

Peterson, Elmor

Working Paper

A Handbook of Geometric Programming

Discussion Paper, No. 101

Provided in Cooperation with:

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

Suggested Citation: Peterson, Elmor (1974) : A Handbook of Geometric Programming, Discussion Paper, No. 101, 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/10419/220461>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

DISCUSSION PAPER NO. 101
A HANDBOOK OF GEOMETRIC PROGRAMMING*

by
Elmor L. Peterson**

July 15, 1974

* To be published as chapter 5 in the forthcoming Handbook of Operations Research (Elmaghraby and Moder, eds.), Van Nostrand Reinhold Company, New York.

** Department of Industrial Engineering/Management Sciences and Department of Mathematics, Northwestern University, Evanston, Illinois 60201. Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-73-2516A.

The United States Government is authorized to reproduce and distribute reprints of this paper for Governmental purposes notwithstanding any copyright notation hereon.

Table of Contents

<u>CHAPTER 5. GEOMETRIC PROGRAMMING</u>	<u>Page</u>
5.1 INTRODUCTION-----	1
5.2 PROBLEM FORMULATION AND EXAMPLES -----	1
5.2.1 The Unconstrained Case -----	2
5.2.2 The Constrained Case -----	11
5.2.3 A Summary -----	19
5.3 BASIC THEORY -----	20
5.3.1 The Unconstrained Case -----	20
5.3.1.1 Optimality conditions -----	21
5.3.1.2 The conjugate transformation, subgradients and convex analysis -----	22
5.3.1.3 Lagrangian saddle points -----	27
5.3.1.4 Duality-----	31
5.3.1.5 Parametric programming and post-optimality analysis-----	40
5.3.1.6 Decomposition principles -----	50
5.3.2 Relations between the Constrained and Unconstrained Cases -----	61
5.3.3 The Constrained Case -----	62
5.3.3.1 Optimality conditions -----	63
5.3.3.2 Lagrangian saddle points -----	65
5.3.3.3 The geometric inequality -----	67
5.3.3.4 Duality -----	68
5.3.3.5 Parametric programming and post-optimality analysis-----	75
5.3.3.6 Decomposition principles -----	78
5.4 REFERENCES -----	78

5.1 INTRODUCTION

Since its inception by Zener [1961,1962] and Duffin [1962a,1962b], geometric programming has undergone rapid development, especially with the appearance of the first book on the subject by Duffin, Peterson, and Zener [1967]. Although its essence and scope have recently been broadened and amplified by Peterson [1970,1973a,1975], major advances in theory, computation, and application are still occurring as more workers enter the field. The main purpose of this chapter is to summarize the present state of the subject and to indicate some of the directions in which it is developing. To keep the length of this chapter within reasonable limits, only the most fundamental aspects are presented, and then only within the context of n -dimensional Euclidean space E_n . (Just to list all relevant papers would itself require several additional pages.) Consequently, some important topics have been omitted, but a much more thorough treatment [Peterson, 1977] is presently in preparation. Since that treatment is to completely subsume the present treatment, only prior references (when available) are cited herein.

With respect to notation, the context alone dictates whether a given vector v in E_n is to be interpreted as a "column vector", or as a "row vector". In all cases, the symbol $\langle \cdot, \cdot \rangle$ indicates the usual "inner product" function.

5.2 PROBLEM FORMULATION AND EXAMPLES

Geometric programming provides a mechanism for formulating and studying in "separable" form many important (usually inseparable) optimization problems. The key to this mechanism is the exploitation of the linearities that are present in a given problem. Such linearities frequently appear as linear equations or linear inequalities, but they can also appear in much more subtle guises, such as matrices associated with nonlinearities.

We shall begin with unconstrained problems and then proceed to (the more

complicated) constrained problems. In each case we consider important examples that arise in operations research.

5.2.1 The Unconstrained Case. Classical optimization theory and ordinary mathematical programming are concerned with the minimization (or maximization) of an arbitrary real-valued function g over some given subset \mathcal{S} of its non-empty domain $\mathcal{C} \subseteq E_n$. In geometric programming, the subset \mathcal{S} is required to be the intersect of the function domain \mathcal{C} with an arbitrary cone $\mathcal{X} \subseteq E_n$ (which is, in fact, a vector space for most examples). For purposes of easy reference and mathematical precision, the resulting geometric programming problem \mathcal{A} is now given the following formal definition in terms of classical terminology and notation.

PROBLEM \mathcal{A} . Using the "feasible solution" set

$$\mathcal{S} \triangleq \mathcal{X} \cap \mathcal{C},$$

calculate both the "problem infimum"

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

and the "optimal solution" set

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

Each optimization problem can generally be formulated as problem \mathcal{A} in more than one way by suitably choosing the function g and the cone \mathcal{X} . For example, one can always let g be the "objective function" for the given problem simply by choosing \mathcal{X} to be E_n , but that choice is generally not the best possible choice. The reason is that most problems involve a certain amount

of linearity (due to the presence of linear equations, linear inequalities, matrices, etc.), which can be conveniently handled through the introduction of an appropriate nontrivial subcone $\mathcal{X} \subset E_n$. The presence of such a subcone \mathcal{X} is one of the distinguishing features of geometric programming.

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

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

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

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

The presence of the "exponent matrix" (a_{ij}) (which is of course associated with algebraic nonlinearities) is the key to applying geometric programming to signomial optimization. To effectively place the problem of minimizing $P(\vec{t})$

in the format of problem \mathcal{A} , simply make the change of variables

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

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

$$\begin{aligned} \mathcal{C} &\triangleq E_n, \\ \mathcal{G}(x) &\triangleq \sum_{i=1}^n c_i e^{x_i}, \end{aligned}$$

and

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

The advantages of studying this problem \mathcal{A} rather than its signomial predecessor are numerous. For example, unlike the signomial P , the exponential function \mathcal{G} is completely separable (in that it is a sum of terms, each of which depends on only a single independent variable x_i). Moreover, if P is actually a posynomial, then \mathcal{G} is of course strictly convex (even though P itself clearly need not even be convex). Consequently, if t^* minimizes a posynomial P , then the corresponding x^* must be a unique optimal solution to problem \mathcal{A} ; in which event the set of all t that minimize P can be obtained from x^* simply by solving the displayed system of equations (a task that is relatively **easy** because the system is clearly linear in terms of $\log t_j$, $j = 1, 2, \dots, m$). In [Duffin, Peterson, Zener, 1967], [Avriel, Williams, 1970], [Duffin, Peterson, 1972a, 1972b, 1972c, 1973], as well as [Abrams, Bunting, 1975], and some of the references cited therein, these properties and others that are too complicated to describe here have been combined into a very comprehensive existence, uniqueness, and characterization theory for signomial (and especially posynomial) optimization. Moreover, in [Falk, 1973] the complete separability induced

into signomial optimization forms the basis for a branch-and-bound algorithm that converges to globally optimal solutions to (intrinsically nonconvex) signomial optimization problems.

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

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

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

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

$$H = D^t D - \theta^t \theta,$$

where t indicates the transpose operation. In terms of D and θ the quadratic function

$$Q(z) = (\frac{1}{2}) (\langle Dz, Dz \rangle - \langle \theta z, \theta z \rangle) + \langle h, z \rangle.$$

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

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

$$c \triangleq E_{2m+1},$$

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

and

$$x \triangleq \text{column space of } \begin{bmatrix} D \\ \theta \\ h \end{bmatrix}.$$

Notice that, unlike the quadratic function Q , the quadratic function \mathcal{G} is completely separable, a fact that can be exploited both theoretically and computationally.

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

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

where b_i and p_i are arbitrary constants, and $|\cdot|$ designates the absolute value function. Notice that the function \mathcal{G} is still completely separable and can be specialized to the quadratic case by choosing $b_i = 0$ and $p_i = 2$ for each i .

Another interesting specialization is obtained by choosing $p_i = p$ for each i while choosing $D = 0$ and $h = 0$. The resulting problem consists essentially of finding the "best ℓ_p -norm approximation" to the fixed vector (b_1, \dots, b_m) by vectors in the column space of the matrix D , a fundamental problem in linear regression analysis.

A detailed analysis of this rather broad class of optimization problems can be found in [Peterson, Ecker, 1970] and the references cited therein.

Example 3. Our third example comes from the optimal location of a new facility relative to existing facilities. We suppose that there are D existing facilities with fixed locations b^1, b^2, \dots, b^D in E_n , and we assume that for each facility i there is a cost $d_i(z, b^i)$ of choosing the new facility location z relative to b^i . In many instances the functions d_i are just "metrics" that reflect the cost of shipping materiel between the two locations. (Such metrics are usually determined by the available transportation systems.) The problem then is to choose a new

location z that minimizes the total cost $d(z) = \sum_{i=1}^p d_i(z, b^i)$.

In this problem statement there is no matrix that serves as the key to effectively applying geometric programming. However, minimizing $d(z)$ is clearly equivalent to solving problem \mathcal{A} when

$$c \triangleq E_{pm},$$

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

and

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

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

Notice that, unlike the function d , the function g is at least partially separable in that it is a sum of terms, each of which depends on only a single independent vector variable x^i . This separability occurs in even more complicated location problems and has been exploited both theoretically and computationally in [Peterson, Wendell, 1975] and the references cited therein.

Example 4. Our fourth example comes from discrete optimal control with linear dynamics (or dynamic programming with linear transition equations). We suppose that for each "stage" i there is a cost $g_i(r^i, a^i)$ that depends on the i 'th "state" r^i and the i 'th "decision" a^i , where the domain of the cost function g_i is the cartesian product $R_i \times D_i$ of the i 'th "state set" R_i and the i 'th "decision set" D_i . We also suppose that the "initial state" r^1 is determined by the "initial decision" a^1 through the equation $r^1 = B_1 a^1$ and that each subsequent state r^i is determined by both the i 'th decision a^i and the $(i-1)$ 'th state r^{i-1} through the "transition equation" $r^i = A_i r^{i-1} + B_i a^i$, where A_i and B_i

are constant matrices. Given that there is a total of p stages, the problem is to make sequential decisions d^i that minimize the total cost

$$\sum_{i=1}^p g_i(r^i, d^i).$$

The presence of the matrices A_i and B_i is the key to applying geometric programming to discrete optimal control. To effectively place the preceding control problem in the format of problem \mathcal{A} , simply let

$$x = (r^1, d^1, r^2, d^2, \dots, d^{p-1}, r^p, d^p)$$

and then observe that the preceding control problem is equivalent to problem

\mathcal{A} when

$$C \triangleq \prod_{i=1}^p (R_i \times D_i),$$

$$g(x) \triangleq \sum_{i=1}^p g_i(r^i, d^i),$$

and

$$\mathcal{X} \triangleq \{x \mid r^1 = B_1 d^1, \text{ and } r^i = A_i r^{i-1} + B_i d^i, \quad i = 2, \dots, p\}.$$

The partial separability of g and the "sparsity" of the matrix whose columns span \mathcal{X} has been exploited both theoretically and computationally in [Dinkel, Peterson, 1975] and the references cited therein.

Example 5. Our fifth example comes from an analysis of multicommodity transportation networks. We consider a "graph" whose "arcs" are enumerated from 1 through n in such a way that for $1 \leq i \leq v$ (return) arc i connects the origin and destination of commodity i , while for $v + 1 \leq i \leq n$ (roadway) arc i represents a collection of unidirectional lanes over which traffic can flow between two adjacent (intersectional) "nodes". Each return arc i is "directed" from the destination of the corresponding commodity i back to the origin of the same commodity i ,

and each roadway arc i has the given direction in which traffic is permitted to flow.

Each commodity i can flow from its origin to its destination only over certain (predetermined) feasible (roadway) "paths" p^j , which are enumerated by a finite index set $[i]$. Each path p^j is, in essence, an n -vector whose k 'th component p_{k}^j is to be identified with arc k . In particular, for a given j in $[i]$, component p_{k}^j is one for (return arc) $k = i$, zero for all other (return arcs) k between 1 and v , and either one or zero for all (roadway arcs) k between $v + 1$ and n , depending respectively on whether roadway arc k is or is not part of path p^j .

For notational convenience in describing feasible flow patterns, all feasible paths p^j are enumerated sequentially by letting $[i] = \{m_i, m_i + 1, \dots, n_i\}$, where $1 = m_1 \leq n_1$, $n_1 + 1 = m_2 \leq n_2, \dots, n_{v-1} + 1 = m_v \leq n_v$. Thus, there is a total of m feasible paths over which traffic can flow, where $m \triangleq n_v$. Moreover, a potential commodity flow pattern z is then just a non-negative m -vector whose j 'th component z_j is simply the "input flow" on path p^j of that commodity i for which j is in $[i]$. Of course, each potential commodity flow pattern z produces a potential total flow pattern $x \triangleq \sum_{j=1}^m z_j p^j$, whose k 'th component x_k is simply the resulting total flow of all commodities on arc k . The feasible flow patterns are then those potential flow patterns for which $x_i = d_i$ for $1 \leq i \leq v$, where d_i is a given (non-negative) total input flow of commodity i .

We now assume that traffic flow on each roadway arc i produces a cost $g_i(x_i)$ that depends only on the total flow x_i . The problem then is to determine those feasible flow patterns that minimize the total cost $\sum_{i=v+1}^n g_i(x_i)$. Of course, this problem is relevant only if the given transportation network can be centrally controlled - which is usually not the case for highway networks.

For highway networks, it is far more realistic to use the same type of separable objective function, but let each function g_i be the (indefinite) integral of a travel cost function c_i (rather than let each g_i be a cost function itself). For many highway networks, the most realistic travel cost function c_i is actually just the travel time $c_i(x_i)$ required to traverse arc i when it is carrying a total traffic flow x_i . In any case, the reason for integrating the cost functions c_i prior to forming the objective function $\sum_{i=v+1}^n g_i$ is that under certain (relatively weak) conditions, the resulting optimal flow patterns (which are, of course, not usually cost optimal) are in a state of "Wardrop equilibrium"; that is, the origin-to-destination travel cost (e.g. the origin-to-destination travel time) for a given commodity i is identical on all paths used by that commodity and is not greater than what it would be on its unused feasible paths (given the same total flow pattern). Such flow patterns are of interest to highway traffic analysts, because highway traffic scientists contend that the traffic flow patterns of many complicated real-world highway networks are in, or at least tend toward, a state of Wardrop equilibrium.

In any event, the presence of the feasible path vectors p^j is the key to applying geometric programming to the preceding network problems. To effectively place those problems in the format of problem \mathcal{A} , first enlarge the set of functions

g_i by letting

$$c_i \triangleq \begin{cases} \{d_i\} & 1 \leq i \leq v \\ [0, +\infty) & v+1 \leq i \leq n \end{cases}$$

and

$$g_i(x_i) \triangleq \begin{cases} 0 & 1 \leq i \leq v \\ g_i(x_i) & v+1 \leq i \leq n. \end{cases}$$

Then, observe that the preceding network problems are equivalent to problem

\mathcal{A} when

$$c = \sum_{i=1}^n c_i,$$

$$g(x) = \sum_{i=1}^n g_i(x_i),$$

and

$$\mathcal{X} = \left\{ x = \sum_{j=1}^m z_j p^j \mid z_j \geq 0 \text{ for } 1 \leq j \leq m \right\}.$$

Note that \mathcal{X} is not generally a vector space, but is instead a polyhedral cone generated by the feasible path vectors p^j . This important class of problems is studied more thoroughly in [Hall, Peterson, 1975] and some of the references cited therein.

A closely related (but totally different) class of network flow problems occurs in the context of electric and hydraulic networks. Such problems involve only a single commodity (electricity or fluid) and can be effectively placed in the format of problem \mathcal{A} in the following way: let $c_i(x_i)$ be the voltage drop or pressure drop, respectively, across arc i as a function of the commodity flow x_i in arc i (in which event $g_i(x_i)$ is termed the "content" of arc i and is frequently just the power dissipated in arc i), and let \mathcal{X} be the vector space of all (non-unidirectional) feasible flows (i.e. all those flows that satisfy the "Kirchoff current conservation laws"). Such problems are studied more thoroughly in [Duffin, 1947], [Minty, 1960], and [Rockafellar, 1967b], and the references cited therein.

The reader who wishes to avoid the complications inherent with constraints can skip the next section and begin again with section 5.2.3 on page 19.

5.2.2 The Constrained Case. To generalize geometric programming by incorporating explicit constraints into the preceding problem formulation, we introduce two non-intersecting (possibly empty) positive-integer index sets I and J with finite cardinality $o(I)$ and $o(J)$ respectively. In terms of these index sets

I and J we also introduce the following notation and hypotheses:

(1a) For each $k \in \{0\} \cup I \cup J$ there is a function g_k with domain $C_k \subseteq E_{n_k}$, and there is a set $D_j \subseteq E_{n_j}$ for each $j \in J$.

(2a) For each $k \in \{0\} \cup I \cup J$ there is an independent vector variable x^k in E_{n_k} , and there is an independent vector variable κ with components κ_j for each $j \in J$.

(3a) x^I denotes the cartesian product of the vector variables x^i , $i \in I$, and x^J denotes the cartesian product of the vector variables x^j , $j \in J$. Hence, the cartesian product $(x^0, x^I, x^J) \triangleq x$ of x^0 , x^I , and x^J is an independent vector variable in E_n , where

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

(4a) there is a cone $X \subseteq E_n$.

For purposes of easy reference and mathematical precision, the resulting geometric programming problem A is now given the following formal definition in terms of classical terminology and notation.

PROBLEM A. Consider the objective function G whose domain

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

and whose functional value

$$G(x, \kappa) \triangleq g_0(x^0) + \sum_J g_j^+(x^j, \kappa_j),$$

where

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

and

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

Using the feasible solution set

$$S \triangleq \{ (x, \kappa) \in C \mid x \in X, \text{ and } g_i(x^i) \leq 0, i \in I \},$$

calculate both the problem infimum

$$\varphi \triangleq \inf_{(x, \kappa) \in S} G(x, \kappa)$$

and the optimal solution set

$$S^* \triangleq \{ (x, \kappa) \in S \mid G(x, \kappa) = \varphi \}.$$

Of course, the unconstrained case occurs when $I = J = \emptyset$, $g_0: C_0 \triangleq \mathcal{C}$, and $X \triangleq \mathcal{X}$.

When $D_j = E_{n_j}$ (which is frequently the situation), a simplification results from noting that $\sup_{d^j \in D_j} \langle x^j, d^j \rangle < +\infty$ if and only if $x^j = 0$, in which event $\sup_{d^j \in D_j} \langle x^j, d^j \rangle = 0$. In particular then, the functional domain

$$C_j^+ = \{ (x^j, \kappa_j) \mid \kappa_j \geq 0 \text{ and } x^j \in \kappa_j C_j \};$$

and the functional values

$$g_j^+(x^j, \kappa_j) = \kappa_j g_j(x^j / \kappa_j),$$

with the understanding that $0g_j(0/0) \triangleq 0$.

In defining the feasible solution set S it is important to make a sharp distinction between the cone condition $x \in X$ and the constraints $g_i(x^i) \leq 0$, $i \in I$, both of which restrict the vector variable (x, κ) . In many cases the cone X is polyhedral (and hence is finitely generated); and in most examples of practical significance X is actually a vector space (and hence has a finite basis). Consequently, the cone condition $x \in X$ can frequently be automatically satisfied and therefore explicitly eliminated by a linear transformation of x that results from the introduction of generating vectors or basis vectors for X (whereas the generally nonlinear constraints $g_i(x^i) \leq 0$, $i \in I$, usually can not be explicitly eliminated by even a nonlinear transformation). Nevertheless, even when it is possible to do so, we do not explicitly eliminate the cone condition $x \in X$, because such a linear transformation would clearly introduce a common vector variable into the arguments of g_0 , g_i , and g_j^+ . Such a

common vector variable only tends to camouflage one of the extremely useful characteristics of geometric programming - its (partial) separability. Such separability is clearly present even when the functions $g_k: C_k$, $k \in \{0\} \cup I \cup J$ are inseparable.

Since each optimization problem can generally be formulated as problem A in more than one way by suitably choosing the functions $g_k: C_k$, $k \in \{0\} \cup I \cup J$ and the cone X, a very important aspect of applied geometric programming is the exploitation of this flexibility in such a way that a given inseparable problem is formulated as an equivalent problem A with as much function separability as possible. As in the unconstrained case, the key to such a formulation is usually the introduction of an appropriate nontrivial cone X to handle the linearities that are present in a given problem. Such linearities frequently appear as linear equations or linear inequalities, but they can also appear in rather subtle guises, such as matrices associated with nonlinearities.

The function separability induced in the objective function for each of the unconstrained examples given in section 5.2.1 can also be induced in any constraint function of the same general type. We now use signomial optimization to illustrate the general procedure for doing so.

Example 6. First, make the following choices:

$$I = \{1, 2, \dots, p\} \text{ and } J = \emptyset;$$

$$C_k \stackrel{\Delta}{=} E_{n_k}$$

and

$$g_k(x^k) \stackrel{\Delta}{=} \sum_{[k]} c_q e^{x_q} - d_k,$$

where

$$[k] = \{m_k, m_k + 1, \dots, n_k\}$$

and

$$1 = m_0 \leq n_0, \quad n_0 + 1 = m_1 \leq n_1, \quad \dots, \quad n_{p-1} + 1 = m_p \leq n_p = n;$$

$$X \stackrel{\Delta}{=} \text{column space of } [a_{qr}],$$

where

$[a_{qr}]$ is any $n \times m$ matrix.

Now, note that all functions in problem A are completely separable.

To relate problem A to constrained signomial optimization, explicitly eliminate the vector space condition $x \in X$ by the (essentially linear) transformation

$$x_q = \sum_{r=1}^m a_{qr} \log t_r, \quad q = 1, 2, \dots, n.$$

By virtue of the laws of exponents, problem A is now clearly equivalent to the following (generally inseparable) signomial optimization problem:

$$\text{Minimize } \sum_{[0]} c_q \prod_{l=1}^m t_r^{a_{qr}}$$

subject to

$$\sum_{[k]} c_q \prod_{l=1}^m t_r^{a_{qr}} \leq d_k, \quad k = 1, 2, \dots, p,$$

and

$$t > 0.$$

Of course, the preceding procedure is usually reversed in practice; that is, the signomial form of problem A tends to occur more often than problem A in real-world applications, but is transformed into problem A so that the complete separability of the resulting exponential functions can be exploited. Actually, signomial optimization problems (as well as more general "algebraic optimization problems") should usually be reduced to much simpler signomial optimization problems prior to their transformation to an appropriate problem A. To see how to reduce such problems to signomial optimization problems in which each signomial has at most two terms, both of which have the same sign, consult [Duffin, 1970] and [Duffin, Peterson, 1973, 1972b]. If such problems could be further reduced to signomial problems in which each signomial has only a single term, all algebraic optimization problems (and hence essentially

all optimization problems involving only continuous functions) could be reduced to (finite-dimensional) linear programming problems. Even though such a reduction will not be accomplished in the future, the reductions given in the preceding references are already starting to be exploited, both theoretically and computationally.

The reader who is interested in the applications of signomial and posynomial optimization should consult the recent book by Zener [1971] as well as the comprehensive list of papers compiled in [Rijckaert,1973].

The procedure for inducing function separability into constrained versions of each of the other four examples given in section 5.2.1 will be left to the imagination of the interested reader, who can also consult the references already cited in section 5.2.1.

Linear programming can be viewed as a special case of geometric programming in at least three different ways. We now present the easiest of the three ways.

Example 7. First, make the following choices:

$$J = \phi ;$$

$$g_0 : E_1 \rightarrow E_1 \quad \text{such that} \quad g_0(x^0) \triangleq x^0,$$

and

$$g_i : E_1 \rightarrow E_1 \quad \text{such that} \quad g_i(x^i) \triangleq x^i - b_i, \quad i \in I,$$

where the b_i , $i \in I$ are arbitrary constants;

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

where a is an arbitrary vector in E_m , M is an arbitrary $o(I) \times m$ matrix, and

\emptyset is any subset of $\{1, 2, \dots, m\}$. Now, note that the most complicated function g_k in problem A is just the simplest kind of affine function.

To relate problem A to linear programming, explicitly eliminate the cone condition $x \in X$ by the linear transformation used in the defining equation for X . Problem A is then clearly equivalent to the following very general linear programming problem:

$$\begin{aligned} & \text{Minimize } \langle a, z \rangle \\ & \text{subject to} \\ & \quad Mz \leq b, \\ & \quad z_j \geq 0, \quad j \in \emptyset. \end{aligned}$$

Two other ways in which to view linear programming from a geometric programming point of view are given in [Peterson, 1973a], but none of the three ways have yet had other than pedagogical influence on linear programming.

In all of the examples given here, the index set J is empty. Probably the most important example for which J is not empty is the "chemical equilibrium problem" -- a problem that lies outside the scope of operations research but is thoroughly discussed in [Peterson, 1973a] and the references cited therein. The reason for including J in the present problem formulation is that its inclusion is a prerequisite for the duality symmetry described in subsection 5.3.3.4.

The following example indicates the generality of geometric programming.

Example 8. First, make the following choices:

$$\begin{aligned} & J = \emptyset ; \\ & n_k = m \text{ and } C_k \stackrel{\Delta}{=} C_0, \quad k \in \{0\} \cup I, \end{aligned}$$

where C_0 is an arbitrary subset of E_m ;

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

where there is a total of $1+o(I)$ identity matrices U that are $m \times m$.

To relate problem A to ordinary programming, explicitly eliminate the vector space condition $x \in X$ by the linear transformation

$$\begin{pmatrix} x^0 \\ x^I \end{pmatrix} = \begin{bmatrix} U \\ U \\ \vdots \\ U \end{bmatrix} z.$$

Problem A is then clearly equivalent to the following very general "ordinary programming problem":

$$\begin{aligned} &\text{Minimize } g_0(z) \\ &\text{subject to} \\ &g_i(z) \leq 0, \quad i \in I \\ &z \in C_0. \end{aligned}$$

Thus, ordinary programming can be viewed as a special case of geometric programming.

Yet, in a certain sense, ordinary programming is no more special than geometric programming -- as can be seen by the following reverse specialization:

$$\begin{aligned} C_0 &\triangleq \left\{ z = (x, \kappa) \mid (x, \kappa) \in C \text{ and } x \in X \right\}, \\ g_0(z) &\triangleq G(x, \kappa), \\ g_i(z) &\triangleq g_i(x^i), \quad i \in I. \end{aligned}$$

Note though that the important structural features of geometric programming are obscured in this ordinary programming formulation. On the other hand, ordinary programming is actually made partially separable in its geometric programming formulation, namely:

$$\begin{array}{ll}
 \text{Minimize } & g_0(x^0) \\
 \text{subject to} & g_1(x^1) \leq 0, \quad g_2(x^2) \leq 0, \dots, \quad g_p(x^p) \leq 0 \\
 & x^0 - x^1 = 0 \\
 & \quad \quad x^1 - x^2 = 0 \\
 & \quad \quad \quad \cdot \quad \quad \quad \cdot \\
 & \quad \quad \quad \quad \quad \quad \cdot \\
 & \quad \quad \quad \quad \quad \quad \quad \quad \quad x^{p-1} - x^p = 0 \\
 & x^0, \quad x^1, \quad x^2, \quad \dots \quad x^{p-1}, \quad x^p \in C_0,
 \end{array}$$

where

$$\{1, 2, \dots, p\} = I.$$

5.2.3 A Summary. Experience seems to indicate that any optimization problem involving matrices, linear equations, affine sets, or cones (and even certain other optimization problems, such as optimal location problems) can probably be transformed into a geometric programming problem that is considerably more separable and hence much more amenable to analysis and solution than the original problem. This is especially true for each of the preceding examples, but the complete exploitation of this fact is way beyond the scope of this chapter.

In the remaining part of this chapter we present only the most basic theory of geometric programming -- a theory that is equally applicable to all problem classes and does not, for the most part, actually require sepa-

rability of the functions $g : C$ and $g_k : C_k$, $k \in \{0\} \cup I \cup J$. Such function separability does, however, become extremely useful when specific problem classes are to be thoroughly investigated and solved.

5.3 BASIC THEORY

The basic theory of geometric programming can be conveniently partitioned into several topics. Certain "optimality conditions" describe important properties possessed by all optimal solutions, and in many cases collectively characterize all optimal solutions. Appropriate "Lagrangians" provide important "saddle-point" characterizations of optimality, and can also be used to introduce the even more significant concepts of "duality". The latter topic provides important "existence and uniqueness theorems" for optimal solutions, as well as useful "algorithmic stopping criteria". Duality is also a key ingredient in "parametric programming" and "post-optimality analysis", which are in turn key ingredients in certain important "decomposition principles".

Since many important problem classes are unconstrained (e.g., the network flow problems given as example 5 in section 5.2.1), and since the theory for the unconstrained case is far simpler than that for the constrained case, we initially limit our attention to the unconstrained case. Actually, in doing so there is no loss of generality, as explained in section 5.3.2.

5.3.1. The Unconstrained Case. Let \mathcal{Y} be the "dual" of the cone \mathcal{X} ; that is,

$$\mathcal{Y} \triangleq \{y \in E_n \mid 0 \leq \langle x, y \rangle \text{ for each } x \in \mathcal{X}\} .$$

Rather elementary considerations show that \mathcal{Y} is generally a closed convex cone. Moreover, \mathcal{Y} is "polyhedral" (i.e., "finitely generated") when \mathcal{X} is polyhedral; and \mathcal{Y} is the "orthogonal complement" \mathcal{X}^\perp of \mathcal{X} when \mathcal{X} is actually a vector space. In fact, \mathcal{Y} can be computed via elementary linear algebra for each of the examples given in section 5.2.1.

Some of the following subsections can be omitted. In particular, either subsection 5.3.1.1 (on optimality conditions) or subsection 5.3.1.3 (on Lagrangian saddle points) can be omitted without serious loss of continuity. Moreover, subsection 5.3.1.2 can be omitted by those readers who are already sufficiently familiar with the "conjugate transformation, subgradients and convex analysis".

5.3.1.1. Optimality conditions. We begin with the following fundamental definition.

DEFINITION. A critical solution (stationary solution, equilibrium solution, P solution) for problem \mathcal{A} is any vector x^* that satisfies the following P optimality conditions

$$\begin{aligned}x^* &\in \mathcal{X} \cap \mathcal{C}, \\ \nabla g(x^*) &\in \mathcal{Y},\end{aligned}$$

and

$$0 = \langle x^*, \nabla g(x^*) \rangle.$$

If the cone \mathcal{X} is actually a vector space (which is the case for each of the examples given in section 5.2.1, except example 5), then $\mathcal{Y} = \mathcal{X}^\perp$ and hence the last P optimality condition $0 = \langle x^*, \nabla g(x^*) \rangle$ is obviously redundant and can be deleted from the preceding definition. Furthermore, if \mathcal{X} is actually the whole vector space E_γ (which is the situation in the unconstrained case of

ordinary programming), then $\mathcal{Y} = E_{\mathcal{Y}}^{\perp} = \{0\}$ and hence the remaining P optimality conditions clearly become the (more familiar) optimality conditions

$$x^* \in \mathcal{C} \text{ and } \nabla g(x^*) = 0.$$

The following theorem gives two convexity conditions that guarantee the necessity and/or sufficiency of the P optimality conditions for optimality.

Theorem 1. Under the hypothesis that g is differentiable at x^* ,

(1) given that \mathcal{X} is convex, if x^* is an optimal solution to problem \mathcal{A} , then x^* is a critical solution for problem \mathcal{A} (but not conversely),

(2) given that g is convex on \mathcal{C} , if x^* is a critical solution for problem \mathcal{A} , then x^* is an optimal solution to problem \mathcal{A} .

The proof of this theorem is not difficult but will of course be omitted.

It is worth noting that g is differentiable for most of the examples given in section 5.2.1. Moreover, \mathcal{X} is polyhedral and hence convex for each of those examples; and g is convex on \mathcal{C} for important special cases of each of those examples. Consequently, the P optimality conditions frequently characterize the optimal solution set \mathcal{A}^* for problem \mathcal{A} .

Characterizations of \mathcal{A}^* that do not require differentiability of g , but do require the concepts described in the following subsection, are given in subsection 5.3.1.3.

5.3.1.2. The conjugate transformation, subgradients and convex analysis.

The conjugate transformation evolved from the classical Legendre transformation but was first studied in great detail only rather recently by Fenchel [1949, 1951]. For a very thorough and modern treatment of both transformations see the recent book by Rockafellar [1970]. We now briefly describe only those of their properties that are relevant to geometric programming. All such properties

are quite plausible when viewed geometrically in the context of two and three dimensions.

The conjugate transformation maps functions into functions in such a way that the "conjugate transform" $\omega:\Omega$ of a given function $w:W$ has functional values

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

Of course, the domain Ω of ω is defined to be the set of all those vectors ζ for which this supremum is finite, and the conjugate transform $\omega:\Omega$ exists only when Ω is not empty.

The conjugate transform of a separable function is clearly the sum of the conjugate transforms of its individual terms - a fact that simplifies the conjugate transform computations for many of the geometric programming examples given in section 5.2.1. For purposes of illustration we now perform two of those computations in the convex case - the only case in which the conjugate transformation proves to be extremely effective.

Example 1. If

$$w(z) \triangleq \sum_{i=1}^{\eta} c_i e^{z_i} \text{ and } W \triangleq E_{\eta}, \text{ where } c_i > 0, i=1,2,\dots,\eta,$$

$$\text{then } \omega(\zeta) = \sup_{z \in E_{\eta}} [\langle \zeta, z \rangle - \sum_{i=1}^{\eta} c_i e^{z_i}] = \sum_{i=1}^{\eta} \sup_{z_i \in E_1} [\zeta_i z_i - c_i e^{z_i}], \text{ which is}$$

clearly finite if and only if $\zeta_i \geq 0, i=1,2,\dots,\eta$; in which case an

application of the differential calculus shows that $\omega(\zeta) = \sum_{i=1}^{\eta} \zeta_i \log (\zeta_i/c_i) - \sum_{i=1}^{\eta} \zeta_i$, with the understanding that $\zeta_i \log \zeta_i \triangleq 0$ when $\zeta_i = 0$. Consequently,

$$\omega(\zeta) = \sum_{i=1}^{\eta} \zeta_i \log (\zeta_i/c_i) - \sum_{i=1}^{\eta} \zeta_i \text{ and } \Omega = \{\zeta \in E_{\eta} \mid \zeta_i \geq 0, i=1,2,\dots,\eta\}.$$

Example 2. If

$$w(z) \triangleq \sum_{i=1}^{\eta-1} p_i^{-1} |z_i - b_i|^{p_i} + z_\eta - b_\eta \text{ and } W \triangleq E_\eta, \text{ where } p_i > 1, i = 1, 2, \dots, \eta - 1,$$

$$\text{then } \omega(\zeta) = \sup_{z \in E_\eta} [\langle \zeta, z \rangle - \sum_{i=1}^{\eta-1} p_i^{-1} |z_i - b_i|^{p_i} - (z_\eta - b_\eta)] =$$

$$\sum_{i=1}^{\eta-1} \sup_{z_i \in E_1} [\zeta_i z_i - p_i^{-1} |z_i - b_i|^{p_i}] + \sup_{z_\eta \in E_1} [\zeta_\eta z_\eta - (z_\eta - b_\eta)],$$

which is clearly finite if and only if $\zeta_\eta = 1$; in which case an application of the differential calculus shows that $\omega(\zeta) = \sum_{i=1}^{\eta-1} (q_i^{-1} |\zeta_i|^{q_i} + b_i \zeta_i) + b_\eta$, where q_i is determined from p_i by the equation $p_i^{-1} + q_i^{-1} = 1$. Consequently,

$$\omega(\zeta) = \sum_{i=1}^{\eta-1} (q_i^{-1} |\zeta_i|^{q_i} + b_i \zeta_i) + b_\eta \text{ and } \Omega = \{\zeta \in E_\eta \mid \zeta_\eta = 1\}.$$

Geometrical insight into the conjugate transformation can be obtained by considering the "subgradient" set for w at z , namely,

$$\partial w(z) \triangleq \{\zeta \in E_\eta \mid w(z) + \langle \zeta, z' - z \rangle \leq w(z') \text{ for each } z' \in W\}.$$

Subgradients are related to ∇w , but considerably different from ∇w , the more familiar gradient. The gradient provides a "tangent hyperplane" while a subgradient provides a "supporting hyperplane" (in that the defining inequality obviously states that the hyperplane with equation $w' = w(z) + \langle \zeta, z' - z \rangle$ intersects the "graph" of w at the point $(z, w(z))$ and lies entirely "on or below" it). It is, of course, clear that a subgradient may exist and not be unique even when the gradient does not exist. On the other hand it is also clear that a subgradient may not exist even when the gradient exists. There is, however, an important class of functions whose gradients are also subgradients -

the class of convex functions. In fact, the notions of gradient and subgradient coincide for the class of differentiable convex functions defined on open sets, a class that arises in many of the examples given in section 5.2.1.

To relate the conjugate transform to subgradients, observe that if $\zeta \in \partial w(z)$ then $\langle \zeta, z' \rangle - w(z') \leq \langle \zeta, z \rangle - w(z)$ for each $z' \in W$, which in turn clearly implies that $\zeta \in \Omega$ and that $\omega(\zeta) = - [w(z) + \langle \zeta, -z \rangle]$. Hence, $\omega(\zeta)$ is simply the negative of the intercept of the corresponding supporting hyperplane with the w' -axis. Consequently, the conjugate transform ω exists when w has at least one subgradient ζ , a condition that is known to be fulfilled when w is convex. Actually, the conjugate transform ω restricted (in the set-theoretic sense) to the domain $\bigcup_{z \in W} \partial w(z)$ is termed the "Legendre transform" of w and has been a major tool in the study of classical mechanics, thermodynamics, and differential equations (as described, for example, in [Courant, Hilbert, 1953]). Usually, the domain Ω of the conjugate transform ω consists of both $\bigcup_{z \in W} \partial w(z)$ and some of its limit points.

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

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

which is termed the "conjugate inequality" (or "Young's inequality") and which is clearly valid for every point $z \in W$ and every point $\zeta \in \Omega$ (as can be seen from the defining equation for $\omega(\zeta)$). Moreover, we have just shown that equality holds if

$$\zeta \in \partial w(z),$$

a condition that actually characterizes equality by virtue of another

elementary computation.

When it exists, the conjugate transform $w:\Omega$ is known to be both convex and "closed"; that is, its "epigraph" (which consists of all those points in $E_{\eta+1}$ that are "on or above" its graph) is both convex and (topologically) closed. Moreover, the conjugate transform of $w:\Omega$ is the "closed convex hull" $\overline{w:W}$ of $w:W$; and thus the conjugate transformation maps the family of all closed convex functions onto itself in one-to-one symmetric fashion. Consequently, the conjugate transformation is its own inverse on the family of all such functions; and given two such "conjugate functions" $w:W$ and $\omega:\Omega$, the relation $\zeta \in \partial w(z)$ and $z \in \partial \omega(\zeta)$ are equivalent and hence "solve" one another.

In geometric programming we must deal with both an arbitrary cone Z and its "dual"

$$Z \triangleq \{\zeta \in E_{\eta} \mid 0 \leq \langle z, \zeta \rangle \text{ for each } z \in Z\},$$

which is clearly itself a cone. Now, it is obvious that the conjugate transform of the zero function with domain Z is just the zero function with domain $-Z$. Consequently, the theory of the conjugate transformation implies that Z is convex and closed (a fact that can be established by more elementary considerations). Furthermore, if the cone Z is also convex and closed, the symmetry of the conjugate transformation readily implies that the dual of Z is just Z . If, in particular, Z is a vector space in E_{η} , this symmetry readily implies the better-known symmetry between orthogonal complementary subspaces Z and Z .

In convex analysis the "relative interior" ($\text{ri}W$) of a convex set $W \subseteq E_{\eta}$ is defined to be the "interior" of W "relative to" the "Euclidean topology" for the "affine hull" of W (i.e. the "smallest affine set [or linear manifold]"

containing W). The reason is that the (riW) defined in this way is not empty, even when the "interior" of W is empty.

This completes our prerequisites for the remaining subsections of this section.

5.3.1.3. Lagrangian saddle points. Let $h:D$ be the conjugate transform of the function $g:C$; that is,

$$D \triangleq \{y \in E_n \mid \sup_{x \in C} [\langle y, x \rangle - g(x)] < +\infty\}$$

and

$$h(y) \triangleq \sup_{x \in C} [\langle y, x \rangle - g(x)].$$

If $h:D$ exists (which is the case, for example, when $g:C$ is convex), then $h:D$ is a closed convex function that inherits any separability present in $g:C$. In fact, $h:D$ can be computed via the calculus in the convex case for each example given in section 5.2.1. Actually, the required computations for examples 1 and 2 have already been given in (the preceding) subsection 5.3.1.2; and the required computations for examples 3 through 5 can be found in the cited references.

The following definition is of fundamental importance.

DEFINITION. For a consistent problem \mathcal{Q} with a finite infimum ω , a P vector is any vector y^* with the two properties

$$y^* \in D$$

and

$$\omega = \inf_{x \in \mathcal{X}} L_{\theta}(x; y^*),$$

where the (geometric) Lagrangian

$$L_{\varrho}(x; y) \triangleq \langle x, y \rangle - h(y).$$

It should be noted that L_{ϱ} is generally as easy to compute as $h: \mathcal{D}$.

The following "saddle-point theorem" provides several characterizations of optimality via P vectors.

Theorem 2. Given that $\varrho: \mathcal{C}$ is convex and closed, let $x^* \in \mathcal{X}$ and let $y^* \in \mathcal{D}$. Then x^* is optimal for problem \mathcal{A} and y^* is a P vector for problem \mathcal{A} if and only if the ordered pair $(x^*; y^*)$ is a "saddle point" for the Lagrangian L_{ϱ} , that is,

$$\sup_{y \in \mathcal{D}} L_{\varrho}(x^*; y) = L_{\varrho}(x^*; y^*) = \inf_{x \in \mathcal{X}} L_{\varrho}(x; y^*);$$

in which case L_{ϱ} has the saddle point value

$$L_{\varrho}(x^*; y^*) = \varrho(x^*) = \varphi.$$

Moreover,

$$\sup_{y \in \mathcal{D}} L_{\varrho}(x^*; y) = L_{\varrho}(x^*; y^*)$$

if and only if x^* and y^* satisfy both the feasibility condition

$$x^* \in \mathcal{C}$$

and the subgradient condition

$$y^* \in \partial \varrho(x^*);$$

in which case

$$L_{\varrho}(x^*; y^*) = \varrho(x^*).$$

Furthermore,

$$L_{\varrho}(x^*; y^*) = \inf_{x \in \mathcal{X}} L_{\varrho}(x; y^*)$$

if and only if x^* and y^* satisfy both the feasibility condition

$$y^* \in \mathcal{D}$$

and the orthogonality condition

$$0 = \langle x^*, y^* \rangle;$$

in which case

$$L_{\vartheta}(x^*; y^*) = -h(y^*).$$

Since the second assertion of Theorem 2 gives certain conditions that are equivalent to the first saddle-point equation, and since the third assertion of Theorem 2 gives other conditions that are equivalent to the second saddle-point equation, Theorem 2 actually provides four different characterizations of all ordered pairs $(x^*; y^*)$ of optimal solutions x^* and P vectors y^* .

Of course, each of those four characterizations provides a characterization of all optimal solutions x^* in terms of a given P vector y^* , as well as a characterization of all P vectors y^* in terms of a given optimal solution x^* .

The symmetry of the preceding statement suggests that all P vectors may, in fact, constitute all optimal solutions to a closely related optimization problem. Actually, the appropriate optimization problem can be motivated by the following inequalities

$$\inf_{x \in \mathcal{X}} L_{\vartheta}(x; y^*) \leq \sup_{y \in \mathcal{D}} [\inf_{x \in \mathcal{X}} L_{\vartheta}(x; y)] \leq \inf_{x \in \mathcal{X}} [\sup_{y \in \mathcal{D}} L_{\vartheta}(x; y)] \leq \sup_{y \in \mathcal{D}} L_{\vartheta}(x^*; y),$$

which are valid for each $y^* \in \mathcal{D}$ and each $x^* \in \mathcal{X}$, by virtue of the single fact that x and y reside in independent sets \mathcal{X} and \mathcal{D} respectively (i.e. the vector $(x; y)$ resides in a cartesian product $\mathcal{X} \times \mathcal{D}$). In particular, note that each of these inequalities must be an equality when $(x^*; y^*)$ is a saddle-point for L_{ϑ} ; in which case x^* is obviously an optimal solution to the minimization problem

$$\inf_{y \in \mathcal{D}} [\sup_{x \in \mathcal{X}} L_{\vartheta}(x; y)],$$

and y^* is obviously an optimal solution to the maximization problem

$$\sup_{y \in \mathcal{D}} [\inf_{x \in \mathcal{X}} L_g(x; y)].$$

Now, when g is convex and closed, the definition of L_g and the symmetry of the conjugate transformation clearly imply that the preceding minimization problem is essentially problem \mathcal{A} . Consequently, it is not unnatural to consider the preceding maximization problem simultaneously with problem \mathcal{A} and term it the "geometric dual problem" \mathcal{B} .

PROBLEM \mathcal{B} . Using the feasible solution set

$$\mathcal{J} \triangleq \{y \in \mathcal{D} \mid \inf_{x \in \mathcal{X}} L_g(x; y) \text{ is finite}\}$$

and the objective function

$$\mathcal{K}(y) \triangleq \inf_{x \in \mathcal{X}} L_g(x; y),$$

calculate both the problem supremum

$$\Psi \triangleq \sup_{y \in \mathcal{J}} \mathcal{K}(y)$$

and the optimal solution set

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

Even though problem \mathcal{B} is essentially a "maximin problem" - a type of problem that tends to be relatively difficult to analyse - the minimization problems that must be solved to obtain the objective function $\mathcal{K}:\mathcal{J}$ have trivial solutions. In particular, the definition of L_g and the hypothesis that \mathcal{X} is a cone clearly imply that $\inf_{x \in \mathcal{X}} L_g(x; y)$ is finite if and only if $y \in \mathcal{Y}$, in which case $\inf_{x \in \mathcal{X}} L_g(x; y) = -h(y)$. Consequently, $\mathcal{J} = \mathcal{Y} \cap \mathcal{D}$ and $\mathcal{K}(y) = -h(y)$, so problem \mathcal{B} can actually be rephrased in the following more direct way:

Using the feasible solution set

$$\mathcal{F} = \mathcal{Y} \cap \mathcal{D},$$

calculate both the problem infimum

$$\psi \triangleq \inf_{y \in \mathcal{F}} h(y) = -\Psi$$

and the optimal solution set

$$\mathcal{F}^* = \{y \in \mathcal{F} \mid h(y) = \psi\}.$$

When phrased in this way, problem \mathcal{B} closely resembles problem \mathcal{A} , and is in fact a geometric programming problem. Of course, the geometric dual problem \mathcal{B} can actually be defined in this way - an approach that is exploited in the following subsection. Nevertheless, the preceding derivation serves as an important link between Lagrangians and duality.

5.3.1.4. Duality. Let $h:\mathcal{D}$ be the conjugate transform of the function $g:\mathcal{C}$; that is,

$$\mathcal{D} \triangleq \{y \in \mathbb{E}_n \mid \sup_{x \in \mathcal{C}} [\langle y, x \rangle - g(x)] < +\infty\}$$

and

$$h(y) \triangleq \sup_{x \in \mathcal{C}} [\langle y, x \rangle - g(x)].$$

If $h:\mathcal{D}$ exists (which is the case, for example, when $g:\mathcal{C}$ is convex), then $h:\mathcal{D}$ is a closed convex function that inherits any separability present in $g:\mathcal{C}$. In fact, $h:\mathcal{D}$ can be computed via the calculus in the convex case for each example given in section 5.2.1. Actually, the required computations for examples 1 and 2 have already been given in subsection 5.3.1.2; and the required computations for examples 3 through 5 can be found in the cited references.

Now, consider the following geometric programming problem \mathcal{B} .

PROBLEM \mathcal{B} . Using the feasible solution set

$$\mathcal{J} \triangleq \mathcal{Y} \cap \mathcal{D},$$

calculate both the problem infimum

$$\psi \triangleq \inf_{y \in \mathcal{J}} h(y)$$

and the optimal solution set

$$\mathcal{J}^* \triangleq \{y \in \mathcal{J} \mid h(y) = \psi\}.$$

It should be noted that problem \mathcal{B} is generally as easy to compute as $h:\mathcal{D}$ and \mathcal{Y} . Moreover, problem \mathcal{B} is always a convex programming problem, because both $h:\mathcal{D}$ and \mathcal{Y} are always convex and closed (even when $g:\mathcal{C}$ and \mathcal{X} are not convex and closed).

Problems \mathcal{A} and \mathcal{B} are termed geometric dual problems. When both $g:\mathcal{C}$ and \mathcal{X} are convex and closed, this duality is clearly symmetric, in that problem \mathcal{A} can then be constructed from problem \mathcal{B} in the same way that problem \mathcal{B} has just been constructed from problem \mathcal{A} . This symmetry induces a symmetry on the theory that relates \mathcal{A} to \mathcal{B} , in that each statement about \mathcal{A} and \mathcal{B} automatically produces an equally valid "dual statement" about \mathcal{B} and \mathcal{A} . To be concise, each dual statement will be left to the reader's imagination.

It is worth mentioning that there are cases in which problems \mathcal{A} and \mathcal{B} have additional interesting symmetries. In particular, problem \mathcal{B} turns out to be a "reversed-time" discrete optimal control problem when problem \mathcal{A} is taken to be a (forward-time) discrete optimal control problem (i.e. example 4 in section 5.2.1) whose linear dynamics are such that each matrix B_i is identical to a nonsingular matrix B that commutes with each matrix A_i for $i = 1, 2, \dots, p$.

Unlike the usual min-max formulations of duality in mathematical programming, both problem \mathcal{A} and its geometric dual problem \mathcal{B} are minimization

problems. The relative simplicity of this min-min formulation will soon become clear, but the reader who is accustomed to the usual min-max formulation must bear in mind that a given duality theorem will generally have slightly different statements depending on the formulation in use. In particular, a theorem that asserts the equality of the min and max in the usual formulation will assert that the sum of the mins is zero (i.e. $\varphi + \psi = 0$) in the present formulation.

The following definition is almost as important as the definition of the dual problems \mathcal{A} and \mathcal{B} .

DEFINITION. The extremality conditions (for unconstrained geometric programming) are:

- (I) $x \in \mathcal{X}$ and $y \in \mathcal{Y}$
- (II) $0 = \langle x, y \rangle$
- (III) $y \in \partial \varphi(x)$.

Extremality conditions (I) are simply the "cone conditions" for problems \mathcal{A} and \mathcal{B} respectively. Extremality condition (II) is termed the "orthogonality condition", and extremality condition (III) is termed the "subgradient condition".

If the cone \mathcal{X} is actually a vector space (which is the case for each of the examples given in section 5.2.1, except example 5), then $\mathcal{Y} = \mathcal{X}^\perp$ and hence the orthogonality condition (II) is redundant and can be deleted from the preceding definition (and everywhere that definition is used).

The following "duality theorem" is the basis for many others to come.

Theorem 3. If x and y are feasible solutions to problems \mathcal{A} and \mathcal{B} respectively (in which case the extremality conditions (I) are satisfied), then

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

with equality holding if and only if the extremality conditions (II) and (III) are satisfied; in which case x and y are optimal solutions to problems \mathcal{A} and \mathcal{B} respectively.

In essence, the proof of this key theorem consists only of combining the defining inequality $0 \leq \langle x, y \rangle$ for \mathcal{Y} with the conjugate inequality $\langle x, y \rangle \leq g(x) + h(y)$ for $h: \mathcal{B}$.

The following important corollary is an immediate consequence of Theorem 3.

Corollary 3A. If the dual problems \mathcal{A} and \mathcal{B} are both consistent, then

(i) the infimum φ for problem \mathcal{A} is finite, and

$$0 \leq \varphi + h(y) \text{ for each } y \in \mathcal{Y},$$

(ii) the infimum ψ for problem \mathcal{B} is finite, and

$$0 \leq \varphi + \psi.$$

The strictness of the inequality in conclusion (ii) plays a crucial role in almost all that follows.

DEFINITION. Consistent dual problems \mathcal{A} and \mathcal{B} for which

$$0 < \varphi + \psi$$

have a duality gap of $\varphi + \psi$.

It is well known that duality gaps do not occur in finite linear programming, but they do occasionally occur in infinite linear programming

where this phenomenon was first encountered by Duffin [1956] and Kretchmer [1961]. Although duality gaps occur very frequently in the present (generally

nonconvex) formulation of geometric programming, we shall eventually see that they can occur only very rarely in the convex case, in that they can then be excluded by very weak conditions on the geometric dual problems \mathcal{A} and \mathcal{B} . Yet, they do occur in the convex case, and examples (due originally to J.J. Stoer) can be found in Appendix C of [Peterson, 1970].

Geometric programming problems \mathcal{A} that are convex are usually much more amenable to study than those that are nonconvex, mainly because of the relative lack of duality gaps in the convex case. Duality gaps are undesirable from a theoretical point of view because we shall see that relatively little can be said about the corresponding geometric dual problems. They are also undesirable from a computational point of view because they usually destroy the possibility of using the inequality $0 \leq g(x) + h(y)$ to provide an algorithmic stopping criterion.

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

$$g(x^\dagger) + h(y^\dagger) \leq 2\epsilon.$$

Because conclusion (i) to Corollary 3A along with the definition of φ shows that $-h(y^\dagger) \leq \varphi \leq g(x^\dagger)$, we conclude from the preceding tolerance inequality that

$$\left| \varphi - \frac{g(x^\dagger) - h(y^\dagger)}{2} \right| \leq \epsilon.$$

Hence, φ can be approximated by $[g(x^\dagger) - h(y^\dagger)]/2$ with an error no greater than $\pm \epsilon$. Moreover, duality (i.e. symmetry) implies that ψ can be approximated by $[h(y^\dagger) - g(x^\dagger)]/2$ with an error no greater than $\pm \epsilon$.

Note though that problems \mathcal{A} and \mathcal{B} have a duality gap if and only if there is a positive tolerance ϵ so small that $2\epsilon < \varphi + \psi$. Because the definitions for φ and ψ imply that $\varphi + \psi \leq g(x) + h(y)$ for feasible solutions x and y ,

we infer that when ϵ satisfies the preceding inequality there are no feasible solutions x^\dagger and y^\dagger for which $g(x^\dagger) + h(y^\dagger) \leq 2\epsilon$; in which event the algorithms being used are never terminated.

The following corollary provides a useful characterization of dual optimal solutions x^* and y^* in terms of the extremality conditions.

Corollary 3B. If the extremality conditions have a solution x' and y' , then

- (i) $x' \in \mathcal{X}^*$ and $y' \in \mathcal{Y}^*$
- (ii) $\mathcal{Y}^* = \{y \in \mathcal{Y} \cap \partial g(x') \mid 0 = \langle x', y \rangle\}$
- (iii) $0 = \varphi + \psi$.

On the other hand, if the dual problems \mathcal{A} and \mathcal{B} are both consistent and if $0 = \varphi + \psi$, then $x \in \mathcal{X}^*$ and $y \in \mathcal{Y}^*$ if and only if x and y satisfy the extremality conditions.

The proof of this corollary is an immediate consequence of Theorem 3 and the conjugate transform relation $\partial g(x) \subseteq \mathcal{B}$.

The first part of Corollary 3B shows that problems \mathcal{A} and \mathcal{B} can be viewed as "variational principles" for finding solutions to the extremality conditions. Actually, in many contexts, the extremality conditions are, in one form or another, the natural objects of study (rather than either problem \mathcal{A} or problem \mathcal{B}). For the highway network problem described in example 5 of section 5.2.1, the extremality conditions (I) through (III) are equivalent to the conditions that define "Wardrop equilibrium". For the electric and hydraulic network problems alluded to in the same example: the cone conditions (I) are simply the "Kirchoff current and potential conservation laws" respectively, the orthogonality condition (II) is redundant because \mathcal{X} and \mathcal{Y} are orthogonal complementary vector spaces, and the sub-

gradient condition (III) is just "Ohm's law". Moreover, in the context of electric and hydraulic networks, problems \mathcal{A} and \mathcal{B} are frequently termed the "Maxwell-Duffin complementary variational principles". Needless to say, Corollary 3B shows that the lack of a duality gap is fundamental in all such contexts.

It is worth noting that if $g:\mathcal{C}$ is convex and closed, then the symmetry of the conjugate transformation implies that the subgradient condition (III) can be replaced by the equivalent subgradient condition

$$(IIIa) \quad x \in \partial h(y)$$

without changing the validity of Corollary 3B.

In that case Corollary 3B and its (unstated) dual are of direct use when $0 = \varpi + \psi$ and both \mathcal{S}^* and \mathcal{T}^* are known to be nonempty; because they then provide a method for calculating all optimal solutions from the knowledge of only a single optimal solution. For example, if x^* in \mathcal{S}^* is a known optimal solution to problem \mathcal{A} , then

$$\mathcal{T}^* = \{y \in \mathcal{U} \cap \partial g(x^*) \mid 0 = \langle x^*, y \rangle\};$$

and for each $y^* \in \mathcal{T}^*$, the set

$$\mathcal{S}^* = \{x \in \mathcal{X} \cap \partial h(y^*) \mid 0 = \langle x, y^* \rangle\}.$$

The definition of the Lagrangian L_g (in subsection 5.3.1.3) and the fact that \mathcal{X} is a cone readily imply that the cone condition $y \in \mathcal{Y}$ and the orthogonality condition $0 = \langle x, y \rangle$ can both be replaced by the single equivalent condition $L_g(x; y) = \inf_{x' \in \mathcal{X}_g} L_g(x'; y)$. Moreover, if $g:\mathcal{C}$ is convex

and closed, conjugate transform theory readily implies that the subgradient condition $y \in \partial g(x)$ can be replaced by the equivalent condition

$$\sup_{y' \in \mathcal{D}} L_g(x; y') = L_g(x; y).$$

Consequently, the saddle-point condition discussed in subsection 5.3.1.3 is equivalent to the extremality conditions when

$g:\mathcal{C}$ is convex and closed. Nevertheless, it seems that the extremality conditions given in the definition are the most convenient to work with.

The following theorem provides an important tie between dual problem \mathcal{B} and the P vectors defined in subsection 5.3.1.3.

Theorem 4. Given that problem \mathcal{A} is consistent with a finite infimum φ ,

- (1) if problem \mathcal{A} has a P vector, then problem \mathcal{B} is consistent and $0 = \varphi + \psi$,
- (2) if problem \mathcal{B} is consistent and $0 = \varphi + \psi$, then

$$\{y^* \mid y^* \text{ is a P vector for problem } \mathcal{A}\} = \mathcal{J}^*.$$

The proof of this theorem is not difficult but will of course be omitted.

An important consequence of Theorem 4 is that, when they exist, all P vectors for problem \mathcal{A} can be obtained simply by computing the dual optimal solution set \mathcal{J}^* . However, there are cases in which the vectors in \mathcal{J}^* are not P vectors for problem \mathcal{A} ; though such cases can occur only when $0 < \varphi + \psi$, in which event Theorem 4 implies that there can be no P vectors for problem \mathcal{A} .

The absence of a duality gap (i.e. the assumption that $0 = \varphi + \psi$) is crucial to the preceding computational and theoretical techniques, as well as others to come. Although there are numerous conditions that guarantee the absence of a duality gap, the most useful and widely used ones can be viewed as special manifestations of the hypotheses in the following theorem. In addition to guaranteeing the absence of a duality gap, this (geometric programming) version of "Fenchel's theorem" also serves as a very important existence theorem.

Theorem 5. Suppose that both $g:\mathcal{C}$ and \mathcal{X} are convex and closed. If the dual

problem β has a feasible solution $y^0 \in (\text{ri}\mathcal{Y}) \cap (\text{ri}\mathcal{B})$, and if problem β has a finite infimum ψ , then $0 = \varphi + \psi$ and $\mathcal{L}^* \neq \emptyset$.

A proof of this theorem that is quite different from Fenchel's original proof (as given in [Fenchel, 1951] or [Rockafellar, 1970]) can be found in [Peterson, 1970].

There are several facts about relative interiors that help in the application of Theorem 5. For example, if the cone \mathcal{Y} is actually a vector space, then $(\text{ri}\mathcal{Y}) = \mathcal{Y}$ and hence the hypothesis $y^0 \in (\text{ri}\mathcal{Y}) \cap (\text{ri}\mathcal{B})$ is implied by the hypothesis $y^0 \in \mathcal{Y} \cap (\text{ri}\mathcal{B})$. Also, if the set \mathcal{B} turns out to be a vector space, then $(\text{ri}\mathcal{B}) = \mathcal{B}$ and hence the hypothesis $y^0 \in (\text{ri}\mathcal{Y}) \cap (\text{ri}\mathcal{B})$ is implied by the hypothesis $y^0 \in (\text{ri}\mathcal{Y}) \cap \mathcal{B}$; which in turn is always satisfiable when $\mathcal{B} = E_n$.

Theorem 5 and the preceding facts are the key ingredients needed to show that there are no duality gaps for many of the examples given in section 5.2.1, such as:

- (1) posynomial programming problems (example 1) that are "canonical" (as defined in [Duffin, Peterson, 1966] or [Duffin, Peterson, Zener, 1967]),
- (2) convex ℓ_p programming problems (example 2) that are "canonical" (as defined in [Peterson, Ecker, 1970]),
- (3) optimal location problems (example 3) whose costs $d_i(z, b^i) = \|z - b^i\|_i$ for some "norm" $\|\cdot\|_i$ (as first shown in [Peterson, Wendell, 1975]),
- (4) discrete optimal control problems (example 4) that are "canonical" (as defined in [Dinkel, Peterson, 1975]).
- (5) highway network equilibrium problems (example 5) whose roadway arc travel times are monotone nondecreasing and unbounded from above as functions of the corresponding arc total traffic flows (as first shown in [Hall, Peterson, 1975]),

(6) electric and hydraulic network equilibrium problems (also example 5) that contain only "current sources", "potential sources", and linear or nonlinear "monotone resistors" (as first shown essentially in [Duffin, 1947], [Minty, 1960], and [Rockafellar, 1967b]).

It is worth mentioning here that, when both $g:\mathcal{C}$ and \mathcal{X} are convex and closed, the absence of a duality gap can actually be characterized in terms of the changes induced in the problem infimum φ by small changes in certain problem input parameters - a characterization that is given in the following subsection.

5.3.1.5. Parametric programming and post-optimality analysis. For both practical and theoretical reasons, problem \mathcal{A} should not be studied entirely in isolation. It should also be embedded in a parameterized family \mathcal{F} of closely related geometric programming problems $\mathcal{A}(u)$ that are generated by simply translating (the domain \mathcal{C} of) g through all possible displacements - $u \in E_n$, while keeping \mathcal{X} fixed. (For gaining insight, we recommend making a sketch of a typical case in which n is 2 and \mathcal{X} is a one-dimensional vector space.) Problem \mathcal{A} then appears in the parameterized family \mathcal{F} as problem $\mathcal{A}(0)$ and is studied in relation to all other geometric programming problems $\mathcal{A}(u)$, with special attention given to those problems $\mathcal{A}(u)$ in \mathcal{F} that are close to $\mathcal{A}(0)$ in the sense that (the "norm" of) u is small.

The parameterized family \mathcal{F} of all problems $\mathcal{A}(u)$ (for fixed $g:\mathcal{C}$ and \mathcal{X}) is termed a geometric programming family. For purposes of easy reference and mathematical precision, problem $\mathcal{A}(u)$ is now given the following formal definition, which should be compared with the formal definition of problem \mathcal{A} at the beginning of section 5.2.1.

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

$$\mathcal{A}(u) \triangleq \mathcal{X} \cap (\mathcal{C} - u),$$

calculate both the problem infimum

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

and the optimal solution set

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

Note that (in a rather general set-theoretic sense) the symbols \mathcal{A} , \mathcal{A} , φ and \mathcal{A}^* now represent functions of u , though they originally represented only the particular functional values $\mathcal{A}(0)$, $\mathcal{A}(0)$, $\varphi(0)$ and $\mathcal{A}^*(0)$ respectively. Needless to say, the reader must keep this notational discrepancy in mind when comparing subsequent developments with previous developments.

For a given u , problem $\mathcal{A}(u)$ is either consistent or inconsistent, depending on whether the feasible solution set $\mathcal{A}(u)$ is nonempty or empty. It is, of course, obvious that the parameterized family \mathcal{F} contains infinitely many consistent problems $\mathcal{A}(u)$. The domain of the infimum function φ is taken to be the corresponding nonempty set \mathcal{U} of all those vectors u for which $\mathcal{A}(u)$ is consistent. Thus, the range of φ may contain the point $-\infty$; but if $\varphi(u) = -\infty$ then the **optimal solution set** $\mathcal{A}^*(u)$ is clearly empty.

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

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

the feasible perturbation set for problem $\mathcal{A}(0)$ (relative to the family \mathcal{F}).

The functions φ and \mathcal{A}^* usually show the dependence of optimality on actual external influences and hence are of prime interest in "cost-benefit analysis" and other such subjects. In fact, for the examples given in section 5.2.1, it is easy to see that the perturbation vector u :

(1) alters in example 1 the (log of the absolute value of the) signomial coefficients c_i (which are generally determined by such external influences as design requirements, performance requirements, materiel costs, and so forth),

(2) alters in the linear regression analysis case of example 2 the vector (b_1, \dots, b_m) being optimally approximated,

(3) alters in example 3 the fixed facility locations b^i , provided that each cost $d_i(z, b^i) = \|z - b^i\|_i$ for some "norm" $\|\cdot\|_i$,

(4) translates in example 4 the "state sets" R_i and the "decision sets" D_i ,

(5) alters in example 5 the total input flows d_i , as well as other network parameters.

The properties of the parameterized family \mathcal{J} brought out in the following theorem are of fundamental theoretical significance and also have rather obvious applications in parametric programming.

Theorem 6. The feasible perturbation set \mathcal{U} is given by the formula

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

Moreover, if both \mathcal{C} and \mathcal{X} are convex, then so is \mathcal{U} , and the point-to-set function ρ is "concave" on \mathcal{U} in that

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

for each "convex combination" $\delta_1 u^1 + \delta_2 u^2$ of arbitrary points $u^1, u^2 \in \mathcal{U}$.

Furthermore, if both $g: \mathcal{C}$ and \mathcal{X} are convex, then so is problem $\sigma(u)$

for each $u \in \mathcal{U}$, and the infimum function φ is either finite and convex on \mathcal{U} or $\varphi(u) \equiv -\infty$ for each $u \in (\text{ri}\mathcal{U})$.

A proof for a slightly more limited version of this theorem is given in

[Peterson, 1970].

It should be mentioned that there are instances in which both $\varphi: \mathcal{C}$ and \mathcal{X} are convex (as well as closed) and for which $\varphi(u)$ is finite for at least one $u \in (\text{rb}\mathcal{U})$ (the "relative boundary" of \mathcal{U}) even though $\varphi(u) \equiv -\infty$ for each $u \in (\text{ri}\mathcal{U})$. An example (due originally to J.J. Stoer) can be found in appendix C of [Peterson, 1970].

In cases for which \mathcal{X} is actually a vector space, the following theorem reduces a study of the parameterized family \mathcal{F} to a study of only those problems $\mathcal{A}(u)$ in \mathcal{F} for which $u \in \mathcal{U} = \mathcal{X}^\perp$. In such cases, it is convenient to adopt the notation $u_{\mathcal{X}}$ and $u_{\mathcal{Y}}$ for the "orthogonal projection" of an arbitrary vector $u \in E_n$ onto the orthogonal complementary subspaces \mathcal{X} and \mathcal{Y} respectively.

Theorem 7. Suppose that \mathcal{X} and \mathcal{U} are orthogonal complementary subspaces of E_n . Then, for each vector $u \in E_n$, either the feasible solution sets $\mathcal{S}(u)$ and $\mathcal{S}(u_{\mathcal{Y}})$ are both empty, or both are nonempty, with the latter being the case if and only if $u \in \mathcal{U}$, in which case

$$\mathcal{S}(u) = \mathcal{S}(u_{\mathcal{Y}}) - u_{\mathcal{X}}$$

and

$$\varphi(u) = \varphi(u_{\mathcal{Y}}).$$

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

$$\mathcal{S}^*(u) = \mathcal{S}^*(u_{\mathcal{Y}}) - u_{\mathcal{X}}.$$

The proof of this theorem is not difficult and can be found in [Peterson, 1970].

In addition to its rather obvious applications in parametric programming the preceding reduction theorem and its (unstated) dual can be

used to relate the present geometric programming formulation of duality to both the original Fenchel formulation of duality [Fenchel, 1951] and the more recent Rockafellar formulations of duality [Rockafellar, 1967a, 1968, 1970]. All such relations can be found in [Peterson, 1970].

The following (geometric programming) version of a theorem due originally to Rockafellar [1967a, 1968, 1970] provides a direct link between the infimum function φ and the dual problem \mathcal{B} . This link serves as the key to deriving very important properties of φ via conjugate transform theory.

Theorem 8. The infimum function φ is finite everywhere on its domain \mathcal{U} and possesses a conjugate transform if and only if the dual problem \mathcal{B} is consistent, in which case the dual objective function $h:\mathcal{F}$ is the conjugate transform of $\varphi:\mathcal{U}$.

Although this theorem has a relatively simple and direct proof (due originally to Rockafellar), it can also be viewed as an immediate corollary to other theorems that produce micro-economic interpretations of the dual problem \mathcal{B} . In fact, a detailed analysis of the situation (given in [Peterson, 1970]) shows that a rather interesting class of micro-economic problems can be solved explicitly in terms of problem \mathcal{B} .

The following corollary plays a crucial role in most applications of Theorem 8.

Corollary 8A. If the dual problem \mathcal{B} is consistent, then the conjugate transform of its objective function $h:\mathcal{F}$ is the closed convex hull $\overline{\varphi:\mathcal{U}}$ of the infimum function $\varphi:\mathcal{U}$ (which is, of course, identical to $\varphi:\mathcal{U}$ when $\varphi:\mathcal{U}$ happens to be both convex and closed).

This corollary is an immediate consequence of Theorem 8 and conjugate

transform theory.

The preceding theorem and its corollary provide a method for constructing, without the use of numerical optimization techniques, the closed convex hull $\bar{\varphi}:\bar{\mathcal{U}}$ of $\varphi:\mathcal{U}$ (which, by virtue of Theorem 6, is essentially the desired infimum function $\varphi:\mathcal{U}$ when both \mathcal{C} and \mathcal{X} are convex). In particular, if the dual feasible solution set \mathcal{J} is not empty (which, as indicated by Theorem 6, is the only really interesting nontrivial case), it can of course be covered with a "mesh"

$$\mathcal{M} \triangleq \{y^1, y^2, \dots, y^s\} \subseteq \mathcal{J} \triangleq \mathcal{Y} \cap \mathcal{D}.$$

From Corollary 8A and conjugate transform theory $\bar{\varphi}$ is clearly bounded from below on $\bar{\mathcal{U}}$ by the (polyhedral) approximating function $\bar{\varphi}_b$ whose functional values

$$\bar{\varphi}_b(u) \triangleq \max_{i=1,2,\dots,s} [\langle y^i, u \rangle - h(y^i)] \text{ for each } u \in \bar{\mathcal{U}}.$$

Moreover, the conjugate inequality (for $\bar{\varphi}:\bar{\mathcal{U}}$ and $h:\mathcal{J}$) can be used to show that

$$\bar{\varphi}(u) = \langle y^i, u \rangle - h(y^i) \text{ for each } u \in \mathcal{H}(y^i) - \{\chi \in \mathcal{X} \mid \langle \chi, y \rangle = 0\}, i = 1, 2, \dots, s;$$

and the convexity of $\bar{\varphi}$ then implies that $\bar{\varphi}$ is bounded from above by "affine interpolations" between such functional values. Furthermore, it is a consequence of conjugate transform theory that these lower and upper approximations can be made with arbitrary accuracy simply by choosing the mesh \mathcal{M} to be sufficiently "dense" in \mathcal{J} . To be practical though, this method requires an explicit construction of the dual objective function $h:\mathcal{J}$, a construction that is rather easy for virtually all of the important (convex) examples given in section 5.2.1.

The preceding corollary also helps to motivate the following definition.

DEFINITION. Problem $\mathcal{A}(u)$ is said to be quasi-consistent if the closed convex

hull $\bar{\varphi}:\bar{\mathcal{U}}$ of the infimum function $\varphi:\mathcal{U}$ exists and if $u \in \bar{\mathcal{U}}$, in which case $\bar{\varphi}(u)$ is termed the quasi-infimum for problem $\mathcal{A}(u)$.

Since $\mathcal{U} \subseteq \bar{\mathcal{U}}$, each consistent problem $\mathcal{A}(u)$ is quasi-consistent, but not conversely.

The following theorem leads to a complete explanation of duality gaps, and is also at the heart of many post-optimal "sensitivity analyses".

Theorem 9. Suppose that the dual problem \mathcal{B} is consistent. Then, its infimum ψ is finite if and only if problem $\mathcal{A}(0)$ is quasi-consistent, in which case

$$0 = \bar{\varphi}(0) + \psi \text{ and } \partial\bar{\varphi}(0) = \mathcal{J}^{\star}.$$

The proof of this theorem is a rather direct consequence of Corollary 8A and conjugate transform theory. The details for somewhat limited versions of this theorem can be found in either [Peterson, 1970] or [Rockafellar, 1970].

The following corollary identifies duality gaps as simply the difference between $\varphi(0)$ and $\bar{\varphi}(0)$.

Corollary 9A. Suppose that the dual problems $\mathcal{A}(0)$ and \mathcal{B} are both consistent. Then, problem $\mathcal{A}(0)$ is quasi-consistent and

$$\varphi(0) - \bar{\varphi}(0) = \varphi(0) + \psi.$$

The proof of this corollary comes from Corollary 3A and the addition of $\varphi(0) - \bar{\varphi}(0)$ to both sides of the equation $0 = \bar{\varphi}(0) + \psi$ given in Theorem 9.

The preceding corollary helps to motivate the following definition.

DEFINITION. A consistent problem $\mathcal{A}(u)$ for which $\varphi(u) = \bar{\varphi}(u)$ is said to be normal.

The following corollary formalizes the equivalence of normality and the lack of a duality gap, while showing that either condition guarantees the validity of the equation on which many sensitivity analyses are based.

Corollary 9B. Suppose that the dual problems $\mathcal{A}(0)$ and \mathcal{B} are both consistent. Then, problem $\mathcal{A}(0)$ is normal if and only if problems $\mathcal{A}(0)$ and \mathcal{B} have no duality gap, in which case

$$\partial_{\text{co}}(0) = \mathcal{J}^*.$$

The proof of this corollary uses both Theorem 9 and Corollary 9A along with the fact that $\partial_{\text{co}}(0) = \partial_{\text{co}}(0)$ when $\varphi(0) = \bar{\varphi}(0)$.

An additional prerequisite for sensitivity analyses is provided by the following definition.

DEFINITION. A consistent problem $\mathcal{A}(u)$ with a finite infimum $\varphi(u)$ is termed stably set (relative to the family \mathcal{F}) when the (one-sided) "directional derivative"

$$D_d \varphi(u) \triangleq \lim_{s \rightarrow 0^+} \frac{\varphi(u+sd) - \varphi(u)}{s}$$

exists and is finite for each feasible direction d (i.e. each direction d such that $u+sd \in \mathcal{U}$ for sufficiently small $s > 0$).

The following theorem ties stability directly to the existence of certain subgradients, and shows how each guarantees normality.

Theorem 10. Let both $\varphi: \mathcal{C}$ and \mathcal{X} be convex, and suppose that problem $\mathcal{A}(0)$ is consistent and has a finite infimum $\varphi(0)$. Then, problem $\mathcal{A}(0)$ is stably set if and only if $\partial_{\text{co}}(0)$ is not empty, in which case problem $\mathcal{A}(0)$ is normal.

This theorem is a simple consequence of Theorem 6 and the differentiability properties of convex functions (as described, for example, in [Rockafellar, 1970]).

Given a consistent problem $\mathcal{A}(0)$ with a known finite infimum $\varphi(0)$, a sensitivity analysis consists of estimating other infima $\varphi(u)$ for very small u . First-order estimates can, of course, be based on the directional derivatives $D_u \varphi(0)$ when problem $\mathcal{A}(0)$ is stably set. In that case, the defining equation for $D_u \varphi(0)$ provides the usual estimation formula

$$\varphi(u) \approx \varphi(0) + D_u \varphi(0),$$

whose use requires a computation of $D_u \varphi(0)$. Toward that end, Fenchel [1951] and, more recently, Rockafellar [1970] have appropriately extended the well-known formula $D_u \varphi(0) = \langle u, \nabla \varphi(0) \rangle$ by showing that $D_u \varphi(0) = \max_{y \in \partial \varphi(0)} \langle u, y \rangle$ when φ is convex on \mathcal{U} - which is indeed the case when both $g: \mathcal{C}$ and \mathcal{X} are convex, by virtue of Theorem 6. Moreover, Corollary 9B shows that the preceding formula can be rewritten as

$$D_u \varphi(0) = \max_{y \in \mathcal{J}^*} \langle u, y \rangle$$

when the dual problem \mathcal{B} is consistent and there is no duality gap - which is almost always the case, as indicated by Theorem 6, Theorem 8, and (Fenchel's) Theorem 5. Consequently, it is of interest to know \mathcal{J}^* in addition to $\varphi(0)$ and $\varphi^*(0)$, so that the preceding displayed formulas can be used to estimate $\varphi(u)$ for a very small u . But \mathcal{J}^* can usually be calculated from an arbitrary $x^* \in \mathcal{S}^*(0)$ by employing the extremality conditions, as explained after Corollary 3B.

To further characterize duality gaps, we also embed the dual problem \mathcal{B} in a parameterized family \mathcal{L} of closely related geometric programming problems $\mathcal{B}(v)$ that are generated by simply translating (the domain \mathcal{D} of) h

through all possible displacements - $v \in E_n$, while keeping \mathcal{V} fixed. Problem \mathcal{B} then appears in the parameterized family \mathcal{L} as problem $\mathcal{B}(0)$ and is studied in relation to all other geometric programming problems $\mathcal{B}(v)$, with special attention given to those problems $\mathcal{B}(v)$ in \mathcal{L} that are close to $\mathcal{B}(0)$ in the sense that (the "norm" of) v is small.

Naturally, the symbols \mathcal{B} , \mathcal{F} , ψ and \mathcal{F}^* now represent functions of v , though they originally represented only the particular functional values $\mathcal{B}(0)$, $\mathcal{F}(0)$, $\psi(0)$ and $\mathcal{F}^*(0)$ respectively. Needless to say, the reader must keep this notational discrepancy in mind when comparing subsequent developments with previous developments.

Since the main results having to do with the dual geometric programming family \mathcal{L} are essentially dual to those already given for the family \mathcal{F} , we shall feel free to use them without further discussion.

The following important theorem involves the dual families \mathcal{F} and \mathcal{L} in a reflexive (i.e. self-dual) way.

Theorem 11. Let both \mathcal{C} and \mathcal{X} be convex and closed, and suppose that the dual problems $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are both consistent. Then, the following three conditions are equivalent:

- (i) problem $\mathcal{A}(0)$ is normal,
- (ii) problem $\mathcal{B}(0)$ is normal,
- (iii) problems $\mathcal{A}(0)$ and $\mathcal{B}(0)$ do not have a duality gap.

Moreover, if any of these three conditions are satisfied, then

$$\partial \mathcal{B}(0) = \mathcal{F}^*$$

and

$$\partial \psi(0) = \mathcal{L}^*.$$

This theorem can be proved simply by a repeated application of Corollary 9B and its (unstated) dual.

There are various degrees of consistency, ranging all the way from (the very weak) quasi-consistency through (the intermediate) consistency to "strong consistency".

DEFINITION. Problem $\beta(v)$ is said to be strongly consistent if $v \in (\text{ri}\mathcal{V})$.

Since $(\text{ri}\mathcal{V}) \subsetneq \mathcal{V}$, each strongly consistent problem $\beta(v)$ is consistent, but not conversely.

The following theorem indicates the importance of strong consistency.

Theorem 12. Each strongly consistent dual problem $\beta(0)$ with a finite infimum $\psi(0)$ is stably set (and hence normal).

This theorem is an immediate consequence of the convexity of $h:\mathcal{D}$, the (unstated) dual of Theorem 6, and the differentiability properties of convex functions (which are given in both [Fenchel, 1951] and [Rockafellar, 1970]).

Since $(\text{ri}\mathcal{V})$ is "almost all" of \mathcal{V} , Theorem 12 implies that almost all consistent problems $\beta(0)$ with a finite infimum $\psi(0)$ are normal, so Theorem 11 shows that duality gaps are rather rare phenomena when both $g:\mathcal{C}$ and \mathcal{X} are convex and closed. Nevertheless, duality gaps can occur when both $g:\mathcal{C}$ and \mathcal{X} are convex and closed, and examples (due originally to J.J. Stoer) can be found in Appendix C of [Peterson, 1970].

5.3.1.6 Decomposition principles. In all problems $\mathcal{A}(0)$ known to the author to be of practical significance (including all examples given in section

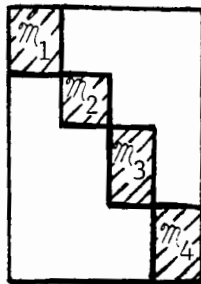
5.2.1) the cone \mathcal{X} is polyhedral and hence "finitely generated". Suppose then without any known loss of practical significance that there is at least one $n \times m$ matrix \mathcal{M} with a corresponding index set $\theta \subseteq \{1, 2, \dots, m\}$ for which

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

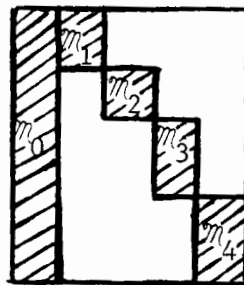
The index set θ can of course be taken to be the empty set when \mathcal{X} is in fact a vector space.

The main prerequisites for decomposing problem $\mathcal{A}(0)$ into smaller (more manageable) subproblems are "sparsity" of the matrix \mathcal{M} and separability of the function $g:\mathcal{C}$. Sparsity of \mathcal{M} is frequently a natural occurrence with the modeling of large systems, while separability of $g:\mathcal{C}$ comes from making the appropriate problem transformations (as illustrated by the examples given in section 5.2.1).

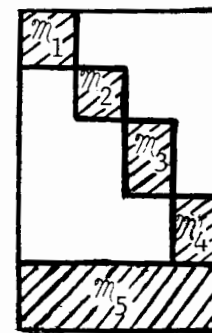
Three different decomposition principles for three different types of sparsity are described here. The three different types of sparsity are indicated by the three different types of "block diagonal structure" illustrated below.



Type 1



Type 2



Type 3

SPARSITY TYPES

The enumerated submatrices \mathcal{M}_k are of course the only submatrices of \mathcal{M} that need not be zero matrices. Assume in general that r such submatrices $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_r$, with $r \geq 2$, are arranged diagonally. (In particular, r is 4 for each of the three illustrated examples.) Matrices \mathcal{M} of type 1 are then those that have no additional nonzero submatrices; matrices \mathcal{M} of type 2 are those that have a single additional nonzero submatrix \mathcal{M}_0 consisting of entire columns of \mathcal{M} ; and matrices \mathcal{M} of type 3 are those that have a single additional nonzero submatrix \mathcal{M}_{r+1} consisting of entire rows of \mathcal{M} . Actually, some matrices of neither type can be effectively transformed into one of the three types simply by row and/or column permutations.

Type 1 sparsity is the easiest to exploit; and its exploitation is at the heart of exploiting both type 2 sparsity and, to some extent, type 3 sparsity. Types 1 and 2 are exploited directly, but type 3 requires the use of geometric Lagrangians and duality.

Each of the three sparsity types induces a partitioning of the rows of \mathcal{M} and hence the components of x in such a way that

$$x = (x^1, x^2, \dots, x^r) \text{ for problems of types 1 and 2}$$

while

$$x = (x^1, x^2, \dots, x^r, x^{r+1}) \text{ for problems of type 3,}$$

where the components of the vector variable x^k are enumerated exactly the same as those rows of \mathcal{M} that contain rows of the submatrix \mathcal{M}_k . Of course, the rows of \mathcal{M}_{r+1} for a given problem of type 3 are the "coupling rows" that must be contended with in reducing such a problem to an equivalent problem of type 1.

Similarly, each of the three sparsity types induces a partitioning of the columns of \mathcal{M} and hence the components of z in such a way that

$$z = (z^1, z^2, \dots, z^r) \text{ for problems of types 1 and 3}$$

while

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

where the components of the vector variable z^k are enumerated exactly the same as those columns of \mathcal{M} that contain columns of the submatrix \mathcal{M}_k . Of course, the columns of \mathcal{M}_0 for a given problem of type 2 are the "coupling columns" that must be contended with in reducing such a problem to an equivalent problem of type 1.

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

$$\mathcal{C} = \prod_{k=1}^r \mathcal{C}_k \text{ and } g(x) = \sum_{k=1}^r g_k(x^k) \text{ for problems of types 1 and 2}$$

while

$$\mathcal{C} = \prod_{k=1}^{r+1} \mathcal{C}_k \text{ and } g(x) = \sum_{k=1}^{r+1} g_k(x^k) \text{ for problems of type 3.}$$

This assumption is, of course, automatically satisfied when $g:\mathcal{C}$ is completely separable, a condition that holds for many of the examples given in section 5.2.1.

Decomposition principles for type 1 and type 2 problems utilize the cones

$$\mathcal{X}_k \triangleq \{x^k \in E_{\mathcal{M}_k} \mid x^k = \mathcal{M}_k z^k \text{ for at least one } z^k \in E_{\mathcal{M}_k} \text{ for which } z_j^k \geq 0, j \in \theta\}.$$

There are of course r such cones \mathcal{X}_k , $k=1,2,\dots,r$ for a problem of type 1, and $r+1$ such cones \mathcal{X}_k , $k=0,1,2,\dots,r$ for a problem of type 2. Only the extra (coupling) cone \mathcal{X}_0 for a problem of type 2 is a subcone of \mathcal{X} .

Problems of type 1. Observe that the cone \mathcal{X} is separable in that

$$\mathcal{X} = \bigtimes_{k=1}^r \mathcal{X}_k.$$

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

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

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

$$\mathcal{A}^{1*}(0) = \bigtimes_{k=1}^r \mathcal{A}_k^*(0).$$

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

It is important to note that the previous assumptions about problem $\mathcal{A}^1(0)$ are inherited by all problems $\mathcal{A}^1(u)$ in the geometric programming family \mathcal{A}^1 . In particular, the cone \mathcal{X} and hence the block diagonal structure of its matrix representation \mathcal{M} remain invariant of u ; and the separability of the function $g:\mathcal{C}$ is clearly inherited by all functions $g(\cdot+u):(\mathcal{C}-u)$.

Consequently, the decomposition principle just described in the context of problem $\mathcal{A}^1(0)$ is just as applicable to all other problems $\mathcal{A}^1(u)$ in the family \mathcal{F}^1 .

In treating such problems $\mathcal{A}^1(u)$, the components of u must be partitioned in the same way that the components of x have been partitioned, namely,

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

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

Now, an application of the decomposition principle just described in the context of problem $\mathcal{A}^1(0)$ to each problem $\mathcal{A}^1(u)$ in a given family \mathcal{F}^1 shows that the (desired) functions $\varphi^1: \mathcal{U}^1$ and $\varphi^{1*}: \mathcal{U}^1$ are determined by the corresponding functions $\varphi_k: \mathcal{U}_k$ and $\varphi_k^*: \mathcal{U}_k$ that are associated with the geometric programming families \mathcal{F}_k that are of course constructed from the respective functions $g_k: \mathcal{C}_k$ and the respective cones \mathcal{X}_k , $k=1, 2, \dots, r$. This direct decomposition of the family \mathcal{F}^1 into r smaller families \mathcal{F}_k , $k=1, 2, \dots, r$ can be concisely described by the formulas

$$\mathcal{U}^1 = \times_{k=1}^r \mathcal{U}_k,$$

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

and

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

where

$$\mathcal{U}_k = \mathcal{C}_k - \mathcal{X}_k, \quad k=1, 2, \dots, r.$$

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

In particular, given that the (one-sided) directional derivative

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

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

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

If a given function $\varphi_k: \mathcal{U}_k$ is convex (which is the case when both \mathcal{C}_k and \mathcal{X}_k are convex), then other relatively mild conditions (as discussed in the preceding subsection 5.3.1.5) guarantee that

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

where $\mathcal{J}_k^*(0; u^k)$ is the optimal solution set for the geometric dual $\beta_k(0; u^k)$ of problem $\mathcal{A}_k(u^k)$. Since $\mathcal{J}_k^*(0, u^k)$ can usually be calculated from a single known optimal solution $x^{k*} \in \mathcal{S}_k^*(u^k)$ by employing the appropriate extremality conditions (as explained after Corollary 3B), and since $\mathcal{J}_k^*(0; u^k)$ is frequently polyhedral (and sometimes a singleton), the directional derivatives $D_{\vec{a}} \varphi^1(u)$ can often be determined by the preceding two displayed formulas by little more than linear programming (and sometimes much less). This means of course that first-order methods can often be used in conjunction with the preceding decomposition principle to minimize $\varphi^1(u)$ over a given subset of \mathcal{U}^1 - a fundamental technique to be used in decomposing problems of type 2.

Problems of type 2. Observe that a given problem $\mathcal{A}(0)$ of type

2, now designated problem $\mathcal{A}^2(0)$, reduces to a problem $\mathcal{A}^1(u)$ of type 1 when the coupling vector variable z^0 is (temporarily) fixed and u is chosen to be $m_0 z^0$. Now, very elementary arguments show that the (desired) infimum $\varphi^2(0)$ for problem $\mathcal{A}^2(0)$ can in fact be determined by the formula

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

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

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

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

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

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

In summary, problem $\mathcal{A}^2(0)$ can be torn into r smaller problems $\mathcal{A}_k(u^k)$, $k=1,2,\dots,r$ that are judiciously selected by the master problem (with the possible help of geometric programming duality in the convex case).

This tearing may increase computational efficiency but can in fact be a necessity when $\mathcal{A}^2(0)$ is itself too large for computer storage.

It is worth noting that the preceding decomposition principle can easily be applied to each problem $\mathcal{A}^2(u)$ in a given family \mathcal{A}^2 whose (unperturbed) problem $\mathcal{A}^2(0)$ is of type 2. However, the result of such an application is notationally cumbersome to describe and is left to the imagination of the interested reader.

Decomposition principles for type 3 problems implicitly utilize the dual problem $\mathcal{B}(0)$. In doing so, they require $g:\mathcal{C}$ to be convex and closed, and they require the absence of a duality gap.

Problems of type 3. Recall from the end of subsection 5.3.1.3 that problem $\mathcal{A}(0)$, now designated problem $\mathcal{A}^3(0)$, can be described in terms of the geometric Lagrangian L_g when $g:\mathcal{C}$ is convex and closed. In particular, the (desired) infimum

$$\omega^3(0) = \inf_{x \in \mathcal{X}} [\sup_{y \in \mathcal{D}_g} L_g(x; y)].$$

Now, the separability inherited by L_g from $g:\mathcal{C}$ (via $h:\mathcal{D}$) clearly implies that

$$\inf_{x \in \mathcal{X}} [\sup_{y \in \mathcal{D}_g} L_g(x; y)] = \inf_{x \in \mathcal{X}} [\sup_{y^{r+1} \in \mathcal{D}_{r+1}} \mathcal{L}_g(x; y^{r+1})],$$

where the "contracted geometric Lagrangian"

$$\mathcal{L}_g(x; y^{r+1}) \triangleq \sum_{k=1}^r g_k(x^k) + \langle y^{r+1}, x^{r+1} \rangle - h_{r+1}(y^{r+1}).$$

Moreover, the absence of a duality gap implies that

$$\inf_{x \in \mathcal{X}} [\sup_{y^{r+1} \in \mathcal{D}_{r+1}} \mathcal{L}_g(x; y^{r+1})] = \sup_{y^{r+1} \in \mathcal{D}_{r+1}} [\inf_{x \in \mathcal{X}} \mathcal{L}_g(x; y^{r+1})]$$

and hence that

$$\varphi^3(0) = \sup_{y^{r+1} \in \mathcal{D}_{r+1}} [\inf_{x \in \mathcal{X}} \mathcal{L}_\theta(x; y^{r+1})].$$

Consequently, the sparsity of \mathcal{M} produces a master problem

$$\varphi^3(0) = \sup_{y^{r+1} \in \mathcal{D}_{r+1}} \Psi(y^{r+1})$$

whose objective function value

$$\Psi(y^{r+1}) \triangleq \sum_{k=1}^r \psi_k(y^{r+1}) - h_{r+1}(y^{r+1}),$$

where the subproblem infima

$$\psi_k(y^{r+1}) \triangleq \inf_{z^k} [\theta_k(\mathcal{M}_k z^k) + \langle y^{r+1}, \mathcal{M}_{r+1}^k z^k \rangle] \quad k=1,2,\dots,r,$$

with \mathcal{M}_{r+1}^k representing that submatrix of \mathcal{M}_{r+1} whose columns correspond to the components of z^k .

It is worth noting that the k 'th subproblem is in fact a geometric programming problem with an objective function value $\theta_k(x^k) + \langle y^{r+1}, x^{r+1} \rangle$ and a cone generated by the columns of the matrix $\begin{bmatrix} \mathcal{M}_k \\ \mathcal{M}_{r+1}^k \end{bmatrix}$ relative to θ .

Moreover, the master problem turns out to be a "suboptimized" version of the geometric dual problem $\mathcal{B}(0)$; that is, $\Psi(y^{r+1})$ is finite if and only if there exists a vector (y^1, y^2, \dots, y^r) such that the augmented vector $y = (y^1, y^2, \dots, y^r, y^{r+1}) \in \mathcal{J}(0)$, in which case $\Psi(y^{r+1})$ is just the negative of the (sub)infimum of $h(y)$ over all such (y^1, y^2, \dots, y^r) .

The preceding fact implies that the master problem consists of maximizing a concave function Ψ over a convex set (which makes it a convex programming problem). Given that $\theta_k : \mathcal{C}_k$, $k = 1, 2, \dots, r, r+1$ is convex and closed, any directional derivatives $D_\theta \Psi(y^{r+1})$ that are needed to implement an appropriate algorithm for solving the master problem can be

obtained from the subgradient representation

$$-D_{\bar{d}}\Psi(y^{r+1}) = \sup_{u^{r+1} \in -\partial\Psi(y^{r+1})} \langle \bar{d}, u^{r+1} \rangle ,$$

where the subgradient set

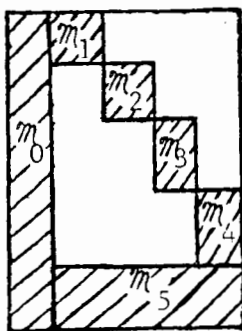
$$\begin{aligned} -\partial\Psi(y^{r+1}) = \{ & u^{r+1} \mid \text{there exist } x^k \in \mathcal{X}_k, k = 1, 2, \dots, r, r+1 \text{ for} \\ & \text{which } 0 = \sum_1^{r+1} \langle x^k, y^k \rangle, x^k \in \partial h_k(y^k), \\ & k = 1, 2, \dots, r \text{ and } x^{r+1} + u^{r+1} \in \partial h_{r+1}(y^{r+1}) \}. \end{aligned}$$

Of course, y^{r+1} is actually an optimal solution to the master problem if and only if $0 \in -\partial\Psi(y^{r+1})$, in which event the (desired) optimal solution set

$$\begin{aligned} \mathcal{S}^{3*}(0) = \{ & (x^1, x^2, \dots, x^r, x^{r+1}) \mid x^k \in \mathcal{X}_k, k = 1, 2, \dots, r, r+1, \\ & 0 = \sum_1^{r+1} \langle x^k, y^k \rangle, \text{ and} \\ & x^k \in \partial h_k(y^k), k = 1, 2, \dots, r, r+1 \}. \end{aligned}$$

It is worth noting that the preceding decomposition principle can easily be applied to each problem $\mathcal{A}^3(u)$ in a given family \mathcal{A}^3 whose (unperturbed) problem $\mathcal{A}^3(0)$ is of type 3. However, the result of such an application is notationally cumbersome to describe and is left to the imagination of the interested reader.

All of the preceding decomposition principles can be combined to treat problems $\mathcal{A}(0)$ for which \mathcal{M} has the following type of block diagonal structure.



Type 4

As indicated by the partitioning of the matrix \mathcal{M} between the submatrices \mathcal{M}_0 and \mathcal{M}_5 (in the preceding illustration), a series of type 3 decompositions is to be guided by a type 2 decomposition.

More detailed descriptions of the preceding decomposition principles can be found in [Peterson, 1973b, 1976].

5.3.2 Relations between the Constrained and Unconstrained Cases. The constrained case can, of course, be specialized to the unconstrained case, simply by letting both index sets I and J be empty while choosing $g_0 : C_0$ to be $g : C$ and X to be \mathcal{X} . A somewhat surprising fact is that this specialization can be reversed; that is, the unconstrained case can actually be specialized to the constrained case.

To do so, let the functional domain

$$C \triangleq \{ (x^0, x^I, \alpha, x^J, \kappa) \in E_n \mid x^0 \in C_0; x^i \in C_i, \alpha_i \in E_1, \text{ and} \\ g_i(x^i) + \alpha_i \leq 0, i \in I; (x^j, \kappa_j) \in C_j^+, j \in J \};$$

and let the functional value

$$g(x^0, x^I, \alpha, x^J, \kappa) \triangleq g_0(x^0) + \sum_J g_j^+(x^j, \kappa_j) \triangleq G(x, \kappa),$$

while letting the cone

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

Then, (the unconstrained) problem \mathcal{A} is clearly identical to (the constrained) problem A. The additional independent vector variable α with components α_i , $i \in I$ may seem superfluous, but is included so that the induced family \mathcal{F} is essentially identical to a family F that includes certain important constraint parameters μ_i (as described in subsection 5.3.3.5).

Of course, the preceding choice of $g : \mathcal{C}$ and \mathcal{X} also induces a choice of the dual problem \mathcal{B} and dual family \mathcal{G} corresponding to problem \mathcal{A} (i.e. problem A). In fact, computations of the resulting conjugate transform $h : \mathcal{D}$ and dual cone \mathcal{Y} (given in Peterson [1975]) show that the induced dual problem \mathcal{B} and dual family \mathcal{G} are essentially identical to the dual problem B and dual family G respectively described in subsections 5.3.3.4 and 5.3.3.5. This identification turns out to be highly significant because it provides an efficient mechanism for extending to the constrained case most of the important theorems already described for the unconstrained case.

5.3.3 The Constrained Case. Only the key ideas and results from the unconstrained case are generalized here, but hopefully the reader can easily fill in the remaining details by using section 5.3.1 as a guide.

Let Y be the "dual" of the cone X; that is,

$$Y \triangleq \{ y \in E_n \mid 0 \leq \langle x, y \rangle \text{ for each } x \in X \} .$$

Of course, the fact that $x \triangleq (x^0, x^I, x^J)$ means that $y \triangleq (y^0, y^I, y^J)$,

with y^I and y^J constructed in the same manner as x^I and x^J .

5.3.3.1. Optimality conditions. We begin with the following fundamental definition.

DEFINITION. A critical solution (stationary solution, equilibrium solution, P solution) for problem A is any vector (x^*, κ^*) for which there is a vector λ^* in $E_{o(I)}$ such that (x^*, κ^*) and λ^* jointly satisfy the following P optimality conditions

$$\begin{aligned} x^* &\in X, \\ g_i(x^{*i}) &\leq 0 && i \in I, \\ \lambda_i^* &\geq 0 && i \in I, \\ \lambda_i^* g_i(x^{*i}) &= 0 && i \in I, \\ y^* &\in Y, \\ 0 &= \langle x^*, y^* \rangle, \end{aligned}$$

and

$$\langle x^{*j}, y^{*j} \rangle = g_j^+(x^{*j}, \kappa_j^*) \quad j \in J,$$

where

$$\begin{aligned} y^{*0} &\triangleq \nabla g_0(x^{*0}), \\ y^{*i} &\triangleq \lambda_i^* \nabla g_i(x^{*i}), && i \in I, \end{aligned}$$

and

$$y^{*j} \triangleq \nabla g_j(x^{*j}/\kappa_j^*) \quad j \in J.$$

For ordinary programming (i.e. example 8 in section 5.2.2), it is worth noting that the P optimality conditions are essentially the (more familiar) "Kuhn-Tucker optimality conditions"

$$\begin{aligned} g_i(z^*) &\leq 0 & i \in I \\ \lambda_i^* &\geq 0 & i \in I \\ \lambda_i^* g_i(z^*) &= 0 & i \in I \end{aligned}$$

and

$$\nabla g_0(z^*) + \sum_I \lambda_i^* \nabla g_i(z^*) = 0.$$

On the other hand, the following important concept from ordinary programming plays a crucial role in the constrained case of geometric programming.

DEFINITION. For a consistent problem A with a finite infimum φ , a Kuhn-Tucker vector is any vector λ^* in $E_{0(I)}$ with the two properties

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

and

$$\varphi = \inf_{\substack{(x, \kappa) \in C \\ x \in X}} L_0(x, \kappa; \lambda^*),$$

where the (ordinary) Lagrangian

$$L_0(x, \kappa; \lambda) \triangleq G(x, \kappa) + \sum_I \lambda_i g_i(x^i).$$

The following theorem gives two convexity conditions that guarantee the necessity and /or sufficiency of the P optimality conditions for optimality.

Theorem 13. Under the hypotheses that g_k is differentiable at x^{*k} , $k \in \{0\} \cup I \cup J$ and that g_j^+ is differentiable at (x^{*j}, κ_j^*) , $j \in J$,

(1) given that X is convex, if (x^*, κ^*) is an optimal solution to problem A, and if λ^* is a Kuhn-Tucker vector for problem A, then (x^*, κ^*) is a critical solution for problem A relative to λ^* (but not conversely),

(2) Given that g_k is convex on C_k , $k \in \{0\} \cup I \cup J$, if

(x^*, κ^*) is a critical solution for problem A relative to λ^* , then (x^*, κ^*) is an optimal solution to problem A, and λ^* is a Kuhn-Tucker vector for problem A.

5.3.3.2. Lagrangian saddle points. Let $h_k: D_k$ be the conjugate transform of the function $g_k: C_k$, $k \in \{0\} \cup I \cup J$, and let these sets D_j determine the sets D_j , $j \in J$ postulated at the beginning of section 5.2.2.

The following definition is of fundamental importance.

DEFINITION. Consider the function H whose domain

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

and whose functional value

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

where

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

and

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

For a consistent problem A with a finite infimum φ , a P vector is any vector (y^*, λ^*) with the two properties

$$(y^*, \lambda^*) \in D$$

and

$$\varphi = \inf_{\substack{x \in X \\ \kappa \geq 0}} L_g(x, \kappa; y^*, \lambda^*),$$

where the (geometric) Lagrangian

$$L_g(x, \kappa; y, \lambda) \triangleq \langle x, y \rangle - H(y, \lambda) - \sum_J \kappa_j h_j(y^j).$$

For a given problem A note that the geometric Lagrangian L_g is entirely different from the ordinary Lagrangian L_o . Unlike the geometric Lagrangian L_g , the ordinary Lagrangian L_o simply reduces to the objective function G when $I = \emptyset$, but L_o exists even when the conjugate transforms $h_k: D_k$, $k \in \{0\} \cup I \cup J$ do not exist. However, Corollary 17B in subsection 5.3.3.4 shows that Kuhn-Tucker vectors and P vectors are closely related.

The following saddle point theorem provides several characterizations of optimality via P vectors.

Theorem 14. Given that $g_k: C_k$, $k \in \{0\} \cup I \cup J$ is convex and closed, let (x^*, κ^*) be such that $x^* \in X$ and $\kappa^* \geq 0$, and let $(y^*, \lambda^*) \in D$. Then, (x^*, κ^*) is optimal for problem A and (y^*, λ^*) is a P vector for problem A if and only if the ordered pair $(x^*, \kappa^*; y^*, \lambda^*)$ is a "saddle point" for the Lagrangian L_g , that is,

$$\sup_{(y, \lambda) \in D} L_g(x^*, \kappa^*; y, \lambda) = L_g(x^*, \kappa^*; y^*, \lambda^*) = \inf_{\substack{x \in X \\ \kappa \geq 0}} L_g(x, \kappa; y^*, \lambda^*);$$

in which case L_g has the saddle point value

$$L_g(x^*, \kappa^*; y^*, \lambda^*) = G(x^*, \kappa^*) = \varphi.$$

Moreover,

$$\sup_{(y, \lambda) \in D} L_g(x^*, \kappa^*; y, \lambda) = L_g(x^*, \kappa^*; y^*, \lambda^*)$$

if and only if (x^*, κ^*) and (y^*, λ^*) satisfy both the feasibility conditions

$$(x^*, \kappa^*) \in C,$$

$$g_i(x^{*i}) \leq 0 \quad i \in I,$$

and the subgradient and "complementary slackness" conditions

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

either $\lambda_i^* = 0$ and $\langle x^{*i}, y^{*i} \rangle = \sup_{c^i \in C_i} \langle c^i, y^{*i} \rangle$, or $\lambda_i^* > 0$ and $y^{*i} \in \lambda_i^* \partial g_i(x^{*i})$, $i \in I$,

$$\lambda_i^* g_i(x^{*i}) = 0 \quad i \in I,$$

either $\kappa_j^* = 0$ and $\langle x^{*j}, y^{*j} \rangle = \sup_{d^j \in D_j} \langle x^{*j}, d^j \rangle$, or $\kappa_j^* > 0$ and $y^{*j} \in \partial g_j(x^{*j}/\kappa_j^*)$, $j \in J$;

in which case

$$L_g(x^*, \kappa^*; y^*, \lambda^*) = G(x^*, \kappa^*).$$

Furthermore,

$$L_g(x^*, \kappa^*; y^*, \lambda^*) = \inf_{\substack{x \in X \\ \kappa \geq 0}} L_g(x, \kappa; y^*, \lambda^*)$$

if and only if (x^*, κ^*) and (y^*, λ^*) satisfy both the feasibility conditions

$$y^* \in Y,$$

$$h_j(y^{*j}) \leq 0 \quad j \in J,$$

and the orthogonality and complementary slackness conditions

$$0 = \langle x^*, y^* \rangle,$$

$$\kappa_j^* h_j(y^{*j}) = 0 \quad j \in J;$$

in which case

$$L_g(x^*, \kappa^*; y^*, \lambda^*) = -H(y^*, \lambda^*).$$

5.3.3.3. The geometric inequality. To introduce duality into the constrained case, the conjugate inequality (given in subsection 5.3.1.2) must be extended in a very special way [Peterson, 1973a].

The resulting "geometric inequality" can actually be derived directly from the conjugate inequality by introducing a scalar variable $\tau \geq 0$. First, suppose that $\tau > 0$ and that ζ/τ is in Ω , so that ζ/τ can be substituted for ζ in the conjugate inequality. Then, multiply the resulting inequality by τ to establish the nontrivial part of the geometric inequality

$$\langle z, \zeta \rangle \leq \tau w(z) + w^+(\zeta, \tau) \text{ for } z \in W \text{ and } (\zeta, \tau) \in \Omega^+,$$

where

$$\Omega^+ \triangleq \{(\zeta, \tau) \in E_{\eta+1} \mid \text{either } \tau = 0 \text{ and } \sup_{z' \in W} \langle z', \zeta \rangle < +\infty, \text{ or } \tau > 0 \text{ and } \zeta \in \tau\Omega\}.$$

and

$$w^+(\zeta, \tau) \triangleq \begin{cases} \sup_{z' \in W} \langle z', \zeta \rangle & \text{if } \tau = 0 \text{ and } \sup_{z' \in W} \langle z', \zeta \rangle < +\infty \\ \tau w(\zeta/\tau) & \text{if } \tau > 0 \text{ and } \zeta \in \tau\Omega. \end{cases}$$

Of course, the trivial part of this geometric inequality is an immediate consequence of the definition of $w^+(\zeta, \tau)$ for $\tau = 0$. Moreover, it is clear from the equality characterization of the conjugate inequality that equality holds if and only if

$$\text{either } \tau = 0 \text{ and } \langle z, \zeta \rangle = \sup_{z' \in W} \langle z', \zeta \rangle, \text{ or } \tau > 0 \text{ and } \zeta \in \tau \partial w(z).$$

Of course, another geometric inequality can be derived from the same conjugate inequality simply by introducing another scalar variable $t \geq 0$ and substituting z/t for z in the conjugate inequality. The details of that inequality are left to the imagination of the reader.

If w is convex and closed, the symmetry of the conjugate transformation clearly implies that the condition $z \in \partial w(\zeta/\tau)$ can be replaced by the condition $\zeta \in \tau \partial w(z)$ in the characterization of equality for the geometric inequality; in which case the relations $z \in \partial w(\zeta/\tau)$ and $\zeta \in \tau \partial w(z)$ are equivalent and hence "solve" one another when $\tau > 0$.

This completes our prerequisites for all of the remaining subsections.

5.3.3.4. Duality. Let $h_k : D_k$ be the conjugate transform of the function $g_k : C_k$, $k \in \{0\} \cup I \cup J$, and let these sets D_j determine the sets D_j , $j \in J$ postulated at the beginning of section 5.2.2.

Now, consider the following geometric programming problem B.

PROBLEM B. Consider the objective function H whose domain

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

and whose functional value

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

where

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

and

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

Using the feasible solution set

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

calculate both the problem infimum

$$\psi \triangleq \inf_{(y, \lambda) \in T} H(y, \lambda)$$

and the optimal solution set

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

Problems A and B are, of course, termed geometric dual problems. When $g_k: C_k$, $k \in \{0\} \cup I \cup J$ and X are convex and closed, this duality is clearly symmetric, in that problem A can then be constructed from problem B in the same way that problem B has just been constructed from problem A. Actually, this duality is the only completely symmetric duality that is presently known for general (closed) convex programming with explicit constraints.

In linear programming (example 7), elementary computations show that dual problem B is, in essence, just the usual linear dual problem. However, in ordinary programming (example 8), elementary considerations show that dual problem B is not just the (Wolfe) "ordinary dual problem" (as properly defined for the first time in [Falk, 1967]). Actually, the ordinary dual problem results from a (sub)optimization of the geometric dual problem

B over y - a fact that indicates why the ordinary dual problem can almost never be computed as easily as the geometric dual problem. Unlike the ordinary dual problem, the geometric dual problem can frequently be computed in terms of elementary functions, particularly for the convex examples described or alluded to in section 5.2.2.

The following definition is almost as important as the definition of the dual problems A and B.

DEFINITION. The extremality conditions (for constrained geometric programming) are:

- (I) $x \in X$ and $y \in Y$
- (II) $g_i(x^i) \leq 0, i \in I$ and $h_j(y^j) \leq 0, j \in J,$
- (III) $0 = \langle x, y \rangle$
- (IV) $y^0 \in \partial g_0(x^0),$
- (V) either $\lambda_i = 0$ and $\langle x^i, y^i \rangle = \sup_{c^i \in C_i} \langle c^i, y^i \rangle$ or $\lambda_i > 0$ and $y^i \in \lambda_i \partial g_i(x^i), i \in I,$
- (VI) either $\kappa_j = 0$ and $\langle x^j, y^j \rangle = \sup_{d^j \in D_j} \langle x^j, d^j \rangle$ or $\kappa_j > 0$ and $y^j \in \partial g_j(x^j/\kappa_j), j \in J,$
- (VII) $\lambda_i g_i(x^i) = 0, i \in I,$ and $\kappa_j h_j(y^j) = 0, j \in J.$

When $C_i = E_{n_i}$ (which is frequently the situation), a simplification results from noting that $\langle x^i, y^i \rangle = \sup_{c^i \in C_i} \langle c^i, y^i \rangle$ if and only if $y^i = 0$. In particular then, the corresponding extremality condition (V) can be replaced by the equivalent extremality condition

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

Extremality conditions (I) are simply the "cone conditions" for problems A and B respectively, and extremality conditions (II) are simply the "constraints" for problems A and B respectively. Extremality condition (III) is termed the "orthogonality condition", extremality conditions (IV) through (VI) are termed the "subgradient conditions", and extremality conditions (VII) are (of course) termed the "complementary slackness conditions".

The following duality theorem is the basis for many others.

Theorem 15. If (x, κ) and (y, λ) are feasible solutions to problems A and B respectively (in which case the extremality conditions (I) through (II) are satisfied), then

$$0 \leq G(x, \kappa) + H(y, \lambda),$$

with equality holding if and only if the extremality conditions (III) through (VI) are satisfied; in which case (x, κ) and (y, λ) are optimal solutions to problems A and B respectively.

In essence, the proof of this key theorem consists of little more than combining the defining inequality $0 \leq \langle x, y \rangle$ with an inequality that comes from summing the conjugate inequality $\langle x^0, y^0 \rangle \leq g_0(x^0) + h_0(y^0)$, the geometric inequalities $\langle x^i, y^i \rangle \leq \lambda_i g_i(x^i) + h_i^+(y^i, \lambda_i)$, $i \in I$, and the geometric inequalities $\langle x^j, y^j \rangle \leq g_j^+(x^j, \kappa_j) + \kappa_j h_j(y^j)$, $j \in J$.

The preceding theorem has two important corollaries that are left to the imagination of the reader because they differ only slightly from the two corollaries to (the analogous) Theorem 3. The first corollary is fundamental to the definition of "duality gap" and is helpful in studying the constrained version of the algorithmic stopping criterion presented in subsection 5.3.1.4. The second corollary provides a useful characterization of dual optimal solutions (x^*, κ^*) and (y^*, λ^*) in terms of the extremality conditions.

It is worth noting that if $g_0: C_0$ is convex and closed, then the symmetry of the conjugate transformation implies that the subgradient condition (IV) can be replaced by the equivalent subgradient condition

$$(IVa) \quad x^0 \in \partial h_0(y^0).$$

Likewise, if $g_i: C_i$, $i \in I$ is convex and closed, then the subgradient condition

(V) can be replaced by the equivalent subgradient condition

$$(Va) \quad \text{either } \lambda_i = 0 \text{ and } \langle x^i, y^i \rangle = \sup_{c^i \in C_i} \langle c^i, y^i \rangle, \text{ or } \lambda_i > 0 \text{ and } x^i \in \partial h_i(y^i / \lambda_i), \quad i \in I;$$

and if $g_j: C_j$, $j \in J$ is convex and closed, then the subgradient condition (VI) can be replaced by the equivalent subgradient condition

$$(VIa) \quad \text{either } \kappa_j = 0 \text{ and } \langle x^j, y^j \rangle = \sup_{d^j \in D_j} \langle x^j, d^j \rangle, \text{ or } \kappa_j > 0 \text{ and } x^j \in \kappa_j \partial h_j(y^j), \quad j \in J.$$

These equivalent subgradient conditions (IVa) through (VIa) are, of course, especially helpful when using the extremality conditions to compute all primal optimal solutions (x^*, κ^*) in S^* from the knowledge of only a single dual optimal solution (y^*, λ^*) in T^* .

It should be emphasized that problem A need not always be solved directly. Under appropriate conditions it can actually be solved indirectly by solving either the extremality condition (I-VII) or problem B. In some cases it may be advantageous to solve the extremality conditions (I-VII), especially when they turn out to be (essentially) linear (e.g. linearly constrained quadratic programming). In other cases it may be advantageous to solve problem B, especially when the index set J is empty (e.g. quadratically constrained quadratic programming and posynomial constrained posynomial programming); in which event problem B has no constraints (even when problem A does). Of course, in all such cases the absence of a duality gap is crucial.

The definition of the Lagrangian L_g (in subsection 5.3.3.2) and the fact that X is a cone readily imply that the cone condition $y \in Y$, the constraints $h_j(y^j) \leq 0$, $j \in J$, the orthogonality condition $0 = \langle x, y \rangle$, and the complementary slackness conditions $\kappa_j h_j(y^j) = 0$, $j \in J$ can all be replaced by the single equivalent condition $L_g(x, \kappa; y, \lambda) = \inf_{\substack{x' \in X \\ \kappa' \geq 0}} L_g(x', \kappa'; y, \lambda)$. Moreover, if $g_k: C_k$,

$k \in \{0\} \cup I \cup J$ is convex and closed, conjugate transform theory implies that the constraints $g_i(x^i) \leq 0$, $i \in I$, the subgradient conditions (IV) through (VI), and the complementary slackness conditions $\lambda_i g_i(x^i) = 0$, $i \in I$ can all be replaced by the two equivalent conditions $\kappa \geq 0$ and $\sup_{(y', \lambda') \in D} L_g(x, \kappa; y', \lambda') = L_g(x, \kappa; y, \lambda)$.

Consequently, the saddle-point condition discussed in subsection 5.3.3.2 is equivalent to the extremality conditions when $g_k: C_k$, $k \in \{0\} \cup I \cup J$ is convex and closed. Nevertheless, it seems that the extremality conditions given in the definition are the most convenient to work with.

The following theorem provides an important tie between dual problem B and the P vectors defined in subsection 5.3.3.2.

Theorem 16. Given that problem A is consistent with a finite infimum φ ,

- (1) if problem A has a P vector, then problem B is consistent and $0 = \varphi + \psi$,
- (2) if problem B is consistent and $0 = \varphi + \psi$, then

$$\{(y^*, \lambda^*) \mid (y^*, \lambda^*) \text{ is a P vector for problem A}\} = T^*.$$

The following theorem provides an important tie between dual problem B and the Kuhn-Tucker vectors defined in subsection 5.3.3.1.

Theorem 17. Given that problems A and B are both consistent and that $0 = \varphi + \psi$, if there is a "minimizing sequence" $\{(y^q, \lambda^q)\}_1^\infty$ for problem B (i.e. $(y^q, \lambda^q) \in T$ and $\lim_{q \rightarrow +\infty} H(y^q, \lambda^q) = \psi$) such that $\lim_{q \rightarrow +\infty} \lambda^q$ exists and is finite, then $\lambda^* \triangleq \lim_{q \rightarrow +\infty} \lambda^q$ is a Kuhn-Tucker vector for problem A.

The following corollary ties the dual optimal solution set T^* directly to Kuhn-Tucker vectors.

Corollary 17A. Given that problems A and B are both consistent and that $0 = \varphi + \psi$, each dual optimal solution $(y^*, \lambda^*) \in T^*$ provides a Kuhn-Tucker vector λ^* for problem A.

The following corollary ties the set of all P vectors (y^*, λ^*) directly to the set of all Kuhn-Tucker vectors λ^* .

Corollary 17B. Given that problem A is consistent with a finite infimum φ , each P vector (y^*, λ^*) for problem A provides a Kuhn-Tucker vector λ^* for problem A.

The proof of this corollary requires Theorem 16 as well as Corollary 17A.

As the preceding theory indicates, the absence of a duality gap is a highly desired situation. The following (geometric programming) version of "Fenchel's theorem" provides useful conditions that guarantee both the absence of a duality gap and the existence of primal optimal solutions $(x^*, \kappa^*) \in S^*$.

Theorem 18. Suppose that both $g_k : C_k$, $k \in \{0\} \cup I \cup J$ and X are convex and closed. If

(i) problem B has a feasible solution (y', λ') such that

$$h_j(y'j) < 0 \quad j \in J,$$

(ii) problem B has a finite infimum ψ ,

(iii) there exists a vector (y^+, λ^+) such that

$$y^+ \in (\text{ri}Y),$$

$$y^{+k} \in (\text{ri}D_k) \quad k \in \{0\} \cup J,$$

$$(y^{+i}, \lambda_i^+) \in (\text{ri}D_i^+) \quad i \in I,$$

then $0 = \varphi + \psi$ and $S^* \neq \emptyset$.

This theorem is perhaps the deepest theorem in geometric programming. Its most direct proof utilizes (the corresponding unconstrained) Theorem 5 along with rather intricate arguments based on "convex analysis".

Note that the vector (y^+, λ^+) in hypothesis (iii) need not be a feasible solution to problem B. However, (y^+, λ^+) is obviously such a solution when J is empty (which is the case for all examples described in section 5.2.2). Consequently, hypothesis (i) can clearly be replaced by the hypothesis

(i') J is empty

without disturbing the validity of Theorem 18. Moreover, when the cone Y is in fact a vector space (which is the case for all examples described or alluded to in section 5.2.2), the condition $y^+ \in (riY)$ is implied by the weaker condition $y^+ \in Y$.

Theorem 18 can be used to show that there are no duality gaps for the constrained versions of problem classes (1) through (3) listed after Theorem 5. Furthermore, its (unstated) dual can be used to strengthen the "Kuhn-Tucker-Slater theorem" in ordinary programming.

5.3.3.5. Parametric programming and postoptimality analysis. As mentioned in section 5.3.2, the parameterized family F into which problem A is to be embedded can be obtained by translating the prescribed domain C (given at the beginning of section 5.3.2) through all possible displacements $-u$, while keeping the prescribed cone X fixed.

Now, the structure of C obviously induces a partitioning of the components of u into vectors $u^0, u^I, \mu, u^J,$ and u_j that correspond respectively to vectors $x^0, x^I, \alpha, x^J,$ and κ . Clearly, each component u_j , of u_j does not influence the problem infimum, but simply translates through $-u_j$ only the optimal value of κ_j (if such a value exists). Hence, setting u_j equal to zero deletes from the resulting family F only problems $A(u)$ that are essentially superfluous. Consequently, the family F is actually taken to be the resulting family F with all such superfluous problems deleted. Of course, the vectors u^0, u^I, u^J and μ still needed to parameterize F constitute a single vector parameter (u, μ) where $(u^0, u^I, u^J) \triangleq u$.

The parameterized family F of all problems $A(u, \mu)$ (for fixed $g_k: C_k, k \in \{0\} \cup I \cup J$ and X) is termed a geometric programming family. For purposes of easy reference and mathematical precision, problem $A(u, \mu)$ is now given the following formal

definition, which should be compared with the formal definition of problem A at the beginning of section 5.2.2.

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

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

and whose functional value

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

where

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

and

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

Using the feasible solution set

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

calculate both the problem infimum

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

and the optimal solution set

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

Of course, problem A appears in the parameterized family F as problem A(0,0); and the symbols A, S, φ and S* now represent function of (u,μ), though they originally represented only the particular functional values A(0,0), S(0,0),

$\varphi(0,0)$ and $S^*(0,0)$ respectively - a notational discrepancy that must be kept in mind when comparing subsequent developments with previous developments.

For the examples given in section 5.2.2, it is easy to see that the perturbation vector (u,μ) :

(6) alters in example 6 the (log of the absolute value of the) signomial coefficients c_q and the constraint upper bounds d_k ,

(7) alters in example 7 the affine objective function constant 0 and the linear constraint upper bounds b_i ,

(8) translates in (the ordinary programming) example 8 the common function domain C_0 in several simultaneous directions while altering the constraint upper bounds 0.

To extend the unconstrained theorems given in subsection 5.3.1.5 into corresponding constrained theorems:

(1) each hypothesis that the set \mathcal{C} be convex should be replaced by the hypothesis that the sets C_0 and C_j^+ , $j \in J$ along with the functions $g_i: C_i$, $i \in I$ be convex,

(2) each hypothesis that the function $g: \mathcal{C}$ be convex (closed) should be replaced by the hypothesis that the functions $g_k: C_k$, $k \in \{0\} \cup I$ and the functions $g_j^+: C_j^+$, $j \in J$ be convex (closed),

(3) each hypothesis that the cone \mathcal{X} be convex (closed) should be replaced by the hypothesis that the cone X be convex (closed),

(4) make the obvious notational alterations in both the hypotheses and the conclusions, as well as the definitions.

In carrying out (4): the symbol \mathcal{L} should be replaced by the symbol G , the symbol ν should be replaced by the symbol (v,ν) , where $(v^0, v^I, v^J) \triangleq \nu$ translates sets and ν influences constraint upper bounds.

Needless to say, the remaining details are left to the interested reader.

5.3.3.6. Decomposition principles. To generalize the decomposition principles given in subsection 5.3.1.6, simply view the constrained case in the context of the unconstrained case via the prescribed choices of \mathcal{C} and \mathcal{X} given in section 5.3.2.

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

It is equally important to realize that the function \mathcal{C} inherits any separability that is present in the functions $g_0:C_0$ and $g_j^+:C_j^+$, $j \in J$. Although \mathcal{C} clearly does not generally inherit any of the separability that is present in a given function $g_i:C_i$ (unless a corresponding Kuhn-Tucker multiplier λ_i^* is known, in which case the constraint $g_i(x^i) + \alpha_i \leq 0$ can be deleted from the defining equation for \mathcal{C} while the expression $\lambda_i^*[g_i(x^i) + \alpha_i]$ is added to the defining equation for $\mathcal{C}(x^0, x^I, \alpha, x^J, \kappa)$), \mathcal{C} does inherit sufficient partial separability when the components of x^i belong to a single vector x^k .

5.4 REFERENCES

- Abrams, R.A., and M.L. Bunting, "Reducing Reversed Posynomial Programs", SIAM J. Appl. Math., to appear, (probably 1975).
- Avriel, M., and A.C. Williams, "Complementary Geometric Programming", SIAM J. Appl. Math., 19, (1970), 125-141.
- Courant, R., and D. Hilbert, Methods of Mathematical Physics, I, Interscience, New York, (1953).

Dinkel, J.J., and E.L. Peterson, "Discrete Optimal Control (with Linear Dynamics) via Geometric Programming," J. Math. Anal. Appls., to appear, (probably 1975).

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

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

_____. "Dual Programs and Minimum Cost," SIAM J. Appl. Math., 10, (1962a), 119.

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

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

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

_____. "Geometric Programs Treated with Slack Variables," J. Appl. Anal., 2, (1972a), 255-267.

_____. "The Proximity of (Algebraic) Geometric Programming to Linear Programming," Math. Prog., 3, (1972b), 250-253.

_____. "Reversed Geometric Programs Treated by Harmonic Means," Indiana Univ. Math. J., 22, (1972c), 531-550.

_____. "Geometric Programming with Signomials," J. Opt. Th. Appls., 11, (1973), 3-35.

Duffin, R.J., E.L. Peterson and C. Zener, Geometric Programming - Theory and Applications, Wiley, New York, (1967) - Russian translation by D. Babayev, Mir, Moscow, (1972).

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

_____. "Global Solutions of Signomial Programs," Technical Report #T-274, George Washington Univ. Prog. in Logistics, Washington, D.C., June, (1973).

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

_____. "Convex Cones, Sets and Functions," Mathematics Department mimeographed lecture notes, Princeton Univ., Princeton, N.J., (1951).

Hall, M.A., and E.L. Peterson, "Multicommodity Flows in Nonlinear Transportation Networks," to appear (probably 1975).

Kretschmer, K.S., "Programmes in Paired Spaces," Canadian J. Math., 13, (1961), 221.

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

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

_____. "Mathematical Foundations of Geometric Programming", Appendix to "Geometric Programming and Some of Its Extensions," Optimization and Design, M. Avriel, M.J. Rijckaert and D.J. Wilde (Eds.), Prentice-Hall, Englewood Cliffs, N.J., (1973a), 244-289.

_____. "The Decomposition of Large (Generalized) Geometric Programming Problems by Tearing," Proc. NATO Inst. Large Scale Systems, D.M. Himmelblau (Ed.), North Holland, Amsterdam, (1973b), 525.

- _____. "Generalization and Symmetrization of Duality in Geometric Programming," to appear, (probably 1975).
- _____. "The Decomposition of Large (Generalized) Geometric Programming Problems by Geometric Lagrangians," to appear, (probably 1976).
- _____. The Mathematical Foundations of Convex and Nonconvex Programming, to appear, (probably 1977).
- Peterson, E.L., and J.G. Ecker, "Geometric Programming: Duality in Quadratic Programming and ℓ_p - approximation, I," Proc. International Sym. Math. Prog., H.W. Kuhn (Ed.), Princeton Univ. Press, Princeton, N.J., (1970), 445-480.
- Peterson, E.L., and R.E. Wendell, "Optimal Location by Geometric Programming," to appear, (probably 1975).
- Rijckaert, M.J., "Engineering Applications of Geometric Programming," Optimization and Design, M. Avriel, M.J. Rijckaert and D.J. Wilde (Eds.), Prentice-Hall, Englewood Cliffs, N.J., (1973), 196-220.
- Rockafellar, R.T., "Duality and Stability in Extremum Problems Involving Convex Functions," Pacific J. Math., 21, (1967a), 167-187.
- _____. "Convex Programming and Systems of Elementary Monotonic Relations," J. Math. Anal. Appls., 19, (1967b), 543-564.
- _____. "Duality in Nonlinear Programming," Amer. Math. Soc. Lects. Appl. Math., 11, G.B. Dantzig and A.F. Veinott (Eds.), Amer. Math. Soc., Providence, R.I., (1968), 401-422.
- _____. Convex Analysis, Princeton Univ. Press, Princeton, N.J., (1970).