j approaches zero, the corresponding stresses $\sigma_{jk} \equiv f_{jk}/A_j$ may, and in most cases do, approach nonzero limits [see Eq. (7)]. Therefore, it may happen that min W is located at a point where some $A_j = 0$ but one of the corresponding stresses σ_{jk} exceeds the stress limit at all neighboring designs with the $A_j \neq 0$. In this case, the stress constraint will repel an iteration of the existing techniques approach to min W through its neighborhood with the $A_j > 0$ as shown in Fig. 3c.

While the foregoing discussion deals with stress constraints, similar situations can arise with respect to other constraints. For example, the displacements of an unloaded joint may approach nonzero limits as the size of all the connecting members approaches zero. Therefore, the corresponding displacement constraints may be violated in the neighborhood of a minimum weight design with these members omitted.

Furthermore, if $A_j^l > 0$, the conditional size constraints (3) will divide the A_j design space into several disjoint regions with no admissible designs between any two of them (see Fig. 3d). Under this circumstance, when starting within one region, the existing techniques with continuous variables can only search for a minimum within that region, and before a minimum or a near minimum is reached, it is difficult to decide to shift to another region.

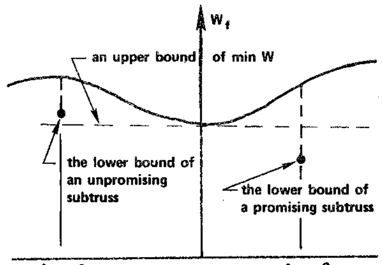
To overcome the difficulties and find a min W, Sved and Ginos[7] suggested that a "search of all perfect structures obtained by omitting members¶ is necessary." In the R_{lk} space, this means in addition to finding a min W_f on the continuous part of the W_f surface, it is necessary to search down to the bottom of every possible ditch to find out if min W lies there. No matter to what extent this is correct, the effort that would be required to attack this problem directly, by seeking optimum cross-sectional areas for all stable subtrusses, is, in general, overwhelmingly large. Since even in a primary truss configuration with a moderate number of members and degrees of redundancy, a very large number of stable subtrusses can be obtained, it is highly undesirable to "search them all" using an NLP technique (for example, a 15-member truss with six redundants can generate more than five thousand subtrusses). The foregoing considerations, therefore, raise an important fundamental question. Namely, is there an efficient scheme for seeking out promising candidate configurations from a large number of subtrusses that will drastically reduce the number of cases to be treated using an NLP technique? A bounding technique is set forth in subsequent sections as an affirmative response to this question.

IV. Linear Programing For Lower Bounds

Since interest is centered on determining the global minimum weight design rather than on finding the minimum weight design for each subtruss and since each search for a minimum weight subtruss design by an NLP technique is very costly, one should ask, "Is it worth searching this ditch for its bottom?", before starting each NLP search (for a minimum of a subtruss). The question may be answered if a low enough upper bound on min W is already available and a high enough lower bound on the minimum design weight of the subtruss can be easily obtained so that an unpromising subtruss can be distinguished from an attractive one. A subtruss with its lower bound of minimum equal to or greater than an upper bound on min W is called unpromising because it is obviously not the optimum. Any lowest design weight among the existing feasible designs can serve as an upper bound. Initially, a min W_f on the continuous part of the W_f surface (see Figs. 3a and b) can be obtained by an NLP technique to serve as a first upper bound. Figure 4 illustrates the foregoing idea schematically.

Whatever lower bound is chosen, it should be one which is much easier to obtain than the corresponding minimum determined by an NLP technique, if a substantial reduction in total effort is to be realized. In Sec. II, a lower bound min W_m on min W was found by considering equilibrium conditions and stress limits only. This min W_m can be interpreted as a minimum weight truss in the limit design sense if the σ_{jk}^t and σ_{jk}^c are the yield stresses of the members and the p_{ik} are the plastic collapse loads of the truss. To obtain the value of a min W_m, one can search from one vertex of the subpolyhedrons in R_{lk} space to

Fig. 4 Promising and unpromising subtrusses.



‡ A design which satisfies all constraints is called feasible.

§ When the constrained quantity is at or close to its upper or lower limits.

¶ Here "perfect" should be understood to mean statically stable. Hereinafter the term "subtruss" refers to stable truss configurations obtained from the given primary truss configuration by omitting members.

another along the edge connecting them in the decreasing weight direction. This is equivalent to solving the following linear programing (hereafter abbreviated as LP) problem[5,6] by the simplex method:

$$\min W_m \equiv \sum_{j=1}^{n} \rho_j l_j A_j^m \tag{13}$$

subject to

$$\sigma_{jk}{}^t A_j^m \geqq f_{jk} \geqq -\sigma_{jk}{}^c A_j^m \tag{14}$$

$$\sum_{j=1}^{n} b_{ij} f_{jk} = p_{ik} \tag{15}$$

and

$$A_j^u \geqq A_j^m \geqq 0 \tag{16}$$

where $i = 1,2,\ldots,m$; $j = 1,2,\ldots,n$; $k = 1,2,\ldots,q$. Note that the upper limits A_j^u of A_j are also imposed on A_j^m here. Indeed, any constraints which do not destroy the linearity of the foregoing LP problem and do not divide the design space into disjoint feasible regions should be included to raise the lower bound of the minimum as high as possible. The lower limits A_j^l on A_j^m are not included by the restriction just mentioned (also see Fig. 3d). Excluding these lower limits may decrease the lower bound of the minimum in the foregoing LP problem and/or those of the subtrusses we are going to find subsequently. This can have the effect of making a particular subtruss appear more promising than it would if the lower limits A_j^l had been included. However, when the A_j^l are small, their exclusion will have little effect on the lower bounds obtained and hence, the over-all result may not be influenced at all. Furthermore, excluding the lower limits A_j^l will prove convenient in that it [Eq. (1)] leads to a significant efficiency advantage when finding the lower bounds of the minimum for a large number of subtrusses and Eq. (2) facilitates development of a simpler main searching algorithm.

To obtain a lower bound of the minimum (hereafter called LBM) for a subtruss, it is only necessary to impose one additional constraint $A_j^m = 0$ in the foregoing LP problem for each member omitted (it would be necessary to replace (16) with $A_j^m = 0$ if $A_j^l > 0$ were included). In general, the more constraints $A_j^m = 0$ imposed, the larger the LBM obtained. For example, a subtruss with one member omitted will yield a LBM less or equal to the LBM obtained with one additional member omitted. This would not necessarily be true if the A_j^l had not been excluded in Eq. (16) and were reached in the former LBM. By excluding A_j^l, the LBM exhibits the foregoing property, and, therefore, some unpromising subtrusses can be identified even without computing their LBM. Furthermore, if the LBM solution shows that in addition to the imposed condition $A_j^m = 0, j \in J_1$ there are some other $A_j^m = 0, j \in J_2$ in the solution, this solution is obviously also a LBM for a subtruss with $A_j^m = 0, j \in J_3$ where $J_1 \subseteq J_3 \subseteq (J_1 \cup J_2)$. The minimum obtained without imposing any of these additional constraints is the smallest one and will be called the first LBM. After the optimum tableau of the first LBM is obtained by a simplex algorithm, it can be used as an initial tableau in a dual simplex algorithm to find the LBM of any subtruss with some $A_j^m = 0$. In practical computation, no additional equations $A_j^m = 0$ need to be added to the original tableau. It is only necessary to convert A_j^m from the basic (or the nonbasic at its upper bound) to the nonbasic at its lower bound, zero, adjust the values of the constant column and delete the column to which A_j^m now belongs. Then use the dual simplex method to find new basic variables until a new optimal tableau is obtained.

Actual computation in a modern high-speed computer has shown that the cost of finding a LBM for a subtruss is considerably smaller and sometimes negligible compared to that of finding a minimum by an NLP technique. Because of the relatively low cost of finding LBM solutions using a dual simplex algorithm (hereafter called DSA), these solutions provide an effective means for guiding the selection of promising candidate configurations (subtrusses). Furthermore, the LBM solutions also furnish starting values of A_j for upper bound NLP solutions that are often as good as those available from other methods.

V. An NLP Technique for Upper Bounds

Having an LBM for a subtruss, an upper bound of the global minimum (hereafter called UBGM) is needed in order to judge whether or not the subtruss is a promising candidate configuration for the global minimum (based on its LBM). The smaller the UBGM obtained, the more likely it is that many subtrusses can be rapidly identified as unpromisng. As promising subtrusses are identified, the UBGM can often be reduced by seeking a solution of the NLP problem (1–7) for one of the promising candidate configurations. Each time an effort is made to reduce the UBGM an NLP problem must be solved. Therefore, it is desirable to employ an efficient NLP algorithm. An algorithm primarily based on Zoutendijk's feasible direction method[8] has thus been adopted. In the actual computer program several special devices have been introduced to accelerate the speed of convergence thus improving over-all efficiency. The details of the specific NLP algorithm employed to find upper bound solutions are not central to the basic bounding technique presented here. A detailed description of the NLP algorithm used and an evaluation of its efficiency in comparison with other existing capabilities will be reported separately.

VI. Main Searching Algorithm

Having the tools to find lower and upper bound solutions for a subtruss, one needs now an over-all sorting strategy which will minimize the total cost of finding an over-all optimum design. A searching algorithm based on the bounding technique is presented in this section. The basic approach is: a) find the LBM solutions for those subtrusses which have an equal chance to become a most promising candidate among those still under consideration; b) seek a reduced UBGM by finding the NLP solution for the most promising candidate still surviving, and repeating this step until one or more subtrusses have an equal chance to become the most promising candidate; then return to a and continue. In step a of each cycle, the LBM solutions of some unexpected subtrusses may be found simultaneously with those under consideration. Some of these subtrusses may be identified as unpromising when compared to the current UBGM and the LBM solution of the most promising surviving candidate is increased. In step b, on the other hand, a new UBGM may be found so that its value is decreased, and some additional unpromising subtrusses may also be identified. The recurrent application of these two steps finally yields the smallest UBGM one can find and it is considered an optimum design.

Development of the main searching algorithm, described in some detail in the sequel, was based primarily on the foregoing fundamental strategy and partially on convenience in generating the computer program. The algorithm will search a smallest UBGM within a given list of subtrusses which may contain only a preselected set of subtrusses or may consist of all possible subtrusses in case an over-all global minimum is sought. Candidates on the list with an equal number of omitted members are grouped together. These groups are ordered according to the number of omitted members, i.e., the group of subtrusses with one member omitted is labelled group 1, the group of subtrusses with two members omitted is labelled group 2, and so on. The algorithm proceeds as follows:

1) Use the NLP method to find a minimum of the primary truss with no members omitted by declaration. This furnishes the first UBGM. This step can be optionally omitted in which case the first UBGM is set equal to an extremely large value.

2) Use the simplex method to find the first LBM (i.e., without declaring the omission of any members by imposing $A_j^m = 0$ conditions). Store the optimal tableau and the solution of A_j^m. If some $A_j^m = 0$ in the solution, this is also the LBM solution for those candidates which have members omitted by declaration corresponding to those members that have vanished (i.e., $A_j^m = 0$) at the first LBM solution. The following steps 3–5 represent a cycle that is to be repeated when more than one group of subtrusses is to be searched.

3) Apply the DSA on the optimal tableau obtained in 2 to find

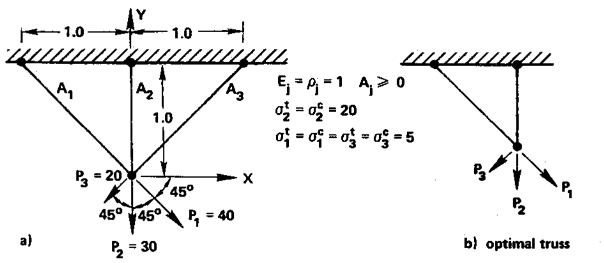


Fig. 5 Schmit and Sved's example of 3-bar truss.

the LBM solutions for the candidates in the lowest ordered group which have not yet been processed by this step (but skip the candidates whose LBM solutions have unexpectedly been found before). As in step 2 the LBM solutions thus obtained may also be the LBM solutions for some candidates in the higher groups, since some A_j^m not omitted by declaration may also vanish. a) If a candidate has its LBM less than the current UBGM, it will be temporarily classified as promising, and its solution for A_j^m is stored. These A_j^m will be used as starting values in the NLP problem in case this candidate is selected for seeking a smaller UBGM. b) If a candidate has its LBM equal to or greater than the current UBGM, it is classified as unpromising. Furthermore, if a candidate in a higher numbered group has a set of declared omitted members that contains those of an unpromising candidate, then the LBM of the candidate in the higher numbered group is larger as described in Sec. IV, so it is also classified as unpromising. All these unpromising candidates are deleted from the list.

Let W_c denote the smallest LBM found in 3 in the current cycle. Note that W_c is not necessarily the smallest LBM generated in processing the current group since the latter may have unexpectedly been found in a previous cycle.

4) Pick up from the accumulative list of promising candidates a configuration with an LBM that does not exceed W_c. Priority is given to a candidate with the smallest LBM, and with the lowest redundancy if there is more than one candidate having the smallest LBM. Find the NLP minimum for this candidate using the A_j^m stored in step 3 as the starting point. If the NLP minimum found is not less than the current UBGM, delete the candidate from the list,** if the minimum is less than the UBGM, replace it and examine the remaining "temporarily promising candidates." Delete from the list of promising candidates those configurations with LBM equal to or greater than the new UBGM, including candidates from higher groups for which it is apparent that the LBM cannot be less than the new UBGM.

Repeat this step until none of the surviving candidates have an LBM equal to or less than W_c.

5) Go to step 3 if there is a higher ordered group which has not been processed by that step, otherwise go to step 4 but forget W_c (i.e., set W_c extremely large).

The main searching algorithm terminates when the list of promising candidates is exhausted and the last UBGM obtained is taken to be the optimum design.

The main searching algorithm just described illustrates the central idea of how to employ bounding technique in structural design optimization. It is apparent that the main search strategy could be subject to a variety of more or less sophisticated modifications. For example, since the search for an LBM costs much less, one could find the LBM for more candidates (particularly those with less redundancy when only stress constraints are imposed) before choosing the most promising candidate for which to seek a reduced UBGM using the NLP algorithm. Also in selecting a promising candidate priority could be given to the candidate with the smallest "modified LBM" rather than to the one with the smallest LBM. The modification could be made by increasing the LBM a few preselected percent for each subtruss according to its degree of redundancy and configuration. No modifications of this sort have, however, been incorporated into the computer program developed to generate the numerical examples presented in the next section.

** This step assumes that the NLP technique finds the global minimum for the candidate configuration. In the absence of sufficient information to demonstrate that the NLP problem is convex it cannot be guaranteed that the NLP technique yields to global minimum.

VII. Numerical Results for Examples

A computer program based upon the algorithms described in previous sections has been used to solve several problems. Because of space limitations, only three illustrative examples are presented here. The problems presented in this section are not those which have been solved most successfully by the bounding technique but rather they represent selected results that illustrate some interesting points and suggest possible further improvement of the bounding technique approach. In each of the following examples, all members are assumed to be made from the same material and the stress and displacement limits are independent of loading condition. In each case, the results presented include the optimal design for the best configuration, the minimum weight design for the primary configuration, if different from the optimum and the first LBM solution presented for comparison. An asterisk is used to identify fully stressed designs. The executing times given are for the FORTRAN IV program on a UNIVAC 1108 machine.

A. Four Node, Three-Bar Truss

This example is taken from Ref. 9 and was used by Sved and Ginos[7] to point out the present problem. The truss is subject to three alternative loadings which are shown in Fig. 5a along with the stress limits. The displacement limits and the upper limits A_j^u are set high on purpose so that these constraints are inactive. Although $A_j^l = 0$, the results in Table 1 show that the NLP algorithm, starting with $A_j > 0$ (the primary configuration), did not find the global optimum which can only be reached by declaring that bar 3 is omitted (see Fig. 5b). Note that A_3 is not the smallest of the A_j eiher in the primary minimum or in the 1st LBM. Therefore, it is difficult to predict from these minima that bar 3 should be omitted. If the member with the smallest A_j in the primary optimum and in the first LBM is omitted by declaration, the result is min $W = 17.3137$ with $A_1 = 8$ and $A_3 = 4.2426$ which indicates that this is the worst configuration since for the subtruss with A_1 omitted by declaration min $W = 14.1421$ with $A_2 = 2.8284$ and $A_3 = 8$.

Table 1 Design results for example A

	A_1	A_2	A_3	W
Global optimum*	8.0	1.5	0.0	12.8137
Primary optimum	7.0241	2.1391	2.7554	15.9694
1st LBM	5.8787	0.75	2.1213	12.0637

B. Five Node, Nine-Bar Truss

The truss shown in Fig. 6 is subject to two alternative loading conditions. In view of the load and support conditions, a symmetrical optimum design is anticipated and only subtrusses with symmetrical configuration are considered. In view of the symmetry, the conditions $A_1 = A_2$, $A_3 = A_4$, $A_5 = A_6$, and $A_7 = A_8$ are imposed and consequently only one loading condition needs to be considered. Tables 2a and 2b display the results in the absence of and with active displacement constraints, respectively. The optimum design configuration for both cases is shown in Fig. 6b. An examination of Tables 2a and 2b indicates that in both cases the global optimum exhibits a considerable weight reduction from its corresponding primary optimum. Although symmetry considerations lead to operational consideration of only one load condition, the optimum designs achieved satisfy all constraints on the original two-load condition problem and the optimal truss is not necessarily a statically determinate

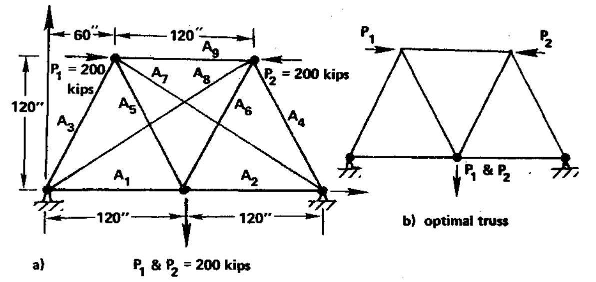


Fig. 6 A 9-bar truss.

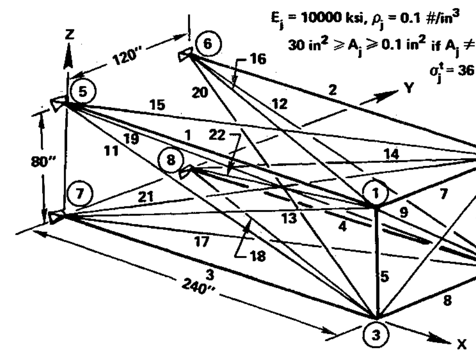


Fig. 7 22-bar cantilever truss.

structure. Again, it is observed that the members with the smallest A_j in both the primary design ($A_1 = A_2 = A_j^l = 2.0$) and the 1st LBM design ($A_1 = A_2 = 0$) do not correspond to the members omitted from the global optimum design ($A_7 = A_8 = 0$).

C. Eight Node, 22-Bar Space Truss

In the primary truss shown in Fig. 7 each joint is connected to every other joint except that members between the fixed support points 5, 6, 7, and 8 are excluded from the outset. The structure is subject to eight alternative load conditions. However, symmetry considerations are such that only three of these load conditions need to be considered after imposing symmetry conditions on the bar areas, i.e., $A_1 = A_2 = A_3 = A_4$, $A_5 = A_6$, $A_7 = A_8$, $A_9 = A_{10}$, $A_{11} = A_{12} = A_{13} = A_{14}$, $A_{15} = A_{16} = A_{17} = A_{18}$, $A_{19} = A_{20} = A_{21} = A_{22}$. Only symmetrical configurations are considered as candidates when seeking the global optimum design. The reduced set of three alternative loading conditions is shown in Table 3a where 1x denotes the x component of loading at joint 1, etc. Tables 3b and 3c display the results obtained in the absence of and with active displacement constraints, respectively. The results show that in the absence of active displacement constraints (see Table 3b) the optimal truss is a subtruss with members 9 and 10 omitted. It is observed that these are not the longest members in the primary truss, while in the previous example 2 the omitted members 7 and 8 were the longest members in the primary truss. Furthermore, in the absence of active displacement constraints it is found that there is another NLP minimum W= 758.5 lb (with members 1–4 and 19–22 omitted) in addition to three NLP minima close to the primary minimum weight. In the case with active displacement constraints (see Table 3c) the primary truss optimum design and the global optimum are identical and it is seen that none of the members are omitted. However, there are two NLP minima W= 1028.07 (with members 9, 10 omitted) and W= 1029.35 (with members 7, 8 omitted) as well as a third NLP minimum all very close to the global optimum shown. In both cases (without and with active displacement constraints) there are NLP minima so close to the global optimum obtained that they can also be considered optima in practice.

Table 2a Designs without active displacement constraints ($|u_i| \leq 3$ in.)

	$A_1 = A_2$, in.^{2}	$A_3 = A_4$	$A_5 = A_6$	$A_7 = A_8$	A_9	W, lb
Global optimum*	2.7778	13.9754	11.1803	0.0	11.1111	2476.25
Primary optimum	2.0	7.7588	6.3096	13.3955	10.7352	3208.80
1st LBM	0.0	11.355	5.5902	5.6337	12.5	2401.08

Table 2b Designs with active displacement constraints ($|u_i| \leq 1.5$ in.)[a]

	$A_1 = A_2$, in.^{2}	$A_3 = A_4$	$A_5 = A_6$	$A_7 = A_8$	A_9	W, lb
Global optimum	4.3620	13.9754	11.1803	0.0	12.4992	2630.99
Primary optimum	2.0	8.2847	6.4475	12.8115	11.7921	3223.48

[a] The first LBM is the same as that in Table 2a.

Table 3a The reduced loadings by symmetrical option

(Kips)	1x	1y	1z	2x	2y	2z	3x	3y	3z	4x	4y	4z
I	−20	0	−5	−20	0	−5	−20	0	−30	−20	0	−30
II	−20	−5	0	−20	−50	0	−20	−5	0	−20	−50	0
III	−20	0	35	−20	0	0	−20	0	0	−20	0	−35

Table 3b Designs without active displacement constraints ($|u_i| \leqq 10$ in.)

	A_1 to A_4	A_5, A_6	A_7, A_8	A_9, A_{10}	A_{11} to A_{14}	A_{15} to A_{18}	A_{19} to A_{22}	W, lb
Global optimum	1.3207	0.5505	0.5649	0.0	1.4088	2.1579	2.0610	754.18
Primary minimum	1.3049	0.5554	0.3257	0.5499	1.4543	2.2416	1.9433	763.24
1st LBM	2.1203	0.3688	1.0842	0.0	1.9082	2.1961	0.0	664.28

Table 3c Designs with active displacement constraints ($|u_i| \leqq 2$ in.)

	A_1 to A_4	A_5, A_6	A_7, A_8	A_9, A_{10}	A_{11} to A_{14}	A_{15} to A_{18}	A_{19} to A_{22}	W, lb
Global[a] optimum	2.6288	1.1624	0.3433	0.4231	2.7823	2.1726	1.9523	1024.80
An NLP min	2.6101	1.4234	0.587	0.0	2.7861	2.0891	2.0935	1028.07
An NLP min	2.5657	1.1331	0.0	0.6461	2.6738	2.1769	2.1613	1029.35

[a] Global optimum and primary minimum are identical.

Finally, Table 4 furnishes some information showing the efficiencies of the bounding technique and the searching algorithm used in the computer program.

Table 4 Computer run data for numerical examples

Example	A	B		C	
		I	II[a]	I	II[a]
No. of candidates	4	6		38	
LP problems solved	4	4	4	16	22
No. of LBM $\leqq$ optimum	2	3	3	13	20
NLP problem solved[b]	2[c]	3[c]	3	13	20
Executing time (sec)	0.71	1.48	1.61	45.34	68.21

[a] With active displacement constraints.
[b] The min. is one since the primary min. is sought.
[c] One solution is obtained from its LBM, directly in A and with slightly modified starting point in BI.

The number of NLP problems solved divided by the number of candidates indicates the efficiency of the bounding technique. Furthermore, the number of NLP problems solved divided by "No. of LBM $\leqq$ optimum" combined with the number of LP problems solved divided by "No. of candidates" provides a measure of the efficiency of the main searching algorithm (the lower the quotients, the higher the efficiencies). As mentioned previously, these examples are not chosen from the solved problems exhibiting the highest efficiency of the bounding technique, and, indeed, they are among those showing the lowest efficiency. The ones showing the highest efficiency are those whose optima coincide with their first LBM (i.e., a statically determinate subtruss without active displacement constraint). In these cases an NLP problem does not need to be solved unless the primary minimum is sought. However, based on the results of all problems solved to date the following tendencies are observed: 1) in a truss with a high bar-density configuration as in example 3, the primary minimum is often close to the global minimum, therefore, in order to save computational time, it may be wise, in practice, to use the primary minimum or try a few more subtrusses whose omitted members are contained in the set of $A_j = 0$ in the first LBM; 2) the present bounding technique is less efficient in problems with active displacement constraints. The second of the foregoing tendencies is not surprising. Indeed, the more active nonlinear constraints ignored in finding the LBM, the wider the gap between it and a real minimum. Thus, in displacement limited problems, the possibility of the LBM for any candidate configuration being below the current UBGM is enhanced. Hence, each candidate has a greater chance of being selected for search by the NLP algorithm in the quest for a lower UBGM. To overcome this inefficiency, if it is known in advance that the displacement limits govern the design, the authors suggest the following possible approaches: 1) use some form of modified LBM such as that suggested previously in Sec. IV or 2) use the LBM obtained by the following procedure. For each load condition maximize the external work subject to the stress and displacement limits[6] and then using the stresses obtained, minimize the weight subject to the equilibrium conditions. The largest minimum weight obtained by considering all of the load conditions, one at a time, would then be used as a LBM. However, it is noted that the first of the foregoing suggestions requires experience in order to preselect the percentage increase to be applied to the LBM of each subtruss. Furthermore, the theoretical basis of the second suggested approach and its effectiveness require futher investigation.

VIII. Discussion

The stability problem is not included in the previous discussion since it requires a more sophisticated treatment. Although buckling constraints for members in compression can be included in the NLP problem it is difficult to directly include these constraints in the LP problem for finding the LBM solutions. However, it is possible to indirectly guard against compression member buckling within the LP problem by appropriate choice of the allowable compressive stresses $\sigma_{jk}{}^c$ and the lower limits $A_j{}^l$ of the member cross-sectional areas. For example, when a thin walled circular tube is used with the compression force estimated in advance from static considerations, the formulas in Ref. 10 can provide a basis for selecting $\sigma_{jk}{}^c$ and $A_j{}^l$.

To accelerate convergence of the NLP algorithm used the iteration from a starting design to an upper bound minimum weight design is divided into three stages. Each stage employs an increasingly stringent termination criterion and the constraint tolerances are reduced for each successive stage. From the results of the NLP problems solved, it is found that the weight is always reduced very little (often about 1% or less) in the final stage of iteration in the NLP problem. Therefore, a great deal of computational time could be saved if the search for an NLP minimum were terminated when the weight is still considerably larger than the current UBGM (say 5%) at the beginning of the final stage of iteration.

In principle, the present approach can be extended to treat the problem where candidates are selected from different primary configurations. This, of course, would require a more sophisticated main searching algorithm and consequently a more complicated computer program in order to retain the efficiency of the method. Because of the mathematical similarity or equivalence, the present approach is also thought to be potentially applicable to some other kinds of structures, such as bar-shear panel representations.

References

[1] Cilley, F. H., "The Exact Design of Statically Indeterminate Frameworks, an Exposition of its Possibility, but Futility," *Transactions of the ASCE* 43, June 1900, pp. 353–407.

[2] Drymael, J., "The Design of Trusses and its Influence on Weight and Stiffness," *Journal of the Royal Aeronautical Society*, Vol. 46, 1942, pp. 287–308.

[3] Sved, G., "The Minimum Weight of Certain Redundant Structures," *Aust. Journal of Applied Sciences*, Vol. 5, 1954, pp. 1–9.

[4] Barta, J., "On the Minimum Weight of Certain Redundant Structures," *Acta Technica, Academiae Scientiarum*, Hungaricae, Vol. 18, No. 1-2, 1957, pp. 67–76.

[5] Pearson, C. E., "Structural Design by High Speed Computing Machines," *Conference on Electronic Computation of ASCE*, Kansas City, Nov. 1958, pp. 417–436. *Proceedings of the First Conference on Electronic Computation*, ASCE, New York, 1958, pp. 417–436.

[6] Dorn, W. S., Gomory, R. E., and Greenberg, H. J., "Automatic Design of Optimal Structures," *Journal de Mecanique*, Vol. 3, 1964, pp. 25–52.

[7] Sved, G. and Ginos, Z., "Structural Optimization under Multiple Loading," *International Journal of Mechanical Sciences*, Vol. 10, 1968, pp. 803–805.

[8] Zoutendijk, G., *Methods of Feasible Directions*, American Elsevier, New York, 1960.

[9] Schmit, L. A., Jr., "Structural Design by Systematic Synthesis," *Second Conference on Electronic Computation of ASCE*, Pittsburgh, Sept. 1960, pp. 105–132.

[10] Gerard, G., *Minimum Weight Analysis of Compression Structures*, New York University Press, Interscience, New York, 1956, pp. 18–19.

FEBRUARY 1972 AIAA JOURNAL VOL. 10, NO. 2

A Theoretical Study of the Generation of Atmospheric-Clear Air Turbulence

COLEMAN DUP. DONALDSON,* ROGER D. SULLIVAN,† AND HAROLD ROSENBAUM‡
Aeronautical Research Associates of Princeton Inc., Princeton, N.J.

This paper concerns a numerical study of the time history of the growth and decay of turbulence in the atmosphere. The study was motivated by a desire to better understand the behaviour of shear-generated turbulence in the Earth's atmosphere—the problem of clear air turbulence (CAT). The variant modeling technique is used to derive a closed set of equations that govern this phenomenon. The resulting set of equations allows, for the first time, a step-by-step numerical calculation of the growth of turbulent disturbances in atmospheric shear flows and the dependence of this growth on atmospheric instabilities.

Nomenclature§

A, B, C = const
a_2, a_3 = const
c, c_1, c_2, c_3, c_4 = const
c_p = specific heat at constant pressure
f = unknown function
g = acceleration due to gravity
g_{ij} = the metric tensor
K = $\langle u^{m\prime} u_m{}' \rangle$
k = conductivity
k_T = eddy conductivity
l = Prandtl's mixing length
p = pressure
q = $K^{1/2} = \langle u^{m\prime} u_m{}' \rangle^{1/2}$
q_i = heat-flow vector
Re_Λ = $\rho \Lambda q / \mu$
T = temperature
t = time
u, v, w = x, y, and z components of velocity
u_i = velocity vector
X = body force
x, y, z = Cartesian coordinates
x_i = general coordinate system
δ = boundary-layer thickness or scale of mean motion
$\delta_i{}^j$ = Kronecker delta
ε_{ij} = $\bar{u}_{i,j} + \bar{u}_{j,i}$
η = function defined in Eq. (66)
λ, Λ = scalar measures of length
μ, μ^* = first and second coefficients of viscosity
μ_T = eddy viscosity
ξ = function defined in Eq. (64)
ρ = density
σ = $(1/\rho_0)\, \partial \rho_0 / \partial z$
τ_{ij} = stress tensor
ϕ = gravitational potential
ω = $2[(g/T_0)(\partial \bar{T}/\partial z)]^{1/2}$

Subscript

0 = equilibrium conditions of the atmosphere

Superscripts

= departure from equilibrium
= mean
= departure from the mean

Presented as Paper 70-55 at the AIAA 8th Aerospace Sciences Meeting, New York, January 19–21, 1970; submitted May 4, 1971; revision received August 23, 1971. This research was supported by NASA under Contract NASW-1868.

Index categories: Atmospheric, Space, and Oceanographic Science; Jets, Wakes, and Viscid-Inviscid Flow Interactions.

* President and Senior Consultant. Associate Fellow AIAA.

† Vice President and Consultant. Member AIAA.

‡ Associate Consultant; presently Senior Consulting Scientist, Avco Systems Division, Avco Corporation, Wilmington, Mass. Member AIAA.

§ The notation of general tensor analysis is used. In particular, the summation convention is implied, and a comma preceding a subscript indicates covariant differentiation.