

**A STRUCTURAL OPTIMIZATION SOLUTION TO A
 BRANCH-AND-BOUND PROBLEM***

By

O. E. LEV

University of Illinois at Urbana-Champaign

Abstract. A simple algorithm, developed for a least-weight structural optimization problem, is used to force the selection of the same n components of the vectors X and Y , containing b elements ($b > n$) so that the objective function $\bar{L} \max_{x_i, y_i} \{|X|, |Y|\}$ is minimized subject to n equality constraints on each vector, $AX = b_1, AY = b_2$. The method has an obvious advantage over integer programming or branch-and-bound techniques that would, in this case, seek the best selection of n out of b elements which satisfy the constraints.

1. Introduction. The following problem, herein referred to as Problem A, arises in the design of least-weight pin-jointed trusses [4]:

Problem A:

$$\text{minimize } W_0 = \sum_{i=1} L_i \max_{1,2} \{|F_i^1|, |F_i^2|\}, \quad i \in B \quad (1)$$

$$\text{subject to } \tilde{N}F^1 = [\tilde{N}_T, \tilde{N}_R] \begin{bmatrix} F_T^1 \\ F_R^1 \end{bmatrix} = P^1, \quad T \cup R = B \quad (2)$$

$$\tilde{N}F^2 = [\tilde{N}_T, \tilde{N}_R] \begin{bmatrix} F_T^2 \\ F_R^2 \end{bmatrix} = P^2 \quad T \cap R = \emptyset$$

$$\text{and } F_K^1 = F_K^2 = 0, \quad k \in R \quad (3)$$

where the set B contains b elements and consists of two disjoint subsets to be found: R of (at least) r elements and T of (at most) n elements. The unknown vectors F^1 and F^2 and the given vector L contain b elements each; the matrix N ($b \times n$) and the n -element vectors P^1 and P^2 are also given.

The b elements of the set B correspond to the bars of a truss, which are assumed to develop forces F^1 and F^2 under two independent loadings P^1 and P^2 with components corresponding to the n displacement components of the truss joints. The forces and the loads are related by means of n equilibrium equations under each set of loads as shown in Eqs. (2). Superscripts denote loading conditions. In general $b \geq n$ is a condition required for stability, so that $r (= b - n)$ force components must be independently specified in order that the remaining n components can be evaluated. The truss has thus r degrees of freedom and is termed "indeterminate" to the r th degree, where

$$r = b - n \geq 0. \quad (4)$$

* Received March 28, 1975.

It is therefore possible to partition the forces F to a set of r "redundants" (R) and a complementary "tree" set (T) of n members such that

$$F_T^{1,2} = \tilde{N}_T^{-1}(P^{1,2} - \tilde{N}_R F_R^{1,2}) \quad (5)$$

which follows directly from Eqs. (2). Whenever Eq. (4) holds as an equality ($r = 0$) and no set R can be selected or equivalently $F_R = 0$, the second term in the parentheses of Eq. (5) vanishes and F_T is directly determined from the given loads using \tilde{N}_T which is ($n \times n$) and hence invertible in general. Such a truss is termed "statically determinate", since the equilibrium equations (statics) are sufficient to determine its forces. After the forces are evaluated the cross-sectional areas of the members are computed. Assuming that the stresses are not to exceed some fraction of the most severe of F^1 and F^2 (regardless of signs), the term $\max_{1,2} \{|F_i^1|, |F_i^2|\}$ is proportional to the cross-section of the i th element, so that each product $L_i \max_{1,2} \{|F_i^1|, |F_i^2|\}$ in Eq. (1), where L_i is the length of the bar, is proportional to its volume or weight.

Problem A expressed in words using the above terminology can thus be stated as follows: find the least-weight statically determinate truss with given geometry (N) under two given loading conditions (P^1 and P^2). Equivalent formulations in Sec. 3 suggest solutions using integer programming or branch-and-bound [5] techniques. These serve to highlight a simple algorithm which avoid the difficulties and limitations of such techniques and is central to this paper. In a previous work [1] Problem A, with the constraint of Eq. (3) relaxed, was shown to decompose into two simpler linear programming [3] problems. This decomposition is briefly presented in Sec. 2. The algorithm developed in Sec. 4 starts at the point where the decomposed optimal solutions have been obtained. Since upon introducing Eq. (3) these are no longer feasible, they are used as a lower bound and are modified step by step until the desired optimum is reached.

2. The decomposed problem. Consider the following problem, referred to as Problem B:

Problem B:

$$\text{minimize } W = \sum_{i \in B} L_i \max_{1,2} \{|F_i^1|, |F_i^2|\} \quad (1)$$

$$\text{subject to } \tilde{N}F^1 = P^1 \quad \text{and} \quad \tilde{N}F^2 = P^2. \quad (2)$$

Except for the absence of the constraint in Eq. (3), Problems A and B are identical. It follows that the optimal solution of Problem B, \bar{W} , is the smallest value that W_0 of Problem A will reach. Furthermore, the solution of the latter is a subset of the former. Since these facts are used to develop the algorithm in Sec. 4, the solution of Problems B, given in [1], will be briefly described.

Using the identity

$$\max \{|x|, |y|\} = \frac{1}{2} |x + y| + \frac{1}{2} |x - y|$$

the vectors F^S , F^D , P^S , P^D defined as

$$\begin{aligned} P^S &= \frac{1}{2}(P^1 + P^2); & P^D &= \frac{1}{2}(P^1 - P^2); \\ F^S &= \frac{1}{2}(F^1 + F^2); & F^D &= \frac{1}{2}(F^1 - F^2) \end{aligned} \quad (6)$$

can be substituted in Eq. (1) and Eq. (2), yielding

Problem B₁ :

$$\text{minimize } W = \sum L_i |F_i^S| + \sum L_i |F_i^D| = W^S + W^D \quad (7)$$

$$\text{subject to } \tilde{N}F^S = P^S \quad \text{and} \quad \tilde{N}F^D = P^D. \quad (8)$$

Problem B₁, in which the superscripts *S* and *D* stand for "sum" and "difference", can be decomposed into two independent problems, each of which corresponds to weight minimization under a single load P^S and P^D :

The decomposed problem:

$$\begin{array}{ll} \text{"sum" problem} & \text{minimize } W^S = \sum_i L_i |F_i^S|, \\ & \text{subject to } \tilde{N}F^S = P^S \end{array} \quad (9)$$

$$\begin{array}{ll} \text{"difference" problem} & \text{minimize } W^D = \sum_i L_i |F_i^D|, \\ & \text{subject to } \tilde{N}F^D = P^D. \end{array} \quad (10)$$

The optimal solution of Problem B, \bar{W} , can thus be obtained as the sum of the two solutions of the decomposed problem:

$$\bar{W} = \bar{W}^S + \bar{W}^D. \quad (11)$$

Similarly, the optimal forces \bar{F}^1 and \bar{F}^2 may be obtained using

$$\bar{F}^1 = \bar{F}^S + \bar{F}^D, \quad \bar{F}^2 = \bar{F}^S - \bar{F}^D \quad (12)$$

which is true by definition (Eq. (6)).

It should be noted that the above results, reviewed from [1], were obtained using linear programming methods by Hemp [6] and, without using these methods, by Nagtegaal and Prager [7, 8].

The optimal structures in the "sum" and "difference" problem are both statically determinate. This is a well-known result in structure design. It can also be reached using the formal theory of linear programming [2]. Since there are n equality constraints, it follows from Eq. (4) that (at least) $r = b - n$ bars must have vanishing forces in the basic optimal solution as they correspond to nonbasic variables.

Let S and D be the n -bar sets which form the optimal bases and hence have non-zero forces \bar{F}^S and \bar{F}^D respectively; the optimal truss of Problem B can be obtained by forming the union of these sets:

$$S \cup D = B. \quad (13a)$$

Here it should be remarked that a certain bar may have vanishing forces in both Eqs. (9) and (10), i.e., for some bar L , $\bar{F}_L^S = \bar{F}_L^D = 0$. It is assumed that such a bar will be eliminated from the original set, the dimensionality of the problem will be reduced and the set B can be redefined still as having b bars, where b has been properly adjusted. The overlap set SD of S and D is defined as the set containing nonvanishing forces in either Eq. (9) or Eq. (10):

$$SD = S \cap D \quad (13b)$$

Similarly, the complementary r -bar sets of vanishing forces will be designated by S_0 and E_0 . It follows that

$$S \cup S_0 = B, \quad S \cap S_0 = \emptyset \tag{13c}$$

$$D \cup D_0 = B, \quad D \cap D_0 = \emptyset. \tag{13d}$$

The above relationships among the sets are illustrated in Fig. 1. Physically, the optimal b -bar solution \bar{W} of Problem B contains:

- a. a set SD of $(n - r)$ bars common to both optimal solutions of the decomposed problems, $\bar{F}_i^S \neq 0, \bar{F}_i^D \neq 0, i \in SD$;
- b. a set S_0 of r bars with vanishing forces under $P^S, \bar{F}_j^S = 0, \bar{F}_j^D \neq 0, j \in S_0$;
- c. a set D_0 of r bars with vanishing forces under $P^D, \bar{F}_k^S \neq 0, \bar{F}_k^D = 0, k \in D_0$.

The above results become significant in the next sections.

3. Branch-and-bound and integer programming formulations. Restating the objective in Problem A, the best partition of the set B into the disjoint sets R and T , defined in the introduction, is to be found such that F_R^1 and F_T^2 computed from Eq. (5) with $F_R^1 = F_T^2 = 0$ as required by Eq. (3) yield the desired minimum \bar{W}_0 of W_0 .

Using branch-and-bound methods [5], an efficient technique could probably be found to exhaust all the possibilities of selecting the r elements of R from the $b (= n + r)$ elements of B , which amount to $b!/r! n!$. Such a technique, with its dimensionality drawback, would be further complicated by the fact that not every selection of r members qualifies as a set of redundants for reasons of structural stability.

Perhaps because integer programming [5] does not offer any formal known technique to solve such a problem, the formulation of Problem A as an *IP* problem is interesting by itself:

$$\begin{aligned} \text{minimize } W_0 &= \sum_{i=1}^b L_i \delta_i \max_{1,2} \{|F_i^1|, |F_i^2|\} \\ \text{subject to } \sum_b N_{ij} \delta_i F_i^{1,2} &= P_j^{1,2} \quad j = 1, 2, \dots, n \\ \text{and } \sum_{i=1} \delta_i &= n, \text{ where } \delta_i = 1 \text{ or } 0. \end{aligned} \tag{14}$$

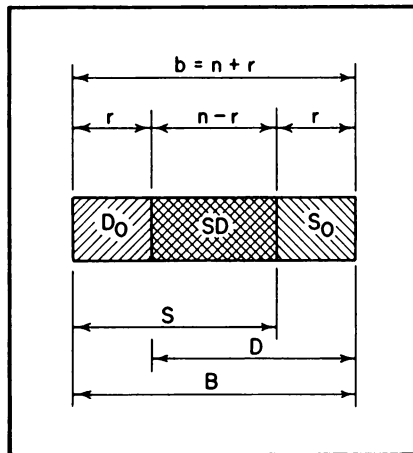


FIG. 1. The disjoint subsets of B .

When we use the decomposition shown in Sec. 2, Eqs. (14) become

$$\text{minimize } W_0^S = \sum L_i \delta_i^S |F_i^S| \text{ subject to } N_{ij} \delta_i^S F_i^S = P_j^S; \quad (15)$$

$$\text{minimize } W_0^D = \sum L_i \delta_i^D |F_i^D| \text{ subject to } N_{ij} \delta_i^D F_i^D = P_j^D, \quad (16)$$

where $j = 1, 2, \dots, n, i \in B$, and

$$\delta_i^S = \delta_i^D, \text{ where } \delta_i^S \text{ and } \delta_i^D \text{ are 1 or 0.} \quad (17)$$

It is remarkable that Eqs. (15) and (16) represent the same objective function and same type of constraint with a modified vector of constants, as shown for simplicity below:

$$\text{minimize } \bar{L} |X| \text{ subject to } AX = b_1, \quad (15a)$$

$$\text{minimize } \bar{L} |Y| \text{ subject to } AY = b_2. \quad (16a)$$

The additional constraint, Eq. (17), requires the n elements of X and Y in the bases to have identical indices, thus forcing the same determinate n -bar set to be selected in each case.

4. A simplex-in-reverse algorithm. In this section an algorithm based on modifying the simplex [3] optimal solution of Problem B is developed to solve Problem A. \bar{W} , the optimal b -element solution of the former, is used as the greatest lower bound of W_0 , which is any feasible solution of the latter containing only n elements. Since $\bar{W}_0 = \min W_0$ is sought, it can be obtained by minimizing the difference $W_0 - \bar{W}$. This fact will be used as a criterion for eliminating r members out of the given b .

Consider the two final linear programming simplex tableaux corresponding to the optimal basic solutions \bar{W}^S and \bar{W}^D of the decomposed problem (Eqs. (9) and (10)). Fig. 2 shows them after rearrangement in sets which were defined in Sec. 2. Thus, each tableau contains three disjoint subsets (Fig. 1) S_0 , SD and D_0 . For convenience, these subsets are referred to as follows:

$$\begin{aligned} \text{in the "sum" tableau} \quad & S_0 = S_1, \quad SD = S_2, \quad D_0 = S_3, \\ \text{in the "difference" tableau} \quad & D_0 = D_1, \quad SD = D_2, \quad S_0 = D_3. \end{aligned} \quad (18)$$

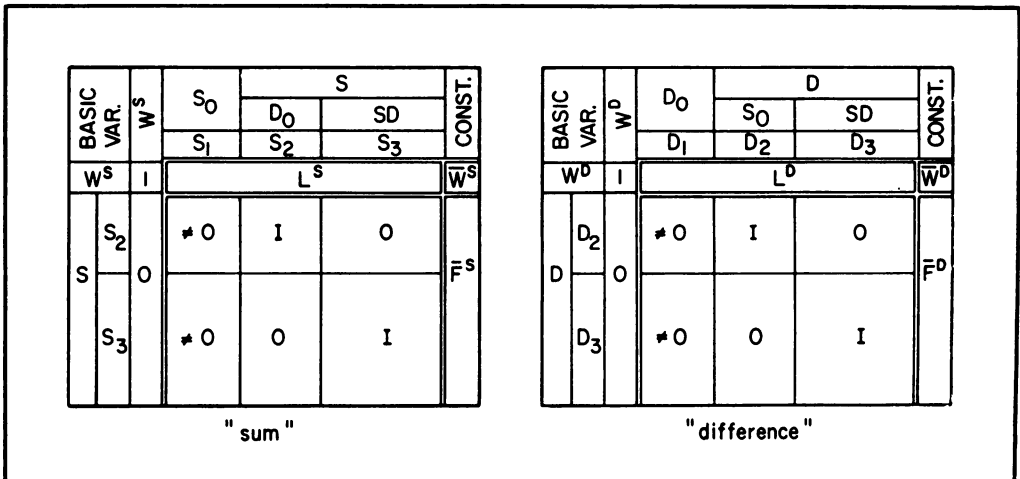


FIG. 2. The final simplex "decomposed" tableaux.

Thus $S_2 \cup S_3$ and $D_2 \cup D_3$ correspond to S and D respectively and contain the basic variables, while S_1 and D_1 correspond to S_0 and D_0 and contain the nonbasic (zero) variables.

The two tableaus are merged, as shown in Fig. 3, to become the initial tableau for the procedure specified below. It should be remarked that the initial tableau represents an unfeasible solution for Problem A since it contains $r = b - n$ extra elements.

4.1 The algorithm.

1. (a) Select a member $\alpha^s \in S_3$ as a candidate for leaving the basis in the "sum" tableau.
 (b) Select, as a candidate to enter the basis, the member $\beta^s \in S_1$ that produces the least increase ΔW_0^s when entering the basis, thus replacing α^s .
2. Repeat step 1 on the "difference" tableau, tentatively selecting $\alpha^d \equiv \beta^s \in D_3$ to leave and $\beta^d \in D_1$ to enter, thus producing ΔW_0^d .
3. Compare ΔW_0^s and ΔW_0^d and change basis according to step 1 if $\Delta W_0^s \leq \Delta W_0^d$, otherwise change basis according to step 2.
4. Compare the sets S_1 and D_1 after rearranging the variables as a result of step 3. If $S_1 \neq D_1$ go to step 1, otherwise stop.

4.2 Discussion.

1. Since S_1 and D_1 are mutually exclusive, and since by definition $D_3 = S_1$ and $S_3 = D_1$, a member α leaving D_3 (or S_3) becomes immediately a member of D_1 (or S_1), so that S_1 and D_1 both contain α after a single iteration. Similarly, since $S_2 = D_2 = SD$, β , which is exclusive to either S or D , becomes common to both following the exchange and hence augments the set SD . This implies that after a single iteration a member α is eliminated while another, β , becomes common to both bases.

2. The change of basis in step 3 corresponds to moving from a vertex i in the constraints hypersurface, which can be shown to be convex, to the next closest vertex j such that $(W_{0j} - W_{0i})$ is minimum. In particular, after the first iteration the weight W_{01} closest to \bar{W} is attained. Thus, the objective function W_0 keeps increasing while ascending a minimal amount relative to the initial position. In this sense the algorithm resembles the simplex method performed in reverse.

3. The procedure is repeated until r bars selected from either S_1 or D_1 are eliminated so that all the n remaining elements are in the common set SD , thus leaving S_3 and D_3 empty. Since the vertex reached then is the closest to \bar{W} with n bars, it is the desired optimal solution \bar{W}_0 .

4. Note that whenever the basis changes according to step 1, β^s need not be considered further for elimination. Similarly, step 2 implies that α^s should remain in the basis and β^d should not be eliminated. This indicates that it is sufficient to perform the algorithm only r times and that this can be done considering either S_1 (or D_1) for entering or S_2 (or D_2) for leaving.

5. Finally, it should be noted that it is possible to show that members of the set SD (the initial and obviously the one in any iteration) are not good candidates for elimination, although this is intuitively obvious. The proof hinges on the fact that variation of SD variables corresponds to motion along the hyperplanes and not the edges of the hypersurface of constraints. This in effect limits the search for the leaving variable considerably, as shown in the shaded area in Fig. 3.

		β^s			β^D				
	W_0	S_0	S		D_0	D		CONST.	
			D_0	SD		S_0	SD		
		S_1	S_2	S_3	D_1	D_2	D_3		
	W_0	I	L^s			L^D			\bar{W}
a^s	S_2	0	I	O	O			\bar{F}^s	
	S_3	$\neq 0$	O	I	O				
	S	O	O			O			\bar{F}^D
a^D	D_2	O			0	I	O		
(β^s)	D	O	O			$\neq 0$	O	I	
	D_3	O			O				

PIVOT FOR BASIS CHANGE TO BE FOUND IN SHADED AREA

FIG. 3. The initial "simplex-in-reverse" tableau.

5. Conclusions. The algorithm presented in Sec. 4 avoids the limitations of integer programming and branch-and-bounds procedures. The initial tableaus may be conveniently generated by library simplex routines and the procedure described very easily programmed. From the structural point of view, it has yet a further advantage. An optimal statically indeterminate truss, with decreasing degrees of freedom from r (\bar{W}) to zero (\bar{W}_0), is generated as a by-product at every iteration.

REFERENCES

- [1] W. R. Spillers and O. Lev, *Design for two loading conditions*, Int. J. Solids Structures. 7, 1261-1267 (1971)
- [2] W. R. Spillers and J. Farrell, *On the analysis of structural design*, J. Math. Anal. Appl. 25, 285-295 (1969)
- [3] M. Simonnard, *Linear programming*, Prentice Hall, Englewood Cliffs, New Jersey, 1966
- [4] O. E. Lev, *Optimum choice of determinate trusses under multiple loads*, Proc. Amer. Soc. Civil Eng., J. Struct. Div., 1977 (to appear)
- [5] F. S. Hillier and G. J. Lieberman, *Operations research*, 2nd ed., Holden Day, 1974
- [6] W. S. Hemp, *Abstract of lecture course, optimum structures*, 2nd ed., Engineering Laboratory, University of Oxford, 1968
- [7] J. C. Nagtegaal and W. Prager, *Optimal layout of a truss for alternative loads*, Int. J. Mech. Scis. 15, 583-592 (1973)
- [8] J. C. Nagtegaal, *A superposition principle in optimal plastic design for alternative loads*, Int. J. Solids Structures 9, 1465-1471 (1973)