JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 11, No. 1, 1973

Geometric Programming with Signomials[1]

R. J. DUFFIN[2] AND E. L. PETERSON[3]

Communicated by G. B. Dantzig

Abstract. The difference of two *posynomials* (namely, polynomials with arbitrary real exponents, but positive coefficients and positive independent variables) is termed a *signomial.*

Each signomial program (in which a signomial is to be either minimized or maximized subject to signomial constraints) is transformed into an equivalent posynomial program in which a posynomial is to be minimized subject only to inequality posynomial constraints. The resulting class of posynomial programs is substantially larger than the class of (prototype) *geometric programs* (namely, posynomial programs in which a posynomial is to be minimized subject only to upper-bound inequality posynomial constraints). However, much of the (prototype) geometric programming theory is generalized by studying the *equilibrium solutions* to the *reversed geometric programs* in this larger class. Actually, some of this theory is new even when specialized to the class of prototype geometric programs. On the other hand, all of it can indirectly, but easily, be applied to the much larger class of well-posed *algebraic programs* (namely, programs involving real-valued functions that are generated solely by addition, subtraction, multiplication, division, and the extraction of roots).

1. Introduction

Originally developed by Duffin, Peterson, and Zener (Ref. 1), geometric programming with posynomials provides a powerful method for studying many problems in optimal engineering design (Refs. 2–6).

[1] This research was partially supported by Research Grant No. DA-AROD-31-124-6680 from the Army Research Office, Durham, North Carolina, and by a Summer Fellowship Grant from Northwestern University.

[2] Professor, Department of Mathematics, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[3] Associate Professor, Department of Industrial Engineering and Management Sciences and Department of Mathematics, Northwestern University, Evanston, Illinois.

However, many other important optimization problems can be modeled accurately only by using signomials and 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, Section III.4 of Ref. 1 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 (Ref. 7) and Blau and Wilde (Ref. 8). They generalized some of the prototype concepts and theorems in order to treat signomial programs; but most of the important protoype theorems are not valid in that more general setting. Nevertheless, this paper advances their work in such a way that those difficulties are at least partially overcome, even in the still more general setting of algebraic programs.

More recently, Avriel and Williams (Ref. 9) have shown how to reduce the study of each *rational program* to the study of a family of approximating prototype geometric programs. That reduction forms the basis of a potentially useful algorithm for which they have established convergence. It seems that similar algorithms have been proposed independently by Broverman, Federowicz, and McWhirter (Ref. 10), Pascual and Ben-Israel (Ref. 11), and Passy (Ref. 12), but for somewhat smaller classes of programs and without convergence proofs. Actually, the same ideas can be further exploited both theoretically and computationally by reducing the study of each algebraic program to the study of a family of approximating linear programs. In fact, a special application of that reduction combined with the original duality theory for linear programming (Ref. 13–14) provides an alternative proof (Ref. 15) of the main theorems from the *refined duality theory* for prototype geometric programming (Ref. 16 or Chapter VI of Ref. 1). However, in overall philosophy and approach, all of that work (on reducing the study of various programs to the study of other families of programs with nicer properties) is not nearly as closely related to this paper as it is to a parallel and independent companion paper (Ref. 17).

Other work of that general type has been done by Charnes and Cooper (Ref. 18), who proposed methods for approximating signomial programs with prototype geometric programs. However, the errors involved in their approximations have never been investigated.

With the exception of a single isolated theorem whose proof makes use of the *refined duality theory* (Ref. 16 or Chapter VI of Ref. 1), this paper is essentially self-contained.

2. Signomial Programs Transformed into Equivalent Posynomial Programs

By employing the well-known elementary transformations from mathematical programming and by using rather obvious extensions of the transformations given in Section III.4 of Ref. 9, 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 developed in this section. 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 the appendix. In this section, we shall confine our attention to the more easily described, but much smaller, class of signomial programs.

A *signomial* is a (generalized) polynomial

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

(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, one 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 contraints. Moreover, it is clear that each of the resulting constraints can be formulated in one of the following three forms:

$$f(t) \leqslant -1, \qquad f(t) \leqslant 0, \qquad f(t) \leqslant 1. \tag{1}$$

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 the following two forms:

$$g(t) \leqslant 1, \qquad g(t) \geqslant 1. \tag{2}$$

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 that 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 nonnegative, we should minimize the positive independent variable t_0 subject to the original constraints and the additional constraint $f_0(t) \leqslant t_0$, 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 $f_0(t) + t_0 \leqslant 0$, 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, the sign of the constrained infimum of $f_0(t)$ may not be known in advance. In that event, one 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, it is clear that the additional signomial constraint can be formulated in at least two of the three forms (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 (1). Thus, suppose that we wish to minimize a posynomial $g_0(t)$ subject to the signomial constraints

$$f_1(t) \leqslant -1, \qquad f_2(t) \leqslant 0, \qquad f_3(t) \leqslant 1.$$

If $f_1(t)$ is a posynomial, the constraint $f_1(t) \leqslant -1$ 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 $-f_1(t) \geqslant 1$, which already has the second of the desired forms (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 $f_2(t) \leqslant 0$ 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 $f_3(t) \leqslant 1$ is already a posynomial constraint that has the first of the desired forms (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 that we wish to minimize a posynomial $g_0(t)$ subject to the constraints

$$h_1(t) - h_4(t) \leqslant -1,$$
$$h_2(t) - h_5(t) \leqslant 0,$$
$$h_3(t) - h_6(t) \leqslant 1,$$

where the $h_k(t)$, $k = 1, 2,..., 6$, are posynomials and $t = (t_1, t_2,..., t_m)$. 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 iff 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

$$1 + h_1(t) \leqslant t_{m+1} \leqslant h_4(t),$$
$$h_2(t) \leqslant t_{m+2} \leqslant h_5(t),$$
$$h_3(t) \leqslant t_{m+3} \leqslant h_6(t) + 1.$$

But these constraints are clearly equivalent to the constraints

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

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 (2).

It is now apparent from the preceding considerations 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 (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 sections.

3. Reversed Geometric Programs and Their Equilibrium Solutions

The proceeding section shows how to transform each signomial program into an equivalent posynomial program having a special form. Posynomial programs having that special form have been termed *reversed geometric programs* (Ref. 15), because some of their inequality posynomial constraints have a direction $g(t) \geqslant 1$ that is the reverse of the direction $g(t) \leqslant 1$ required for the (prototype) *geometric programs* treated in Ref. 1 and 16.

The most general *reversed geometric program* is now stated for future reference as follows.

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

$$g_k(t) \leqslant 1, \qquad k = 1, 2,..., p, \tag{3}$$

and

$$g_k(t) \geqslant 1, \qquad k = p + 1,..., p + r \triangleq q. \tag{4}$$

Here,

$$g_k(t) \triangleq \sum_{i \in [k]} u_i(t), \qquad k = 0, 1,..., q, \tag{5}$$

and

$$u_i(t) \triangleq \begin{cases} c_i t_1^{a_{i1}} t_2^{a_{i2}} \cdots t_m^{a_{im}}, & i \in [k], \quad k = 0, 1,..., p, \quad (6) \\ c_i t_1^{-a_{i1}} t_2^{-a_{i2}} \cdots t_m^{-a_{im}}, & i \in [k], \quad k = p + 1,..., q, \quad (7) \end{cases}$$

where

$$[k] \triangleq \{m_k, m_k + 1,..., n_k\}, \qquad k = 0, 1,..., q, \tag{8}$$

and

$$1 \triangleq m_0 \leqslant n_0, \qquad n_0 + 1 \triangleq m_1 \leqslant n_1,..., n_{q-1} + 1 \triangleq m_q \leqslant n_q \triangleq n. \tag{9}$$

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 (7) in order to obtain a notational simplification in the ensuing developments. To provide other notational simplifications, we introduce the index sets

$$P \triangleq \{1, 2, ..., p\}, \tag{10}$$

$$R \triangleq \{p + 1, ..., q\}, \tag{11}$$

and

$$[K] \triangleq \bigcup_{k \in K} [k] \qquad \text{for each} \qquad K \subseteq \{0\} \cup P \cup R. \tag{12}$$

For purposes requiring pronunication, $[K]$ is called *block* K.

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

As in prototype geometric programming (Ref. 1), each posynomial term $u_i(t)$ in primal program A gives rise to an independent *dual variable* δ_i, $i \in [0] \cup [P] \cup [R]$, and each posynomial $g_k(t)$ gives rise to a dependent dual variable $\lambda_k(\delta) \triangleq \sum_{i \in [k]} \delta_i$, $k \in \{0\} \cup P \cup R$. To define the *geometric dual* of primal program A, it is convenient to extend the notation of the preceding paragraph by introducing the symbols

$$K(\delta) \triangleq \{k \in K \mid \lambda_k(\delta) \neq 0\} \qquad \text{for each} \qquad K \subseteq \{0\} \cup P \cup R, \tag{13}$$

$$[K](\delta) \triangleq \{i \in [K] \mid \delta_i \neq 0\} \qquad \text{for each} \qquad K \subseteq \{0\} \cup P \cup R. \tag{14}$$

Then, corresponding to primal program A is the following *geometric dual program*.

Dual Program B. Find the supremum M_B of the *dual objective function*

$$v(\delta) \triangleq \left\{ \left[\prod_{[0](\delta)} (c_i/\delta_i)^{\delta_i} \right] \left[\prod_{[P](\delta)} (c_i/\delta_i)^{\delta_i} \right] \left[\prod_{[R](\delta)} (c_i/\delta_i)^{-\delta_i} \right] \right\}$$
$$\times \left\{ \left[\prod_{P(\delta)} \lambda_k(\delta)^{\lambda_k(\delta)} \right] \left[\prod_{R(\delta)} \lambda_k(\delta)^{-\lambda_k(\delta)} \right] \right\} \tag{15}$$

subject to the *dual constraints* that consist of the *positivity conditions*

$$\delta_i \geqslant 0, \qquad i \in \{1, 2, \ldots, n\} = [0] \cup [P] \cup [R], \tag{16}$$

the *normality condition*

$$\lambda_0(\delta) = 1, \tag{17}$$

and the *orthogonality conditions*

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \qquad j = 1, 2, \ldots, m. \tag{18}$$

Here,

$$\lambda_k(\delta) \triangleq \sum_{i\in[k]} \delta_i\,, \qquad k \in \{0, 1, \ldots, q\} = \{0\} \cup P \cup R, \tag{19}$$

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.

The source of those differences is most easily revealed by considering the logarithm of the dual objective function. Of course, the monotonicity of the logarithmic function guarantees that $v(\delta)$ can be maximized by maximizing $\log v(\delta)$. Consequently, the following theorem shows that, unlike prototype geometric programming, reversed geometric programming is not essentially a branch of convex programming.

Theorem 3.1. The transformed dual objective function

$$\begin{aligned}\log v(\delta) \triangleq & \left[\sum_{[0](\delta)} \delta_i(\log c_i - \log \delta_i)\right] \\ & + \left[\sum_{[P](\delta)} \delta_i(\log c_i - \log \delta_i) + \sum_{P(\delta)} \lambda_k(\delta) \log \lambda_k(\delta)\right] \\ & - \left[\sum_{[R](\delta)} \delta_i(\log c_i - \log \delta_i) + \sum_{R(\delta)} \lambda_k(\delta) \log \lambda_k(\delta)\right]\end{aligned}$$

is concave in the variables δ_i, $i \in [0] \cup [P]$, but convex in the variables δ_i, $i \in [R]$.

Proof. Differentiation shows that the Hessian matrix of second partial derivatives for the function

$$\left[\sum_{[0](\delta)} \delta_i(\log c_i - \log \delta_i)\right]$$

is negative definite, so this function is concave. Differentiation and an application of the Cauchy–Schwartz inequality show that the Hessian matrix for the function

$$\left[\sum_{[P](\delta)} \delta_i(\log c_i - \log \delta_i) + \sum_{P(\delta)} \lambda_k(\delta) \log \lambda_k(\delta)\right]$$

is negative semidefinite, so this function is also concave. (For the complete details of this step, see page 122 of Ref. 1.) It follows that the function

$$-\left[\sum_{[R](\delta)} \delta_i(\log c_i - \log \delta_i) + \sum_{R(\delta)} \lambda_k(\delta) \log \lambda_k(\delta)\right]$$

is convex, so the proof of Theorem 3.1 is complete.

The convex nature of prototype geometric programming is reflected in its *main lemma* (Lemma 1 on page 114 of Ref. 1), which asserts that the primal objective function evaluated at each primal feasible solution is greater than or equal to the dual objective function evaluated at each dual feasible solution, with equality holding iff the primal and dual feasible solutions satisfy certain *extremality conditions* (a term that is used in Refs. 19–23 although not in Refs. 1 and 16.)

With suitable but very weak hypotheses, one of the main duality theorems of prototype geometric programming asserts the existence of primal and dual feasible solutions that satisfy the extremality conditions, in which event the primal infimum equals the dual supremum, and the primal and dual optimal solutions (namely, *minimizing points* for the primal program and *maximizing points* for the dual program) are characterized as those primal and dual feasible solutions that satisfy the extremality conditions.

The preceding facts and the linearity of the dual constraints lead to algorithms for finding primal and dual optimal solutions to prototype geometric programs; and it is our ultimate goal to devise such algorithms for reversed geometric programming. However, the lack of total convexity in reversed geometric programming will force us to be content with devising algorithms for finding *equilibrium solutions* that need not always be optimal.

Thus, the preceding remarks and the extremality conditions for prototype geometric programming help to motivate the following definition.

Definition 3.1. A feasible solution t^* to primal program A is termed a *primal equilibrium solution* if there is a feasible solution δ^* to dual program B such that

$$\delta_i^* g_0(t^*) = u_i(t^*), \qquad i \in [0], \tag{20-1}$$

$$\delta_i^* = \lambda_k(\delta^*)\, u_i(t^*), \qquad i \in [k], \qquad k \in P \cup R, \tag{20-2}$$

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

The rest of this paper is devoted to studying the properties of equilibrium solutions. With that goal in mind, the following theorem is fundamental in that it brings out the most elementary properties to be repetitively used in subsequent developments.

Theorem 3.2. Each primal equilibrium solution t^* and its corresponding dual equilibrium solution δ^* to programs A and B respectively have the following properties.

(i) The nonzero components of the vector δ^* are positive, more specifically,

$$\delta_i^* > 0 \qquad \text{for} \qquad i \in [0] \qquad \text{and} \qquad \text{for} \qquad i \in [P \cup R](\delta^*).$$

(ii) The nonzero components of the vector $\lambda(\delta^*)$ are positive, more specifically,

$$\lambda_0(\delta^*) = 1 \qquad \text{and} \qquad \lambda_k(\delta^*) > 0 \qquad \text{for} \qquad k \in (P \cup R)(\delta^*).$$

(iii) The vectors t^* and $\lambda(\delta^*)$ satisfy the *complementary slackness conditions*

$$\lambda_k(\delta^*)[g_k(t^*) - 1] = 0, \qquad k \in P,$$

$$\lambda_k(\delta^*)[1 - g_k(t^*)] = 0, \qquad k \in R.$$

Proof. The equilibrium conditions (20-1) and the positivity of both $g_0(t^*)$ and $u_i(t^*)$ imply that $\delta_i^* > 0$ for $i \in [0]$; and Definition 3.1 requires that each dual equilibrium solution δ^* satisfy the positivity conditions (16), so $\delta_i^* > 0$ for $i \in [P \cup R](\delta^*)$. Definition 3.1 also requires that δ^* satisfy the normality condition (17), so $\lambda_0(\delta^*) = 1$; and from conclusion (i) we see that $\lambda_k(\delta^*) > 0$ for $k \in (P \cup R)(\delta^*)$. Finally, we sum the equilibrium conditions (20-2) over i to show that $\lambda_k(\delta^*) = \lambda_k(\delta^*)\, g_k(t^*)$, $k \in P \cup R$. This completes our proof of Theorem 3.2.

From Theorem 3.2, we might guess that equilibrium solutions are intimately related to the *Lagrangian* for primal program A. Even though they are, we need not, nor do we, make explicit use of those relations in this paper. Nevertheless, those relations do 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 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. Thus, we devote the rest of this section to a study of those relations so that the practical significance of succeeding sections is established.

Corresponding to primal program A is the following *Lagrange problem.*

Lagrange Problem C. For the *Lagrangian*

$$L(t, \mu) \triangleq g_0(t) + \sum_P \mu_k[g_k(t) - 1] + \sum_R \mu_k[1 - g_k(t)],$$

find a *critical solution* (t^*, μ^*), namely, a vector $(t^*, \mu^*) \in E_{m+q}$ such that

(I) the vector t^* satisfies both the prototype posynomial constraints

$$g_k(t) - 1 \leqslant 0, \qquad k \in P,$$

and the reversed posynomial constraints

$$1 - g_k(t) \leqslant 0, \qquad k \in R;$$

(II) the vector μ^* satisfies the positivity conditions

$$\mu_k \geqslant 0, \qquad k \in P \cup R,$$

(III) the partial derivatives $D_jL(t, \mu)$ of the Lagrangian L with respect to the t_j at (t, μ), $j = 1, 2,..., m$, satisfy the conditions

$$D_jL(t^*, \mu^*) = 0, \qquad j = 1, 2,..., m,$$

(IV) the vectors t^* and μ^* satisfy the complementary slackness conditions

$$\mu_k^*[g_k(t^*) - 1] = 0, \qquad k \in P,$$
$$\mu_k^*[1 - g_k(t^*)] = 0, \qquad k \in R.$$

Here, the posynomials $g_k(t)$, $k \in \{0\} \cup P \cup R$, are, of course, as given in primal program A.

By characterizing the equilibrium solutions to primal program A as the component vectors t^* of the critical solutions (t^*, μ^*) to problem C, the following theorem relates the main concepts of this paper to the more standard concepts of mathematical programming.

Theorem 3.3. Each primal equilibrium solution t^* and its corresponding dual equilibrium solution δ^* to programs A and B respectively produce a critical solution (t^*, μ^*) to the Lagrange problem C by letting

$$\mu_k^* \triangleq \lambda_k^*(\delta^*)\, g_0(t^*), \qquad k \in P \cup R.$$

Conversely, each critical solution (t^*, μ^*) to the Lagrange problem C produces corresponding equilibrium solutions t^* and δ^* to primal program A and its dual program B respectively by letting

$$\delta_i^* \triangleq \begin{cases} [1/g_0(t^*)]\, u_i(t^*), & i \in [0], \\ [\mu_k^*/g_0(t^*)]\, u_i(t^*), & i \in [k], \qquad k \in P \cup R. \end{cases}$$

Proof. First, observe from Definition 3.1 that t^* is a feasible solution to program A, and hence possesses property (I) of the Lagrange problem C. Then note that the positivity of the posynomial $g_0(t)$ and the nonnegativity of $\lambda_k(\delta^*)$ asserted in conclusion (ii) of Theorem 3.2 show that μ^* as defined satisfies property (II). Now, write the orthogonality conditions (18) in terms of $g_0(t^*)$, $u_i(t^*)$, and $\lambda_k(\delta^*)$ by using the equilibrium conditions (20) to eliminate δ_i^*; and then eliminate $\lambda_k(\delta^*)$ in favor of μ_k^* by using our defining formula for μ^*, so that multiplication of the resulting conditions by $g_0(t^*)/t_j^*$, $j = 1, 2,..., m$, implies that t^* has property (III). Finally, observe from our defining formula for μ^* that multiplication of the complementary slackness conditions in conclusion (iii) of Theorem 3.2 by $g_0(t^*)$ verifies the validity of the complementary slackness conditions in property (IV). This completes our proof of the first half of Theorem 3.3.

To prove the second half, observe that property (I) of the Lagrange problem C asserts that t^* is a feasible solution to program A.

Due to the positivity of the posynomial terms $u_i(t)$, property (II) and our defining formulas for δ^* show that δ^* satisfies the positivity conditions (16). Moreover, a summation over i of our defining formula for δ_i^*, $i \in [0]$, shows that δ^* satisfies the normality condition (17); and multiplication of the derivative conditions in property (III) by $t_j^*/g_0(t^*)$, $j = 1, 2,..., m$, shows that δ^* satisfies the orthogonality conditions (18). Consequently, δ^* is a feasible solution to program B.

Now, our defining formula for δ_i^*, $i \in [0]$, clearly implies the validity of the equilibrium conditions (20-1). Moreover, if μ_k^* is zero; then δ_i^*, $i \in [k]$, must obviously be zero, so $\lambda_k(\delta^*)$ is clearly zero; and hence the validity of the corresponding equilibrium conditions (20–2) is established. On the other hand, if μ_k^* is positive, then the corresponding complementary slackness property (IV) implies that $g_k(t^*) = 1$, so a summation over i of our defining formula for δ_i^*, $i \in [k]$, shows that $\lambda_k(\delta^*) = \mu_k^*/g_0(t^*)$, which in turn implies that our defining formula for δ_i^*, $i \in [k]$, is identical to the corresponding equilibrium conditions (20-2). This completes our proof of Theorem 3.3.

By characterizing the critical solutions to Lagrange problem C in terms of those feasible solutions to primal program A that are *tangentially optimal* in a certain weakly global sense, the following theorem relates some of the standard concepts of mathematical programming to more practically relevant concepts.

Theorem 3.4. Suppose that t^* is a feasible solution to primal program A, and let

$$Z(t^*) \triangleq \{k \in P \cup R \mid g_k(t^*) = 1\}.$$

Then, t^* is a component vector of a critical solution (t^*, μ^*) to Lagrange problem C iff

$$g_0(t^*) \leqslant g_0(t)$$

for every vector t with positive components t_j whose logarithms $\log t_j$ satisfy the linear system

$$\sum_{j=1}^{m} A_{kj}[\log t_j - \log t_j^*] \leqslant 0, \qquad k \in Z(t^*),$$

where

$$A_{kj} \triangleq \sum_{i\in[k]} a_{ij} u_i(t^*), \qquad k \in Z(t^*), \qquad j = 1, 2,..., m.$$

Proof. Performing most of the partial differentiations in the equations of property (III) for Lagrange problem C, and then multiplying the resulting equations by the positive numbers t_j^*, $j = 1, 2,..., m$, we readily see that t^* is a component vector of a critical solution (t^*, μ^*) to Lagrange problem C iff there exist nonnegative numbers μ_k^*, $k \in Z(t^*)$, for which

$$t_j^* D_j g_0(t^*) + \sum_{Z(t^*)} \mu_k^* A_{kj} = 0, \qquad j = 1, 2,..., m.$$

Now, according to the well-known Farkas lemma concerning linear systems (for example, see Lemma 1 on page 17 of Ref. 1), such numbers $\mu_k{}^*$, $k \in Z(t^*)$ are known to exist iff

$$0 \leqslant \sum_{j=1}^{m} t_j{}^* D_j g_0(t^*)[\log t_j - \log t_j{}^*] \tag{21}$$

for every vector t with positive components t_j whose logarithms $\log t_j$ satisfy the linear system

$$\sum_{j=1}^{m} A_{kj}[\log t_j - \log t^*{}_j] \leqslant 0, \qquad k \in Z(t^*). \tag{22}$$

Consequently, to complete our proof, we need only show that Ineq. (21) can be replaced by the inequality $g_0(t^*) \leqslant g_0(t)$ without disturbing the validity of the preceding statement.

To do so, we make the change of independent variables

$$t_j \triangleq \exp(z_j), \qquad j = 1, 2,..., m, \tag{23}$$

so that primal program A is transformed into an equivalent *reversed convex program* to which we can apply an elementary theorem from convex analysis. This equivalent program clearly consists of minimizing the convex function $G_0(z)$ subject to both the *prototype convex constraints*

$$G_k(z) \leqslant 1, \qquad k \in P, \tag{24}$$

and the *reversed convex constraints*

$$G_k(z) \geqslant 1, \qquad k \in R, \tag{25}$$

where

$$G_k(z) \triangleq \sum_{i \in [k]} U_i(z), \qquad k \in \{0\} \cup P \cup R, \tag{26}$$

and

$$U_i(z) \triangleq \begin{cases} c_i \exp(a_{i1}z_1 + a_{i2}z_2 + \cdots + a_{im}z_m), & i \in [k], \quad k \in \{0\} \cup P, \quad (27) \\ c_i \exp(-a_{i1}z_1 - a_{i2}z_2 - \cdots - a_{im}z_m), & i \in [k], \quad k \in R. \quad (28) \end{cases}$$

(Of course, the convexity of these functions G_k, $k \in \{0\} \cup P \cup R$, follows easily from the positivity of the coefficients c_i, $i \in [0] \cup [P] \cup [R]$.) In terms of this notation and the inner product notation $\langle \cdot, \cdot \rangle$, Ineq. (21) is simply

$$0 \leqslant \langle \nabla G_0(z^*), z - z^* \rangle, \tag{29-1}$$

and Ineqs. (22) are simply

$$\langle \nabla G_k(z^*), z - z^* \rangle \begin{cases} \leqslant 0, & k \in P \cap Z(t^*), \\ \geqslant 0, & k \in R \cap Z(t^*). \end{cases} \tag{30}$$

From the convexity of G_0 we know that

$$\langle \nabla G_0(z^*), z - z^* \rangle \leqslant G_0(z) - G_0(z^*),$$

so the validity of Ineq. (29-1) implies the validity of the inequality

$$G_0(z^*) \leqslant G_0(z). \tag{29-2}$$

On the other hand, the solution set for the linear inequalities (30) is obviously a cone with vertex z^*, so the validity of Ineq. (29-2) for each vector z in that solution cone implies the validity of Ineq. (29-1) for each such solution vector z, by virtue of the differential calculus. Finally, we observe that Ineq. (29-2) is equivalent to the inequality

$$g_0(t^*) \leqslant g_0(t). \tag{29-3}$$

This completes our proof of Theorem 3.3.

The way in which a feasible solution $t^* = \exp(z^*)$ can be tangentially optimal in a weakly global sense is indicated by the solution cone for the linear inequalities (30). The tangential nature is indicated by the presence of $\nabla G_k(z^*)$, $k \in P \cup R$, in (30); the global nature is indicated by the fact that this solution cone need not be sufficiently small; and the weak nature is indicated by the fact that this solution cone does not contain the entire set of feasible solutions to the constraint inequalities (24)–(25) unless $R \cap Z(t^*) = \varnothing$.

We now have enough machinery to establish the optimal nature of the equilibrium solutions to primal program A.

Corollary 3.1. Suppose that t^* is a feasible solution to primal program A, and let

$$Z(t^*) \triangleq \{k \in P \cup R \mid g_k(t^*) = 1\}.$$

Then, t^* is an equilibrium solution to primal program A iff

$$g_0(t^*) \leqslant g_0(t)$$

for every vector t with positive components t_j whose logarithms $\log t_j$ satisfy the linear system

$$\sum_{j=1}^{m} A_{kj}[\log t_j - \log t_j^*] \leqslant 0, \qquad k \in Z(t^*),$$

where

$$A_{kj} \triangleq \sum_{i \in [k]} a_{ij} u_i(t^*), \qquad k \in Z(t^*), \qquad j = 1, 2, \ldots, m.$$

This corollary follows immediately from Theorems 3.3 and 3.4.

It is worth mentioning that equilibrium solutions to primal program A are also tangentially optimal in an even more weakly global but more computationally exploitable sense, as described in Ref. 17. Moreover, they are actually (globally) optimal when primal program A is a prototype geometric program (that is, $R = \varnothing$), as can be seen from the main lemma of prototype geometric programming (Lemma 1 on page 114 of Ref. 1). In contrast, they need not even be *locally optimal* when primal program A is not a prototype geometric program (that is, $R \neq \varnothing$).

For example, notice that the vector $t^* \triangleq (1, 1)$ and the vector $\mu^* \triangleq 1$ produce a critical solution (t^*, μ^*) to the Lagrange problem C corresponding to the primal program A that consists of minimizing the posynomial $g_0(t) \triangleq t_1 + t_2$ subject to the single reversed posynomial constraint $g_1(t) \triangleq (1/2)\, t_1^2 + (1/2)\, t_2^2 \geqslant 1$. Hence, Theorem 3.3 asserts that $t^* \triangleq (1, 1)$ is an equilibrium solution to this primal program; but the contours of g_0 and g_1 obviously show that this equilibrium solution is not locally optimal even though it is tangentially optimal. Such (undesired) equilibrium solutions are clearly unstable and hence, due to round-off error, are possibly less likely to be obtained by most numerical algorithms, especially those proposed in Refs. 9 and 17.

It is worth recalling the well-known fact that every locally optimal solution to a general nonlinear program under any of several rather weak *constraint qualifications* is always part of a critical solution to the corresponding Lagrange's problem (for example, see Chapter 5 of Ref. 24 or Chapter 2 of Ref. 25). Thus, we infer from Theorem 3.3 that the (desired) set of all (globally) optimal solutions to primal program A is almost always a subset of the set of all primal equilibrium solutions and hence can almost always be found by sharpening the methods to be used for computing equilibrium solutions.

The initial work on equilibrium solutions for reversed geometric programs was performed by Passy and Wilde (Ref. 7) in the setting of *generalized polynomial programs* (i.e., signomial programs); but they used the terminology *pseudominimum* rather than equilibrium solution. Subsequent work of a more detailed nature on the general relationships between locally optimal solutions, stable equilibrium solutions, and unstable equilibrium solutions was performed by Avriel and Williams (Section 4 of Ref. 9); but they used the terminology *quasiminimum* rather than equilibrium solution. In addition to studying important new questions and phenomena, this paper and its companion paper (Ref. 17) present a self-contained alternative approach to almost all of the important questions and phenomena studied in Ref. 7 and 9.

The remaining sections of this paper bring to light some important properties of equilibrium solutions, which lead to a family of *indirect methods* for computing them. Other important properties that lead to families of *direct methods* are brought to light in Refs. 9 and 17.

4. Basic Properties of Equilibrium Solutions

The last part of the preceding section tended to concentrate on the properties of primal equilibrium solutions. In this and the next section, the emphasis shifts somewhat toward the properties of dual equilibrium solutions. Those properties are more nearly linear in nature, and hence dual equilibrium solutions are somewhat more amenable to computation.

In addition to showing that the nonzero components of dual equilibrium solutions occur in *blocks*, the following fundamental theorem also presents a useful extension of an identity that was first obtained by Zener in prototype geometric programming.

Theorem 4.1. If δ^* is an equilibrium solution to dual program B, then the following results hold.

(i) For each $k \in P \cup R$, either $\delta_i^* = 0$ for each $i \in [k]$ or $\delta_i^* > 0$ for each $i \in [k]$, with the latter being the case iff $\lambda_k(\delta^*) > 0$; hence,

$$[P \cup R](\delta^*) = [(P \cup R)(\delta^*)].$$

(ii) Given the equilibrium value $E_A \triangleq g_0(t^*)$ for a corresponding equilibrium solution t^* to primal program A, the identity

$$E_A^{\lambda_0(y)} \equiv \left\{\left[\prod_{[0](\delta^*)} (c_i/\delta_i^*)^{y_i}\right]\left[\prod_{[P](\delta^*)} (c_i/\delta_i^*)^{y_i}\right]\left[\prod_{[R](\delta^*)} (c_i/\delta_i^*)^{-y_i}\right]\right\}$$
$$\times \left\{\left[\prod_{P(\delta^*)} \lambda_k(\delta^*)^{\lambda_k(y)}\right]\left[\prod_{R(\delta^*)} \lambda_k(\delta^*)^{-\lambda_k(y)}\right]\right\}$$

is valid for every vector y that satisfies both the orthogonality conditions

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

and the condition

$$y_i = 0 \qquad \text{for each } i \text{ for which } \delta_i^* = 0.$$

Proof. From conclusion (ii) of Theorem 3.2, we know that $\lambda_k(\delta^*) \geqslant 0$ for $k \in P \cup R$. Consequently, the equilibrium conditions (20-2) and the positivity of $u_i(t^*)$ imply that either $\delta_i^* = 0$ for each $i \in [k]$ or $\delta_i^* > 0$ for each $i \in [k]$, with the latter being the case iff $\lambda_k(\delta^*) > 0$. This establishes conclusion (i).

To prove conclusion (ii), first divide the equilibrium conditions (20-1) by c_i and raise both sides to the power y_i to obtain the relations

$$(\delta_i^*/c_i)^{y_i} E_A^{y_i} = (u_i^*/c_i)^{y_i}, \qquad i \in [0](\delta^*).$$

Then, for each $i \in [k](\delta^*)$ and each $k \in P(\delta^*)$, divide the equilibrium conditions (20-2) by c_i and raise both sides to the power y_i to obtain the relations

$$(\delta_i^*/c_i)^{y_i} = (\lambda_k^*)^{y_i}(u_i^*/c_i)^{y_i}, \qquad i \in [k](\delta^*), \qquad k \in P(\delta^*).$$

Also, for each $i \in [k](\delta^*)$ and each $k \in R(\delta^*)$, divide the equilibrium conditions (20-2) by c_i and raise both sides to the power $-y_i$ to obtain the relations

$$(\delta_i^*/c_i)^{-y_i} = (\lambda_k^*)^{-y_i}(u_i^*/c_i)^{-y_i}, \qquad i \in [k](\delta^*), \qquad k \in R(\delta^*).$$

Now, multiply all of these relations together, and use the defining equations (6) and (7) for u_i^*, to obtain the relation

$$E_A^{\lambda_0}\left[\prod_{[0](\delta^*)} (\delta_i^*/c_i)^{y_i}\right]\left[\prod_{[P](\delta^*)} (\delta_i^*/c_i)^{y_i}\right]\left[\prod_{[R](\delta^*)} (\delta_i^*/c_i)^{-y_i}\right]$$
$$= \left[\prod_{P(\delta^*)} (\lambda_k^*)^{\lambda_k}\right]\left[\prod_{R(\delta^*)} (\lambda_k^*)^{-\lambda_k}\right]\left[\prod_{[T](\delta^*)} (p_i^*)^{y_i}\right],$$

where $T \triangleq \{0\} \cup P \cup R$ and $p_i \triangleq \prod_{j=1}^m t_j^{a_{ij}}$ for each $i \in [T]$. This establishes our identity; because the condition that $y_i = 0$ for each $i \notin [T](\delta^*)$, the definition $p_i \triangleq \prod_{j=1}^m t_j^{a_{ij}}$ and the orthogonality conditions $\sum_{i=1}^n a_{ij}\, y_i = 0$, $j = 1, 2,..., m$, imply that

$$\prod_{[T](\delta^*)} (p_i)^{y_i} = \prod_{i=1}^{n} (p_i)^{y_i} = t_1^{\sum_{i=1}^n a_{i1} y_i} \cdots t_m^{\sum_{i=1}^n a_{im} y_i} = 1$$

for each $t > 0$. Thus, our proof of Theorem 4.1 is seen to be complete.

The following corollary to Theorem 4.1 extends to reversed geometric programming a somewhat weakened version of the important prototype geometric programming theorem that asserts the equality of the primal program infimum and its corresponding dual program supremum.

Corollary 4.1. Corresponding primal and dual equilibrium values $E_A \triangleq g_0(t^*)$ and $E_B \triangleq v(\delta^*)$ are always equal.

This corollary follows immediately from the identity in conclusion (ii) by choosing y to be δ^*, because δ^* is dual feasible and hence satisfies the normality condition $\lambda_0(\delta^*) = 1$.

Other important properties of dual equilibrium solutions can be conveniently described in terms of the *nullity vectors* that were used in prototype geometric programming (page 84 of Ref. 1). A *nullity vector* is simply any solution ν to the homogeneous counterpart of the normality and orthogonality conditions, namely,

$$\lambda_0(\nu) = 0 \tag{31}$$

and

$$\sum_{i=1}^{n} a_{ij}\nu_i = 0, \qquad j = 1, 2,..., m. \tag{32}$$

The following corollary to Theorem 4.1 is especially useful because it isolates each dual equilibrium solution δ^* and the posynomial coefficient vector $c = (c_1 ,..., c_n)$ on the opposite sides of an identity.

Corollary 4.2. If δ^* is an equilibrium solution to dual program B, then every nullity vector ν such that

$$\nu_i = 0 \qquad \text{for each } i \text{ for which} \qquad \delta_i^* = 0$$

satisfies the identity

$$F(\delta^*, \nu) \equiv K(c, \nu),$$

where

$$F(\delta^*, \nu) \triangleq \left\{\left[\prod_{[0](\delta^*)} (\delta_i^*)^{\nu_i}\right]\left[\prod_{[P](\delta^*)} (\delta_i^*)^{\nu_i}\right]\left[\prod_{[R](\delta^*)} (\delta_i^*)^{-\nu_i}\right]\right\}$$
$$\times \left\{\left[\prod_{P(\delta^*)} \lambda_k(\delta^*)^{-\lambda_k(\nu)}\right]\left[\prod_{R(\delta^*)} \lambda_k(\delta^*)^{\lambda_k(\nu)}\right]\right\},$$

$$K(c, \nu) \triangleq \left[\prod_{[0]} (c_i)^{\nu_i}\right]\left[\prod_{[P]} (c_i)^{\nu_i}\right]\left[\prod_{[R]} (c_i)^{-\nu_i}\right].$$

This corollary follows immediately from the identity in conclusion (ii) by choosing y to be ν, because ν is a nullity vector and hence satisfies the condition $\lambda_0(\nu) = 0$ and because the condition $\nu_i = 0$ for each $i \notin [P](\delta^*) \cup [R](\delta^*)$ implies that

$$\left[\prod_{[P](\delta^*)} (c_i)^{\nu_i}\right]\left[\prod_{[R](\delta^*)} (c_i)^{-\nu_i}\right] = \left[\prod_{[P]} (c_i)^{\nu_i}\right]\left[\prod_{[R]} (c_i)^{-\nu_i}\right].$$

The following theorem is important in that it sheds considerable light on the nature of the *equilibrium identity* $F(\delta^*, \nu) \equiv K(c, \nu)$ by providing a fundamental link between the *basic function* $F(\cdot, \nu)$, the *basic constant* $K(c, \nu)$, and the directional derivative function $D_\nu v(\cdot)$ of the dual objective function v in a given direction ν.

Theorem 4.2. If δ is a feasible solution to dual program B, then $\delta + r\nu$ is also a feasible solution to dual program B for each scalar r in some sufficiently small neighborhood of zero iff ν is a nullity vector such that

$$\nu_i = 0 \qquad \text{for each } i \text{ for which} \qquad \delta_i = 0,$$

in which case the dual objective function v has a directional derivative $D_\nu v(\delta)$ at δ in the direction ν that is given by the formula

$$D_\nu v(\delta) = \{\log K(c, \nu) - \log F(\delta, \nu) - \lambda_0(\nu)\} v(\delta),$$

where

$$F(\delta, \nu) \triangleq \left\{\left[\prod_{[0](\delta)} (\delta_i)^{\nu_i}\right]\left[\prod_{[P](\delta)} (\delta_i)^{\nu_i}\right]\left[\prod_{[R](\delta)} (\delta_i)^{-\nu_i}\right]\right\}$$
$$\times \left\{\left[\prod_{P(\delta)} \lambda_k(\delta)^{-\lambda_k(\nu)}\right]\left[\prod_{R(\delta)} \lambda_k(\delta)^{\lambda_k(\nu)}\right]\right\},$$

$$K(c, \nu) \triangleq \left[\prod_{[0]} (c_i)^{\nu_i}\right]\left[\prod_{[P]} (c_i)^{\nu_i}\right]\left[\prod_{[R]} (c_i)^{-\nu_i}\right].$$

Proof. From elementary linear algebra, we know that $\delta + r\nu$ satisfies the normality and orthogonality conditions for at least one nonzero scalar r iff ν is a nullity vector, in which case $\delta + r\nu$ satisfies the normality and orthogonality conditions for every scalar r. Moreover, it is clear that $\delta + r\nu$ satisfies the positivity conditions for each scalar r in some sufficiently small neighborhood of zero iff $\nu_i = 0$ for each i for which $\delta_i = 0$. This proves the first assertion in Theorem 4.2.

The second assertion can be established under much weaker hypotheses than those that are given. In fact, we see from the defining formula (15) for v that, to keep imaginary numbers from being generated the domain of v need only be limited to those vectors δ that satisfy the positivity conditions. Given such a vector δ, we have already observed that the vector $\delta + r\nu$ is also such a vector for each scalar r in some sufficiently small neighborhood of zero iff $\nu_i = 0$ for each i for which $\delta_i = 0$. Under these conditions the defining formula (15) for v shows that, at $r = 0$, the function $V(r) \triangleq v(\delta + r\nu)$ has the following logarithmic derivative:

$$\begin{aligned} D \log V(0) = & \sum_{[0](\delta)} [\log c_i - \log \delta_i - 1]\, \nu_i \\ & + \sum_{[P](\delta)} [\log c_i - \log \delta_i - 1]\, \nu_i + \sum_{P(\delta)} [\log \lambda_k(\delta) + 1]\, \lambda_k(\nu) \\ & - \sum_{[R](\delta)} [\log c_i - \log \delta_i - 1]\, \nu_i - \sum_{R(\delta)} [\log \lambda_k(\delta) + 1]\, \lambda_k(\nu). \end{aligned}$$

Using our defining equation for $F(\delta, \nu)$ and the linear homogeneous condition on ν, we see that

$$D \log V(0) = \sum_{[0](\delta)} \nu_i \log c_i + \sum_{[P](\delta)} \nu_i \log c_i - \sum_{[R](\delta)} \nu_i \log c_i - \log F(\delta, \nu) - \lambda_0(\nu).$$

This equation establishes the desired formula because of our defining equation for $K(c, \nu)$ and the linear homogeneous condition on ν. Thus, our proof of Theorem 4.2 is seen to be complete.

The following corollary to Theorem 4.2 shows that dual equilibrium solutions are *stationary solutions* to dual program B.

Corollary 4.3. If δ^* is an equilibrium solution to dual program B, then the identity

$$D_\nu v(\delta^*) \equiv 0$$

is valid for every vector ν such that $\delta^* + r\nu$ is a feasible solution to dual program B for each scalar r in some sufficiently small neighborhood of zero.

This corollary follows immediately from choosing δ to be δ^* in Theorem 4.2 and then applying Corollary 4.2.

As indicated by the lack of total convexity brought to light in Theorem 3.1, equilibrium solutions to dual program B need not either minimize or maximize the dual objective function v, even though they are stationary solutions to dual program B. However, dual equilibrium solutions are *tangentially optimal* in a *strongly* global sense, as explained in Ref. 17.

The following theorem shows that dual equilibrium solutions are almost characterized by the properties that have been brought to light in this section.

Theorem 4.3. If

(i) δ^* is a feasible solution to dual program B,

(ii) for each $k \in \{0\} \cup P \cup R$, either $\delta_i^* = 0$ for each $i \in [k]$, or $\delta_i^* > 0$ for each $i \in [k]$,

(iii) every nullity vector ν such that $\nu_i = 0$ for each i for which $\delta_i^* = 0$ satisfies the identity

$$F(\delta^*, \nu) \equiv K(c, \nu),$$

where

$$F(\delta^*, \nu) \triangleq \left\{ \left[\prod_{[0](\delta^*)} (\delta_i^*)^{\nu_i} \right] \left[\prod_{[P](\delta^*)} (\delta_i^*)^{\nu_i} \right] \left[\prod_{[R](\delta^*)} (\delta_i^*)^{-\nu_i} \right] \right\}$$
$$\times \left\{ \left[\prod_{P(\delta^*)} \lambda_k(\delta^*)^{-\lambda_k(\nu)} \right] \left[\prod_{R(\delta^*)} \lambda_k(\delta^*)^{\lambda_k(\nu)} \right] \right\},$$

$$K(c, \nu) \triangleq \left[\prod_{[0]} (c_i)^{\nu_i} \right] \left[\prod_{[P]} (c_i)^{\nu_i} \right] \left[\prod_{[R]} (c_i)^{-\nu_i} \right],$$

then deletion of the zero components of δ^* produces an equilibrium solution ζ^* to the geometric dual program B' corresponding to the primal program A' that results from deleting those constraints in primal program A for which $\lambda_k(\delta^*) = 0$.

Proof. Primal program A' consists of minimizing the posynomial $g_0(t)$ subject to both the prototype posynomial constraints

$$g_k(t) \leqslant 1, \qquad k \in P', \tag{33}$$

and the reversed posynomial constraints

$$g_k(t) \geqslant 1, \qquad k \in R', \tag{34}$$

where the index set

$$P' \triangleq P(\delta^*) \tag{35}$$

and the index set

$$R' \triangleq R(\delta^*). \tag{36}$$

In the following developments, it is notationally convenient to also employ both the symbol

$$Q' \triangleq P' \cup R' \tag{37}$$

and the symbol

$$T' \triangleq \{0\} \cup Q'. \tag{38}$$

Accordingly, the geometric dual program B' corresponding to primal program A' can be described by introducing an independent dual vector variable ζ whose components ζ_i, $i \in [T']$, are not consecutively ordered unless $T' = T \triangleq \{0, 1, 2, ..., q\}$. To give such a description, we also introduce the dependent dual variables $\omega_k(\zeta) \triangleq \sum_{i\in[k]} \zeta_i$, $k \in T'$, and we adapt our other notation so that

$$K'(\zeta) \triangleq \{k \in K' \mid \omega_k(\zeta) \neq 0\} \qquad \text{for each} \qquad K' \subseteq T', \tag{39}$$

$$[K'](\zeta) \triangleq \{i \in [K'] \mid \zeta_i \neq 0\} \qquad \text{for each} \qquad K' \subseteq T'. \tag{40}$$

Then, dual program B' consists of maximizing the objective function

$$v'(\zeta) \triangleq \left\{\left[\prod_{[0](\zeta)} (c_i/\zeta_i)^{\zeta_i}\right]\left[\prod_{[P'](\zeta)} (c_i/\zeta_i)^{\zeta_i}\right]\left[\prod_{[R'](\zeta)} (c_i/\zeta_i)^{-\zeta_i}\right]\right\}$$
$$\times \left\{\left[\prod_{P'(\zeta)} \omega_k(\zeta)^{\omega_k(\zeta)}\right]\left[\prod_{R'(\zeta)} \omega_k(\zeta)^{-\omega_k(\zeta)}\right]\right\}, \tag{41}$$

subject to the positivity conditions

$$\zeta_i \geqslant 0, \qquad i \in [T'], \tag{42}$$

the normality conditions

$$\omega_0(\zeta) = 1, \tag{43}$$

and the orthogonality conditions

$$\sum_{i\in[T']} a_{ij}\zeta_i = 0, \qquad j = 1, 2, ..., m, \tag{44}$$

where

$$\omega_k(\zeta) \triangleq \sum_{i\in[k]} \zeta_i, \qquad k \in T'. \tag{45}$$

From hypothesis (i), we easily infer that the vector ζ^* with components

$$\zeta_i^* \triangleq \delta_i^*, \qquad i \in [T'], \tag{46}$$

is a feasible solution to dual program B'; and from hypothesis (ii), we immediately see that all components of ζ^* are strictly positive. Thus, introducing another independent vector variable v whose components v_i, $i \in [T']$, are not consecutively ordered unless $T' = T \triangleq \{0, 1, 2, ..., q\}$, we readily deduce from hypothesis (iii) that every nullity vector v for program B' satisfies the equilibrium identity

$$F'(\zeta^*, v) \equiv K'(c, v), \tag{47-1}$$

where

$$F'(\zeta^*, v) \triangleq \left\{\left[\prod_{[0](\zeta^*)} (\zeta_i^*)^{v_i}\right]\left[\prod_{[P'](\zeta^*)} (\zeta_i^*)^{v_i}\right]\left[\prod_{[R'](\zeta^*)} (\zeta_i^*)^{-v_i}\right]\right\} \times \left\{\left[\prod_{P'(\zeta^*)} \omega_k(\zeta^*)^{-\omega_k(v)}\right]\left[\prod_{R'(\zeta^*)} \omega_k(\zeta^*)^{\omega_k(v)}\right]\right\}, \tag{47-2}$$

$$K'(c, v) \triangleq \left[\prod_{[0]} (c_i)^{v_i}\right]\left[\prod_{[P']} (c_i)^{v_i}\right]\left[\prod_{[R']} (c_i)^{-v_i}\right]. \tag{47-3}$$

Consequently, ζ^* inherits from δ^* every property relative to program B' that δ^* has relative to program B. In addition, ζ^* has important properties that δ^* need not have; namely, all components of ζ^* are strictly positive, and hence all numbers $\omega_k(\zeta^*)$, $k \in T'$, are strictly positive. Using all of these properties of ζ^*, we shall now carry out our proof by demonstrating the existence of a vector t^* such that

$$\zeta_i^* g_0(t^*) = u_i(t^*), \qquad i \in [0], \tag{48-1}$$

$$\zeta_i^* = \omega_k(\zeta^*)\, u_i(t^*), \qquad i \in [k], \qquad k \in P' \cup R'. \tag{48-2}$$

Such a vector t^* is automatically a feasible solution to program A'; because relation (48-2) shows that

$$\omega_k(\zeta^*) \triangleq \sum_{i\in[k]} \zeta_i^* = \omega_k(\zeta^*) \sum_{i\in[k]} u_i(t^*) = \omega_k(\zeta^*)\, g_k(t^*), \qquad k \in P' \cup R',$$

which in turn implies that

$$g_k(t^*) = 1, \qquad k \in P' \cup R',$$

by virtue of the positivity of $\omega_k(\zeta^*)$, $k \in P' \cup R'$.

The key to establishing the existence of a vector t^* that satisfies relations (48) is the application of a powerful existence theorem from prototype geometric programming to a certain prototype primal program A^* and its geometric dual program B^*. Corresponding to programs A^* and B^* is a new coefficient vector c^* with components

$$c_i^* \triangleq \begin{cases} c_i, & i \in [0] \cup [P'], \qquad (49\text{-}1) \\ [\zeta_i^*/\omega_k(\zeta^*)]^2\, c_i^{-1}, & i \in [k], \quad k \in R', \qquad (49\text{-}2) \end{cases}$$

which are clearly positive. Now, the defining equation (49-2) readily implies that every vector v satisfies the identity

$$\prod_{R'(\zeta^*)} \left[\prod_{[k]} (\zeta_i^*/\omega_k(\zeta^*))^{2v_i} \right] \equiv \prod_{[R']} (c_i^* c_i)^{v_i},$$

because it is easy to verify that $R'(\zeta^*) = R'$. Taking account of the defining equation (49-1), and multiplying the left-hand and right-hand sides of the preceding identity into the left-hand and right-hand sides respectively of the equilibrium identity (47) shows that every nullity vector v for program B' satisfies the identity

$$F^*(\zeta^*, v) \equiv K^*(c^*, v), \tag{50-1}$$

where

$$F^*(\zeta^*, v) \triangleq \left\{ \left[\prod_{[0](\zeta^*)} (\zeta_i^*)^{v_i} \right] \left[\prod_{[Q'](\zeta^*)} (\zeta_i^*)^{v_i} \right] \right\} \left\{ \left[\prod_{Q'(\zeta^*)} \omega_k(\zeta^*)^{-\omega_k(v)} \right] \right\}, \tag{50-2}$$

$$K^*(c^*, v) \triangleq \left[\prod_{[0]} (c_i^*)^{v_i} \right] \left[\prod_{[Q']} (c_i^*)^{v_i} \right]. \tag{50-3}$$

This identity suggests that we consider a geometric dual program B^* which differs from program B' only in that the objective function for program B^* is

$$v^*(\zeta) \triangleq \left\{ \left[\prod_{[0](\zeta)} (c_i^*/\zeta_i)^{\zeta_i} \right] \left[\prod_{[Q'](\zeta)} (c_i^*/\zeta_i)^{\zeta_i} \right] \right\} \left\{ \left[\prod_{Q'(\zeta)} \omega_k(\zeta)^{\omega_k(\zeta)} \right] \right\}. \tag{51}$$

The fact that programs B^* and B' have the same constraints implies that they have the same set of nullity vectors v, so our identity (50) is actually the equilibrium identity corresponding to program B^*. The fact that programs B^* and B' have the same constraints also implies that ζ^* is a feasible solution to program B^*, because we have already observed that ζ^* is a feasible solution to program B'. Moreover, the defining equation (46) shows that each component of ζ^* is strictly positive, so ζ^* is in the (relative) interior of the feasible solution set for program B^*.

With the preceding properties of ζ^* at hand, it is now expedient to apply to program B^* the theory already developed in this paper. To do so, we temporarily identify program B^* with program B by choosing $c_i = c_i^*$, $i \in [T']$, while letting $P = Q'$ and $R = \varnothing$. Then, Theorem (4.2) along with our equilibrium identity (50) for program B^* shows that ζ^* is a stationary solution to program B^*. Hence, the presence of ζ^* in the (relative) interior of the feasible solution set for program B^* implies that ζ^* is actually an optimal solution to program B^*, because program B^* is convex by virtue of Theorem 3.1 and the relation $R = \varnothing$.

This same relation $R = \varnothing$ also shows that program B^* is the geometric dual of a prototype primal program A^*, so we can now apply the refined duality theory of prototype geometric programming (Ref. 16 or Chapter VI of Ref. 1).

First, using the defining equations (49) for the coefficients c_i^*, we observe that primal program A^* consists of minimizing the posynomial $g_0(t)$ subject to the standard posynomial constraints

$$g_k(t) \leqslant 1, \qquad k \in P', \tag{52-1}$$

$$g_k^*(t) \leqslant 1, \qquad k \in R', \tag{52-2}$$

where

$$g_k^*(t) \triangleq \sum_{[k]} [\zeta_i^*/\omega_k(\zeta^*)]^2[u_i(t)]^{-1}, \qquad k \in R'. \tag{53}$$

Now, programs A^* and B^* are *canonical* (see page 169 of Ref. 1), because the dual program B^* has a feasible solution ζ^* with strictly positive components. Thus, the optimality of ζ^* along with Theorem 1 on page 169 of Ref. 1 implies that program A^* has an optimal solution t^* such that $g_0(t^*) = v^*(\zeta^*)$. Then, the main lemma of prototype geometric programming (Lemma 1 on page 167 of Ref. 1) shows that t^* and ζ^* satisfy the extremality conditions

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

and

$$\zeta_i^* = \begin{cases} \omega_k(\zeta^*)\, u_i(t^*), & i \in [k], \quad k \in P', \\ \omega_k(\zeta^*)[\zeta_i^*/\omega_k(\zeta^*)]^2[u_i(t^*)]^{-1}, & i \in [k], \quad k \in R'. \end{cases}$$

Algebraic manipulation of the second part of the latter relation enables us to rewrite the latter relation in the more compact form

$$\zeta_i^* = \omega_k(\zeta^*)\, u_i(t^*), \qquad i \in [k], \qquad k \in P' \cup R'.$$

This establishes the validity of relations (48) and hence completes our proof of Theorem 4.3.

It is worth nothing that Theorem 4.3 cannot be sharpened by strengthening only its conclusion. For example, consider an arbitrary pair of primal and dual programs A' and B' that possess corresponding primal and dual equilibrium solutions t^* and ζ^*, respectively; and then append to program A' an additional primal constraint that cannot be satisfied. Then, ζ^* with appropriate zero components appended produces a vector δ^* that satisfies the hypotheses of Theorem 4.3 relative to the resulting pair of primal and dual programs A and B; but programs A and

B cannot have an equilibrium solution because program A is clearly inconsistent.

Although computationally oriented, the next (and final) section of this paper does shed additional light on the nature of equilibrium solutions.

5. An Indirect Method for Obtaining Equilibrium Solutions

The theory developed in the preceding section leads to useful necessary conditions that help to determine equilibrium solutions. Such necessary conditions for dual equilibrium solutions δ^* come from observing that the set of all those nullity vectors ν that satisfy the δ^* *zero-condition* (namely, $\nu_i = 0$ for each i for which $\delta_i{}^* = 0$) forms a vector subspace of E_n.

If this *nullity subspace* corresponding to δ^* contains only the zero vector, then the elementary theory of linear algebra asserts that the normality condition, the orthogonality conditions, and the δ^* zero-condition have a unique solution δ^*, in which case δ^* can easily be computed by elementary linear algebra, and there are no other dual equilibrium solutions that satisfy the δ^* zero-condition.

On the other hand, if the nullity subspace corresponding to δ^* contains more than just the zero vector, then it has positive dimension $d \geqslant 1$ and d basis vectors $b^1, b^2,..., b^d$. Each such basis determines a set of d basic constants $K(c, b^j)$, $j = 1, 2,..., d$, and a set of d basic functions $F(\cdot, b^j)$, $j = 1, 2,..., d$, that give rise to a corresponding set of d *equilibrium equations*

$$F(\delta, b^j) = K(c, b^j), \qquad j = 1, 2,..., d, \tag{54}$$

which (according to Corollary 4.2) must be satisfied by each dual equilibrium solution that satisfies the δ^* zero-condition. From the construction of the nullity vectors $b^1, b^2,..., b^d$, we know that $n - d$ linearly independent equations can be selected from the normality condition, the orthogonality conditions, and the δ^* zero-condition. Such a selection will always contain the normality condition; otherwise, the elementary theory of linear algebra would imply that the dual feasible solution δ^* does not exist. Each such set of $n - d$ linear equations and the d nonlinear equilibrium equations (54) provide n necessary conditions to help determine the n components of δ^*. However, the last part of Section 4 shows that these n necessary conditions are not always sufficient in that they may have solutions that are not dual equilibrium solutions. A conclusive test for a given solution consists of showing the existence or nonexistence of a corresponding primal equilibrium solution t^*.

Before elaborating on a method for constructing corresponding primal equilibrium solutions, it is important to note that not all dual equalibrium solutions need satisfy our n conditions; but those that do not must satisfy a different set of n conditions obtained from equating different component *blocks* of δ^* to zero. Only complete component blocks of δ^* are equated to zero because of the necessary condition given in conclusion (i) of Theorem 4.1. The determination of all such sets of n conditions is, of course, an elementary (but lengthy) task in combinatorics and linear algebra.

From a computational point of view, it is worth remarking that the only nonlinear equations in a given set are the d equilibrium equations (54), so such a set is said to have *degree of difficulty d*. It may be worth noting that the equilibrium equations are actually linear in the variables $\log \delta_i$ and $\log \lambda_k$ when the logarithm of both sides of these equations is taken. Furthermore, the resulting equations are linear in the parameters $\log c_i$, so the family of all reversed geometric programs with a fixed exponent matrix $[a_{ij}]$ and a given dual equilibrium solution δ^* can be found by constructing the general solution of this linear system. Of course, not every reversed geometric program constructed in this manner need have δ^* as a dual equilibrium solution, but those and only those programs for which there is a corresponding primal equilibrium solution t^*.

To obtain the primal equilibrium solutions corresponding to a given dual equilibrium solution δ^* for fixed coefficients c_i, Corollary 4.1 shows that $v(\delta^*)$ can be substituted for $g_0(t^*)$ into the equilibrium conditions (20-1) for the corresponding primal equilibrium solutions t^*. After making this substitution, one can produce a linear system in $\log t_j^*$ by taking the logarithm of both sides of the resulting conditions and those other equilibrium conditions (20-2) for which $\lambda_k(\delta^*) > 0$. A summation of these latter conditions shows that $\lambda_k(\delta^*) = \lambda_k(\delta^*)\, g_k(t^*)$, so $g_k(t^*) = 1$, and hence the corresponding prototype and reversed primal constraints are automatically satisfied by each such solution t^*; but the other primal constraints [for which $\lambda_k(\delta^*) = 0$] need not be satisfied by such a solution t^*.

Consequently, the primal equilibrium solutions t^* corresponding to a given solution δ^* of the n necessary conditions that result from equating certain component blocks to zero are readily characterized as the solutions to a system of linear equations and (usually) nonlinear inequalities. (Those inequalities arising from single-term posynomial constraints are clearly linear.) Of course, this system may not have a solution, in which case the vector δ^* is not a dual equilibrium solution even though it satisfies the appropriate n necessary conditions. However, Theorem 4.3 shows that such a vector δ^* with its zero components deleted is always a dual equilibrium solution to the dual of the primal program that results from deleting those primal constraints for which $\lambda_k(\delta^*) = 0$.

The presence of nonlinear equations and frequently nonlinear inequalities is the main difficulty with using the preceding indirect method for finding equilibrium solutions. In Ref. 17, we develop direct methods based on solving appropriate sequences of prototype geometric programs.

Finally, it is worth mentioning that when reversed constraints are not present (that is, $R = \varnothing$) and when the primal objective function has only a single term (that is, $[0] = \{1\}$), dual program B is essentially the *chemical equilibrium problem* that consists of minimizing *Gibbs' free energy function* $-\log v(\delta)$ subject to the *mass balance equations*

$$\sum_{i=2}^{n} a_{ij}\delta_i = -a_{1j}, \qquad j = 1, 2,..., m,$$

to obtain the *equilibrium mole fraction* δ_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*. In this context, the nullity vector components are called *stoichiometric coefficients*, the basic constants are termed *equilibrium constants*, and the equilibrium equations are known as the *mass action laws*. For complete details, see Ref. 26, Appendix C of Ref. 1, and the references cited therein.

6. Appendix

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 Section 2.

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

$$\sqrt{\{[\sqrt{[f_1(t)]} + f_3(t)]/[\sqrt{[f_2(t)]} + f_4(t)]\}}, \tag{55}$$

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

$$0 \leqslant f_1(t), \tag{56}$$

$$0 \leqslant f_2(t). \tag{57}$$

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

$$0 \leqslant \sqrt{[f_1(t)]} + f_3(t), \tag{58-1}$$

$$0 \leqslant \sqrt{[f_2(t)]} + f_4(t), \tag{59-1}$$

or the constrainsts

$$\sqrt{[f_1(t)]} + f_3(t) \leqslant 0, \tag{58-2}$$

$$\sqrt{[f_2(t)]} + f_4(t) \leqslant 0. \tag{59-2}$$

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_1 with constraints (58-1) and (59-1) and the program P_2 with constraints (58-2) and (59-2), 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_1 whose consistency we shall assume.

To test for the possible occurrence of the interminate form $\sqrt{(0/0)}$, we should first minimize just the numerator $\sqrt{[f_1(t)]} + f_3(t)$ subject, of course, to the constraints (56)–(57), (58-1), and (59-1). This program P_1' has an optimal value that is either zero or positive by virtue of constraint (58-1). If it is zero, then constraint (59-1) shows that either there is a minimizing sequence such that the denominator $\sqrt{[f_2(t)]} + f_4(t)$ is bounded from below by a positive number, or $\sqrt{[f_2(t)]} + f_4(t)$ approaches zero from above for each minimizing sequence. In the first case, the optimal value of program P_1 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 shouldn't arise when the original program P is propertly formulated. The remaining possibility is that the optimal value for program P_1' 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_1.

Before proceeding, we should observe that program P_1' 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_1.

Introducing an additional positive independent variable t_0, we see that program P_1 consists essentially of minimizing the posynomial

$$\sqrt{t_0} \tag{60}$$

subject to both the constraints (56), (57), (58-1), (59-1) and the additional algebraic constraint $\{\sqrt{[f_1(t)]} + f_3(t)\}/\{\sqrt{[f_2(t)]} + f_4(t)\} \leqslant t_0$, which can conveniently be rewritten as

$$0 \leqslant -\sqrt{[f_1(t)]} + t_0 \sqrt{[f_2(t)]} - f_3(t) + t_0 f_4(t), \tag{61}$$

by virtue of constraint (59-1). To achieve our goal, we must still transform the algebraic functions in constraints (58-1), (59-1), and (61) into signomials. Toward that end, we introduce two additional positive independent variables t_{m+1} and t_{m+2} so that (58-1) and (59-1) can be replaced by

$$0 \leqslant \sqrt{t_{m+1}} + f_3(t), \tag{62-1}$$

$$t_{m+1} \leqslant f_1(t), \tag{62-2}$$

and

$$0 \leqslant \sqrt{t_{m+2}} + f_4(t), \tag{63-1}$$

$$t_{m+2} \leqslant f_2(t). \tag{63-2}$$

Finally, we introduce another positive independent variable t_{m+3} so that (61) can be replaced by

$$0 \leqslant -\sqrt{t_{m+3}} + t_0 \sqrt{t_{m+2}} - f_3(t) + t_0 f_4(t), \tag{64-1}$$

$$f_1(t) \leqslant t_{m+3}\,. \tag{64-2}$$

Thus, program P_1 actually reduces to minimizing the posynomial (60) subject to the signomial constraints (56)–(57) and (62)–(64). This program is obviously a signomial program, and hence can be further transformed into a posynomial program with the aid of the techniques given in Section 2.

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

1. Duffin, R. J., Peterson, E. L., and Zener, C., *Geometric Programming—Theory and Application*, John Wiley and Sons, New York, 1967.
2. Zener, C., *A Mathematical Aid in Optimizing Engineering Designs*, Proceedings of the National Academy of Sciences of the USA, Vol. 47, No. 4, 1961.
3. Zener, C., *A Further Mathematical Aid in Optimizing Engineering Designs*, Proceedings of the National Academy of Sciences of the USA, Vol. 48, No. 4, 1962.
4. Avriel, M., and Wilde, D. J., *Optimal Condenser Design by Geometric Programming*, Stanford University, Chemical Engineering Report, 1966.
5. Federowicz, A. J., and Mazumdar, M., *The Use of Geometric Programming to Maximize Reliability Achieved by Redundancy*, ORSA, Vol. 16, No. 5, 1968.
6. Wilde, D. J., and Beightler, C. S., *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.
7. Passy, U., and Wilde, D. J., *Generalized Polynomial Optimization*, SIAM Journal of Applied Mathematics, Vol. 15, No. 5, 1967.
8. Blau, G., and Wilde, D. J., *Second Order Characterization of Generalized Polynomial Programs*, Paper presented at the Princeton International Symposium on Mathematical Programming, 1967.
9. Avriel, M., and Williams, A. C., *Complementary Geometric Programming*, SIAM Journal on Applied Mathematics, Vol. 19, No. 1, 1970.
10. Broverman, A., Federowicz, A. J., and McWhirter, J., personal communication, 1966.
11. Pascual, L. D., and Ben-Israel, A., *Constrained Maximization of Posynomials by Geometric Programming*, Journal of Optimization Theory and Applications, Vol. 5, No. 2, 1970.
12. Passy, U., *Condensing Generalized Polynomials*, Journal of Optimization Theory and Applications (to appear).
13. Gale, D., Kuhn, H. W., and Tucker, A. W., *Linear Programming and the Theory of Games*, Cowles Commission Monograph No. 13, 1951.
14. Goldman, A. J., and Tucker, A. W., *Theory of Linear Programming*, Linear Inequalities and Related Systems, Edited by H. W. Kuhn and A. W. Tucker, Princeton University Press, Princeton, New Jersey, 1956.
15. Duffin, R. J., *Linearizing Geometric Programs*, SIAM Review, Vol. 12, No. 2, 1970.
16. Duffin, R. J., and Peterson, E. L., *Duality Theory for Geometric Programming*, SIAM Journal on Applied Mathematics, Vol. 14, No. 6, 1966.
17. Duffin, R. J., and Peterson, E. L., *Reversed Geometric Programs Treated by Harmonic Means*, Indiana University Mathematical Journal (to appear).

18. Charnes, A., and Cooper, W. W., *A Convex Approximant Method for Nonconvex Extensions of Geometric Programming*, Proceedings of the National Academy of Sciences of the USA, Vol. 56, No. 5, 1966.
19. Peterson, E. L., *Symmetric Duality for Generalized Unconstrained Geometric Programming*, SIAM Journal on Applied Mathematics, Vol. 19, No. 3, 1970.
20. Peterson, E. L., *Generalization and Symmetrization of Duality in Geometric Programming* (to appear).
21. Rockafellar, R. T., *Duality and Stability in Extremum Problems Involving Convex Functions*, Pacific Journal of Mathematics, Vol. 21, No. 1, 1967.
22. Rockafellar, R. T., *Duality in Nonlinear Programming*, Mathematics of the Decision Sciences, Part 1, Edited by G. B. Dantzig and A. F. Veinott, American Mathematical Society, Providence, Rhode Island, 1968.
23. Rockafellar, R. T., *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.
24. Mangasarian, O. L., *Nonlinear Programming*, McGraw-Hill Book Company, New York, 1969.
25. Zangwill, W. I., *Nonlinear Programming—A Unified Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
26. Passy, U., and Wilde, D. J., *A Geometric Programming Algorithm for Solving Chemical Equilibrium Problems*, SIAM Journal on Applied Mathematics, Vol. 16, No. 2, 1968.