Rizk-Aziz, Kemil A.; Hurter, Arthur P.

Working Paper

A Relaxation Algorithm for a Class of Decomposable Polynomial Programming Problems

Discussion Paper, No. 56

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Rizk-Aziz, Kemil A.; Hurter, Arthur P. (1973) : A Relaxation Algorithm for a Class of Decomposable Polynomial Programming Problems, Discussion Paper, No. 56, Northwestern University, Kellogg School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston, IL

This Version is available at:
https://hdl.handle.net/10419/220416

Discussion Paper No. 56

A RELAXATION ALGORITHM FOR A CLASS OF DECOMPOSABLE POSYNOMIAL PROGRAMMING PROBLEMS*

PART I: THE GENERAL CASE

PART II: A SPECIAL CASE

by

Kemil A. Rizk-Aziz, and Arthur P. Hurter, Jr.

October 8, 1973

*Work partially supported by N.S.F. through the Urban Systems Engineering Center, The Technological Institute, Northwestern University, and by the National Science Foundation under contract number GK38336. The support is tratefully acknowledged.

A RELAXATION ALGORITHM FOR A CLASS OF DECOMPOSABLE POSYNOMIAL PROGRAMMING PROBLEMS

Part I

by

Kemil A. Rizk-Aziz*

Arthur P. Hurter, Jr.

ABSTRACT

A large scale posynomial programming problem amenable to decomposition is considered. It exhibits a vector of coupling variables z which tie together N disjoint vectors of subsystem variables (t^i, i = 1, 2,...,N). The problem is similar to that considered by Heymann and Avriel [6] but subsystem posynomial problems of degrees of difficulty greater than zero are permitted here. The relaxation algorithm, originally developed by Benders [1] for mixed integer linear programs, is extended to the class of posynomial programs described above. A result similar to the Farkas lemma for linear equalities and inequalities is derived for posynomial inequalities.

* The authors gratefully acknowledge support of the National Science Foundation and the Urban Systems Engineering Center, Northwestern University.

1. INTRODUCTION:

The literature on optimization problems includes a substantial class of problems referred to as "large-scale programming problems". These are characterized by a large number of constraints and variables and often exhibit a special constraint set structure.

The strategy for the solution of these problems involves the advantageous usage of the structure of the constraint set. It is common to find a structure that combines variables that belong uniquely to subsystems with variables which tie together the different subsystems. The former are referred to as "subsystem variables" and the latter as "coupling variables". A decomposition procedure may be applied to these problems, reducing the solution of the problem to a solution of a series of smaller problems combined together in different ways.

Analogous problems also appear in posynomial programming (i.e., geometric programming [2,9]). Zener [12] decomposed certain cost minimization problems with a special structure by minimizing the cost of specific overlapping systems. Heymann and Avriel [6] extended Zener's results to a large class of posynomial programming problems. However, their results are limited to problems where the structure permits only subsystem problems which are themselves posynomial programming problems with zero degrees of difficulty. This paper develops a relaxation algorithm to solve problems of the same general structure. However, there is no restriction on the generality of the subsystem, i.e., subproblems which are posynomial programming problems with degrees of difficulty different from zero are allowed.

Decomposition procedures for more general geometric programming problems (i.e., not necessarily posynomial programming problems) have

recently been developed by Peterson [10]. The procedures in this paper exploit the special features of posynomial programming problems.

2. The Problem:

A general posynomial programming problem consisting of N+1 interrelated subsystems is considered. Two kinds of variables are present; "coupling" variables $\underline{z}$ and "subsystems" variables $\underline{t}$. The vector of coupling variables z is common to all subsystem $\underline{\lambda}$ (λ=1,2,...,N).

Associated with each subsystem $\underline{\lambda}$ ($\lambda = 1, 2,..., N$) there is a vector of variables $\underline{t}^{\lambda}$ which appears only in subsystem λ. The vector of subsystems variables t is defined by the vectors t^{ℓ}, $\ell = 1, 2,..., N$, i.e., $t \triangleq (t^1, t^2,..., t^N)$. The vector of variables (z,t) will be indexed as follows: m(0) and n(0) are the indices given to the first and last elements in the vector of coupling variables $\underline{z}$ respectively. For each t^{λ} $\lambda = 1, 2,..., N$, m(λ) and n(λ) are the indices given to the first and last elements in t^{λ}, respectively. The following vectors of variables will be indexed sequentially in the order given: $z, t^1, t^2,..., t^N$. $m(0) \triangleq 1$, and n(0) is the number of z variables involved. $m(1) \triangleq n(0)+1$ Thus and n(1) - m(1) is the number of variables in t^1.

In general, $t^{\lambda} \triangleq (t_{m(\lambda)},..., t_{n(\lambda)})$ and $m(\lambda) = n(\lambda-1) + 1$ for $\lambda = 1,2,...,N$. The total number of variables in (z,t) is the index of the last variable in t^N, i.e., $t_{n(N)}$, or $n(N) \triangleq m$.

DEFINITION 2.1: A posynomial term, $u_i(x)$ is an expresion of the form

$$u_i(x) \triangleq c_i \prod_j x_j^{a_{ij}}$$

where c_i is a positive scalar

a_{ij}, for all j are arbitrary real numbers

and the x_j, for all j, are positive variables.

Here, Π indicates product over the j's.

DEFINITION 2.2: A posynomial function g(x) is the sum of posynomial terms, i.e.

$$g(x) \triangleq \sum_i u_i(x) \triangleq \sum_i c_i \prod_j x_j^{a_{ij}}$$

Each subsystem has a posynomial objective function and "prototype" posynomial constraints i.e. (posynomial functions kept less than or equal to 1). The subsystem whose objective function and constraints depend only on the z variables, is termed subsystem 0, and those subsystems having objective function and constraints depending on the $\underline{z}$ and the $\underline{t}^{\ell}$ variables are termed subsystem ℓ, for $\ell = 1, 2, \ldots, N$.

For the purpose of exposing the structure of the problem, it is convenient to identify those functions belonging to different subsystems. Thus, let $g_0(z)$ and $g_\ell(z,t^\ell)$, $\ell = 1, 2, \ldots, N$ be the posynomial objective functions corresponding to subsystems $\ell = 0, 1, 2, \ldots, N$ respectively. Let $g_k(z)$, for $k \in P_0$ and $g_k(z,t^\ell)$, for $k \in P_\ell$, $\ell = 1, 2, \ldots, N$ be the posynomial constraint functions corresponding to subsystems $\ell = 0, 1, 2, \ldots, N$. P_ℓ, $(\ell = 0,1,2,\ldots,N)$ are the index sets of the constraints functions belonging to subsystem ℓ, $(\ell = 0,1,2,\ldots,N)$ and will be defined below.

For notational simplification and because of the structure of dual posynomial programs, it is also convenient to index sequentially (1) the posynomial functions and (2) the posynomial terms of the objective functions and constraints belonging to different subsystems.

(1) Let $\mathcal{O} \triangleq \{0,1,2,\ldots,N\}$ be the index set identifying the N+1 posynomial objective functions corresponding to subsystems $\ell = 0,1,2,\ldots,N$.

Let $P_0 \triangleq \{m_0', m_0'+1,\ldots,n_0'\}$ be the index set of the $n_0' - m_0'$ constraint functions of subsystem 0, while $P_\ell \triangleq \{m_\ell', m_\ell'+1,\ldots,n_\ell'\}$ is the corresponding index set of the $n_\ell' - m_\ell'$ constraint functions of subsystem ℓ, $\ell = 1,2,\ldots,N$.

The posynomial functions will be indexed sequentially by ordering the index sets defined above, as follows:

$$\overbrace{\{0,1,2,\ldots,N}^{\mathcal{O}}, \overbrace{m_0',m_0'+1,\ldots,n_0'}^{P_0}, \overbrace{m_1',\ldots,n_1'}^{P_1}, \ldots\ldots, \overbrace{m_N',m_N'+1,\ldots,n_N'}^{P_N}\}$$

Thus, $m_0' \triangleq N+1$, $m_1' \triangleq n_0'+1$ and in general, $m_\ell' \triangleq n_{\ell-1}'+1$ $\ell=1,2,\ldots,N$.

The total number of posynomial functions p + 1 is given by the index assigned to the last element in P_N plus one, i.e., $n_N' \triangleq p$. The index $\underline{k}$ is used to identify a particular posynomial function. Thus, if $k \in \mathcal{O}$, k identifies the posynomial objective function of subsystem k. If $k \in P_\ell$ for some $\ell \in \mathcal{O}$, k identifies a particular posynomial constraint function which belongs to the constraint set P_ℓ of subsystem ℓ. Let $P \triangleq P_1 \cup P_2 \cup \ldots \cup P_N$, then $\mathcal{O} \cup P \triangleq \mathcal{O} \cup P_1 \cup P_2 \cup \ldots \cup P_N = \{0,1,2,\ldots,p\}$.

Each $k \in \mathcal{O}$ identifies a posynomial function of the objective function and each $k \in P$ identifies a particular constraint function.

(2) Recall that each posynomial function g is defined as the sum of a number of "posynomial terms" (see definition 2.2). Consider the function g_k, for any $k \in \mathcal{O} \cup P$, and define m_k and n_k to be the indices of the first and last posynomial terms of g_k respectively. The index set of the posynomial terms of g_k is defined by

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

$[k]$ is called "block k" [2,9]. Each posynomial function g_k, $k \in G \cup P$ has an associated block of indices $[k]$ which index the posynomial terms of which it is formed.

The indices representing posynomial terms are ordered in much the same way posynomial functions were ordered. The ordering of blocks $[k]$ will follow the ordering of the functions k, i.e.

$$\{\underbrace{[0],[1],[2],\ldots,[N]}_{[G]},\ \underbrace{[m_0'],[m_0'+1],\ldots,[n_0']}_{[P_0]},\ \ldots\ldots,\ \underbrace{[m_N'],[m_N'+1],\ldots,[n_N']=[p]}_{[P_N]}\}$$

here, $m_0' \triangleq N+1$ and $m_\ell' \triangleq n_{\ell-1}'+1 \quad \ell = 0,1,2,\ldots,N$

and $[G] \triangleq [0] \cup [1] \cup [2] \cup \ldots \cup [N] \qquad [P_\ell] \triangleq [m_\ell'] \cup [m_\ell'+1] \cup \ldots \cup [n_\ell']$

The subsystems objective functions were ordered before the constraints. Therefore, blocks of terms corresponding to each objective function $g_0, g_1, \ldots, g_N$ are ordered sequentially as:

$$[G] \triangleq \{\underbrace{m_0,m_0+1,\ldots,n_0}_{[0]},\ \underbrace{m_1,m_1+1,\ldots,n_1}_{[1]},\ m_2\ldots,\ \underbrace{m_N,m_N+1,\ldots,n_N}_{[N]}\}$$

here, $m_0 \triangleq 1$ and $m_k \triangleq n_k + 1 \qquad k \in G = \{0,1,2,\ldots,N\}$

$[G]$ is the set of indices identifying all the "posynomial terms" in the objective function. The index sets P_ℓ for the constraints of subsystem ℓ ($\ell = 0,1,2,\ldots,N$) were defined as

$$P_\ell = \{m_\ell', m_\ell'+1,\ldots,n_\ell'\} \qquad \ell = 0,1,2,\ldots,N$$

Then $[n_\ell' + j] \triangleq \{m_{m_\ell'+j}, m_{m_\ell'+j}+1,\ldots,n_{m_\ell'+j}\}$ identify all the terms in the constraint $k = m_\ell' + j$ of subsystem ℓ, where $m_\ell' \le m_\ell' + j \le n_\ell'$.

The vector of indices for all the "terms" in all the functions g_k, $k \in G \cup P$ is:

$$[G \cup P] \triangleq \{\overbrace{m_0, m_0+1, \ldots, n_0}^{[0]}, \overbrace{m_1, m_1+1, \ldots, n_1}^{[1]}, \ldots, \overbrace{m_N, \ldots, n_N}^{[N]},$$

$$\overbrace{\overbrace{m_{m_0'}, \ldots, n_{m_0'}}^{[m_0']}, \overbrace{m_{m_0'+1}, \ldots, n_{m_0'+1}}^{[m_0'+1]}, \ldots \overbrace{m_{n_0'}, \ldots, n_{n_0'}}^{[n_0']}}^{[P_0]}, \ldots \overbrace{m_{m_1'}, \ldots, n_{n_1'}}^{[P_1]},$$

$$\overbrace{m_{m_2'}, \ldots, n_{n_2'}}^{[P_2]}, \ldots, \overbrace{m_{m_N'}, \ldots, n_{n_N'} \triangleq n_p \triangleq n}^{[P_n]}\}$$

where, $m_0' \triangleq N+1$, $m_\ell' \triangleq n_{\ell-1}' + 1 \quad \ell = 1,2,\ldots,N$

$m_0 \triangleq 1$, $m_{m_\ell'} \triangleq n_{n_{\ell-1}'} + 1 \quad \ell = 1,2,\ldots,N \qquad m_{m_{\ell+1}'} \triangleq n_{m_\ell'} + 1 \quad \ell = 0,1,2,\ldots,N$

and $[P_\ell] \triangleq \bigcup_{k \in P_\ell} [k] \qquad \ell = 0,1,2,\ldots,N$

$$[G \cup P] \triangleq [G] \cup [P_1] \cup [P_2] \cup \ldots \cup [P_N].$$

The index i is used to refer to "term i" in some "block k" i.e. $i \in [k]$, so i identifies a particular posynomial term of a posynomial function g_k. If $i \in [G]$, i identifies a particular term in the objective function. If $i \in [P_\ell]$, i identifies a particular term in the constraint functions of subsystem ℓ. Suppose index i refers to the p-th term of the j-th constraint of subsystem ℓ. Then it would be denoted $(m_{m_\ell'} + j + p)$. Thus, index i runs over $[m_\ell' + j]$, that is, over the terms of constraint $k = m_\ell' + j$ where $k \in P_\ell$. Finally, for subsystems ℓ, $(\ell = 0,1,2,\ldots,N)$, $k = \ell$ identifies the objective function of subsystem ℓ and $k \in P_\ell$ identifies the constraint functions of subsystem ℓ.

It is useful to define sets $P_\ell' \triangleq P_\ell \cup \{\ell\}$ which identify all the

posynomial functions corresponding to subsystem ℓ (objective function and constraints). Additionally $[P'_\ell] \triangleq [P_\ell] \cup [\ell]$ identifies all the posynomial terms of all functions belonging to subsystem ℓ ($\ell = 0,1,2,\ldots,N$).

The posynomial terms corresponding to subsystem 0 are a function of the z variables only, hence according to definition 2.1,

$$(2.1) \qquad u_i(z) \triangleq c_i \prod_{j=1}^{n(0)} z_j^{a_{ij}} \qquad i \in [P'_0]$$

Each posynomial function in subsystem 0 is defined in definition 2.1 as a sum of posynomial terms involving the z variables (i.e. 2.1)

$$(2.2) \qquad g_k(z) \triangleq \sum_{i \in [k]} u_i(z) \qquad k \in P'_0$$

For any subsystem ℓ ($\ell = 1,2,\ldots,N$) the posynomial terms are a function of the z variables as well as the t^ℓ variables, i.e.

$$(2.3) \qquad w_i(z,t^\ell) \triangleq c_i \prod_{j=1}^{n(0)} z_j^{a_{ij}} \prod_{j=m(\ell)}^{n(\ell)} t_j^{a_{ij}} \qquad i \in [P'_\ell]$$

Note that, (2.3) can be written as

$$(2.4) \qquad w_i(z,t^\ell) \triangleq u_i(z) \prod_{j=m(\ell)}^{n(\ell)} t_j^{a_{ij}} \qquad i \in [P'_\ell]$$

If one defines

$$(2.5) \qquad u_i(z) \triangleq c_i \prod_{j=1}^{n(0)} z_j^{a_{ij}} \qquad i \in [P'_\ell]$$

Moreover, the posynomial functions of subsystem ℓ ($\ell=1,2,\ldots,N$) are

$$(2.6) \qquad g_k(z,t^\ell) \triangleq \sum_{i \in [k]} w_i(z,t^\ell) \qquad k \in P'_\ell$$

Because of (2.1) and (2.5), there exists a posynomial term $u_i(z)$ for $i \in [P'_0] \cup [P'_1] \cup \ldots \cup [P'_N] = [P]$, i.e., for every term in every posynomial function defined. Indexing the posynomial terms $u_i(z)$ indexes the compound

posynomial terms $w_i(z,t^\ell)$.

Using (2.1) - (2.6) the problem of interest can now be written:

Program A:

$$(2.7) \qquad \min\ g_0(z) + \sum_{\ell=1}^{N} g_\ell(z,t^\ell)$$

Subject to

$$(2.8) \qquad g_k(z) \le 1 \qquad\qquad k \in P_0$$

$$(2.9) \qquad g_k(z,t^\ell) \le 1 \qquad\qquad k \in P_\ell \qquad\qquad \ell = 1,2,\ldots,N$$

with $\qquad z > 0,\ t^1 > 0,\ t^2 > 0,\ldots,t^N > 0$

Each posynomial term $u_i(z)$, $i \in [G \cup P]$ i.e. in the objective function and in each constraint, gives rise to an independent dual variable labeled δ_i, $i \in [G \cup P]$. The sum of the posynomial terms of the objective function and each posynomial g_k, $k \in P$, gives rise to dependent dual variables $\lambda_G(\delta) \triangleq \sum_{i \in [G]} \delta_i$ and $\lambda_k(\delta) \triangleq \sum_{i \in [k]} \delta_i$, $k \in P$ respectively. The geometric programming dual associated with program A is program B [2,9]:

Program B:

$$(2.10) \qquad \max\ v(\delta) \triangleq \left\{ \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \prod_{k \in P} \lambda_k(\delta)^{\lambda_k(\delta)} \right\}$$

subject to:

$$(2.11) \qquad \delta_i \ge 0 \qquad i \in [G \cup P] = \{1,2,\ldots,n\}$$

$$(2.12) \qquad \lambda_G(\delta) \triangleq \sum_{i \in [G]} \delta_i = 1$$

$$(2.13) \qquad \sum_{i=1}^{n} a_{ij} \delta_i = 0 \qquad\qquad j = 1,2,\ldots,n(0)$$

$$(2.14) \qquad \sum_{i \in [P_\ell]} a_{ij} \delta_i = 0 \qquad\qquad j = m(\ell),\ldots,n(\ell)$$

$$\ell = 1,2,\ldots,N$$

where $\lambda_k(\delta) = \sum_{i \in [k]} \delta_i \qquad k \in P$

The following assumptions are made:

(i) Program A and its dual B are canonical programs, ie., there exists a vector $\hat{\delta} > 0$ satisfying the constraints of program B [2,9].

(ii) Program A is superconsistent [2,9], i.e., there is at least one vector $(\hat{z}, \hat{t}^1, \hat{t}^2, \ldots, \hat{t}^N)$ with positive components satisfying the constraints of program A, with strict inequality.

The strong duality theorem for geometric programming [2,9] guarantees a finite optimal solution for program A and its dual B under assumptions (i) and (ii).

Program A can be rewritten using a projection procedure described by Geoffrion [3]. Projection into the z variables yields:

Program MA:

$$(2.15) \qquad \text{Min } g_0(z) + \sum_{\ell=1}^{N} \phi_\ell(z)$$

subject to:

$$(2.16) \qquad z \in Z'$$

$$(2.17) \qquad g_k(z) \leq 1 \qquad k \in P_0$$

where

$$Z' \triangleq \bigcap_{\ell=1}^{N} Z'_\ell$$

and

$$Z'_\ell \triangleq \{z > 0 \mid \text{there exists } t^\ell > 0 \ni g_k(z, t^\ell) \leq 1 \quad k \in P_\ell\}$$

Each of the subsystem objectives after projection, the $\phi_\ell(z)$ in (2.15) is defined by program $SA_\ell(z)$ for each $\ell = 1, 2, \ldots, N$.

Program $SA_\ell(z)$: Given $z > 0$, find

$$\sigma_\ell(z) \triangleq \underset{t^\ell > 0}{\text{INF}} \; g_\ell(z,t^\ell) \text{ subject to } g_k(z,t^\ell) \leq 1, \quad k \in P_\ell$$

The potential advantage of this alternative formulation of program A becomes apparent when it is observed that, for a given $z > 0$, satisfying (2.16) and (2.17), (2.15) decomposes into the solution of N independent subproblems $SA_\ell(z)$, $\ell = 1,2,\ldots,N$. The solutions to program MA and to program A are related through the use of the projection theorem [4].

THEOREM 2.1: If (z^*, t^*) is optimal in program A then z^* solves program MA. If z^* is optimal in program MA and $(t^\ell)^*$ achieves the minimum in $SA_\ell(z^*)$ for $\ell = 1,2,\ldots,N$, then (z^*, t^*) is optimal in program A.

An algorithm for the solution to program A is developed using program MA and subproblems $SA_\ell(z)$, $\ell = 1,2,\ldots,N$. Theorem 2.1 specifies the conditions which ensure a solution to program A as well.

In program MA, dual representations of $\sigma_\ell(z)$ and z'_ℓ, for each $\ell = 1,2,\ldots,N$ are necessary. Sections 3 and 4 are addressed to these dual representations respectively. In what follows, a program is consistent iff there exists a point satisfying its constraints.

Since z is a parameter in $\sigma_\ell(z)$ and Z'_ℓ, the following simplifying notation is helpful.

DEFINITION 2.3: Let

$$u_i \triangleq u_i(z) \quad \text{for} \quad i = 1,2,\ldots,N$$

and $g_k(t^\ell) \triangleq g_k(z,t^\ell) \qquad k \in P_\ell \cup \{\ell\} \triangleq P_\ell'$, where

$$(2.18) \qquad g_k(t^\ell) \triangleq \sum_{i\in[k]} u_i \prod_{j=m(\ell)}^{j=n(\ell)} t_j^{a_{ij}} \qquad k \in P'_\ell$$

Using (2.18), the program $SA_\ell(z)$ and the set Z'_ℓ are rewritten below.

Program $SA_\ell(z)$: Given $z > 0$, find

$$\varphi_\ell(z) \triangleq \inf_{t^\ell > 0} g_\ell(t^\ell) \quad \text{subject to} \quad g_k(t^\ell) \le 1 \quad k \in P_\ell$$

and

$$Z'_\ell = \{z > 0 \mid \text{there exists a } t^\ell > 0 \text{ s.t. } g_k(t^\ell) \le 1,\ k \in P'_\ell\}$$

3. REPRESENTATION OF $\varphi_\ell(z)$:

The dual representation of $\varphi_\ell(z)$ can be obtained by the geometric dual program of $SA_\ell(z)$, denoted by $SB_\ell(z)$ where

Program $SB_\ell(z)$: Given $z > 0$, find

$$\psi_\ell(z) \triangleq \sup \left[\prod_{i\in[P'_\ell]} \left(\frac{u_i}{\delta_i} \right)^{\delta_i} \right] \left[\prod_{k\in P_\ell} \lambda_k(\delta^\ell)^{\lambda_k(\delta^\ell)} \right]$$

subject to

$$\delta^\ell \ge 0$$

$$\lambda_\ell(\delta^\ell) \triangleq \sum_{i\in[\ell]} \delta_i = 1$$

$$\sum_{i\in[P'_\ell]} a_{ij}\delta_i = 0 \qquad j = m(\ell),\ldots,n(\ell)$$

where

$$\lambda_k(\delta) \triangleq \sum_{i\in[k]} \delta_i, \qquad \delta^\ell \triangleq (\delta_i,\ i\in[P'_\ell])$$

and $$P'_\ell \triangleq P_\ell \cup \{\ell\}$$

In comparing $SB_\ell(z)$ and B note the use of u_i in a role similar to that of c_i in B. This is explained below. Lemma 3.1 makes use of assumptions (i) and (ii) concerning programs A and B to show that for each $\ell = 1,2,\ldots,N$, $SA_\ell(z)$ and $SB_\ell(z)$ are canonical programs.

Lemma 3.1: For each $\ell = 1,2,\ldots,N$;

Programs $SA_\ell(z)$ and $SB_\ell(z)$ are canonical Programs.

Proof: Programs A and B are canonical programs by assumption (i). Hence there exists a $\bar{\delta} > 0$ satisfying the constraints of Dual program B. For each $\ell = 1,2,\ldots,N$, $\hat{\delta}^\ell > 0$ satisfies the constraints of Dual programs $SB_\ell(z)$ where

$$(1) \qquad \hat{\delta}_i \triangleq \frac{\bar{\delta}_i}{\lambda_\ell(\bar{\delta})} \qquad \text{for} \quad i \in [P_\ell] \cup [\ell] \triangleq [P'_\ell]$$

and

$$\lambda_\ell(\bar{\delta}) \triangleq \sum_{i\in[\ell]} \bar{\delta}_i$$

As shown below: For each $\ell = 1,2,\ldots,N$

(a) $\hat{\delta}_i > 0 \quad i \in [P'_\ell]$ since $\bar{\delta}_i > 0 \quad i\in[P'_\ell]$ and $\lambda_\ell(\bar{\delta}) > 0$

(b) $$\sum_{i\in[\ell]} \hat{\delta}_i = \sum_{i\in[\ell]} \frac{\bar{\delta}_i}{\lambda_\ell(\bar{\delta})} = 1 \quad \text{by definition}$$

(c) $$\sum_{i\in[P'_\ell]} a_{ij}\hat{\delta}_i = \sum_{i\in[P'_\ell]} a_{ij} \frac{\bar{\delta}_i}{\lambda_\ell(\bar{\delta})} = 0 \quad j = m(\ell),\ldots,n(\ell)$$

Since

$$\sum_{i\in[P'_\ell]} a_{ij}\bar{\delta}_i = 0 \quad \text{for } j = m(\ell),\ldots,n(\ell) \quad \text{and} \quad \lambda_\ell(\bar{\delta}) > 0 .$$

Hence, since $\hat{\delta}^{\ell} > 0$ satisfies the constraints of $SB_{\ell}(z)$ for each $\ell = 1,2,\ldots,N$, then the pairs of programs $SA_{\ell}(z)$, $SB_{\ell}(z)$, for each $\ell = 1,2,\ldots,N$ are canonical programs. The lemma is proved.

Given $z > 0$, program $SB_{\ell}(z)$ is always consistent. However, for some values of z, $SA_{\ell}(z)$ can be inconsistent, i.e., no t^{ℓ} can be found with $t^{\ell} > 0$ and the constraints satisfied. The strong duality theorem of Geometric Programming for canonical programs [2,9], applied to $SA_{\ell}(z)$ and $SB_{\ell}(z)$ asserts that program $SB_{\ell}(z)$ can be either unbounded or finite and positive. If it is unbounded $SA_{\ell}(z)$ is inconsistent. If it is finite and positive, $SA_{\ell}(z)$ is consistent and the optimal values of $SA_{\ell}(z)$ and $SB_{\ell}(z)$ are equal.

In program MA, the choice of $z \in Z'$ insures the consistency of every subproblem $SA_{\ell}(z)$. Hence, if $z \in Z'$ the optimal values of $SA_{\ell}(z)$ and its dual $SB_{\ell}(z)$; $\ell = 1,2,\ldots,N$ are finite positive and equal.

Define Δ_{ℓ} to be the set of δ^{ℓ} satisfying the dual constraints of program $SB_{\ell}(z)$:

$$\Delta_{\ell} \triangleq \left\{ \delta^{\ell} \geq 0 \;\middle|\; \sum_{i \in [\ell]} \delta_i = 1, \qquad \sum_{i \in [P'_{\ell}]} a_{ij} \delta_i = 0 \quad j = m(\ell),\ldots,n(\ell) \right\}.$$

If z is chosen to be in Z'_{ℓ}, then the optimal values of $SA_{\ell}(z)$ and $SB_{\ell}(z)$ are finite and positive and $v_{\ell}(z) = v_{\ell}(z)$.

Hence

$$(3.1) \qquad v_{\ell}(z) = \max_{\delta^{\ell} \in \Delta_{\ell}} \left[\prod_{i \in [P'_{\ell}]} \left(\frac{u_i}{\delta_i} \right)^{\delta_i} \prod_{k \in P_{\ell}} \lambda_k(\delta^{\ell})^{\lambda_k(\delta^{\ell})} \right]$$

In (3.1) replace u_i by its value given in definition 2.1 and let

$$K(\delta^{\ell}) \triangleq \left[\prod_{i \in [P'_{\ell}]} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{k \in P_{\ell}} \lambda_k(\delta^{\ell})^{\lambda_k(\delta^{\ell})} \right]$$

Then,

$$(3.2) \qquad \varphi_\ell(z) = \underset{\delta^\ell \in \Delta_\ell}{\text{MAX}} \left\{ K(\delta^\ell) \prod_{i \in [P'_\ell]} \prod_{j=1}^{n(0)} z_j^{a_{ij}\delta_i} \right\} .$$

Notice that $\prod_{i \in [P'_\ell]} \prod_{j=1}^{n(0)} z_j^{a_{ij}\delta_i} = \prod_{j=1}^{n(0)} z_j^{\sum_{i \in [P'_\ell]} a_{ij}\delta_i}$.

Now, define $\xi'_j(\delta^\ell) \triangleq \sum_{i \in [P'_\ell]} a_{ij}\delta_i \qquad j = 1,2,\ldots,n(0)$ and then write (3.2) as

$$(3.3) \qquad \varphi_\ell(z) = \underset{\delta^\ell \in \Delta_\ell}{\text{MAX}} \left\{ K(\delta^\ell) \prod_{j=1}^{n(0)} z_j^{\xi'_j(\delta^\ell)} \right\}$$

which is, the desired dual representation of $\varphi_\ell(z)$ for each $\ell = 1,2,\ldots,N$.

4. REPRESENTATION OF Z'_ℓ :

Recall that $Z' \triangleq \bigcap_{\ell=1}^{N} Z'_\ell$ and

$$Z'_\ell \triangleq \{ z > 0,\ z \in R^{n(0)} \mid \exists \text{ a } t^\ell > 0 \text{ such that } g_k(t^\ell) \le 1 \quad k \in P_\ell \}$$

Treating z as a parameter is equivalent to treating u as a parameter and the set Z'_ℓ can be replaced by the set U'_ℓ where

$$U'_\ell \triangleq \{ u^\ell > 0,\ u^\ell \in R^{o([P_\ell])} \mid \exists \text{ a } t^\ell > 0 \text{ such that } g_k(t^\ell) \le 1 \quad k \in P_\ell \}$$

with $R^{o([P_\ell])} \triangleq$ space of real numbers of dimension $o([P_\ell])$ where $o([P_\ell])$ is the number of elements in $[P_\ell]$.

For the balance of section 4, consider a set U' instead of U'_ℓ obtained by replacing t^ℓ with an arbitrary vector $(t_j,\ j=1,2,\ldots,m')$ and P_i by an

arbitrary set $\{1,2,\ldots,p'\}$. This notational change is for ease of exposition only. Then

$$U' \triangleq \{u > 0,\ u \in R^{n'} \mid \exists \text{ a } t > 0 \text{ such that } g_k(t) \leq 1 \quad k = 1,2,\ldots,p'\}$$

U' is the set of all "admissible" coefficients for the set of posynomial inequalities $g_k(t)$, $k = 1,2,\ldots,p'$

where $$g_k(t) \triangleq \sum_{i \in [k]} w_i(t) \qquad k = 1,2,\ldots,p'$$

and $$w_i(t) \triangleq u_i t_1^{a_{i1}} t_2^{a_{i2}} \cdots\cdots t_m^{a_{im}} \qquad i \in \{[k];\ k=1,\ldots,p'\} \triangleq \{1,2,\ldots,n'\}.$$

Each a_{ij} is an arbitrary real number and is an element of the exponent matrix A defined by

$$A \triangleq [a_{ij}] \quad i=1,2,\ldots,n' \ , \quad j=1,2,\ldots,m'.$$

Consider the set D' defined to be the intersection of the non-negative orthant with the null space of the exponent matrix A^T, i.e.

$$D' \triangleq \{\ \gamma \geq 0,\ \gamma \in R^{n'} \mid \sum_{i=1}^{n'} a_{ij}\gamma_i = 0 \quad j = 1,\ldots,m'\}$$

Conditions for U' to be non-empty are given in theorem 4.1.

THEOREM 4.1 Suppose there exists $\hat{\gamma} \in D'$, $\ni \hat{\gamma}_i > 0$ for $i = 1,2,\ldots,n'$. Then $u \in U'$, i.e., there exists $t > 0$ such that $g_k(t) \leq 1$ for $k = 1,2,\ldots,p'$ iff

$$(4.1) \qquad \sup_{\gamma \in D'} \left\{ \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \right\} = 1$$

where $$\lambda_k(\gamma) \triangleq \sum_{i \in [k]} \gamma_i$$

in which case the sup is a max.

PROOF: The strategy of the proof involves formulation of a posynomial program which has a dual of the form given by (4.1).

Consider a posynomial program in which

$[0] = \{0\}$ $u_0 = 1$ $a_{0j} = 0$ $j = 1,2,\ldots,m'$, that is,

(1) MIN $g_0(t) \stackrel{\Delta}{=} 1$ subject to $g_k(t) \leq 1$ $k = 1,2,\ldots,p'$

The geometric dual of (1) is:

$$(2) \quad \sup \left\{ \left(\frac{u_0}{\gamma_0} \right)^{\gamma_0} \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \right\}$$

Subject to:

$$\gamma_i \geq 0 \quad i = 1,2,\ldots,n'$$

$$\gamma_0 = 1$$

$$\sum_{i=1}^{n'} a_{ij}\gamma_i = 0 \quad j = 1,2,\ldots,m'$$

where $\lambda_k(\gamma) = \sum_{i \in [k]} \gamma_i$

Program (2) can be rewritten noting that $\left(\frac{u_0}{\gamma_0}\right)^{\gamma_0} = 1$, as

$$(3) \quad \sup_{\gamma \in D'} \left\{ \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i}\right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \right\}$$

Since $\exists\, \hat{\gamma} \in D'$ $\ni \hat{\gamma}_i > 0$ for $i = 1,2,\ldots,n'$ and $\gamma_0 = 1$, programs (1) and (2) are canonical programs (by assumption), the main Duality theorem in Geometric Programming [2,9] insures that the optimal values of programs (1) and (2) are equal. Since the optimal value of (1) is by definition equal to 1, then the optimal value of (2) is also 1. Further,

since the optimal value of (2) is finite and positive, i.e., equal to 1, the duality theorem additionally states that program (1) is consistent, i.e. $u \in U'$ iff the optimal value of (2) is finite and positive, hence

$$\sup_{\gamma \in D'} \left\{ \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \right\} = 1$$

iff there exists a $t > 0$ such that

$$g_k(t) \leq 1 \qquad k = 1,2,\ldots,p'.$$

The theorem is proved.

Theorem 4.1 states that $u \in U'$ iff (4.1) holds, hence the following lemma is immediate.

Lemma 4.1: $u \in U'$ iff $u \in U$ where

$$(4.2) \qquad U = \left\{ u > 0,\ u \in R^{n'} \mid \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \leq 1 \quad \gamma \in D' \right\}$$

Lemma 4.2 also follows immediately, since if $u \notin U'$ program (1) in the proof of theorem 4.1 is inconsistent so (2) is unbounded.

Lemma 4.2: If $u \notin U'$ then $\sup_{\gamma \in D'} \left\{ \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \right\} = +\infty$

A third lemma, useful in later work, requires proof.

Lemma 4.3: A vector $u \in U$ iff $x \in X$ where

$$X \triangleq \left\{ x \in R^{n'} \mid \sum_{i=1}^{n'} \gamma_i x_i \leq \sum_{k=1}^{p'} \sum_{i \in [k]} \gamma_i \log \gamma_i - \lambda_k(\gamma) \log \lambda_k(\gamma) \text{ for all } \gamma \in D' \right\}$$

and $x_i \triangleq \log u_i \quad i = 1,2,\ldots,n'$

Proof: Using (4.2) $u \in U$ implies

$$(1) \quad \prod_{i=1}^{n'} \left(\frac{u_i}{\gamma_i} \right)^{\gamma_i} \prod_{k=1}^{p'} \lambda_k(\gamma)^{\lambda_k(\gamma)} \leq 1 \quad \text{for all } \gamma \in D'$$

Taking logarithm of both sides of (1) yields

$$(2) \quad \sum_{i=1}^{n'} \gamma_i \log u_i - \sum_{i=1}^{n'} \gamma_i \log \gamma_i + \sum_{k=1}^{p'} \lambda_k(\gamma) \log \lambda_k(\gamma) \leq 0 \quad \text{for all } \gamma \in D'$$

Since $\{[k], k=1,2,..,p'\} = \{1,2,\ldots,n'\}$, (2) can be written as

$$(3) \quad \sum_{i=1}^{n'} \gamma_i \log u_i \leq \sum_{k=1}^{p'} \left(\sum_{i \in [k]} \gamma_i \log \gamma_i - \lambda_k(\gamma) \log \lambda_k(\gamma) \right) \quad \text{for all } \gamma \in D' .$$

Define $x_i \triangleq \log u_i \quad i = 1,2,\ldots,n'$, then

$$(4) \quad \sum_{i=1}^{n'} \gamma_i x_i \leq \sum_{k=1}^{p'} \left(\sum_{i \in [k]} \gamma_i \log \gamma_i - \lambda_k(\gamma) \log \lambda_k(\gamma) \right) \quad \text{for all } \gamma \in D'$$

finally; let $X \triangleq \{ x \mid (4) \text{ is satisfied} \}$, then $u \in U$ iff $x \in X$. The lemma is proved.

For programs (1) and (2), as defined in the proof of Theorem 4.1, to be canonical, the existence of $\hat{\gamma} \in D'$ such that $\hat{\gamma}_i > 0$ is required. This is not restrictive since otherwise programs (1) and (2) are partially degenerate programs. Two canonical programs can be derived from them by deleting some of their terms. These two programs, termed, the "reduced forms" of programs (1) and (2) are defined and explained in references [2,9]. Observe that a restatement of lemma 4.3 in the following form shows its parallel with the Farkas lemma for linear equalities [2].

Lemma 4.4: There exists a vector $t > 0$ satisfying

$$g_k(t) \leq 1 \quad \text{for } k = 1,2,\ldots,p'$$

where

$$g_k(t) \triangleq \sum_{i \in [k]} u_i \prod_{j=1}^{m'} t_j^{a_{ij}} \qquad k = 1,2,\ldots,p'$$

if and only if

$$\sum_{i=1}^{n'} \gamma_i x_i - \sum_{i=1}^{n} \nu_i \log \nu_i + \sum_{k=1}^{p'} \lambda_k(\gamma) \log \lambda_k(\gamma) \leq 0$$

for every $(\gamma_i,\ i = 1,2,\ldots,n')$ satisfying

$$\nu_i \geq 0 \quad i = 1,2,\ldots,n'$$

and $$\sum_{i=1}^{n'} a_{ij}\gamma_i = 0 \qquad j = 1,2,\ldots,m'$$

where $\{[k],\ k = 1,2,\ldots,p'\} = \{1,2,\ldots,n'\}$

and $$x_i \triangleq \log u_i \qquad i = 1,2,\ldots,n' .$$

The set X as defined in lemma 4.3 can be written as

$$(4.3) \qquad X = \{\, x \mid < x,\gamma > \leq f^*(\gamma|X) \text{ for all } \gamma \in D' \,\}$$

where

$$f^*(\gamma|X) \triangleq \sum_{k=1}^{p'} \left(\sum_{i\in[k]} \nu_i \log \gamma_i - \lambda_k(\nu) \log \lambda_k(\nu) \right) .$$

Theorem A1 in appendix A (after Part II) shows that $f^*(\gamma|X)$ is a <u>non-negative</u>, <u>positively homogeneous</u>, <u>convex</u> function of γ for $\gamma \in \{\nu \geq 0\}$. Additionally at any point $\gamma > 0$, f^* is a differentialble function and $\nabla f^*(\gamma|X) < 0$.

Since $f^*(\gamma|X)$ is a positively homogeneous convex function, X is a convex set and $f^*(\gamma|X)$ is its support function at $\gamma \in D'$. (See corollary 13.2.1 [11]). Additionally, X is not a polyhedral set since $f^*(\gamma|X)$ is not a polyhedral function (see corollary 19.2.1 [11]). Moreover, X is a convex set generated by an infinite number of supports, one for each γ in D'. Observe that D' is a polyhedral cone.

Lemma 4.3 indicates the equivalence of the set X and the set U as defined by (4.2), and Lemma 4.1 established the equivalence between the set U and the set U' for which a dual representation was desired. U is

the desired dual representation of the set U'. Recall that at the beginning of this section, there was for each $\ell = 1,2,\ldots,N$ a set U'_ℓ similar in nature to U', but with the vector t replaced by t^ℓ, the set $\{1,2,\ldots,n'\}$ replaced by $\{ i \in [k],\ k = 1,2,\ldots,p' \}$ and the set $\{1,2,\ldots,p'\}$ replaced by P_ℓ. The corresponding dual representations U_ℓ and X_ℓ for U'_ℓ are given by lemmas 4.1 and 4.3 respectively once the correct changes in variables and index sets are taken into consideration, that is,

$$U_\ell = \Big\{ u^\ell > 0,\ u^\ell \in R^{o([P_\ell])} \ \Big|\ \prod_{i\in[P_\ell]} \Big(\frac{u_i}{\gamma_i}\Big)^{\gamma_i} \prod_{k\in P_\ell} \lambda_k(\gamma)^{\lambda_k(\gamma)} \leq 1 \text{ for each } \gamma^\ell \in D_\ell \Big\}$$

where $D_\ell \triangleq \Big\{ \gamma^\ell \geq 0,\ \gamma^\ell \in R^{o([P_\ell])} \ \Big|\ \sum_{i\in[P_\ell]} a_{ij}\gamma_i = 0 \quad j = m(\ell),\ldots,n(\ell) \Big\}$

and $X_\ell \triangleq \Big\{ x^\ell \in R^{o([P_\ell])} \ \Big|\ \sum_{i\in[P_\ell]} \gamma_i x_i \leq \sum_{k\in P_\ell} \Big(\sum_{i\in[k]} \gamma_i \log \gamma_i - \lambda_k(\gamma) \log \lambda_k(\gamma) \Big)$ for all $\gamma^\ell \in D_\ell \Big\}$

Finally by writing the definition of u^ℓ in terms of z and the equivalence of Z'_ℓ and U'_ℓ, the original set Z'_ℓ can be dually represented, as follows:

$u^\ell \in U'_\ell$ iff $u^\ell \in U_\ell$ that is, $u^\ell \in U_\ell$ iff (4.4) holds where

$$(4.4) \qquad \prod_{i\in[P_\ell]} \Big(\frac{u_i}{\gamma_i}\Big)^{\gamma_i} \prod_{k\in P_\ell} \lambda_k(\gamma)^{\lambda_k(\gamma)} \leq 1 \text{ for each } \gamma^\ell \in D_\ell \ .$$

Since $u_i^\ell \triangleq c_i \prod_{j=1}^{n(0)} z_j^{a_{ij}} \quad i \in [P_\ell]$, then $u^\ell \in U_\ell$ iff z satisfies (4.5) where

$$(4.5) \qquad \prod_{i\in[P_\ell]} \Big(\frac{c_i}{\gamma_i}\Big)^{\gamma_i} \prod_{k\in P_\ell} \lambda_k(\gamma)^{\lambda_k(\gamma)} \cdot \prod_{j=1}^{n(0)} z_j^{\sum_{i\in[P_\ell]} a_{ij}\gamma_i} \leq 1, \text{ for each } \gamma^\ell \in D_\ell$$

(4.6) Let $C(\gamma^{\ell}) \triangleq \prod_{i \in [P_{\ell}]} \left(\frac{c_i}{\gamma_i}\right)^{\gamma_i} \prod_{k \in P_{\ell}} \lambda_k(\gamma)^{\lambda_k(\gamma)}$ and $\xi_j(\gamma^{\ell}) \triangleq \sum_{i \in [P_{\ell}]} a_{ij}\gamma_i$

for each $j = 1,2,\ldots,n(0)$. Using (4.6), $u^{\ell} \in U_{\ell}$ iff z satisfies (4.7) where

$$(4.7) \qquad C(\gamma^{\ell}) \prod_{j=1}^{n(0)} z_j^{\xi_j(\gamma^{\ell})} \leqq 1 \text{ for each } \gamma^{\ell} \in D_{\ell} .$$

Using (4.7) the dual representation of the set Z_{ℓ}' for each $\ell=1,2,\ldots,N$ is the set Z_{ℓ} where

$$(4.8) \qquad Z_{\ell} \triangleq \left\{ z > 0,\ z \in R^{n(0)} \mid C(\gamma^{\ell}) \prod_{j=1}^{n(0)} z_j^{\xi_j(\gamma^{\ell})} \leqq 1 \text{ for all } \gamma^{\ell} \in D_{\ell} \right\}$$

Observe that the following Lemma is immediate:

Lemma 4.5: $z \in Z_{\ell}$ iff $u^{\ell} \in U_{\ell}$ iff $x^{\ell} \in X_{\ell}$

where $x_i^{\ell} \triangleq \log u_i^{\ell} \qquad i \in [P_{\ell}]$

$$u_i^{\ell} \triangleq c_i \prod_{j=1}^{n(0)} z_j^{a_{ij}} \qquad i \in [P_{\ell}] .$$

5. THE MASTER PROGRAM:

Using the dual representations of $SA_{\ell}(z)$ and Z_{ℓ}' given by (3.3) and (4.8) respectively program MA can be written as follows:

$$(5.1) \qquad \text{MIN } g_0(z) + \sum_{\ell=1}^{N} \text{MAX}_{\delta^{\ell} \in \Delta_{\ell}} \left\{ K(\delta^{\ell}) \prod_{j=1}^{n(0)} z_j^{\xi_j'(\delta^{\ell})} \right\}$$

subject to:

$$(5.2) \qquad C(\gamma^{\ell}) \prod_{j=1}^{n(0)} z_j^{\xi_j(\gamma^{\ell})} \leqq 1 \quad \text{for all } \gamma^{\ell} \in D_{\ell} \qquad \ell = 1,2,\ldots,N$$

$$(5.3) \qquad g_k(z) \leqq 1 \qquad k \in P_0$$

$$z > 0$$

Program MA can be further transformed through the introduction of new variables $(y_\ell,\ \ell = 1,2,\ldots,N)$ yielding

Program MTA:

$$(5.4) \qquad \text{MIN } g_0(z) + \sum_{\ell=1}^{N} y_\ell$$

subject to:

$$(5.5) \qquad K(\delta^\ell) \prod_{j=1}^{n(0)} z_j^{\xi_j(\delta^\ell)} \leq y_\ell \quad \text{for every } \delta^\ell \in \Delta_\ell \text{ and } \ell=1,2,\ldots,N$$

$$(5.6) \qquad C(\gamma^\ell) \prod_{j=1}^{n(0)} z_j^{\xi_j(\gamma^\ell)} \leq 1 \quad \text{for every } \gamma^\ell \in D_\ell \text{ and } \ell=1,2,\ldots,N$$

$$(5.7) \qquad g_k(z) \leq 1$$

$$(5.8) \qquad z > 0,\ y > 0.$$

Program MTA is a posynomial programming problem in the variables $z = (z_1,\ldots,z_{n(0)})$ and $y = (y_1,\ldots,y_N)$ with one term posynomial constraint (5.5) for every feasible solution to $SB_\ell(z)$, $\ell = 1,2,\ldots,N$; and one term posynomial constraint (5.6) for every vector in the polyhedral cone D_ℓ for each $\ell = 1,2,\ldots,N$. There are a very large number of these constraints.

Perhaps it is useful to review our procedure at this point. Program A is the original problem and program B is its dual. A projection technique is used whereby program A is rewritten as program MA. Two things are important about (MA): (i) the constraints $z \in Z'$ and (ii) the subproblems $(SA_\ell(z),\ \ell = 1,2,\ldots,N)$ it generates. In the subproblems, z is treated as a parameter. If $z \in Z'$, a finite optimal solution to each of the

subproblems can be found. It becomes important, then, to find a useful characterization of Z'. The equivalence of treating the u_i and z as parameters, since the u_i depends only on the elements of z and some constants, is recognized. Sets (U'_ℓ, ℓ=1,2,...,N) are then defined representing the sets of values of u^ℓ that leave subproblem ℓ consistent, for each ℓ = 1,2,...,N. Conditions for U'_ℓ to be non-empty and for $u^\ell \in U'_\ell$ were given in Theorem 4.1. These conditions were transformed into posynomial type constraints in Lemma 4.1 and 4.2. The constraints were further transformed by taking logs in Lemma 4.3 with the introduction of variables $x^\ell \triangleq \log u^\ell$. The results are convex sets X_ℓ for ℓ = 1,2,...,N. Then $u^\ell \in U'_\ell$ iff $x^\ell \in X_\ell$ for each ℓ = 1,2,...,N. Lemma 4 demonstrates the equivalence between consistency of $SA_\ell(z)$ and $x^\ell \in X_\ell$. All these manimpulations result in constraints (5.2).

The posynomial dual problems $SB_\ell(z)$ of $SA_\ell(z)$, ℓ = 1,2,...,N provide a dual representation of $SA_\ell(z)$, ℓ = 1,2,...,N given by (3.3). (5.2) and (3.3) ultimately result in (5.5) and (5.6) in program MTA. Of course, only the solution for each $z \in Z'$ that minimizes (5.1) is of ultimate interest. When (5.1) is transformed to (5.4) in program MTA, a constraint (5.5) appears for each feasible solution of subproblems $SB_\ell(z)$, ℓ = 1,2,...,N.

In order to develop a manageable algorithm for program MTA, a "relaxation procedure" is employed which is begun by ignoring some or all of the constraints (5.5) and (5.6) and solving the relaxed version of MTA. The solutions obtained in this way are used in each of the subproblems to test whether a solution to the relaxed version of MTA is optimal for A according to Theorem 2.1. This procedure has been

suggested by Geoffrion [4] and Lasdon [8], and was developed by Benders [1] for mixed integer programs.

Assume $(\hat{z}, \hat{y})$ is an optimal solution to a relaxed version of MTA at a certain step of the algorithm. The following illustrates the test for optimality. The first step is to solve $SB_\ell(\hat{z})$, $\ell = 1,2,\ldots,N$; then either the value of $v_\ell(\hat{z})$ is finite positive or unbounded for $\ell = 1,2,\ldots,N$.

On the one hand, assume that $v_\ell(\hat{z})$ is finite and positive for $\ell = 1,2,\ldots,N$. Let $(\hat{\delta}^\ell, \ell = 1,2,\ldots,N)$ solve $SB_\ell(\hat{z})$, $\ell = 1,2,\ldots,N$. If for each $\ell = 1,2,\ldots,N$,

$$(5.9) \qquad K(\hat{\delta}^\ell) \prod_{j=1}^{n(0)} \hat{z}_j^{\Xi_j^!(\hat{\delta}^\ell)} \leq \hat{y}_\ell,$$

then $(\hat{z}, \hat{y})$ is optimal in MTA. Let $\hat{t} = (\hat{t}^\ell, \ell = 1,\ldots,N)$ solve the corresponding primal subproblems $SA_\ell(\hat{z})$, $\ell = 1,2,\ldots,N$. Theorem 2.1 assures that $(\hat{z}, \hat{y})$ solves program A.

However, if for some subset $S_\Delta \subseteq L \triangleq \{1,2,\ldots,N\}$, (5.9) is violated, then the set of constraints to be added to the relaxed form of MTA at the next iteration is

$$(5.10) \qquad K(\hat{\delta}^\ell) \prod_{j=1}^{n(0)} z_j^{\Xi_j^!(\hat{\delta}^\ell)} \leq y_\ell \qquad \ell \in S_\Delta \ .$$

A new value of (z,y) is obtained that minimizes the updated version of MTA which includes constraints (5.10).

On the other hand, suppose $v_\ell(\hat{z})$ is unbounded, for some subset S_0 of the subsystems that is, the corresponding program $SA_\ell(\hat{z})$ is feasible for $z = \hat{z}$. Find the generator, $\hat{\gamma}^\ell$, for each cone D_ℓ such that $v_\ell(\hat{z})$ is unbounded. Add the following constraints to the relaxed version of MTA, i.e.,

$$(5.11) \qquad C(\hat{\gamma}^{\ell}) \cdot \prod_{j=1}^{n(0)} z_j^{\xi_j(\hat{\gamma}^{\ell})} \leq 1 \qquad \ell \in S_D$$

A new value of (z,y) is then obtained that minimizes the modified version of MTA.

Observe that in order to generate (5.11) the algorithm to solve the dual subproblems $SB_{\ell}(z)$ should be able to develop the generator vector $\hat{\gamma}^{\ell}$ that makes $\nu_{\ell}(z)$ unbounded, in case that for a given z, $SA_{\ell}(z)$ is inconsistent. Most dual type algorithms addressed to solve $SA_{\ell}(z)$ by means of solving $SB_{\ell}(z)$ will produce $\hat{\gamma}^{\ell}$ as a byproduct. The procedure is further described in the following algorithm:

THE ALGORITHM:

Step 1: Initiate the procedure by solving MTA with only a few (or no) constraints (5.5) and (5.6). Set $y = \underline{y} > 0$, a lower bound for vector y. Let (z^0, y^0) solve the relaxed MTA. Set I = 0.

Step 2: Solve subproblems $SA_{\ell}(z^I)$, $SB_{\ell}(z^I)$, $\ell = 1,2,\ldots,N$.

Step 2.1: If $\nu_{\ell}(z^I) > 0$ finite for every $\ell = 1,2,\ldots,N$, let $\delta^{I\ell}$ and $t^{I\ell}$ solve respectively $SB_{\ell}(z^I)$ and $SA_{\ell}(z^I)$ for each $\ell = 1,2,\ldots,N$. Then

(i) If (5.9) is satisfied for $(\delta^{I\ell}, z^I, y^{I\ell}; \ell = 1,2,\ldots,N)$, then (z^I, y^I) is optimal in program MTA and (z^I, t^I) solves program A, hence, terminate.

(ii) If (5.9) is not satisfied for a subset S_{Δ}, generate a set of constraints (5.10) with $\delta^{I\ell}$; $\ell \in S_{\Delta}$; update the relaxed version of MTA. Go to Step 3.

Step 2.2: If $\nu_{\ell}(z^I) = +\infty$ for $\ell \in S_D$,

(i) Let $\delta^{I\ell}$ and $t^{I\ell}$ solves $SB_{\ell}(z^I)$ and $SA_{\ell}(z^I)$ for $\ell \in S_D$.

If (5.9) is not satisfied for a subset $S'_\Delta \subseteq L - S_D$, generate a set of constraints (5.10) with $\delta^{I\ell}$, $\ell \in S'_\Delta$.

(ii) For each $\ell \in S_D$, find $\gamma^{I\ell}$ i.e., the generator of the cone D_ℓ and generate a set of constraints (5.11). With constraints generated at (i) and (ii), update the relaxed version of MTA.

Step 3: Solve the new relaxed version of MTA. Let (z^{I+1}, y^{I+1}) be optimal in the current relaxed version of MTA. Let I = I+1. Go to Step 2.

Note that program MTA is a minimization problem and at each iteration constraints are added to the relaxed version of MTA. Therefore, the value of the objective function increases or remains the same at each subsequent step in the algorithm. Thus, a non-decreasing sequence

$$\left\{ g_0(z^I) + \sum_{\ell=1}^{N} y_\ell^I \right\} \qquad I = 0,1,2,.....$$

is generated. At each step when (5.9) is violated,

$$K(\delta^{\ell I}) \prod_{j=1}^{n(0)} z_j^{I\,\Gamma_j'(\delta^{\ell I})} > y_\ell^I \,, \quad \ell \in S_\Delta$$

hence an upper bound on the value of the y_ℓ variable at iteration M exists, i.e.,

$$y_\ell^M \leq \min_{0 \leq I \leq M} \left\{ K(\delta^{\ell I}) \prod_{j=1}^{n(0)} z_j^{I\,\Gamma_j'(\delta^{\ell I})} \right\} .$$

The sequence generated by the algorithm is bounded above and non-decreasing; hence the algorithm converges to an optimal solution of program MTA. However, an optimal solution of MTA may not be obtained in a finite number of iterations since it takes an infinite

number of supports (5.6) to completely characterize each set (z_ℓ, ℓ=1,2,...,N). The algorithm can lead to expenditures of a great deal of time in step 2.2 in an attempt to remain feasible while disregarding the minimization of (5.4).

This difficulty can be avoided when MTA is a linear program. This is the special case described in part II, where inner-linearization of the set X_ℓ is employed to permit removal of step 2.2 completely. This implies more work when solving the relaxed master programs since the optimal z will be forced to lie inside of Z'_ℓ. In part II a special case of program A, labeled program D, is considered. This program has the attribute that MTA is a linear program. An algorithm for this special case is developed using inner linearization of the sets X_ℓ. Relaxation is applied to constraints (5.5) only.