Int. J. Solids Structures, 1972, Vol. 8, pp. 847 to 864. Pergamon Press. Printed in Great Britain

STRUCTURAL OPTIMIZATION BY GEOMETRIC PROGRAMMING†

A. J. MORRIS

Structures Department, Royal Aircraft Establishment, Farnborough, Hampshire, England

Abstract—Details are given on the technique of geometric programming together with an explanation as to how it can be used to obtain solutions for certain problems in structural optimization. Emphasis is laid on the flexibility of the method and on its capacity to give lower bounds to the minimum value of an objective function by means of simple calculations. It is also shown how the method may be used to rapidly ascertain the influence of any constraints which are imposed on the objective function in order that inactive ones may be omitted before the main minimization calculation is started. A discussion of possible extensions to the method by the use of approximation techniques is also included.

1. INTRODUCTION

ONE of the most refreshing developments in the recent history of optimization theory was the introduction of a remarkable technique known as geometric programming. The method first came to light in 1961 when Zener [1] observed that a sum of component costs can sometimes be minimized almost by inspection when each cost depends on products of the design variables each raised to a known power. An account of this early work and its subsequent development is given in the expository book by Duffin *et al.* [2], some extensions to the original theory are discussed by Wilde and Beightler [3] and a modern treatment is presented by Peterson [4] who also references the more recent theoretical papers. Essentially the method of geometric programming consists of a sequence of operations upon a set of terms which are expressed as generalized positive polynomials or, briefly, posynomials. An example of such a posynomial containing two terms and three variables might be

$$f(\bar{x}) = ax_1^{0\cdot7}x_2^{-1}+bx_2^{0\cdot5}x_3.$$

A posynomial may thus be defined as a function consisting of a sum of terms which comprise of a positive coefficient multiplied by products of variables and with each variable raised to an arbitrary power. The technique of minimizing a function by means of geometric programming has proved incisive and simple in operation when applied to engineering design [2] but at the present time has been mainly employed in solving problems encountered in the chemical process industry.

In the case of structural optimization the problem is normally one of minimizing a suitable objective function whilst conforming to a set of constraints which are imposed by either physical or design considerations. The most usual method employed for the handling of this problem [5] is to consider the objective function as a point in a constrained function space where the minimum is sought by moving along certain preferred directions in a sequential manner. However, general computer programs of this type quite often require large amounts of storage and many of the methods currently available can encounter convergence difficulties. Further, if the program must be stopped due to the effects of rank annihilation or excessive time, because the function space may be locally non-convex, there is no way of knowing how near the current value of the objective function is to its minimum. These methods can take little advantage of problems with simple mathematical structures to obtain rapid answers and they only offer limited help in ascertaining the influence of variables within a given formulation. Geometric programming, on the other hand, is able to give bounds on the minimum value for an objective function together with a better overall picture of the relative importance of various parameters and can take advantage of certain mathematical structures to obtain rapid answers. It has also a most unusual property in that problems which have local minima are easily treated since a dual formulation renders them strictly concave and, in this respect, it is unique amongst the known optimization methods. Even under the most unfavourable conditions when used in conjunction with a more orthodox sequential technique it will reduce a constrained minimization problem to an unconstrained one with positive variables which is usually easier to handle sequentially. But the method does suffer from the disadvantage that the problem must be cast in posynomial form and it is by no means obvious that all the structural optimization problems currently of interest can be moulded into this form. However, an approximate form of geometric programming is presented which offers the possibility of circumventing this problem.

Despite this disadvantage there are, at least, two areas in the field of structural optimization where the method can be applied without difficulty. If, for example, a structure is to be designed according to a set of simple design or cost codes the objective function will depend upon the dimensions of the structure and these in turn will be constrained by the demands of the code. In such a case the objective function and the constraints will probably reduce to simple algebraic expressions which may be transformed into posynomial form. The other class of problem which offers no immediate difficulty is that of a determinate structure where the stresses in each element can be obtained from static calculations. Many problems in this field are of a very simple type which give rise to linear objective functions with posynomial constraints and can be treated in a very straightforward manner.

The main purpose of the present work is, therefore, to show that the method of geometric programming can be used to its full advantage when treating structural problems. In order to make the paper reasonably self contained the next section contains a description of the method in some detail together with an outline of the well-known duality theorem for convex sets as it applies to geometric programming. The third section illustrates the principles of the method by minimizing a simple two-bar truss which calls into play all the ideas of section two. A procedure for computerizing the method is then explained and section five shows how this may be used to solve a ship bulkhead problem. This particular problem has been solved in previous papers and the primary purpose for its inclusion is to illustrate how the computerized geometric programming technique may be successfully applied to the minimization of meaningful structural problems.

Finally, geometric programming has an advantage which is not immediately obvious but adds to the attractiveness of the method. It is possible to show that by a simple transformation any linear programme can be expressed as a geometric programme of a special type in which each posynomial has only a single term. Although this point is not pursued in the present paper additional information can be obtained by consulting Duffin *et al.*

2. THE FUNDAMENTALS OF GEOMETRIC PROGRAMMING

Consider any objective function g_0 which is the sum of n_0 terms dependent upon a set of variables $x_1, x_2, \ldots, x_m$,

$$g_0(\bar{x}) = \sum_{i=1}^{n_0} c_i x_1^{a_{i1}} x_2^{a_{i2}}, \ldots, x_m^{a_{im}}$$

where the c_i's are positive constants and the powers a_{ij} are real numbers which may be positive, negative or zero. This expression may be re-written in a more compact notation as,

$$g_0(\bar{x}) = \sum_{i=1}^{n_0} P_i(\bar{x}), \tag{2.1a}$$

where

$$P_i(\bar{x}) = c_i \prod_{j=1}^{m} x_j^{a_{ij}} \tag{2.1b}$$

and constitutes a posynomial formulation for the objective function. If this function is to be minimized subject to a set of constraints which can also be written in posynomial form the resulting problem is known as the primal problem of geometric programming or simply the primal programme.

Primal programme

Find a vector $\bar{x}$ which minimizes (2.1) subject to the constraints

$$x_i \geq 0, \qquad i = 1, \ldots, m \tag{2.2}$$

and

$$g_1(\bar{x}) \leq 1, \qquad g_2(\bar{x}) \leq 1, \ldots, g_p(\bar{x}) \leq 1 \tag{2.3}$$

where these additional g's are given by

$$g_k(\bar{x}) = \sum_{i=l_k}^{n_k} P_i(\bar{x}), \qquad k = 1, \ldots, p \tag{2.4}$$

and

$$l_1 = n_0 + 1, \ldots, l_k = n_{(k-1)} + 1, \qquad k = 1, \ldots, p$$

with

$$\begin{aligned} n_0 &= \text{number of terms in } g_0(x) \\ n_1 - n_0 &= \text{number of terms in } g_1(x) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ n_p - n_{p-1} &= \text{number of terms in } g_p(x) \end{aligned}$$

noting that $n = n_p$ = the total number of terms, i.e. the summation of all the terms in all the posynomials.

As before the term $P_i(\bar{x})$ is given by (2.1b) where the exponents a_{ij}, are once more arbitrary real numbers, and the coefficients c_i are all positive. The matrix (a_{ij}) is termed the exponent matrix, it has n rows and m columns.

In the above formulation the objective function $g_0(\bar{x})$ is termed the *primal function*, the variables $x_1, x_2, \ldots, x_m$ are called *primal variables*. The constraints (2.2) are termed *natural constraints*, and those given at (2.3) are called the *forced constraints*. Collectively these constraints are referred to as *primal constraints*.

The real power of geometric programming will be seen to lie in its ability to maximize sets of product functions called the dual programme. We now state this dual problem of geometric programming in the following way.

Dual programme

Find a vector $\bar{\delta}$ that maximizes the product function

$$V(\bar{\delta}) = \left[\prod_{i=1}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right]\prod_{k=1}^{p}\lambda_k(\bar{\delta})^{\lambda_k(\bar{\delta})} \tag{2.4a}$$

where

$$\lambda_k(\bar{\delta}) = \sum_{i=l_k}^{n_k}\delta_i, \qquad k = 1, \ldots, p$$

subject to the linear constraints

$$\delta_1 \geq 0, \qquad \delta_2 \geq 0, \ldots, \delta_n \geq 0 \tag{2.5}$$

$$\sum_{i=1}^{n_0}\delta_i = 1 \tag{2.6}$$

and

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \qquad j = 1, 2, \ldots, m \tag{2.7}$$

with a_{ij}, c_i, l_k, n_k the same as in the primal programme.

In evaluating the product function $V(\bar{\delta})$ it is to be understood that $x^x = x^{-x} = 1$ for $x = 0$.

The product function $V(\bar{\delta})$ is termed the *dual function* and the variables $\delta_1, \delta_2, \ldots, \delta_n$ are called *dual variables*. Relation (2.5) is termed the *positivity condition* and (2.6) the *normality condition* whilst (2.7) constitutes the *orthogonality condition*. Collectively, these conditions are known as the *dual constraints*.

In examining these two formulations it is seen that δ_i is associated with the ith term $c_i x_1^{a_{i1}}, \ldots, x_m^{a_{im}}$ of the primal programme, so that each term of $g_k(\bar{x}), k = 0, 1, 2, \ldots, p$, is associated with one and only one of the dual variables $\delta_1, \ldots, \delta_n$. Similarly, each factor $\lambda_k(\bar{\delta})^{\lambda_k(\bar{\delta})}$ of $V(\bar{\delta})$ comes from a forced constraint $g_k(x) \leq 1$. It may be noted that no such factor appears from the primal function because the normality condition forces $\lambda_0(\bar{\delta})$ to be one.

Even though the two families of primal and dual programmes are mutually exclusive there does exist a correspondence between the class of all primal programmes and the class of all dual programmes, and a duality theory relates the properties of each primal programme to the properties of its corresponding dual.

Before stating any of the duality theorems of geometric programming certain details of nomenclature require elaboration. First of all a programme (either primal or dual) is said to be *consistent* if there is at least one point (vector) that satisfies its constraints. Second, the primal programme is said to be *superconsistent* if there is at least one vector $\bar{x}^*$ which has positive components such that,

$$g_k(\bar{x}^*) < 1, \qquad k = 1, 2, \ldots, p.$$

In terms of the preceding concepts we state theorem 1, which is called the first duality theorem of geometric programming and is the main theorem of the current formulation.

Theorem 1

Suppose that the primal programme is superconsistent and that the primal function $g_0(\bar{x})$ attains its constrained minimum value at a point which satisfies the primal constraints. Then:

(i) The corresponding dual programme is consistent and the dual function $V(\bar{\delta})$ attains its constrained maximum value at a point which satisfies the dual constraints.

(ii) The constrained maximum value of the dual function is equal to the constrained minimum value of the primal function.

(iii) If $\bar{\delta}^*$ is a maximizing point for the dual programme each minimizing point $\bar{x}^*$ for the primal programme satisfies the system of equations

$$\delta_i^* = \begin{cases} P_i(\bar{x}^*)/g_0(\bar{x}^*), & i = 1 \ldots n_0 \\ \lambda_k(\bar{\delta}^*)P_i(\bar{x}^*) & i = n_0+1 \ldots n \\ & k = 1 \ldots p, \end{cases} \tag{2.8}$$

where each P_i in the expression $\lambda_k(\bar{\delta}^*)P_i(\bar{x}^*)$ is a term contained in the posynomial constraint equation $g_k(\bar{x})$. $\bar{\delta}^*$ further satisfies the system,

$$\lambda_k(\bar{\delta}^*)[1 - g_k(\bar{x}^*)] = 0, \qquad k = 1 \ldots p. \tag{2.9}$$

It may be noted that equations (2.8) and (2.9) provide a method whereby the minimizing vector $\bar{x}^*$ can be found from a knowledge of a maximizing vector $\bar{\delta}^*$. Although it has not been stated above it is possible to show [2] that theorem 1 may be extended to obtain upper and lower bounds to the primal and dual programmes. Thus for a set of feasible vectors $\bar{\delta}$ and $\bar{x}$ giving $g_0(\bar{x})$ and $V(\bar{\delta})$, respectively, we have that

$$g_0(\bar{x}) > g_0(\bar{x}^*) = V(\bar{\delta}^*) > V(\bar{\delta}). \tag{2.10}$$

Before leaving the first theorem it may be observed by way of equation (2.9) that for any maximizing vector $\bar{\delta}^*$, those dual variables which correspond to tight constraints $[g_k(\bar{x}^*) = 1]$ are positive, i.e. $\delta_k^* > 0$, whilst those associated with loose constraints $[g_k(\bar{x}^*) < 1]$ are zero, i.e. $\delta_k^* = 0$.

Theorem 1 is the most important theorem from a practical viewpoint but there are others which may be found in Duffin *et al.* [2] together with theorems applicable to the case of extended geometric programming.

When a solution to a given geometric programming problem is required the procedure is to form the dual function and then maximize this subject to the linear equations and nonnegativity constraints on the dual variables. A particularly easy instance arises where the linear equations have a unique solution which occurs when the number of equations in the dual constraints is the same as the number of dual variables. If it turns out that there are insufficient independent equations the difference between the number of variables and the number of independent linear equations is conventionally called the number of degrees of freedom. In the present case there are m orthogonality conditions, one for each variable x_m, a single normality condition and n dual variables giving a system with $n-m-1$ degrees of freedom. It has been suggested by Duffin and his colleagues that this quantity should be called the degree of difficulty. The degree of difficulty is then identical to the number of independent variables over which the dual function is to be maximized.

When the degree of difficulty is zero the objective function (2.1) is easily minimized by forming the dual function and solving the set of equations (2.6) and (2.7). The required minimum is then calculated by substituting these values for the dual variables into the dual function. The values for the primal variables at the minimum point can then be found from theorem 1, part III. If the degree of difficulty is greater than zero the procedure described will not yield values for all the dual functions. Solving the normality and orthogonality equations in this case leads to a set of dependent dual variables being expressed in terms of a set of independent ones. These independent variables are then equal in number to the degree of difficulty. The resulting expressions for the dual variables may then be substituted into (2.4) to produce a dual function which depends upon a set of known constants c_i and a set of unknown dual variables. The method of finding the maximum of the dual programme with a non zero degree of difficulty is left to a later section where it will also be seen that much information can be extracted from this formulation without ever finding a maximizing vector δ^*.

3. ILLUSTRATIVE EXAMPLE WITH ZERO DEGREE OF DIFFICULTY

Consider the two-bar pinned truss with tubular steel members shown in Fig. 1 which carries a vertical load of $2P$. This problem was also used by Fox [6] in a similar illustrative

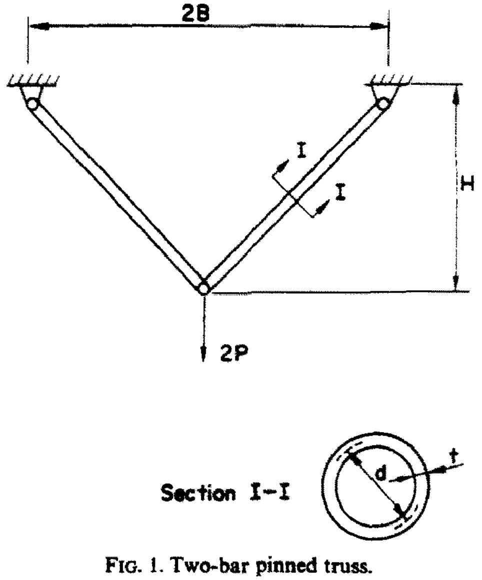

FIG. 1. Two-bar pinned truss.

role. Since the thickness of the tube t and the half span B are assumed fixed the problem is to find the minimum weight by selecting appropriate values for the mean diameter of the tubes d and the height H subject to the constraint that the stresses in the tubes should not exceed the yield stress. The following numerical values are taken: $P = 33{,}000$ lb; $t = 0{\cdot}1$ in.; $B = 30$ in.; σ_0 (yields stress) $= 60{,}000$ lb/in.^{2}; ρ (density of material) $= 0{\cdot}3$ lb/in.^{3}.

Thus the primal problem consists of minimizing the weight

$$W = \rho 2\pi dt(B^2+H^2)^{\frac{1}{2}}$$

$$= 0{\cdot}188d(900+H^2)^{\frac{1}{2}} \tag{3.1}$$

subject to

$$\frac{P}{\pi t}\frac{(B^2+H^2)^{\frac{1}{2}}}{dH} \le 60{,}000$$

which may be re-written as

$$1 \ge 1{\cdot}75\frac{(900+H^2)^{\frac{1}{2}}}{dH}. \tag{3.2}$$

Clearly both (3.1) and (3.2) are not posynomials due to the presence of the term $(900+H^2)^{\frac{1}{2}}$. This difficulty may be overcome by introducing a related function together with an additional independent variable and constraint. The related function for (3.1) is then

$$W = 0{\cdot}188dz$$

where z is the new independent variable, and the new constraint is,

$$1 \ge \frac{900}{z^2}+\frac{H^2}{z^2}.$$

Introducing the notation $x_1 = z$, $x_2 = H$, $x_3 = d$ the primal problem becomes

$$\text{minimize } W = 0{\cdot}188x_1x_3, \tag{3.3}$$

subject to

$$1 \ge 1{\cdot}75x_1x_2^{-1}x_3^{-1}, \tag{3.4}$$

$$1 \ge 900x_1^{-2}+x_2^2x_1^{-2}. \tag{3.5}$$

The associated dual problem is, therefore, to maximize

$$V(\delta) = \left(\frac{0{\cdot}188}{\delta_1}\right)^{\delta_1}(1{\cdot}75)^{\delta_2}\left(\frac{900}{\delta_3}\right)^{\delta_3}\left(\frac{1}{\delta_4}\right)^{\delta_4}(\delta_3+\delta_4)^{(\delta_3+\delta_4)}, \tag{3.6}$$

subject to the dual constraints,

$$\delta_i \ge 0 \qquad i = 1, 2, 3 \tag{3.7}$$

$$\delta_1 = 1, \text{(normality)} \tag{3.8}$$

and (orthogonality)

$$
\begin{aligned}
&x_1) \quad && \delta_1+\delta_2-2\delta_3-2\delta_4 = 0,\\
&x_2) \quad && -\delta_2 \qquad +2\delta_4 = 0, \qquad\qquad (3.9)\\
&x_3) \quad && \delta_1-\delta_2 \qquad\qquad = 0.
\end{aligned}
$$

Where the notation $x_i)$ is used to indicate that the equation has been derived by considering the orthogonality condition for the coefficients of the given x_i. The solution to the set of equations (3.7)–(3.9) is straightforward since the problem is one with zero degrees of difficulty. Hence the maximizing vector for the dual programme is given by,

$$\delta_1^* = 1, \qquad \delta_2^* = 1, \qquad \delta_3^* = \tfrac{1}{2}, \qquad \delta_4^* = \tfrac{1}{2},$$

and the maximum value of the dual function [and thus the minimum for (3.1)]

$$V(\bar{\delta}^*) = \left(\frac{0{\cdot}188}{1}\right)^1 (1{\cdot}75)^1 \left(\frac{900}{1/2}\right)^{\frac{1}{2}} \left(\frac{1}{1/2}\right)^{\frac{1}{2}} \left(\frac{1}{2}+\frac{1}{2}\right)^{(\frac{1}{2}+\frac{1}{2})}$$

which yields

$$V(\bar{\delta}^*) = 19{\cdot}8 \text{ lb}.$$

The minimizing values for the individual x_i's can now be obtained directly by applying equations (2.8) and (2.9). The value of x_1 can be immediately found from the first term of the constraint (3.5),

$$\frac{\delta_3}{\delta_3+\delta_4} = \tfrac{1}{2} = 900 x_1^{-2}$$

therefore

$$x_1 = 42{\cdot}426$$

similarly x_2 is found from

$$\frac{\delta_4}{\delta_3+\delta_4} = \tfrac{1}{2} = x_2^2 x_1^2,$$

giving

$$x_2 = H = 30,$$

and x_3 is found from (3.4) to be 2·475. Hence the solution to the primal problem is,

$$W = 19{\cdot}8 \text{ lb}$$

with

$$H = 30, \qquad d = 2{\cdot}475.$$

Achieving a solution vector δ^* to the dual problem by using only equations (3.7)–(3.9) focuses attention on an important invariance property of zero degree of difficulty geometric programmes. This states that solution vectors to the dual programme are independent of the coefficients c_i. Thus, if one solution has already been found, a variation in the maximum value of the dual function for variations in the primal coefficients c_i can be obtained simply by substituting the modified c_i values into (3.6) and calculating $V(\delta^*)$ with the aid of the known δ^*'s since these remain the same. In this way one complete solution giving a maximum value to a given problem is the key to a whole family of related maxima (or minima).

4. PROBLEMS WITH NON-ZERO DEGREE OF DIFFICULTY

Turning our attention to a geometric programme with positive degree of difficulty d, it is still convenient to operate with the dual programme. The first step in obtaining a solution is to construct basis vectors $\bar{b}^{(j)}$, $j = 0, 1, 2 \ldots d$ so that the general solution to the dual constraints is

$$\bar{\delta} = b^{(0)} + \sum_{j=1}^{d} r_j \bar{b}. \tag{4.1}$$

The r_j are arbitrary real numbers satisfying the positivity condition

$$b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)} \geq 0, \qquad i = 1, 2, \ldots, n_1$$

where $b_i^{(j)}$ is the ith component of the vector $\bar{b}^{(j)}$. In this new formulation the variables r_j are called *basic variables* and the vector $\bar{b}^{(0)}$, which satisfies both normality and orthogonality, is the *normality vector*. The vectors $\bar{b}^{(j)}, j = 1, 2, \ldots, d$ which are linearly independent solutions to the homogeneous counterpart of the normality and orthogonality conditions are termed nullity vectors.

The dual programme has now undergone a transformation and may be re-formulated in the following manner:

Transformed dual programme

Find the maximum value of a product function

$$V(\bar{r}) = \left[\prod_{i=1}^{n} \left(\frac{c_i}{\delta_i(\bar{r})}\right)^{\delta_i(\bar{r})}\right] \prod_{k=1}^{p} \lambda_k(\bar{r})^{\lambda_k(\bar{r})} \tag{4.2}$$

where

$$\delta_i(\bar{r}) = b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)}, \qquad i = 1 \ldots n \tag{4.3}$$

and

$$\lambda_k(\bar{r}) = \lambda_k^{(0)} + \sum_{j=1}^{d} r_j \lambda_k^{(j)}, \qquad k = 1, \ldots, p. \tag{4.4}$$

The vector $\bar{r}$ is subject to the positivity constraints

$$b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)} \geq 0, \qquad i = 1, \ldots, n \tag{4.5}$$

and

$$\lambda_k^{(i)} = \sum_{j=l_k}^{n_k} b_j^{(i)} \qquad i = 0, 1, \ldots, d. \tag{4.6}$$

The factor c_i and l_k have the same meaning here as in the definition of the primal programme. The normality vector $\bar{b}^{(0)}$ satisfies the normality condition

$$\sum_{i=1}^{n_0} b_i^{(0)} = 1 \tag{4.7}$$

and the orthogonality condition.

$$\sum_{i=1}^{n} a_{ij} b_i^{(0)} = 0, \qquad j = 1, \ldots, m. \tag{4.8}$$

The matrix (a_{ij}) is the exponent matrix of the primal programme.

Since the basis vector $\bar{b}$ is obtained directly from the exponent matrix and is thus known, a lower bound on the primal function can be obtained by simply finding any vector $\bar{r}$ whose components satisfy equations (4.5) and then calculating the value of the transformed dual function $V(\bar{r})$. An upper bound can be obtained by selecting a set of primal variables which do not cause the primal constraints to be violated and then calculating a new value for the primal function. On many occasions a few calculations performed in this way can achieve close bounds on the desired optimum and are able to rapidly indicate the closeness or otherwise of a given design to its optimum. This is of practical importance in many areas of engineering design where long experience can allow an engineer to use his intuition and obtain a near optimum design without the need of special optimizing routines.

If it is necessary to find the maximum to the transformed function $V(\bar{r})$ this, as we have seen, requires that the normality and nullity vectors $b^{(J)}$ be determined from the exponent matrix (a_{ij}). The method normally adopted in determining these vectors closely parallels that employed by Brand [7] in his matrix algebraic treatment of the Pi theorem of dimensional analysis. An explanation of the details of this procedure are left to a later paper which will deal with computational aspects of geometric programming.

Once the nullity and normality vectors have been determined the problem reduces to one of maximizing $V(\bar{r})$ subject to the single set of constraints (4.5). At this stage it is appropriate to use a sequential minimization (maximization) scheme to move to a complete solution and in the present context a suitably modified version of the method of "conjugate directions" due to Powell [8] proved to be satisfactory. However, $V(\bar{r})$ is not always concave and it is better to use log $V(\bar{r})$ as the cost function, when sequential methods are employed, since the logarithmic function does not suffer this disadvantage. It is, indeed, one of the remarkable features of this method that even if the primal programme has a set of local minima the function log $V(\bar{r})$ is always strictly concave.

A computer program has been written on the basis of the procedure explained in this section which requires, as input data, a knowledge of the exponent matrix together with other details of the primal problem. The routine then calculates the normality and nullity vectors and hands over the problem to Powell's subroutine which finds the minimum of 1/log $V(\bar{r})$ in terms of the variables r_j. If it is not necessary to find the maximizing vector or if good starting values are required before initiating the Powell routine provision for calculating lower bounds on the maximum $V(\bar{r})$, by selecting values for the r_j's which satisfy equations (4.5), is incorporated in the routine.

5. A DESIGN CODE PROBLEM

The problem to be considered is that shown in Figs. 2 and 3 and consists of optimizing the design of a vertically corrugated transverse bulkhead of an oil tanker. The cost function is taken to be the total weight of the bulkhead and is subject to constraints on performance

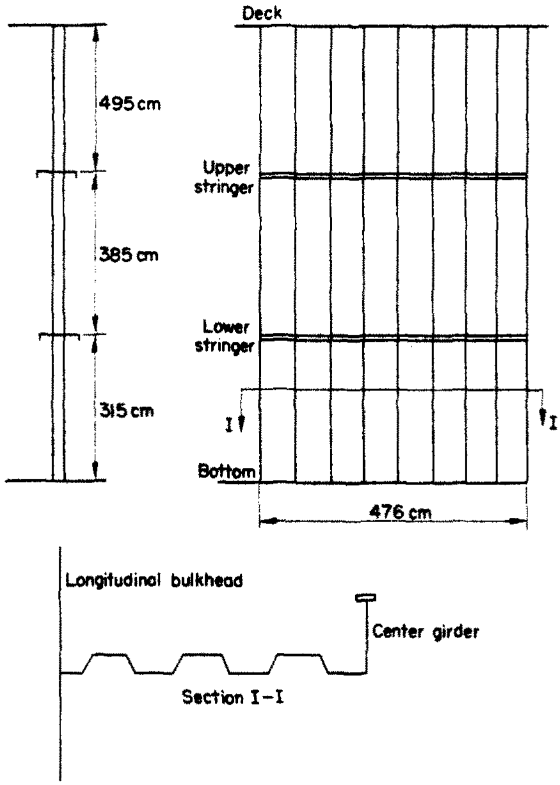

FIG. 2. Bulkhead layout.

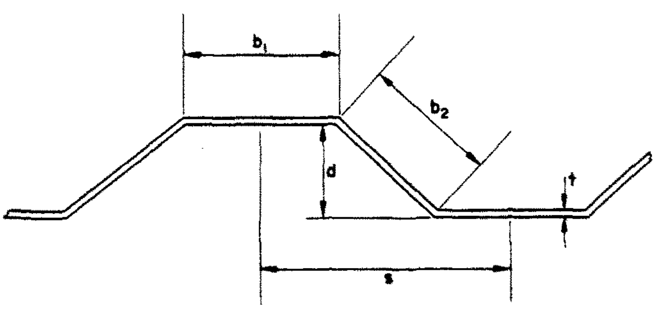

FIG. 3. Description of variables.

characteristics and on certain dimensions imposed by specifications laid down in *Det Norske Veritas*. This problem has been discussed by Kavlie *et al.* [9] and more recently by Bracken and McCormick [10], in both cases the solution was sought using one of the normal sequential techniques.

In formulating the problem it is assumed that the shape of the corrugations are identical in all the panels and that the positions of the stringers and the width of the bulkheads are fixed. The basic design variables are shown in Fig. 3 for one panel and thus for all three panels are:

b_1 = width of flange (cm);
b_2 = length of web (cm);
d = depth of corrugation (cm);
t_t = thickness of plate in top panel (cm);
t_m = thickness of plate in middle panel (cm);
t_s = thickness of plate in bottom panel (cm).

By means of these variables and the distance s, defined in Fig. 3, the total weight may be defined as,

$$W = \frac{\gamma 476(b_1+b_2)(495t_t+385t_m+315t_b)}{s} \tag{5.1}$$

where γ = the density of the material in ton/cm^3.

This cost function (5.1) is then subject to constraints on the section modules for each panel, on the moments of inertia and on the minimum plate thickness all of which are imposed by D.N.V. This procedure leads to a problem requiring the minimization of (5.1) subject to 16 constraints which can be solved by the geometric programming subroutine without much difficulty and yields answers which agree with those given by Bracken and McCormick.

In discussing the problem in detail it is more convenient not to deal with the entire problem together with its 16 constraints but to examine the optimizing of the bottom panel alone. In this case the cost function (5.1) reduces to

$$W = g(0) = 1{\cdot}177x_4(x_1+x_3)/x_2, \tag{5.2}$$

where the density γ ($7{\cdot}850 \times 10^{-6}$) which has been used is that employed by Kavlie *et al.* [9]. The constraint equations are not explained in detail in the present formulation since this aspect is adequately covered in Refs. [9, 10], but in the case of the bottom panel these are found to be

$$\left.\begin{aligned}
1 &\geq g(1) = \frac{53{\cdot}64x_2x_4^{-1}}{(2{\cdot}4x_1+x_3)[x_3^2-(x_2-x_1)^2]^{\frac{1}{2}}},\\
1 &\geq g(2) = \frac{26{\cdot}4(8{\cdot}94x_2)^{\frac{1}{2}}x_4^{-1}}{(2{\cdot}4x_1+x_3)[x_3^2-(x_2-x_1)^2]},\\
1 &\geq g(3) = 0{\cdot}0156x_1x_4^{-1}+0{\cdot}15x_4^{-1}\\
1 &\geq g(4) = 0{\cdot}0156x_3x_4^{-1}+0{\cdot}15x_4^{-1}\\
1 &\geq g(5) = 1{\cdot}05x_4^{-1}
\end{aligned}\right\} \tag{5.3}$$

where in both (5.2) and (5.3) the free variables are,

$$x_1 = b_1, \quad x_2 = s, \quad x_3 = b_2 \quad \text{and} \quad x_4 = t_b.$$

The first thing to notice in the constraint equations is that neither g(1) nor g(2) are in posynomial form due to the presence of the two terms $(2{\cdot}4x_1 + x_3)$ and $x_3^2 - (x_2 - x_1)^2$ in the denominators. The first of these expressions can be approximated by using the formulae given in the appendix and to this end (A.4), (A.5) are selected rather than (A.1) because the resulting approximation form is accurate for wide variations of x_1, x_3. The second term $[x_3^2 - (x_2 - x_1)^2]$ could be dealt with in exactly the same manner but it is more accurate to introduce a related function as in Section 3 and then to approximate the remaining non-posynomial terms. Thus a new variable x_5 is introduced and a new constraint

$$1 \geq x_5^2 x_3^{-2} + (x_2 - x_1)^2 x_3^{-2} \tag{5.4}$$

where the term $(x_2 - x_1)^2$ requires an approximate posynomial form obtained from (A.4), (A.5).

Thus, the primal problem is to minimize,

$$g(0) = 1{\cdot}177(x_1 x_2^{-1} x_4 + x_2^{-1} x_3 x_4), \tag{5.5}$$

subject to the primal constraints,

$$\left.\begin{aligned}
1 &\geq g(1) = 15{\cdot}77 x_1^{-0{\cdot}706} x_2 x_3^{-0{\cdot}294} x_4^{-1} x_5^{-1}, \\
1 &\geq g(2) = 143{\cdot}65 x_1^{-0{\cdot}706} x_2^{1{\cdot}333} x_3^{-0{\cdot}294} x_4^{-1} x_5^{-2}, \\
1 &\geq g(3) = x_5^2 x_3^{-2} + 0{\cdot}045 x_1^{-2{\cdot}5} x_2^{4{\cdot}5} x_3^{-2}, \\
1 &\geq g(4) = 0{\cdot}0156 x_1 x_4^{-1} + 0{\cdot}15 x_4^{-1}, \\
1 &\geq g(5) = 0{\cdot}0156 x_3 x_4^{-1} + 0{\cdot}15 x_4^{-1}, \\
1 &\geq g(6) = 1{\cdot}05 x_4^{-1},
\end{aligned}\right\} \tag{5.6}$$

with the approximating point chosen as $x_1^* = x_3^* = 50$, $x_2^* = 90$. It is at this stage that the problem is handed over to the computer and the description which follows gives an outline of the method whereby a solution is achieved and, in the main, uses the information obtained from the subroutine. The dual of equation (5.5) is to maximize,

$$\begin{aligned}
V = &\left(\frac{1{\cdot}177}{\delta_1}\right)^{\delta_1} \left(\frac{1{\cdot}177}{\delta_2}\right)^{\delta_2} (15{\cdot}77)^{\delta_3} (143{\cdot}65)^{\delta_4} \left(\frac{1}{\delta_5}\right)^{\delta_5} \left(\frac{0{\cdot}045}{\delta_6}\right)^{\delta_6} \left(\frac{0{\cdot}0156}{\delta_7}\right)^{\delta_7} \\
&\left(\frac{0{\cdot}15}{\delta_8}\right)^{\delta_8} \left(\frac{0{\cdot}0156}{\delta_9}\right)^{\delta_9} \left(\frac{0{\cdot}15}{\delta_{10}}\right)^{\delta_{10}} (1{\cdot}05)^{\delta_{11}} (\delta_5 + \delta_6)^{(\delta_5 + \delta_6)} (\delta_7 + \delta_8)^{(\delta_7 + \delta_8)} (\delta_9 + \delta_{10})^{(\delta_9 + \delta_{10})}
\end{aligned} \tag{5.7}$$

subject to the normality condition

$$\delta_1 + \delta_2 = 1$$

and the orthogonality conditions

$$\begin{pmatrix} 1 & 0 & -0.706 & -0.706 & 0 & -2.5 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1.333 & 0 & 4.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.294 & -0.294 & -2 & -2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & -2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \\ \delta_8 \\ \delta_9 \\ \delta_{10} \\ \delta_{11} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{5.8}$$

together with the positivity conditions $\delta_i \geq 0, i = 1, \ldots, 11$.

The problem must now be transformed in terms of the variables r_j and suitable normality and nullity vectors found. In order to find these vectors the algebraic Pi theorem analysis [7] requires that the matrix comprising the first five columns of (5.8) be non-singular, which turns out not to be true for the present formulation. In order to overcome the difficulty it is only necessary to interchange the columns of (5.8) until a suitable matrix is found and simply placing δ_3 beneath δ_{11} and moving all the dual variables, with the exception of δ_1 and δ_2, up one place is sufficient to give a non-singular matrix. Performing the algebraic operations gives a set of transformed dual variables

$$\left.\begin{aligned} \delta_1(r) &= -0.077947+0.642501r_1+0.625047r_2+0.642501r_3+0.642501r_4 \\ &\quad +1.065801r_5, \\ \delta_2(r) &= 1.077947-0.642501r_1-0.625047r_2-0.642501r_3-0.642501r_4 \\ &\quad -1.065801r_5, \\ \delta_3(r) &= 0.374953-0.374953r_1-0.34953r_3-0.374953r_4-1.87420r_5 \\ \delta_4(r) &= 0.374953-0.374953r_1-0.374953r_3-0.374953r_4-0.624578r_5, \\ \delta_5(r) &= 0.108902+0.10882r_1+0.10882r_3+0.108820r_4-0.000204r_5, \\ \delta_6(r) &= 0.625047-0.625047r_1-0.625047r_2-0.625047r_3-0.625047r_4 \\ &\quad -0.625047r_5, \\ \delta_7(r) &= r_1, \\ \delta_8(r) &= 0.625047r_2, \\ \delta_9(r) &= r_3, \\ \delta_{10}(r) &= r_4, \\ \delta_{11}(r) &= 2.49925r_5 \end{aligned}\right\} \tag{5.9}$$

where the basic variables r_j must comply with (4.5). With the aid of (4.5), (5.9) and (5.7) we can find lower bounds on the function to be minimized. For example, setting

$$r_2 = r_3 = r_4 = r_5 = 0$$

means that a satisfactory value for r_1 must lie in the range

$$1 \geq r_1 \geq 0{\cdot}121318.$$

Any number from within this range may be selected and substituted into (5.9) to give numerical values to the dual variables which can then be used to calculate $V(\bar{r})$ (5.7). If $r_1 = 0{\cdot}5$ is selected the vector $(\delta_1, \delta_2, \ldots, \delta_{11})$ is (0·243, 0·757, 0·187, 0·187, 0·163, 0·312, 0·500, 0, 0, 0, 0) which gives $V(\bar{r}) = 1{\cdot}11$. However, the original design given by Kavlie *et al.* [9] before any optimization was attempted gave a weight 1·79 tons for the bottom panel. Thus the minimum weight for this panel is $1{\cdot}45 \pm 0{\cdot}34$ tons and a trivial calculation has yielded an answer accurate to within a maximum possible error of 23·4 per cent.

Two courses now lie open either, these bounds can be improved, or the computer routine can be allowed to initiate the final sequential search subroutine. In fact a compromise between the two alternatives was sought in that an improved lower bound was found and the values for the basic variables used as the starting values for the final sequential part. The eventual result gave the minimum weight of the bottom panel as 1·35 tons and this is achieved with a minimizing vector $x_1 = 57{\cdot}69$, $x_2 = 105{\cdot}52$, $x_3 = 57{\cdot}69$, $x_4 = 1{\cdot}05$. Obtaining this result required approximately 5 sec of central processing time on the Royal Aircraft Establishment's ICL 1907 computer. This weight remained unchanged (to two decimal places) when the values for the operating point in the approximating scheme were varied by ± 20 per cent though changes in the minimizing vector of about 1 per cent were noted. The comparison result given in Ref. [9] was $x_1 = 56{\cdot}3$, $x_2 = 100{\cdot}8$, $x_3 = 58$, $x_4 = 1{\cdot}05$ and a minimum weight of 1·40 ton but this appears to be in error since it violates one of the constraints.

6. ASCERTAINING THE INFLUENCE OF CONSTRAINTS

The ability of geometric programming to find lower bounds on the minimum value for the objective function permits the taking of certain liberties which would be totally inappropriate in the context of a standard sequential minimization routine. Consider the problem of the previous section, if we had decided to seek a lower bound on the minimum at the point where the equations (5.8) had been formulated it would have been quite acceptable to put $\delta_5 = \delta_6 = 0$. Such a procedure would require that the terms $(1/\delta_5)^{\delta_5}$, $(0{\cdot}045/\delta_6)^{\delta_6}$ in the dual function (5.7) be put equal to 1 and any influence exerted on the solution by the constraint $g(3) \leq 1$ would disappear. Thus removing a constraint, or groups of constraints, from the primal problem simply gives rise to a lower bound on the solution. The structural engineer is then at liberty to ascertain the influence of the constraints by solving a group of simple problems each using a different selection of the total number of constraint equations. In many instances these simple problems may be so arranged that they reduce to zero degree of difficulty.

7. CONCLUSIONS

The preceding sections have shown how the geometric programming technique may be used to solve certain classes of structural optimization problems. In particular emphasis has been placed on the computerized version of this method which can be made to handle large numbers of variables with the aid of a bounding technique. At the present moment this automated version can only deal with posynomial formulations and any approximation must be done before the problem is given over to the computer. However, judging from hand calculations there is no reason why the program should not be extended to form its own approximate posynomials and then proceed to solve a sequence of geometric programmes. In this situation it would be perfectly feasible for the method to handle complicated constraints that are not expressible as explicit functions of the design variables. It would then be possible to optimize statically indeterminant structures both with and without variable geometries. Of course, an evaluation of the efficiency of geometric programming in performing this type of calculation must await the appearance of numerical results.

REFERENCES

[1] C. M. ZENER, A mathematical aid in optimising engineering designs. *Proc. natn. Acad. Sci.* **47**, 537–539 (1961).
[2] R. J. DUFFIN, E. L. PETERSON and C. M. ZENER, *Geometric Programming*. John Wiley (1967).
[3] D. J. WILDE and C. S. BEIGHTLER, *Foundations of Optimisation*. Prentice-Hall (1967).
[4] E. L. PETERSON, Symmetric duality for generalised unconstrained geometric programming. *J. appl. Math.* **19**, 487–526 (1970).
[5] G. G. POPE and L. A. SCHMIT, Structural Design Applications of Mathematical Programming Techniques, AGARDograph No. 149, AGARD-AG-149-71 (1971).
[6] R. L. FOX, Notes on Mathematical Methods in Optimization, Seminar on Structural Optimization, University of Waterloo, Canada (1968).
[7] L. BRAND, The Pi theorem of dimensional analysis. *Archs. ration. Mech. Analysis* **1**, 33–47 (1957).
[8] M. J. D. POWELL, An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**, 155–162 (1964).
[9] D. KAVLIE, J. KOWALIK and J. MOE, Structural Optimisation by means of Non-linear Programming, *International Symposium on the Use of Digital Computers in Structural Engineering*, University of Newcastle upon Tyne (1967).
[10] J. BRACKEN and G. MCCORMICK, *Selected Applications of Non-linear Programming*. John Wiley (1968).
[11] M. AVRIEL and A. C. WILLIAMS, Complementary geometric programming. *J. appl. Math.* **19**, 125–141 (1970).

APPENDIX

In this section several transformations are discussed which permit the primal programme to include functions which are more general than posynomials. Suppose that the objective function or one of the constraint equations contains a positive function of the form

$$F = \frac{f_1(x)}{f_2(x)},$$

where either or both of f_1 and f_2 could be non-posynomial forms. If f_1 is defined by a function of the form

$$\left(\sum_{i=1}^{r} c_i x_1^{a_{i1}} \dots x_m^{a_{im}}\right)^n$$

where all the c_i's are positive real numbers and n is any real number we may proceed as in Section 3 and introduce a related variable Z_1 such that $f_1 \leq Z_1$. Similarly if f_2 is defined by a function of the form

$$\left(\sum_{i=1}^{r} c_i x_1^{a_{i1}} \dots x_m^{a_{im}}\right)^n$$

where c_1 is positive but all other c_i's are negative and n is any real number then following the procedure of Section 5 we use the related function Z_2 such that $f_2 \geq Z_2$. The inequalities in the expressions relating the f's and Z's are important since they ensure that f_1 and Z_1 share the same infinum and f_2 and Z_2 share the same supremum. Thus any approximation scheme must satisfy the same inequality relationships and if we have $f'_1 \simeq f_1$ and $f'_2 \simeq f_2$ then $f_1 \leq f'_1$ and $f_2 \geq f'_2$.

In examining the expressions for f_1 and f_2 in the above paragraph it may be observed that the idea of using a related function for these terms cannot be applied when some of the c_i's in the form for f_1 are negative or some of the c_i's in the form for f_2 are positive (other than c_1). This occurs because the related constraint equation which must be introduced when either Z_1 or Z_2 are used does not have the required posynomial form. However, before discussing a general method of approximation consider the case when both f_1 and f_2 are ordinary posynomials with f_2 having at least two terms rendering the function F a non-posynomial form. In order to deal with this situation it is possible to employ the geometric inequality on f_2 such that

$$f_2 = \sum_{i=1}^{r} c_i x_i^{a_{i1}}, \dots, x_m^{a_{im}} \geq \prod_{i=1}^{r} \left(\frac{c_i x_i^{a_{i1}} \dots x_m^{a_{im}}}{\delta_i}\right)^{\delta_i} = f'_2, \tag{A.1}$$

where

$$\delta_1 + \delta_2 + \dots + \delta_m = 1.$$

Thus values for the δ_i's must be chosen and subsequently improved by iteration. Consider now the cases when

$$f_1 = \left(\sum_{i=1}^{r} c_i x_1^{a_{i1}} \dots a_m^{a_{im}}\right)^n \tag{A.2}$$

where c_1 is positive and all other c_i's are negative, and

$$f_2 = \left(\sum_{i=1}^{r} c_i x_1^{a_{i1}} \dots a_m^{a_{im}}\right)^n \tag{A.3}$$

where all the c_i's are positive. If mixed cases occur when some of the c_i's in (A.2) are positive and some of the c_i's in (A.3) are negative these coefficients can be removed by introducing related functions and we return to forms defined in (A.2) and (A.3). It is immediately clear that the approximation scheme introduced at (A.1) may be generalized to cope with (A.3), and since (A.2) can be transformed into the form of (A.3) then,

$$f_2 = \left(\sum_{i=1}^{r} c_i x_1^{a_{i1}} \dots x_m^{a_{im}}\right)^n \geq \left\{\prod_{i=1}^{r} \left(\frac{c_i x_1^{a_{i1}} \dots x_m^{a_{im}}}{\delta_i}\right)^{\delta_i}\right\}^n = f'_2$$

where, as before, $\delta_1 + \delta_2 + \dots + \delta_r = 1$ and the correct values for the individual δ's must be chosen by means of an iteration process. However, Duffin *et al.* [2] have presented an alternative approximation scheme which may be used for either f_1 or f_2. If it is necessary to find an approximate form $h'(x)$ (which might be f_1 or f_2) for a function $h(x)$ then an appropriate single-term posynomial is given by

$$h'(\bar{x}) \simeq h(\bar{x}^*)\left(\frac{x_1}{x_1^*}\right)^{a_1}\left(\frac{x_2}{x_2^*}\right)^{a_2}\cdots\left(\frac{x_m}{x_m^*}\right)^{a_m}, \tag{A.4}$$

where

$$a_j = \left(\frac{x_j}{h}\frac{\partial h}{\partial x_j}\right)_{\bar{x}=\bar{x}^*} \quad j = 1, 2, \ldots, m. \tag{A.5}$$

The vector $(x_1^*, \bar{x}_2^*, \ldots, \bar{x}_m^*)$ is an estimate to the mean value of the range of the variables and is called the operating point. Once this point has been chosen and the formulae (A.4) and (A.5) applied the ordinary techniques of geometric programming may be used to find a minimum to the primal problem. However, if the operating point turns out to be distant from the minimizing point it is necessary to up-date the values of $x_1^*, x_2^*, \ldots, x_m^*$ and repeat the calculation.

Although Duffin *et al.* introduce (A.4) they do not indicate whether or not it satisfies the inequality conditions enumerated in the first paragraph of this section. In order to answer this point consider the expression

$$h^*/h = R(\bar{x}')$$

where $R(\bar{x})$ is equivalent to the Lagrangian form of the remainder in Taylor's series and is evaluated somewhere between $\bar{x}^*$ and the current value of $\bar{x}$, namely $\bar{x}'$. A fairly simple calculation reveals that

$$R(\bar{x}') = \left(\frac{x_1}{x_1'}\right)^{\mu_1}\left(\frac{x_2}{x_2'}\right)^{\mu_2}\cdots\left(\frac{x_m}{x_m'}\right)^{\mu_m},$$

where for the cases under discussion the parameters $\mu_i, i = 1 \ldots m$, take the form such that for f_1 the approximating function δ'_1, found by using (A.4), (A.5), gives $f_1 \leq f'_1$ and for $f_2, f_2 \geq f'_2$.

In the case of the strictly posynomial term f_2 the form of (A.4) is exactly that given by Avriel and Williams [11] in their discussion of complementary geometric programming.

(*Received* 9 *August* 1971; *revised* 3 *January* 1972)

Абстракт—Даются некоторые подробности метода геометрического программирования, вместе с выяснением использования этой задачи для получения решений, касающихся некоторых задач оптимализации конструкций. Особенно подчеркивается пригодность метода и возможность получения болучения более низких пределов для минимального значения заданной функции, путём несложных расчетов. Указывается также возможность использования метода дла быстрого выяснения влияния всех ограничений, наложенных на заданную функцию, с целвю пропуска огранигелий без никого действия, зачем начнется основной расчет минимализации. Обсуждается также возможность обобщения метода, путем использования приближенных расчетов.