DISCUSSION PAPER NO. 36

THE SCOPE OF GEOMETRIC PROGRAMMING

by

ELMOR L. PETERSON

MARCH 1, 1973

THE SCOPE OF GEOMETRIC PROGRAMMING*

Elmor L. Peterson**

Abstract

Contrary to popular belief geometric programming is not just a special technique for studying the very important class of "posynomial programs". It is really a very general mathematical theory that is especially useful for studying a large class of "separable programs". Its practical efficacy is due mainly to the fact that many very important (seemingly nonseparable) programs can actually be formulated as (separable) geometric programs by fully exploiting their (linear) algebraic structure. Some examples are: nonlinear multicommodity network flow problems, dynamic programs with linear transition functions, facility location problems, (ℓ_p constrained) ℓ_p regression problems, (quadratically constrained) quadratic programs, and general algebraic programs. The theory of geometric programming includes (i) very strong existence, uniqueness, and characterization theorems, (ii) useful parametric and economic analyses, and (iii) powerful numerical solution techniques.

This paper is primarily expository in nature and includes only the simplest derivations and results that provide insight into the subject. No prior knowledge of geometric programming is assumed.

---

*Originally prepared for the Advanced NATO Summer School on the Impact of Optimization Theory on Technological Design, held in Louvain, Belgium, July, 1971. To be published in the form of a chapter "Geometric Programming and Some of Its Extensions" along with an appendix "The Mathematical Foundations of Geometric Programming" in Optimization and Design (Avriel, Rijckaert, and Wilde, eds.), Prentice-Hall, 1973.

**Department of Industrial Engineering/Management Sciences and Department of Mathematics, Northwestern University, Evanston, Illinois 60201.

TABLE OF CONTENTS

Section	Page
1. Introduction	1
2. Geometric Programming Families	2
3. Geometric Programming Dual Families	7
4. Duality Gaps and the Extremality Conditions	14
5. Three Economics Problems in Management Science	17
6. Duality with Explicit Constraints	22
7. Posynomial Programming Algorithms	42
8. Signomial Programs Treated by Geometric and Harmonic Means	45
9. Example of Arbitrary Algebraic Programs	53
References	56

1. INTRODUCTION

Geometric programming provides a systematic method for analyzing and optimizing a large class of technological designs. The original prototype theory treats the important class of optimization problems that consist of minimizing a "posynomial" (i.e., a generalized positive polynomial) subject to upper bound inequality posynomial constraints. In essence, this is accomplished by transforming such (nonconvex nonseparable) problems into equivalent convex separable problems and applying Cauchy's "arithmetic-geometric mean inequality." The resulting "geometric dual problem" is at least as important in the study of posynomial programming as the linear dual problem is important in the study of linear programming. In fact, the geometric dual problem has only linear constraints, provides lower bounds on the minimum value of the corresponding posynomial problem, and gives the dependence of the minimum value on the posynomial coefficients.

Analogous transformations and inequalities can be used to treat other important classes of optimization problems, such as chemical equilibrium problems, quadratically-constrained quadratic programs, ℓ_p-constrained ℓ_p-regression problems, facility location problems, dynamic programs with linear transition functions, and multicommodity network flow problems. The appropriate transformations are suggested by the (linear) algebraic structure of the given problem, and the appropriate inequalities are generated by taking the "conjugate transform" of the resulting functions.

In addition to indicating the potential scope of geometric programming and its potential impact on technological design, the discussion of its mathematical foundations given in Secs. 2 through 6 can serve either as an introduction to the subject or as an enlightening review of the prototype theory.

First-generation geometric programming algorithms have encountered certain numberical difficulties when slack constraints are present. Those difficulties have been at least partially overcome by the introduction of slack variables (in a manner that differs considerably from the corresponding approach in linear programming). The method is closely related to "penalty function methods" and also provides a somewhat simpler proof of the original "refined duality theory" of prototype geometric programming. This topic is discussed in Sec. 7.

Well-posed algebraic programs (namely, optimization problems involving only real-valued functions that are generated solely by addition, subtraction, multiplication, division, and the extraction of roots) can be transformed into equivalent posynomial programs. The resulting class of posynomial programs is substantially larger than the class of prototype geometric programs. However, much of the prototype theory can be generalized by studying the "equilibrium solutions" to the "reversed geometric programs" in this larger class. Moreover, the "arithmetic-harmonic mean inequality" can be used to "invert" the "reversed constraints" and hence reduce the study of each reversed geometric program to the study of a corresponding "robust" family of "conservatively approximating" prototype geometric programs. Hence, algebraic programs can be analyzed by the well-developed techniques of prototype geometric programming. This topic is discussed in Secs. 8 and 9.

2. GEOMETRIC PROGRAMMING FAMILIES

Classical optimization theory and 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 the subset S is required to be the intersect of C with an arbitrary vector subspace $X \subseteq E_n$. 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 A_1 of closely related minimization problems $A_1(u)$ 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 n is 2 and the dimension of X is 1.) The problem of minimizing g over $X \cap C$ appears in the family A_1 as problem $A_1(0)$ and is studied in relation to all other problems $A_1(u)$, with special attention given to those problems $A_1(u)$ that are close to $A_1(0)$ in the sense that (the norm of) u is small.

Each problem $A_1(u)$ is said to be a <u>geometric programming problem</u>, and the family A_1 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 $A_1(u)$ is now given the following formal definition in terms of classical terminology and notation.

Problem $A_1(u)$. Using the "feasible solution" set

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

calculate both the "problem infimum"

$$\phi_1(u) \triangleq \inf_{x \in S(u)} g(x + u)$$

and the "optimal solution" set

$$S^*(u) \triangleq \{x \in S(u) \mid g(x + u) = \phi_1(u)\}.$$

For a given u, problem $A_1(u)$ 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 A_1 contains infinitely many consistent problems $A_1(u)$. The domain of the infimum function ϕ_1 is taken to be the corresponding nonempty set U of all those vectors u for which $A_1(u)$ is consistent. Thus, the range of ϕ_1 may contain the point $-\infty$; but if $\phi_1(u) = -\infty$, 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 $A_1(0)$ in more than one way by suitably choosing the function g and the vector subspace X. For example, we 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 problems in technological design, because they involve a certain amount of linear analysis (due to the presence of matrices, linear equations, etc.) which can be conveniently handled through the introduction of an appropriate nontrivial vector space X. The presence of this vector space X is the distinguishing feature of the geometric programming point of view.

Due to the pre-eminence of problem $A_1(0)$, we shall find it useful to interpret problem $A_1(u)$ as a perturbed version of $A_1(0)$, so we term the set

$$U \triangleq \{u \in E_n \mid S(u) \text{ is not empty}\}$$

the feasible perturbation set for problem $A_1(0)$ relative to the family A_1. We shall soon see that the functions ϕ_1 and S^* usually show the dependence of an optimal design on actual external influences such as design requirements, materiel costs, and so forth.

Example 1. Perhaps the most striking example of the utility of the geometric programming point of view comes from using it to study the minimization of "generalized polynomials." This was first done by Zener (1961, 1962) and Duffin (1962a,b), and served as the initial development (as well as the main stimulus for subsequent developments) of geometric programming.

A generalized polynomial is any function with the form

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

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.

The presence of the "exponent matrix" (a_{ij}) is the key to applying geometric programming to generalized polynomial optimization. To effectively place the problem of minimizing $P(t)$ in the format of problem $A_1(0)$, simply make the change of variables

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

and then use the laws of exponents to infer that minimizing $P(t)$ is equivalent to solving problem $A_1(0)$ when

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

and $\qquad X \triangleq$ column space of (a_{ij}).

The advantages of studying this problem $A_1(0)$ rather than its polynomial predecessor come mainly from the fact that, unlike the function P, the function g is "separable" in that it is the sum of terms, each of which depends on only a single independent variable x_i. Notice also that u_i is a logarithmic perturbation of the coefficient c_i, 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. We shall discuss other aspects of polynomial optimization in later sections.

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

$$Q(z) = \tfrac{1}{2}\langle z, Mz\rangle + \langle h, z\rangle,$$

where M is an arbitrary constant matrix and h is an arbitrary constant vector. Using linear algebra, we can compute matrices H_1 and H_2 such that

$$M = H_1^t H_1 - H_2^t H_2,$$

where ^{t} indicates the transpose operation. In terms of H_1 and H_2 the quadratic function is

$$Q(z) = \tfrac{1}{2}(\langle H_1 z, H_1 z\rangle - \langle H_2 z, H_2 z\rangle) + \langle h, z\rangle.$$

Of course, the expression $\langle H_1 z, H_1 z\rangle$ is not present when $Q(z)$ is "negative semi-definite"; and the expression $-\langle H_2 z, H_2 z\rangle$ is not present when $Q(z)$ is "positive semi-definite" (i.e., a convex function).

From elementary linear algebra we now infer that minimizing $Q(z)$ is equivalent to solving problem $A_1(0)$ when

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

and $\qquad X \triangleq$ column space of $\begin{bmatrix} H_1 \\ H_2 \\ h \end{bmatrix}$.

Notice that, unlike the quadratic function Q, the quadratic function g is 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 each b_i is an arbitrary constant and each constant $p_i \geq 1$. Notice that the function g is still 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 H_2 to be the zero matrix and h the zero vector. 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 H_1, a fundamental problem in "regression analysis." It is worth noting that u is a perturbation of the vector $(b_1, \ldots, b_m)$ being approximated, a rather useful type of perturbation to consider. We shall discuss other aspects of "ℓ_p programming" in later sections.

Example 3. Our third example comes from the problem of optimally locating a new facility relative to existing facilities. We suppose that there are p existing facilities with fixed locations $b^1, b^2, \ldots, b^p$, and we assume that for each $k = 1, 2, \ldots, p$ there is a cost $d_k(z,b^k)$ of choosing the new facility location z relative to b^k. In many instances the functions d_k are just "metrics" (i.e., generalized distance functions) that reflect the cost of shipping materiel between the two locations. Such metric functions are usually determined by the available transportation systems. The problem then is to choose the new location z so that the total cost

$$d(z) = \sum_{k=1}^{p} d_k(z,b^k)$$

is minimized. However, minimizing d(z) is clearly equivalent to solving problem $A_1(0)$ when

$$g(x) \triangleq \sum_{k=1}^{p} d_k(x^k,b^k)$$

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

where $x = (x^1, x^2, \ldots, x^p)$ and there are a total of p identity matrices I.

Notice that, unlike the function d, the function g is at least partially separable in that it is a sum of terms, each of which depends on only a single independent vector variable x^k. Furthermore, it is clear that each component of u perturbs a corresponding component of one of the fixed locations b^k when each metric d_k is generated by a "norm" $||\cdot||_k$; that is, $d_k(x^k,b^k) \triangleq ||x^k - b^k||_k$. The effect of such perturbations is clearly of interest in studying the potential pay-off from relocating the existing facilities.

Examples 4 and 5. There are two other special examples that should be mentioned: "dynamic programming" with linear "transition functions," and (nonlinear) "multicommodity network flow" problems. In the dynamic programming example the linear transition functions are used to define the vector space X. In the network flow example the (linear) "conservation laws" are used to define X. The details for these two examples are left to the imagination of the interested reader.

It will prove helpful to establish some elementary properties of the family A_1 prior to giving an additional important example of great generality. We begin by giving a more precise description of A_1.

Theorem 2A. The feasible perturbation set U is the nonempty "cylinder"

$$U = C - X.$$

Moreover, if C is a convex set, then so is U, and the point-to-set mapping $u \to S(u)$ is "concave" in that

$$\delta_1 S(u^1) + \delta_2 S(u^2) \subseteq S(\delta_1 u^1 + \delta_2 u^2)$$

for each "convex combination" $\delta_1 u^1 + \delta_2 u^2$ of arbitrary points $u^1, u^2 \in U$. Furthermore, if g is a convex function on C, then either the infimum function ϕ_1 is finite and convex on U or $\phi_1(u) \equiv -\infty$ for each $u \in$ (int U) (the "interior" of U).

The proof of this theorem is omitted (Peterson, 1970b).

It should be mentioned that there are convex families A_1 such that $\phi_1(u)$ is finite for at least one $u \in$ (bd U) (the "boundary" of U) even though $\phi_1(u) \equiv -\infty$ for each $u \in$ (int U). An example can be found in Appendix C of Peterson (1970b).

The following theorem reduces the study of the family A_1 to a study of only those problems $A_1(u)$ for which $u \in Y$, where $Y \triangleq X^{\perp}$. In this context, it will be convenient to adopt the notation u_X and u_Y for the orthogonal projection of an arbitrary vector u onto the orthogonal complementary subspaces X and Y.

Theorem 2B. For each vector $u \in E_n$, either the feasible solution sets S(u) and $S(u_Y)$ are both empty, or both are nonempty, with the latter being the case if, and only if, $u \in U$, in which case

$$S(u) = S(u_Y) - u_X$$

and

$$\phi_1(u) = \phi_1(u_Y).$$

Furthermore, if $u \in U$, then either the optimal solution sets $S^*(u)$ and $S^*(u_Y)$ are both empty, or both are nonempty and

$$S^*(u) = S^*(u_Y) - u_X.$$

The proof of this theorem can be found in Peterson (1970b).

Example 6. The preceding reduction theorem provides the prerequisites for introducing our sixth example, the recently developed Rockafellar formulation (1968) of optimization theory.

To obtain the Rockafellar formulation, let

$$X \triangleq \text{column space of } \begin{bmatrix} I_p \\ O_q \end{bmatrix},$$

where I_p is the $p \times p$ identity matrix, O_q is the $q \times p$ zero matrix, and $p + q = n$. Then the orthogonal complement Y of X is, of course, given by the equation

$$Y = \text{column space of } \begin{bmatrix} O_p \\ I_q \end{bmatrix},$$

where I_q is the $q \times q$ identity matrix, and 0_p is the $p \times q$ zero matrix. Taking account of Theorem 2B, we see that the resulting specialized family A_1 is equivalent to the following family A_1' of mathematical programming problems $A_1'(u')$.

Problem $A_1'(u')$. Using the feasible solution set

$$S'(u') \triangleq \{x' \in E_p \mid (x',u') \in C'\},$$

calculate both the problem infimum

$$\phi_1'(u') \triangleq \inf_{x' \in S'(u')} g'(x',u')$$

and the optimal solution set

$$S'^*(u') \triangleq \{x' \in S'(u') \mid g'(x',u') = \phi_1'(u')\}.$$

It is to be understood that u' is a vector parameter in E_q so that the Cartesian product (x',u') is in E_n; and the function $g':C'$ is, of course, identical to the function $g:C$.

Each optimization problem can be put into the form of problem $A_1'(0)$, and a given $A_1'(0)$ can be embedded in a very large class of permissible families A_1', with the particular choice of an A_1' to be determined by the particular "perturbations" being studied. With this in mind, Rockafellar (1968) has termed each such mapping $u' \to [S'(u'),g'(\cdot,u')]$ a "bifunction," and the corresponding problem $A_1'(0)$ together with a local analysis of the infimum function ϕ_1' at $u' = 0$, a "generalized program." Note that this Rockafellar formulation A_1' shows that the class of perturbations permitted in the geometric programming formulation A_1 of a given optimization problem $A_1(0)$ is not nearly as small as we might suspect from a superficial examination of problem $A_1(u)$.

Note also that Appendix A of Peterson (1970b) shows that the present geometric programming formulation of optimization theory can also be viewed as an example of the Rockafellar formulation, so the two formulations are actually equivalent. The geometric programming point of view tends to be the most useful when we are trying to simplify the statement of an optimization problem (e.g., write it in separable form). The Rockafellar point of view tends to be the most useful when we are more concerned with the perturbations to be studied.

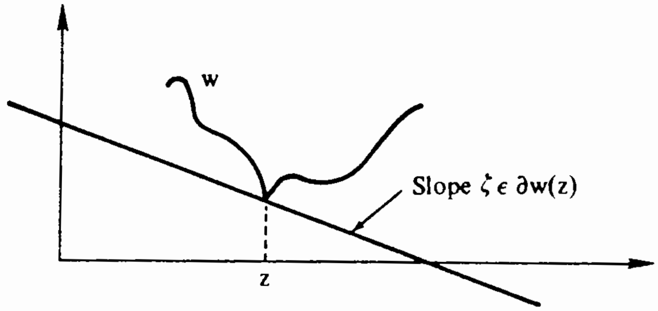
Both the geometric programming and Rockafellar formulations were preceded by another formulation of optimization theory, the original Fenchel formulation (1951). Although the Fenchel formulation is equivalent to both the geometric programming and Rockafellar formulations, it does not seem to have any particular virtue for studying problems in technological design.

3. GEOMETRIC PROGRAMMING DUAL FAMILIES

To introduce the extremely important concept of a dual family and its associated dual problem, we need the "conjugate transformation." The conjugate transformation maps functions into functions in such a way that the "conjugate transform" ω of an arbitrary function w has functional values

$$\omega(\zeta) \triangleq \sup_{z \in W} \{\langle\zeta,z\rangle - w(z)\},$$

where W is the domain of w. Of course, the domain Ω of ω is defined to be the set of all those vectors ζ for which this supremum is finite, and the conjugate transform ω exists only when Ω is not empty.



Geometrical insight into the conjugate transformation can be obtained by considering the "subgradient" set

$$\partial w(z) \triangleq \{\zeta | w(z) + \langle\zeta, z' - z\rangle \leqq w(z') \text{ for each } z' \in W\}$$

for w at z. The subgradient concept is related to, but considerably different from, the more familiar "gradient" concept. The gradient provides a "tangent hyperplane" for the function's graph; while the subgradient provides a "supporting hyperplane" for the function's graph, in that the defining inequality simply states that the hyperplane with equation $w' = w(z) + \langle\zeta, z' - z\rangle$ intersects the graph of w at the point $[z, w(z)]$ and lies entirely on or below it.

It is clear from the displayed example that a subgradient may exist and not be unique even when the gradient does not exist. On the other hand, it is also clear that a subgradient may not exist even when the gradient exists. There is, however, an important class of functions whose gradients are also subgradients — the class of convex functions. In fact, the notions of gradient and subgradient are identical for the class of differentiable convex functions defined on open sets, a class that we shall be working with in many of our examples.

To relate the conjugate transform to subgradients, observe that if $\zeta \in \partial w(z)$, then

$$\langle\zeta, z'\rangle - w(z') \leqq \langle\zeta, z\rangle - w(z) \text{ for each } z' \in W,$$

which implies that $\zeta \in \Omega$ and that

$$\omega(\zeta) = -\{w(z) + \langle\zeta, -z\rangle\}.$$

Hence, $\omega(\zeta)$ is simply the negative of the intercept of the supporting hyperplane with the w'-exis. Actually, the conjugate transform ω restricted (in the set-theoretic sense) to the domain $\bigcup_{z \in W} \partial w(z)$ is termed the "Legendre transform" of w and has been a major tool in the study of classical mechanics, thermodynamics, and differential equations. The domain Ω of the conjugate transform ω generally consists of both $\bigcup_{z \in W} \partial w(z)$ and some of its limit points.

When it exists, the conjugate transform is known to be both "closed" (i.e., "lower semicontinuous") and convex. In fact, Fenchel (1949, 1951) has shown that the conjugate transformation provides a one-to-one mapping of the family of all closed convex functions onto itself in symmetric fashion (i.e., the mapping is its own inverse). Two such functions are said to be "conjugate functions" when they are the conjugate transform of one another.

Each function w and its conjugate transform ω give rise to an important inequality

$$\langle z, \zeta\rangle \leqq w(z) + \omega(\zeta)$$

that is clearly valid for every point $z \in W$ and every point $\zeta \in \Omega$ [as can be seen from the defining equation for $\omega(\zeta)$]. Moreover, elementary computations show that this "conjugate inequality" is actually an equality if, and only if, $\zeta \in \partial w(z)$. Of course, the condition $\zeta \in \partial w(z)$ is equivalent to the condition $z \in \partial\omega(\zeta)$ when w is closed and convex (by virtue of the symmetry of the conjugate transformation when operating on such functions).

This completes the prerequisites for introducing the concept of duality into geometric programming.

The "geometric dual" B_1 of the family A_1 exists only when the conjugate transform h of the function g exists; that is, when the domain D of h is nonempty. The family B_1 is then defined in terms of h and the orthogonal complement Y of X. In particular, the family B_1 consists of the following geometric programming problems $B_1(v)$.

Problem $B_1(v)$. Using the feasible solution set

$$T(v) \triangleq Y \cap (D - v),$$

calculate both the problem infimum

$$\psi_1(v) \triangleq \inf_{y \in T(v)} h(y + v)$$

and the optimal solution set

$$T^*(v) \triangleq \{y \in T(v) \,|\, h(y + v) = \psi_1(v)\}.$$

Families A_1 and B_1 are clearly of the same type, except that B_1 contains only convex programming problems because of the nature of the conjugate transformation. Notice how B_1 is obtained from A_1; the function g is replaced by its conjugate transform h, and the vector subspace X is replaced by its orthogonal complement Y. Hence, when g is closed and convex, the symmetry of the conjugate transformation and the symmetry of the orthogonal complement relation imply that the family obtained by applying the same transformation to B_1 is again A_1. Because of this symmetry (in the closed convex case), A_1 and B_1 are termed dual families of geometric programming problems.

Each of the dual families A_1 and B_1 contains a problem of special interest, namely, problems $A_1(0)$ and $B_1(0)$. Due to the apparent symmetry between them (in the closed convex case), $A_1(0)$ and $B_1(0)$ are termed dual problems. To avoid confusion, it is important to bear in mind that problems $A_1(u)$ and $B_1(v)$ are termed dual problems only when u and v are zero.

However, problem $A_1(u)$ does have a dual problem, and it can be obtained by observing that $A_1(u)$ is essentially problem $A_1(0)$ with the function $g:C$ replaced by the function $g(\cdot + u):C - u$. Thus the dual of problem $A_1(u)$ is problem $B_1(0)$ with $h:D$ replaced by the conjugate transform $h_u:D_u$ of $g(\cdot + u):C - u$. To compute $h_u:D_u$, notice that

$$h_u(\zeta) = \sup_{z \in C-u} \{\langle\zeta,z\rangle - g(z + u)\}$$

$$= \sup_{c \in C} \{\langle\zeta,c\rangle - g(c)\} - \langle u,\zeta\rangle,$$

which shows that $D_u = D$ and that $h_u(\cdot) = h(\cdot) - \langle u,\cdot\rangle$. It follows that the dual of problem $A_1(u)$ has the same feasible solution set T(0) as the dual problem $B_1(0)$; only the objective function $h(\cdot):D$ for $B_1(0)$ is altered to give the objective function $h(\cdot) - \langle u,\cdot\rangle:D$ for the dual of

$A_1(u)$. Hence, we might expect the feasible solution set $T(0)$ for the dual problem $B_1(0)$ to play an important role in a study of the family A_1. This expectation is confirmed by the results to be given here.

Because of the symmetry between the families A_1 and B_1, it is clear that problem $B_1(v)$ also has a dual, namely problem $A_1(0)$ with $g(\cdot):C$ replaced by $g(\cdot) - \langle v,\cdot\rangle:C$. The analogs of the remarks made about problem $A_1(u)$ and its dual are left to the reader.

Unlike the usual min-max formulations of duality in mathematical programming, both problem $A_1(0)$ and its dual problem $B_1(0)$ are minimization problems. The relative simplicity of this min-min formulation will soon become clear, but the reader who is accustomed to the usual min-max formulation must bear in mind that a given duality theorem will generally have slightly different statements, depending on the formulation in use. In particular, a theorem that asserts the equality of the min and max in the usual formulation will assert that the sum of the mins is zero [i.e., $\phi_1(0) + \psi_1(0) = 0$] in the present formulation.

The symmetry between families A_1 and B_1 induces a symmetry on the theory that relates A_1 and B_1 (in the closed convex case). Thus each mathematical statement about A_1 and B_1 automatically produces an equally valid "dual statement" about B_1 and A_1. To be concise, our attention will be focused on the family A_1, and each dual statement will be left to the reader's imagination.

It is now instructive to compute the geometric dual B_1 of the family A_1 for each of the examples given in Sec. 2. To do so, we need only compute the corresponding conjugate transforms $h:D$ and the corresponding orthogonal complements Y. The former computations are possible (in fact, easy) because all example functions $g:C$ are separable. Of course, the latter computations are always easy because they involve only elementary linear algebra. Hence, we shall now take advantage of one of the main features of the geometric programming point of view — the resulting separability of the functions $g:C$.

Example 1. To obtain the geometric dual B_1 of the family A_1 corresponding to generalized polynomial optimization, we compute the conjugate transform $h:D$ of $\sum_{i=1}^{n} c_i e^{x_i}$. This conjugate transform $h:D$ exists only if each coefficient $c_i > 0$, in which case we are dealing only with convex functions and "posynomial" minimization. However, we are able to remove this restriction by employing additional devices as in Sec. 4. To compute $h:D$, observe that

$$h(y) = \sup_{x\in E_n} \left\{\langle y,x\rangle - \sum_{i=1}^{n} c_i e^{x_i}\right\}$$

$$= \sum_{i=1}^{n} \sup_{x_i\in E_1} \{y_i x_i - c_i e^{x_i}\}$$

$$= \sum_{i=1}^{n} y_i \log\left(\frac{y_i}{c_i}\right) - \sum_{i=1}^{n} y_i$$

for $y \in D = \{y \in E_n \mid y_i \geq 0 \text{ for } i = 1, 2, \ldots, n\}$, with the understanding that $y_i \log y_i = 0$ when $y_i = 0$. The orthogonal complement Y of the column space X of the exponent matrix (a_{ij}) is simply the set of all those vectors $y \in E_n$ that satisfy the "orthogonality conditions" $\sum_{i=1}^{n} a_{ij}y_i = 0$ for $j = 1, 2, \ldots, m$.

Hence, the resulting specialized family B_1 is obtained by letting

$$h(y) = \sum_{i=1}^{n} y_i \log\left(\frac{y_i}{c_i}\right) - \sum_{i=1}^{n} y_i,$$

$$D = \{y \in E_n \mid y_i \geq 0 \text{ for } i = 1, 2, \ldots, n\},$$

and $$Y = \{y \in E_n \mid \sum_{i=1}^{n} a_{ij}y_i = 0 \text{ for } j = 1, 2, \ldots, m\}.$$

The resulting dual problem $B_1(0)$ was first obtained by Duffin (1962a), but it is not the same dual problem that was eventually incorporated into the "prototype formulation" (Duffin, 1966; Duffin, Peterson, and Zener, 1967) of geometric programming. To obtain the prototype dual problem, we must perform a "suboptimization" (Duffin, 1962a) whose details can also be found on page 500 of Peterson (1970b).

Example 2. To obtain the geometric dual B_1 of the family A_1 corresponding to ℓ_p programming, we compute the conjugate transform $h:D$ of $\sum_{i=1}^{n-1} p_i^{-1}|x_i - b_i|^{p_i} + x_n - b_n$. In essence we are assuming that H_2 is the zero matrix so that the negative terms $-p_i^{-1}|x_i - b_i|^{p_i}$ need not appear in the original definition of $g:C$. Of course, this restricts our attention to only convex ℓ_p programming, but this is the only case in which the conjugate transform $h:D$ exists. However, this restriction can probably be eliminated with further research. To compute $h:D$, observe that

$$h(y) = \sup_{x \in E_n} \left\{\langle y,x\rangle - \sum_{i=1}^{n-1} p_i^{-1}|x_i - b_i|^{p_i} - x_n + b_n\right\}$$

$$= \sum_{i=1}^{n-1} \sup_{x_i \in E_1} \{y_i x_i - p_i^{-1}|x_i - b_i|^{p_i}\} + \sup_{x_n \in E_1} \{(y_n - 1)x_n\} + b_n$$

$$= \sum_{i=1}^{n-1} (q_i^{-1}|y_i|^{q_i} + b_i y_i) + b_n$$

for $y \in D = \{y \in E_n \mid y_n = 1\}$, where q_i is determined from p_i by the relation $p_i^{-1} + q_i^{-1} = 1$ for $i = 1, 2, \ldots, n-1$. The orthogonal complement Y of the column space X of the partitioned matrix $\begin{bmatrix} H \\ h \end{bmatrix}$ is simply the set of all those vectors $y \in E_n$ that satisfy the conditions $\sum_{i=1}^{n-1} H_{ij}y_i + h_j y_n = 0$ for $j = 1, 2, \ldots, m$.

Hence, the resulting specialized family B_1 is obtained by letting

$$h(y) = \sum_{i=1}^{n-1} (q_i^{-1}|y_i|^{q_i} + b_i y_i) + b_n,$$

$$D = \{y \in E_n | y_n = 1\},$$

and
$$Y = \{y \in E_n | \sum_{i=1}^{n-1} H_{ij} y_i + h_j y_n = 0 \text{ for } j = 1, 2, \ldots, m\}.$$

The resulting dual problem $B_1(0)$ was first obtained by Peterson and Ecker (1968), but another geometric dual problem corresponding to a slightly different formulation of the special ℓ_p-regression problem had previously been introduced and studied in Chapter VII of Duffin, Peterson, and Zener (1967). The latter dual problem can be obtained from the former by a suboptimization similar to the one described on page 500 of Peterson (1970b) for posynomial problems.

Example 3. To obtain the geometric dual B_1 of the family A_1 corresponding to the problem of optimally locating a new facility, we compute the conjugate transform $h{:}D$ of $\sum_{k=1}^{p} d_k(x^k,b^k)$. This conjugate transform $h{:}D$ exists only when the conjugate transform $h_k{:}D_k$ exists for each function $d_k(\cdot,b^k)$, a condition that is always satisfied when each function $d_k(\cdot,\cdot)$ is a metric. To compute $h{:}D$ in such a case, observe that

$$\begin{aligned} h(y) &= \sup_{x \in E_n} \{ \sum_{k=1}^{p} \langle y^k,x^k \rangle - \sum_{k=1}^{p} d_k(x^k,b^k)\} \\ &= \sum_{k=1}^{p} \sup_{x^k \in E_{n_k}} \{\langle y^k,x^k \rangle - d\ (x^k,b^k)\} \\ &= \sum_{k=1}^{p} h_k(y^k,b^k) \end{aligned}$$

for $y \in D = \bigtimes_{k=1}^{p} D_k$. If, in particular, $d_k(x^k,b^k) = ||x^k - b^k||_k$ where $||\cdot||_k$ is a norm, then

$$\begin{aligned} h_k(y^k,b^k) &= \sup_{x^k \in E_{n_k}} \{\langle y^k,x^k \rangle - ||x^k - b^k||_k\} \\ &= \sup_{w^k \in E_{n_k}} \{\langle y^k,w^k + b^k \rangle - ||w^k||_k\} \\ &= \sup_{w^k \in E_{n_k}} \{\langle y^k,w^k \rangle - ||w^k||_k\} + \langle b^k,y^k \rangle \\ &= \langle b^k,y^k \rangle \end{aligned}$$

for $y^k \in D_k = \partial||0||_k$. Now, the orthogonal complement Y of the column space X of the partitioned matrix $\begin{bmatrix} U \\ U \\ \vdots \\ U \end{bmatrix}$ is simply the set of all those vectors $y \in E_n$ that satisfy the condition $\sum_{k=1}^{p} y^k = 0$.

Hence, the resulting specialized family B_I is obtained by letting

$$h(y) = \sum_{k=1}^{p} h_k(y^k, b^k),$$

$$D = \mathop{\times}_{k=1}^{p} D_k,$$

and $$Y = \{y \in E_n | \sum_{k=1}^{p} y^k = 0\}.$$

In particular, $$h_k(y^k, b^k) = \langle b^k, y^k \rangle$$

and $$D_k = \partial||0||_k$$

when $d_k(x^k, b^k) = ||x^k - b^k||_k$. These problems have recently been investigated by Wendell and Peterson (1971).

Examples 4 and 5. A computation of the geometric dual B_I of the family A_I arising in dynamic programming is, of course, left to the imagination of the interested reader, as is the corresponding computation for multicommodity network flow problems.

Example 6. To obtain the geometric dual B_I of the family A_I corresponding to the Rockafellar formulation of optimization theory, we use the assumption that X is the column space of the special matrix $\begin{bmatrix} I_p \\ 0_q \end{bmatrix}$, and the resulting fact that Y is the column space of the special matrix $\begin{bmatrix} 0_p \\ I_q \end{bmatrix}$. Then, taking account of the (unstated) dual of the reduction theorem, Theorem 2B, we see that the resulting specialized family B_I is equivalent to the following family B_I' of programming problems $B_I'(v')$.

Problem $B_I'(v')$. Using the feasible solution set

$$T'(v') \triangleq \{y' \in E_q | (v', y') \in D'\},$$

calculate both the problem infimum

$$\psi_I'(v') \triangleq \inf_{y' \in T'(v')} h'(v', y')$$

and the optimal solution set

$$T'^*(v') \triangleq \{y' \in T'(v') | h'(v', y') = \psi_I'(v')\}.$$

It is to be understood that v' is a vector parameter in E_p so that the Cartesian product (v', y') is in E_n; and the function $h':D'$ is, of course, identical to the closed convex function $h:D$.

Families A_1' and B_1' are clearly of the same type, except that B_1' contains only convex programming problems because of the nature of the conjugate transformation. Notice how B_1' can be obtained directly from A_1'; simply replace the function $g':C'$ by its conjugate transform $h':D'$, replace the variable vector $x' \in E_p$ by the perturbation vector $v' \in E_p$, and replace the perturbation vector $u' \in E_q$ by the variable vector $y' \in E_q$. Hence, when $g:C$ is closed and convex, the symmetry of the conjugate relation and the symmetry involved in permuting Cartesian products imply that the family obtained by applying the same transformation to B_1' is again A_1'. Because of this symmetry and the resulting symmetry between problems $A_1'(0)$ and $B_1'(0)$, Rockafellar (1968) has termed them "dual generalized programs." Actually, Rockafellar formulates B_1' as a family of "concave programs" by placing minus signs at crucial (and difficult to remember) places, but that version requires a parallel discussion of the maximization of concave functions and hence is not as simple as the present version.

Appendix A of Peterson (1970b) shows that the present geometric programming formulation of duality can also be viewed as a special case of the preceding Rockafellar formulation, so the two formulations are actually equivalent. However, an examination of Appendix A should convince the reader that Rockafellar's formulation is more easily obtained from the geometric programming formulation than vice versa. Moreover, the presence of explicit constraints ruins the symmetry of Rockafellar's formulation, and it is not presently known how to overcome that difficulty. In contrast, we shall soon see that the present geometric programming formulation can be modified without any loss of symmetry to include arbitrary (convex) programming problems with explicit constraints.

Both the geometric programming and Rockafellar formulations were preceded by another formulation of duality, the original Fenchel formulation (1951). Although the Fenchel formulation is equivalent to both the geometric programming and Rockafellar formulations, it does not seem to have any particular virtue for studying problems in technological design.

4. DUALITY GAPS AND THE EXTREMALITY CONDITIONS

We begin with the most basic and easily proved duality theorem, which leads directly to both the duality gap concept and the extremality conditions. *Unless otherwise stated, we assume that the function $g:C$ is both closed and convex throughout the rest of this paper.* This guarantees the existence of the conjugate transform $h:D$ and leads to much stronger and more useful conclusions in our theorems.

Theorem 4A. *If x and y are feasible solutions to the dual problems $A_1(0)$ and $B_1(0)$ respectively, then*

$$0 \leqq g(x) + h(y),$$

with equality holding if, and only if, $x \in \partial h(y)$ or, equivalently, $y \in \partial g(x)$, in which case x and y are optimal solutions to $A_1(0)$ and $B_1(0)$ respectively.

Proof. Because x and y are in the orthogonal complementary subspaces X and Y, we see from the conjugate inequality that $0 = \langle x,y\rangle \leq g(x) + h(y)$. with equality holding if, and only if, $x \in \partial h(y)$, or $y \in \partial g(x)$. Moreover, it is a direct consequence of this inequality that x and y are optimal solutions to problems $A_1(0)$ and $B_1(0)$ when $0 = g(x) + h(y)$. This completes the proof of Theorem 4A.

The basic inequality provided by Theorem 4A implies important properties of the dual infima $\phi_1(0)$ and $\psi_1(0)$.

Corollary 4A1. If the geometric dual problems $A_1(0)$ and $B_1(0)$ are both consistent, then

(i) the infimum $\phi_1(0)$ for problem $A_1(0)$ is finite, and

$$0 \leqq \phi_1(0) + h(y)$$

for each feasible solution y to problem $B_1(0)$;

(ii) the infimum $\psi_1(0)$ for problem $B_1(0)$ is finite, and

$$0 \leqq \phi_1(0) + \psi_1(0).$$

The proof of this corollary is, of course, a trivial application of Theorem 4A.

Consistent dual problems $A_1(0)$ and $B_1(0)$ for which $0 < \phi_1(0) + \psi_1(0)$ are said to have a duality gap of $\phi_1(0) + \psi_1(0)$. It is well known that duality gaps do not occur in finite linear programming, but they do occur in infinite linear programming where this phenomenon was first encountered by Duffin (1956). They also occur in the present formulation of geometric programming, and examples due originally to J.J. Stoer are given in Appendix C of Peterson (1970b). However, we shall see that duality gaps in the present formulation are extremely rare (when $g:C$ is assumed to be closed and convex) in that they are excluded by very weak conditions on the dual problems $A_1(0)$ and $B_1(0)$. This extreme scarcity of duality gaps is very fortunate because of their highly undesirable properties.

Duality gaps are undesirable from a theoretical point of view because we shall see that relatively little can be said about the corresponding dual problems. They are also undesirable from a computational point of view because they usually destroy the possibility of using the inequality $0 \leqq g(x) + h(y)$ to provide an algorithm stopping criterion.

Such a criterion results from specifying a positive tolerance ε so that the numerical algorithms being used to minimize both $g(x)$ and $h(y)$ are terminated when they produce a pair of feasible solutions x^{+} and y^{+} for which

$$g(x^{+}) + h(y^{+}) \leqq 2\varepsilon.$$

Because conclusion (i) to Corollary 4A1 and the defining property for $\phi_1(0)$ show that

$$-h(y^{+}) \leqq \phi_1(0) \leqq g(x^{+}),$$

we conclude from the preceding tolerance inequality that

$$\left|\phi_1(0) - \frac{g(x^{+}) - h(y^{+})}{2}\right| \leqq \varepsilon.$$

Hence, $\phi_1(0)$ can be approximated by $[g(x^{+}) - h(y^{+})]/2$ with an error no greater than $\pm\varepsilon$; and, dually, $\psi_1(0)$ can be approximated by $[h(y^{+}) - g(x^{+})]/2$, also with an error no greater than $\pm\varepsilon$.

Note, however, that the defining properties for $\phi_1(0)$ and $\psi_1(0)$ imply that

$$\phi_1(0) + \psi_1(0) \leqq g(x) + h(y)$$

for each pair of feasible solutions x and y. Now, suppose that the dual problems $A_1(0)$ and $B_1(0)$ have a duality gap, and let the positive tolerance ε be chosen so small that

$$2\varepsilon < \phi_1(0) + \psi_1(0).$$

[Of course, such a choice for ε is possible if, and only if, the dual problems $A_1(0)$ and $B_1(0)$ actually have a duality gap.] From the preceding two inequalities we easily infer that there are no feasible solutions $x^\dagger$ and $y^\dagger$ for which

$$g(x^\dagger) + h(y^\dagger) \leqq 2\varepsilon;$$

so the numerical algorithms being used to minimize both g(x) and h(y) will never be terminated. Consequently, this algorithm stopping criterion may not be very useful for solving dual problems $A_1(0)$ and $B_1(0)$ that have a duality gap, especially if the gap is rather large.

For those dual problems $A_1(0)$ and $B_1(0)$ that do not have a duality gap, Theorem 4A provides a useful characterization of dual optimal solutions x^* and y^* in terms of the following <u>extremality conditions</u>:

(I) $\qquad x \in X \quad$ and $\quad y \in Y,$

(II) $\qquad$ either $\quad x \in \partial h(y) \quad$ or $\quad y \in \partial g(x).$

We formalize this characterization as the following corollary.

<u>Corollary</u> 4A2. <u>Suppose that the geometric dual problems</u> $A_1(0)$ <u>and</u> $B_1(0)$ <u>are both consistent and</u> $0 = \phi_1(0) + \psi_1(0)$. <u>Then arbitrary vectors</u> x <u>and</u> y <u>are optimal solutions to problems</u> $A_1(0)$ <u>and</u> $B_1(0)$, <u>respectively</u>, <u>if</u>, <u>and only if</u>, x <u>and</u> y <u>satisfy the extremality conditions</u> (I) <u>and</u> (II).

The proof of this corollary is an immediate consequence of Theorem 4A and the conjugate transform relations $\partial g(x) \subseteq D$ and $\partial h(y) \subseteq C$ that were described in Sec. 3.

Note that the preceding characterization is especially useful when both $A_1(0)$ and $B_1(0)$ are known to have nonempty optimal solution sets $S^*(0)$ and $T^*(0)$. In that case, Corollary 4A2 provides <u>a direct method for calculating all optimal solutions from the knowledge of only a single optimal solution</u>. For example, if x^* is a known optimal solution to $A_1(0)$, then $T^*(0) = Y \cap \partial g(x^*)$, and $S^*(0) = X \cap \partial h(y^*)$ for each $y^* \in T^*(0)$.

The extremality conditions (I) and (II) constitute the analog of the "complementary slackness conditions" in linear programming, and they can be specialized to give the "extremality conditions" stated in Rockafellar (1967a) for the Fenchel-Rockafellar formulation of duality.

The following theorem, which was first established by Duffin (1962a), has much stronger hypotheses than those required <u>to exclude duality gaps</u>, but its proof is relatively simple and very informative.

<u>Theorem</u> 4B. <u>If problem</u> $B_1(0)$ <u>has an optimal solution</u> y^* <u>at which</u> h:D <u>is differentiable</u>, <u>then the dual problem</u> $A_1(0)$ <u>has a unique optimal solution</u> $x^* \triangleq \nabla h(y^*)$, <u>and</u> $0 = \phi_1(0) + \psi_1(0)$.

<u>Proof</u>. First, observe that $x^* \in X$; otherwise, there would exist a direction vector $d \in Y$ such that $\langle \nabla h(y^*), d \rangle < 0$, which would imply (by virtue of the differential calculus) the existence of a sufficiently small $\varepsilon > 0$ such that $h(y^* + \varepsilon d) < h(y^*)$, a contradiction of the assumed optimality of y^*. Next, observe that $x^* \in C$ because $\nabla h(y^*) \in \partial h(y^*) \subseteq C$. Hence, x^* and y^* are feasible solutions to $A_1(0)$ and $B_1(0)$, respectively, so Theorem 4A and the assumption that $x^* = \nabla h(y^*)$ imply that

$0 = g(x^*) + h(y^*)$ and therefore that $0 = \phi_1(0) + \psi_1(0)$. The uniqueness of x^* follows from Corollary 4A2 and the fact that $\partial h(y^*) = \{\nabla h(y^*)\}$ by virtue of the properties of subgradients and gradients for convex functions (as described in Sec. 3).

Deeper duality theorems are established in Sec. 9 of Peterson (1970b) with the help of the mathematical machinery described in the next section. That machinery is also of intrinsic interest because it provides both an economic interpretation of duality and a method for constructing the infimum function $\phi_1:U$ without employing numerical optimization techniques.

5. THREE ECONOMICS PROBLEMS IN MANAGEMENT SCIENCE

To motivate the additional programming problems to be introduced in this section, let us review the problem of Zener Corporation. (Editor's note: See Sec. 2 of "An Introduction to Mathematical Programming" by Peterson.) Thus, suppose that Zener Corporation manufactures m different raw materials, and assume that Zener Corporation's technology limits its feasible "product mixes" to vectors $z \in E_m$ for which $Mz \in C - u$, where M is essentially an appropriate $(n \times m)$ "technology matrix," and where $u \in E_n$ is an arbitrary perturbation of Zener Corporation's warehouse store of raw materials. In particular, a positive value for u_i is to indicate the external disposal of u_i units of raw material i, and a negative value for u_i is to indicate the acquisition of $-u_i$ additional units of raw material i. Also, suppose that $g(Mz + u)$ is the difference between Zener Corporation's marketing cost and the price it receives for the feasible product mix z, and let X be the column space of M. Then, given that Zener Corporation carries out a feasible raw material perturbation $u \in U$, its set of feasible transformed product mixes $x = Mz$ is just the nonempty feasible solution set $S(u)$; its minimum possible cost is the problem infimum $\phi_1(u)$; and its set of optimal transformed product mixes x^* is the optimal solution set $S^*(u)$. Of course, a negative cost infimum $\phi_1(u)$ is to be interpreted as a positive profit supremum $-\phi_1(u)$.

Presumably, Zener Corporation can externally dispose of some of its raw material by selling it, and can acquire additional raw material by buying it. We shall assume in this paper that Zener Corporation can either buy or sell each raw material i for y_i dollars per unit quantity on the "raw material market," where the raw material price vector y is an arbitrary, but fixed, vector in E_n that cannot be influenced by Zener Corporation's actions.

Thus, Zener Corporation's total maximum profit is $\langle y,u\rangle - \phi_1(u)$ for a feasible perturbation $u \in U$, and it is reasonable that Zener Corporation should want to adjust u to maximize its total maximum profit. For purposes of easy reference and mathematical precision, this economics problem from management science can now be formally stated.

<u>Problem $A_2(y)$.</u> Using the feasible perturbation set

$$U \triangleq \{u \in E_n | S(u) \text{ is not empty}\},$$

calculate both the problem supremum

$$\phi_2(y) \triangleq \sup_{u \in U} \{\langle y,u\rangle - \phi_1(u)\}$$

and the "optimal perturbation" set

$$U^*(y) \triangleq \{u \in U | \langle y,u\rangle - \phi_1(u) = \phi_2(y)\}.$$

In addition, given $u^* \in U^*(y)$, calculate the optimal solution set

$$S^*(u^*) \triangleq \{x \in S(u^*) | g(x + u^*) = \phi_1(u^*)\}.$$

According to Theorem 2A, the feasible perturbation set U is the nonempty cylinder C - X, so problem $A_2(y)$ is consistent for each y in E_n. Thus, the domain of the supremum function ϕ_2 is all of E_n. Note, however, that the range of ϕ_2 may contain the point $+\infty$. But in contrast to problem $A_1(u)$, observe that the optimal perturbation set $U^*(y)$ for $A_2(y)$ need not be empty even though $\phi_2(y) = +\infty$. Nevertheless, it is clear that each optimal solution set $S^*(u^*)$ must be empty when $\phi_2(y) = +\infty$.

It is possible to give a complete solution of problem $A_2(y)$ in terms of the feasible solution set T(0) and the objective function h:T(0) for the dual problem $B_1(0)$. This close relationship between the problem family A_2 and problem $B_1(0)$ is not very surprising when we observe that the supremum function ϕ_2 is just the conjugate transform of the infimum function ϕ_1 which, in turn, is a constrained infimum of the function g whose conjugate transform is h; but the actual derivation is not straightforward, and it helps to first solve another closely related economics problem A(y).

To motivate problem A(y), observe that $A_2(y)$ is the second stage of a two-stage sequential optimization problem: first, minimize Zener Corporation's "production" cost $g(x + u)$ by adjusting its transformed product mix x subject to the feasibility constraint $x \in S(u)$ to obtain its minimum production cost $\phi_1(u)$ for each $u \in U$, and then maximize its total maximum profit $\langle y,u\rangle - \phi_1(u)$ by adjusting its raw material perturbation u subject to the feasibility constraint $u \in U$ to obtain its maximum total maximum profit $\phi_2(y)$. From Zener Corporation's point of view, this sequential optimization problem clearly seems to have no more economic relevance than the more easily stated nonsequential optimization problem: maximize Zener Corporation's total profit $\langle y,u\rangle - g(x + u)$ by simultaneously adjusting its transformed product mix x and its raw material perturbation u subject to the feasibility constraint $x \in S(u)$ to obtain its maximum total profit.

For purposes of easy reference and mathematical precision, the preceding nonsequential optimization problem is now stated formally.

<u>Problem A(y)</u>. Using the "feasible strategy" set

$$W \triangleq \{(x,u) \in E_{2n} | x \in S(u)\},$$

calculate both the problem supremum

$$\phi(y) \triangleq \sup_{(x,u) \in W} \{\langle y,u\rangle - g(x + u)\}$$

and the "optimal strategy" set

$$W^*(y) \triangleq \{(x,u) \in W | \langle y,u\rangle - g(x + u) = \phi(y)\}.$$

Notice that the feasible strategy set W is not empty by virtue of Theorem 2A, so problem A(y) is consistent for each $y \in E_n$. Thus, the domain of the supremum function ϕ consists of all of E_n, and the range of ϕ may, of course, contain the point $+\infty$; but if $\phi(y) = +\infty$, then the optimal strategy set $W^*(y)$ must clearly be empty.

Obviously, <u>problems</u> $A_2(y)$ <u>and</u> A(y) <u>are closely related</u>. In fact, the following theorem shows that they are equivalent in the sense that a solution of either one automatically provides a solution of the other.

Theorem 5A. The supremum $\phi_2(y)$ for problem $A_2(y)$ and the supremum $\phi(y)$ for problem $A(y)$ are identical for each $y \in E_n$. Moreover, if these suprema are finite for a particular $y \in E_n$, then

(i) the optimal perturbation set $U^*(y)$ and the optimal strategy set $W^*(y)$ are related by the set containment

$$U^*(y) \supseteq \{u \in E_n \mid (x,u) \in W^*(y) \text{ for some } x \in E_n\},$$

with set equality holding if, and only if, the optimal solution set $S^*(u^*)$ is nonempty for each $u^* \in U^*(y)$,

(ii) for each $u^* \in U^*(y)$ the optimal solution set $S^*(u^*)$ and the optimal strategy set $W^*(y)$ are related by the set equation

$$S^*(u^*) = \{x \in E_n \mid (x,u^*) \in W^*(y)\}.$$

The proof of this theorem is rather routine and can be found in Sec. 7 of Peterson (1970b).

The nonsequential optimization problem $A(y)$ seems to be easier to solve than the sequential optimization problem $A_2(y)$. Thus, we first solve $A(y)$ and then use Theorem 5A to translate that solution into a solution for $A_2(y)$.

The solution for problem $A(y)$ can be conveniently summarized as the following theorem.

Theorem 5B. The feasible strategy set W is nonempty and hence problem $A(y)$ is consistent for each $y \in E_n$. Moreover, the supremum $\phi(y)$ for problem $A(y)$ is finite if, and only if, $y \in T(0) \triangleq Y \cap D$, in which case:

(i) $\phi(y) = h(y)$;

(ii) the optimal strategy set $W^*(y)$ is nonempty if, and only if, $\partial h(y)$ is nonempty; in which event

$$W^*(y) = \{(x,y) \in E_{2n} \mid x \in X \text{ and } u \in \partial h(y) - x\}.$$

The essential idea in proving this theorem comes from observing that if $y \in Y \cap D$, then

$$\phi(y) = \sup_{(x,u) \in W} \{\langle y, x+u\rangle - g(x+u)\}$$

because $x \in X$ for each $(x,u) \in W$. But

$$\sup_{(x,u) \in W} \{\langle y, x+u\rangle - g(x+u)\} = \sup_{c \in C} \{\langle y, c\rangle - g(c)\} \triangleq h(y)$$

because $\{x + u \in E_n \mid (x,u) \in W\}$ is clearly identical to C. Hence,

$$\phi(y) = h(y)$$

and

$$W^*(y) = \{(x,u) \in E_{2n} \mid x \in X \text{ and } x + u \in \partial h(y)\}$$

by virtue of the conditions that characterize equality for the conjugate inequality.

Notice that the solution to problem $A(y)$ is intimately related to the dual problem $B_1(0)$; in particular, Theorem 5B can be summarized by the following statement. The supremum $\phi(y)$ for problem $A(y)$ is finite if, and only if, the vector y is a feasible solution to the dual problem $B_1(0)$; in which case, (i) $\phi(y)$ is identical to the dual objective function value $h(y)$, and (ii) a strategy $(x,u) \in E_{2n}$ is optimal if, and only if, it satisfies the extremality conditions $x \in X$ and $x \in \partial h(y) - u$. Of course, these extremality conditions are not the same as those provided in Sec. 4 for the dual problems $A_1(0)$ and $B_1(0)$, but are the extremality conditions that correspond to problem $A_1(u)$ and its dual problem, namely, problem $B_1(0)$ with $h(\cdot):D$ replaced by $h(\cdot) - \langle u,\cdot\rangle:D$.

The solution of problem $A_2(y)$ can now be readily obtained by using Theorem 5A to translate the solution of problem $A(y)$ provided in the preceding Theorem 5B. The resulting solution of problem $A_2(y)$ is stated here as the following theorem.

Theorem 5C. The feasible perturbation set U is nonempty and hence problem $A_2(y)$ is consistent for each $y \in E_n$. Moreover, the supremum $\phi_2(y)$ for problem $A_2(y)$ is finite if, and only if, $y \in T(0) \triangleq Y \cap D$, in which case:

(i) $\phi_2(y) = h(y)$;

(ii) either the optimal perturbation set $U^*(y)$ and its intersection with Y are both empty, or both are nonempty; in which event $U^*(y)$ is the cylinder

$$U^*(y) = (U^*(y) \cap Y) - X,$$

which is bounded from outside by the relation

$$U^*(y) \subseteq \partial[h:T(0)](y);$$

moreover, if $\partial h(y)$ is nonempty, then $U^*(y)$ is bounded from inside by the relation

$$\partial h(y) - X \subseteq U^*(y),$$

with the corresponding set theoretic difference being empty if, and only if, the optimal solution set $S^*(u^*)$ is nonempty for each $u^* \in U^*(y)$;

(iii) if $\partial h(y)$ is nonempty, then

$$S^*(u^*) = X \cap [\partial h(y) - u^*]$$

for each $u^* \in U^*(y)$, with $S^*(u^*)$ being nonempty for each $u^* \in U^*(y)$ when $y \in (\text{int } D)$; but if $\partial h(y)$ is empty, then $S^*(u^*)$ is empty for each $u^* \in U^*(y)$.

As with problem $A(y)$, the preceding solution to problem $A_2(y)$ can be phrased in terms of the dual problem $B_1(0)$ and the extremality conditions for problem $A_1(u)$ and its dual problem. However, such a statement of Theorem 5C will be left to the interested reader.

Note that Theorems 2A and 5C provide a method for constructing the infimum function $\phi_1:U$ without employing numerical optimization techniques. In particular, if the dual feasible solution set $T(0)$ is empty, then Theorem 2A asserts that $\phi_1(u) \equiv -\infty$ for each $u \in (\text{int } U)$; so there is virtually nothing left to know about $\phi_1:U$ in this case. The only other (more interesting) possibility is that $T(0)$ is nonempty, in which case we can cover it with a "mesh" of points

$$\{y^1, y^2, \ldots, y^s\} \subseteq T(0) \triangleq Y \cap D.$$

Then Theorem 5C implies that

$$\phi_1(u) = \langle y^k,u\rangle - h(y^k) \text{ for each } u \in \partial h(y^k) - X,$$

for $k = 1, 2, \ldots, s$. Moreover, it is a consequence of the properties of the conjugate transformation that the set of all such vectors u can be made as "dense" in U as desired, simply by choosing the mesh $\{y^1, y^2, \ldots, y^s\}$ to be sufficiently dense in $T(0)$. Of course, this method depends on an ability to construct the dual objective function $h:T(0)$, a construction that is rather easy for the important examples given in Secs. 2 and 3.

Aside from its preceding significance, Theorem 5C provides the following basis for proving important duality relations between the geometric dual problems $A_1(0)$ and $B_1(0)$.

Corollary 5C1. The infimum function $\phi_1:U$ for the family A_1 is finite on its entire domain U if, and only if, problem $B_1(0)$ is consistent, in which case the objective function $h:T(0)$ for problem $B_1(0)$ is the conjugate transform of the function $\phi_1:U$.

Proof. Theorem 2A asserts that either $\phi_1(u) \equiv -\infty$ on (int U) or $\phi_1:U$ is finite and convex on U; and problem $A_2(y)$ shows that the supremum function ϕ_2 is the conjugate transform of $\phi_1:U$. With these facts in mind, the proof of this corollary is an immediate consequence of Theorem 5C.

The preceding corollary was first obtained by Rockafellar within the Fenchel-Rockafellar formulation of duality. However, he was not aware of its economic significance and did not explicitly introduce the analogs of problems A(y) and $A_2(y)$, nor did he even implicitly study them as thoroughly as we have explicitly done in Theorems 5A, 5B and 5C. Rockafellar was interested primarily in applications to his dual problems. Analogous applications to the geometric dual problems $A_1(0)$ and $B_1(0)$ are given in Sec. 9 of Peterson (1970b). We reproduce only one of them here in the form of the following corollary.

Corollary 5C2. If the dual problems $A_1(0)$ and $B_1(0)$ are both consistent and $0 = \phi_1(0) + \psi_1(0)$, then

$$\partial\phi_1(0) = T^*(0).$$

Proof. Corollary 5C1 asserts that $h:T(0)$ is the conjugate transform of $\phi_1:U$; so the conjugate inequality implies that $\langle u,y\rangle \leq \phi_1(u) + h(y)$ for each $u \in U$ and each $y \in T(0)$, with equality holding if, and only if, $y \in \partial\phi_1(u)$. In particular, because $0 \in U$, we see that

$$0 \leq \phi_1(0) + h(y)$$

for each $y \in T(0)$, with equality holding if, and only if, $y \in \partial\phi_1(0)$. But from the assumption that $0 = \phi_1(0) + \psi_1(0)$ and the definition of $\psi_1(0)$, we also know that the preceding inequality is actually an equality if, and only if, $y \in T^*(0)$. This completes the proof of Corollary 5C2.

Corollary 5C2 is important because of its application to "sensitivity analysis." In particular, given the infimum $\phi_1(0)$ for problem $A_1(0)$, it is desirable to be able to estimate $\phi_1(u)$ for u close to 0. Such an estimate can, of course, be based on the directional derivative of ϕ_1 at 0 in the direction u. In particular, the defining equation

$$D_u\phi_1(0) \triangleq \lim_{s\to 0^+} \frac{\phi_1(su) - \phi_1(0)}{s}$$

provides the estimation formula

$$\phi_1(u) \approx \phi_1(0) + D_u\phi_1(0); \qquad (5.1)$$

and, hence, it is of interest to be able to compute $D_u\phi_1(0)$.

Fenchel (1951) has shown (in the more general setting of convex functions) that

$$D_u\phi_1(0) = \max_{y \in \partial\phi_1(0)} \langle u,y\rangle \tag{5.2a}$$

which, incidentally, reduces to the well-known formula $D_u\phi_1(0) = \langle\nabla\phi_1(0),u\rangle$ when $\partial\phi_1(0) = \{\nabla\phi_1(0)\}$. Moreover, under appropriate conditions (such as those provided in Corollary 5C2), this formula can be rewritten as

$$D_u\phi_1(0) = \max_{y \in T^*(0)} \langle u,y\rangle. \tag{5.2b}$$

Thus, it is of interest to know $T^*(0)$ in addition to $\phi_1(0)$ and $S^*(0)$, so that formulas (A5.1) and (A5.2b) can be used to estimate $\phi_1(u)$ for u close to 0. But $T^*(0)$ can usually be calculated from an arbitrary $x^* \in S^*(0)$ by employing the extremality conditions as explained immediately following Corollary 4A2.

6. DUALITY WITH EXPLICIT CONSTRAINTS

The conjugate inequality must be generalized to handle problems with explicit constraints. To make that generalization, recall that the conjugate inequality corresponding to a pair of conjugate functions $g:C$ and $h:D$ is

$$\langle x,y\rangle \leqq g(x) + h(y) \text{ for } x \in C \text{ and } y \in D, \tag{6.1}$$

with equality holding if, and only if, $y \in \partial g(x)$ or, equivalently, $x \in \partial h(y)$.

Given a scalar variable $\lambda > 0$ such that $y \in \lambda D$, we can substitute y/λ for y in the preceding conjugate inequality and then multiply through by λ to establish the nontrivial part of the corresponding <u>geometric inequality</u>

$$\langle x,y\rangle \leqq \lambda g(x) + h^+(y,\lambda) \text{ for } x \in C \text{ and } (y,\lambda) \in D^+, \tag{6.2}$$

where $$D^+ \triangleq \{(y,\lambda) \mid \text{either } \lambda = 0 \text{ and } \sup_{c \in C} \langle c,y\rangle < +\infty, \text{ or } \lambda > 0 \text{ and } y \in \lambda D\}$$

and $$h^+(y,\lambda) \triangleq \begin{cases} \sup_{c \in C} \langle c,y\rangle & \text{if } \lambda = 0 \text{ and } \sup_{c \in C} \langle c,y\rangle < +\infty \\ \lambda h(y/\lambda) & \text{if } \lambda > 0 \text{ and } y \in \lambda D. \end{cases}$$

Of course, the trivial part of this geometric inequality is an immediate consequence of the definition of $h^+(y,\lambda)$ for $\lambda = 0$. Moreover, it is clear that equality holds if, and only if, either $\lambda = 0$ and $\langle x,y\rangle = \sup_{c \in C} \langle c,y\rangle$, or $\lambda > 0$ and $y \in \lambda\partial g(x)$.

Similarly, given a scalar variable $\varkappa > 0$ such that $x \in \varkappa C$, we can substitute $x/\varkappa$ for x in the conjugate inequality and then multiply through by $\varkappa$ to establish the nontrivial part of the corresponding <u>geometric inequality</u>

$$\langle x,y\rangle \leqq g^+(x,\varkappa) + \varkappa h(y) \text{ for } (x,\varkappa) \in C^+ \text{ and } y \in D, \tag{6.3}$$

where $$C^+ \triangleq \{(x,\varkappa) \mid \text{either } \varkappa = 0 \text{ and } \sup_{d \in D} \langle x,d\rangle < +\infty, \text{ or } \varkappa > 0 \text{ and } x \in \varkappa C\}$$

$$\text{and}\quad g^{+}(x,\varkappa) \triangleq \begin{cases} \sup\limits_{d\in D} \langle x,d\rangle & \text{if } \varkappa = 0 \text{ and } \sup\limits_{d\in D} \langle x,d\rangle < +\infty \\ \varkappa g(x/\varkappa) & \text{if } \varkappa > 0 \text{ and } x \in \varkappa C. \end{cases}$$

Of course, the trivial part of this geometric inequality is an immediate consequence of the definition of $g^{+}(x,\varkappa)$ for $\varkappa = 0$. Moreover, it is clear that equality holds if, and only if, either $\varkappa = 0$ and $\langle x,y\rangle = \sup\limits_{d\in D} \langle x,d\rangle$, or $\varkappa > 0$ and $y \in \partial g(x/\varkappa)$.

To keep track of the explicit constraints that will now appear in our family of geometric programming problems, 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 <u>notation and hypotheses</u>:

(Ia) For each $k \in \{0\} \cup I \cup J$, suppose that $g_k\!:\!C_k$ is a closed convex function with a nonempty domain $C_k \subseteq E_{n_k}$.

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

(IIIa) 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

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

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

(Va) 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.

(VIa) Suppose that X is a vector subspace of E_n.

Now consider the following family A_1 of <u>"geometric programming" problems</u> $A_1(u,\mu)$.

<u>Problem $A_1(u,\mu)$.</u> Consider the objective function $G(\cdot + u, \varkappa)\!:\!C(u)$ whose domain

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

and whose functional value

$$G(x + u,\varkappa) \triangleq g\,(x^0 + u^0) + \sum_J g^{+}_j\,(x^j + u^j, \varkappa_j),$$

where, of course,

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

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

Using the feasible solution set

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

calculate both the problem infimum

$$\phi_1(u,\mu) \triangleq \inf_{(x,\varkappa) \in S(u,\mu)} G(x + u, \varkappa)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{(x,\varkappa) \in S(u,\mu) \,|\, G(x + u, \varkappa) = \phi_1(u,\mu)\}.$$

In defining the feasible solution set $S(u,\mu)$, it is important to make a sharp distinction between the <u>vector space condition</u> $x \in X$ and the <u>constraints</u> $g_i(x^i + u^i) + \mu_i \leq 0$, $i \in I$, both of which restrict the vector variable $(x, \varkappa)$. The vector space condition is linear and, hence, can be eliminated by a linear transformation of the vector variable x, but the (generally nonlinear) constraints usually cannot be eliminated by even a nonlinear transformation. Nevertheless, we do not explicitly eliminate the vector space 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 tend only to camouflage one of the extremely useful characteristics possessed by the geometric programming point of view, namely, the separability that is built into problem $A_1(u,\mu)$. We shall eventually illustrate this separability by introducing constraints into the examples of Sec. 2.

Analogous to the unconstrained case discussed in Sec. 2, we shall find it useful to interpret problem $A_1(u,\mu)$ as a perturbed version of $A_1(0,0)$, so we term the set

$$U \triangleq \{(u,\mu) \,|\, S(u,\mu) \text{ is not empty}\}$$

the <u>feasible perturbation set</u> for problem $A_1(0,0)$ relative to the family A_1. Notice that the special perturbations μ appear only with the constraints. It is, of course, obvious that the unconstrained case is obtained by taking the index sets I and J to be the empty set.

Closely related to the family A_1 is its geometric dual family B_1. To obtain B_1 from A_1, we need the following <u>additional notation and hypotheses</u>:

(Ib) For each $k \in \{0\} \cup I \cup J$, let $h_k{:}D_k$ be the conjugate transform of $g_k{:}C_k$. Then $h_k{:}D_k$ is a closed convex function with a nonempty domain $D_k \subseteq E_{n_k}$.

(IIb) For each $k \in \{0\} \cup I \cup J$, let v^k be an independent vector parameter in E_{n_k}, and let ν be an independent vector parameter with components ν_j for each $j \in J$.

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

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

(IVb) For each $k \in \{0\} \cup I \cup J$, let y^k be an independent vector variable in E_{n_k}, and let λ be an independent vector variable with components λ_i for each $i \in I$.

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

(VIb) Let Y be the orthogonal complement of X in E_n. Then Y is, of course, a vector subspace of E_n.

Now, consider the following family B_1 of "geometric programming" problems $B_1(v,\nu)$.

Problem $B_1(v,\nu)$. Consider the objective function $H(\cdot + v, \lambda):D(v)$ whose domain

$$D(v) \triangleq \{(y,\lambda) \mid y^k + v^k \in D_k,\ k \in \{0\} \cup J, \text{ and } (y^i + v^i, \lambda_i) \in D_i^+,\ i \in I\},$$

and whose functional value

$$H(y + v, \lambda) \triangleq h_0(y^0 + v^0) + \sum_I h_i^+ (y^i + v^i, \lambda_i),$$

where, of course,

$$D_i^+ \triangleq \{(z^i, \lambda_i) \mid \text{either } \lambda_i = 0 \text{ and } \sup_{c^i \in C_i} \langle c^i, z^i \rangle < +\infty, \text{ or } \lambda_i > 0 \text{ and } z^i \in \lambda_i D_i\}$$

and
$$h_i^+(z^i, \lambda_i) \triangleq \begin{cases} \sup_{c^i \in C_i} \langle c^i, z^i \rangle & \text{if } \lambda_i = 0 \text{ and } \sup_{c^i \in C_i} \langle c^i, z^i \rangle < +\infty \\ \lambda_i h_i(z^i/\lambda_i) & \text{if } \lambda_i > 0 \text{ and } z^i \in \lambda_i D_i. \end{cases}$$

Using the feasible solution set

$$T(v,\nu) \triangleq \{(y,\lambda) \in D(v) \mid y \in Y, \text{ and } h_j(y^j + v^j) + \nu_j \leq 0,\ j \in J\},$$

calculate both the problem infimum

$$\psi_1(v,\nu) \triangleq \inf_{(y,\lambda) \in T(v,\nu)} H(y + v, \lambda)$$

and the optimal solution set

$$T^*(v,\nu) \triangleq \{(y,\lambda) \in T(v,\nu) \mid H(y + v, \lambda) = \psi_1(v,\nu)\}.$$

Families A_1 and B_1 are clearly of the same type, so the observations made about A_1 are equally valid for B_1. Notice how B_1 is obtained from A_1: the closed convex functions $g_k:C_k$, $k \in \{0\} \cup I \cup J$, are replaced by their respective conjugate transforms $h_k:D_k$, $k \in \{0\} \cup I \cup J$; the vector subspace X is replaced by its orthogonal complement Y; and the roles played by the two index sets I and J are interchanged. Hence, the symmetry in these three operations implies that the family obtained by applying the same transformation to B_1 is again A_1. Because of this symmetry, A_1 and B_1 are termed dual families of geometric programming problems.

Each of the dual families A_1 and B_1 contains a problem of special interest, namely, problems $A_1(0,0)$ and $B_1(0,0)$. Due to the apparent symmetry between them, $A_1(0,0)$ and $B_1(0,0)$ are termed dual problems. To avoid confusion, it is important to bear in mind that problems $A_1(u,\mu)$

and $B_1(v,\nu)$ are termed dual problems only when (u,μ) and (v,ν) are zero.

The symmetry between families A_1 and B_1 induces a symmetry on the theory that relates A_1 to B_1. Thus, each mathematical statement about A_1 and B_1 automatically produces an equally valid dual statement about B_1 and A_1. To be concise, our attention will be focused on the family A_1, and each dual statement will be left to the reader's imagination.

We shall now consider the unperturbed problem $A_1(0,0)$ and its geometric dual problem $B_1(0,0)$. Many of the most important properties of these dual problems are direct consequences of the conjugate and geometric inequalities. In fact we need only make repeated use of the following fundamental lemma that results from those inequalities.

Lemma 6a. *If* $(x,\varkappa)$ *is in the domain*

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

of the objective function $G(\cdot,\varkappa)$ *for problem* $A_1(0,0)$, *and if* (y,λ) *is in the domain*

$$D(0) \triangleq \{(y,\lambda) \mid y^k \in D_k,\ k \in \{0\} \cup J, \text{ and } (y^i,\lambda_i) \in D_i^+,\ i \in I\}$$

of the objective function $H(\cdot,\varkappa)$ *for problem* $B_1(0,0)$, *then*

$$\langle x,y\rangle \le G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda) + \sum_J \varkappa_j h_j(y^j),$$

with equality holding if, and only if,

$$y^0 \in \partial g_0(x^0),$$

$$\text{either}\quad \lambda_i = 0 \text{ and } \langle x^i,y^i\rangle = \sup_{c^i\in C_i} \langle c^i,y^i\rangle,$$

$$\text{or}\quad \lambda_i > 0 \text{ and } y^i \in \lambda_i \partial g_i(x^i),\quad i \in I,$$

$$\text{either}\quad \varkappa_j = 0 \text{ and } \langle x^j,y^j\rangle = \sup_{d^j\in D_j} \langle x^j,d^j\rangle,$$

$$\text{or}\quad \varkappa_j > 0 \text{ and } y^j \in \partial g_j(x^j/\varkappa_j),\quad j \in J.$$

Moreover, if y *also satisfies the constraints*

$$h_j(y^j) \le 0,\quad j \in J,$$

of problem $B_1(0,0)$, *then*

$$G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda) + \sum_J \varkappa_j h_j(y^j)$$

$$\le G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda),$$

with equality holding if, and only if,

$$\varkappa_j h_j(y^j) = 0,\quad j \in J.$$

Furthermore, if x *also satisfies the constraints*

$$g_i(x^i) \le 0,\quad i \in I,$$

of problem $A_1(0,0)$, *then*

$$G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda) \le G(x,\varkappa) + H(y,\lambda),$$

with equality holding if, and only if,

$$\lambda_i g_i(x^i) = 0,\quad i \in I.$$

Proof. From the conjugate inequality (6.1) we know that

$$<x^0,y^0> \leqq g_0(x^0) + h_0(y^0),$$

with equality holding if, and only if,

$$y^0 \in \partial g_0(x^0).$$

From the geometric inequality (6.2) we know that

$$<x^i,y^i> \leqq \lambda_i g_i(x^i) + h_i^+(y^i,\lambda_i),$$

with equality holding if, and only if,

$$\text{either} \quad \lambda_i = 0 \text{ and } <x^i,y^i> = \sup_{c^i \in C_i} <c^i,y^i>$$

$$\text{or} \quad \lambda_i > 0 \text{ and } y^i \in \lambda_i \partial g_i(x^i), \quad i \in I.$$

From the geometric inequality (6.3) we also know that

$$<x^j,y^j> \leqq g_j^+(x^j, \varkappa_j) + \varkappa_j h_j(y^j),$$

with equality holding if, and only if,

$$\text{either} \quad \varkappa_j = 0 \text{ and } <x^j,y^j> = \sup_{d^j \in D_j} <x^j,d^j>,$$

$$\text{or} \quad \varkappa_j > 0 \text{ and } y^j \in \partial g_j(x^j/\varkappa_j), \quad j \in J.$$

Adding all $1 + o(I) + o(J)$ of these inequalities and taking account of the defining equations for x, y, G, and H proves the first assertion of Lemma 6a. The second assertion is an immediate consequence of the fact that $\varkappa_j \geqq 0$ when $(x^j, \varkappa_j) \in C_j^+$, $j \in J$. Similarly, the third assertion is an immediate consequence of the fact that $\lambda_i \geqq 0$ when $(y^i,\lambda_i) \in D_i^+$, $i \in I$. This completes our proof of Lemma 6a.

We now give the most basic and easily proved duality theorem. This theorem generalizes Theorem 4A and hence leads directly to both the duality gap concept and the extremality conditions for problems with explicit constraints.

Theorem 6A. *If $(x, \varkappa)$ and (y,λ) are feasible solutions to problem $A_1(0,0)$ and its geometric dual problem $B_1(0,0)$, respectively, then*

$$0 \leqq G(x, \varkappa) + H(y,\lambda),$$

with equality holding if, and only if,

$$y^0 \in \partial g_0(x^0),$$

$$\text{either} \quad \lambda_i = 0 \text{ and } <x^i,y^i> = \sup_{c^i \in C_i} <c^i,y^i>,$$

$$\text{or} \quad \lambda_i > 0 \text{ and } y^i \in \lambda_i \partial g_i(x^i), \quad i \in I,$$

$$\text{either} \quad \varkappa_j = 0 \text{ and } <x^j,y^j> = \sup_{d^j \in D_j} <x^j,d^j>,$$

$$\text{or} \quad \varkappa_j > 0 \text{ and } y^j \in \partial g_j(x^j/\varkappa_j), \quad j \in J,$$

$$\lambda_i g_i(x^i) = 0,\ i \in I, \text{ and } \varkappa_j h_j(y^j) = 0,\ j \in J.$$

Proof. A sequential application of all three assertions of Lemma 6a shows that

$$\begin{aligned}\langle x,y\rangle &\leqq G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda) + \sum_J \varkappa_j h_j(y^i)\\ &\leqq G(x,\varkappa) + \sum_I \lambda_i g_i(x^i) + H(y,\lambda)\\ &\leqq G(x,\varkappa) + H(y,\lambda).\end{aligned}$$

Thus it is a consequence of Lemma 6a that

$$\langle x,y\rangle \leqq G(x,\varkappa) + H(y,\lambda),$$

with equality holding if, and only if, the final four conditions in the theorem are satisfied. Now x and y are in the orthogonal complementary subspaces X and Y, so $\langle x,y\rangle = 0$ and, hence, our proof of Theorem 6A is complete.

The basic inequality provided by Theorem 6A implies important properties of the dual infima $\phi_1(0,0)$ and $\psi_1(0,0)$. In particular, we obtain the following generalization of Corollary 4A1.

Corollary 6A1. If the geometric dual problems $A_1(0,0)$ and $B_1(0,0)$ are both consistent, then

(i) the infimum $\phi_1(0,0)$ for problem $A_1(0,0)$ is finite, and

$$0 \leqq \phi_1(0,0) + H(y,\lambda)$$

for each feasible solution (y,λ) to problem $B_1(0,0)$;

(ii) the infimum $\psi_1(0,0)$ for problem $B_1(0,0)$ is finite, and

$$0 \leqq \phi_1(0,0) + \psi_1(0,0).$$

The proof of this corollary is, of course, a trivial application of Theorem 6A.

Naturally, consistent dual problems $A_1(0,0)$ and $B_1(0,0)$ for which $0 < \phi_1(0,0) + \psi_1(0,0)$ are said to have a duality gap of $\phi_1(0,0) + \psi_1(0,0)$. The discussion (following Corollary 4A1) of duality gaps for problems without explicit constraints is equally valid for problems with explicit constraints. In fact, dual problems with explicit constraints can be studied within the framework of dual problems without explicit constraints. Only the extremality conditions must be modified to reflect the presence of explicit constraints.

That modification is easily obtained by observing that Theorem 6A provides a characterization of dual optimal solutions $(x^*, \varkappa^*)$ and (y^*,λ^*) in terms of the following extremality conditions:

(I) $\quad x \in X$ and $y \in Y$

(II) $\quad g_i(x^i) \leqq 0,\ i \in I$ and $h_j(y^j) \leqq 0,\ j \in J,$

(III) $\quad y^0 \in \partial g_0(x^0),$

(IV) $\quad$ either $\lambda_i = 0$ and $\langle x^i,y^i\rangle = \sup\limits_{c^i \in C_i} \langle c^i,y^i\rangle,$

or $\lambda_i > 0$ and $y^i \in \lambda_i \partial g_i(x^i), \quad i \in I,$

(V) $\quad$ either $\varkappa_j = 0$ and $\langle x^j,y^j\rangle = \sup\limits_{d^j \in D_j} \langle x^j,d^j\rangle,$

or $\varkappa_j > 0$ and $y^j \in \partial g_j(x^j/\varkappa_j), \quad j \in J,$

(VI) $\quad \lambda_i g_i(x^i) = 0,\ i \in I,$ and $\varkappa_j h_j(y^j) = 0,\ j \in J.$

We formalize this characterization as the following corollary.

Corollary 6A2. Suppose that the geometric dual problems $A_1(0,0)$ and $B_1(0,0)$ are both consistent and $0 = \phi_1(0,0) + \psi_1(0,0)$. Then arbitrary vectors $(x, \varkappa)$ and (y,λ) are optimal solutions to problems $A_1(0,0)$ and $B_1(0,0)$ respectively if, and only if, $(x, \varkappa)$ and (y,λ) satisfy the extremality conditions (I-VI).

The proof of this corollary is an immediate consequence of Theorem 6A and the conjugate transform relation $\partial g_0(x^0) \subseteq D_0$ that was described in Sec. 3.

The extremality conditions (I) are simply the vector space conditions for problems $A_1(0,0)$ and $B_1(0,0)$; and the extremality conditions (II) are simply the constraints for problems $A_1(0,0)$ and $B_1(0,0)$. The extremality conditions (III-V) are termed the subgradient conditions; and the extremality conditions (VI) are, of course, termed the complementary slackness conditions.

Note that the subgradient conditions (III-V) have several equivalent formulations that result from the symmetry of the conjugate transformation. In particular, the condition $y^0 \in \partial g_0(x^0)$ can be replaced by the equivalent condition $x^0 \in \partial h_0(y^0)$; the conditions $y^i \in \lambda_i \partial g_i(x^i)$, $i \in I$, can be replaced by the equivalent conditions $x^i \in \partial h_i(y^i/\lambda_i)$, $i \in I$; and the conditions $y^j \in \partial g_j(x^j/\varkappa_j)$, $j \in J$, can be replaced by the equivalent conditions $x^j \in \varkappa_j \partial h_j(y^j)$, $j \in J$.

In our fundamental Lemma 6a the reader probably noticed the appearance of the "Lagrangians" $G(x, \varkappa) + \sum_I \lambda_i g_i(x^i)$ and $H(y,\lambda) + \sum_J \varkappa_j h_j(y^j)$ for problems $A_1(0,0)$ and $B_1(0,0)$, respectively. Indeed, Lagrangians and their associated "Kuhn-Tucker multipliers" and "saddlepoints" play a fundamental role in geometric programming. We shall now indicate those roles by relating Kuhn-Tucker multipliers to geometric programming.

A consistent problem $A_1(0,0)$ whose infimum $\phi_1(0,0)$ is finite can usually be replaced by a minimization problem with the same infimum $\phi_1(0,0)$, but without the inequality constraints $g_i(x^i) \leq 0$, $i \in I$. The unconstrained minimization problem is obtained by introducing a Kuhn-Tucker multiplier, namely, a vector λ^* in $E_{o(I)}$ with the two properties

$$\lambda_i^* \geqq 0, \quad i \in I,$$

and

$$\phi_1(0,0) = \inf_{\substack{(x,\varkappa) \in C(0) \\ x \in X}} L_A(x, \varkappa;\, \lambda^*),$$

where the Lagrangian L_A has the functional values

$$L_A(x, \varkappa;\, \lambda) \triangleq G(x, \varkappa) + \sum_I \lambda_i g_i(x^i).$$

The following theorem is fundamental in that it relates certain "minimizing sequences" for problem $B_1(0,0)$ to the Kuhn-Tucker multipliers for problem $A_1(0,0)$.

Theorem 6B. Suppose that the geometric dual problems $A_1(0,0)$ and $B_1(0,0)$ are both consistent and $0 = \phi_1(0,0) + \psi_1(0,0)$. If there is a minimizing sequence $\{(y^q,\lambda^q)\}_1^\infty$ for problem $B_1(0,0)$ [that is, each vector (y^q,λ^q) is in the feasible solution set $T(0,0)$, and $\lim_{q\to\infty} H(y^q,\lambda^q) = \psi_1(0,0)$] such that $\lim_{q\to\infty} \lambda^q$ exists, then the limit vector $\lambda^* \triangleq \lim_{q\to\infty} \lambda^q$ is a Kuhn-Tucker multiplier for problem $A_1(0,0)$.

Proof. The assumed feasibility of the vectors (y^q,λ^q) and the defining properties for the sets D_i^+, $i \in I$, imply that $\lambda^q{}_i \geqq 0$, $i \in I$, for each q; so we infer from the hypothesis $\lim_{q\to\infty} \lambda^q = \lambda^*$ that

$$\lambda_i^* \geqq 0, \quad i \in I.$$

The assumed feasibility of the vectors (y^q,λ^q) and a sequential application of the first two assertions of Lemma 6a show that

$$\begin{aligned} \langle x,y^q\rangle &\leqq G(x,\varkappa) + \sum_I \lambda_i^q g_i(x^i) + H(y^q,\lambda^q) + \sum_J \varkappa_j h_j(y^{qj}) \\ &\leqq G(x,\varkappa) + \sum_I \lambda_i^q g_i(x^i) + H(y^q,\lambda^q) \end{aligned}$$

for each vector $(x,\varkappa) \in C(0)$ and for each q. Consequently, for each vector $(x,\varkappa) \in C(0)$ such that $x \in X$ we deduce that

$$0 \leqq G(x,\varkappa) + \sum_I \lambda_i^q g_i(x^i) + H(y^q,\lambda^q)$$

for each q, because the vectors $y^q \in Y \triangleq X^\perp$. This inequality along with the hypotheses $\lim_{q\to\infty} H(y^q,\lambda^q) = \psi_1(0,0)$ and $\lim_{q\to\infty} \lambda^q = \lambda^*$ clearly imply that

$$0 \leqq G(x,\varkappa) + \sum_I \lambda_i^* q_i(x^i) + \psi_1(0,0)$$

for each vector $(x,\varkappa) \in C(0)$ such that $x \in X$. Using the defining formula $L_A(x,\varkappa;\lambda^*) \triangleq G(x,\varkappa) + \sum_I \lambda_i^* g_i(x^i)$ and the hypothesis $0 = \phi_1(0,0) + \psi_1(0,0)$, we infer from the preceding inequality that

$$\phi_1(0,0) \leq \inf_{\substack{(x,\varkappa)\in C(0)\\ x\in X}} L_A(x,\varkappa;\lambda^*).$$

Now choose a minimizing sequence $\{(x^q,\varkappa^q)\}_1^\infty$ for problem $A_1(0,0)$, and then observe for each q that

$$L_A(x^q,\varkappa^q;\lambda^*) \leqq G(x^q,\varkappa^q),$$

because $\lambda_i^* \geqq 0$ and $g_i(x^i) \leqq 0$, $i \in I$. From the construction of the sequence $\{(x^q,\varkappa^q)\}_1^\infty$ we know that the vectors $(x^q,\varkappa^q) \in C(0)$, the vectors $x^q \in X$, and $\phi_1(0,0) = \lim_{q\to\infty} G(x^q,\varkappa^q)$; so we conclude from the preceding two displayed inequalities that

$$\phi_1(0,0) = \inf_{\substack{(x,\varkappa)\in C(0)\\ x\in X}} L_A(x,\varkappa;\lambda^*).$$

This completes our proof of Theorem 6B.

The following corollary to Theorem 6B is important because it shows that each optimal solution to problem $B_1(0,0)$ provides a Kuhn-Tucker multiplier for problem $A_1(0,0)$ when $A_1(0,0)$ and $B_1(0,0)$ are both consistent and do not have a duality gap.

Corollary 6B1. If the geometric dual problems $A_1(0,0)$ and $B_1(0,0)$ are both consistent and $0 = \phi_1(0,0) + \psi_1(0,0)$, then each optimal solution (y^*,λ^*) to problem $B_1(0,0)$ provides a Kuhn-Tucker multiplier λ^* for problem $A_1(0,0)$.

To prove this corollary, simply choose the minimizing sequence $\{(y^q,\lambda^q)\}_1^\infty$ in Theorem 6B so that all its vectors are identical to (y^*,λ^*).

We shall devote the remaining part of this section to various interesting examples.

Example 0. Interestingly, the geometric programming formulation of duality is related to the various formulations of duality in linear programming. We now specialize our geometric programming formulation to obtain the well-known symmetric formulation in linear programming.
To do so, choose the functions $g_k:C_k$, $k \in \{0\} \cup I \cup J$, by letting

$$g_0:E_1 \rightarrow \{0\},$$
$$g_j:\{1\} \rightarrow \{a_j\}, \qquad j \in J,$$
$$g_i:E_1 \rightarrow R \text{ such that } g_i(x^i) \triangleq x^i - b_i, \quad i \in I,$$

where the numbers a_j and b_i are components of given vectors a and b respectively. In addition, choose the vector subspace X to be the column space of a specially structured matrix, namely,

$$X \triangleq \text{column space of } \begin{bmatrix} 0 \\ M \\ U \end{bmatrix},$$

where 0 denotes a row of zeros, M is a given matrix with o(I) rows and o(J) columns, and U is the o(J) × o(J) identity matrix.
Then the resulting specialized family A_1 consists of the following linear programming problems $A_1(u,\mu)$.

Problem $A_1(u,\mu)$. Using the objective function

$$G(x + u, \varkappa) \triangleq \sum_J a_j \varkappa_j$$

and the feasible solution set $S(u,\mu)$ consisting of all those vectors $(x, \varkappa)$ that satisfy both the conditions

(1a) $x^k + u^k \in E_1 \qquad k \in \{0\} \cup I,$

(2a) $\varkappa_j \geqq 0$ and $x^j + u^j = \varkappa_j \qquad j \in J,$

(3a) $$\begin{cases} x^0 = 0 & \\ x^i = \sum_J M_{ij} z_j & i \in I, \\ x^j = z_j & j \in J \end{cases}$$

and the constraints

(4a) $x^i \leqq b_i - (\mu_i + u^i) \qquad i \in I,$

calculate both the problem infimum

$$\phi_1(u,\mu) \triangleq \inf_{(x,\varkappa) \in S(u,\mu)} G(x + u, \varkappa)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{(x, \varkappa) \in S(u,\mu) \mid G(x + u, \varkappa) = \phi_1(u,\mu)\}.$$

The constraints (4a) show that the parameters u^i, $i \in I$, are redundant, so we can set them equal to zero. The conditions (1a) are automatically satisfied, so we can ignore them. The conditions (2a) and (3a) show that the variables x^j and z_j depend on $\varkappa_j$, so we can eliminate x^j and z_j in favor of $\varkappa_j$. Then the constraints (4a) become $\sum_J M_{ij} \varkappa_j \leq b_i - \mu_i + \sum_J M_{ij} u^j$, so the parameters u^j, $j \in J$, are obviously redundant and hence can be set equal to zero. Thus the linear programming family A_1 consists essentially of computing

$$\phi_1'(\mu) \triangleq \inf_{\varkappa \in S'(\mu)} \langle a, \varkappa \rangle$$

where $S'(\mu) \triangleq \{\varkappa \in E_{o(J)} \mid M\varkappa \leqq b - \mu \text{ and } \varkappa \geqq 0\}$.

The computation of the infimum function ϕ_1' is frequently referred to as "parametric linear programming."

The conjugate transforms $h_k{:}D_k$, $k \in \{0\} \cup I \cup J$, of our chosen functions $g_k{:}C_k$, $k \in \{0\} \cup I \cup J$, are clearly obtained by letting

$$h_0{:}\{0\} \to \{0\},$$

$$h_j{:}E_1 \to R \text{ such that } h_j(y^j) \triangleq y^j - a_j, \quad j \in J,$$

$$h_i{:}\{1\} \to \{b_i\}, \qquad i \in I.$$

Moreover, the orthogonal complement Y of our chosen vector subspace X is easily seen to be the column space of a specially structured matrix, namely,

$$Y = \text{column space of } \begin{bmatrix} 1 & 0 \\ 0 & U^{\dagger} \\ 0 & -M^T \end{bmatrix},$$

where $U^{\dagger}$ is the $o(I) \times o(I)$ identity matrix, and $-M^T$ is the negative transpose of M.

Thus, the corresponding specialized family B_1 consists of the following linear programming problems $B_1(v,\nu)$.

<u>Problem $B_1(v,\nu)$.</u> Using the objective function

$$H(y + v,\lambda) \triangleq \sum_I b_i\lambda_i$$

and the feasible solution set $T(v,\nu)$ consisting of all those vectors (y,λ) that satisfy both the conditions

(1b) $y^0 + v^0 = 0$ and $y^j + v^j \in E_1$ $\qquad j \in J,$

(2b) $\lambda_i \geqq 0$ and $y^i + v^i = \lambda_i$ $\qquad i \in I,$

(3b)
$$\begin{cases} y^0 = z_0 & \\ y^i = z_i & i \in I, \\ y^j = -\sum_I M^t{}_{ji} z_i & j \in J \end{cases}$$

and the constraints

(4b) $y^j \leqq a_j - (\nu_j + v^j)$ $\qquad j \in J,$

calculate both the problem infimum

$$\psi_1(v, \ ) \triangleq \inf_{(y,\lambda) \in T(v,\nu)} H(y + v,\lambda)$$

and the optimal solution set

$$T^*(v,\nu) \triangleq \{(y,\lambda) \in T(v,\nu) \mid H(y + v,\lambda) = \psi_1(v,\nu) .$$

The constraints (4b) show that the parameters v^j, $j \in J$, are redundant, so we can set them equal to zero. The variables y^0 and z_0 appear only in the first condition (1b) and the first condition (3b), which show that y^0 and z_0 must be fixed and equal to the parameter $-v^0$; but v^0 appears nowhere else in problem $B_1(v,\nu)$, so y^0, z^0, and v^0 can be ignored. Moreover, the remaining conditions (1b) are automatically satisfied, so we can ignore them. Now the conditions (2b) and (3b) show that the variables y^i and z_i depend on λ_i, so we can eliminate y^i and z_i in favor of λ_i. Then the constraints (4b) become $-\sum_I M_{ij}\lambda_i \leq a_j - \nu_j - \sum_I M_{ij}v^i$, so the parameters v^i, $i \in I$, are obviously redundant and hence can be set equal to zero. Thus, the linear programming family B_1 consists essentially of computing

$$\psi_1'(\nu) \triangleq \inf_{\lambda \in T'(\nu)} \langle b,\lambda\rangle$$

where $\quad T'(\nu) \triangleq \{\lambda \in E_{o(I)} \mid -M^t\lambda \leq a - \nu \text{ and } \lambda \geq 0\}.$

These symmetric linear programming dual families have been thoroughly studied in Peterson (1970a), and nothing new about them will be brought to light here. It is, of course, easy to see that the rather unorthodox dual linear problems $A_1(0,0)$ and $B_1(0,0)$ are equivalent to the classical symmetric dual linear problems: simply observe that problem $A_1(0,0)$ actually consists of minimizing $\langle a,x\rangle$ subject to the constraints $-Mx \geq -b$ and $x \geq 0$, whose classical dual consists of maximizing $\langle -b,y\rangle$, subject to the constraints $-M^ty \leq a$ and $y \geq 0$. It is then clear that the classical dual is equivalent to problem $B_1(0,0)$, because $\max \langle -b,y\rangle = -\min \langle b,y\rangle$.

We shall now specialize our geometric programming formulation of duality to obtain the well-known <u>unsymmetric formulation</u> of duality in linear programming. To do so, choose the functions $g_k:C_k$, $k \in \{0\} \cup I \cup J$, by letting

$$g_0:E_1 \to R \text{ such that } g_0(x^0) \triangleq x^0,$$
$$g_j:E_1 \to \{0\}, \qquad j \in J,$$
$$g_i:E_1 \to R \text{ such that } g_i(x^i) \triangleq x^i - b_i, \quad i \in I,$$

where the numbers b_i are components of a given vector b. In addition, choose the vector subspace X to be the column space of another specially structured matrix, namely,

$$X \triangleq \text{column space of } \begin{bmatrix} a \\ M \\ U \end{bmatrix},$$

where a is a given row vector, M is a given matrix with o(I) rows and o(J) columns, and U is the $o(J) \times o(J)$ identity matrix.

Then, the resulting specialized family A_1 consists of the following linear programming problems $A_1(u,\mu)$.

<u>Problem $A_1(u,\mu)$</u>. Using the objective function

$$G(x + u, \varkappa) \triangleq x^0 + u^0$$

and the feasible solution set $S(u,\mu)$ consisting of all those vectors $(x,\varkappa)$ that satisfy both the conditions

(1a) $\quad x^k + u^k \in E_1 \qquad k \in \{0\} \cup I,$

(2a) $\quad \varkappa_j \geq 0$ and $x^j + u^j \in \varkappa_j E_1 \qquad j \in J,$

$$\text{(3a)} \qquad \begin{cases} x^0 = \sum_J a_j z_j & \\ x^i = \sum_J M_{ij} z_j & i \in I, \\ x^j = z_j & j \in J \end{cases}$$

and the constraints

$$\text{(4a)} \qquad x^i \leqq b_i - (\mu_i + u^i) \qquad\qquad i \in I,$$

calculate both the problem infimum

$$\phi_1(u,\mu) \triangleq \inf_{(x,\varkappa) \in S(u,\mu)} G(x + u, \varkappa)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{(x,\varkappa) \in S(u,\mu) \mid G(x + u, \varkappa) = \phi_1(u,\mu)\}.$$

The constraints (4a) show that the parameters μ_i, $i \in I$, are redundant, so we can set them equal to zero. The conditions (1a) are automatically satisfied, so we can ignore them. The variables $\varkappa_j$, $j \in J$, appear only in the conditions (2a) whose second parts are automatically satisfied when each $\varkappa_j > 0$; hence, the conditions (2a) can be ignored. The conditions (3a) show that the variables x^0 and x^i depend on the variable z_j, so we can eliminate x^0 and x^i in terms of z_j. Then the objective function $x^0 + u^0$ becomes $\langle a,z \rangle + u^0$, and the constraints (4a) become $\sum_J M_{ij} z_j \leqq b_i - u^i$. Thus, the linear programming family A_1 consists essentially of computing

$$\phi_1''(u^0,u^I) \triangleq \inf_{z \in S''(u^I)} \langle a,z \rangle + u^0$$

where $\quad S''(u^I) \triangleq \{z \in E_{o(J)} \mid Mz \leqq b - u^I\}$.

Notice that this family A_1 has the characteristic feature of unsymmetric duality in linear programming; that is, the components z_j of z are not restricted to be nonnegative.

Now the conjugate transforms $h_k{:}D_k$, $k \in \{0\} \cup I \cup J$, of our chosen functions $g_k{:}C_k$, $k \in \{0\} \cup I \cup J$, are clearly obtained by letting

$$h_0{:}\{1\} \to \{0\},$$
$$h_j{:}\{0\} \to \{0\}, \qquad\qquad j \in J,$$
$$h_i{:}\{1\} \to \{b_i\}, \qquad\qquad i \in I.$$

Moreover, the orthogonal complement Y of our chosen vector subspace X is easily seen to be the column space of a specially structured matrix, namely,

$$Y = \text{column space of} \begin{bmatrix} 1 & 0 \\ 0 & U^{\dagger} \\ -a^t & -M^t \end{bmatrix},$$

where $U^{\dagger}$ is the $o(I) \times o(I)$ identity matrix, $-a^t$ is the negative transpose of a, and $-M^t$ is the negative transpose of M.

Thus, the corresponding specialized family B_1 consists of the following linear programming problems $B_1(v,\nu)$.

<u>Problem $B_1(v,\nu)$.</u> Using the objective function

$$H(y + v,\lambda) \triangleq \sum_I b_i\lambda_i$$

and the feasible solution set $T(v,\nu)$ consisting of all those vectors (y,λ) that satisfy both the conditions

$$(1b)\quad y^0 + v^0 = 1 \text{ and } y^j + v^j = 0 \qquad j \in J,$$

$$(2b)\quad \lambda_i \geq 0 \text{ and } y^i + v^i = \lambda_i \qquad i \in I,$$

$$(3b)\quad \begin{cases} y^0 = z_0 & \\ y^i = z_i & i \in I \\ y^j = -a_j z_0 - \sum_I M^T{}_{ji} z_i & j \in J, \end{cases}$$

and the constraints

$$(4b)\quad 0 \leq -\nu_j \qquad j \in J,$$

calculate both the problem infimum

$$\psi_1(v,\nu) \triangleq \inf_{(y,\lambda) \in T(v,\nu)} H(y + v,\lambda)$$

and the optimal solution set

$$T^*(v,\nu) \triangleq \{(y,\lambda) \in T(v,\nu) \mid H(y + v,\lambda) = \psi_1(v,\nu)\}.$$

When the parameters $\nu_j \leq 0$, $j \in J$, the constraints (4b) are automatically satisfied and, hence, can be ignored. The conditions (1b) and the first condition (3b) show that the variables y^0, y^j, and z_0 must be fixed and equal to the parameters $1 - v^0$, $-v^j$, and $1 - v^0$, respectively. The second parts of conditions (2b) and (3b) show that the variables y^i and z_i depend on λ_i, so we can eliminate y^i and z_i in favor of λ_i. Then the last of conditions (3b) become $-\sum_I M^t{}_{ji}\lambda_i = a_j - v^j - a_j v^0 - \sum_I M^t{}_{ji} v^i$, so the parameters v^0 and v^i, $i \in I$, are obviously redundant and hence can be set equal to zero. Thus, the linear programming family B_1 consists essentially of computing

$$\psi_1''(v^J,\nu) \triangleq \inf_{\lambda \in T''(v^J,\nu)} \langle b,\lambda\rangle$$

where

$$T''(v^J,\nu) \triangleq \begin{cases} \emptyset \text{ if } \nu_j > 0 \text{ for at least one } j \in J \\ \{\lambda \in E_{o(I)} \mid -M^t\lambda = a - v^J \text{ and } \lambda \geq 0\} \text{ if } \nu_j \leq 0,\ j \in J. \end{cases}$$

These unsymmetric linear programming dual families have not been studied as thoroughly as the symmetric linear programming dual families. Moreover, the theory given here can be strengthened considerably in this special case, but the details will be omitted here.

In our remaining examples the index set J is assumed to be empty. Then the vector parameter u^J, the vector variables x^J and $\varkappa$, and their corresponding conditions do not appear in problem $A_1(u,\mu)$. Moreover, the vector parameters v^J and ν, the vector variable y^J, and their corresponding constraints do not appear in problem $B_1(v,\nu)$. We shall eventually see that this lack of constraints tends to make problem $B_1(v,\nu)$ more computationally attractive than problem $A_1(u,\mu)$.

Example 0+. We now indicate the generality of geometric programming by showing that "ordinary mathematical programming" can be viewed as a special case.

To obtain all ordinary mathematical programming problems, choose the vector subspace X to be the column space of a specially structured matrix, namely,

$$X \triangleq \text{column space of} \begin{bmatrix} U \\ U \\ \vdots \\ U \end{bmatrix},$$

where the $o(I) \times o(I)$ identity matrix U appears in a total of $1 + o(I)$ positions. Then, the resulting specialized family A_1 consists of the following mathematical programming problems $A_1(u,\mu)$.

Problem $A_1(u,\mu)$. Using the objective function

$$G(x + u) \triangleq g_0(x^0 + u^0)$$

and the feasible solution set $S(u,\mu)$ consisting of all those vectors x that satisfy both the conditions

$$\text{(1a)} \quad x^k + u^k \in C_k \qquad k \in \{0\} \cup I,$$

$$\text{(3a)} \quad x^k = z \qquad k \in \{0\} \cup I,$$

and the constraints

$$\text{(4a)} \quad g_i(x^i + u^i) + \mu_i \leq 0 \qquad i \in I,$$

calculate both the problem infimum

$$\phi_1(u,\mu) \triangleq \inf_{x \in S(u,\mu)} G(x + u)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{x \in S(u,\mu) \,|\, G(x + u) = \phi_1(u,\mu)\}.$$

The conditions (3a) show that we can eliminate the vector variable x in favor of the vector variable z so that the mathematical programming problem $A_1(u,\mu)$ consists essentially of computing

$$\phi_1(u,\mu) \triangleq \inf_{z \in S'(u,\mu)} g_0(z + u^0)$$

where $\quad S'(u,\mu) \triangleq \{z \in \cap \; (C_k - u^k) \,|\, g_i(z + u^i) \leq -\mu_i, \; i \in I\}.$

We term the subfamily that results from choosing the perturbation vectors $u^k = 0$, $k \in \{0\} \cup I$ the ordinary mathematical programming family because its problem $A_1(0,0)$ and its perturbation vector μ have been the focus of most of the past work in mathematical programming.

The orthogonal complement Y of our chosen vector subspace X is easily seen to be the column space of a specially structured matrix, namely,

$$Y = \text{column space of} \begin{bmatrix} -U & -U & \cdots & -U \\ U & 0 & \cdots & 0 \\ 0 & U & & \\ \vdots & & \ddots & \\ 0 & & & U \end{bmatrix},$$

where the $o(I) \times o(I)$ identity matrix U appears in a total of $2\,o(I)$ positions, and the $o(I) \times o(I)$ zero matrix 0 appears in the remaining positions.

Thus the corresponding specialized family B_1 consists of the following unconstrained convex programming problems $B_1(v)$.

Problem $B_1(v)$. Using the objective function

$$H(y+v,\lambda) \triangleq h_0(y^0+v^0) + \sum_I h_i^+(y^i+v^i,\lambda_i)$$

and the feasible solution set $T(v)$ consisting of all those vectors (y,λ) that satisfy the conditions

(1b) $\quad y^0 + v^0 \in D_0,$

(2b) $\quad (y^i + v^i,\lambda_i) \in D_i^+ \qquad i \in I,$

(3b) $\quad \begin{cases} y^0 = -\sum_I w^i & \\ y^i = w^i & i \in I, \end{cases}$

calculate both the problem infimum

$$\psi_1(v) \triangleq \inf_{(y,\lambda)\in T(v)} H(y+v,\lambda)$$

and the optimal solution set

$$T^*(v) \triangleq \{(y,\lambda) \in T(v) \,|\, H(y+v,\lambda) = \psi_1(v)\}.$$

The conditions (3b) show that the convex programming problem $B_1(v)$ consists essentially of computing

$$\psi_1(v) \triangleq \inf_{(y,\lambda)\in T(v)} \{h_0(y^0+v^0) + \sum_I h_i^+(y^i+v^i,\lambda_i)\} \tag{6.4}$$

where

$$T(v) \triangleq \{(y,\lambda)\,|\,y^0+v^0 \in D_0; \; (y^i+v^i,\lambda_i) \in D_i^+,\ i \in I; \text{ and } y^0 + \sum_I y^i = 0\}. \tag{6.5}$$

It is informative to relate the preceding geometric dual problem $B_1(0)$ to the "ordinary dual problem" β, which is defined as follows.

Problem β. Using the objective function

$$\mathcal{H}(\lambda) \triangleq \inf_{x\in C(0)\cap X} L_A(x;\lambda)$$

and the "feasible multiplier" set $\mathcal{J}$ consisting of all those vectors $\lambda \geq 0$ that satisfy the condition

(0a) $\quad \inf_{x\in C(0)\cap X} L_A(x;\lambda)$ is finite,

where the Lagrangian

$$L_A(x;\lambda) \triangleq g_0(x^0) + \sum_I \lambda_i g_i(x^i)$$

and the set $C(0) \cap X$ consists of all those vectors x that satisfy the conditions

(1a) $\quad x^k \in C_k \qquad k \in \{0\} \cup I,$

(3a) $\quad x^k = z \qquad k \in \{0\} \cup I,$

calculate both the problem supremum

$$\Psi \triangleq \sup_{\lambda \in \mathcal{J}} \mathcal{H}(\lambda)$$

and the optimal multiplier set

$$\mathcal{J}^* \triangleq \{\lambda \in \mathcal{J} \mid \mathcal{H}(\lambda) = \Psi\}.$$

The conditions (3a) show that we can eliminate the vector variable x in favor of the vector variable z so that problem β consists essentially of computing

$$\Psi \triangleq \sup_{\lambda \in \mathcal{J}} \inf_{z \in \cap C_k} \{g_0(z) + \sum_I \lambda_i g_i(z)\} \qquad (6.6)$$

where
$$\mathcal{J} \triangleq \{\lambda \geq 0 \mid \inf_{z \in \cap C_k} \{g_0(z) + \sum_I \lambda_i g_i(z)\} \text{ is finite}\}. \qquad (6.7)$$

To tie the geometric dual problem $B_1(0)$ to the ordinary dual problem β, recall from the conjugate inequality (6.1) that

$$\langle z,y^0\rangle \leqq g_0(z) + h_0(y^0) \text{ when } z \in C_0 \text{ and } y^0 \in D_0,$$

with equality holding if $y^0 \in \partial g_0(z)$. Also, for each $i \in I$, recall from the geometric inequality (6.2) that

$$\langle z,y^i\rangle \leqq \lambda_i g_i(z) + h_i^+(y^i,\lambda_i) \text{ when } z \in C_i \text{ and } (y^i,\lambda_i) \in D_i^+,$$

with equality holding if $y^i \in \lambda_i \partial g_i(z)$. Adding these inequalities, we see that

$$0 \leqq \{g_0(z) + \sum_I \lambda_i g_i(z)\} + \{h_0(y^0) + \sum_I h_i^+(y^i,\lambda_i)\} \qquad (6.8a)$$

when
$$z \in \cap C_k;\ y^0 \in D_0;\ (y^i,\lambda_i) \in D_i^+,\ i \in I;$$
$$\text{and } y^0 + \sum_I y^i = 0, \qquad (6.8b)$$

with equality holding if

$$y^0 \in \partial g_0(z) \text{ and } y^i \in \lambda_i \partial g_i(z),\ i \in I. \qquad (6.8c)$$

These relations (6.8) bind the geometric dual problem $B_1(0)$ directly to the (original) Dennis-Dorn-Wolfe version (Dennis, 1959) of the ordinary dual problem β, namely, "the dual problem" β' that consists of computing

$$\Psi' \triangleq \sup_{(z,\lambda) \in Z} \{g_0(z) + \sum_I \lambda_i g_i(z)\},$$

where the feasible solution set

$$Z \triangleq \{(z,\lambda) \mid z \in \cap C_k;\ \lambda_i \geq 0,\ i \in I; \text{ and } 0 \in \partial g_0(z) + \sum_I \lambda_i \partial g_i(z)\}.$$

[Notice that for each $(z,\lambda) \in Z$ the subgradient condition $0 \in \partial g_0(z) + \sum_I \lambda_i \partial g_i(z)$ implies that $\lambda \in \mathcal{J}$ and $\mathcal{H}(\lambda) = g_0(z) + \sum_I \lambda_i g_i(z)$; so each feasible solution (z,λ) to problem β' provides a feasible solution λ to problem β, but the converse is generally not true. Actually, in the original "Wolfe formulation" (1961) of problem β', only differentiable convex functions $g_k:C_k$, $k \in \{0\} \cup I$, were considered; in which case the subgradient condition $0 \in \partial g_0(z) + \sum_I \lambda_i \partial g_i(z)$ reduces to the more familiar gradient condition $0 = \nabla g_0(z) + \sum_I \lambda_i \nabla g_i(z)$. Falk (1967) seems to have been the first person to remove the differentiability restrictions and broaden the definition of the ordinary dual problem by replacing problem β' with problem β.]

In particular, the preceding definition of Z shows that each feasible solution (z,λ) to problem β' produces at least one vector (y,λ) that satisfies both the equation $y^0 + \sum_I y^i = 0$ in condition (6.8b) and all of condition (6.8c). Such a vector (y,λ) must also satisfy the remaining relations in condition (6.8b) by virtue of the definition of D_i^+. Hence, each feasible solution (z,λ) to problem β' produces at least one feasible solution (y,λ) to problem $B_1(0)$ such that inequality (6.8a) is an equality. Consequently, introducing the feasible solution set

$$T_s(\lambda) \triangleq \{y \mid (y,\lambda) \in T(0)\},$$

we deduce from inequality (6.8a) the validity of the identity

$$0 \equiv \{g_0(z) + \sum_I \lambda_i g_i(z)\} + \operatorname*{submin}_{y \in T_s(\lambda)} \{h_0(y^0) + \sum_I h_i^+(y^i,\lambda_i)\}$$

for each $(z,\lambda) \in Z$. This identity shows that the (original) ordinary dual objective function is essentially a subminimized and hence restricted version (in the function-theoretic sense) of the corresponding geometric dual objective function. However, the subminimization and resulting restriction may not be desirable because they tend to suppress the vector variable y and hence conceal important sensitivity analyses. Moreover, unlike the feasible solution set T(0) for the geometric dual problem $B_1(0)$, the feasible solution set Z for the (original) ordinary dual problem β' is generally not convex. However, the (broadened) ordinary dual problem β does not share this disadvantage in that a rather elementary computation (left to the interested reader) shows that its feasible solution set $\mathcal{J}$ is always convex, as is its objective function $\mathcal{H}$. Nevertheless, problem β is clearly a max-min problem and hence difficult to treat numerically.

Detailed discussions of other aspects of ordinary mathematical programming have been given by Falk (1967), Gould (1969, 1971), and Geoffrion (1971). The preceding tie between the ordinary and geometric dual problems is generalized and strengthened in Peterson (1971).

Example 1. We now begin our study of the minimization of posynomials subject to the upper-bound posynomial constraints (i.e., "prototype geometric programming"). To make our notation consistent with the notation already in the literature, let

$$I = \{1, 2, \ldots, p\} \triangleq P.$$

Then, choose the functions $g_k : C_k$, $k \in \{0\} \cup P$, to have the form

$$g(x) \triangleq \ln \left(\sum_i c_i e^{x_i}\right) \text{ where } c_i > 0,$$

and let $\qquad X \triangleq$ column space of (a_{ij}),

where (a_{ij}) is an arbitrary $n \times m$ real matrix. Such functions g are known to be closed and convex even though the function ln is concave. We could proceed without introducing the ln into g (as we did for the unconstrained Example 1 in Sec. 2), but we introduce the ln here so that the geometric dual problem will turn out to be the generalized "chemical equilibrium problem."

The resulting specialized family A_1 consists of the following convex programming problems $A_1(u,\mu)$.

Problem $A_1(u,\mu)$. Using the objective function

$$G(x+u) \triangleq \ln \sum_{[0]} (c_i e^{u_i})\, e^{x_i}$$

and the feasible solution set $S(u,\mu)$ consisting of all those vectors x that satisfy both the conditions

(1a) $\quad x^k + u^k \in E_{n_k}, \qquad\qquad k \in \{0\} \cup P,$

(3a) $\quad x_i = \sum_{j=1}^{m} a_{ij} \ln t_j$ where $t_j > 0, \qquad j = 1, 2, \ldots, m;\ i = 1, 2, \ldots, n$

and the constraints

(4a) $\quad \ln \sum_{[k]} (c_i e^{u_i}) e^{x_i} \leq -\mu_k, \qquad k \in P,$

calculate both the problem infimum

$$\phi_1(u,\mu) \triangleq \inf_{x \in S(u,\mu)} G(x+u)$$

and the optimal solution set

$$S^*(u,\mu) \triangleq \{x \in S(u,\mu) \mid G(x+u) = \phi_1(u,\mu)\}.$$

Here the index set $[k]$, termed "block k," is defined as

$$[k] \triangleq \{m_k, m_k + 1, \ldots, n_k\}, \qquad k \in \{0\} \cup P,$$

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

Using the monotonicity of the ln function along with the laws of exponents, we see that the convex programming problem $A_1(0,0)$ is actually equivalent to the nonconvex programming problem that consists of minimizing the posynomial $\sum_{[0]} c_i \prod_{j=1}^{m} t_j^{a_{ij}}$ subject to the upper-bound posynomial constraints $\sum_{[k]} c_i \prod_{j=1}^{m} t_j^{a_{ij}} \leq 1$, $k \in P$. Moreover, it is obvious from the formula for $\phi_1(u,\mu)$ that the components of u constitute logarithmic perturbations of the coefficients c_i; and it is clear that the components of μ constitute logarithmic perturbations of the constraint upper-bounds 1. Notice that the perturbations μ are redundant in that there is no need to explicitly consider perturbations of the constraint upper-bounds 1; a posynomial constraint with a (positive) upper-bound other than 1 can always be divided by that upper-bound and, hence, can always be replaced by an equivalent posynomial constraint with the upper-bound 1 and appropriately perturbed coefficients $c_i e^{u_i}$.

To obtain the geometric dual B_1 of the preceding family A_1, we now compute the conjugate transform $h{:}D$ of $\ln \sum_i c_i e^{x_i}$. To do so, observe that

$$h(y) = \sup_x \{\langle y,x\rangle - \ln \sum_i c_i e^{x_i}\}$$

is finite only if $y_i \geq 0$ for each i, and $\sum_i y_i = 1$; in which case an elementary maximization via the differential calculus produces the function value

$$h(y) = \sum_i y_i \ln \left(\frac{y_i}{c_i}\right),$$

with the understanding that $y_i \ln \left(\frac{y_i}{c_i}\right) = 0$ when $y_i = 0$. It follows then that the function domain

$$D = \{y \mid y_i \geq 0 \text{ for each } i, \text{ and } \sum_i y_i = 1\}.$$

Thus, the corresponding specialized family B_1 consists of the following linearly constrained convex programming problems $B_1(v)$.

Problem $B_1(v)$. Using the objective function

$$H(y + v,\lambda) = \sum_{[0]} (y_i + v_i) \ln \left(\frac{y_i + v_i}{c_i}\right) + \sum_P \left[\sum_{[k]} (y_i + v_i) \ln \left(\frac{y_i + v_i}{c_i}\right) - \lambda_k \ln \lambda_k\right]$$

and the feasible solution set $T(v)$ consisting of all those vectors (y,λ) that satisfy the conditions

(1b) $y_i + v_i \geq 0$, $i \in [0]$, and $\sum_{[0]} (y_i + v_i) = 1$,

(2b) $y_i + v_i \geq 0$, $i \in [k]$, and $\lambda_k = \sum_{[k]} (y_i + v_i)$, $k \in P$,

(3b) $\sum_{i=1}^{n} a_{ij} y_i = 0$, $j = 1, 2, \ldots, m$,

calculate both the problem infimum

$$\psi_1(v) \triangleq \inf_{(y,\lambda) \in T(v)} H(y + v,\lambda)$$

and the optimal solution set

$$T^*(v) \triangleq \{(y,\lambda) \in T(v) \mid H(y + v,\lambda) = \psi_1(v)\}.$$

In contrast with the "primal program" $A_1(0,0)$, notice that the "dual program" $B_1(0)$ has no constraints and only the following linear conditions: the "positivity conditions" $y_i \geq 0$, $i = 1, 2, \ldots, n$; the "normality condition" $\sum_{[0]} y_i = 1$; the "orthogonality conditions" $\sum_{i=1}^{n} a_{ij} y_i = 0$, $j = 1, 2, \ldots, m$; and the conditions $\lambda_k = \sum_{[k]} y_i$, $k = 1, 2, \ldots, p$, which can, of course, be used to eliminate the explicit use of the vector variable λ.

It is worth mentioning that, when $[0] = \{1\}$, dual program $B_1(0)$ is the "chemical equilibrium problem" that consists of minimizing "Gibbs' free energy function" $H(y,\lambda(y))$ subject to the "mass balance equations" $\sum_{i=2}^{n} a_{ij} y_i = -a_{1j}$, $j = 1, 2, \ldots, m$, to obtain the "equilibrium mole fraction" y^*_i/λ^*_k for each "chemical species" i that can be chemically formed from the m "elements" present in "phase" k of a p-phase "ideal chemical system." This fact and many of its consequences were first established by Passy and Wilde (1968), and have been included in Appendix C of Duffin, Peterson, and Zener (1967). More recently, the corresponding specialized primal program $A_1(0,0)$ has been intimately related to the Darwin-Fowler method in statistical mechanics by Duffin and Zener (1969), who also treat more general chemical systems in Duffin and Zener (1971).

The original geometric dual program formulated by Duffin and Peterson (1966) is essentially the same as dual program $B_1(0)$, and consists of maximizing the objective function

$$\exp\{-H(y,\lambda)\} = \prod_{i=1}^{n} \left(\frac{c_i}{y_i}\right) \prod_{k=1}^{p} \lambda_k^{\lambda_k}$$

subject to the positivity, normality, and orthogonality conditions. The original geometric primal program is, of course, essentially the same as primal program $A_1(0,0)$, and consists of minimizing the objective function

$$\exp\{G(x)\} = \sum_{[0]} c_i \prod_{j=1}^{m} t_j^{a_{ij}}$$

subject to the upper-bound posynomial constraints.

The reader should have no trouble now introducing quadratic and ℓ_p constraints into Example 2 given in Secs. 2 and 3. The resulting classes of "quadratically-constrained quadratic programs" and "ℓ_p-constrained ℓ_p-regression problems" have been thoroughly studied by Peterson and Ecker (1968, 1969, 1970a, 1970b, 1971).

The reader should also have no trouble introducing location constraints into Example 3 given in Secs 2 and 3. The resulting classes of facility location problems are being studied by Wendell and Peterson. The special unconstrained problems that employ only the Euclidean norm have a long history, beginning with the work of Fasbender (1846) and including the more recent work of Kuhn and Kuenne (1962), Kuhn (1967), and Francis and Cabot (1971).

The dynamic programming problems alluded to as Example 4 in Secs. 2 and 3 are being studied by Dinkel and Peterson (1971) who were stimulated by the work of Bellman and Karush (1962, 1963a,b).

The multicommodity network flow problems alluded to as Example 5 in Secs. 2 and 3 are being studied by Morlok and Peterson who expect to apply the results to the analysis and planning of transportation networks. The more specialized single commodity (electrical) network flow problems have a rather long history, beginning with the (Kirchoff-stimulated) work of Weyl (1923) and including the more recent work of Duffin (1947), Bott and Duffin (1951, 1953), Minty (1960, 1966), and Rockafellar (1970). For related work, see Duffin's recent survey papers (1969, 1970).

Although stimulated mainly by Zener's initial work (1961, 1962) on geometric programming, Duffin's initial work (1962a,b) on geometric programming was primarily a mathematical outgrowth of his earlier work on electrical networks and mechanical systems.

This completes our survey of the mathematical foundations of geometric programming.

7. POSYNOMIAL PROGRAMMING ALGORITHMS

A posynomial program $A_1(0,0)$ and its geometric dual program $B_1(0)$ are said to be "degenerate" if at least one of the posynomial terms $c_i \prod_{j=1}^{m} t_j^{a_{ij}}$

can approach zero without causing any of the other posynomial terms to approach plus infinity. Programs $A_1(0,0)$ and $B_1(0)$ that are not degenerate are said to be "canonical," because Sec. VI.5 of Duffin, Peterson, and Zener (1967) "reduces" the study of degenerate programs to the study of equivalent canonical programs. Actually, it seems that all posynomial programs arising in technological design are canonical to begin with; but in any event we need only restrict our attention to canonical programs.

For canonical dual programs $B_1(0)$, Sec. VI.4 of the same reference shows that a linear programming algorithm can be used to construct dual feasible solutions with strictly positive components (which, incidently, do not exist for degenerate dual programs). Such dual feasible solutions are numerically desirable because the dual objective function V defined by the formula

$$V(y) \triangleq - H(y,\lambda(y))$$

is differentiable (in fact, "analytic") only at points with strictly positive components. Moreover, the set of all strictly positive dual feasible solutions is clearly the (relative) interior of the dual feasible solution set, and it is not difficult to show that V is continuous on the (relative) boundary of that set.

Hence, any of the efficient numerical algorithms for maximizing an analytic concave function subject to linear constraints can be used to generate a strictly positive maximizing sequence $\{y^q\}_0^\infty$ from a given strictly positive dual feasible solution y^0 (that can be determined by linear programming). Such a maximizing sequence may, of course, converge to an optimal solution that does not have strictly positive components.

These maximizing sequences can be terminated by the algorithm stopping criterion discussed in Sec. 4 because Sec. VI.4 of Duffin, Peterson, and Zener (1967) proves that canonical programs do not have duality gaps. But such procedures require the simultaneous generation of a minimizing sequence for program $A_1(0,0)$, presumably by an appropriate numerical algorithm for minimizing convex functions subject to nonlinear convex constraints. Such algorithms are known to be relatively slow and sometimes unreliable, so alternative procedures are desirable.

One alternative procedure employs the extremality conditions (Editor's note: The extremality conditions are stated explicitly in Sec. 6 of this paper.) corresponding to programs A_1 (0,0) and $B_1(0)$, namely:

$$\text{(I)} \qquad x \in X \qquad \text{and} \qquad y \in Y,$$

$$\text{(II)} \qquad \sum_{[k]} c_j e^{x_j} \leq 1 , \qquad k \in P,$$

$$\text{(III)} \qquad y_i \sum_{[0]} c_j e^{x_j} = c\ e^{x_i} , \qquad i \in [0],$$

$$\text{(IV)} \qquad y_i = \frac{\lambda_k(y) c_i e^{x_i}}{\sum_{[k]} c_j e^{x_j}} , \qquad i \in [k],\ k \in P,$$

$$\text{(VI)} \qquad \Big(\sum_{[k]} c_j e^{x_j}\Big)^{\lambda_k(y)} = 1 , \qquad k \in P.$$

Of course, some of the extremality conditions, such as (V), do not appear because the index set J is empty, and because the $\sup_{c_k \in E_{n_k}} \langle c^k, y^k \rangle$ is finite if, and only if, $y^k = 0$. Furthermore, by using the equation $\lambda_k(y) = \sum_{[k]} y_i$, it is not difficult to show that conditions (IV) and (VI) are equivalent to the conditions

$$\text{(IV')} \qquad y_i = \lambda_k(y) c_i e^{x_i}, \qquad i \in [k],\ k \in P,$$

whose validity when $\lambda_k(y) > 0$ clearly implies the validity of the corresponding condition (II) as an equality (i.e., the corresponding primal constraint is "active").

Now, Sec. VI.4 of Duffin, Peterson, and Zener (1967) guarantees the existence of an optimal solution x^* to a consistent canonical program $A_1(0,0)$; and, when $A_1(0,0)$ has a Kuhn-Tucker multiplier [which is, of course, the case when $A_1(0,0)$ is super-consistent], Sec. VI.4 also guarantees the existence of an optimal solution y^* to the corresponding dual program $B_1(0)$. Such optimal solutions x^* and y^* must satisfy the extremality conditions (I-III), conditions (IV') and the objective function condition

$$\text{(V')} \qquad \log \sum_{[0]} c_j e^{x_j} = V(y) ,$$

because of the lack of duality gaps. In particular, observe that x^* is immediately determined from y^* when y^* has strictly positive components, by simply taking the logarithm of conditions (III) and (IV') while taking account of condition (V'). If y^* has a zero component, say, $y^*_i = 0$ for some $i \in [k]$, then the corresponding condition (IV') implies that $\lambda_k(y^*) = 0$ and hence that $y^*_i = 0$ for each $i \in [k]$; in which case, x^*_i for each $i \in [k]$ cannot be determined by taking the logarithm of conditions (IV') [although such unknown components x^*_i can sometimes be determined from the known components of x^* by using extremality condition (I)].

Generally, we do not have a dual optimal solution y^* at hand, so we must work only with a strictly positive maximizing sequence $\{y^q\}_0^\infty$ such that $y^q \to y^*$. Using such a sequence $\{y^q\}_0^\infty$ and the (obviously) continuous extremality conditions (III), (IV'), and (V'), we can generate an image sequence $\{x^q\}_0^\infty$ such that $x^q \to x^*$ when y^* has strictly positive components. Of course, the vectors x^q may not be feasible solutions to program $A_1(0,0)$; so the algorithm stopping criterion must also include tolerance limits for the primal constraints, each of which is active when y^* has strictly positive components.

The preceding algorithm has been quite successful only at solving those programs $A_1(0,0)$ and $B_1(0)$ that have active primal constraints, because the resulting positivity of the Kuhn-Tucker multipliers λ^*_k implies that the optimal solutions y^* to program $B_1(0)$ are strictly positive. However, Kochenberger (1969) has discovered how to extend this algorithm to those programs $A_1(0,0)$ and $B_1(0)$ that have inactive primal constraints, by essentially taking up the slack in such constraints.

To do so, he adds "slack variables" T_k to the posynomials that appear in potentially inactive constraints; and then he adds their reciprocals bT_k^{-1} to the posynomial objective function. The coefficient parameter b is chosen to be positive so that the positive variables T_k will tend to become as large as possible and hence make the corresponding constraints active. Consequently, this "augmented" posynomial program can be solved by the preceding algorithm for each choice of the positive parameter b. In particular, Kochenberger and Woolsey (in private communication) state that when b is very close to zero, the resulting optimal solutions are very good approximations to the (desired) optimal solutions to the original programs $A_1(0,0)$ and $B_1(0)$. More recently, Duffin and Peterson (1971d) have shown that these approximations can, in fact, be made arbitrarily accurate by choosing b sufficiently close to zero. In addition, they have used this device to give a new and somewhat simpler proof of the fact that canonical programs do not have duality gaps.

Since the primal constraints in the augmented program are known to be active, they can clearly be replaced by the corresponding equality constraints without changing the augmented optimal solutions. These equality constraints can then be solved for T_k and hence eliminated by substituting the resulting expression for T_k into the augmented objective function. This substitution produces an equivalent unconstrained program that is not a posynomial program but has the same objective function used in the "penalty function" methods of Carroll, Fiacco, and McCormick (1969).

Clearly, there are many posynomials like bT_k^{-1} that produce active constraints when added to the original posynomial objective function; the only requirement is that such posynomials are not themselves minimized by any $T_k < 1$. Each posynomial corresponds to a different penalty function, and each penalty function is known to produce a numerical method [Fiacco and McCormick (1969)] for solving the posynomial program $A_1(0,0)$ directly. Perhaps a hybrid of the purely penalty function method and Kochenberger's method would be most effective. Such a hybrid method could conceivably exploit the fact that a primal constraint is inactive when its corresponding dual positivity conditions are active.

Actually, it is worth mentioning that Kochenberger and Woolsey no longer use the posynomial bT_k^{-1} because their experience indicates that it is numerically better to introduce an additional positive parameter r and use the posynomial $bT_k^{r} + bT_k^{-r}$. In particular, they have obtained sufficiently accurate approximate optimal solutions to a number of programs of practical significance by choosing $b = r = 0.01$.

Finally, we should mention that these and other techniques have been coded and studied by Thomas Jefferson who intends to use them to solve the "northeast corridor transportation problem" as part of his Ph.D. thesis at Northwestern University. His computer software can, of course, be used to solve any posynomial program (and, hence, any algebraic program by virtue of the developments described in the next section). A very comprehensive software package, including a postoptimal sensitivity analysis for the coefficients , is now commercially available.

8. SIGNOMIAL PROGRAMS TREATED BY GEOMETRIC AND HARMONIC MEANS

Although geometric programming with posynomials provides a powerful method for studying many optimization problems in technological design, many other important optimization problems can be modeled accurately only by using more general types of algebraic functions. Hence, the question of extending the applicability of geometric programming to those larger classes of programs has received considerable attention.

In particular, Sec. III.4 of Duffin, Peterson, and Zener (1967) presents various techniques for transforming a limited class of algebraic programs into equivalent prototype geometric programs, but many of the most important optimization problems are not within that limited class.

Initial attempts at rectifying this situation were made by Passy and Wilde (1967). They generalized some of the prototype concepts and theorems in order to treat "signomial programs"; but most of the important prototype theorems are not valid in that more general setting. Nevertheless, their work was subsequently advanced by Duffin and Peterson (1971a) in such a way that those difficulties have been at least partially overcome, even in the still more general setting of algebraic programs. By employing the well-known elementary transformations from mathematical programming and by using rather obvious extensions of the transformations given in Sec. III.4 of Duffin, Peterson, and Zener (1967), each well-posed algebraic program can be transformed into an equivalent signomial program — and, hence, ultimately into an equivalent posynomial program by exploiting the transformations to be described here. Due to the inherent difficulty in giving a general analytical description of the class of algebraic programs, we only illustrate their transformation into equivalent signomial programs with an example in Sec. 5. Here we confine our attention to the more easily described, but much smaller, class of signomial programs.

A signomial is a (generalized) polynomial

$$f(t_1, t_2, \ldots, t_m) \triangleq \sum_{i=1}^{N} c_i \, t_1^{a_{i1}} \, t_2^{a_{i2}} \ldots t_m^{a_{im}}$$

(with arbitrary real exponents a_{ij}) whose independent variables t_j are all restricted to be positive. It is convenient to arrange the terms of a signomial $f(t)$ so that those with positive coefficients c_i (if any) appear first in the summation. Then each signomial $f(t)$ is seen to be either a posynomial (i.e., all coefficients c_i are positive), the negative of a posynomial, or the difference of two posynomials.

By using the well-known elementary transformations employed in mathematical programming, we can easily transform each signomial program into an equivalent signomial program in which a signomial is to be minimized, subject only to upper-bound inequality signomial constraints. Moreover, each of the resulting constraints can be formulated in one of three forms:

$$f(t) \leqq -1 \ , \quad f(t) \leqq 0 \ , \quad f(t) \leqq 1 \ . \tag{8.1}$$

We now show how to transform each of these signomial programs into an equivalent posynomial program in which a posynomial is to be minimized, subject only to inequality posynomial constraints having one of these forms:

$$g(t) \leqq 1 \ , \quad g(t) \geqq 1 \ . \tag{8.2}$$

Unless the objective function is already a posynomial, we first transform it by introducing a new positive independent variable t_0. To see how this is done, suppose we wish to minimize a signomial $f_0(t)$, subject to inequality signomial constraints. The transformation to be used depends on the sign of the constrained infimum of $f_0(t)$. If this sign is not negative, we should minimize the positive independent variable t_0, subject to the original constraints and the additional constraint $f_0(t) \leqq t_0$; in which case the constrained infimum of t_0 clearly gives the constrained infimum of $f_0(t)$. If the constrained infimum of $f_0(t)$ is negative, we should maximize t_0, subject to the original constraints and the additional constraint $f_0(t) + t_0 \leqq 0$; in which case the negative of the constrained supremum of t_0 clearly gives the constrained infimum of $f_0(t)$. Now, maximizing t_0 can obviously be accomplished by minimizing t_0^{-1}; so in all cases we are left with an equivalent program that consists of minimizing a posynomial, subject only to inequality signomial constraints.

Of course, when the sign of the constrained infimum of $f_0(t)$ is not known in advance, we should probably make an educated guess at the appropriate sign and, hence, the appropriate transformation. If the first transformation is chosen and the resulting infimum turns out to be zero, then the second transformation should also be tried in order to see whether the desired infimum is actually less than zero. If the second transformation is chosen and the resulting program turns out to be inconsistent, then the first transformation should also be tried in order to see whether the original program is actually inconsistent or just has a nonnegative infimum. In any event, clearly the additional signomial constraint can be formulated in at least two of the three forms (2.1).

The additional transformations required to obtain an equivalent posynomial program are most easily described within the context of a special case in which there are only three signomial constraints, each representing one of the three possible forms (8.1). Thus, suppose we wish to minimize a posynomial $g_0(t)$ subject to the signomial constraints

$$f_1(t) \leqq -1 , \quad f_2(t) \leqq 0 , \quad f_3(t) \leqq 1 .$$

If $f_1(t)$ is a posynomial, the constraint $f_1(t) \leqq -1$ clearly cannot be satisfied, so the program is inconsistent. If $f_1(t)$ is the negative of a posynomial, this constraint is equivalent to the posynomial constraint $-f_1(t) \geqq 1$, which already has the second of the desired forms (2.2). Hence, we need to give further consideration only to the case in which $f_1(t)$ is the difference of two posynomials.

If $f_2(t)$ is a posynomial, the constraint $f_2(t) \leqq 0$ clearly cannot be satisfied, so the program is inconsistent. If $f_2(t)$ is the negative of a posynomial, this constraint is automatically satisfied and, therefore, can be ignored. Hence, we need to give further consideration only to the case in which $f_2(t)$ is the difference of two posynomials.

If $f_3(t)$ is a posynomial, the constraint $f_3(t) \leqq 1$ is already a posynomial constraint that has the first of the desired forms (2.2). If $f_3(t)$ is the negative of a posynomial, this constraint is automatically satisfied and, therefore, can be ignored. Hence, we need to give further consideration only to the case in which $f_3(t)$ is the difference of two posynomials.

Thus, suppose we wish to minimize a posynomial $g_0(t)$, subject to the constraints

$$\begin{aligned} h_1(t) - h_4(t) &\leqq -1 , \\ h_2(t) - h_5(t) &\leqq \ \ 0 , \\ h_3(t) - h_6(t) &\leqq \ \ 1 , \end{aligned}$$

where the $h_k(t)$, $k = 1, 2, \ldots, 6$, are posynomials and $t = (t_1, t_2, \ldots t_m)$. Introducing three new positive independent variables t_{m+1}, t_{m+2}, and t_{m+3}, we see that t is a feasible solution to these constraints if, and only if, there are positive values for t_{m+1}, t_{m+2}, and t_{m+3} such that the augmented vector $(t, t_{m+1}, t_{m+2}, t_{m+3})$ is a feasible solution to the constraints

$$\begin{aligned} 1 + h_1(t) \leqq\ & t_{m+1} \leqq h_4(t) , \\ h_2(t) \leqq\ & t_{m+2} \leqq h_5(t) , \\ h_3(t) \leqq\ & t_{m+3} \leqq h_6(t) + 1 . \end{aligned}$$

But these constraints are clearly equivalent to the constraints

$$g_k(t,\ t_{m+1},\ t_{m+2},\ t_{m+3}) \leqq 1\ , \qquad k = 1,\ 2,\ 3,$$
$$g_k(t,\ t_{m+1},\ t_{m+2},\ t_{m+3}) \geqq 1\ , \qquad k = 4,\ 5,\ 6,$$

where

$$g_k(t,\ t_{m+1},\ t_{m+2},\ t_{m+3}) \triangleq \begin{cases} t_{m+k}^{-1}\ [1 + h_k(t)] & k = 1 \\ t_{m+k}^{-1}\ h_k(t) & k = 2,\ 3 \\ t_{m+(k-3)}^{-1}\ h_k(t) & k = 4,\ 5 \\ t_{m+(k-3)}^{-1}\ [h_k(t) + 1] & k = 6. \end{cases}$$

Moreover, it is obvious that these functions $g_k(t,\ t_{m+1},\ t_{m+2},\ t_{m+3})$ are posynomials and that each of the preceding six constraints has one of the two desired forms (8.2).

It is now apparent that each signomial program can easily be transformed into an equivalent posynomial program in which a posynomial $g_0(t)$ is to be minimized, subject only to inequality posynomial constraints having one of the two forms (8.2). Hence, there is no loss of generality in restricting our attention to this special class of posynomial programs, so we make this simplifying restriction in the following developments.

Such posynomial programs have been termed "reversed geometric programs" because some of their inequality posynomial constraints have a direction $g(t) \geqq 1$ that is the reverse of the direction $g(t) \leqq 1$ required for prototype geometric programs.

The most general reversed geometric program is now stated for future reference as:

Primal Program A. Find the infimum M_A of a posynomial $g_0(t)$ subject to the posynomial constraints

$$g_k(t) \leqq 1\ , \qquad k = 1,\ 2,\ \ldots,\ p,$$

and $$g_k(t) \geqq 1\ , \qquad k = p+1,\ \ldots,\ p+r \triangleq q.$$

Here, $$g_k(t) \triangleq \sum_{[k]} u_i(t)\ , \qquad k = 0,\ 1,\ \ldots,\ q,$$

and $$u_i(t) \triangleq \begin{cases} c_i t_1^{a_{i1}}\ t_2^{a_{i2}} \ldots\ t_m^{a_{im}},\ i \in [k], & k = 0,\ 1,\ \ldots,\ p, \\ c_i t_1^{-a_{i1}}\ t_2^{-a_{i2}} \ldots\ t_m^{-a_{im}},\ i \in [k], & k = p+1,\ \ldots,\ q, \end{cases}$$

where $$[k] \triangleq \{m_k,\ m_k + 1,\ \ldots,\ n_k\}\ , \qquad k = 0,\ 1,\ \ldots,\ q,$$

and $$1 \triangleq m_0 \leq n_0,\quad n_0 + 1 \triangleq m_1 \leq n_1,\quad \ldots,\quad n_{q-1} + 1 \triangleq m_q \leq n_q \triangleq n.$$

The exponents a_{ij} and $-a_{ij}$ are arbitrary real numbers, but the coefficients c_i and the independent variables t_j are assumed to be positive.

We have placed minus signs in the exponents for the reversed constraint terms in order to obtain a notational simplification in the ensuing developments.

To provide other notational simplifications, we use the index sets

$$P \triangleq \{1, 2, \ldots, p\},$$

$$R \triangleq \{p+1, \ldots, q\},$$

and $$[K] \triangleq \bigcup_{k \in K} [k] \qquad \text{for each } K \subseteq \{0\} \cup P \cup R.$$

For purposes requiring pronunciation, $[K]$ is called "block K."

In terms of the preceding symbols, primal program A consists of minimizing the "primal objective function," $g_0(t)$ subject to the prototype "primal constraints" $g_k(t) \leq 1$, $k \in P$, and subject to the reversed primal constraints $g_k(t) \geq 1$, $k \in R$, where: the posynomial $g_k(t) \triangleq \sum_{[k]} u_i(t)$ for each $k \in \{0\} \cup P \cup R$; the posynomial term $u_i(t) \triangleq c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}$ for each $i \in [0] \cup [P]$; and the posynomial term $u_i(t) \triangleq c_i t_1^{-a_{i1}} t_2^{-a_{i2}} \ldots t_m^{-a_{im}}$ for each $i \in [R]$.

As in prototype geometric programming, each posynomial term $u_i(t)$ in primal program A gives rise to an independent "dual variable" y_i, $i \in [0] \cup [P] \cup [R]$; and each posynomial $g_k(t)$ gives rise to a dependent dual variable $\lambda_k(y) \triangleq \sum_{[k]} y_i$, $k \in \{0\} \cup P \cup R$. To define the "geometric dual" of primal program A, it is convenient to extend the notation of the preceding paragraph by introducing the symbols

$$K(y) \triangleq \{k \in K \mid \lambda_k(y) \neq 0\} \qquad \text{for each } K \subseteq \{0\} \cup P \cup R,$$

and $$[K](y) \triangleq \{i \in [K] \mid y_i \neq 0\} \qquad \text{for each } K \subseteq \{0\} \cup P \cup R.$$

Then, corresponding to primal program A is the following <u>geometric dual program</u>:

<u>Dual Program B</u>. Find the supremum M_B of the dual objective function

$$v(y) \triangleq \left\{\left[\prod_{[0](y)} \left(\frac{c_i}{y_i}\right)^{y_i}\right]\left[\prod_{[P](y)} \left(\frac{c_i}{y_i}\right)^{y_i}\right]\left[\prod_{[R](y)} \left(\frac{c_i}{y_i}\right)^{-y_i}\right]\right\}$$

$$\times \left\{\left[\prod_{P(y)} \lambda_k(y)^{\lambda_k(y)}\right]\left[\prod_{R(y)} \lambda_k(y)^{-\lambda_k(y)}\right]\right\}$$

subject to the positivity conditions

$$y_i \geqq 0\ , \qquad i \in \{1, 2, \ldots, n\} = [0] \cup [P] \cup [R],$$

the normality condition

$$\lambda_0(y) = 1\ ,$$

and the orthogonality conditions

$$\sum_{i=1}^{n} a_{ij} y_i = 0\ , \qquad j = 1, 2, \ldots, m.$$

Here $\lambda_k(y) \triangleq \sum_{[k]} y_i$, $\qquad k \in \{0, 1, \ldots, q\} = \{0\} \cup P \cup R,$

and the numbers a_{ij} and c_i are as given in primal program A.

The dual constraints are identical to their analogs in prototype geometric programming; and they are linear, so the dual feasible solution set is either empty or polyhedral and convex. The dual objective function differs from its analog only by the presence of minus signs in the exponents of the factors corresponding to the reversed primal constraints; but those minus signs result in very large theoretical and computational differences between reversed and prototype geometric programming.

In particular, the logarithm of the dual objective function v is no longer concave, because the minus signs make the terms corresponding to the reversed primal constraints convex. Consequently, unlike prototype geometric programming, reversed geometric programming is not essentially a branch of convex programming. Moreover, this lack of (total) convexity will force us to be content with studying "equilibrium solutions" that need not always be optimal.

Thus, the preceding remarks and the extremality conditions (I-III) and (IV') (given in Sec. 3) for prototype geometric programs help to motivate the following definition:

Definition. A feasible solution t^* to primal program A is termed a primal equilibrium solution if there is a feasible solution y^* to dual program B such that

$$y^*_i g_0(t^*) = u_i(t^*) , \qquad i \in [0],$$

and $$y^*_i = \lambda_k(y^*) u_i(t^*) , \qquad i \in [k] \qquad k \in P \cup R;$$

in which case y^* is termed a dual equilibrium solution. Given corresponding primal and dual equilibrium solutions t^* and y^*, the numbers $E_A \triangleq g_0(t^*)$ and $E_B \triangleq v(y^*)$ are said to be corresponding primal and dual equilibrium values.

In view of our previous discussion of prototype geometric programs, we might guess that equilibrium solutions are intimately related to the Lagrangian for primal program A. Actually, those relations serve as a convenient vehicle for establishing two illuminating facts that indicate the practical relevance of equilibrium solutions: first, the set of all equilibrium solutions to primal program A is identical to the set of all those primal feasible solutions that are "tangentially optimal" in a certain weakly global sense; and, second, almost every "locally optimal" solution to primal program A is also a primal equilibrium solution. These and other facts are established and further discussed in Duffin and Peterson (1971a). In particular, corresponding primal and dual equilibrium values are shown to be equal, and dual equilibrium solutions are characterized as the solutions to certain "equilibrium equations." The equilibrium equations generalize the "mass action laws" for the chemical equilibrium problem, and they also provide "indirect methods" for computing equilibrium solutions.

The study of each reversed geometric program will now be reduced to the study of either of two different corresponding families of approximating prototype geometric programs. This reduction is based on the classical inequalities relating the arithmetic, geometric, and harmonic means [Hardy, Littlewood, and Polya (1959)]. For our purposes, it is convenient to state those inequalities in somewhat disguised form as the following lemma.

Lemma 8a. *If $u_1, \ldots, u_N$ are positive quantities, and if $\alpha_1, \ldots, \alpha_N$ are positive numbers such that*

$$\sum_{i=1}^{N} \alpha_i = 1 ,$$

then $$\left(\sum_{i=1}^{N} u_i\right)^{-1} \leqq \prod_{i=1}^{N} \left(\frac{\alpha_i}{u_i}\right)^{\alpha_i} \leqq \sum_{i=1}^{N} \left(\frac{\alpha_i^{\,2}}{u_i}\right) .$$

Moreover, these inequalities are strict unless

$$u_i = \alpha_i \left(\sum_{j=1}^{N} u_i\right) , \qquad i = 1, \ldots, N,$$

in which case they are equalities.

Proof. Given positive quantities $T_1, \ldots, T_N$ and the positive "weights" $\alpha_1, \ldots, \alpha_N$, Cauchy's arithmetic-geometric mean inequality asserts that

$$\left(\sum_{i=1}^{N} \alpha_i T_i\right) \geqq \prod_{i=1}^{N} (T_i)^{\alpha_i} \tag{8.3}$$

with equality holding if and only if

$$T_i = \sum_{j=1}^{N} \alpha_j T_j , \qquad i = 1, \ldots, N. \tag{8.4}$$

(Actually, this inequality is equivalent to the conjugate inequality when $g(x) \triangleq \log \sum_{i=1}^{N} e^{x_i}$.) Replacing the positive quantities T_i with their positive reciprocals T_i^{-1} gives the classical geometric-harmonic mean inequality

$$\prod_{i=1}^{N} (T_i)^{\alpha_i} \geqq \left(\sum_{i=1}^{N} \alpha_i T_i^{-1}\right)^{-1} , \tag{8.5}$$

with equality holding if, and only if,

$$T_i^{-1} = \sum_{j=1}^{N} \alpha_j T_j^{-1} , \qquad i = 1, \ldots, N. \tag{8.6}$$

Moreover, it is easily seen that the normalization $\sum_{i=1}^{N} \alpha_i = 1$ implies the equivalence of the equality conditions (8.4) and (8.6). Now, choose $T_i = u_i/\alpha_i$ for $i = 1, \ldots, N$, and invert each of the inequalities resulting from (8.3) and (8.5) to complete our proof of Lemma 8a.

Given a posynomial

$$g(t) \triangleq \sum_{i=1}^{N} u_i(t)$$

and positive weights $\alpha_1, \ldots, \alpha_N$, the corresponding *geometric inverse* $g'(\cdot;\alpha)$ of g, and the corresponding *harmonic inverse* $g''(\cdot;\alpha)$ of g, are posynomials defined by the following formulas:

$$g'(t;\alpha) \triangleq \prod_{i=1}^{N} \left[\frac{\alpha_i}{u_i(t)}\right]^{\alpha_i}$$

and $$g''(t;\alpha) \triangleq \sum_{i=1}^{N} \frac{\alpha_i{}^2}{u_i(t)} .$$

Then, Lemma 8a shows that

$$1/g(t) \leqq g'(t;\alpha) \leqq g''(t;\alpha)$$

for each $t > 0$, so we have the following implications:

$$g'(t;\alpha) \leqq 1 \Rightarrow g(t) \geqq 1 , \tag{8.7}$$

and $$g''(t;\alpha) \leqq 1 \Rightarrow g'(t;\alpha) \leqq 1 \Rightarrow g(t) \geqq 1 . \tag{8.8}$$

Given a reversed geometric program A, the implication (8.7) suggests the introduction of a condensed program $A'(\alpha)$ in which the reversed inequality constraints $g(t) \geqq 1$ are replaced by the corresponding prototype inequality constraints $g'(t;\alpha) \leqq 1$. Then the resulting condensed program $A'(\alpha)$ is a prototype geometric program, and the implication (2.7) shows that the infima M_A and $M_{A'(\alpha)}$ for programs A and $A'(\alpha)$, respectively, satisfy the inequality

$$M_{A'(\alpha)} \geqq M_A .$$

A detailed analysis of the family of all condensed programs $A'(\alpha)$ was, in essence, first given by Avriel and Williams (1970), although similar analyses for somewhat smaller classes of programs were independently made by Broverman, Federowicz, and McWhirter (personal communication), Pascual and Ben-Israel (1970a), and Passy (1971). Actually, the condensation process can be further exploited to reduce the study of each reversed geometric program to the study of an infinite family of approximating linear programs. In particular, that process combined with the duality theory of linear programming provides another alternative proof [Duffin (1970)] of the fact that canonical prototype programs do not have duality gaps.

Given a reversed geometric program A, the implication (8.8) suggests the introduction of a harmonized program $A''(\alpha)$ in which the reversed inequality constraints $g(t) \geqq 1$ are replaced by the corresponding prototype inequality constraints $g''(t;\alpha) \leqq 1$. Then the resulting harmonized program $A''(\alpha)$ is a prototype geometric program, and the implication (2.8) shows that

$$M_{A''(\alpha)} \geqq M_{A'(\alpha)} \geqq M_A ,$$

where $M_{A''(\alpha)}$ is the infimum for program $A''(\alpha)$.

A detailed analysis of the family of all harmonized programs $A''(\alpha)$ was first given by Duffin and Peterson (1971b), although a few of their results were independently obtained by Passy (1971).

Condensed programs and harmonized programs have many common features that we will now summarize. The most fundamental common feature is that each approximates a reversed geometric program with a prototype geometric program — actually, with an infinite family of prototype geometric programs because the weights α are not unique.

The approximations are clearly conservative in that the arithmetic-geometric mean inequality and the arithmetic-harmonic mean inequality imply that each feasible solution to an arbitrary geometric program in the approximating families is also a feasible solution to the reversed geometric program. Thus, the infimum for each of those geometric programs is not less than the infimum for the reversed geometric program.

The approximating families are robust in that each feasible solution to the reversed program turns out to be a feasible solution to at least one of the approximating programs in each family. Hence, the infima for the geometric programs in each family come arbitrarily close to the infimum for the reversed program.

Under suitable conditions, a sequence of approximating programs can be chosen from either family so that the corresponding infima sequence converges monotonely to the infimum for the reversed program. Thus, a reversed geometric program can frequently be solved by solving a sequence of approximating geometric programs, each of which can be solved by the techniques of prototype geometric programming (as discussed in Sec. 2).

The harmonic-mean approach has an important feature not possessed by the geometric-mean approach in that its approximating exponent matrices clearly depend only on the reversed geometric program being approximated, and not on the given approximation. Only the posynomial coefficients change with the approximation, so many matrix computations need not be repeated during the solution of a sequence of approximating programs. This feature also leads to a variety of strategies for determining such program sequences. For example, we can employ the coefficient sensitivity analyses alluded to in Sec. 2 and developed in Appendix B of Duffin, Peterson, and Zener (1967). Those sensitivity analyses cannot be used with the geometric-mean approach because of its lack of invariance for the exponent matrix.

However, the geometric-mean approach has potentially useful features not possessed by the harmonic-mean approach. In particular, we have already seen that its approximations are generally not as conservative as those in the harmonic-mean approach, so it may require fewer iterations. Furthermore, it tends to reduce the "degree of difficulty" [page 11 of Duffin, Peterson, and Zener (1967)], an invariant in the harmonic-mean approach; so its approximating geometric programs may be easier to solve.

Consequently, the relative computational merits of the two approaches may not become apparent until considerable computational experience is obtained.

Finally, we mention that Charnes and Cooper (1966) have proposed an independent and completely different approach for approximating signomial programs with prototype geometric programs. However, the errors involved in their approximations have never been investigated.

9. EXAMPLE OF ARBITRARY ALGEBRAIC PROGRAM

We now illustrate with an example how to transform an arbitrary algebraic program into an equivalent signomial program so that it can be further transformed into an equivalent posynomial program with the aid of the transformations introduced in Sec. 4.

Without loss of generality, we assume that the independent variables are restricted to be positive, a condition that can, of course, always be achieved by replacing each unrestricted independent variable with the difference of two new positive independent variables.

Thus, suppose that we wish to minimize the algebraic function

$$\sqrt{\left[\sqrt{f_1(t)} + f_3(t)\right] \Big/ \left[\sqrt{f_2(t)} + f_4(t)\right]}\ , \tag{9.1}$$

where the $f_k(t)$, $k = 1, 2, 3, 4$, are signomials and $t = (t_1, t_2, \ldots, t_m)$. To keep imaginary numbers from being generated and, hence, make this a well-posed algebraic program, we must obviously include the constraints

$$0 \leqq f_1(t) \tag{9.2}$$

and

$$0 \leqq f_2(t) \ . \tag{9.3}$$

For the same reason, we must also include either the constraints

$$0 \leqq \sqrt{f_1(t)} + f_3(t) \tag{9.4a}$$

and

$$0 \leqq \sqrt{f_2(t)} + f_4(t) \ , \tag{9.5a}$$

or the constraints

$$\sqrt{f_1(t)} + f_3(t) \leqq 0 \tag{9.4b}$$

and

$$\sqrt{f_2(t)} + f_4(t) \leqq 0 \ . \tag{9.5b}$$

In general, more than a single program must be solved to solve one algebraic program. In our example we must solve both the program P_a with constraints (9.4a) and (9.5a), and the program P_b with constraints (9.4b) and (9.5b); after which we must choose the smaller of the two optimal values. To be concise, we shall illustrate our additional techniques on only one of these two programs, namely, program P_a whose consistency we shall assume.

To test for the possible occurrence of the indeterminate form $\sqrt{0/0}$, we should first minimize just the numerator $\left[\sqrt{f_1(t)} + f_3(t)\right]$, subject, of course, to the constraints (9.2), (9.3), (9.4a), and (9.5a). This program P_a' has an optimal value that is either zero or positive by virtue of constraint (9.4a). If it is zero, then constraint (9.5a) shows that either there is a minimizing sequence such that the denominator $\left[\sqrt{f_2(t)} + f_4(t)\right]$ is bounded from below by a positive number, or $\left[\sqrt{f_2(t)} + f_4(t)\right]$ approaches zero from above for each minimizing sequence. In the first case, the optimal value of program P_a and, hence, the original program P is zero; in the second case, there is presumably a common factor that needs to be removed from the numerator and denominator, a situation that should not arise when the original program P is properly formulated. The remaining possibility is that the optimal value for program P_a' is positive; in which event the indeterminate form $\sqrt{0/0}$ cannot occur, and we must consider both the numerator and the denominator simultaneously, that is, program P_a.

Before proceeding, we should observe that program P_a' is generally not a signomial program; but for the sake of conciseness we shall not carry out its transformation into an equivalent signomial program. Instead, we assume that its optimal value is positive so that we must actually come to grips with the more complicated program P_a.

Introducing an additional positive independent variable t_0, we see that program P_a consists essentially of minimizing the posynomial

$$\sqrt{t_0} \ , \tag{9.1a}$$

subject to both the constraints (3.2), (3.3), (3.4a), (3.5a), and the additional algebraic constraint

$$\left[\sqrt{f_1(t)} + f_3(t)\right] \Big/ \left[\sqrt{f_2(t)} + f_4(t)\right] \leqq t_0 \ ,$$

which can conveniently be rewritten as

$$0 \leqq -\sqrt{f_1(t)} + t_0\sqrt{f_2(t)} - f_3(t) + t_0 f_4(t) \qquad (9.6a)$$

by virtue of constraint (9.5a). To achieve our goal, we must still transform the algebraic functions in constraints (9.4a), (9.5a), and (9.6a) into signomials. Toward that end, we introduce two additional positive independent variables t_{m+1} and t_{m+2} so that (9.4a) and (9.5a) can be replaced by

$$0 \leqq \sqrt{t_{m+1}} + f_3(t) \;, \qquad (9.4a1)$$

$$t_{m+1} \leqq f_1(t) \;, \qquad (9.4a2)$$

and

$$0 \leqq \sqrt{t_{m+2}} + f_4(t) \;, \qquad (9.5a1)$$

$$t_{m+2} \leqq f_2(t) \;. \qquad (9.5a2)$$

Finally, we introduce another positive independent variable t_{m+3} so that (9.6a) can be replaced by

$$0 \leqq -\sqrt{t_{m+3}} + t_0\sqrt{t_{m+2}} - f_3(t) + t_0 f_4(t) \;, \qquad (9.6a1)$$

$$f_1(t) \leqq t_{m+3} \;. \qquad (9.6a2)$$

Thus, program P_a actually reduces to minimizing the posynomial (9.1a), subject to the signomial constraints (9.2), (9.3), (9.4a1), (9.4a2), (9.5a1), (9.5a2), (9.6a1), and (9.6a2). This program is obviously a signomial program which can be further transformed into a posynomial program with the aid of the techniques given in Sec. 4.

The variety of optimization problems that can be expressed as well-posed algebraic programs is worth stressing. For example, by virtue of the Stone-Weierstrass approximation theorem, each program involving continuous functions with bounded domains can be approximated with arbitrary accuracy by a rather limited class of algebraic programs, namely, the class of polynomial programs.

REFERENCES

Avriel, M., and D. J. Wilde. "Optimal Condenser Design by Geometric Programming," Indust. and Engrg. Chem. Process Design and Development, 6, (1967), 256-263.

Avriel, M., and A. C. Williams. "Complementary Geometric Programming," SIAM J. Appl. Math., 19, (1970), 125-141.

Beckenbach, E., and R. Bellman. An Introduction to Inequalities, L. W. Singer Co., 1961.

Bellman, R., and N. Karush. "Mathematical Programming and the Maximum Transform," SIAM Jour. Appl. Math., 1, (1962), 550-567.

____________. "On the Maximum Transform," Jour. Math. Anal. Appls., 6, (1963), 67-74.

____________. "Functional Equations in the Theory of Dynamic Programming XII: An Application of the Maximum Transform," Jour. Math. Anal. Appls., 6, (1963), 155-157.

Beightler, C. S., R. M. Crisp, and W. L. Meier. "Optimization by Geometric Programming," J. Indust. Engrg., 19, (1968), 117-120.

Ben-Israel, A., and L. D. Pascual. "Vector-valued Criteria in Geometric Programming," ORSA, 19, (1971), 98-104.

Bott, R., and R. J. Duffin. "On the Wang Algebra of Networks," Bull. Amer. Math. Soc., 57, (1951), 136.

____________. "On the Algebra of Networks," Trans. Amer. Math. Soc., 74, (1953), 99-109.

Charnes, A., and W. W. Cooper. "A Convex Approximant Method for Nonconvex Extensions of Geometric Programming," Proc. Nat. Acad. Sci. U.S.A., 56, (1966), 1361-1364.

Charnes, A., and M. Kirby. "Modular Design, Generalized Inverses and Convex Programming," ORSA, 13, (1965), 836-847.

Dennis, J. B. Mathematical Programming and Electrical Networks, New York, John Wiley and Sons, 1959.

Dinkel, J. J., and E. L. Peterson. "A Duality Theory for Dynamic Programming Via Geometric Programming," in preparation.

Dorn, W. S. "A Duality Theorem for Convex Programs," IBM Jour. Res. Dev., 4, (1960), 407-413.

Duffin, R. J. "Nonlinear Networks, IIa," Bull. Amer. Math. Soc., 53, (1947), 963-971.

____________. "Infinite Programs," in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker (Eds.), Princeton, N. J., Princeton University Press, 1956, 157-170.

———. "Dual Programs and Minimum Cost," SIAM Jour. Appl. Math., 10, (1962), 119.

———. "Cost Minimization Problems Treated by Geometric Means," ORSA, 10, (1962), 668.

———. "Linearizing Geometric Programs," SIAM Rev., 12, (1970), 211-227.

———. "Network Models," Proc. Sym. Mathematical Aspects of Electrical Network Theory, Amer. Math. Soc., 1969.

———. "Duality Inequalities of Mathematics and Science," Proc. Sym. Nonlinear Programming, O. L. Mangasarian and J. B. Rosen (Eds.), New York, Academic Press, 1970.

Duffin, R. J., and E. L. Peterson. "Duality Theory for Geometric Programming," SIAM J. Appl. Math., 14, (1966), 1307-1349.

———. "Geometric Programming with Signomials," J. Optimization Th. Appls., to appear, (1971a).

———. "Reversed Geometric Programs Treated by Harmonic Means," submitted to Jour. Math. Prog., (1971b).

———. "Degeneracy in Reversed Geometric Programming," in preparation, (1971c).

———. "Geometric Programs Treated with Slack Variables," Jour. Appl. Anal., in press (1971d).

Duffin, R. J., E. L. Peterson, and C. Zener. Geometric Programming-Theory and Applications, New York, John Wiley and Sons, 1967.

Duffin, R. J., and C. Zener. "Geometric Programming, Chemical Equilibrium, and the Anti-entropy Function," Proc. Nat. Acad. Sciences, 63, (1969b), 629-639.

———. "Geometric Programming and the Darwin-Fowler Method in Statistical Mechanics," Jour. Phys. Chem., 74, (1970), 2419-2923.

Ecker, J. G., and J. R. McNamara. "Geometric Programming and the Preliminary Design of Industrial Waste Treatment Plants," Water Resources Research, 7, (1971), 18-22.

Falk, J. E. "Lagrange Multipliers and Nonlinear Programming," Jour. Math. Anal. Appl., 19, (1967), 141-159.

Fasbender, E. Jour. Math., 30, (1846), 230-231.

Federowicz, A. J., and M. Mazumdar. "The Use of Geometric Programming to Maximize Reliability Achieved by Redundancy," ORSA, 16, (1968), 948-954.

Fenchel, W. "On Conjugate Convex Functions," Canadian Jour. Math., 1, (1949), 73-77.

————————. "Convex Cones, Sets and Functions," Mathematics Department mimeographed lecture notes, Princeton University, 1951.

Fiacco, A. V., and G. P. McCormick. Sequential Unconstrained Minimization Techniques for Nonlinear Programming, New York, John Wiley and Sons, 1969.

Francis, R. L., and A. V. Cabot. "Properties of a Multifacility Location Problem Involving Euclidian Distances," Naval Res. Log. Quart., in press.

Geoffrion, A. M. "Duality in Nonlinear Programming: A Simplified Applications-Oriented Development," SIAM Rev., 13, (1971), 1-38.

Gould, F. J. "Extensions of Lagrange Multipliers in Nonlinear Programming," SIAM Jour. Appl. Math., 17, (1969), 1280-1297.

Gould, F. J., and J. W. Tolle. "A Necessary and Sufficient Qualification for Constrained Optimization," SIAM Jour. Appl. Math., 20, (1971), 164-172.

Hardy, G. H., J. E. Littlewood, and G. Polya. Inequalities, Cambridge, England, Cambridge University Press, 1959.

Kazarinoff, N. D. Geometric Inequalities, New York, Random House, 1961.

Kochenberger, G. A. "Geometric Programming-Extension to Deal with Degrees of Difficulty and Loose Constraints," Ph.D. Thesis, Univ. of Colorado School of Business, 1969.

Kuhn, H. W. "On A Pair of Dual Nonlinear Problems," Nonlinear Programming, J. Abadie (Ed.), Amsterdam, North-Holland Publishing Company, 1967, 37-54.

Kuhn, H. W., and R. E. Kuenne. "An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics," Jour. Regional Science, 4, (1962), 21-33.

Minty, G. J. "Monotone Networks," Proc. Roy. Soc. London Ser. A, 257, (1960), 194-212.

————————. "On the Axiomatic Foundations of the Theories of Directed Linear Graphs, Electrical Networks, and Network Programming," Jour. Math. Mech., 15, (1966), 485-520.

Pascual, L. D., and A. Ben-Israel. "Constrained Maximization of Posynomials by Geometric Programming," Jour. Optimization Th. Appls., 5, (1970a) 73-80.

————————. "On the Solution of Maximization Problems of Optimal Design by Geometric Programming," J. Engr. Math., 4, (1970b) 349-360.

__________. "Vector Valued Criteria in Geometric Programming," Operations Research, 19, 1, (Jan./Feb. 1971), 98-104.

Passy, U. "Condensing Generalized Polynomials," J. Optimization Th. Appls., April, 1972.

Passy, U., and D. J. Wilde. "Generalized Polynomial Optimization," SIAM J. Appl. Math., 15, (1967), 1344-3156.

__________. "A Geometric Programming Algorithm for Solving Chemical Equilibrium Problems," SIAM Jour. Appl. Math., 16, (1968), 363-373.

Peterson, E. L. "An Economic Interpretation of Duality in Linear Programming," Jour. Math. Anal. Appls., 30, (1970a) 172-196.

__________. "Symmetric Duality for Generalized Unconstrained Geometric Programming," SIAM J. Appl. Math., 19, (1970b) 487-526.

__________. "Generalization and Symmetrization of Duality in Geometric Programming," in preparation, (1971).

Peterson, E. L., and J. G. Ecker. "Geometric Programming: A Unified Duality Theory for Quadratically Constrained Quadratic Programs and ℓ_p-constrained ℓ_p-approximation Problems," Bull. Amer. Math. Soc., 74, (1968), 316-321.

__________. "Geometric Programming: Duality in Quadratic Programming and ℓ_p-approximation II (Canonical Programs)," SIAM Jour. Appl. Math., 17, (1969), 317-340.

__________. "Geometric Programming: Duality in Quadratic Programming and ℓ_p-approximation I," Proc. International Sym. Math. Prog., H. W. Kuhn (Ed.), Princeton, N.J., Princeton Univ. Press, 1970a.

__________. "Geometric Programming: Duality in Quadratic Programming and ℓ_p-approximation III (Degenerate Programs)," Jour. Math. Anal. Appls., 29, (1970b) 365-383.

__________. "Geometric Programming: Duality in Quadratic Programming and ℓ_p-approximation IV (Computational Procedures)," in preparation.

Rockafellar, R. T. "Duality and Stability in Extremum Problems Involving Convex Functions," Pacific J. Math., 21, (1967a), 167-187.

__________. "Convex Programming and Systems of Elementary Monotonic Relations," Jour. Math. Anal. Appls., 19, (1967b), 543-564.

__________. "Duality in Nonlinear Programming," Amer. Math. Soc. Lects. Appl. Math., 11, G. B. Dantzig and A. F. Veinott (Eds.), Providence, R. I., Amer. Math. Soc., 1968, 401-422.

____________. Convex Analysis, Princeton, N. J., Princeton University Press, 1970.

Wendell, R. E., and E. L. Peterson. "Duality in Generalized Location Problems," in preparation.

Weyl, H. "Reparticion de Couriente en una Red Conductora," Revista Matematica Hispano-Americana, 5, (1923), 153-164.

Wilde, D. J., and C. S. Beightler. Foundations of Optimization, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1967.

Wolfe, P. "A Duality Theorem for Nonlinear Programming," Quart. Appl. Math., 19, (1961), 239-244.

Zener, C. "A Mathematical Aid in Optimizing Engineering Designs," Proc. Nat. Acad. Sci. U.S.A., 47, (1961), 537-539.

____________. "A Further Mathematical Aid in Optimizing Engineering Designs," Proc. Nat. Acad. Sci. U.S.A., 48, (1962), 518-522.

____________. Engineering Design by Geometric Programming, New York, John Wiley and Sons, 1971.