Geometric Programming Strategies in Large-Scale Structural Synthesis

Prabhat Hajela*
University of Florida, Gainesville, Florida

A geometric programming approach is proposed for the optimum design of complex structural configurations. The strategy is particularly attractive in that it reduces the solution of a nonlinearly constrained optimization problem to one with strictly linear constraints. Under certain conditions, the geometric programming solution is equivalent to solving a set of linear algebraic equations. The computational advantages of the geometric programming methodology are examined, particularly in the context of using well-documented approximation concepts.

Introduction

SIGNIFICANT advances in matrix methods of structural analysis and an increased digital computing capability have resulted in the emergence of automated structural synthesis as a viable design tool. The use of optimality criteria methods[1] and general nonlinear mathematical programming algorithms in conjunction with discrete analysis procedures[2] has dictated the primary trend for development in this discipline. Geometric programming is a relatively new approach to solving mathematical programming problems with nonlinear constraints. References 3-6 focus on the theoretical developments in this field. The method has its mathematical origin in the arithmetic-geometric mean inequality relationships between sums and products of positive numbers.[7] More rigorous Lagrangian duality concepts establish a relationship between geometric programming and convex programming.[8]

Geometric programming algorithms provide a very elegant solution procedure to an optimization problem with nonlinear constraints if the primary problem statement is in a special form. More precisely, geometric programming is applicable when the objective function and the associated design constraints can be expressed as *posynomials*, an adaptation of the word "polynomial" to the case in which all coefficients are positive. Under these conditions, the Lagrangian duality relations can be used to transform the primal nonlinear optimization problem to a dual linear programming problem for which very efficient solution techniques are available.

The dimensionality of the dual problem is not defined directly by the number of design variables and problem constraints. Instead, it is based on the difference between the number of primal design variables and the total number of posynomial terms in the objective function and constraints. The dimensionality is referred to as the degree of difficulty of the geometric programming problem. As shown in subsequent sections, the degree of difficulty is often significantly lower than the number of primal variables, thereby providing a reduction in the dimensionality of the problem. In the fortuitous case of a zero degree of difficulty, the geometric programming form is particularly powerful in that the solution of the optimization problem is reduced to solving a set of linear algebraic equations.

Received April 2, 1985; presented as Paper 85-0698 at the AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics and Materials Conference, Orlando, FL, April 15-17, 1985; revision received Oct. 7, 1985.

*Assistant Professor, Department of Engineering Sciences. Member AIAA.

The present paper describes a systematic implementation of the geometric programming principles in an optimum structural synthesis environment. In most realistic structural configurations, the analysis has to be performed by discrete numerical procedures such as the finite element method. Furthermore, to utilize the geometric programming solution strategy, the objective function and constraints must be formulated as posynomial functions on the basis of discrete analysis. Methods of complementary geometric programming that reduce multiterm polynomials to single-term posynomial functions by a log-linear approximation[9] are used to construct the appropriate forms for the geometric programming problem. These methods, in conjunction with the cumulative constraint concept,[10] are shown to lead to a sequence of very efficient zero-degree-of-difficulty problems. A strategy is proposed to enhance the efficiency of these methods for redundant configurations. Representative numerical solutions are presented for test problems with stress, displacement, and frequency constraints.

Mathematical Problem Statement

The general structural optimization problem can be stated as follows:

Minimize

$$W(d) \tag{1}$$

subject to the constraints

$$g_j(d) \le 0 \qquad j=1,2,\ldots m \tag{2}$$

$$h_k(d) = 0 \qquad k=1,2,\ldots m_1 \tag{3}$$

$$d_i^\ell \le d_i \le d_i^u \qquad i=1,2,\ldots n \tag{4}$$

where $W(\boldsymbol{d})$ is typically the structural weight and $\boldsymbol{d}$ is the n-dimensional vector of design variables, each component of which is limited by lower and upper bounds d_i^ℓ and d_i^u, respectively. The inequality constraints $g_j(\boldsymbol{d})$ represent limits imposed on the structural response, while the equality constraint $h_k(\boldsymbol{d})$ may be necessary to ensure precise design requirements. For most realistic structural configurations, the objective and constraint functions are implicitly dependent on the design variables.

Constrained Geometric Programming

The general constrained geometric programming problem is stated in the following form[11]:

Find

$$\{d\}^T = \{d_1, d_2, \ldots d_n\}$$

$$\text{Min } [F(\{d\})] \equiv \sum_{j=1}^{N_0} C_{0j} \prod_{i=1}^{n} d_i^{a_{0ij}} \tag{5}$$

subject to the constraints

$$g_k(\{d\}) = \sum_{j=1}^{N_k} C_{kj} \prod_{i=1}^{n} d_i^{a_{kij}} \lesseqgtr 1 \qquad k = 1,2,\ldots m \tag{6}$$

and

$$d_i \geq 0 \qquad i = 1,2,\ldots n \tag{7}$$

The above primal problem statement is in a posynomial form when the coefficients c_{ij} are positive and the exponents a_{kij} are real numbers. The inequality constraints, Eq. (6), can be rewritten as

$$f_k \equiv \sigma_k(1 - g_k(\{d\})) \geq 0 \qquad k = 1,2,\ldots m \tag{8}$$

where σ_k is a signum function taking on a value of $+1$ or -1 for $g_k(\{d\}) \leq 1$ and $g_k(\{d\}) \geq 1$, respectively. Transforming the primal variables as

$$d_i = e^{\zeta_i} \qquad i = 0,1,2,\ldots n \tag{9}$$

and further defining,

$$\Delta_{0j} = C_{0j} \prod_{i=1}^{n} d_i^{a_{0ij}} \Big/ F(\{d\}) > 0 \tag{10a}$$

and

$$\Delta_{kj} = C_{kj} \prod_{i=1}^{n} d_i^{a_{kij}} > 0 \quad k = 1,2,\ldots m, \qquad j = 1,2,\ldots N_k \tag{10b}$$

where

$$\sum_{j=1}^{N_0} \Delta_{0j} = 1 \text{ (always)} \tag{11}$$

and for the kth constraint that is critically satisfied

$$\sum_{j=1}^{N_k} \Delta_{kj} = 1 \tag{12}$$

Taking the natural logarithms of Eqs. (10), and making the substitution

$$\zeta_0 = \ln F, \quad \zeta_i = \ln d_i \tag{13}$$

one can write

$$\ln\left(\frac{\Delta_{0j}}{c_{0j}}\right) = -\zeta_0 + \sum_{i=1}^{n} a_{0ij}\zeta_i \quad j = 1,2,\ldots N_0 \tag{14}$$

and

$$\ln\left(\frac{\Delta_{kj}}{c_{kj}}\right) = \sum_{i=1}^{n} a_{kij}\zeta_i \quad k = 1,2,\ldots m, \quad j = 1,2,\ldots N \tag{15}$$

The primal geometric programming problem is reduced to minimizing ζ_0, subject to the inequality and equality constraints [Eqs. (10-15)]. The Lagrangian for this optimization problem is written as

$$\begin{aligned} L(\zeta,\Delta,\lambda) &= \zeta_0 + \lambda_0\left(\sum_{j=1}^{N_0} \Delta_{0j} - 1\right) + \sum_{k=1}^{m} \lambda_k \sigma_k \left(\sum_{j=1}^{N_k} \Delta_{kj} - 1\right) \\ &+ \sum_{j=1}^{N_0} \lambda_{0j}\left[\sum_{i=1}^{n} a_{0ij}\zeta_i - \zeta_0 - \ln\left(\frac{\Delta_{0j}}{C_{0j}}\right)\right] \\ &+ \sum_{k=1}^{m}\sum_{j=1}^{N_k} \lambda_{kj}\sigma_k\left[\sum_{i=1}^{n} a_{kij}\zeta_i - \ln\left(\frac{\Delta_{kj}}{C_{kj}}\right)\right] \end{aligned} \tag{16}$$

An application of the necessary conditions for optimality permit a restatement of the Lagrangian in the form

$$\begin{aligned} L(\zeta,\lambda,f) &= (\zeta_0 - 1)\left(1 - \sum_{j=1}^{N_0} \lambda_{0j}\right) \\ &+ \sum_{i=1}^{n} \zeta_i \sum_{k=0}^{m}\sum_{j=1}^{N_k} \sigma_k \lambda_{kj} a_{kij} \\ &+ \sum_{k=1}^{m} \lambda_k f_k + \sum_{k=0}^{m}\sum_{j=1}^{N_k} \sigma_k \lambda_{kj} \ln\left(\frac{C_{kj}}{\lambda_{kj}} \sum_{\ell=1}^{N_k} \lambda_{k\ell}\right) \end{aligned} \tag{17}$$

This form suggests a reformulation of the optimization problem in terms of dual variables λ_{kj} as follows:

Maximize

$$D(\lambda) = \sum_{k=0}^{m}\sum_{j=1}^{N_k} \sigma_k \lambda_{kj} \ln\left(\frac{C_{kj}}{\lambda_{kj}} \sum_{\ell=1}^{N_k} \lambda_{k\ell}\right) \tag{18}$$

subject to

$$1 - \sum_{j=1}^{N_0} \lambda_{0j} = 0 \tag{19}$$

$$\sum_{k=0}^{m}\sum_{j=1}^{N_k} \sigma_k \lambda_{kj} a_{kij} = 0 \quad i = 1,2,\ldots n \tag{20}$$

and

$$\lambda_k \equiv \sum_{j=1}^{N_k} \lambda_{kj} \geq 0 \quad k = 1,2,\ldots m \tag{21}$$

In this formulation, $\zeta_0 - 1$, ζ_i, and f_k play the role of Lagrange multipliers. It can be shown that a maximum of the dual function $D(\lambda)$ also minimizes the primal objective function $F(\{d\})$. The optimization problem posed in Eqs. (18-21) forms the basis for an efficient geometric programming solution strategy. The objective function to be maximized is a nonlinear function in λ_{kj} and is constrained by strictly linear constraints with added requirements of positivity for the dual variables. In the geometric programming terminology, Eqs. (19) and (20) are referred to as the normality and orthogonality conditions, respectively. Highly efficient solution techniques are available for this class of problems.[12] Once the dual solution is known, the primal design variables are obtained from n independent equations of the set (14) and (15).

The equality constraints of the dual problem can be used to eliminate $(n+1)$ dual variables from the original set. Combining the orthogonality and normality conditions, one can write

$$\underset{n+1 \times \bar{N}}{[B]} \; \underset{\bar{N}\times 1}{\{\lambda\}} = \underset{n \times 1}{\{i\}} \tag{22}$$

where $\bar{N}$ is the total number of posynomial terms in the objective function and the constraints, n the number of design variables in the original primal problem statement, $[B]$ a coefficient matrix, and $\{\lambda\}$ an $\bar{N}$-dimensional vector of dual variables. The vector $\{i\}$ is defined as

$$\underset{(n\times 1)}{\{i\}} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \vdots \\ 0 \end{Bmatrix} \tag{23}$$

The dual variable vector $\{\lambda\}$ and matrix $[B]$ are partitioned as follows

$$\{\lambda\} = \begin{Bmatrix} \lambda_I \\ \lambda_{II} \end{Bmatrix} \begin{matrix} (n+1)\times 1 \\ d\times 1 \end{matrix} \tag{24}$$

$$[B] = [\underset{(n+1)\times(n+1)}{R} \;\vdots\; \underset{(n+1)\times d}{S}] \tag{25}$$

where d is referred to as the degree of difficulty of the geometric programming problem

$$(d = \bar{N} - (n+1))$$

The set of variables $\{\lambda_I\}$ to be eliminated from the original set of $\{\lambda\}$ can be expressed as

$$\{\lambda_I\} = [R]^{-1}(\{i\} - [S]\{\lambda_{II}\}) \tag{26}$$

The dual optimization problem is thus expressed in terms of independent dual variables $\{\lambda_{II}\}$ and is constrained only by the positivity requirement on the variables.

The Structural Resizing Problem

The foregoing sections detail the powerful optimization tool that a geometric programming methodology affords. The successful implementation of this approach to a structural sizing problem is critically dependent on the ability of the designer to pose the objective function and constraints in an appropriate posynomial form. In the design of structural components, it is often possible to express the response parameters in this form.[13] For more realistic structural configurations that require discrete analysis procedures, approximating posynomial forms can be obtained for the objective function and the constraints. The log-linear approximation that forms the core of complementary geometric programming provides a very useful method to approximate the structural optimization problem by a sequence of geometric programming posynomial forms. A function $\ln[\phi(d)]$ can be approximated by a linear extrapolation about a design point $(\bar{d})$ as

$$\ln[\phi(d)] = \ln[\phi(\bar{d})] + \sum_{i=1}^{n} \frac{\partial[\ln\phi(d)]}{\partial[\ln d_i]}\bigg|_{d=\bar{d}} [\ln d_i - \ln\bar{d}_{is}] \tag{27}$$

This can be rewritten as

$$\ln[\phi(d)] = \ln[\phi(\bar{d})] + \sum_{i=1}^{n} \frac{\bar{d}_i}{\phi(\bar{d})} \frac{\partial[\phi(d)]}{\partial d_i}\bigg|_{d=\bar{d}} \ln\left(\frac{d_i}{\bar{d}_i}\right) \tag{28}$$

or

$$\phi(d) = \phi(\bar{d}) \prod_{i=1}^{n} \left(\frac{d_i}{\bar{d}_i}\right)^{\frac{\bar{d}_i}{\phi(\bar{d})} \frac{\partial[\phi(d)]}{\partial d_i}\big|_{d=\bar{d}}} \tag{29}$$

Equation (29) is the desired posynomial form and is obtained on the basis of the function value and its derivative at a starting design point. A sequence of such approximations are necessary to obtain the solution to a design problem. Consider constraints on the stress σ, displacement u, and frequency ω for a given structure

$$g_j(\bar{d}) = \frac{\sigma}{\sigma_{\text{all}}} \le 1 \qquad j = 1,2,\ldots m$$

$$= \frac{u}{u_{\text{all}}} \le 1$$

$$= \frac{\omega}{\omega_{\text{all}}} \le 1 \tag{30}$$

The left-hand side of these equations can be represented as single-term posynomials by Eq. (29). Hence, the total number of posynomial terms in the geometric programming problem is equal to the number of constraints plus the number of posynomial terms in the objective function. For most structural design problems with member sizes playing the role of design variables, the objective function can be written analytically as an n-term posynomial. Complementary geometric programming can be used to compact it as a single-term posynomial. The side constraints on the design variables are naturally available in the desired posynomial form

$$d_i/d_i^u \le 1, \quad d_i^\ell d_i^{-1} \le 1 \tag{31}$$

Approximation Concepts

The efficiency of geometric programming strategies can be improved substantially by incorporating approximation concepts developed in the context of NLP-based structural design. As is clearly evident from the formulation in the preceding sections, the most desirable form of the problem statement is one that yields a zero-degree-of-difficulty problem. The strategies experimented with in the present effort can be categorized as follows.

A. Constraint Deletion

The "throw-away" concept proposed in Ref. 14 was implemented in the design algorithm. Constraints with a 65% safety margin were deleted from consideration to decrease the number of posynomial terms and hence also the degree of difficulty. This factor was selected arbitrarily and, with tighter move limits on the design variables, can be further relaxed.

B. Design Variable Deletion

This step is important for the standpoint of reducing the number of gradient computations required to approximate the posynomial form. In the design of redundant structures with multiple load paths, the optimum design is often characterized by a few members taking on minimum gage values. The concept of design-variable dominance, proposed and illustrated in Ref. 15, was used very effectively to delete the inactive variables from the original set. The dominance of a variable is defined by the numerical quantity

$$CME = \frac{\partial\Omega/\partial d_i}{\partial\bar{W}/\partial d_i} \qquad i = 1,2,\ldots n \tag{32}$$

where $\bar{W}$ is the objective function and Ω is a cumulative constraint representation of the form

$$\Omega = \sum_{i=1}^{m} \langle g_i \rangle^r \qquad \text{if } \Omega > \psi$$

$$= \frac{1}{\rho}\ln\left(\sum_{i=1}^{m} \exp(\rho g_i)\right) \qquad \text{if } \Omega \le \psi \tag{33}$$

$$\langle g_i \rangle = \begin{cases} g_i & \text{if } g_i > 0 \\ 0 & \text{if } g_i \le 0 \end{cases} \qquad i = 1,2,\ldots m \tag{34}$$

ψ is a preselected parameter (order 10^{-1}) that allows transition from one formulation to the other and is chosen such that the change occurs close to the constraint boundary; r is typically chosen as 2 but is reduced as the constraint approaches the boundary; ρ is referred to as the constraint participation factor, and its numerical value is typically of order 10^2. The design variables are ordered by decreasing magnitude of CME_i and the first k variables are retained as active variables, where k renders S minimum subject to

$$S = \sum_{i=1}^{k} ABS(CME_i) \ge f \sum_{i=1}^{n} ABS(CME_i) \tag{35}$$

where f is a judgmentally chosen factor (typically f is 0.9-0.97).

C. "Collapsing" Constraints

This feature has not been experimented with extensively and in the present task was implemented only for the six-bar determinate truss with stress constraints. This feature is dependent on the ability to write the objective function as an n-term posynomial function for the case of n design variables. The use of the cumulative constraint [Eq. (33)] creates a zero-degree-of-difficulty problem with all the associated advantages.

In addition to these approximating strategies, linear extrapolation of stiffness and mass matrices, first-order explicit approximations for constraint functions, and approximations to obtain first-order derivatives from a stored data base of function values are being implemented in the programming system.

Numerical Implementation

A flowchart depicting the order of execution of various programs is shown in Fig. 1. The optimization/analysis system has been developed on a VAX 11-750 system with the looping between the processors controlled in the Command Language feature of DEC systems. The primary analysis tool used in this effort is a general-purpose finite element program EAL,[16] and the optimization algorithm CINMIN[17] was used in problems with nonzero degrees of difficulty. A gradient projection algorithm particularly well suited for the class of nonlinear objective, linearly constrained problems is also available in the programming system.

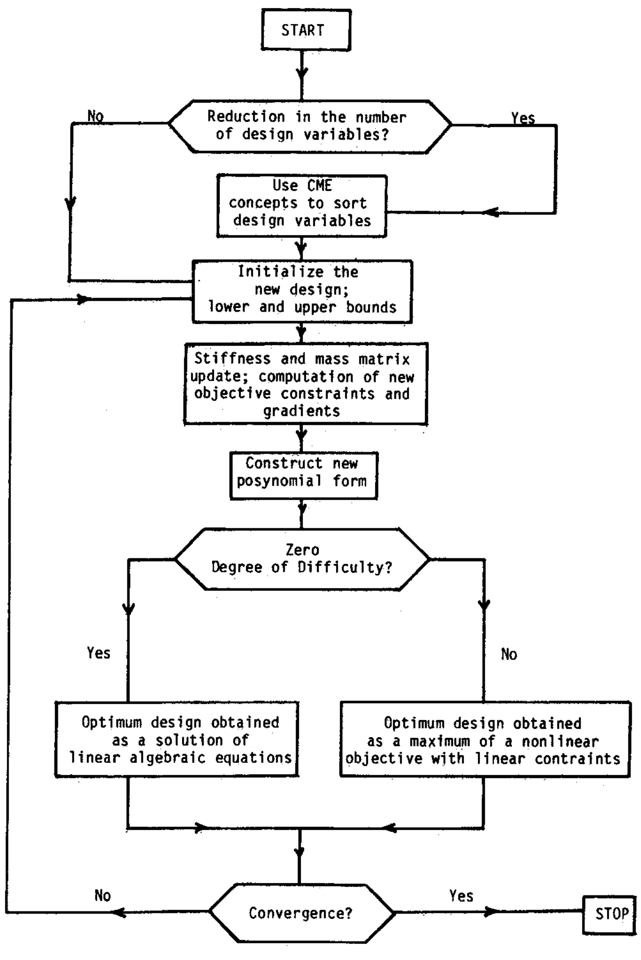

Fig. 1 Flowchart depicting the computer implementation of the geometric programming strategy.

Numerical Results

The dimensionality of the numerical test problems was intentionally kept low, in conformity with the stated objective of establishing the effectiveness of the geometric programming approach. The test problems include statically determinate and indeterminate truss structures and a built-up finite element model of a wing structural box. Constraints on allowable stress, nodal deflections, and natural frequencies were prescribed in the design procedure. A description of these problems and the results obtained can be summarized as follows.

A. Six-Bar Truss

A six-member, ten-degree-of-freedom, statically determinate truss comprised the first test structure for the algorithm. The structure and the loading conditions are shown in Fig. 2. The first design was for stress constraints only, with permissible stresses in tension and compression of 50,000 psi. As expected, a fully stressed design is obtained. The results are presented in Table 1 and compared with solutions from a standard nonlinear programming approach. A maximum permissible y displacement at node 3 of 1.0 in. was included in the constraint set. The results for this case are presented in Table 2. The stress-constrained problem was solved as a zero degree-of-difficulty problem. The solution to the latter problem was obtained both from a one-degree-of-difficulty problem with a single-term posynomial representation of the objective function and from a six-degree-of-difficulty problem with a six-term posynomial form for the objective function. The only reason for considering the latter is the obvious ease in identifying independent equations to solve for the primal variables from a converged dual solution.

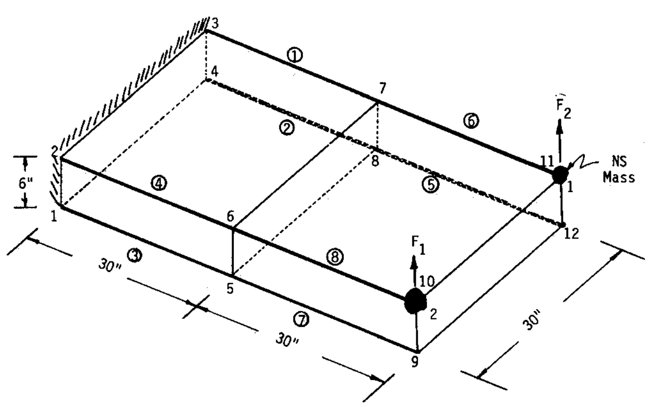

Fig. 2 Six-bar planar truss ($P = 40{,}000$ lb, $E = 10.5 \times 10^6$ psi).

Table 1 Numerical results for the six-bar truss with stress constraints

	Design variable no.										Objective, lb
	1	2	3	4	5	6	7	8	9	10	
G.P. solution, in.^{2}	7.947	8.066	3.943	5.743	5.571	5.573	0.100	0.100	0.100	0.100	1594.12
NLP solution, in.^{2} (Ref. 14)	7.938	8.062	3.938	5.745	5.569	5.569	0.100	0.100	0.100	0.100	1593.23

Table 2 Numerical results for the six-bar truss with stress and displacement constraints

	Design variable no.						Objective,
	1	2	3	4	5	6	lb
G.P. solution, in.^{2}	1.599	1.599	1.789	2.684	0.894	0.799	38.395
NLP solution, in.^{2}	1.599	1.599	1.792	2.690	0.896	0.798	38.435

Table 3 Numerical results for ten-bar truss with stress constraints

	Design variable no.										Objective,
	1	2	3	4	5	6	7	8	9	10	lb
G.P. solution, in.^{2}	29.853	22.496	14.657	8.129	19.659	19.456	0.10	0.10	0.10	0.10	4817.54

Table 4 Numerical results for ten-bar truss with stress and displacement constraints

	Design variable no.						Objective,
	1	2	3	4	5	6	lb
G.P. solution, in.^{2}	2.884	2.884	3.194	3.942	0.895	0.799	60.569
NLP solution, in.^{2}	2.870	2.870	3.353	3.992	0.896	0.798	61.416

Table 5 Numerical results for wing structural box. Skin and web elements have a thickness of 0.12 and 0.2 in., respectively. NSM1 and NSM2 are the nonstructural masses defined in Fig. 4

	Design variable no.										Objective,
	1	2	3	4	5	6	7	8	NSM1	NSM2	lb
G.P. solution, in.^{2}	4.347	4.359	1.143	1.138	0.551	0.579	0.100	0.100	0.1	27.097	121.652
NLP solution, in.^{2}	4.955	4.840	1.312	1.336	0.570	0.681	0.100	0.100	0.1	27.100	126.385

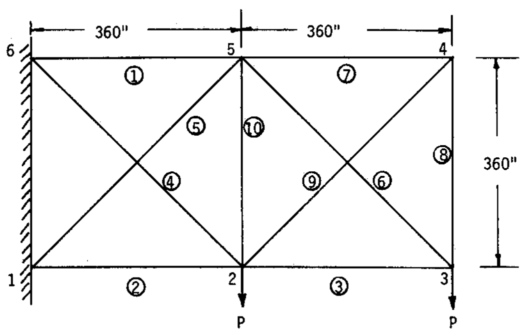

Fig. 3 Ten-bar planar truss ($P = 100{,}000$ lb, $E = 10.5 \times 10^6$ psi).

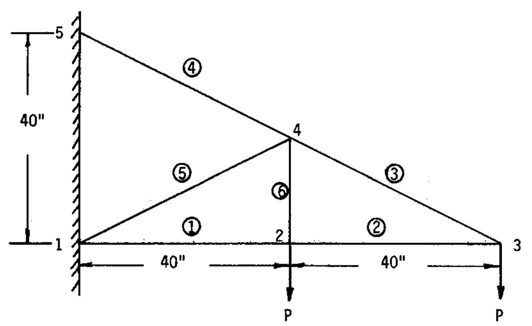

Fig. 4 Finite element model of wing structural box ($F_1 = 15{,}000$ lb, $F_2 = 30{,}000$ lb, $E = 10.5 \times 10^6$ psi).

B. Ten-Bar Truss

This truss configuration, shown in Fig. 3, has been a testing bed for several optimization algorithms. The layout and load conditions are shown in the figure. The first run for this structure was with stress constraints only, permitting maximum stress in compression and tension of 25,000 psi. Design variable deletion was used to reduce the number of active variables to six. A second test case was the modification of the previous problem by adding maximum allowable y displacements at nodes 2-5 to be less than or equal to 2.0 in. The designs for these cases are presented in Tables 3 and 4, respectively. As in the case of the six-bar truss, the objective function was represented both as a single-term and a six-term posynomial to generate the geometric programming solutions.

C. Semimonocoque Structural Box

This test structure, shown in Fig. 4, was sized for stress and frequency constraints. The built-up model consists of axial-force bar elements, quadrilateral membrane elements, and rigid masses placed at outboard locations, as shown in the figure. The bar elements were sized for maximum permissible stresses in compression and tension of 35,000 psi. Sorting of the design variables was done by the variable deletion method described earlier, resulting in six active primal variables. The solution was obtained from a zero-degree-of-difficulty geometric programming problem. The base structure obtained from this exercise was further constrained to have its first natural frequency below 2.013 Hz. The two tuning masses at the tip were treated as the primary design variables, again resulting in a zero-degree-of-difficulty problem. The results for these test cases are shown in Table 5.

For the numerical test cases experimented with in this effort, CPU time savings factors ranged from 2-10 over a standard NLP-based approach using a combination of EAL and CONMIN with finite difference gradients.

Conclusions

Geometric programming methods provide an extremely efficient alternative technique for the optimum design of structural systems. The method is particularly powerful in cases where zero-degree-of-difficulty problems can be formulated. An additional feature of this class of problems is the location of a "global optimum" in that region of the design space where the posynomial approximation is generated. The combination of well-developed approximation concepts with the geometric programming techniques significantly enhances the capabilities of the latter. Additional studies are required to examine the convergence characteristics of this method for a class of problems with mixed element types and more complex constraint formulations,

References

[1]Berke, L. and Venkayya, V. B., "Review of Optimality Criteria Approaches to Structural Optimization," presented at ASME Winter Annual Meeting, New York, Nov. 1974.

[2]Rogers, J. R. et al., "An Implementation of the Programming Structural Synthesis System (PROSSS)," NASA-TM-83180, Dec. 1981.

[3]Duffin, R. J., Peterson, E. L., and Zener, C. M., *Geometric Programming*, John Wiley & Sons, New York, 1967.

[4]Zener, C. M., *Engineering Design by Geometric Programming*, Wiley Interscience, New York, 1971.

[5]Morris, A. J., ed., *Foundations of Structural Optimization: A Unified Approach*, John Wiley & Sons, New York, 1982.

[6]Beightler, C. and Phillips, D. T., *Applied Geometric Programming*, John Wiley & Sons, New York, 1976.

[7]Kazarinoff, N. D., *Geometric Inequalities*, Random House, New York, 1961.

[8]Rockefeller, R. T., *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[9]Avriel, M. and Williams, A. C., "Complementary Geometric Programming," *SIAM Journal of Applied Mathematics*, Vol. 19, 1970, pp. 125-141.

[10]Hajela, P., "Techniques in Optimum Structural Synthesis with Static and Dynamic Constraints," Ph.D. Thesis, Stanford University, Stanford, CA, July 1982.

[11]Rao, S. S., *Optimization-Theory and Application*, Wiley Eastern Ltd., New Delhi, 1978.

[12]Ecker, J. G. and Kupferschmid, M., "A Computational Comparison of Several Nonlinear Programming Algorithms," Rensselaer Polytechnic Institute, Troy, NY, Rept. OR&S-82-2, 1982.

[13]Burns, S. and Ramamurthy, S., "Mathematical Modeling in Structural Design," *Journal of Structural Engineering*, Aug. 1983.

[14]Schmit, L. A. Jr. and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA CR-2552, March 1976.

[15]Hajela, P. and Sobieski, J. E., "The Controlled Growth Method—A Tool for Structural Optimization," *AIAA Journal*, Vol. 20, Oct. 1982, pp. 1440-1441.

[16]Whetstone, D., *SPAR—Reference Manual*, NASA CR-145098 -1, Feb. 1977.

[17]Vanderplaats, G. N., *CONMIN—A FORTRAN Program for Constrained Function Minimization—User's Manual*, NASA TMX-62282, Aug. 1973.