Discussion Paper No. 13

THE DECOMPOSITION OF LARGE (GENERALIZED) GEOMETRIC PROGRAMMING PROBLEMS BY TEARING

by

Elmor L. Peterson

September 25, 1972

Abstract: Decomposition principles are developed for those (generalized) geometric programming problems whose matrices have a certain type of block diagonal structure and whose functions are (at least partially) separable (in a certain compatible way).

TABLE OF CONTENTS

THE DECOMPOSITION OF LARGE (GENERALIZED) GEOMETRIC PROGRAMMING PROBLEMS BY TEARING

ELMOR L. PETERSON
Department of Industrial Engineering/Management Sciences,
and Department of Mathematics
Northwestern University, Evanston, Illinois, 60201, U.S.A.

Abstract: Decomposition principles are developed for those (generalized) geometric programming problems whose matrices have a certain type of block diagonal structure and whose functions are (at least partially) separable (in a certain compatible way).

1. INTRODUCTION

Geometric programming in its most general form provides a mechanism for reformulating many important inseparable optimization problems as separable (generalized) geometric programming problems.

The key to such reformulations is the exploitation of the linearities that are present in a given optimization problem. Such linearities frequently appear as linear equations or linear inequalities, but they can also appear in much more subtle guises as matrices associated with nonlinearities.

The geometric programming problems that result from such reformulations are (at least partially) separable, and many that arise from the modeling of large systems have sparse matrices. Moreover, such separability and sparsity can frequently be exploited by using the decomposition principles to be developed here.

Initial work on geometric programming decomposition was done by Zener (1964), but only for a special type of unconstrained "posynomial" problem that occurs in the design of heat exchangers. However, subsequent extensions and generalizations to various types of constrained posynomial problems have been made by Heymann and Avriel (1969), Ecker (1972), and Rijckaert (1973).

This paper presents two geometric programming decomposition principles that transcend the special class of constrained posynomial problems. The first principle is the most basic and forms an integral part of the second principle. The second principle can be viewed as primarily another manifestation of the general method of "diakoptics" as introduced by Kron (1953). (Diakoptics is a term that stems from the Greek word "kopto" meaning to break or tear apart and the Greek prefix "dia" which reinforces the word that it precedes). The second principle can further be viewed as a nonlinear generalization of the "dualplex method" of Gass (1966), which (according to Dantzig (1970)) is related to the "partition programming" of Rosen (1964) in dual form.

The author suspects that the second principle also generalizes and/or unifies many nonlinear programming decomposition principles that appear in diverse disciplines and fields of application. In fact, the author hopes that this paper will stimulate and facilitate a unification and further development of such principles.

With that goal in mind this paper has been made as self-contained as possible, with a minimum of detailed technical developments. In particular, the geometric programming approach is introduced in section 2 in the simplest possible setting - programming without explicit constraints. Two decomposable problem types are then described in section 3 and are subsequently decomposed in section 4. The first type is decomposed simply by inspection, and the second

type is reduced to the first type by the method of tearing (i.e. diakoptics), guided in some cases by geometric programming duality. The resulting decomposition principles are then specialized in section 5 to the seemingly more general setting of programming with explicit constraints.

2. THE GEOMETRIC PROGRAMMING APPROACH

Classical optimization theory and ordinary mathematical programming are concerned with the minimization (or maximization) of an arbitrary real-valued function g over some given subset S of the function's nonempty domain C. For pedagogical simplicity we shall restrict our attention to the finite-dimensional case in which C is itself a subset of n-dimensional Euclidean space E_n.

In (generalized) geometric programming, as defined by Peterson (1970), the subset S is required to be the intersect of C with an arbitrary cone $X \subseteq E_n$. However, for both practical and theoretical reasons, this problem of minimizing g over $X \cap C$ is not studied in isolation. It is first embedded in the family $\mathcal{A}$ of closely related minimization problems $\mathcal{A}(u)$ that are generated by simply translating (the domain C of) g through all possible displacements $-u \in E_n$ while keeping X fixed. (For gaining insight we recommend making a sketch of a typical case in which X is a one dimensional subspace of E_2. In most known cases of practical significance the cone X is in fact a subspace of E_n.) The problem of minimizing g over $X \cap C$ appears in the family $\mathcal{A}$ as problem $\mathcal{A}(0)$ and is studied in relation to all other problems $\mathcal{A}(u)$, with special attention given to those problems $\mathcal{A}(u)$ that are close to $\mathcal{A}(0)$ in the sense that (the norm of) u is small.

Each problem $\mathcal{A}(u)$ is said to be a <u>geometric programming problem</u>, and the family $\mathcal{A}$ of all such problems (for fixed g, C and X) is termed a <u>geometric programming family</u>. For purposes of easy reference and mathematical precision, problem $\mathcal{A}(u)$ is now given the following formal definition in terms of classical terminology and notation.

PROBLEM $\mathcal{A}(u)$. <u>Using the "feasible solution" set</u>

$$S(u) \triangleq X \cap (C - u),$$

<u>calculate both the "problem infimum"</u>

$$\varphi(u) \triangleq \inf_{x \in S(u)} g(x+u)$$

<u>and the "optimal solution" set</u>

$$S^*(u) \triangleq \{ x \in S(u) \mid g(x+u) = \varphi(u) \}.$$

For a given u, problem $\mathcal{A}(u)$ is either "consistent" or "inconsistent", depending on whether the feasible solution set $S(u)$ is nonempty or empty. It is of course obvious that the family $\mathcal{A}$ contains infinitely many consistent problems $\mathcal{A}(u)$. The domain of the infimum function φ is taken to be the corresponding nonempty set

$$U \triangleq \{ u \in E_n \mid \text{problem } \mathcal{A}(u) \text{ is consistent} \}.$$

Thus, the range of φ may contain the point $-\infty$; but if $\varphi(u) = -\infty$, then the optimal solution set $S^*(u)$ is clearly empty.

Each optimization problem can generally be put into the form of the geometric programming problem $\mathcal{A}(0)$ in more than one way by suitably choosing the function g and the cone X. For example, one can always let g be the "objective function" for the given optimization problem simply by choosing X to be E_n, but we shall soon see that such a choice is generally not the best possible choice for most important optimization problems. The reason is that most such problems involve a certain amount of linearity (due to the presence of linear equations, linear inequalities, matrices, etc.), which can be conveniently handled through the introduction of an appropriate <u>nontrivial</u> subcone $X \subset E_n$. The presence of

such a subcone $\mathcal{K}$ is one of the distinguishing features of the geometric programming point of view.

Due to the pre-eminence of problem $\mathcal{A}(0)$ we shall find it useful to interpret problem $\mathcal{A}(u)$ as a perturbed version of $\mathcal{A}(0)$, so we term the set $\mathcal{U}$ the <u>feasible perturbation set</u> for problem $\mathcal{A}(0)$. Actually, a rather elementary argument (given essentially by Peterson (1970)) shows that

$$\mathcal{U} = \mathcal{C} - \mathcal{K}.$$

This kind of perturbation analysis is intimately related to the geometric programming duality previously studied by Peterson (1970), and it is the only ingredient that must be combined with our first decomposition principle to obtain our second decomposition principle. In fact, for a given problem $\mathcal{A}(0)$ that is of a certain decomposable type 2 the infimum function $\varphi^1:\mathcal{U}^1$ that results from replacing $\mathcal{K}$ with an appropriate "separable subcone" $\mathcal{K}^1 \subset \mathcal{K}$ turns out to be the objective function for a corresponding "master problem" when the domain $\mathcal{U}^1$ is restricted to its intersect with an appropriate "coupling subcone" $\mathcal{K}_0 \subset \mathcal{K}$ whose sum with $\mathcal{K}^1$ is $\mathcal{K}$.

This kind of perturbation analysis is also of direct practical interest. In particular, the following examples of important problem classes indicate that the functions $\varphi:\mathcal{U}$ and $g^*:\mathcal{U}$ often show the dependence of optimality on actual external influences, such as available resources, design requirements, materiel costs, and the data being optimally approximated in linear regression analysis.

<u>Example 1.</u> Perhaps the most striking example of the utility of the geometric programming approach comes from using it to study the minimization of "signomials". This was first done by Zener (1961, 1962) and Duffin (1962), and served as the initial development (as well as the main stimulus for subsequent developments) of geometric programming.

A <u>signomial</u> (sometimes termed a "generalized polynomial") is any function with the form

$$P(t) = \sum_{i=1}^{n} c_i t_1^{a_{i1}} t_2^{a_{i2}} \cdots t_m^{a_{im}},$$

where the coefficients c_i and the exponents a_{ij} are arbitrary real constants but the independent variables t_j are restricted to be positive. After much experience in the physical sciences and engineering Zener clearly recognized that many optimization problems of practical importance can be accurately modeled with such functions. In many cases they come directly from the laws of nature and/or the laws of economics. In other cases this functional form gives a good fit to empirical data over a wide range of the variables t_j.

The presence of the "exponent matrix" (a_{ij}) (which is of course associated with algebraic nonlinearities) is the key to applying geometric programming to signomial optimization. To effectively place the problem of minimizing $P(t)$ in the format of problem $\mathcal{A}(0)$, simply make the change of variables

$$x_i = \sum_{j=1}^{m} a_{ij} \log t_j, \quad i = 1,2,\ldots,n;$$

and then use the laws of exponents to infer that minimizing $P(t)$ is equivalent to solving problem $\mathcal{A}(0)$ when

$$g(x) \triangleq \sum_{i=1}^{n} c_i e^{x_i}$$

and

$$\mathcal{K} \triangleq \text{column space of } (a_{ij}).$$

The advantages of studying this problem $\mathcal{A}(0)$ rather than its signomial predecessor come mainly from the fact that, unlike the signomial P, the exponential

function g is completely separable (in that it is the sum of terms, each of which depends on only a single independent variable x_i). Notice also that the parameter u_i is a logarithmic perturbation of the $|c_i|$, generally a very useful type of perturbation to consider because the coefficients c_i are typically costs per unit quantity of materiel and hence tend to vary somewhat. The exponents a_{ij} are usually geometrical constants or are fixed by the laws of nature and/or economics; so they do not tend to vary, and hence there is little lost in not studying their perturbations.

Example 2. Our second example comes from the minimization of quadratic functions

$$Q(z) = (1/2) \langle z, Hz \rangle + \langle h, z \rangle ,$$

where H is an arbitrary constant matrix and h is an arbitrary constant vector.

A factorization of the coefficient matrix H (which is of course associated with quadratic nonlinearities) is the key to effectively applying geometric programming to quadratic programming. More specifically, linear algebra is used to compute matrices D and $\mathcal{E}$ such that

$$H = D^t D - \mathcal{E}^t \mathcal{E} ,$$

where ^{t} indicates the transpose operation. In terms of D and $\mathcal{E}$ the quadratic function

$$Q(z) = (1/2)(\langle Dz, Dz \rangle - \langle \mathcal{E}z, \mathcal{E}z \rangle) + \langle h, z \rangle .$$

Of course, the expression $\langle Dz, Dz \rangle$ is not present when $Q(z)$ is negative semidefinite; and the expression $-\langle \mathcal{E}z, \mathcal{E}z \rangle$ is not present when $Q(z)$ is positive semidefinite (i.e. a convex function).

From elementary linear algebra we now infer that minimizing $Q(z)$ is equivalent to solving problem $\mathcal{G}(0)$ when

$$g(x) \triangleq (1/2)\left(\sum_{i=1}^{m} x_i^2 - \sum_{i=m+1}^{2m} x_i^2\right) + x_{2m+1}$$

and

$$\mathcal{Y} \triangleq \text{column space of } \begin{bmatrix} D \\ \mathcal{E} \\ h \end{bmatrix} .$$

Notice that, unlike the quadratic function Q, the quadratic function g is completely separable, a fact that can be exploited both theoretically and computationally.

It is useful to introduce some additional parameters into the preceding function g so that a much broader class of optimization problems can be studied. In particular, we redefine g so that

$$g(x) \triangleq \sum_{i=1}^{m} p_i^{-1} |x_i - b_i|^{p_i} - \sum_{i=m+1}^{2m} p_i^{-1} |x_i - b_i|^{p_i} + x_{2m+1} - b_{2m+1}$$

where b_i and p_i are arbitrary constants. Notice that the function g is still completely separable and can be specialized to the quadratic case by choosing $b_i = 0$ and $p_i = 2$ for each i.

Another interesting specialization is obtained by choosing $p_i = p$ for each i, while choosing $\mathcal{E} = 0$ and $h = 0$. The resulting problem consists essentially of finding the "best ℓ_p-norm approximation" to the fixed vector $(b_1, \ldots, b_m)$ by vectors in the column space of the matrix D, a fundamental problem in linear regression analysis. Finally, it is probably worth noting that the parameter u is then a perturbation of the vector $(b_1, \ldots, b_m)$ of data being optimally approximated, a rather useful type of perturbation to consider.

Other important examples that can be effectively treated by geometric pro-

gramming (but have usually been studied without explicitly mentioning the subject) are: generalized "Weber problems" that are of central importance in location theory, discrete optimal control problems with linear dynamics (sometimes called dynamic programming problems with linear transition functions), and various nonlinear network flow problems that arise in the analysis of electrical networks and the analysis of certain (multicommodity) transportation networks.

For those examples the functions $\varphi:\mathcal{U}$ and $S^*:\mathcal{U}$ show the dependence of optimality on the following external influences: existing facility locations in location theory, the control and state sets as well as the initial state and final target sets in control theory, the input currents and the input potential differences in electrical network theory, and the travel demands in transportation network theory.

For a more thorough discussion of such examples, see the recent expository paper by Peterson (1973), and the references cited therein.

In concluding our present discussion of important examples it is probably worth mentioning that the modeling of certain transportation networks that contain at least a single one-way automobile artery gives rise to cones $\mathcal{X}$ that are generally not subspaces. Consequently, the added generality built into this presentation of geometric programming is not without practical significance.

3. TWO DECOMPOSABLE PROBLEM TYPES

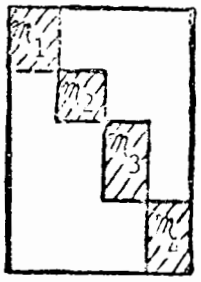
In all geometric programming problems $\mathcal{A}(0)$ known to the author to be of practical significance the cone $\mathcal{X}$ is polyhedral and hence finitely generated. We therefore suppose without any known loss of practical significance that there is at least one $n\times m$ matrix $\mathcal{M}$ with a corresponding index set $\mathcal{P} \subseteq \{1,2,\ldots,m\}$ for which

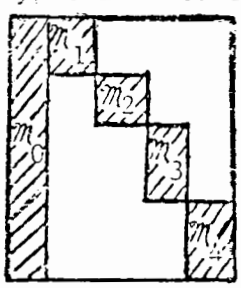
$$\mathcal{X} = \{\, x \in E_n \mid x = \mathcal{M}z \text{ for at least one } z \in E_m \text{ for which } z_j \geq 0,\ j \in \mathcal{P} \,\}.$$

The index set $\mathcal{P}$ can of course be taken to be the empty set when $\mathcal{X}$ is in fact a subspace of E_n.

We also suppose that it has been possible to choose the matrix representation $\mathcal{M}$ and its corresponding index set $\mathcal{P}$ so that $\mathcal{M}$ is sparse with a block diagonal structure that is one of the two types illustrated in Figure 1.

Type 1





Type 2

Figure 1. Two Types of block diagonal structure for $\mathcal{M}$.

The enumerated submatrices $\mathcal{M}_k$ are of course the only submatrices of $\mathcal{M}$ that need not be zero matrices. We assume in general that r such submatrices $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_r$ with $r \geq 2$ are arranged diagonally. (In particular, r is 4 for both of the examples illustrated in Figure 1.) Matrices $\mathcal{M}$ of type 1 can then be characterized as those that have no additional nonzero submatrices, and matrices $\mathcal{M}$ of type 2 are those that have a single additional nonzero submatrix $\mathcal{M}_0$ consisting of entire columns of $\mathcal{M}$.

Both types of block diagonal structure induce a partitioning of the rows of $\mathcal{M}$ and hence the components of x in such a way that

$$x = (x^1, x^2, \ldots, x^r),$$

where the components of the vector variable x^k are enumerated exactly the same as those rows of $\mathcal{M}$ that contain rows of the submatrix $\mathcal{M}_k$.

Similarly, each of the two types of block diagonal structure induces a partitioning of the columns of $\mathcal{M}$ and hence the components of z in such a way that

$$z = (z^1, z^2, \ldots, z^r) \text{ for problems of type 1}$$

while

$$z = (z^0, z^1, z^2, \ldots, z^r) \text{ for problems of type 2,}$$

where the components of the vector variable z^k are enumerated exactly the same as those columns of $\mathcal{M}$ that contain columns of submatrix $\mathcal{M}_k$. The components of the additional vector variable z^0 in a given problem of type 2 are of course the "coupling variables" that must be contended with in reducing such a problem to an equivalent problem of type 1.

To render both problem types amenable to decomposition, we further suppose that the function $g:\mathcal{C}$ is at least partially separable and that its partial separability is compatible with the preceding partitioning of the components of x. In particular then, we assume that there are functions $g_k:\mathcal{C}_k$, $k = 1,2,\ldots,r$, such that

$$\mathcal{C} = \sum_{k=1}^{r} \mathcal{C}_k \qquad \text{and} \qquad g(x) = \sum_{k=1}^{r} g_k(x^k).$$

This assumption is of course automatically satisfied when $g:\mathcal{C}$ is completely separable, a condition that holds for examples 1 and 2 in secton 2.

It is important to note that the preceding assumptions about problem $\mathcal{A}(0)$ are inherited by all problems $\mathcal{A}(u)$ in the geometric programming family $\mathcal{A}$. In particular, the cone Υ and hence the block diagonal structure of its matrix representation $\mathcal{M}$ remain invariant of u; and the separability of the function $g:\mathcal{C}$ is clearly inherited by all functions $g(\cdot + u) : (\mathcal{C} - u)$. Consequently, the decomposition principles to be developed in the context of treating problem $\mathcal{A}(0)$ are just as applicable to all other problems $\mathcal{A}(u)$ in the family $\mathcal{A}$.

In treating such problems $\mathcal{A}(u)$ we shall need to partition the components of u in the same way that the components of x have been partitioned, namely,

$$u = (u^1, u^2, \ldots, u^r),$$

where the components of the vector variable u^k are enumerated exactly the same as those rows of $\mathcal{M}$ that contain rows of the submatrix $\mathcal{M}_k$.

4. DECOMPOSITION BY TEARING

The decomposition principles to be developed here utilize the cones

$$\Upsilon_k \triangleq \{x^k \in E_{n_k} \mid x^k = \mathcal{M}_k z^k \text{ for at least one } z^k \in E_{m_k} \text{ for which } z^k_j \geq 0,\ j \in \mathcal{J}\}.$$

There are of course r such cones Υ_k, $k = 1,2,\ldots,r$ for a problem of type 1, and $r + 1$ such cones Υ_k, $k = 0,1,2,\ldots,r$ for a problem of type 2. Only the extra (coupling) cone Υ_0 for a problem of type 2 is a subcone of Υ.

Problems of type 1. We begin by observing that the cone Υ is separable, in that

$$x \in \Upsilon \text{ if and only if } x^k \in \Upsilon_k,\ k = 1,2,\ldots,r.$$

This separation of the cone Υ into the direct sum of the cones Υ_k, $k = 1,2,\ldots,r$, and the separation of the function $g:\mathcal{C}$ into a sum of the functions $g_k:\mathcal{C}_k$, $k = 1,2,\ldots,r$, immediately imply that problem $\mathcal{A}(0)$, now designated problem $\mathcal{A}^1(0)$, can be solved by solving the smaller geometric programming problems $\mathcal{A}_k(0)$ that are constructed from the respective functions $g_k:\mathcal{C}_k$ and the respective cones Υ_k, $k = 1,2,\ldots r$. In particular, the (desired) infimum $\varphi^1(0)$ for

problem $\mathcal{A}^1(0)$ can clearly be determined from the infima $\varphi_k(0)$ for the respective problems $\mathcal{A}_k(0)$, $k = 1,2,\ldots,r$ by the formula

$$\varphi^1(0) = \sum_{k=1}^{r} \varphi_k(0).$$

Moreover, the (desired) optimal solution set $\mathcal{S}^{1*}(0)$ for problem $\mathcal{A}^1(0)$ can obviously be determined from the optimal solution sets $\mathcal{S}_k^*(0)$ for the respective problems $\mathcal{A}_k(0)$, $k = 1,2,\ldots,r$ by the formula

$$\mathcal{S}^{1*}(0) = \underset{k=1}{\overset{r}{\times}} \mathcal{S}_k^*(0).$$

This direct decomposition of problem $\mathcal{A}^1(0)$ into r smaller problems $\mathcal{A}_k(0)$, $k = 1,2,\ldots,r$ generally increases computational efficiency and can in fact be a necessity when $\mathcal{A}^1(0)$ is itself too large for computer storage.

Using the previously noted fact that the same decomposition principle can be applied to each problem $\mathcal{A}(u)$ in a given family $\mathcal{A}$, we infer that the (desired) functions $\varphi^1: \mathcal{U}^1$ and $\mathcal{S}^{1*}: \mathcal{U}^1$ are determined by the corresponding functions $\varphi_k: \mathcal{U}_k$ and $\mathcal{S}_k^*: \mathcal{U}_k$ that are associated with the geometric programming families $\mathcal{A}_k$ (that are of course constructed from the respective functions $g_k: \mathcal{C}_k$ and the respective cones $\mathcal{K}_k$, $k = 1,2,\ldots,r$). This direct decomposition of the family $\mathcal{A}^1$ into r smaller families $\mathcal{A}_k$, $k = 1,2,\ldots,r$ can be concisely described by the formulas

$$\mathcal{U}^1 = \underset{k=1}{\overset{r}{\times}} \mathcal{U}_k,$$

$$\varphi^1(u) = \sum_{k=1}^{r} \varphi_k(u^k),$$

and

$$\mathcal{S}^{1*}(u) = \underset{k=1}{\overset{r}{\times}} \mathcal{S}_k^*(u^k),$$

where

$$\mathcal{U}_k = \mathcal{C}_k - \mathcal{K}_k, \qquad k = 1,2,\ldots,r.$$

Consequently, the functions $\varphi^1: \mathcal{U}^1$ and $\mathcal{S}^{1*}: \mathcal{U}^1$ can be studied by studying the functions $\varphi_k: \mathcal{U}_k$ and $\mathcal{S}_k^*: \mathcal{U}_k$, $k = 1,2,\ldots,r$.

In particular, given that the (one-sided) directional derivative $D_{d^k} \varphi_k(u^k)$ of the function $\varphi_k: \mathcal{U}_k$ (at a point $u^k \in \mathcal{U}_k$ in the direction $d^k \in E_{n_k}$) exists for $k = 1,2,\ldots,r$, and given that both $u = (u^1,u^2,\ldots,u^r)$ and $d = (d^1,d^2,\ldots,d^r)$, the (one-sided) directional derivative $D_d \varphi^1(u)$ of the function $\varphi^1: \mathcal{U}^1$ (at the point $u \in \mathcal{U}^1$ in the direction $d \in E_n$) clearly exists and is given by the formula

$$D_d \varphi^1(u) = \sum_{k=1}^{r} D_{d^k} \varphi_k(u^k).$$

It is only on rare occasions that the functions $\varphi_k: \mathcal{U}_k$, $k = 1,2,\ldots,r$ can be obtained in terms of elementary formulas. Consequently, the directional derivatives $D_{d^k} \varphi_k(u^k)$, $k = 1,2,\ldots,r$ usually have to be determined by numerical differentiation or other numerical methods.

If a given function $\varphi_k: \mathcal{U}_k$ is convex, then the "convex analysis" of Fenchel (1949, 1951) and Rockafellar (1970) shows that for almost every $u^k \in \mathcal{U}_k$ the directional derivative $D_{d^k} \varphi_k(u^k)$ exists for all the "feasible directions" $d^k \in E_{n_k}$ (i.e. all the directions $d^k \in E_{n_k}$ for which $u^k + sd^k \in \mathcal{U}_k$ for sufficiently small $s > 0$). On the other hand, the paper by Peterson (1970) shows that $\varphi_k: \mathcal{U}_k$ is in fact convex if the function $g_k: \mathcal{C}_k$ is assumed to be convex, an assumption that seems much less restrictive when viewed in the light of some recent work of Duffin and Peterson (1973, 1972). In any event, the paper by Peterson (1970) also shows that this assumption and other relatively mild assumptions actually guarantee that

$$D_{d^k} \varphi_k(u^k) = \max_{y^k \in \mathcal{J}_k^*(0,u^k)} \langle d^k, y^k \rangle,$$

where $\mathcal{G}_k^*(0;u^k)$ is the optimal solution set for the "geometric dual" $\mathcal{G}_k(0;u^k)$ of problem $\mathcal{A}_k(u^k)$. Since the paper by Peterson (1970) also provides "extremality conditions" that frequently lead to a relatively simple computation of $\mathcal{G}_k^*(0;u^k)$ from the knowledge of only a single optimal solution $x^{k*} \in \mathcal{S}_k^*(u^k)$, and since $\mathcal{G}_k^*(0;u^k)$ is frequently polyhedral (and sometimes a singleton), the directional derivatives $D_d\varphi^1(u)$ can often be determined by the preceding two displayed formulas by little more than linear programming (and sometimes much less). This means of course that first-order methods can often be used in conjunction with the preceding decomposition principle to minimize $\varphi^1(u)$ over a given subset of U^1 - a fundamental technique to be employed in developing a decomposition principle for problems of type 2.

Problems of type 2. We begin by observing that a given problem $\mathcal{A}(0)$ of type 2, now designated problem $\mathcal{A}^2(0)$, reduces to a problem $\mathcal{A}^1(u)$ of type 1 when the coupling vector variable x^0 is (temporarily) fixed and u is chosen to be $\mathcal{M}_0 x^0$. Using very elementary arguments, we then infer that the (desired) infimum $\varphi^2(0)$ for problem $\mathcal{A}^2(0)$ can in fact be determined by the formula

$$\varphi^2(0) = \inf_{u \in \mathcal{I}_0 \cap U^1} \varphi^1(u).$$

The minimization problem that appears in this formula is obviously a geometric programming problem and is termed the master problem. Its objective function $\varphi^1: U^1$ has of course already been (partially) separated into a sum of infima functions $\varphi_k: U_k$, $k = 1,2,\ldots,r$ by the decomposition principle previously developed for problems of type 1. That decomposition principle is of course to be used here not only to calculate the functional values $\varphi^1(u)$ but also to calculate any directional derivatives $D_d\varphi^1(u)$ that are needed to implement an appropriate algorithm for solving the master problem. Once the master problem's optimal solution set

$$U^* \triangleq \{u \in \mathcal{I}_0 \cap U^1 \mid \varphi^1(u) = \varphi^2(0)\}$$

has been obtained, the (desired) optimal solution set $\mathcal{S}^{2*}(0)$ for problem $\mathcal{A}^2(0)$ can clearly be determined by the formula

$$\mathcal{S}^{2*}(0) = \bigcup_{u^* \in U^*} \{u^* + \sum_{k=1}^{r} \mathcal{S}_k^*(u^{*k})\}.$$

In the process of determining a $u^* \in U^*$ with the aid of this tearing procedure, one can of course expect to determine an $x^{k*} \in \mathcal{S}_k^*(u^{*k})$, $k = 1,2,\ldots,r$; in which event $u^* + (x^{1*},x^{2*},\ldots,x^{r*})$ is one of the desired optimal solutions to problem $\mathcal{A}^2(0)$. This assumes of course that such optimal solutions exist.

In summary, problem $\mathcal{A}^2(0)$ has been torn into r smaller problems $\mathcal{A}_k(u^k)$, $k = 1,2,\ldots,r$ that are judiciously selected by the master problem, with the possible help of geometric programming duality in the convex case. This tearing may increase computational efficiency but can in fact be a necessity when $\mathcal{A}^2(0)$ is itself too large for computer storage.

As previously noted, the same decomposition principle can be applied to each problem $\mathcal{A}(u)$ in a given family $\mathcal{A}$. The obvious result of such an application here is notationally somewhat cumbersome to describe and is left to the imagination of the interested reader.

Although developed in the context of problems without explicit constraints, the preceding decomposition principles apply equally well to problems with explicit constraints - by virtue of the relations to be given in the next section.

5. PROBLEMS WITH EXPLICIT CONSTRAINTS

To keep track of the explicit constraints, we introduce two nonintersecting (possibly empty) positive-integer index sets I and J with finite cardinality o(I) and o(J) respectively. We also introduce the following notation and hypotheses:

(1) For each $k \in \{0\} \cup I \cup J$ suppose that g_k is a function with a nonempty

domain $C_k \subset E_{nk}$, and for each $j \in J$ suppose that D_j is a nonempty subset of E_{nj}.

(2) For each $k \in \{0\} \cup I \cup J$ let u^k be an independent vector parameter in E_{nk}, and let μ be an independent vector parameter with components μ_i for each $i \in I$.

(3) Denote the cartesian product of the vector parameters u^i, $i \in I$, by the symbol u^I, and denote the cartesian product of the vector parameters u^j, $j \in J$, by the symbol u^J. Then the cartesian product $u \triangleq (u^0, u^I, u^J)$ of the vector parameters, u^0, u^I, and u^J is an independent vector parameter in E_n, where

$$n \triangleq n_0 + \sum_I n_i + \sum_J n_j .$$

(4) For each $k \in \{0\} \cup I \cup J$ let x^k be an independent vector variable in E_{nk}, and let $\varkappa$ be an independent vector variable with components $\varkappa_j$ for each $j \in J$.

(5) Denote the cartesian product of the vector variables x^i, $i \in I$, by the symbol x^I, and denote the cartesian product of the vector variables x^j, $j \in J$, by the symbol x^J. Then the cartesian product $x \triangleq (x^0, x^I, x^J)$ of the vector variables x^0, x^I, and x^J is an independent vector variable in E_n.

(6) Suppose that X is a cone in E_n.

Now, consider the following geometric programming family A of geometric programming problems $A(u,\mu)$ that were first investigated by Peterson (1973, 1974).

PROBLEM $A(u,\mu)$. Consider the objective function $G(\cdot + u, \varkappa)$ whose domain

$$C(u) \triangleq \{(x,\varkappa) \mid x^k + u^k \in C_k,\ k \in \{0\} \cup I, \text{ and } (x^j + u^j, \varkappa_j) \in C_j^+,\ j \in J\},$$

and whose functional value

$$G(x + u, \varkappa) \triangleq g_0(x^0 + u^0) + \sum_J g_j^+(x^j + u^j, \varkappa_j),$$

where

$$C_j^+ \triangleq \{(z^j, \varkappa_j) \mid \text{either } \varkappa_j = 0 \text{ and } \sup_{d^j \in D_j} \langle z^j, d^j \rangle < +\infty, \text{ or } \varkappa_j > 0 \text{ and } z^j \in \varkappa_j C_j\}$$

and

$$g_j^+(z^j, \varkappa_j) \triangleq \begin{cases} \sup_{d^j \in D_j} \langle z^j, d^j \rangle & \text{if } \varkappa_j = 0 \text{ and } \sup_{d^j \in D_j} \langle z^j, d^j \rangle < +\infty \\ \varkappa_j g_j(z^j/\varkappa_j) & \text{if } \varkappa_j > 0 \text{ and } z^j \in \varkappa_j C_j. \end{cases}$$

Using the feasible solution set

$$S(u,\mu) \triangleq \{(x,\varkappa) \in C(u) \mid x \in X, \text{ and } g_i(x^i + u^i) + \mu_i \le 0,\ i \in I\},$$

calculate both the problem infimum

$$\varphi(u,\mu) \triangleq \inf_{(x,\varkappa) \in S(u,\mu)} G(x + u, \varkappa)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{(x,\varkappa) \in S(u,\mu) \mid G(x + u, \varkappa) = \varphi(u,\mu)\}.$$

In defining the feasible solution set $S(u,\mu)$, it is important to make a sharp distinction between the cone condition $x \in X$ and the constraints $g_i(x^i + u^i) + \mu_i \le 0$, $i \in I$, both of which restrict the vector variable $(x,\varkappa)$. The cone X is frequently finitely generated, and hence the cone condition can frequently be eliminated by a linear transformation of the vector variable x; but the (generally nonlinear) constraints usually can not be eliminated by even a nonlinear transformation. Nevertheless, we never explicitly eliminate the cone condition because such a linear transformation would introduce a common vector variable into the arguments of g_0, g_i, and g_j^+. Such a common vector variable would of course only tend to camouflage the separability that is built into problem $A(u,\mu)$.

Analogous to the unconstrained case introduced in section 2, we shall find it useful to interpret problem A(u,v) as a perturbed version of A(0,0), so we term the set

$$U \triangleq \{(u,v) \mid \text{problem } A(u,v) \text{ is consistent}\}$$

the feasible perturbation set for problem A(0,0). It is important to note that there are no perturbations associated with the variables x_j; the reason is that such perturbations would clearly have no effect. Note also that the special perturbations v_i appear only with the constraints and hence have no counterpart in the unconstrained case. The unconstrained case can of course be obtained from the present constrained case by simply letting both index sets I and J be the empty set.

The extreme flexibility of the present geometric programming formulation is clearly illustrated by the following examples of important problem classes.

Example 0. Linear programming can be viewed as a special case of geometric programming in various ways. Probably the most direct way is to let J be the empty set and choose the functions $g_k: C_k$, $k \in \{0\} \cup I$ by letting

$$C_0 \triangleq E_1 \text{ with } g_0(x^0) \triangleq x^0,$$

and

$$C_i \triangleq E_1 \text{ with } g_i(x^i) \triangleq x^i - b_i, \quad i \in I,$$

where the numbers b_i are components of a given vector b. In addition, choose the cone X by letting

$$X \triangleq \{(x^0, x^I) \in E_n \mid x^0 = \langle a,z \rangle \text{ and } x^I = Mz \text{ for at least one } z \in E_m \text{ for which } z_j \geq 0, \ j \in P\},$$

where a is a given vector, M is a given matrix, and P is a given subset of $\{1,2,\ldots,m\}$.

Problem A(0,0) then clearly consists of minimizing the linear function $\langle a,z \rangle$ subject to both the linear constraints $Mz \leq b$ and the nonegativity conditions $z_j \geq 0$, $j \in P$. This is of course the most general linear programming problem, and it is worth noting that while the effect of the parameter u^0 is obvious the parameters u^i and v_i both perturb the constraint upper bound b_i.

Other ways of viewing linear programming as a special case of geometric programming are given in the expository paper by Peterson (1973).

Example 0+. To view ordinary mathematical programming as a special case of geometric programming, let I be the set $\{1,2,\ldots,p\}$ and let J be the empty set. Then, choose the cone X to be the column space of a specially structured matrix, namely,

$$X \triangleq \text{column space of } \begin{bmatrix} \mathcal{I} \\ \mathcal{I} \\ \vdots \\ \mathcal{I} \end{bmatrix}$$

where the m × m identity matrix $\mathcal{I}$ appears in a total of 1 + p positions.

Problem A(0,0) then clearly consists of minimizing $g_0(z)$ subject to both the constraints $g_i(z) \leq 0$, $i = 1,2,\ldots,p$ and the domain condition $z \in \bigcap_{k=0}^{p} C_k$. It is worth noting that the vector parameter u^k perturbs the domain C_k by translation while the parameter v_k perturbs the constraint upper bound 0. The subfamily of A that results from choosing the vector parameters $u^k = 0$ has been termed the "ordinary mathematical programming" family by Rockafellar (1970), because its problem A(0,0) and its vector parameter u have been the focus of most of the past work in mathematical programming.

Actually, there may be advantages in reformulating the ordinary mathematical programming problem as the following geometric programming problem.

$$\text{Minimize } g_0(x^0) \text{ subject to}$$

$$g_1(x^1) \le 0, \qquad g_2(x^2) \le 0, \quad \ldots, \quad g_p(x^p) \le 0$$

$$x^0 - x^1 = 0$$

$$x^1 - x^2 = 0$$

$$\cdots$$

$$x^{p-1} - x^p = 0$$

$$x^0 \in C_0, \qquad x^1 \in C_1, \quad \ldots, \quad x^p \in C_p,$$

In this geometric programming formulation the ordinary programming problem is at least partially separable.

In any event, most important optimization problems have much more structure than that which is present in the ordinary programming problem. Moreover, that extra structure can frequently be exploited by choosing the cone X differently from the way it has been chosen in this example. This fact is clearly illustrated by the following extensions of examples 1 and 2 from section 2.

Example 1. To treat the minimization of signomials subject to signomial inequality constraints, let I be the set $\{1,2,\ldots,p\}$ and let J be the empty set. Then, choose the functions $g_k : C_k$, $k = 0,1,2,\ldots,p$ by letting

$$C_k \triangleq E_{n_k} \text{ with } g_k(x^k) \triangleq \sum_{[k]} c_i e^{x_i} - b_k,$$

where the numbers c_i and b_k are given constants and the index set

$$[k] \triangleq \{m_k, m_k + 1, \ldots, n_k\},$$

with the understanding that

$$1 \triangleq m_0 \le n_0, \qquad n_0 + 1 \triangleq m_1 \le n_1, \qquad \ldots, \qquad n_{p-1} + 1 \triangleq m_p \le n_p = n.$$

Finally, choose the cone X by letting

$$X \triangleq \text{column space of } (a_{ij}),$$

where (a_{ij}) is an arbitrary $n \times m$ real matrix.

Making the change of variables

$$x_i = \sum_{j=1}^{m} a_{ij} \log t_j \text{ where } t_j > 0, \; j = 1,2,\ldots,m \text{ and } i = 1,2,\ldots,n,$$

we easily infer that problem A(0,0) consists essentially of minimizing the signomial $\sum_{[0]} c_i \prod_{j=1}^{m} t_j^{a_{ij}}$ subject to the signomial inequality constraints $\sum_{[k]} c_i \prod_{j=1}^{m} t_j^{a_{ij}} \le b_k$, $k = 1,2,\ldots,p$ (and of course the conditions $t_j > 0$, $j = 1,2,\ldots,m$). It is worth noting that the parameter u_i perturbs the coefficient c_i (actually, the logarithm of the $|c_i|$) while the parameter u_k perturbs the constraint upper bound b_k.

Of course, the main reason for studying the preceding geometric programming formulation of signomial optimization is the complete separability that is introduced. For a more thorough discussion of this topic see the recent papers by Duffin and Peterson (1973, 1972), and the references cited therein. Parts of those papers show that all "algebraic programming" problems can actually be reduced to the "posynomial" case in which all coefficients c_i are positive and each posynomial has at most two terms. To avoid possible confusion, it should also be mentioned that in those papers, as well as the predecessors by Duffin and Peterson (1966) and Duffin, Peterson, and Zener (1967), it has been advantageous

to replace the separable functions $\sum_{[k]} c_i e^{x_i}$ with the (technically) inseparable functions $\log \left(\sum_{[k]} c_i e^{x_i} \right)$ for the study of geometric programming duality.

Example 2. To treat both quadratic programming with quadratic constraints and ℓ_p - regression problems with ℓ_p - norm constraints, let I be the set $\{1,2,\ldots,r\}$ and let J be the empty set. Then, choose the functions $g_k : C_k$, $k = 0,1,2,\ldots,r$ by letting

$$C_k \triangleq F_{n_k} \text{ with } g_k(x^k) \triangleq \sum_{[k]^+} \frac{1}{p_i} \mid x_i - b_i \mid^{p_i} - \sum_{[k]^-} \frac{1}{p_i} \mid x_i - b_i \mid^{p_i} + x_{]k[} - b_{]k[},$$

where the numbers b_i, $b_{]k[}$, and p_i are given constants and the index sets are

$$[k]^+ \triangleq \{m_k^+, m_k^+ + 1, \ldots, n_k^+\}; \; [k]^- \triangleq \{m_k^-, m_k^- + 1, \ldots, n_k^-\}; \text{ and }]k[\triangleq n_k,$$

with the understanding that

$$1 \triangleq m_0^+ \le n_0^+, \quad n_0^+ + 1 \triangleq m_0^- \le n_0^-, \quad n_0^- + 1 \triangleq n_0, \quad n_0 + 1 \triangleq m_1^+ \le n_1^+, \ldots, n_r^- + 1 \triangleq n.$$

Finally, choose the cone X by letting

$$X \triangleq \text{column space of } \begin{bmatrix} D_0 \\ \mathcal{D}_0 \\ h_0 \\ \cdot \\ \cdot \\ \cdot \\ D_r \\ \mathcal{D}_r \\ h_r \end{bmatrix}$$

where D_k is an $(n_k^+ - n_{k-1}) \times m$ real matrix, $\mathcal{D}_k$ is an $(n_k^- - n_k^+) \times m$ real matrix, and h_k is a $1 \times m$ real matrix.

Making the usual change of variables

$$x = Mz$$

where M denotes the preceding partitioned matrix, we obtain quadratic programming and ℓ_p - regression analysis with the following two specializations.

To obtain all quadratic programming problems, let $b_i = 0$ and $p_i = 2$ for $i \in [k]^+ \cup [k]^-$. Then, problem A(0,0) consists essentially of minimizing the quadratic function $(1/2) \langle z, H_0 z \rangle + \langle h_0, z \rangle - b_{]0[}$ subject to the quadratic constraints $(1/2) \langle z, H_k z \rangle + \langle h_k, z \rangle \le b_{]k[}$, $k = 1,2,\ldots,r$, where $H_k \triangleq D_k^t D_k - \mathcal{D}_k^t \mathcal{D}_k$. Although the effect of the parameter u_i is rather uninteresting and complicated, the parameter u_k perturbs the constraint upper bound $b_{]k[}$.

To obtain all ℓ_p - regression problems, take $p_i = p^k$ for $i \in [k]^+ \cup [k]^-$; let $b_{]0[} = 0$ and $b_i = 0$ for $i \in [k]^-$; and choose $\mathcal{D}_k = 0$ and $h_k = 0$. Then, problem A(0,0) consists essentially of minimizing the ℓ_{p^0} - norm function $(1/p^0)(\| D_0 z - b^k \|_{p^0})^{p^0}$ subject to the ℓ_{p^k} - norm constraints $(1/p^k)(\| D_k z - b^k \|_{p^k})^{p^k} \le b_{]k[}$, $k = 1,2,\ldots,r$, where $\| \cdot \|_{p^k}$ denotes the ℓ_{p^k} - norm and $b^k \triangleq (b_{m_k^+}, \ldots, b_{n_k^+})$. While the vector parameter $u^k \triangleq (u_{m_k^+}, \ldots, u_{n_k^+})$ perturbs the vector b^k, both the parameters $u_{]k[}$ and μ_k perturb the constraint upper bound $b_{]k[}$.

Of course, the main reason for studying the preceding geometric programming formulation of both quadratic programming and ℓ_p - regression analysis is the complete separability that is introduced. For a more thorough discussion of this topic in the convex case (in which $\mathcal{D}_k = 0$ for $k = 0,1,2,\ldots,r$) see the papers by Peterson and Ecker (1968, 1970, 1969, 1970).

Other important examples that can be effectively treated by geometric programming (but have usually been studied without explicitly mentioning the subject) are discussed in the recent expository paper by Peterson (1973), and the references cited therein. In one of those examples, the "generalized chemical equilibrium problem", the index set J is not the empty set. That example and an alternative way of viewing linear programming as a special case of geometric programming are at this time the only practical justifications for including the index set J (and hence its resulting complications) in the present formulation of geometric programming. However, "geometric programming duality" as developed by Peterson (1973, 1974) provides an aesthetic justification for doing so.

We now show how geometric programming with explicit constraints can be viewed as a special case of geometric programming without explicit constraints. In doing so, we see how to apply the decomposition principles developed in section 4 to the problems discussed in this section.

Introducing an additional independent vector variable α with components α_i for each $i \in I$, we let the functional domain

$$\mathcal{C} \triangleq \{(x^0, x^I, \alpha, x^J, \varkappa) \in E_\eta \mid x^0 \in C_0;\ x^i \in C_i, \alpha_i \in E_1, \text{ and } g_i(x^i) + \alpha_i \le 0,\ i \in I;\ (x^j, \varkappa_j) \in C_j^+,\ j \in J\};$$

and we let the functional value

$$\mathcal{g}(x^0, x^I, \alpha, x^J, \varkappa) \triangleq g_0(x^0) + \sum_J g_j^+(x^j, \varkappa_j) \triangleq G(x, \varkappa).$$

We also let the cone

$$\mathcal{X} \triangleq \{(x^0, x^I, \alpha, x^J, \varkappa) \in E_\eta \mid (x^0, x^I, x^J) \in X;\ \alpha = 0;\ \varkappa \in E_{o(J)}\}.$$

Then, problem A(0,0) is clearly identical to problem $\mathcal{A}(0)$. In fact, it is easy to see that problem $A(u,\mu)$ is identical to problem $\mathcal{A}(u)$, with the parameter μ_i being identified with the parameter u_i that corresponds to α_i. Of course, the parameter u_j that corresponds to $\varkappa_j$ has no effect and can be set equal to zero.

It is important to realize that the components of $\varkappa = (x^0, x^I, \alpha, x^J, \varkappa)$ can be placed in any order to achieve a block diagonal structure for some matrix representation $\mathcal{M}$ of $\mathcal{X}$. Moreover, it is clear from the definition of $\mathcal{X}$ that the possibility of achieving such a block diagonal structure for some $\mathcal{M}$ depends entirely on the possibility of ordering the components of $x = (x^0, x^I, x^J)$ in such a way that a block diagonal structure is achieved for some matrix representation M of X.

It is equally important to realize that the function $\mathcal{g}:\mathcal{C}$ inherits any separability that is present in the functions $g_0:C_0$ and $g_j^+:C_j^+$. Unfortunately, $\mathcal{g}:\mathcal{C}$ clearly does not generally inherit any of the separability that is present in the functions $g_i:C_i$; unless there are corresponding Kuhn-Tucker (Lagrange) multipliers λ_i, in which case the constraints $g_i(x^i) + \alpha_i \le 0$, $i \in I$ in the defining equation for $\mathcal{C}$ can be deleted, provided that the expression $\sum_I \lambda_i [g_i(x^i) + \alpha_i]$ is added to the defining equation for $\mathcal{g}(x^0, x^I, \alpha, x^J, \varkappa)$. If desired, this manipulation can of course be limited to only some of the constraints for which there are corresponding Kuhn-Tucker multipliers.

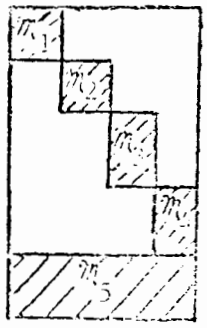
Other ramifications of the preceding relation between constrained and unconstrained geometric programming are given in the recent work of Peterson (1974).

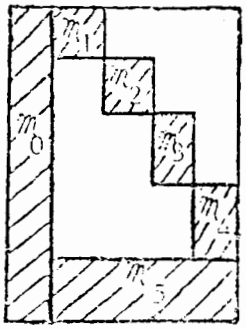
6. CONCLUDING REMARKS

Decomposition principles can of course be developed for geometric programming problems $\mathcal{A}(0)$ that possess matrix representations $\mathcal{M}$ of $\mathcal{X}$ with block structures that are not one of the two types treated here. In fact, a future paper

will be devoted to developing decomposition principles for those problems $\mathcal{A}(0)$ for which $\mathcal{M}$ has one of the two additional types of block diagonal structure illustrated in Figure 2.

Type 3





Type 4

Figure 2. Two additional types of block diagonal structure for $\mathcal{M}$.

Geometric programming duality plays a much more fundamental role in the treatment of problems $\mathcal{A}(0)$ of type 3 than it has played here in the treatment of problems $\mathcal{A}(0)$ of types 1 and 2. Problems $\mathcal{A}(0)$ of type 4 are of course a combination of problems $\mathcal{A}(0)$ of types 2 and 3. In fact, problems $\mathcal{A}(0)$ of type 4 can be treated by combining the decomposition principles that have been developed here for problems $\mathcal{A}(0)$ of type 2 with those to be developed elsewhere for problems $\mathcal{A}(0)$ of type 3. The order in which the principles are to be combined is indicated by the partitioning of the matrix $\mathcal{M}$ between the submatrices $\mathcal{M}_0$ and $\mathcal{M}_5$ in illustration 2 of Figure 2.

REFERENCES

Dantzig, G.B., (1970), Large scale systems and the computer revolution, in Proc. Princeton Symp. Math. Prog., (1967) ed. H.W. Kuhn, Princeton University Press, p. 51.

Duffin, R.J., (1962), Dual programs and minimum cost, SIAM Jour. Appl. Math., 10, p. 119.

Duffin, R.J., (1962), Cost minimization problems treated by geometric means, ORSA Jour., 10, p. 668.

Duffin, R.J., and E.L. Peterson, (1966), Duality theory for geometric programming, SIAM Jour. Appl. Math., 14, p. 1307.

Duffin, R.J., and E.L. Peterson, (1973), Geometric programming with signomials, Jour. Opt. Th. Appls., 11, in press.

Duffin, R.J., and E.L. Peterson, (1972), Reversed geometric programs treated by harmonic means, Indiana University Math. Jour., to appear.

Duffin, R.J., and E.L. Peterson, (1972), The proximity of (algebraic) geometric programming to linear programming, Math. Prog., to appear.

Duffin, R.J., E.L. Peterson, and C. Zener, (1967), Geometric Programming, Wiley, New York.

Ecker, J.G., (1972), Decomposition in separable geometric programming, Jour. Opt. Th. Appls., 9, p. 176.

Fenchel, W., (1949), On conjugate convex functions, Canadian Jour. Math., 1, p. 73.

Fenchel, W., (1951), Convex Cones, Sets, and Functions, Math. Dept. mineographed lecture notes, Princeton University.

Gass, S.I., (1966), The dualplex method for large-scale linear programs, Operations Research Center, 1966-15, University of California, Berkeley.

Heymann, M., and M. Avriel, (1969), On a decomposition for a special class of geometric programming problems, Jour. Opt. Th. Appls., 3, p.

Kron, G., (1953), A set of principles to interconnect the solutions of physical systems, Jour. Appl. Phys., 24, p. 965.

Kron, G., (1963), Diakoptics: the Piecewise Solution of Large-Scale Systems, MacDonald, London.

Peterson, E.L., (1970), Symmetric duality for generalized unconstrained geometric programming, SIAM Jour. Appl. Math., 19, p. 487.

Peterson, E.L., (1973), Geometric programming and some of its extensions, in Perspectives on Optimization, ed. M. Avriel, M. Rijckaert, and D. Wilde, Prentice-Hall, to appear.

Peterson, E.L., (1974), Generalization and symmetrization of duality in geometric programming, SIAM Jour. Appl. Math., to be submitted.

Peterson, E.L., and J.G. Ecker, (1968), Geometric programming: a unified duality theory for quadratically constrained quadratic programs and ℓ_p - constrained ℓ_p - approximation problems, Bull. Am. Math. Soc., 74, p. 316.

Peterson, E.L., and J.G. Ecker, (1970), Geometric programming: duality in quadratic programming and ℓ_p - approximation I, in Proc. Princeton Symp. Math. Prog. (1967), ed. H.W. Kuhn, Princeton University Press, p. 445.

Peterson, E.L., and J.G. Ecker, (1969), Geometric programming: duality in quadratic programming and ℓ_p - approximation II (canonical programs), SIAM Jour. Appl. Math., 17, p. 317.

Peterson, E.L., and J.G. Ecker, (1970), Geometric programming: duality in quadratic programming and ℓ_p - approximation III (degenerate programs), Jour. Math. Anal. Appls., 29, p. 365.

Rijckaert, M., (1973), Decomposition applied to dual geometric programming, in Proc. NATO Inst. on Decomposition (1972), ed. D.M. Himmelblau, North Holland Publ. Com., p.

Rockafellar, R.T., (1970), Convex Analysis, Princeton University Press.

Rosen, J.B., (1964), Primal partition programming for block diagonal matrices, Numerische Math., 6, p. 250.

Zener, C., (1961), A mathematical aid in optimizing engineering designs, Proc. Nat. Acad. Sci., 47, p. 537.

Zener, C., (1962), A further mathematical aid in optimizing engineering designs, Proc. Nat. Acad. Sci., 48, p. 518.

Zener, C., (1962), Minimization of system costs in terms of subsystem costs, Proc. Nat. Acad. Sci., 51, p. 162.

Zener, C., (1971), Engineering Design by Geometric Programming, Wiley, New York.