Computers & Structures, Vol. 5, pp. 249–255. Pergamon Press 1975. Printed in Great Britain

OPTIMAL DESIGN OF MULTI-STAGE STRUCTURES: A NESTED DECOMPOSITION APPROACH†

JAMES K. HO
Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California

(Received 14 May 1974)

Abstract—A new approach to the optimal design of multi-stage structures is presented. The problem considered is that of optimal sizing and configurations for minimal weight, using limit analysis under a single loading condition. It is formulated as a large-scale linear program with the staircase structure. Using the technique of nested decomposition, design problems too large for conventional methods can be solved. Computational results are reported on examples in planar truss design.

1. INTRODUCTION

The problem of designing planar trusses for minimal weight over a class of feasible member sizes and configurations has been studied in [1]. The design is based on limit analysis subject to a single set of loads. Under these circumstances, the optimal trusses will be statically determinate, and, therefore fully-stressed, i.e. each member will be stressed to its maximum allowable value. The design problem is then a linear program [2]. Extensions to include multiple loading conditions and elastic buckling constraints [3] through the use of non-linear programming methods [4, 5] have been treated in [6] and [7]. In this paper, we specialize the formulation in [1] for multi-stage structures. For simplicity in presentation we focus on planar trusses (two dimensional pin-jointed skeletal structures). The results are easily translated into problems for space frames (three dimensional rigidly-jointed skeletal structures) [8]. The linear programs then have the staircase structure [9]. As such, they can be solved with the nested decomposition algorithm developed in [10]. This algorithm solves a sequence of smaller subproblems, each corresponding to one stage in the structure. When an optimal coordination among these subproblems obtains, an optimal solution to the original problem is constructed. Using a FORTRAN IV implementation [11], it is found that the present approach is computationally more efficient than conventional linear programming techniques [12]. Moreover, problems of larger scale can now be solved.

2. THE DESIGN PROBLEM FOR SINGLE-STAGE STRUCTURES

The designer is given a set of N admissible joints whose locations in space are specified. In the case of planar trusses, let (x_i, y_i), $i = 1, \ldots, N$, be the coordinates of these joints. See Fig. 1. All structures considered in a given problem will select joints from this set and only from this set. A single loading condition comprising an external force of P_i at joint i is given. Let $P_i = (P_{ix}, P_{iy})$ where P_{ix}, P_{iy} are the components of P_i in the x and y coordinates respectively.

Each pair of distinct admissible joints define an admissible bar. Therefore the number of admissible bars is given by

$$K = N(N-1)/2. \tag{1}$$

We assume for the moment that all bars are made of the same material. If bar b_k, $k = 1, \ldots, K$, connects joint i and joint j, where $i < j$, we define its positive direction as that of the vector $(x_j - x_i, y_j - y_i)$. We say that b_k leaves joint i and enters joint j. The length of b_k is given by

$$L_k = [(x_j - x_i)^2 + (y_j - y_i)^2]^{1/2}. \tag{2}$$

Let α_k and β_k be the angles between the positive direction of b_k and the positive x and y axes respectively. See Fig. 2. Then the direction cosines δ_k and γ_k of b_k are given by

$$\delta_k = \cos \alpha_k = (x_j - x_i)/L_k, \tag{3}$$

$$\gamma_k = \cos \beta_k = (y_j - y_i)/L_k. \tag{4}$$

Let s_k be the cross-sectional area of bar b_k and u_k the force it carries, taken positive in tension. The structure consisting of all admissible bars b_k, $k = 1, \ldots, K$, is called the ground structure. In general, we can describe the equilibrium condition at each joint by two equations, corresponding to the components of the applied force, P_{ix} and P_{iy}, respectively. Therefore a total of $2N$ equilibrium equations can be written:

$$\sum_{k=1}^{K} a_{lk} u_k = P_l \qquad (l = 1, \ldots, 2N) \tag{5}$$

where we have relabelled $(P_{1x}, P_{1y}, \ldots, P_{ix}, P_{iy}, \ldots, P_{Nx}, P_{Ny})$ as $(P_1, \ldots, P_l, \ldots, P_{2N})$. The coefficients in the transformations are defined as follows.

If l corresponds to ix for some $1 \le i \le N$, then

$$a_{lk} = \begin{cases} -\delta_k & \text{if } b_k \text{ leaves joint } i \\ +\delta_k & \text{if } b_k \text{ enters joint } i \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

If l corresponds to iy for some $1 \le i \le N$, then

$$a_{lk} = \begin{cases} -\gamma_k & \text{if } b_k \text{ leaves joint } i \\ +\gamma_k & \text{if } b_k \text{ enters joint } i \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

A system of self-equilibrating forces in the plane satisfy three equilibrium conditions. Hence, for the ground

†Research and reproduction of this report were partially supported by the U.S. Atomic Energy Commission Contract AT(04-3)-326 PA, No. 18; National Science Foundation, Grant GP 31393X2, and Grant GS 30377A1.

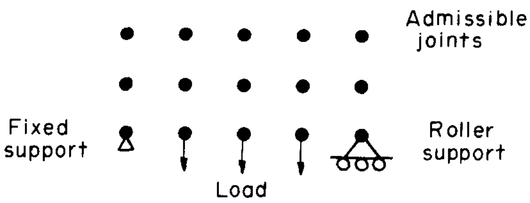

Fig. 1. A single-stage design problem.

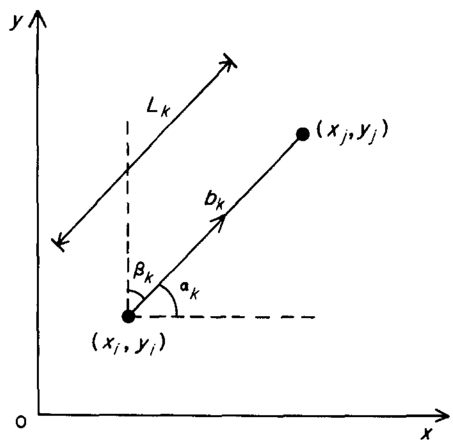

Fig. 2. The basic geometry.

structure to be rigid, exactly

$$m = 2N - 3 \tag{8}$$

of the $2N$ equations in (5) must be linearly independent. This implies that for any system of self-equilibrating external forces applied at the joints, a system of bar forces u_k, $k = 1, \ldots, K$, exists so that equilibrium is maintained. Assuming a rigid ground structure, two of the three equations to be deleted may be attributed to a fixed support at one joint which then has no degree-of-freedom. The third equation may be accounted for by a roller support for another joint which is allowed to move in only, say, the x direction.

Letting $A = [a_{lk}]$, $l = 1, \ldots, m$, $k = 1, \ldots, K$; $u = (u_1, \ldots, u_k)$ and $P = (P_1, \ldots, P_m)$ we write the subsystem of (5) which has rank m as

$$Au = P. \tag{9}$$

Next, assume that the yield stress $\sigma > 0$ of the bar material is given and is the same in tension and compression (otherwise use a linear transformation to reduce to the present case). This implies the constraints

$$-\sigma s_k \le u_k \le \sigma s_k, \quad k = 1, \ldots, K \tag{10}$$

which states that the stress in each bar does not exceed the allowable yield stress.

If ρ is the density of the material, the total weight of the structure is given by

$$W = \rho \sum_{k=1}^{K} s_k L_k. \tag{11}$$

†Unless otherwise stated, all vectors are columns; a prime denotes a transpose.

The design problem is to minimize W over $\{s_k\}$ and $\{u_k\}$ subject to (9) and (10). However, observe that any optimal solution will be fully-stressed, i.e.

$$|u_k| = \sigma s_k, \quad k = 1, \ldots, K. \tag{12}$$

Otherwise, some s_k can be reduced, hence lowering the value of W, while (9) and (10) still hold. Therefore, the design problem for a single-stage structure is to

$$\begin{aligned} &\text{minimize} && \bar{W} = \frac{\rho}{\sigma} \sum_{k=1}^{K} L_k |u_k| \\ &\text{subject to} && \sum_{k=1}^{K} a_{lk} u_k = P_l, \quad l = 1, \ldots, m. \end{aligned} \tag{13}$$

Letting $L = (L_1, \ldots, L_K)$, $|u| = (|u_1|, \ldots, |u_k|)$, A and P as before, and $(\rho/\sigma) = 1$ by a change of units, (13) can be written as†

$$\begin{aligned} &\text{minimize} && L'|u| \\ &\text{subject to} && Au = P. \end{aligned} \tag{14}$$

To put (14) into the canonical form of linear programming, we use the transformation

$$u = u^+ - u^-$$
$$|u| = u^+ + u^-$$

where $0 \le u^+ = \max[0, u]$, $0 \le u^- = -\min[0, u]$, the maximum and minimum operation being taken componentwise. Equation (14) is then equivalent to

$$\begin{aligned} &\text{minimize} && (L, L)'(u^+, u^-) \\ &\text{subject to} && [A, -A]\begin{pmatrix} u^+ \\ u^- \end{pmatrix} = P, \quad u^+, u^- \ge 0. \end{aligned} \tag{15}$$

Suppose (u^+, u^-) is an optimal solution to (15), then an optimal design of minimum weight is obtained by prescribing the following cross-sectional area for each bar:

$$s_k = \frac{|u_k^+ - u_k^-|}{\sigma}. \tag{16}$$

The optimal configuration is obtained by retaining only those b_k in the ground structure for which either u_k^+ or u_k^- is basic. See Fig. 3. Note that some u_k^+ or u_k^- may be basic but at zero value. Then s_k should be assigned an infinitesimal value $\epsilon > 0$ to achieve rigidity. In such cases only ϵ-optimal rigid structures are realizable. Also, we have relaxed a set of compatibility conditions on the bar forces as governed by Hooke's law. However, it can be shown that the optimal solution to (15) satisfies these conditions. This and other characteristics of the optimal

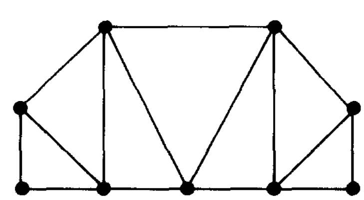

Fig. 3. Example of an optimal configuration.

solution as well as an interpretation of the dual problem to (15) are treated in [1].

3. THE DESIGN PROBLEM FOR MULTI-STAGE STRUCTURES

The structural designer is given a set of admissible joints as in Section 2. However, a partition of the admissible joints into an ordered set of stages is also

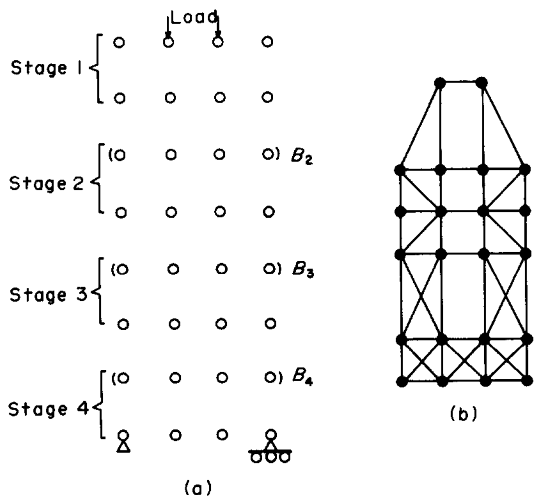

Fig. 4(a). A multi-stage design problem. (b) Example of an admissible structure.

specified. See Fig. 4(a). This is accompanied by the following conditions on the ground structure:

(i) within each stage, every pair of distinct joints define an admissible bar;

(ii) let stage t be separated from stage $t+1$ by the set B_{t+1} of boundary joints in stage $t+1$; then there is an admissible bar from each joint in stage t to each joint in B_{t+1};

(iii) there is no other admissible bar connecting the joints.

An example of a structure composed of a subset of the admissible bars for the joints in Fig. 4(a) is shown in Fig. 4(b). We remark that although rectangular grids are used as examples throughout this paper, the pattern of admissible joints can be perfectly arbitrary.

Denoting the number of joints in stage t by N_t and that in B_t by $|B_t|$, the number of admissible bars in stage t is given by

$$K_t = \frac{N_t(N_t-1)}{2} + N_t|B_{t+1}|, \quad t = 1, \dots, T; \qquad |B_{T+1}| = 0. \tag{17}$$

Therefore, the total number of admissible bars is

$$K = \sum_{t=1}^{T} K_t = \sum_{t=1}^{T}\left[\frac{N_t^2}{2} - \frac{N_t}{2} + N_t|B_{t+1}|\right]; \tag{18}$$

whereas, the total number of admissible bars in the ground structure of the corresponding single-stage structure with $N = \sum_{t=1}^{T} N_t$ joints would be

$$\bar{K} = \frac{N(N-1)}{2} = \sum_{t=1}^{T}\left[\frac{N_t^2}{2} - \frac{N_t}{2}\right] + \sum_{\tau\neq\sigma} N_\tau N_\sigma + \prod_{t=1}^{T} N_t. \tag{19}$$

†The notations used are different from those in previous sections.

Noting that $|B_t|/N_t \leq 1$ and comparing (18) and (19) we have

$$K < \bar{K}. \tag{20}$$

Hence the multi-stage feature results in a simplification in the ground structure.

The assumption of stages is often more realistic than that of a general ground structure. This is obviously true when different portions of the structure are to be built with different materials, or at different times and places. Functional or architectural requirements also give rise to multi-stage structures. Examples are multi-storey frames, transmission towers and multi-span bridges.

The problem of minimal weight design under a single loading condition can again be stated as a linear program similar to (15). Let us examine the constraints more carefully. Arrange the variables and constraint equations so that those corresponding to admissible bars in stage 1 come first, followed by those of stage 2, and so on up to stage T. Then the stage t variables have non-zero coefficients only in the stage t constraints (corresponding to joints in stage t) and those constraints in stage $t+1$ corresponding to boundary joints in B_{t+1}. The overall constraint matrix in (15) therefore exhibits the staircase structure shown in Fig. 5, where non-zeros occur only in the shaded regions.

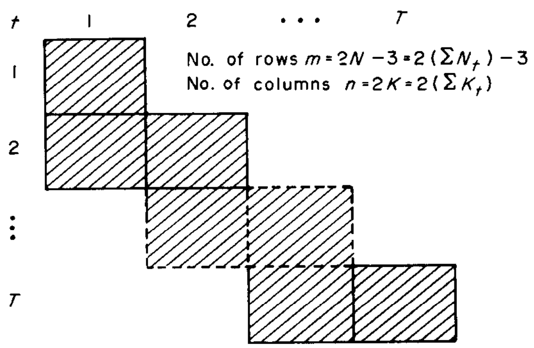

Fig. 5. Profile of the constraint matrix in the multi-stage design problem.

When the number of stages is large or there are many admissible joints in each stage, the resulting linear program can be too large for a direct simplex approach. In the next section we describe an algorithm which is designed to solve large-scale problems of this type, henceforth known as staircase linear programs.

4. NESTED DECOMPOSITION OF STAIRCASE LINEAR PROGRAMS

Consider the following staircase linear program:†

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T} c_t x_t \\ &\text{subject to} \quad A_1 x_1 = d_1 \\ &\qquad B_{t-1}x_{t-1} + A_t x_t = d_t \\ &\qquad\qquad x_t \geq 0,\ t = 1, \dots, T \end{aligned} \tag{21}$$

where x_t is $n_t \times 1$, A_t is $m_t \times n_t$, B_t is $m_{t+1} \times n_t$, c_t is $1 \times n_t$ and d_t is $m_t \times 1$ in dimension.

Our approach is to decompose the T-stage problem (21) into an ordered set of T independent subproblems, coordinated by communication of prices and proposals between adjacent subproblems. An iterative procedure is then used to adjust the prices and proposals for the subproblems until an optimal coordination (if one exists) is achieved. An optimal solution to (21) can be reconstructed from the decomposed system. It is shown in [10] that a repeated application of the Dantzig–Wolfe decomposition principle for linear programs [14] on (21) results in the following system of subproblems† SP_t^k, $t = 1, \ldots, T$, on the kth cycle of the algorithm:

$$\text{minimize} \quad (c_t - \pi_{t+1}^k B_t)x_t^k + \sum_{j=1}^{k-1} p_t^j \lambda_t^{jk} = z_t^k \tag{22}$$

$$\text{subject to} \quad A_t x_t^k + \sum_{j=1}^{k-1} q_t^j \lambda_t^{jk} = d_t \tag{23}$$

$$SP_t^k \quad \sum_{j=1}^{k-1} \delta_t^j \lambda_t^{jk} = 1 \tag{24}$$

$$x_t^k, \lambda_t^k \geq 0$$

where

π_{t+1}^k, the dual variables to (23) in SP_{t+1}^k, are the prices from SP_{t+1}^k to SP_t^k;

$$\begin{pmatrix} p_t^k \\ q_t^k \end{pmatrix} = \begin{pmatrix} c_{t-1}x_{t-1}^k + \sum_{j=1}^{k-1} p_{t-1}^j \lambda_{t-1}^{jk} \\ B_{t-1}x_{t-1}^k \end{pmatrix} \text{ is the proposal‡ from } SP_{t-1}^k \text{ to } SP_t^{k+1}, \tag{25}$$

$$\delta_t^k = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \text{ if } (x_{t-1}^k, \lambda_{t-1}^k s) \text{ is an } \begin{Bmatrix} \text{extreme point} \\ \text{extreme ray} \end{Bmatrix} \text{ solution to } SP_{t-1}^k. \tag{26}$$

The following interpretations of the proposals and prices serve to motivate our decomposition approach. To make stage t independent of x_{t-1}, we replace the latter by proposals from stage $t-1$. By choosing the proper combination of such proposals through the weights λ_t^{jk} we achieve the same effect as including x_{t-1} explicitly in the stage t constraints. At the same time, we relieve x_t from the constraints in stage $t+1$ by the use of prices from stage $t+1$. Through the term $-\pi_{t+1}^k B_t$ in (22), the proper set of prices can force x_t to satisfy the constraints in stage $t+1$ implicitly. Hence, the decomposition is complete.

In the multi-stage design problem, x_t represents the member forces in stage t, while the constraints express equilibrium conditions of the joints. Physically, a proposal to the stage t subproblem can be viewed as a substructure in the first $t-1$ stages which satisfies the equilibrium conditions in those stages. The prices for stage t can be regarded as penalties for perturbing the equilibrium of joints in stage $t+1$. Hence, the task of subproblem t is to choose a weighted combination of the proposed substructures, extend it to include stage t and send the result to stage $t+1$. To make this choice, penalties have to be incorporated into the original weight criterion so as to suppress the generation of proposals incompatible with the latest equilibrium conditions in stage $t+1$. As a result of this operation, stage t achieves a new state of equilibrium with new penalties for stage $t-1$. The algorithm in the next section is a finite scheme to generate an optimal price-proposal coordination among the subproblems from which an optimal solution to the original problem can be constructed.

The main advantage of our approach is that the subproblems are significantly smaller than the original problem, so that each can be solved with relative ease. Moreover, problems too large for conventional approaches may now be solved. A potential drawback is that the subproblems may have to be resolved many times before an optimal coordination among them is attained.

5. THE STAIRCASE ALGORITHM

We assume that a phase 1 procedure§ is used to obtain initial feasible subproblems SP_t^1, $t = 1, \ldots, T$. Phase 2 of the staircase algorithm consists of the following steps. See Fig. 6.

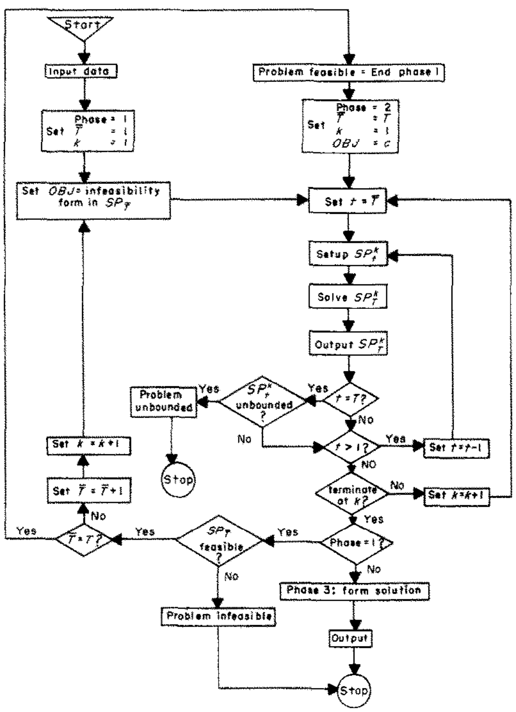

Fig. 6. Flow diagram of the staircase algorithm.

Step (i): Set $k = 1$.
Step (ii): Set $t = T$.
Step (iii): Solve SP_t^k.
If $t < T$, send proposal to SP_{t+1}^{k+1}.
If $t > 1$, send prices to SP_{t-1}^k.
If $t = T$ and subproblem is unbounded, stop: (21) is unbounded from below.
If $t = 1$, go to Step (v).
Step (iv): Set $t = t - 1$. Return to Step (iii).
Step (v): If $z_T^k = \sum_{t=1}^{T} \pi_t^k d_t$, a minimum is achieved, go to Step (vi). Otherwise, set $k = k + 1$ and return to Step (ii).
Step (vi): Set $t = T$, $y_T = x_T^k$, compute $d_T - A_T y_T$.
Step (vii): Set $t = t - 1$. Solve the following system‖

†For SP_1^k, delete the terms involving λ_1^{jk} and equation (24); for SP_T^k, delete the terms $\pi_{T+1}^k B_T$ from (22).

‡$p_2^k = c_1 x_1^k$.

§For $t = 1, \ldots, T$, start with an artificial basis for SP_t^1. Use Steps (i)–(v) on the first t stages in (21) to minimize the sum of the artificial variables in SP_t^1.

‖It is understood that the terms involving μ_t are deleted from SY_1.

$$SY_t:\begin{cases}\text{minimize} & c_t y_t + \sum_{j=1}^{k-1} p_t^j \mu_t^j \\ \text{subject to} & A_t y_t + \sum_{j=1}^{k-1} q_t^j \mu_t^j = d_t \\ & B_t t_y = d_{t+1} - A_{t+1} y_{t+1} \\ & \sum_{j=1}^{k-1} \delta_t^j \mu_t^j = 1 \\ & y_t, \mu_t \geq 0. \end{cases} \tag{27}$$

If $t = 1$, stop. Otherwise compute $d_t - A_t y_t$. Return to Step (vii).

The following results are proved in [10]:

(a) the staircase algorithm terminates in a finite number of cycles;

(b) if (21) is feasible and bounded from below, $y = (y_1, \ldots, y_T)$ obtained in steps (vi)–(vii) (known as phase 3) is an optimal solution.

We remark that the optimal solution obtained in phase 3 of the staircase algorithm is, in general, not a basic solution to the original problem (21). However, for multi-stage design problems it is a simple matter to reduce an optimal solution to a basic optimal solution by grouping together linearly dependent members. The resulting optimal solution then represents a statically determinate optimal structure.

6. COMPUTATIONAL ASPECTS

The data of the subproblems can be stored out-of-core. SP_t^k is read into core only when its solution is called for. Therefore, the core requirement is essentially that for a typical subproblem. To further economize on core size and computational efforts, advanced linear programming techniques should be employed to solve SP_t^k. We have adopted a scheme based on the revised simplex method with inverse in product form[12], together with an inversion technique designed to maintain sparsity in the representation of the basis inverse[15].

Two important aspects of the multi-stage design problem in Section 3 have not been exploited in our experiments. First, recall from (15) that each A_t has the form $[A, -A]$, so that roughly half the storage for A_t can be saved at the expense of some bookkeeping scheme. Secondly, when symmetry exists for a given problem, one needs only solve an equivalent problem of approximately half the size.

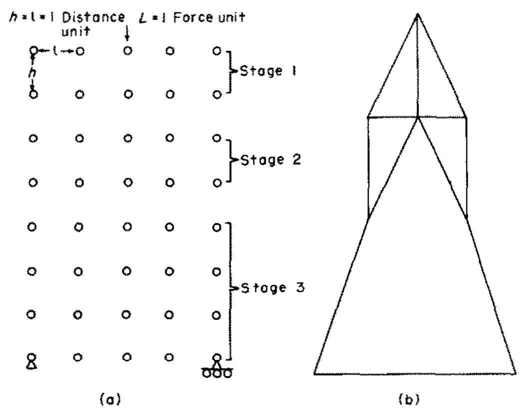

Fig. 7(a). Problem SCSD1. (b) An optimal structure of weight 8·67.

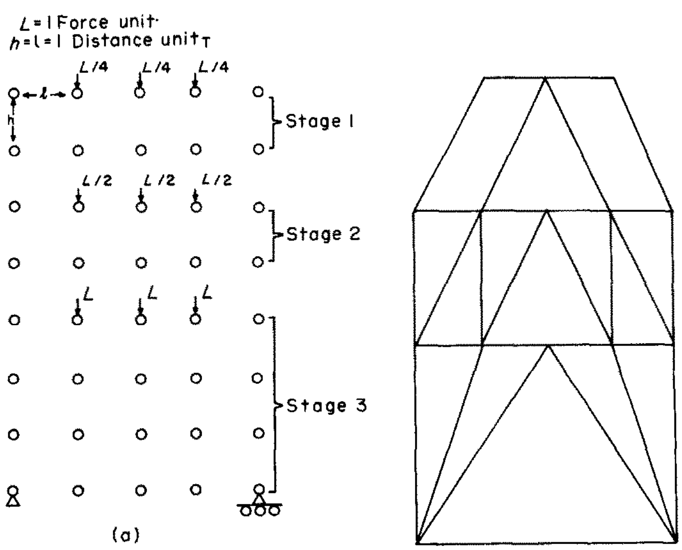

Fig. 8(a). Problem SCSD2. (b) An optimal structure of weight 29·52.

7. COMMENTS ON THE COMPUTER PROGRAM

An experimental code, named SC73, of the staircase algorithm has been written in FORTRAN IV for the IBM/360 and /370 Operating System[11]. Using the H level compiler with option 2, SC73 requires approximately 200 K bytes of core storage when dimensioned for problems with a maximum of 20 periods, each having up to 250 rows and 3000 non-zeros in the constraint matrix (including proposals), and 3000 non-zeros in the basis inverse (eta) file. The input data is in MPS/360 format[13] plus a section on information characterizing a pattern of decomposition. Secondary storage is on 220 tracks (1 track = 7294 bytes) of IBM type 2314 magnetic disc. Data transmission is done by direct access, unformatted I/O using variable length records with a block size of 7294 bytes. SC73 has approximately 2000 statements and takes about 20 sec to compile on an IBM 360/91. A listing of the program is contained in [11].

Table 1. Statistics of the test problems

	STATISTICS \ PROBLEM	SCSD1	SCSD2	SCSD3	SCSD4	SCSD5
Structural	No. of Stages	3	3	4	6	20
	No. of Joints	40	40	75	52	205
	No. of Bars	380	380	950	366	1910
LP	No. of Rows	78	78	148	102	408
	No. of Columns	838	838	2048	834	4228
	No. of Nonzeros	3226	3226	8288	3112	16604
	% Density	4.94	4.94	2.73	3.66	0.96
MPS/360	CPU Seconds	6.3	8.7	33.9	9.7	> 120*
	DISC I/O	194	198	359	209	>1500*
SC73	CPU Seconds	4.1	4.9	12.7	3.6	41.6
	DISC I/O	473	600	871	496	4870

*MPS/360 failed to produce a feasible solution after 120 sec.

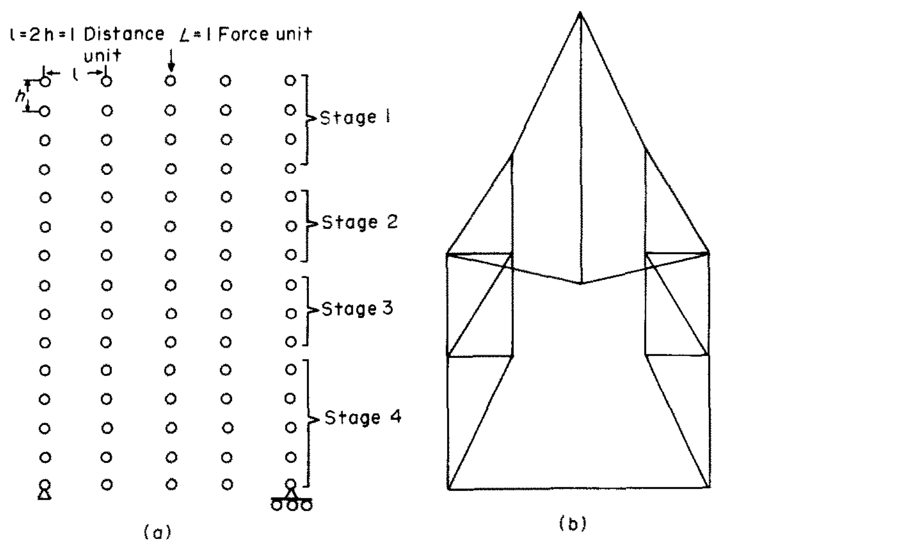

Fig. 9(a). Problem SCSD3. (b) An optimal structure of weight 9·16.

8. COMPUTATIONAL EXPERIENCE

Five examples of the multi-stage structural design problem, ranging from 3 to 20 stages and 40 to 205 admissible joints respectively, are used as test problems. All computations are performed on an IBM360/91 at the Stanford Linear Acceleration Center. The computing times reported are CPU seconds of total solution time excluding input. Disc I/O is measured in number of accesses to IBM type 2314 disc. In order to compare the nested decomposition approach with a direct linear programming approach, both SC73 and MPS/360 are used for each problem, with the same amount of core. For the largest problem SCSD5, SC73 requires approximately 180 K bytes of core. The problem and solution statistics are summarized in Table 1. Examples of optimal designs are illustrated in Figs. 7–9.

9. CONCLUSION

We have demonstrated the potential advantages of nested decomposition as a promising approach to solve multi-stage structural design problems. Our results encourage further work in this application of the staircase algorithm, especially to large-scale problems, as well as extensions to more realistic formulations.

REFERENCES

1. W. S. Dorn, R. E. Gomory and H. J. Greenberg, Automatic design of optimal structures, *J. Méc.* 3, 25–52 (1964).
2. G. B. Dantzig, *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey (1963).
3. K. I. Majid, *Non-Linear Structures.* Butterworths, London (1972).
4. R. L. Fox, *Optimization Methods for Engineering Design.* Addison-Wesley, Reading, Massachusetts (1971).
5. R. E. Gallagher and O. C. Zienkiewicz (Eds), *Optimum Structural Design.* John Wiley, London (1973).
6. M. W. Dobbs and L. P. Felton, Optimization of truss geometry, *Proc. ASCE* ST-10, 2105–2118 (1969).
7. G. N. Vanderplaat and F. Moses, Automated design of trusses for optimum geometry, *Proc. ASCE* ST-3, 671–690 (1972).
8. M. S. Gregory, *Linear Framed Structures.* John Wiley, New York (1966).
9. J. K. Ho and A. S. Manne, Nested decomposition for dynamic models, *Math. Prog.* 6, 121–140 (1974).
10. J. K. Ho, Nested decomposition of large-scale linear programs with the staircase structure, Ph.D. dissertation, Department of Operations Research, Stanford University, California, June 1974.
11. J. K. Ho, SC73 User's Manual, Systems Optimization Laboratory, Department of Operations Research, Stanford University, California, 1974.

12. W. Orchard-Hays, *Advanced Linear Programming Computing Techniques*. McGraw-Hill, New York (1968).
13. IBM application program, Mathematical Programming System/360 Version 2, Linear and Separable Programming, User's Manual H20-0476-2, 1969.
14. G. B. Dantzig and P. Wolfe, Decomposition principle for linear programming, *Ops Res.* **8**, 101–111 (1960).
15. J. A. Tomlin, Pivoting for size and sparsity in linear programming inversion routines, *J. Inst. Maths. Appl.* **10**, 289–295 (1972).