JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 66, No. 3, SEPTEMBER 1990

Posynomial Geometric Programming as a Special Case of Semi-Infinite Linear Programming

J. RAJGOPAL[1] AND D. L. BRICKER[2]

Communicated by M. Avriel

Abstract. This paper develops a wholly linear formulation of the posynomial geometric programming problem. It is shown that the primal geometric programming problem is equivalent to a semi-infinite linear program, and the dual problem is equivalent to a generalized linear program. Furthermore, the duality results that are available for the traditionally defined primal-dual pair are readily obtained from the duality theory for semi-infinite linear programs. It is also shown that two efficient algorithms (one primal based and the other dual based) for geometric programming actually operate on the semi-infinite linear program and its dual.

Key Words. Geometric programming, semi-infinite linear programming, duality, generalized linear programming.

1. Introduction

Since its inception by Duffin and Zener in 1961–62, geometric programming (GP) has undergone a number of developments to emerge as an important method for the analysis of algebraic nonlinear programming problems. While the primal GP problem is wholly nonlinear, the associated dual GP problem has linear constraints and a concave objective function.

This paper develops an alternative structure for the GP primal-dual pair. It is shown that the nonlinear primal GP problem is equivalent to a semi-infinite linear program (SILP), i.e., a linear program with a finite number of decision variables, but infinitely many constraints. The dual GP

[1] Assistant Professor, Department of Industrial Engineering and Operations Research, University of Pittsburgh, Pittsburgh, Pennsylvania.

[2] Associate Professor, Department of Industrial and Management Engineering, University of Iowa, Iowa City, Iowa.

0022-3239/90/0900-0455$06.00/0

problem thus becomes a generalized linear program (GLP), i.e., a linear program with a finite number of constraints, but with columns that are restricted to convex sets. In the case of geometric programming, these columns are restricted to be chosen from the set of extreme points of the convex sets mentioned above, so that the dual problem then becomes a linear program with infinitely many decision variables and a finite number of constraints.

The primary advantage of this structure lies in the complete linearity of the constraints and the objective function. For the dual problem, only a small fraction of the infinitely many decision variables are going to be positive at the optimum. Similarly, for the primal problem only a small fraction of the infinitely many constraints are going to be active at the optimum. This immediately suggests a column-generation procedure for the dual, or equivalently a cutting plane procedure for the primal. Extensions of Kelley's cutting plane method (Ref. 1) for convex programs to the case of semi-infinite programming have been well documented (e.g., Ref. 2). One of the best primal based algorithms (Ref. 3) can also be viewed as a cutting plane method for the semi-infinite program mentioned above. Similarly, a column generation algorithm that works on the generalized linear programming dual (Ref. 4) is also very efficient at overcoming problems traditionally associated with most dual-based procedures, such as computational difficulties, recovery of the primal optima from the dual solutions, and the need to solve subsidiary problems. Furthermore, it would appear that, with the current trends in new linear programming methods, there would be opportunities to develop efficient algorithms for posynomial GP that work on this linear formulation.

2. Primal–Dual Pair in Geometric Programming

We first introduce the GP primal-dual pair. The primal geometric programming problem is defined as follows:

$$\text{(GPP)} \quad \text{minimize} \quad g_0(x), \tag{1}$$

$$\text{s.t.} \quad g_k(x) \leq 1, \qquad k = 1, 2, \ldots, p, \tag{2a}$$

$$x_j > 0, \qquad j = 1, 2, \ldots, m, \tag{2b}$$

$$\text{where} \quad g_k(x) = \sum_{i \in [k]} c_i \prod_{j=1}^{n} x_j^{a_{ij}}, \qquad k = 0, 1, \ldots, p. \tag{2c}$$

The index set I numbers the n terms in the $p+1$ posynomials, and the

index subset $[k]$ numbers the terms in posynomial k,

$$I = \{1, 2, \ldots, n\} = \bigcup_{k=0}^{p} [k], \tag{3a}$$

$$[k] \cap [l] = \varnothing, \qquad k \neq l, \tag{3b}$$

and we assume that $c_i > 0$ for all i. Each function g_k is a generalized polynomial, in that the exponents a_{ij} need not be positive integers but may be any real numbers, and a posynomial because each coefficient c_i is positive.

The corresponding geometric programming dual problem is defined as follows:

$$\text{(GPD)} \qquad \text{maximize} \quad v(\lambda, \delta) = \prod_{i \in I} (c_i/\delta_i)^{\delta_i} \prod_{k=1}^{p} \lambda_k^{\lambda_k}, \tag{4}$$

$$\text{s.t.} \quad \sum_{i \in [k]} \delta_i = \lambda_k, \qquad k = 0, 1, \ldots, p, \tag{5a}$$

$$\sum_{i \in I} a_{ij} \delta_i = 0, \qquad j = 1, 2, \ldots, m, \tag{5b}$$

$$\delta_i \geq 0, \qquad i = 1, 2, \ldots, n, \tag{5c}$$

$$\lambda_0 = 1, \lambda_k \geq 0, \qquad k = 1, 2, \ldots, p. \tag{5d}$$

Unlike the primal program, this program is linearly constrained. As an alternative to maximizing $v(\lambda, \delta)$, one could maximize $\log v(\lambda, \delta)$, since the log function is monotone increasing. This log-dual objective is given by

$$\log v(\lambda, \delta) = \sum_{i \in I} (\delta_i \log c_i - \delta_i \log \delta_i) + \sum_{k=1}^{p} \lambda_k \log \lambda_k. \tag{6}$$

This function has the attractive feature of being separable. Moreover, it is a concave function of δ over the dual feasible region if each λ_k is replaced by $\sum \delta_i$ (Ref. 5). Several algorithms have been developed that make use of this property in order to solve the dual and then compute the primal solution. The duality theory for the above GP primal–dual pair has been well documented, and complete results have been provided (Ref. 5).

Duffin (Ref. 6) was the first to present a linearization of the GP primal problem. In order to do this, he introduced the concept of condensation, where a posynomial function is underestimated by a monomial (a posynomial with a single term). Given the original GP posynomial program and a set of nonnegative weights $\rho_1, \rho_2, \ldots, \rho_n$ such that

$$\sum_{i \in [k]} \rho_i = 1, \qquad k = 0, 1, \ldots, p, \tag{7}$$

the condensed program is obtained by replacing each function $g_k(x)$ with a new function $\bar{g}_k(x)$, where

$$\bar{g}_k(x) = \prod_{i\in[k]} \left\{ c_i \prod_{j=1}^{m} x_j^{a_{ij}} \Big/ \rho_i \right\}^{\rho_i}. \tag{8}$$

Let

$$\prod_{i\in[k]} \{c_i/\rho_i\}^{\rho_i} = \bar{c}_k, \tag{9}$$

$$\sum_{i\in[k]} \rho_i a_{ij} = \bar{a}_{kj}. \tag{10}$$

Then, we have

$$\bar{g}_k(x) = \bar{c}_k \prod_{j=1}^{m} (x_j)^{\bar{a}_{kj}},$$

a monomial for $k = 0, 1, \ldots, p$. By the arithmetic-geometric mean inequality,

$$\bar{g}_k(x) \leq g_k(x), \tag{11}$$

so that the feasible region for the original primal problem is now contained entirely within that of the condensed program. If we now make the transformations

$$\log x_j = z_j, \tag{12}$$

$$\log \bar{c}_k = C_k, \tag{13}$$

we obtain the following linear program:

$$\text{(LP)} \qquad \text{minimize} \quad C_0 + \sum_{j=1}^{m} \bar{a}_{0j} z_j, \tag{14a}$$

$$\text{s.t.} \qquad \sum_{j=1}^{m} \bar{a}_{kj} z_j \leq -C_k, \qquad k = 1, 2, \ldots, p. \tag{14b}$$

Obviously, if we had a vector x which was feasible in the original primal GP problem, and chose a set of nonnegative weights $\rho_1, \rho_2, \ldots, \rho_n$ according to (7), and made the transformations (12) and (13), we could obtain a point z satisfying (14a). The converse is, however, not necessarily true; i.e., any solution to (14a) would not necessarily generate a solution vector x which is feasible in the original problem. Rather, this depends on the choice of the weights $\rho_1, \rho_2, \ldots, \rho_n$ which are used to arrive at the linearized program. In fact, Duffin shows that, if x^* is the optimal solution to the original problem, the linearized program with weights given by

$$\rho_i = c_i \prod_{j=1}^{m} (x_j^*)^{a_{ij}} \Big/ \left[\sum_{w\in[k]} c_w \prod_{j=1}^{m} (x_j^*)^{a_{wj}} \right], \qquad i \in [k], \tag{15}$$

will give rise to a solution z^* to the linearized program and also generate a feasible solution to the original problem; he also shows that

$$\exp\left(C_0 + \sum_{j=1}^{m} \bar{a}_{0j} z_j^*\right) = \bar{g}_0(x^*) = g_0(x). \tag{16}$$

Thus, each linearized program (14) is an approximation of the original program. The GGP algorithm of Avriel, Dembo, and Passy (Ref. 3) makes use of the above principle to solve posynomial GP problems; starting with an initial set of weights, it successively refines the feasible region of (14) until it obtains an optimal solution with the weights given by (15). This would then generate an optimal solution to the original problem. Dinkel, Elliott, and Kochenberger (Ref. 7) also make use of this concept of linearization, but include bounds on the variables in their development; their algorithmic procedure is quite similar to that of Avriel et al (Ref. 3).

The above linearization provides only an approximation of the original feasible region; in fact each set of weights ρ_i, $i = 1, 2, \ldots, n$, provides one such approximation, and the algorithms seek the approximation that is sufficient to determine an optimal solution to the original problem. If all such approximations were to be considered, then the intersection of the various feasible regions would lead to the feasible region of the original problem. The combination of all possible approximations is equivalent to a linear program with infinitely many constraints. This suggests a semi-infinite linear program that is equivalent to the original posynomial GP primal problem.

In the next section, we define a semi-infinite linear program and present some of the important results for the same. These results have been well documented and are therefore presented with references but without explicit proofs. In the succeeding section, we present the development of the GP problem as a semi-infinite LP and use the theoretical results for the latter to prove duality relationships for the GP problem.

3. Semi-Infinite Linear Programming

Semi-infinite linear programming (SILP) problems have a fixed number of decision variables, but infinitely many constraints. An excellent reference on SILP is Glashoff and Gustafson (Ref. 8). In its most general form, the primal SILP problem may be stated as follows:

$$\text{(P)} \qquad \text{minimize} \quad \sum_{r=1}^{n} c_r y_r, \tag{17a}$$

$$\text{s.t.} \quad \sum_{r=1}^{n} a_r(s) y(r) \geq b(s), \qquad s \in \mathbb{S}, \tag{17b}$$

where $c=(c_1, c_2, \ldots, c_n)$ is a fixed vector in $\mathbb{R}^n$ and $\mathbb{S}$ is a parameter set with infinitely many members. Each s in $\mathbb{S}$ determines a constraint for (17b) above. The set of feasible vectors $y=(y_1, y_2, \ldots, y_n)$ is thus the intersection of infinitely many half-spaces, and each s in $\mathbb{S}$ determines a vector $a(s)=[a_1(s), a_2(s), \ldots, a_n(s)]$ in $\mathbb{R}^n$ and a real number $b(s)$.

Associated with this program is a dual program which may be stated as follows:

(D) Determine a finite subset $(s_1, s_2, \ldots, s_q) \subset \mathbb{S}$ and nonnegative numbers $(x_1, x_2, \ldots, x_q)$ so as to

$$\text{minimize} \quad \sum_{i=1}^{q} x_i b(s_i), \tag{18a}$$

$$\text{s.t.} \quad \sum_{i=1}^{q} x_i a_r(s_i) = c_r, \qquad r=1, 2, \ldots, n, \tag{18b}$$

$$x_i \geq 0, \qquad i=1, 2, \ldots, q. \tag{18c}$$

The vector $(s_1, s_2, \ldots, s_q, x_1, x_2, \ldots, x_q)$ is said to be feasible in program D when $s_i \in \mathbb{S}$, $i=1, 2, \ldots, q.$ and the constraints (18b) and (18c) are satisfied. It should be noted at this point that program D is a nonlinear problem in the variables $(s_1, s_2, \ldots, s_q)$ which can be transformed into a linear problem only if the index set $\mathbb{S}$ is finite (Ref. 9). Let $v(\mathrm{P})$ and $v(\mathrm{D})$ denote the values of programs P and D respectively, if they exist. It is then easy to prove the following weak duality theorem.

Theorem 3.1. $v(\mathrm{D}) \leq v(\mathrm{P})$.

We next provide some preliminary definitions required for the statements of the succeeding theorems in this section.

Definition 3.1. If A is any arbitrary set in $\mathbb{R}^n$, then the convex conic hull of A, denoted by $\mathrm{CC}(A)$, is the smallest convex cone containing the convex hull of the set A. It is readily seen that $\mathrm{CC}(A)$ is the set of all nonnegative linear combinations of elements of the set A, i.e.,

$$\mathrm{CC}(A) = \left\{ \sum_{i=1}^{q} x_i a_i,\ x_i \geq 0,\ a_i \in A,\ i=1, 2, \ldots, q,\ q \geq 1 \right\}. \tag{19}$$

Definition 3.2. The convex conic hull $\mathrm{CC}(A_{\mathbb{S}})$ if the set

$$A_{\mathbb{S}} = \{a(s) \mid s \in \mathbb{S}\} \subset \mathbb{R}^n \tag{20}$$

is called the moment cone of program P and is denoted by M_n. It may now be readily seen that $(s_1, s_2, \ldots, s_q)$ is feasible in D if and only if the vector c lines in M_n.

Next, we denote the set of vectors $[b(s), a_1(s), a_2(s), \ldots, a_n(s)]$ from $\mathbb{R}^{n+1}$ by $\tilde{A}_{\mathbb{S}}$; i.e., to each vector in $A_{\mathbb{S}}$, we append the appropriate value of $b(s)$.

Definition 3.3. The moment cone M_{n+1} of program P is defined as the convex conic hull of $\tilde{A}_{\mathbb{S}}$, i.e.,

$$M_{n+1} = \mathrm{CC}(\tilde{A}_{\mathbb{S}}).$$

Definition 3.4. Program P is said to be superconsistent if there exists a vector $y = (y_1, y_2, \ldots, y_n)$ in $\mathbb{R}^n$ such that

$$\sum_{r=1}^{n} a_r(s) y_r > b(s), \qquad \text{for all } s \in \mathbb{S}.$$

Theorem 3.2. If program P is superconsistent and if the set $\mathbb{S}$ is a compact subset of $\mathbb{R}^n$ with real-valued functions $a_1, a_2, \ldots, a_n, b$ which are all continuous on $\mathbb{S}$, then the moment cone M_{n+1} is closed.

We now state an important theorem regarding the solvability of program D.

Theorem 3.3. If M_{n+1} is closed and program D is feasible with the solution being bounded from above, then D has a solution.

Theorem 3.2 provides us with conditions to verify that M_{n+1} is indeed closed, while Theorem 3.3 states that, if D is feasible and bounded from above and if the conditions of Theorem 3.2 are met, then D has an explicit solution. In fact, if the conditions of Theorem 3.2 are met, D is automatically bounded from above if it is feasible (by virtue of Theorem 3.1). The feasibility of program D can be established by means of Haar's extension of the Farkas lemma for the case of semi-infinite systems (Ref. 10). Several versions of this have been stated and proved (e.g., Refs. 9 and 11). This leads to the two strong duality theorems for semi-infinite linear programs.

Theorem 3.4. Suppose that the following conditions are met for the primal-dual pair P-D:

(i) $\mathbb{S}$ is a compact set in $\mathbb{R}^n$ and the functions $a_1, a_2, \ldots, a_n, b$ are all continuous on $\mathbb{S}$;
(ii) P is superconsistent;
(iii) $v(\mathrm{P})$ is finite.

Then, program D has a solution, and $v(\mathrm{P}) = v(\mathrm{D})$.

Theorem 3.4 thus establishes a sufficient condition for D to have a solution; moreover, when these conditions are met, the optimal value of

the dual problem at this solution equals that of the primal problem. The following theorem establishes similar conditions for the solvability of program P.

Theorem 3.5. Suppose that the following conditions hold for the pair P-D:

(i) $v(\mathrm{D})$ is finite;
(ii) the vector $c=(c_1, c_2, \ldots, c_n)$ lies in the interior of the set M_{n+1}.

Then, program P is solvable, and moreover $v(\mathrm{P})=v(\mathrm{D})$.

4. GP as a Special Case of SILP

We now proceed to rewriting the GP primal problem (GPP) as a semi-infinite linear program. We also make use of the duality results of Section 3 to establish primal-dual relationships. The relationship between geometric programming and semi-infinite linear programming was first introduced by Gochet, Smeers, and Kortanek in 1973 (Ref. 12), but there does not seem to be any evidence of further work since then. Gochet *et al.* obtained a semi-infinite programming version of the GP primal by first restating the primal as a convex program and then replacing each nonlinear constraint by the set of all its supporting hyperplanes and adding some limiting constraints. The formulation that follows in this section is along similar lines, but displays the linearity more readily and provides a simple derivation of duality results for canonical programs.

Before stating the GP problem as a semi-infinite linear program, we make the following definitions:

$$Q_k=\left\{\rho\in\mathbb{R}^n \;\middle|\; \rho_i=0,\ i\notin[k],\ \sum_{i\in[k]}\rho_i=1,\ \rho_i\geq 0\right\}, \tag{21}$$

$$A_{kj}(\rho)=\sum_{i=1}^{n} a_{ij}\rho_i, \qquad \rho\in Q_k, \tag{22a}$$

$$A_{kj}(\rho)=\sum_{i\in[k]} a_{ij}\rho_i, \qquad \rho\in Q_k, \tag{22b}$$

$$G_k(\rho)=\sum_{i=1}^{n} \rho_i \log(c_i/\rho_i), \qquad \rho\in Q_k, \tag{23a}$$

$$G_k(\rho)=\sum_{i\in[k]} \rho_i \log(c_i/\rho_i), \qquad \rho\in Q_k. \tag{23b}$$

The primal problem may then be stated as follows:

$$\text{(A)} \quad \text{minimize} \quad \pi_0, \tag{24a}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} \pi_j A_{kj}(\rho) \geq G_k(\rho), \qquad \rho \in Q_k,$$

$$k = 1, 2, \ldots, p, \tag{24b}$$

$$\sum_{j=1}^{m} \pi_j A_{0j}(\rho) + \pi_0 \geq G_0(\rho), \qquad \rho \in Q_0. \tag{24c}$$

It is readily seen that this is a semi-infinite linear program. The $p+1$ sets $Q_0, Q_1, \ldots, Q_p$ assume the role of the set $\mathbb{S}$ defined in Section 3, and each vector ρ corresponds to a value of parameter s. The constraint functions in (24b) and (24c) for a given k may be interpreted as the set of infinitely many supporting hyperplanes for a convex reformulation of the function g_k. It should also be noted that Q_k is not a part of the feasible region of program A; it is a parameter set used to generate the supporting hyperplanes for the constraint k. Each vector from Q_k generates one such hyperplane, and each Q_k thus generates infinitely many constraints in program A.

We now state and prove a theorem that shows that this program does indeed determine a solution to the original GP primal (program GPP).

Theorem 4.1. Suppose that the GP primal program (program GPP) is consistent and attains its constrained minimum at a point satisfying the primal constraints. Then, the vector π given by

$$-\infty < \pi_0 = \log g_0(x) < \infty,$$

$$-\infty < -\pi_j = \log x_j < \infty, \qquad j = 1, 2, \ldots, m,$$

is optimal for program A if and only if x is optimal for program GPP.

Proof. First, suppose that the vector x satisfying the above relationships is optimal for program GPP. Then, since x is feasible in GPP, we have, for $k = 1, 2, \ldots, p$,

$$\sum_{i \in [k]} c_i \prod_{j=1}^{m} x_j^{a_{ij}} \leq 1,$$

implying that

$$\sum_{i \in [k]} c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij} \right) \leq 1,$$

which implies, by virtue of the arithmetic-geometric mean inequality, that

$$\prod_{i\in[k]} \left\{ c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij} \right) \Big/ \rho_i \right\}^{\rho_i} \leq 1, \qquad \rho \in Q_k.$$

Therefore,

$$\sum_{i\in[k]} \rho_i \log\left[c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij} \right) \Big/ \rho_i \right] \leq 0, \tag{25a}$$

$$\sum_{i\in[k]} \rho_i \log(c_i/\rho_i) \leq \sum_{i\in[k]} \rho_i \sum_{j=1}^{m} \pi_j a_{ij}, \tag{25b}$$

$$\sum_{j=1}^{m} \pi_j A_{kj}(\rho) \geq G_k(\rho), \qquad \rho \in Q_k. \tag{25c}$$

Similar to (25), for $k=0$ we have

$$\sum_{i\in[0]} c_i \sum_{j=1}^{m} x_j^{a_{ij}} = g_0(x) = \exp(\pi_0),$$

which implies that

$$\prod_{i\in[0]} \left\{ c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij} \right) \Big/ \rho_i \right\}^{\rho_i} \leq \exp(\pi_0), \qquad \rho \in Q_0.$$

Therefore,

$$\sum_{i\in[0]} \rho_i \log\left[c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij} \right) \Big/ \rho_i \right] \leq \pi_0, \tag{26a}$$

$$\sum_{i\in[0]} \rho_i \log(c_i/\rho_i) \leq \pi_0 + \sum_{i\in[0]} \rho_i \sum_{j=1}^{m} \pi_j a_{ij}, \tag{26b}$$

$$\sum_{j=1}^{m} \pi_j A_{0j}(\rho) + \pi_0 \geq G_0(\rho), \qquad \rho \in Q_0. \tag{26c}$$

From (25c) and (26c), π is feasible in program A. To show that it is also optimal, suppose that the optimal solution is attained at some other $\pi_0' < \pi_0$. Then,

$$\sum_{j=1}^{m} \pi_j A_{kj}(\rho) \geq \begin{cases} G_k(\rho), & k \geq 1, \\ G_k(\rho) - \pi_0', & k = 0, \end{cases}$$

for all $\rho \in Q_k$, $k = 0, 1, \ldots, p$. Then, (25a) and (26a) hold for $\pi = \pi'$, and we have

$$\prod_{i\in[k]} \left\{ c_i \exp\left(\sum_{j=1}^{m} -\pi_j' a_{ij} \right) \Big/ \rho_i \right\}^{\rho_i} \leq \begin{cases} 1, & k \geq 1, \\ \exp(\pi_0'), & k = 0, \end{cases} \tag{27}$$

for all $\rho \in Q_k$. In particular, consider the vector ρ' for which

$$\rho_i' = c_i \exp\left(\sum_{j=1}^{m} -\pi_j' a_{ij}\right) \Big/ \left[\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j' a_{wj}\right)\right], \qquad i \in [k]. \tag{28}$$

From (27) and (28), we have

$$\prod_{i\in[k]} \left\{\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j' a_{wj}\right)\right\}^{\rho_i} \leq \begin{cases} 1, & k \geq 1, \\ \exp(\pi_0'), & k = 0. \end{cases}$$

Then since

$$\sum_{i\in[k]} \rho_i' = 1,$$

we have

$$\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j' a_{wj}\right) \leq \begin{cases} 1, & k \geq 1, \\ \exp(\pi_0'), & k = 0. \end{cases}$$

Letting

$$\exp(-\pi_j') = x_j',$$

we have

$$\sum_{i\in[k]} c_i \sum_{j=1}^{m} x_j'^{a_{ij}} \leq \begin{cases} 1, & k \geq 1, \\ \exp(\pi_0'), & k = 0. \end{cases}$$

Therefore, x' is feasible in program GPP and

$$g_0(x') = \sum_{i\in[0]} c_i \sum_{j=1}^{m} x_j'^{a_{ij}} \leq \exp(\pi_0') < \exp(\pi_0), \qquad \text{since } \pi_0' < \pi_0.$$

But

$$\exp(\pi_0) = g_0(x).$$

Therefore,

$$g_0(x') < g_0(x),$$

which contradicts the initial assumption that x is optimal for program GPP. Therefore, π is optimal for program A.

In order to prove the second clause, suppose that π is optimal for program A. Then, (25) and (26) hold. Further, let us define the vector

$$\rho_i = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \Big/ \left[\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right], \qquad i \in [k].$$

Applying the arithmetic-geometric mean inequality to (25a) and (26a), we see that

$$x_j = \exp(-\pi_j)$$

is feasible in program GPP.

Suppose that this x is not optimal and that the optimal vector is some other x'. Then,

$$g_0(x') < g_0(x).$$

Letting

$$\pi'_j = \log x'_j, \qquad j = 1, 2, \ldots, m,$$

$$\pi'_0 = \log g_0(x'),$$

relationships similar to (25c) and(26c) yield

$$\sum_{j=1}^{m} \pi'_j A_{kj}(\rho) \geq G_k(\rho), \qquad \rho \in Q_k,$$

$$\sum_{j=1}^{m} \pi'_j A_{0j}(\rho) \geq G_0(\rho) - \pi'_0, \qquad \rho \in Q_0.$$

Therefore, π' is feasible in program A and

$$\pi'_0 = \log g_0(x') < \log g_0(x) = \pi_0, \qquad \text{since } g_0(x') < g_0(x).$$

This contradicts the initial assumption that π_0 is optimal for program A. Hence, the above vector x must be optimal for program GPP. This completes the proof for Theorem 4.1. □

In summary, the above theorem states that, given a finite optimal solution π to program A, one could use the relationship

$$x_j = \exp(-\pi_j) \tag{29}$$

to recover an optimal solution to program GPP, the original GP primal problem. We now look at the dual program corresponding to program A.

Recall that the SILP dual (program D) was to find a finite subset $(s_1, s_2, \ldots, s_q)$ of the set $\mathbb{S}$ and the nonnegative numbers $(x_1, x_2, \ldots, x_q)$ so as to

$$\text{maximize} \quad \sum_{i=1}^{q} x_i b(s_i),$$

$$\text{s.t.} \quad \sum_{i=1}^{q} x_i a_r(s_i) = c_r, \qquad r = 1, 2, \ldots, n.$$

Although the number q could be arbitrarily large, we could reduce q to $n+1$ by means of the reduction theorem (Ref. 8) and rewrite the dual as

$$\text{maximize} \quad \sum_{i=1}^{n+1} x_i b(s_i),$$

$$\text{s.t.} \quad \sum_{i=1}^{n+1} x_i a_r(s_i) = c_r, \qquad r = 1, 2, \ldots, n,$$

$$s_i \in \mathbb{S} \text{ and } x_i \geq 0, \qquad i = 1, 2, \ldots, n+1.$$

For our problem, let us use q points from each set Q_k so that the total number of dual variables is at least $(m+1)+1 = m+2$, i.e., $(p+1)q$ should be greater than $m+2$, implying that

$$q \geq (m+2)/(p+1).$$

If $(m+2)/(p+1) < 1$, we choose $q = 1$; otherwise, let q be the smallest integer greater than $(m+2)/(p+1)$. The dual may now be stated as follows:

$$\text{maximize} \quad \sum_{k=0}^{p} \sum_{w=1}^{q} \lambda_k G_k(\rho_w^k),$$

$$\text{s.t.} \quad \sum_{k=0}^{p} \sum_{w=1}^{q} \lambda_k A_{kj}(\rho_w^k) = 0, \qquad j = 1, 2, \ldots, m,$$

$$\sum_{w=1}^{q} \lambda_{0w} = 1, \quad \lambda_{kw} \geq 0, \qquad k = 0, 1, \ldots, p \text{ and } w = 1, 2, \ldots, q,$$

where

$$\rho_w^k = (\rho_{w1}^k, \rho_{w2}^k, \ldots, \rho_{wn}^k)$$

and

$$\rho_1^k, \rho_2^k, \ldots, \rho_q^k \in Q_k, \qquad k = 0, 1, \ldots, p.$$

However, it is easy to show that the function $G_k(\rho)$ is concave; and since this is a maximization problem, the optimal solution would never have any more than one point from each set Q_k. We may therefore always state the dual program (B) as follows:

(B) Find a set of vectors $\rho^0, \rho^1, \ldots, \rho^p$ from the sets $Q_0, Q_1, \ldots, Q_p$, respectively and the nonnegative vector $(\lambda_0, \lambda_1, \ldots, \lambda_p)$ such that

$$\sum_{k=0}^{p} A_{kj}(\rho^k)\lambda_k = 0, \qquad j = 1, 2, \ldots, m, \tag{30a}$$

$$\lambda_0 = 1, \tag{30b}$$

$$\lambda_k \geq 0, \qquad k = 1, 2, \ldots, p, \tag{30c}$$

and the function

$$\sum_{k=0}^{p} G_k(\rho^k)\lambda_k \tag{30d}$$

is maximized.

It should be noted that this program is nonlinear in $\rho^0, \rho^1, \ldots, \rho^p$. It can also be readily seen that this is a generalized linear program, as defined by Dantzig (Ref. 13), in the λ variables, with the columns being restricted to the sets

$$\Gamma_k = \{(\gamma, \alpha) \mid \gamma \in \mathbb{R}, \alpha \in \mathbb{R}^m, \gamma \leq G_k(\rho), \alpha = A_{kj}(\rho), \rho \in Q_k\},$$
$$k = 0, 1, \ldots, p.$$

In fact, as a consequence of the maximization of the objective function, it is sufficient to restrict the columns to come from the set of boundary points of the sets Γ_k. If each set Γ_k were polyhedral (unfortunately, this is not the case), we could actually rewrite the problem as a simple linear program. In this case, however, what we have is a linear program with infinitely many columns.

Consider for this pair of programs (A-B), the moment cones corresponding to the sets M_n and M_{n+1} of Section 3. Corresponding to the set $A_{\mathbb{S}}$ we define a set A_Q as

$$A_Q = (A^k(\rho) \mid \rho \in Q_k, k = 0, 1, \ldots, p) \subset \mathbb{R}^{m+1}, \tag{31a}$$

where

$$A^k(\rho) = (A_{k1}(\rho), A_{k2}(\rho), \ldots, A_{km}(\rho), \beta), \tag{31b}$$

$$\beta = \begin{cases} 0, & \text{if } k \geq 1, \\ 1, & \text{if } k = 0. \end{cases} \tag{31c}$$

A_Q is thus a set of vectors (infinitely many in number), each of which has as its elements the coefficients of $(\pi_1, \pi_2, \ldots, \pi_m, \pi_0)$ in the constraint set of program A. Similarly, we can define, corresponding to $\tilde{A}_S$ of Section 3, the set

$$\tilde{A}_Q = (\tilde{A}^k(\rho) \mid \rho \in Q_k, k = 0, 1, \ldots, p) \subset \mathbb{R}^{m+2}, \tag{32a}$$

where

$$\tilde{A}^k(\rho) = [G_k(\rho), A_k(\rho)], \qquad k = 0, 1, \ldots, p. \tag{32b}$$

Thus, a vector from $\tilde{A}_Q$ is a vector from A_Q with the corresponding $G_k(\rho)$ value appended to it. Also, as in Section 3 let

$$M_n = \mathrm{CC}(A_Q), \tag{33}$$

$$M_{n+1} = \mathrm{CC}(\tilde{A}_Q). \tag{34}$$

Theorem 4.2. Suppose that program A is superconsistent. Then, the moment cone M_{n+1} is closed.

Proof. The result is obtained by applying Theorem 3.2 and noting that:

(i) each set Q_k is a compact subset of $\mathbb{R}^n$, since each element of each vector from Q_k is in the compact set $[0, 1]$;
(ii) the functions $G_k(\rho)$ and $A_{kj}(\rho)$ are continuous on Q_k. □

We are now in a position to state and prove the duality theorems for programs A and B. The first is the weak duality theorem, and the second is the strong duality theorem.

Theorem 4.3. If $(\rho^0, \rho^1, \ldots, \rho^p, \lambda_0, \lambda_1, \ldots, \lambda_p)$ is feasible in program B and $(\pi_0, \pi_1, \ldots, \pi_m)$ is feasible in program A, then

$$\pi_0 \geq \sum_{k=0}^{p} \lambda_k G_k(\rho^k).$$

Proof. It follows directly from the weak duality theorem of Section 3 (Theorem 3.1).

Theorem 4.4. Suppose that program A is superconsistent and attains its minimum value $v(\mathrm{A})$ at a point π satisfying the constraints of program A. Then:

(i) program B is consistent and attains its maximum $v(\mathrm{B})$ at a point $(\rho^0, \rho^1, \ldots, \rho^p, \lambda_0, \lambda_1, \ldots, \lambda_p)$ satisfying the constraints of B; furthermore, $v(\mathrm{B}) = v(\mathrm{A})$;

(ii) $$\rho_i^k = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \Big/ \left[\sum_{w \in [k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right],$$

$$i \in [k],\ k = 0, 1, \ldots, p.$$

Proof. Clause (i) is easily proved. In proving Theorem 4.2, we saw that each set Q_k is a compact subset of $\mathbb{R}^n$ and that the functions $G_k(\rho)$ and $A_{kj}(\rho)$ are continuous on this set. Then, since $v(\mathrm{A})$ is finite and A is superconsistent, Theorem 3.4 implies that program B is feasible and has a solution and that, moreover, $v(\mathrm{A}) = v(\mathrm{B})$.

In order to prove Clause (ii), we need the following lemma.

Lemma 4.1. Given $\pi \in \mathbb{R}^{m+1}$, the vector ρ^* that maximizes the functions

$$\bar{C}_k(\rho) = G_k(\rho) - \sum_{j=1}^{m} A_{kj}(\rho)\pi_j, \qquad k \geq 1, \rho \in Q_k,$$

$$\bar{C}_0(\rho) = G_0(\rho) - \sum_{j=1}^{m} A_{0j}(\rho)\pi_j - \pi_0, \qquad \rho \in Q_0,$$

is given by

$$\rho_i^* = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \Bigg/ \left[\sum_{w \in [k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right], \qquad i \in [k]. \tag{35}$$

Proof of Lemma 4.1. Consider the function $C_k(\rho)$, $k \geq 1$, $\rho \in Q_k$,

$$\bar{C}_k(\rho) = G_k(\rho) - \sum_{j=1}^{m} A_{kj}(\rho)\pi_j$$

$$= \sum_{i \in [k]} \rho_i \log(c_i/\rho_i) - \sum_{j=1}^{m} \pi_j \left(\sum_{i \in [k]} a_{ij}\rho_i\right).$$

We thus solve the problem

$$\begin{aligned} &\text{maximize} \quad \bar{C}_k(\rho), \\ &\text{s.t.} \qquad \sum_{i \in [k]} \rho_i = 1, \qquad \rho_i \geq 0, \qquad i \in [k]. \end{aligned}$$

The Lagrangian for this problem is

$$L(\rho, v) = \bar{C}_k(\rho) + v\left[\left(\sum_{i \in [k]} \rho_i\right) - 1\right].$$

Differentiating this with respect to ρ_i and v, and setting the resulting expression equal to zero, we have

$$\log c_i - 1 - \log \rho_i^* - \left(\sum_{j=1}^{m} \pi_j a_{ij}\right) + v = 0,$$

$$\sum_{i \in [k]} \rho_i^* - 1 = 0,$$

which imply that

$$\rho_i^* = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \exp(v-1), \qquad \sum_{i \in [k]} \rho_i^* = 1.$$

Therefore,

$$\exp(v-1)=1\Big/\left[\sum_{i\in[k]} c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right)\right],$$

$$\rho_i^* = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right)\Big/\left[\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right].$$

For $k=0$, $C_0(\rho)$ includes the additional constant $-\pi_0$, but the derivation of the optimizing ρ is identical to that for $k>0$. This concludes the proof of Lemma 4.1. □

Now, we return to the proof of Theorem 4.4 and recall that π is optimal and feasible for program A. We therefore have

$$\sum_{j=1}^{m} \pi_j\left(\sum_{i\in[k]} \rho_i a_{ij}\right) \geq \sum_{i\in[k]} \rho_i \log(c_i/\rho_i), \qquad \rho\in Q_k,\ k=1,2,\ldots,p,$$

$$\sum_{j=1}^{m} \pi_j\left(\sum_{i\in[0]} \rho_i a_{ij}\right) + \pi_0 \geq \sum_{i\in[0]} \rho_i \log(c_i/\rho_i), \qquad \rho\in Q_0.$$

Using $\lambda_k \geq 0$ and $\rho^k \in Q_k$ that were obtained in proving Clause (i), we have

$$\lambda_0\pi_0 + \sum_{k=0}^{p} \lambda_k\left\{\sum_{i\in[k]} \rho_i^k\left(\sum_{j=1}^{m} \pi_j a_{ij}\right)\right\} \geq \sum_{k=0}^{p} \lambda_k\left\{\sum_{i\in[k]} \rho_i^k \log(c_i/\rho_i^k)\right\}. \tag{36}$$

Furthermore, since the vector $(\rho^0, \rho^1, \ldots, \rho^p, \lambda_0, \lambda_1, \ldots, \lambda_p)$ is feasible in program B, $\lambda_0 = 1$ and

$$\sum_{k=0}^{p} \lambda_k A_{kj}(\rho^k) = 0, \qquad j=1,2,\ldots,m,$$

we have, for $j=1,2,\ldots,m$,

$$\sum_{k=0}^{p} \lambda_k \sum_{i\in[k]} a_{ij}\rho_i^k = 0, \tag{37a}$$

so that

$$\sum_{j=1}^{m} \pi_j \sum_{k=0}^{p} \lambda_k \sum_{i\in[k]} a_{ij}\rho_i^k = 0, \tag{37b}$$

$$\sum_{k=0}^{p} \lambda_k \sum_{i\in[k]} \rho_i^k \sum_{j=1}^{m} \pi_j a_{ij} = 0. \tag{37c}$$

Furthermore, $v(\mathrm{P}) = v(\mathrm{D})$ implies that

$$\pi_0 = \sum_{k=0}^{p} \lambda_k G_k(\rho^k) = \sum_{k=0}^{p} \lambda_k \sum_{i\in[k]} \rho_i^k \log(c_i/\rho_i^k),$$

and the fact that $\lambda_0 = 1$ then implies that

$$\lambda_0 \pi_0 = \sum_{k=0}^{p} \lambda_k \sum_{i \in [k]} \rho_i^k \log(c_i/\rho_i^k). \tag{38}$$

From (36), (37c), and (38), we conclude that (36) is satisfied as a strict equality. This implies that

$$\sum_{k=1}^{p} \lambda_k \left\{ \sum_{i \in [k]} \rho_i^k \left[\log(c_i/\rho_i^k) - \sum_{j=1}^{m} \pi_j a_{ij} \right] \right\}$$
$$+ \lambda_0 \left\{ \sum_{i \in [0]} \rho_i^0 \left[\log(c_i/\rho_i^0) - \sum_{j=1}^{m} \pi_j a_{ij} \right] - \pi_0 \right\} = 0,$$

that is,

$$\sum_{k=1}^{p} \lambda_k \left\{ G_k(\rho^k) - \sum_{j=1}^{m} \pi_j A_{kj}(\rho^k) \right\}$$
$$+ \lambda_0 \left\{ G_0(\rho^0) - \sum_{j=1}^{m} A_{0j}(\rho^0) - \pi_0 \right\} = 0. \tag{39}$$

Furthermore, since the vector $(\rho^0, \rho^1, \ldots, \rho^p, \lambda_0, \lambda_1, \ldots, \lambda_p)$ is feasible in program B,

$$\sum_{j=1}^{m} \pi_j A_{kj}(\rho^k) \geq \begin{cases} G_k(\rho^k), & k = 1, 2, \ldots, p, \\ G_0(\rho^0) - \pi_0, & k = 0, \end{cases}$$

implying that

$$G_k(\rho^k) - \sum_{j=1}^{m} \pi_j A_{kj}(\rho^k) \geq 0, \qquad k = 1, 2, \ldots, p, \tag{40}$$

$$G_0(\rho^0) - \sum_{j=1}^{m} \pi_j A_{0j}(\rho^0) - \pi_0 \geq 0, \qquad k = 0. \tag{41}$$

Then, from (39), (40), (41) and the fact that $\lambda_k \geq 0$ for all k,

$$\lambda_k \left\{ G_k(\rho^k) - \sum_{j=1}^{m} \pi_j A_{kj}(\rho^k) \right\} = \begin{cases} 0, & k = 1, 2, \ldots, p, \\ \pi_0, & k = 0. \end{cases} \tag{42}$$

Note that this is merely the complementary slackness theorem of linear programming, as applied to the semi-infinite program.

Consider Eq. (42) above. We examine two cases.

Case 1. Suppose that $\lambda_k > 0$ for some $k \geq 1$. Note that λ_0 $(=1)$ is always strictly positive. Then,

$$G_k(\rho^k) - \sum_{j=1}^{m} \pi_j A_{kj}(\rho^k) = 0, \tag{43a}$$

$$G_0(\rho^0) - \sum_{j=1}^{m} \pi_j A_{0j}(\rho^0) - \pi_0 = 0. \tag{43b}$$

Furthermore, since π is feasible in program A, (24b) indicates that the quantity

$$\left\{G_k(\rho) - \sum_{j=1}^{m} \pi_j A_{kj}(\rho)\right\} = \bar{C}_k(\rho)$$

is nonpositive for all $\rho \in Q_k$. Then, (43a) indicates that the vector ρ^k maximizes the function $\bar{C}_k(\rho)$. Similarly, (24c) and (43b) together indicate that the vector ρ^0 maximizes the function $\bar{C}_0(\rho)$ over all $\rho \in Q_0$. Thus, for $k=0$ and for k such that $\lambda_k > 0$, Lemma 4.1 allows us to conclude that

$$p_i^k = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \Big/ \left[\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right], \qquad i \in [k].$$

Case 2. Suppose that $\lambda_k = 0$ for some $k \geq 1$. Then, the dual objective (30d) and the constraint functions (30a) are unchanged for any choice of ρ from the set Q_k. The same is also true of Eq. (42). We may therefore actually choose the vector ρ^k so that it satisfies

$$p_i^k = c_i \exp\left(\sum_{j=1}^{m} -\pi_j a_{ij}\right) \Big/ \left[\sum_{w\in[k]} c_w \exp\left(\sum_{j=1}^{m} -\pi_j a_{wj}\right)\right], \qquad i \in [k].$$

This proves clause (ii); thereby, the proof for Theorem 4.4 (strong duality theorem) is concluded. □

5. Algorithmic Aspects

In this section, we briefly discuss two successful GP algorithms and demonstrate how they are actually based upon the SILP formulation of geometric programming (namely, programs A and B). The first (GRIDGP) is a column-generation procedure based on the dual program (Ref. 4). GRIDGP proceeds by generating an initial set of vectors ρ^k for each value of k. Specifically, there are q_k such vectors generated from each set Q_k, where q_k is the cardinality of the set $[k]$, and where each of these vectors is a column from an identity matrix of order q_k. It may be noted that this form is not a rigid requirement and that any initial set could be chosen as long as it has nonnegative elements that summed to 1; the above procedure merely obviates the need for a user-specified starting point and seems to work quite well in practice. The next step is to generate the column corresponding to these vectors, form the initial LP with these columns, solve it, and obtain the simplex multiplier vector. The reduced cost function corresponding to each set $[k]$ is then maximized (these subproblems have a closed form solution given by (35) and proved in Lemma 4.1). If the

maximum value is positive, then the corresponding vector ρ^k is generated and the column from this is added to the LP for the set $[k]$. The procedure is repeated until the maximum reduced cost function for each k is nonpositive at some iteration. This corresponds to the optimal iteration, and the simplex multipliers are exponentiated to obtain the optimal values of the primal variables in the original GP problem.

The other algorithm (GGP, Ref. 3) is a general method which is also applicable to signomial GP problems. It proceeds by solving a sequence of posynomial GP approximations, and the procedure used for this is a cutting-plane algorithm based upon the SILP primal problem. Essentially, an initial set of constraints is generated from a user-specified starting point and the corresponding LP approximation is solved. A test is performed to check whether the current solution would be feasible in the original semi-infinite problem (with its infinitely many constraints). If this test is passed, then the current solution is optimal and the algorithm stops. Otherwise, a condensation is formed around the current solution via (8), (9), and (10), and this generates a new constraint via (14). The extra constraint is added to the current LP, thus cutting off a portion of the current feasible region. The procedure is repeated until the feasible region at some iteration is sufficiently small to determine the feasible region of the original problem, at which point the current solution also becomes the solution to the original GP primal problem.

As the structure of these two algorithms indicates, they are extensions of the primal-simplex and dual-simplex methods to the case of semi-infinite linear programs. One works toward optimality while maintaining feasibility, while the other works toward feasibility while maintaining optimality. In fact, GRIDGP has been coded so as to keep track simultaneously of both the primal and the dual, and at successive iterations it is possible to view the progress being made by each in the direction of the optimum. The obvious advantage here is the absence of any explicit nonlinearity, and the results of computational experiments seem to indicate that these LP-based methods outperform consistently other GP procedures that operate directly on the original nonlinear versions.

6. Summary

The geometric progamming primal problem has been restated as a semi-infinite linear program; based on the well-documented results for semi-infinite linear programming, duality results similar to those developed by Duffin, Peterson, and Zener (Ref. 5) have been presented. Furthermore, an explicit relationship between the optimal primal and dual variables

(including those that correspond to terms in the primal constraints that are slack at the optimum) has been presented. Finally, with the interior point methods that have been developed over the recent past, there should be further opportunities for exploiting this structure of the GP problem.

References

1. KELLEY, J. E., *The Cutting Plane Method for Solving Convex Programs*, SIAM Review, Vol. 8, pp. 703–712, 1960.
2. GUSTAFSON, S.-Å., and KORTANEK, K. O., *Semi-Infinite Programming and Applications*, Mathematical Programming: The State of the Art, Edited by A. Bachem, M. Grotschel, and B. Korte, Springer-Verlag, New York, New York, 1983.
3. AVRIEL, M., DEMBO, R. S., and PASSY, U., *Solution of Generalized Geometric Programs*, International Journal of Numerical Methods in Engineering, Vol. 9, pp. 149–168, 1975.
4. BRICKER, D. L., and RAJGOPAL, J., *Yet Another Geometric Programming Dual Algorithm*, Operations Research Letters, Vol. 2, pp. 177–180, 1982.
5. DUFFIN, R. J., PETERSON, E. L., and ZENER, C., *Geometric Programming: Theory and Applications*, John Wiley and Sons, New York, New York, 1967.
6. DUFFIN, R. J., *Linearizing Geometric Programs*, SIAM Review, Vol. 12, pp. 211–227, 1970.
7. DINKEL, J. J., ELLIOTT, W. H., and KOCHENBERGER, G. A., *A Linear Programming Approach to Geometric Programs*, Naval Research Logistics Quarterly, Vol. 25, pp. 39–53, 1978.
8. GLASHOFF, K., and GUSTAFSON, S.-Å., *Linear Optimization and Approximation*, Springer-Verlag, New York, New York, 1983.
9. GLASHOFF, K., *Duality Theory of Semi-Infinite Programming*, Semi-Infinite Programming: Proceedings of a Workshop, Bad Honnef, Germany, 1978; Edited by R. Heffich, Springer-Verlag, New York, New York, 1979.
10. HAAR, A., *Über Lineare Ungleichungen*, Acta Mathematica, Vol. 2, pp. 1–14, 1924.
11. DUFFIN, R. J., and KARLOVITZ, L. A., *An Infinite Linear Program with Duality Gap*, Management Science, Vol. 12, pp. 122–134, 1965.
12. GOCHET, W., SMEERS, Y., and KORTANEK, K. O., *Using Semi-Infinite Programming in Geometric Programming*, Proceedings of the 20th International Meeting of TIMS, Tel Aviv, Israel, 1973; Edited by E. Skifler, Academic Press, New York, New York, Vol. 2, pp. 430–438, 1973.
13. DANTZIG, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.