A Parametric Successive Underestimation Method for Convex Multiplicative Programming Problems

TAKAHITO KUNO
Department of Social Engineering, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152, Japan

and

HIROSHI KONNO
Institute of Human and Social Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152, Japan

(Received: 19 October 1990; accepted: 15 April 1991)

Abstract. This paper addresses itself to the algorithm for minimizing the product of two nonnegative convex functions over a convex set. It is shown that the global minimum of this nonconvex problem can be obtained by solving a sequence of convex programming problems. The basic idea of this algorithm is to embed the original problem into a problem in a higher dimensional space and to apply a branch-and-bound algorithm using an underestimating function. Computational results indicate that our algorithm is efficient when the objective function is the product of a linear and a quadratic functions and the constraints are linear. An extension of our algorithm for minimizing the sum of a convex function and a product of two convex functions is also discussed.

Key words. nonconvex minimization, global minimization, successive underestimation method, convex multiplicative function, branch-and-bound procedure.

1. Introduction

In a recent series of articles [10, 11], we showed that certain classes of nonconvex minimization problems can be solved by parametric convex minimization algorithms.

The first class of problems is linear multiplicative programming problems (abbreviated as LMP) [10], which is a special type of nonconvex quadratic programming problems whose objective function is the product of two linear functions [1, 2, 16, 20]. We introduced an auxiliary variable and defined the master problem which is equivalent to the original one. Then we applied a parametric simplex algorithm to the master problem. We demonstrated that our algorithm can solve LMP in a little more computational time than needed for solving the associated linear program (i.e., a linear program with the same constraints). The second class of problems is generalized linear multiplicative programming problems (GLMP) [11], whose objective function is the sum of a convex function and a product of two linear functions. We showed that a path

Journal of Global Optimization **1**: 267–285, 1991

following (parametric programming) approach gives us a practical method to calculate a global minimum of GLMP. Konno, Yajima and Matsui [12] proposed an alternative parametric approach for a special case of GLMP in which the convex term of its objective function is affine or quadratic.

We will extend the idea developed in [10, 11] and propose an algorithm for minimizing the product of two nonnegative convex functions over a convex set, which we call a "convex multiplicative programming problem" (CMP). We showed in [10] that CMP can be converted into a parametric convex minimization problem, but did not propose any methods for solving CMP. In this paper, we will solve CMP by applying a branch-and-bound procedure using an underestimating function of its master problem.

The product of two convex functions appears in many areas such as microeconomics [7], VLSI chip design [13], bond portfolio optimization [9], bicriteria optimization problems [6] and so forth (see [16]). Theoretical aspects of this type of nonconvex problems are also dealt with in [18, 19]. Readers are referred to the recent books [8, 17] for the state-of-the-art of nonconvex minimization as well.

In Section 2, we will embed the original problem into its master problem by introducing an auxiliary variable. This reformulation enables us to apply a parametric programming approach. In addition, we will explain several properties of the master problem and show that a successive underestimation method [4, 15, 21] can be applied to the master problem. Section 3 will be devoted to the construction of the algorithm for solving CMP. Results of numerical experiments of our algorithm are also presented. It will be demonstrated that our algorithm can solve fairly large scale problems very efficiently. In Section 4, we will show that the algorithm proposed in Section 3 can be extended to minimizing (i) the product of three convex functions and (ii) the sum of a convex function and a convex multiplicative function.

2. Master Problem for Convex Multiplicative Program

2.1. DEFINITION OF THE MASTER PROBLEM

Let us consider the convex multiplicative programming problem (referred to as CMP) defined below:

$$(\mathrm{CMP})\left|\begin{array}{lll} \text{minimize} & g_0(x) = f_1(x) \cdot f_2(x) & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, m \end{array}\right. \tag{2.1}$$

where $f_1 : R^n \to R^1$, $f_2 : R^n \to R^1$ and $g_i : R^n \to R^1$, $i = 1, \ldots, m$ are twice differentiable convex functions.

g_0 need not be (quasi-)convex nor (quasi-)concave as the following example shows, so that it can have multiple local minima as demonstrated in [10, 20].

EXAMPLE 1. Let $x = (x_1, x_2)^t$ and let $f_1(x) = x_1$, $f_2(x) = x_2^2$. The Hessian matrix of $f_1(x) \cdot f_2(x)$ is

$$\begin{bmatrix} 0 & 2x_2 \\ 2x_2 & 2x_1 \end{bmatrix}$$

which is indefinite. □

We assume in the sequel that the feasible region:

$$X = \{x \in R^n \mid g_i(x) \leq 0, \quad i = 1, \ldots, m\} \tag{2.2}$$

is nonempty and bounded, which implies that (CMP) has a finite optimal solution. We assume further that

$$f_1(x) \geq 0, \quad f_2(x) \geq 0, \quad \forall x \in X. \tag{2.3}$$

If either $f_1(x)$ or $f_2(x)$ attains its lower bound zero at $\tilde{x} \in X$, then $\tilde{x}$ is an optimal solution of (CMP). This can be checked by solving the convex minimization problems:

$$\mathrm{P}_k : \text{minimize}\{f_k(x) \mid x \in X\}, \quad k = 1, 2. \tag{2.4}$$

Therefore the assumption (2.3) can be replaced by

$$f_1(x) > 0, \quad f_2(x) > 0, \quad \forall x \in X \tag{2.5}$$

without loss of generality.

Let us introduce an auxiliary variable $\xi > 0$ and define the following master problem:

$$(\mathrm{MP}) \left| \begin{array}{ll} \text{minimize} & G(x, \xi) = \xi f_1(x) + \frac{1}{\xi} f_2(x) \\ \text{subject to} & x \in X \\ & \xi > 0. \end{array} \right. \tag{2.6}$$

Under the assumption (2.5),

$$\min_{\xi > 0} \left\{ \xi f_1(x) + \frac{1}{\xi} f_2(x) \right\} = 2\sqrt{f_1(x) \cdot f_2(x)}.$$

Hence we obtain the following theorem:

THEOREM 2.1. *Let* (x^*, ξ^*) *be an optimal solution of the master problem* (MP). *Then* x^* *is an optimal solution of* (CMP). □

Let us define the subproblem:

$$\mathrm{P}(\xi)\left|\begin{array}{ll}\text{minimize} & G(x;\xi)=\xi f_1(x)+\dfrac{1}{\xi}f_2(x)\\ \text{subject to} & x\in X\end{array}\right. \tag{2.7}$$

in which ξ has some fixed positive value. Note that $\mathrm{P}(\xi)$ is a convex minimization problem for all $\xi > 0$. Also it has an optimal solution because we assumed that the feasible region X is nonempty and bounded.

Let $x^*(\xi)$ be an optimal solution of $\mathrm{P}(\xi)$ ($\xi > 0$) and let

$$h(\xi) = G(x^*(\xi), \xi) . \tag{2.8}$$

It is easy to see that $h(\xi)$ is continuous for all $\xi > 0$. We need to locate the global minimum point ξ^* of $h(\xi)$ over $\xi > 0$ and $x^*(\xi^*)$, which is guaranteed to be an optimal solution of the original problem (CMP).

2.2. PROPERTIES OF THE MASTER PROBLEM

Let $\mathscr{H}$ be a family of convex functions $H(\xi; a, b)$ which have the following form:

$$H(\xi; a, b) \equiv \xi a + \frac{1}{\xi} b \tag{2.9}$$

where $a, b \in R^1$. Then $h(\xi)$ is the pointwise minimum of an infinite subset of functions $H(\xi; a, b) \in \mathscr{H}$ (see Figure 1) such that

$$a = f_1(x) , \quad b = f_2(x) \tag{2.10}$$

for some $x \in X$.

Tanaka, Thach and Suzuki [21] recently proposed a successive underestimation method to locate the global minimum of $h(\xi)$ over $\xi > 0$ when $h(\xi)$ is the minimum of a finite subset of $\mathscr{H}$. Their method utilizes the following properties of $\mathscr{H}$:

LEMMA 2.2. (i) *Let* (ξ_s, h_s) *and* (ξ_t, h_t) *be any two points in* R^2 *such that* $0 < \xi_s < \xi_t$. *They uniquely determine a function of* $\mathscr{H}$, *namely*

$$H(\xi, \alpha_{st}, b_{st}) = \xi a_{st} + \frac{1}{\xi} b_{st} \tag{2.11}$$

where

$$a_{st} = \frac{h_s\xi_s - h_t\xi_t}{\xi_s^2 - \xi_t^2} , \quad b_{st} = \frac{h_s/\xi_s - h_t/\xi_t}{1/\xi_s^2 - 1/\xi_t^2} . \tag{2.12}$$

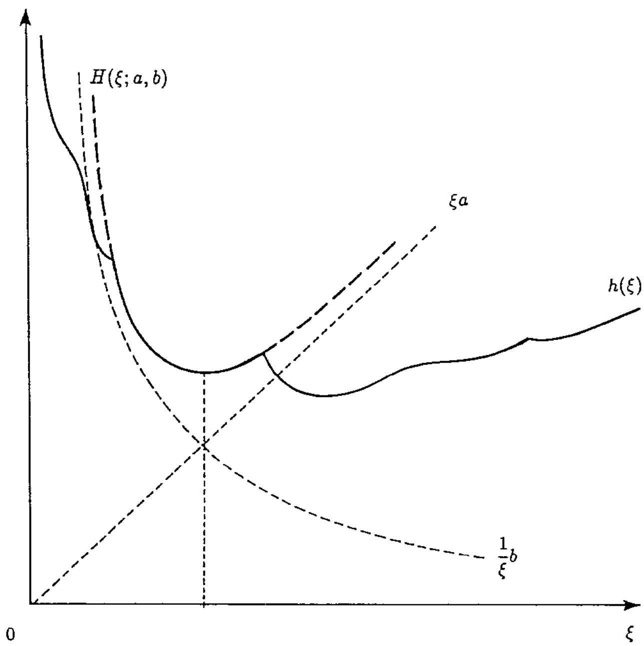

Fig. 1. Relation between $h(\xi)$ and $\mathcal{H}$.

(ii) *Two distinct functions of $\mathcal{H}$ have at most one common point over $\xi > 0$, at which their derivatives are different.*

(iii) *Any function belonging to $\mathcal{H}$ is Lipschitzian over $\xi > \xi_s (>0)$.*

Proof. (i) Simple arithmetic gives the formula (2.11).
(ii) If two functions intersect at more than two distinct points, they are identical because of (i). It is easy to see that the derivatives of two distinct functions of $\mathcal{H}$ are different at the point where they intersect.
(iii) $H(\xi; a, b)$ is Lipschitz continuous with Lipschitz constant $\max\{|a|, |b|/\xi_s^2\}$ over $\xi > \xi_s$. □

THEOREM 2.3. *Let $\xi_t > \xi_s > 0$ and let*

$$U(\xi; \xi_s, \xi_t) \equiv H\left(\xi; \frac{h(\xi_s)\xi_t - h(\xi_t)\xi_t}{\xi_s^2 - \xi_t^2}, \frac{h(\xi_s)/\xi_s - h(\xi_t)/\xi_t}{1/\xi_s^2 - 1/\xi_t^2}\right). \tag{2.13}$$

Then

$$U(\xi; \xi_s, \xi_t) \leq h(\xi), \quad \forall \xi \in [\xi_s, \xi_t]. \tag{2.14}$$

Proof. Assume the contrary. Then there exists $\xi' \in (\xi_s, \xi_t)$ such that

$$U(\xi'; \xi_s, \xi_t) > h(\xi') = \xi' f_1(x^*(\xi')) + \frac{1}{\xi'} f_2(x^*(\xi')).$$

However, the following relationships hold:

$$U(\xi_s\,;\,\xi_s\,,\,\xi_t) = h(\xi_s) \leq \xi_s f_1(x^*(\xi')) + \frac{1}{\xi_s}\, f_2(x^*(\xi'))$$

$$U(\xi_t\,;\,\xi_s\,,\,\xi_t) = h(\xi_t) \leq \xi_t f_1(x^*(\xi')) + \frac{1}{\xi_t}\, f_2(x^*(\xi'))\,.$$

Hence $U(\xi;\,\xi_s\,,\,\xi_t)$ and $\xi f_1(x^*(\xi')) + \frac{1}{\xi} f_2(x^*(\xi'))$ have at least two common points, which contradicts Lemma 2.2 (ii). □

Note that $[h(\xi_s)\xi_s - h(\xi_t)\xi_t]/(\xi_s^2 - \xi_t^2)$ is positive because

$$\begin{aligned} h(\xi_s)\xi_s &= \xi_s^2 f_1(x^*(\xi_s)) + f_2(x^*(\xi_s)) \\ &\leq \xi_s^2 f_1(x^*(\xi_t)) + f_2(x^*(\xi_t)) \\ &< \xi_t^2 f_1(x^*(\xi_t)) + f_2(x^*(\xi_t)) = h(\xi_t)\xi_t\,. \end{aligned}$$

Similarly, $[h(\xi_s)/\xi_s - h(\xi_t)/\xi_t](1/\xi_s^2 - 1/\xi_t^2)$ is positive. Therefore we can obtain the minimum point ξ_1 of $U(\xi;\,\xi_s\,,\,\xi_t)$ over the interval $[\xi_s\,,\,\xi_t]$ by solving a quadratic equation of a single variable. Let us define a function:

$$U_1(\xi) = \begin{cases} U(\xi;\,\xi_s\,,\,\xi_1)\,, & \xi \in [\xi_s\,,\,\xi_1] \\ U(\xi;\,\xi_1\,,\,\xi_t)\,, & \xi \in [\xi_1\,,\,\xi_t]\,. \end{cases} \tag{2.15}$$

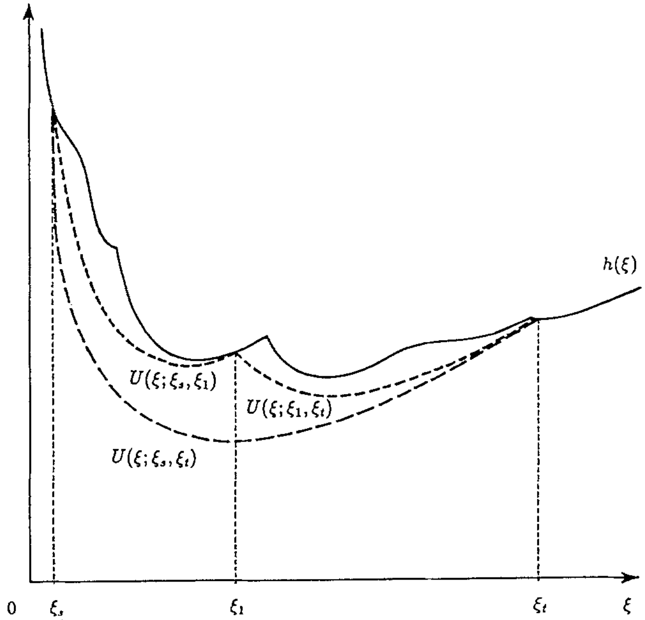

Fig. 2. Underestimating function $U_1(\xi)$.

$U_1(\xi)$ is obviously an underestimating function of $h(\xi)$, i.e., $U_1(\xi) \leq h(\xi)$ over $[\xi_s, \xi_t]$. In addition, We have

$$U(\xi; \xi_s, \xi_t) \leq U_1(\xi) \leq h(\xi), \quad \xi \in [\xi_s, \xi_t]$$

where the first inequality is derived by Lemma 2.2 (ii) (see Figure 2).

Also another underestimating function $U_2(\xi)$ of $h(\xi)$ over $[\xi_s, \xi_t]$ would be generated by applying the above operations to either $[\xi_s, \xi_1]$ or $[\xi_1, \xi_t]$. In this way, we would obtain a sequence of underestimators of $h(\xi)$:

$$U_1(\xi) \leq U_2(\xi) \leq U_3(\xi) \leq \cdots \leq h(\xi), \quad \xi \in [\xi_s, \xi_t]. \tag{2.16}$$

3. Algorithm for Solving the Master Problem

3.1. BRANCH-AND-BOUND PROCEDURE USING THE UNDERESTIMATING FUNCTION

Suppose that the procedure of Section 2.2 generates kth underestimating function $U_k(\xi)$ of $h(\xi)$ over the interval $[\xi_s, \xi_t]$. It can be written as follows:

$$U_k(\xi) = U(\xi; \xi_{kj}, \xi_{k,j+1}), \quad \xi \in [\xi_{kj}, \xi_{k,j+1}], \quad J = 0, 1, \ldots, k \tag{3.1}$$

where $\xi_{k0} = \xi_s$ and $\xi_{k,k+1} = \xi_t$.

Let

$$h(\xi^k) = \min_{0 \leq j \leq k+1} h(\xi_{kj}) \tag{3.2}$$

which gives an upper bound of $h(\xi)$ over $[\xi_s, \xi_t]$. On the other hand, let

$$U_k(\xi_k) = \min_{0 \leq j \leq k} \min\{U(\xi; \xi_{kj}, \xi_{k,j+1}) | \xi \in [\xi_{kj}, \xi_{k,j+1}]\} \tag{3.3}$$

which gives a lower bound of $h(\xi)$. Note that the minimum value of each $U(\xi; \xi_{kj}, \xi_{k,j+1})$ over $[\xi_{kj}, \xi_{k,j+1}]$ can be easily calculated. If

$$h(\xi^k) \leq H_k(\xi_k) \tag{3.4}$$

then ξ^k is obviously the global minimum point of $h(\xi)$ over $[\xi_s, \xi_t]$. Otherwise, we must update the underestimating function of $h(\xi)$ to locate the global minimum point of $h(\xi)$. It should be pointed out that we need not search a subinterval $[\xi_{kj}, \xi_{k,j+1}]$ such that

$$h(\xi^k) \leq \min\{U(\xi; \xi_{kj}, \xi_{k,j+1}) | \xi \in [\xi_{kj}, \xi_{k,j+1}]\}, \tag{3.5}$$

We are now ready to construct a branch-and-bound procedure by putting the above operations together:

Procedure PSUM (j, z, ξ_s, ξ_t)

1. Compute $h(\xi_s)$ and $h(\xi_t)$ by solving the subproblems $P(\xi_s)$ and $P(\xi_t)$, respectively. Generate the underestimation function $U(\xi; \xi_s, \xi_t)$ by using both the values of $h(\xi_s)$ and $h(\xi_t)$.
2. Let
$$\xi_j = \operatorname{argmin}\{U(\xi; \xi_s, \xi_t) | \xi \in [\xi_s, \xi_t]\} .$$
If $U(\xi_j; \xi_s, \xi_t) \geq z$, then return. (Global minimum point of $h(\xi)$ does not exist in $[\xi_s, \xi_t]$.)
3. Compute $h(\xi_j)$ by solving the subproblem $P(\xi_j)$. If $h(\xi_j) < z$, let
$$z = h(\xi_j) , \quad \xi^* = \xi_j . \tag{3.6}$$
If $h(\xi_j) - U(\xi_j; \xi_s, \xi_t) < \epsilon$, then return. ($\xi_j$ is an ϵ-local minimum pint of $h(\xi)$ over the interval $[\xi_s, \xi_t]$).
4. Call Procedure PSUM$(j+1, z, \xi_s, \xi_j)$.
5. Call Procedure PSUM$(j+1, z, \xi_j, \xi_t)$. □

Choosing $\epsilon > 0$ small enough, we can obtain an ϵ-global minimum point ξ^* of $h(\xi)$ by calling the procedure PSUM$(1, +\infty, \xi_{\min}, \xi_{\max})$, where $\xi_{\min}$ and $\xi_{\max}$ are sufficiently small and large positive values, respectively. The solution of $P(\xi^*)$ is an ϵ-optimal solution of (CMP) and $(z/2)^2$ is its ϵ-optimal value.

THEOREM 3.1. *Procedure* PSUM *terminates after finitely many iterations if* $\epsilon > 0$.

Proof. Assume the contrary. Then an infinite sequence ξ_j, $j = l+1, l+2, \ldots$ $(l \geq 0)$ is generated, which converges to some number $\hat{\xi} \in [\xi_s, \xi_t]$. This sequence must satisfy

$$h(\xi_j) - U(\xi_j; \xi_{s_j}, \xi_{t_j}) \geq \epsilon, \quad j = l+1, l+2, \ldots \tag{3.7}$$

where either ξ_{s_j} or ξ_{t_j} is equal to ξ_{j-1}. Since h is the pointwise minimum of functions $H(\xi; a, b) \in \mathcal{H}$, it is Lipschitz continuous with a Lipschitz constant L_h over $[\xi_s, \xi_t]$ (Lemma 2.2(iii)). Also U is Lipschitz continuous with a Lipschitz constant L_U. Thus

$$|h(\xi_j) - U(\xi_j; \xi_{s_j}, \xi_{t_j})| = |h(\xi_j) - U(\xi_{s_{j-1}}; \xi_{s_{j-1}}, \xi_{t_{j-1}})$$
$$+ U(\xi_{s_{j-1}}; \xi_{s_{j-1}}, \xi_{t_{j-1}}) - U(\xi_j; \xi_{s_j}, \xi_{t_j})|$$

$$\leq |h(\xi_j) - h(\xi_{j-1})| + |U(\xi_j : \xi_{s_j}, \xi_{t_j}) - U(\xi_{s_{j-1}}; \xi_{s_{j-1}}, \xi_{t_{j-1}})|$$
$$\leq (L_h + L_U)|\xi_j - \xi_{j-1}|$$

by noting that $U(\xi_{s_{j-1}}; \xi_{s_{j-1}}, \xi_{t_{j-1}}) = h(\xi_j)$. However,

$$\lim_{j\to\infty} |\xi_j - \xi_{j-1}| = 0 ,$$

This contradicts (3.7) □

3.2. ACCELERATION OF CONVERGENCE BY EXPLOITING THE KARUSH–KUHN–TUCKER CONDITIONS

Let us consider the following subproblem:

$$\text{P}(\xi_j) \left| \begin{array}{ll} \text{minimize} & G(x; \xi_j) = \xi_j f_1(x) + \dfrac{1}{\xi_j} f_2(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \end{array} \right. \tag{3.8}$$

where ξ_j is a positive constant. Let us assume that the regularity condition holds at $x^*(\xi_j)$, i.e., that the gradient vectors of binding constraints at $x^*(\xi_j)$ are linearly independent. Then there exists constants λ_i^*, $i = 1, \ldots, m$ for $\text{P}(\xi_j)$, satisfying the Karush–Kuhn–Tucker conditions [14]:

$$\left| \begin{array}{ll} \xi \nabla f_1(x^*(\xi_j)) + \dfrac{1}{\xi_j} \nabla f_2(x^*(\xi_j)) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*(\xi_j)) = 0 & \\ \lambda_i^* \geq 0 , \quad \lambda_i^* g_i(x^*(\xi_j)) = 0 , & i = 1, \ldots, m \\ g_i(x^*(\xi_j)) \leq 0 , & i = 1, \ldots, m . \end{array} \right. \tag{3.9}$$

Also $x^*(\xi_j)$ is an optimal solution of $\text{P}(\xi)$ for all value of $\xi > 0$ as long as the system:

$$\left| \begin{array}{l} \xi \nabla f_1(x^*(\xi_j)) + \dfrac{1}{\xi} \nabla f_2(x^*(\xi_j)) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*(\xi_j)) = 0 \\ \lambda_i \geq 0 , \quad \lambda_i g_i(x^*(\xi_j)) = 0 , \quad i = 1, \ldots, m \end{array} \right. \tag{3.10}$$

has a feasible solution.

From the system (3.10) we can obtain an interval $[\underline{\xi_j}, \overline{\xi_j}]$ such that $x^*(\xi_j)$ is optimal for all $\xi \in [\underline{\xi_j}, \overline{\xi_j}]$ (see [10] for details). This interval is nonempty because ξ_j satisfies (3.10). If $\underline{\xi_j} < \overline{\xi_j}$, we can ignore the interval $[\underline{\xi_j}, \overline{\xi_j}]$ (see Figure 3). In

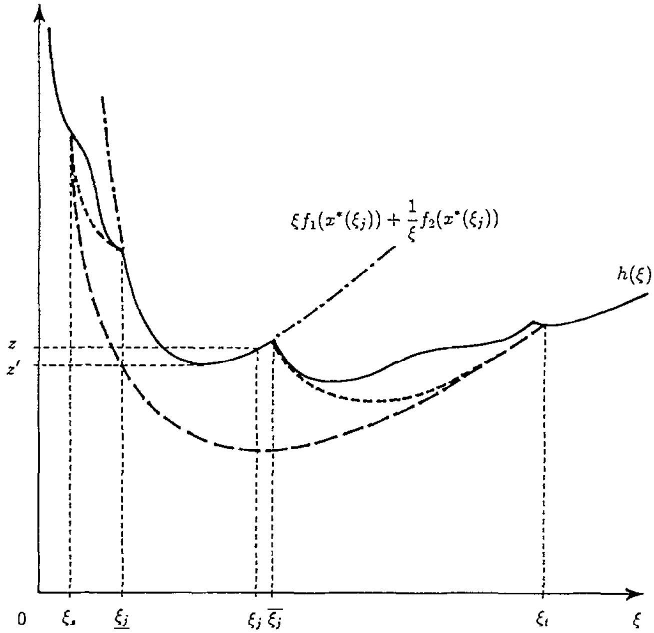

Fig. 3. Improvement by the K–K–T conditions.

addition, a better incumbent will be obtained by replacing (3.6) of Step 3 by

$$z = h(\xi^*) = \min\left\{ \xi f_1(x^*(\xi_j)) + \frac{1}{\xi} f_2(x^*(\xi_j)) \mid \xi \in [\underline{\xi_j}, \overline{\xi_j}] \right\}. \tag{3.11}$$

3.3. COMPUTATIONAL EXPERIMENTS

Let us report the results of the computational experiments of Procedure PSUM for (CMP). We solved problems of the form:

$$\left| \begin{array}{ll} \text{minimize} & g_0(x) = f_1(x) \cdot f_2(x) \\ \text{subject to} & Ax \geqslant b\,, \quad x \geqslant 0 \end{array} \right.$$

where

$$f_1(x) = c^t x + \frac{1}{2}\, x^t Q^t Q x\,, \quad f_2(x) = d^t x$$

and $c, d \in R^n$, $b \in R^m$, $A \in R^{m\times n}$ and $Q \in R^{n\times n}$. All elements of A, b, c, d and Q were randomly generated, whose ranges are $[0, 1.0]$. Matrices A and Q^tQ are almost dense, i.e., about 70 percent of their elements are nonzero.

Let us note that an optimal solution of $\mathrm{P}(\xi)$ is equal to the optimal solution x^0 of a linear program: minimize$\{d^t x \mid Ax \geqslant b, x \geqslant 0\}$ if $\xi > 0$ is small enough. Thus we choose $\xi_{\min}$ as the maximal value of ξ such that x^0 is optimal for P(ξ). $\xi_{\max}$ was always fixed at 10^5.

Subproblems P(ξ) were solved by the reduced gradient method [22]. Direction vectors were generated by applying the conjugate gradient procedure [5]. Programs were coded in C language and ran on a SUN4/280S computer. The size of problems ranges from $(m, n) = (10, 20)$ to $(130, 100)$.

Table I shows the computational results of Procedure PSUM for ten examples for each size when $\xi = 10^{-5}$. The average CPU time (A.v.), its standard deviations (S.d.), the average number of branchings and its minimum (Min.) and maximum (Max.) values are listed in it. The number of branchings corresponds to the number of subproblems P(ξ_j) generated in the course of computation. Table II shows the results of PSUM revised by exploiting the K-K-T conditions. Since the feasible region of the test problem is a polyhedron, the procedure stated in Section 3.2 could be carried out whenever $x^*(\xi_j)$ is a basic solution. The number of basic solutions appeared in the course of computation are also listed in Tables I and II. Table III shows the results when $(m, n) = (30, 50)$ and the value of the tolerance ξ ranges from 10^{-3} to 10^{-7}.

We see from these tables that Procedure PSUM in insensitive to the magnitude of ϵ. This is primarily due to the fact that many branchings were terminated in 2 of Procedure PSUM. It should be emphasized that the number of branchings are very small. This owes very much to an excellent lower bound given by the underestimating function U. It is also worth noting that the number of branchings increase very slowly as the size of problems get larger. Therefore, large sparse problems would be solved efficiently by using sparse matrix techniques. Finally, modification stated in Section 3.2 appears to be fairly effective as is shown in Tables I and II.

Table I. Results of PSUM for (CMP), $\epsilon = 10^{-5}$

m	10	30	30	70	70	130
n	20	20	50	50	100	100
(1) CPU time (in seconds)						
Av.	0.435	2.028	7.828	52.898	167.255	481.695
S.d.	0.204	0.366	1.895	15.509	53.122	146.800
(2) # of branchings						
Av.	7.0	7.9	7.7	11.2	10.9	11.0
Min.	5	6	4	8	7	9
Max.	9	12	11	21	20	15
(3) # of basic solutions						
Av.	4.6	4.4	3.7	6.0	5.5	5.7
Min.	1	1	1	4	1	3
Max.	7	7	7	9	10	9
(1)/(2)						
Av.	0.0621	0.2568	1.0167	4.7231	15.3445	43.7905

Table II. Results of PSUM revised by the K–K–T conditions, $\epsilon = 10^{-5}$

m	10	30	30	70	70	130
n	20	20	50	50	100	100
(1) CPU time (in seconds)						
Av.	0.433	1.938	7.620	48.880	157.808	487.082
S.d.	0.195	0.451	1.949	17.250	50.0299	180.586
(2) # of branchings						
Av.	5.4	5.6	5.8	8.3	8.5	9.1
Min.	3	3	3	4	6	6
Max.	8	12	11	17	14	18
(3) # of basic solutions						
Av.	2.9	2.2	2.4	3.8	3.5	3.8
Min.	1	1	1	2	1	3
Max.	4	3	4	5	6	6
(1)/(2)						
Av.	0.0802	0.3461	1.3138	5.8892	18.5657	53.5255

Table III. Results of PSUM revised by the K–K–T conditions, $(m, n) = (30, 50)$

ϵ	10^3	10^4	10^5	10^6	10^7
(1) CPU time (in seconds)					
Av.	7.643	7.613	7.620	7.713	7.718
S.d.	1.933	1.929	1.949	1.945	1.926
(2) # of branchings					
Av.	5.3	5.5	5.8	6.1	6.2
Min.	3	3	3	3	3
Max.	8	9	11	11	12
(1)/(2)					
Av.	1.4421	1.3842	1.3138	1.2645	1.2449

4. Generalized Convex Multiplicative Program

We show that the parametric successive underestimation method for convex multiplicative programs can be extended to a generalized convex multiplicative programming problem (GCMP):

$$\text{(GCMP)}\left|\begin{array}{ll}\text{minimize} & g_0(x) = f_0(x) + f_1(x) \cdot f_2(x) \\ \text{subject to} & g_i(x) \leqslant 0\,, \quad i = 1, \ldots, m\end{array}\right. \tag{4.1}$$

where $f_k : R^n \to R^1$, $k = 0, 1, 2$ and $g_i : R^n \to R^1$, $i = 1, \ldots, m$ are twice differentiable convex functions. Any method for solving (GCMP) can be used as a procedure for minimizing the product of three convex functions $F_0(x) \cdot F_1(x) \cdot F_2(x)$ over a convex set X, which frequently appears in multicriteria optimization problems, because the objective function of its master problem can be written as $\eta F_0(x) + \frac{1}{\eta} F_1(x) \cdot F_2(x)$ by introducing an auxiliary variable η (Section 2).

As before, we assume that the feasible region:

$$X = \{x \in R^n \mid g_i(x) \leq 0\,, \quad i = 1, \ldots, m\} \tag{4.2}$$

is nonempty and bounded and that

$$f_1(x) > 0\,, \quad f_2(x) > 0\,, \quad \forall x \in X \tag{4.3}$$

The master problem can be defined as follows:

$$(\mathrm{MP}) \left| \begin{array}{ll} \text{minimize} & G(x, \xi) = f_0(x) + \xi \dfrac{[f_1(x)]^2}{2} + \dfrac{1}{\xi} \dfrac{[f_2(x)]^2}{2} \\ \text{subject to} & x \in X \\ & \xi > 0\,. \end{array} \right. \tag{4.4}$$

This problem is equivalent to (GCMP) under the assumption (4.3) (see Theorem 2.1 of [11]).

THEOREM 4.1. *Let (x^*, ξ^*) be an optimal solution of (4.4). Then x^* is an optimal solution of (4.1).* □

For any fixed $\xi > 0$, the subproblem:

$$\mathrm{P}(\xi) \left| \begin{array}{ll} \text{minimize} & G_1(x; \xi) = f_0(x) + \xi \dfrac{[f_1(x)]^2}{2} + \dfrac{1}{\xi} \dfrac{[f_2(x)]^2}{2} \\ \text{subject to} & x \in X \end{array} \right. \tag{4.5}$$

is a convex minimization problem.

Let $x^*(\xi)$ be an optimal solution $\mathrm{P}(\xi)$ and let

$$h(\xi) = G(x^*(\xi); \xi)\,. \tag{4.6}$$

We need to obtain the global minimum point ξ^* of $h(\xi)$ over $\xi > 0$ and $x^*(\xi^*)$, which is an optimal solution of (GCMP).

4.1. UNDERESTIMATING FUNCTIONS FOR THE MASTER PROBLEM

Let $\mathscr{H}'$ be a family of convex functions $H(\xi; a, b, c)$ of the following form:

$$H(\xi; a, b, c) \equiv a + \xi b + \frac{1}{\xi} c \tag{4.7}$$

where

$$a = f_0(x)\,, \quad b = \frac{[f_1(x)]^2}{2}\,, \quad c = \frac{[f_2(x)]^2}{2} \tag{4.8}$$

for some $x \in X$. $h(\xi)$ is the pointwise minimum of an infinite number of functions $H(\xi; a, b, c) \in \mathcal{H}'$.

Let

$$p = \text{minimize}\{f_0(x) | x \in X\} \tag{4.9}$$

and let us define a function:

$$R(\xi; q) \equiv p + \xi q \tag{4.10}$$

for $q \in R^1$.

LEMMA 4.2. *$R(\xi; q)$ and $H(\xi; a, b, c) \in \mathcal{H}'$ have at most one common point over $\xi > 0$, at which their derivatives are different.*

Proof. The intersection points of $R(\xi; q)$ and $H(\xi; a, b, c)$ are given by solving the system:

$$\xi^2(b - q) + \xi(a - p) + c = 0 .$$

If this system has solutions ξ' and ξ'', then the following relationships hold:

$$\xi' + \xi'' = -\frac{a - p}{b - q}, \quad \xi' \cdot \xi'' = \frac{c}{b - q} .$$

Since $p \leq f_0(x)$ for any $x \in X$, either ξ' or ξ'' must be negative. The latter part can be easily checked by a simple arithmetic. □

THEOREM 4.3. *Let $\xi_t > 0$ and let $q_t = [h(\xi_t) - p]/\xi_t$. Then*

$$R(\xi; q_t) \leq h(\xi), \quad \forall \xi \in (0, \xi_t] . \tag{4.11}$$

Proof. Assume that there exits $\xi' \in (0, \xi_t]$ such that

$$\mathrm{R}(\xi'; q_t) > h(\xi') = H(\xi'; a', b', c')$$

where

$$a' = f_0(x^*(\xi')), \quad b' = \frac{[f_1(x^*(\xi'))]^2}{2}, \quad c' = \frac{[f_2(x^*(\xi'))]^2}{2} .$$

However,

$$\lim_{\xi \to +0} [H(\xi; a', b', c') - R(\xi; q_t)] = +\infty$$

and

$$R(\xi_t; q_t) = h(\xi_t) \leq a' + \xi_t b' + \frac{1}{\xi_t} c' .$$

Hence $R(\xi; q_t)$ and $H(\xi; a', b', c')$ have at least two common points, which contradicts Lemma 4.2. □

Let us define a function

$$L(\xi; r) \equiv p + \frac{1}{\xi} r \tag{4.12}$$

for $r \in R^1$. $L(\xi; r)$ has the similar properties to $R(\xi; q)$.

LEMMA 4.4. *$L(\xi; r)$ and $H(\xi; a, b, c) \in \mathcal{H}'$ have at most one common point over $\xi > 0$, at which their derivatives are different.* □

THEOREM 4.5. *Let $\xi_s > 0$ and let $r_s = \xi_s[h(\xi_s) - p]$. Then*

$$L(\xi; r_s) \leq h(\xi), \quad \forall \xi \geq \xi_s \qquad □ \tag{4.13}$$

Let

$$U(\xi; \xi_s, \xi_t) \equiv \max\left\{R\left(\xi; \frac{h(\xi_t) - p}{\xi_t}\right), \quad L(\xi; \xi_s[h(\xi_s) - p])\right\} \tag{4.14}$$

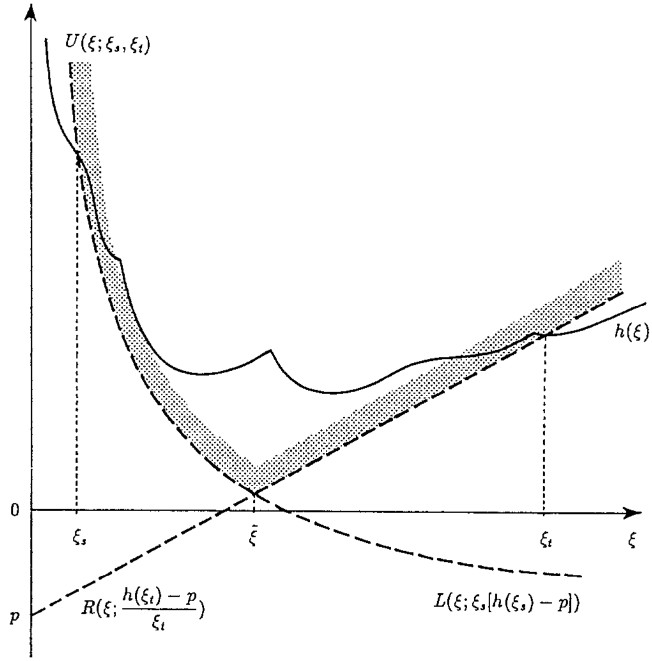

Fig. 4. Underestimating function $U(\xi; \xi_s, \xi_t)$.

then U is a piecewise convex underestimating function of $h(\xi)$ over the interval $[\xi_s, \xi_t]$ $(0 < \xi_s < \xi_t)$ (see Figure 4). We can easily obtain the minimum point $\tilde{\xi}$ of this function, which is given by an intersection of $R(\xi; [h(\xi_t) - p]/\xi_t)$ and $L(\xi; \xi_s[h(\xi_s) - p])$, i.e.,

$$\tilde{\xi} = \sqrt{\xi_s \xi_t \frac{h(\xi_s) - p}{h(\xi_t) - p}} . \tag{4.15}$$

$U(\tilde{\xi}; \xi_s, \xi_t)$ is a lower bound of the global minimum value of $h(\xi)$ over the interval $[\xi_s, \xi_t]$. Thus (GCMP) can be solved by Procedure PSUM using the underestimating function U defined by (4.14). Since both L and R are obviously Lipschitz continuous over $\xi > 0$, the convergence of PSUM is guaranteed in this case as well.

4.2. ALGORITHM WITH LOCAL SEARCH

The minimum point ξ^0 of each $H(\xi; a, b, c) \in \mathcal{H}'$ over $\xi \geqslant 0$ is given by $\xi^0 = \sqrt{c/b}$. By using this property of $\mathcal{H}'$, we can obtain a local minimum point of $h(\xi)$ as follows:

Procedure LS(ξ)

1. Compute $x^*(\xi)$ by solving P(ξ).
2. Let

$$\xi^0 = \frac{f_2(x^*(\xi))}{f_1(x^*(\xi))} .$$

If $\xi^0 = \xi$, then halt. (ξ^0 is a local minimum point.) Otherwise, let $\xi = \xi^0$ and return to Step 1. □

We would be able to accelerate the convergence of PSUM by called Procedure LS($\xi_{\min}$) and LS($\xi_{\max}$) in advance. Let $\xi'_{\min}$ be the output of LS($\xi_{\min}$) and let $\xi'_{\max}$ be that of LS($\xi_{\max}$). Then we would have an interval $[\xi'_{\min}, \xi'_{\max}]$ which is smaller than $[\xi_{\min}, \xi_{\max}]$. Also let

$$z' = \min\{h(\xi'_{\min}), \mathrm{h}(\xi'_{\max})\} \tag{4.16}$$

then z' can be used as an incumbent value of the global minimum value $h(\xi^*)$.

4.3. COMPUTATIONAL EXPERIMENTS

We will report the results of the computational experiments of Procedure PSUM for (GCMP) using the underestimating function stated in the previous subsection.

We solved problems:

$$\left|\begin{array}{ll}\text{minimize} & f_0(x) + f_1(x) \cdot f_2(x) \\ \text{subject to} & Ax \geqslant b\,, \quad x \geqslant 0\end{array}\right.$$

where

$$f_0(x) = c_0^t x + \frac{1}{2}\, x^t Q^t Q x\,, \quad f_1(x) = c_1^t x\,, \quad f_2(x) = c_2^t x$$

$c_k \in R^n$ $(k = 0, 1, 2)$, $b \in R^m$, $A \in R^{m \times n}$ and $Q \in R^{n \times n}$. All elements of A, b, c_k and Q were randomly generated, whose ranges are [0.0, 1.0].

$\xi_{\min}$ and $\xi_{\max}$ were determined by solving minimize $\{c_2^t x | Ax \geqslant b, x \geqslant 0\}$ and minimize $\{c_1^t x | Ax \geqslant b, x \geqslant 0\}$, respectively. As before, Subproblems P(ξ) were solved by the reduced gradient method, programs were coded by C language and ran on a SUN4/280S computer.

Table IV shows the computational results of Procedure LS and PSUM for ten examples for each size when $\epsilon = 10^{-5}$. The average CPU time (AV.), its standard deviations (S.d.), the average number of iterations (or branchings) and its standard deviation of both the procedures are listed in it. The number of iterations as well as branchings corresponds to that of subproblems P(ξ_j) solved in the course of computation. Table V shows the results of PSUM for the different values of the tolerance ϵ when $(m, n) = (30, 50)$.

Table IV includes the average CPU time of the discrete approximation method

Table IV. Results for (GCMP), $\epsilon = 10^{-5}$

m		10	30	30	70	70	130
n		20	20	50	50	100	100
LS:	(1) CPU time (in seconds)						
	Av.	0.685	3.298	15.712	50.368	186.945	449.537
	S.d.	0.245	0.889	6.239	12.087	57.807	121.305
	(2) # of iterations						
	Av.	15.6	23.1	29.1	22.7	32.8	36.9
	S.d.	3.7	12.2	17.6	11.4	6.9	15.2
PSUM:	(3) CPU time (in seconds)						
	Av.	1.170	4.392	24.407	124.457	427.670	553.273
	S.d.	2.945	11.307	34.975	197.759	741.259	732.759
	(2) # of branchings						
	Av.	114.7	171.2	306.8	783.0	372.4	479.0
	S.d.	329.0	510.6	738.1	1212.0	553.8	849.1
Total: (1) + (3)							
	Av.	1.855	7.690	40.118	174.825	614.615	1002.810
DAM:	CPU time (in seconds)						
	Av.	0.985	3.263	16.420	55.463	229.648	511.422

Table V. Results for (GCMP), $(m, n) = (30, 50)$

ϵ		10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
PSUM:	(3) CPU time (in seconds)					
	Av.	13.177	17.267	24.407	36.555	81.820
	S.d.	8.279	16.185	34.975	67.306	177.579
	(4) # of branchings					
	Av.	50.8	107.2	306.8	858.4	2961.6
	S.d.	64.4	190.6	738.1	2230.5	3319.3
Total:	(1) + (3)					
	Av.	28.880	32.958	40.118	52.275	97.522

(DAM) proposed in [11]. We chose the lattice points ξ_j's in DAM by the following formula:

$$x_j = \xi_{\min} + 2^{j\cdot\Delta} - 1 , \quad j = 0, 1, \ldots, 100$$

where $\Delta = \log(\xi_{\max} - \xi_{\min} + 1)/100$. Let

$$\xi_l = \operatorname{argmin}\{h(\xi_j) | 0 \leq j \leq 100\}$$

and call Procedure LS(ξ_l). If the derivative of h at ξ_j was positive, we omitted to compute $h(\xi_j)$.

The underestimating function for (GCMP) defined by (4.14) is not as good as the one for (CMP). However, we see from these tables that the combination of Procedure LS and PSUM can generate a 10^{-5}-optimal solution of the test problem in only twice as much computational time as needed by the discrete approximation method. In addition, the value obtained by PSUM always satisfies the tolerance ϵ. It would be possible to construct a hybrid of PSUM and DAM, which will be discussed elsewhere.

Acknowledgements

This research was supported in part by Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture, Grant No. 63490010. Also the authors would acknowledge the generous support of the Energy Research Institute, Hitachi Co. Ltd.

References

1. Aneja, Y. P., V. Aggarwal, and K. P. K. Nair (1984) 'On a Class of Quadratic Programming', *European J. of Oper. Res.* **18**, 62–70.
2. Bector, C. R. and M. Dahl (1970), 'Simplex Type Finite Iteration Technique and Reality for a Special Type of Pseudo-Concave Quadratic Functions', *Cahiers du Centre d'Etudes de Recherche Operationnelle* **16**, 207–222.

3. Chvătal, V. (1983), *Linear Programming*, W. H. Freedman and Company.
4. Falk, J. E. and K. R. Hoffmann (1976), 'A Successive Underestimating Method for Concave Minimization Problems', *Mathematics of Operations Research* **1**, 251–259.
5. Fletcher, R. and C. M. Reeves, (1964), 'Function Minimization by Conjugate Gradients', *Computer Journal* **7**, 149–154.
6. Geoffrion, A. (1967), 'Solving Bicriterion Mathematical Programs', *Oper. Res.* **15**, 39–54.
7. Henderson, J. M. and R. E. Quandt (1971), *Microeconomic Theory*, McGraw-Hill.
8. Horst, R. and H. Tuy (1990), *Global Optimization: Deterministic Approaches*, Springer-Verlag.
9. Konno, H. and M. Inori (1988), 'Bond Portfolio Optimization by Bilinear Fractional Programming', *J. of Oper. Res. Soc. of Japan* **32**, 143–158.
10. Konno, H. and T. Kuno (1989), 'Linear Multiplicative Programing', IHSS 89-13, Institute of Human and Social Sciences, Tokyo Institute of Technology (to appear in *Mathematical Programming, Ser. A*).
11. Konno, H. and T. Kuno (1990) 'Generalized Linear Multiplicative and Fractional Programming', *Annals of Operations Research* **25**, 147–162.
12. Konno, H., Y. Yajima, and T. Matsui (1991), 'Parametric Simplex Algorithm for Solving a Special Class of Nonconvex Minimization Problems', *J. of Global Optimization* **1**, 65–81.
13. Maling, K., S. H. Mueller, and W. R. Heller (1982), 'On Finding Most Optimal Rectangular Package Plans', Proceedings of the 19th Design Automation Conference, 663–670.
14. Mangasarian, O. L. (1969), *Nonlinear Programming*, McGraw-Hill.
15. McCormick, G. P. (1976), 'Computability of Global Solutions of Factorable Nonconvex Programs: Part I Convex Underestimating Problems', *Mathematical Programming* **10**, 147–175.
16. Pardalos, P. M. (1988), 'Polynomial Time Algorithms for Some Classes of Constrained Nonconvex Quadratic Problems' Computer Science Department, The Pennsylvania State University.
17. Pardalos, P. M. and J. B. Rosen (1987), *Global Optimization: Algorithms and Applications*, Springer-Verlag, Lecture Notes in Computer Science 268.
18. Schaible, S. (1974), 'Maximization of Quasiconcave Quotients and Products of Finitely Many Functions', *Cahiers du Centre d'Etudes de Recherche Operationnelle* **16**, 45–53.
19. Schaible, S. (1973), 'Quasiconcavity and Pseudoconcavity of Cubic Functions', *Mathematical Programming* **5**, 243–247.
20. Swarup, K. (1966), 'Programming with Indefinite Quadratic Function with Linear Constraints', *Cahiers du Centre d'Etudes de Recherche Operationnelle* **8**, 133–136.
21. Tanaka, T., P. T. Thach, and S. Suzuki (1988), 'Methods for an Optimal Ordering Policy for Jointly Replenished Products by Nonconvex Minimization Techniques', Department of Mechanical Engineering, Sophia University.
22. Wolfe, P. (1967), 'Methods of Nonlinear Programming', in J. Abadie (ed.), *Nonlinear Programming*, North-Holland.