Applied Mathematical Sciences, Vol. 2, 2008, no. 29, 1401 - 1414

A Global Optimization Method for Minimizing a Product of Two Positive Convex Functions[1]

Guorong Wu[2,3], Yuelin Gao[4] and Jie Ouyang[3]

[2] Department of Mathematics
Hezuo Minorities Teachers' College
Hezuo, 747000, P. R. China

[3] School of Natural and Applied Science
Northwestern Polytechnical University
Xi'an, 710072, P. R. China

[4] School of Information and Computation Science
North University for Nationalities
Yinchuan, 750021, P. R. China
gaoyuelin@263.net, gor1234yi@tom.com

Abstract

This paper presents a new algorithm for minimizing a product of two positive convex functions. The given problem is firstly converted to an equivalent quasi-concave minimization problem in the outcome space R^2. This quasi-concave minimization problem is then solved by a new branch and bound method. The convergence property is proved. Also tested by the numerical results, the new algorithm is very effective.

Keywords: Global optimization, multiplicative programming, convex multiplicative programming, branch and bound

1 Introduction

This article studies minimization problem of a product of two positive convex functions

$$(SP_D) \quad \begin{cases} global \min & \phi(x) = f_1(x)f_2(x) \\ \text{s.t.} & g_j(x) \leq 0, j = 1, \ldots, m \end{cases}$$

[1]The work is supported by Natural Science Foundation of Ningxia in China(No. NZ0676).

where $f_1(x), f_2(x)$ and $g_j(x), j = 1, \dots, m$, are all finite convex functions in R^n. Let D denote the feasible region of problem (SP_D), obviously D is a convex set. Moreover, suppose D is bounded, and for all $x \in D,\ f_1(x) > 0$ and $f_2(x) > 0$. This problem (SP_D) is a special case of convex multiplicative programming problem where the objective function is a product of p $(p \geq 2)$ convex functions. Compared with convex multiplicative programming problem, problem (SP_D) often costs less computation. But it is still a global optimization problem known to be NP-hard.

There have been quite a few algorithms for solving problem (SP_D) when $f_1(x),\ f_2(x)$ and $g_j(x), j = 1, \dots, m$, are all linear functions. Some are parametric simplex-based methods of Konno and Kuno [8], Konno et al. [11], Schaible and Sodini [15]. Others are branch and bound methods of Pardalos [14], Konno and Kuno [7], Aneja et al. [3] and Tuy and Tam [17] which combine enumeration, discrete approximation, outer approximation, and polyhedral annexation with branch and bound respectively. For the case where $f_1(x)$ and $f_2(x)$ are both finite convex functions in R^n and D is compact and convex, at least three exact global solution algorithms are available for solving problem (SP_D). One of these, an outer approximation method due to Konno et al. [9], applies to the problem of minimizing the sum of k products of two convex functions each, which includes problem (SP_D) as a special case. The other two of these algorithms, by Kuno and Konno [12] and Thoai [16] use underestimation and outer approximation, respectively.

In this article the algorithm proposed for solving problem (SP_D) works in the outcome space R^2 in order to achieve computational efficiency, since n is generally much greater than 2. It combines simplicial reduce with branch and bound in a way of a new flexible branch process. Numerical results show that the new optimization algorithm performs pretty well.

Section 2 converts (SP_D) to a equivalent quasi-concave problem in the outcome space R^2. Section 3 describes a new branch and bound method based on the simplicial reduce. Section 4 constructs a new branch and bound algorithm and proves its convergent property. Section 5 lists the numerical results. In the last section, some concluding remarks are given.

2 Equivalent quasi-concave minimization problem

In this section, we show how to convert problem (SP_D) to a quasi-concave minimization problem (SP_Y) in the outcome space suitable for the new algorithm.

For $i = 1$ and 2, consider the following two convex programming problems

$$\begin{cases} \min & f_i(x) \\ \text{s.t.} & g_j(x) \leq 0, j = 1, \ldots, m \end{cases}$$

whose optimal values are noted as a_1^0 and a_2^0 respectively. Solve the following convex programming problem

$$\begin{cases} \min & f_1(x) \\ \text{s.t.} & f_2(x) \leq a_2^0 \\ & g_j(x) \leq 0, j = 1, \ldots, m \end{cases}$$

to get the optional value b_1^0. Then solve

$$\begin{cases} \min & f_2(x) \\ \text{s.t.} & f_1(x) \leq a_1^0 \\ & g_j(x) \leq 0, j = 1, \ldots, m \end{cases}$$

to get the optimal value b_2^0.

Let $b^0 = (b_1^0, b_2^0)^T, f(x) = (f_1(x), f_2(x))^T$, thus we get the following optimization problem in the outcome space

$$(SP_Y) \quad \begin{cases} \textit{global} \min & z(y) = y_1 y_2 \\ \text{s.t.} & y \in Y, \end{cases}$$

where

$$Y = \{y \in R^2 \mid f(x) \leq y \leq b^0, x \in D\}.$$

The following three theorems are similar to those in the related literature [4], so we don't prove them.

Theorem 1 Problem (SP_Y) consists of the minimization of a function z that is continuous on R^2 and quasi-concave on the nonempty compact convex set Y.

Theorem 2 Problem (SP_Y) has a global optimal solution, and any global optimal solution is in $\partial Y (\partial Y$ denotes the boundary of the set $Y)$.

Theorem 3 Problem (SP_D) and (SP_Y) are equivalent in the following sense: if y^* is a global optimal solution for problem (SP_Y), then any $x^* \in D$ such that $f(x^*) = y^*$ is a global optimal solution for problem (SP_D), and $\phi(x^*) = z(y^*) = f_1(x^*)f_2(x^*)$. Conversely, if x^* is a global optimal solution for problem (SP_D), then $y^* = f(x^*)$ is a global optimal solution for problem (SP_Y), and $\phi(x^*) = z(y^*) = f_1(x^*)f_2(x^*)$.

Define the following set

$$C_0 = \{(y_1, y_2) \mid (y_1, y_2) \in Y, y_2 = \min\{f_2(x) \mid f_1(x) = y_1, x \in D\}\}. \quad (1)$$

As can be seen from (1) that C_0 is a curve if not in the special case where C_0 is a simple point. From Theorem 2, we get the following deduction.

Deduction 1 All the global optimal solutions for problem (SP_Y) lie in C_0.

3 Branch and bound method

3.1 Partition of a simplex

The previous section shows that problem (SP_D) can be converted to problem (SP_Y), so one can solve the former by solving the latter. Moreover, since all the global optimal solutions of problem (SP_Y) are in C_0 according to Deduction 1, one may select a certain set which contains C_0 and thus can serve as an initial outer approximation to C_0. Let $a^0 = (a_1^0, a_2^0)^T$, $c^0 = (c_1^0, c_2^0)^T = (a_1^0, b_2^0)^T$, $d^0 = (d_1^0, d_2^0)^T = (b_1^0, a_2^0)^T$, and $S_0 = co\ \{a^0,\ c^0,\ d^0\}$ represent the convex hull of a^0, c^0 and d^0. Generally S_0 is a 2-dimensional simplex with vertex set $\{a^0,\ c^0,\ d^0\}$. Then $C_0 \subset S_0$, and S_0 is such an initial outer approximation to C_0.

In iteration $k(k \geq 0)$, let the current simplex $S_k = co\ \{a^k,\ c^k,\ d^k\}$ where $c^k = (c_1^k, c_2^k)^T \in C_0$, $d^k = (d_1^k, d_2^k)^T \in C_0$, and $a^k = (c_1^k, d_2^k)^T$. For S_k, the next section will show how to get a new feasible point $\bar{y}^k = (\bar{y}_1^k, \bar{y}_2^k)^T$ in C_0 such that $\bar{y}^k \in C_k$ where C_k is the portion of C_0 between the two points c^k and d^k. We partition S_k along $\bar{y}^k$ into one rectangle and three simplexes below:

$$R_k = [a^k, \bar{y}^k],$$

$$S_{k1} = co\ \{\ c^k, (c_1^k, \bar{y}_2^k)^T, \bar{y}^k\},$$

$$S_{k2} = co\ \{\bar{y}^k, (\bar{y}_1^k, d_2^k)^T, d^k\},$$

$$S_{k3} = co\ \{c^k, \bar{y}^k, d^k\}.$$

Later we will also see, based on the above branching process, in each iteration division sets can be effectively deleted or reduced. Furthermore, a tighter bounding technique is conveniently available which doesn't solve linear programming.

3.2 Finding a feasible point and a corresponding tangent

In iteration $k(k \geq 0)$, consider the following optimization problem

$$(SCP) \quad \begin{cases} \min & (c_2^k - d_2^k) f_1(x) + (d_1^k - c_1^k) f_2(x) \\ \text{s.t.} & g_i(x) \leq 0, j = 1, \dots, m. \end{cases}$$

Theorem 4 Problem (SCP) is a convex programming problem.

Proof Since both the objective function and the constraints of problem (SCP) are convex, the statement is true. □

Theorem 5 Let $\bar{x}^k$ be an optimal solution for problem (SCP), then $\bar{y}^k = (f_1(\bar{x}^k), f_2(\bar{x}^k))^T \in C_0$.

Proof Suppose $\bar{y}^k$ is not in C_0. We define the close line segment as $L(O, \bar{y}^k)$ whose terminals are the origin O and $\bar{y}^k$. Let $\bar{\bar{y}}^k = L(O, \bar{y}^k) \bigcap \partial Y$, then $\bar{\bar{y}}^k \in Y$, $\bar{\bar{y}}^k \leq \bar{y}^k$, and there exists a point $\bar{\bar{x}}^k \in D$ such that $f(\bar{\bar{x}}^k) < \bar{\bar{y}}^k \leq b^0$. Because

$$\begin{aligned} (c_2^k - d_2^k) f_1(\bar{x}^k) + (d_1^k - c_1^k) f_2(\bar{x}^k) &= (c_2^k - d_2^k) \bar{y}_1^k + (d_1^k - c_1^k) \bar{y}_2^k \\ &\geq (c_2^k - d_2^k) \bar{\bar{y}}_1^k + (d_1^k - c_1^k) \bar{\bar{y}}_2^k \\ &> (c_2^k - d_2^k) f_1(\bar{\bar{x}}^k) + (d_1^k - c_1^k) f_2(\bar{\bar{x}}^k), \end{aligned}$$

$\bar{x}^k$ is impossible to be an optimal solution for problem (SCP) and the previous supposition is impossible. Thus $\bar{y}^k \in C_0$. □

Theorem 6 Let $\bar{x}^k$ is an optimal solution for problem (SCP), correspondingly $\bar{y}^k = (\bar{y}_1^k, \bar{y}_2^k)^T = (f_1(\bar{x}^k), f_2(\bar{x}^k))^T$, then

$$(c_2^k - d_2^k)(\bar{y}_1^k - y_1) + (d_1^k - c_1^k)(\bar{y}_2^k - y_2) \leq 0, \forall y = (y_1, y_2)^T \in Y. \tag{2}$$

Proof Let

$$(c_2^k - d_2^k) f_1(\bar{x}^k) + (d_1^k - c_1^k) f_2(\bar{x}^k) = t,$$

i.e.,

$$(c_2^k - d_2^k) \bar{y}_1^k + (d_1^k - c_1^k) \bar{y}_2^k = t. \tag{3}$$

Consider the following optimization problem

$$\begin{cases} \min & (c_2^k - d_2^k) y_1 + (d_1^k - c_1^k) y_2 \\ \text{s.t.} & y = (y_1, y_2)^T \in Y. \end{cases} \tag{4}$$

It holds that t is also the global optimal value of problem (4), so

$$(c_2^k - d_2^k)y_1 + (d_1^k - c_1^k)y_2 \geq t. \tag{5}$$

From (3) and (5), we know that (2) is true. □

From Theorem 6 we can define a tangent passing $\bar{y}^k$ and sustaining Y. The slope of the tangent $K_k = (d_2^k - c_2^k)/(d_1^k - c_1^k)$, then

$$H_k = \{y \in R^p \mid (K_k, -1)(y - \bar{y}) = 0\} \tag{6}$$

is right the tangent. And the corresponding semi-plane

$$H_k^- = \{y \in R^p \mid (K_k, -1)(y - \bar{y}) \leq 0\} \tag{7}$$

contains the feasible region Y of (SP_Y) according to Theorem 6.

3.3 Lower bounds over S_{k1} and S_{k2}

Let in iteration k $(k \geq 0)$, $c^k = (c_1^k, c_2^k)^T$, $d^k = (d_1^k, d_2^k)^T$ and $\bar{y}^k = (\bar{y}_1^k, \bar{y}_2^k)^T$. To define lower bounds over the simplexes S_{k1} and S_{k2} (denoted by $\mu(S_{k1})$ and $\mu(S_{k2})$ respectively), generally one may consider the following convex envelopes of $z(y)$ in S_{k1} and S_{k2}:

$$\omega_{k1}(y) = \bar{y}_2^k y_1 + c_1^k y_2 - c_1^k \bar{y}_2^k, \; y = (y_1, y_2)^T \in S_{k1}, \tag{8}$$

$$\omega_{k2}(y) = d_2^k y_1 + \bar{y}_1^k y_2 - \bar{y}_1^k d_2^k, \; y = (y_1, y_2)^T \in S_{k2}. \tag{9}$$

But the following two aspects make us abandon this idea of the convex envelope method. Firstly, duo to (6) there exist three tangents which pass c^k, $\bar{y}^k$, and d^k respectively, and serve as secants to reduce S_{k1} and S_{k2} further. Let y^{k1} is the crossing point of the two tangents passing c^k and $\bar{y}^k$ respectively, while y^{k2}, the crossing point of the two tangents passing $\bar{y}^k$ and d^k respectively. Then S_{k1} and S_{k2} are respectively reduced to S'_{k1} and S'_{k2}:

$$S'_{k1} = co \; \{ \, c^k, y^{k1}, \bar{y}^k\},$$

$$S'_{k2} = co \; \{\bar{y}^k, y^{k2}, d^k\}.$$

The convex envelopes (8) and (9) don't generate tighter lower bounds over the reduced simplexes S'_{k1} and S'_{k2}. On the other hand, the objective function z of (SP_Y) in S_k is quasi-concave, so enumeration which only compares function values of vertexes of S'_{k1} or S'_{k2} is conveniently available for the definition of lower bound over S'_{k1} or S'_{k2}.

Let K_{k1} and K_{k2} denote the slope of the two tangents passing c^k and d^k respectively, and in the initial iteration $K_{01} = -\infty$, and $K_{02} = 0$. Then we need only to renew K_{k1} and K_{k2} when renewing c^k and d^k at the beginning of iteration k. As is known the slope of the tangent passing $\bar{y}^k$ generated by (6) is $K_k = (d_2^k - c_2^k)/(d_1^k - c_1^k)$. Compute y^{k1} and y^{k2} to get

$$y^{k1} = (t_1, K_{k1}t_1 - K_{k1}c_1^k + c_2^k)^T,$$

where

$$t_1 = (K_{k1}c_1^k - K_k\bar{y}_1^k + \bar{y}_2^k - c_2^k)/(K_{k1} - K_k)$$

and

$$y^{k2} = (t_2, K_k t_2 - K_k\bar{y}_1^k + \bar{y}_2^k)^T,$$

where

$$t_2 = (K_k\bar{y}_1^k - K_{k2}d_1^k + d_2^k - \bar{y}_2^k)/(K_k - K_{k2}).$$

Thus the lower bounds over S_{k1} and S_{k2} (or over S'_{k1} and S'_{k2}) are given as follows:

$$\mu(S_{k1}) = \mu(S'_{k1}) = \min\{z(c^k), z(y^{k1}), z(\bar{y}^k)\}, \quad (10)$$

$$\mu(S_{k2}) = \mu(S'_{k2}) = \min\{z(\bar{y}^k), z(y^{k2}), z(d^k)\}. \quad (11)$$

3.4 Deleting rule

As can be known from the branching process that in iteration k only one feasible point $\bar{y}^k = (\bar{y}_1^k, \bar{y}_2^k)^T$ is in the rectangle $R_k = [a^k, \bar{y}^k]$, and only $\bar{y}^k$, $c^k = (c_1^k, c_2^k)^T$ and $d^k = (d_1^k, d_2^k)^T$ are in both C_0 and $S_{k3} = co\{c^k, \bar{y}^k, d^k\}$. As each of these three points is either in $S_{k1} = co\ \{c^k, (c_1^k, \bar{y}_2^k)^T, \bar{y}^k\}$ or in $S_{k2} = co\ \{\bar{y}^k, (\bar{y}_1^k, d_2^k)^T, d^k\}$, we have a first deleting rule.

(D1): delete R_k and S_{k3} directly.

In iteration k, for each simplex S_{k1}, S_{k2} or any one of the remainder, we give another deleting rule.

(D2): delete any simplex S which satisfies

$$1 - \frac{\mu(S)}{\gamma_k} < \varepsilon,$$

where γ_k denotes the current upper bound.

Thus when the relative precision ε is achieved, the algorithm terminates. Moreover, Rule (D2) signifies if

$$\mu(S_{k1}) = \min\{z(c^k), z(y^{k1}), z(\bar{y}^k)\} = \min\{z(c^k), z(\bar{y}^k)\}, \quad (12)$$

S_{k1} will be deleted, and so will S_{k2} if

$$\mu(S_{k2}) = \min\{z(\bar{y}^k), z(y^{k1}), z(d^k)\} = \min\{z(\bar{y}^k), z(d^k)\}. \tag{13}$$

In fact, suppose without loss of generality (12) is satisfied, then

$$\mu(S_{k1}) = \min\{z(c^k), z(\bar{y}^k)\} = \gamma(S_{k1}) \geq \gamma_k,$$

where $\gamma(S_{k1})$ denotes upper bound over S_{k1}. According to Rule (D2), S_{k1} is deleted.

In section 5, numerical results will show that for all 20 typical random problems, the algorithm needs to save only one subproblem during the iterative process. This shows Rule (D1) and (D2) both work in each iteration, and the deleting techniques based on the branching process are very effective.

4 Algorithm and its convergence

4.1 The outcome space algorithm

Based on the previous discussion, we construct an outcome space algorithm for solving problem (SP_D).

Algorithm 1

step 1 Compute $a^0 = (a_1^0, a_2^0)^T$ and $b^0 = (b_1^0, b_2^0)^T$. If $a^0 = b^0$, stop (arg $z(a^0)$ which denotes some $x \in D$ such that $\phi(x) = z(a^0)$ is an optimal solution, and $z(a^0)$ is the optimal value). Otherwise, $c^0 \leftarrow (a_1^0, b_2^0)^T, d^0 \leftarrow (b_1^0, a_2^0)^T, K_{01} \leftarrow -\infty, K_{02} \leftarrow 0, \upsilon \leftarrow b^0$, and $k \leftarrow 0$.

step 2

$$S_k \leftarrow co\ \{c^k, a^k, d^k\},$$

$$H_{k1} \leftarrow \{y \in R^p \mid (K_{k1}, -1)(y - c^k) = 0\},$$

$$H_{k2} \leftarrow \{y \in R^p \mid (K_{k2}, -1)(y - d^k) = 0\}.$$

Solve convex problem (SCP), from the optimal solution $\bar{x}^k$ compute $\bar{y}^k \in R^p$ and K_k, and carry out (D1).

$$H_k \leftarrow \{y \in R^p \mid (K_k, -1)(y - \bar{y}^k) = 0\},$$

$$y^{k1} \leftarrow H_{k1} \cap H_k,$$

$$y^{k2} \leftarrow H_{k2} \cap H_k,$$

$$\mu(S_{k1}) \leftarrow \min\{z(c^k), z(y^{k1}), z(\bar{y}^k)\},$$

$$\mu(S_{k2}) \leftarrow \min\{z(\bar{y}^k), z(y^{k2}), z(d^k)\},$$

$$\mu_k \leftarrow \min\{\mu_k, \mu(S_{k1}), \mu(S_{k2})\},$$

$$Q \leftarrow \{\upsilon, c^k, \bar{y}^k, d^k\},$$

$$\gamma_k \leftarrow \min\{z(y) \mid y \in Q\},$$

$$\upsilon \leftarrow \arg \gamma_k.$$

Carry out (D2), and if $S = \emptyset$, $\mu_k \leftarrow \gamma_k$; otherwise, select a simplex S from the remainder such that $\mu(S) = \mu_k$.

step 3 If $\mu_k = \gamma_k$, stop (υ is an optimal solution, and γ_k is the optimal value).

step 4 $k \leftarrow k + 1$, renew c^k, d^k, K_{k1}, K_{k2} according to S, and go to *step* 2.

4.2 Convergent proof

Based on Algorithm 1, we give the following convergent theorem.

Theorem 7 Algorithm 1 either generates a global solution for (SP_D) in finite times of iteration, or generates a infinite sequence $\{\bar{y}^k\}$, such that any cluster point y^* of $\{\bar{y}^k\}$ is a global optimal solution for (SP_Y) and any $x^* \in D$ such that $f(x^*) = y^*$ is a global optimal solution for (SP_D).

Proof According to Algorithm 1, $\mu_k \leq z(\bar{y}^k) \leq \gamma_k$, $k = 0, 1, \ldots$. If in iteration k the algorithm terminates, then $\mu_k = z(\bar{y}^k) = \gamma_k$, i.e., $\bar{y}^k$ is a global optimal solution for problem (SP_Y). According to Theorem 3, $\bar{x}^k$ is a global optimal solution for (SP_D).

Otherwise, if the algorithm doesn't terminate in finite times of iteration, let $\{S_{k_q}\}$ be any successively decreasing infinite subsequence of $\{S_k\}$. If $\{K_{k_q}\}$ is unbounded, then $\{d^{k_q}\}$ must be sufficiently close to $c^0 = (a_1^0, b_2^0)^T$, i.e.,

$$\lim_{q\to\infty} c^{k_q} = \lim_{q\to\infty} d^{k_q} = c^0.$$

the previous expression signifies that S_{k_q} is sufficiently small, then

$$\lim_{q\to\infty} \mu_{k_q} = \lim_{q\to\infty} z(\bar{y}^{k_q}) = z(c^0) = \lim_{q\to\infty} \phi(\bar{x}^{k_q}) = \phi(\arg z(c^0)) = \lim_{q\to\infty} \gamma_{k_q},$$

According to the fundamental convergent theory of branch and bound method, c^0 is a global solution of (SP_Y), $\arg z(c^0)$ is a global solution of (SP_D).

If $\{K_{k_q}\}$ is bounded, there exists a positive real number M such that $-M \leq K_{k_q} \leq M (q = 1, 2, \ldots)$. Then in the iteration process of Algorithm 1, $K_{k_q 1} \leq$

$K_{k_q} \leq M$. Since $\{K_{k_q 1}\}$ is monotonously increasing, there must exist a real number α such that

$$\lim_{q\to\infty} K_{k_q 1} = \alpha.$$

On the other hand, let $K_{k_q} \geq -M$, then $K_{k_q 2} \geq K_{k_q} \geq -M$. Since $\{K_{k_q 2}\}$ is monotonously decreasing, there must exist a real number β such that

$$\lim_{q\to\infty} K_{k_q 2} = \beta.$$

Since either $K_{k_{q-1}} = K_{k_q 1}$ or $K_{k_{q-1}} = K_{k_q 2}$ due to definitions of $\{K_{k_q}\}, \{K_{k_q 1}\}$ and $\{K_{k_q 2}\}$, when q is sufficiently great, K_{k_q} is sufficiently close to either α or β. That is to say, the area of S_{k_q} is sufficiently close to zero. Then $\{S_{k_q}\}$ contracts either to a point, or to a segment. The later is impossible, because if so,

$$\mu(S_{k_q 1}) = \min\{z(c^{k_q}), z(y^{k_q 1}), z(\bar{y}^{k_q})\} = \min\{z(c^{k_q}), z(\bar{y}^{k_q})\}, q \longrightarrow \infty$$

and

$$\mu(S_{k_q 2}) = \min\{z(\bar{y}^{k_q}), z(y^{k_q 1}), z(d^{k_q})\} = \min\{z(\bar{y}^{k_q}), z(d^{k_q})\}, q \longrightarrow \infty$$

are both true, i.e., (12) and (13) both hold. From the deleting rule (D2) of Algorithm 1, $S_{k_q 1}$ and $S_{k_q 2}$ are both deleted, i.e., S_{k_q} is deleted. This contradicts to the premise that $\{S_{k_q}\}$ is a successively decreasing infinite sequence. Then the sequence $\{S_{k_q}\}$ must contract to a point $\bar{y}^*$ such that $\lim_{q\to\infty} \bar{y}^{k_q} = \bar{y}^*$ since $\bar{y}^{k_q} \in S_{k_q}$. Thus

$$\lim_{q\to\infty} \mu_{k_q} = \lim_{q\to\infty} z(\bar{y}^{k_q}) = z(\bar{y}^*) = \lim_{q\to\infty} \phi(\bar{x}^{k_q}) = \phi(arg\ \bar{y}^*) = \lim_{q\to\infty} \gamma_{k_q}.$$

So $\bar{y}^*$ is a global solution of (SP_Y), arg $z(\bar{y}^*)$ is a global solution of (SP_D).

Thus the statement is proved. □

5 Numerical results

Consider the two numerical examples in Brigitte et al. [5] to show the effectiveness of Algorithm 1.

Example 1 First consider the randomly generated example

$$(LSP_D) \quad \begin{cases} \min & \alpha_1^T x\ \alpha_2^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0, \end{cases}$$

where the parameters are defined as follows:

—— α_1, α_2 is n dimensional vector with each component in [0,1];

—— $A = (a_{ij}) \in R^m \times R^n$ is a random matrix with each element in [-1,1];

—— $b = (b_1, \ldots, b_m)^T$ is a random vector such that

$$b_i = \sum_{i=1}^{n} a_{ij} + 2b_0,$$

where $i = 1, \ldots, m$, and b_0 is a real number in [0,1].

Example 2 Then consider the randomly generated example

$$(QSP_D) \quad \begin{cases} \min & \alpha_1^T x \ (\alpha_2^T x + x^T D x) \\ \text{s.t.} & Ax \leq b \\ & x \geq 0, \end{cases}$$

where the parameters α_1, α_2, A and b are defined as example 1, and $D = (d_{ij}) \in R^n \times R^n$ is a random diagonal matrix with each diagonal element in [0,1].

The procedure of the algorithm is entirely compiled with Matlab 7.01. And we use standard algorithms in it to solve linear and quadratic programming subproblems. All computations are carried out on personal computer with CPU:c41.7G and RAM:256M. Like Brigitte et al. [5], for each example 1 and example 2, we solve 10 problems and list mean values in the results. And we use the following denotation:

ε: the computational precision,

E: Example 1 (E1) or Example 2 (E2),

n: dimensional of deterministic space,

m: number of constraints,

I: number of iterations,

N: number of nonlinear convex programming problems solved,

M: maximum number of subproblems stored in any iteration,

T: computational time(second).

Table 1 gives numerical results of the 20 examples.

Table 1 Numerical results of Algorithm 1

ε	E	n	m	I	N	M	T
10^{-6}	E1	100	100	8.9	0	1.0	4.84
10^{-6}	E2	100	100	12.2	13.2	1.0	28.36

Brigitte et al. used two different methods (here denoted Method 1 and Method 2 respectively) to define lower bound. Table 2 gives numerical results of Method 1, and Table 3 gives numerical results of Method 2. The same denotation is used in Table 2 and 3 as that in Table 1.

Table 2 Numerical results of Method 1 in [5]

ε	E	n	m	I	N	M	T
10^{-6}	E1	100	100	22.6	74.8	3.4	34.72
10^{-6}	E2	100	100	26.0	85.0	3.8	54.65

Table 3 Numerical results of Method 2 in [5]

ε	E	n	m	I	N	M	T
10^{-6}	E1	100	100	13.0	31.0	1.6	5.96
10^{-6}	E2	100	100	15.2	35.4	2.0	32.28

As can be seen from the three tables Algorithm 1 solves only linear and convex quadratic program for problem (LSP_D) and (QSP_D), avoiding solving difficult nonlinear programming problems which were inevitable in [5]. Besides, in Table 1 number of iterations, number of convex programming problems solved, maximum number of subproblems stored in any iteration and computational time are all more satisfying than those in Table 2 and 3. It is shown that Algorithm 1 is more effective.

6 Concluding remark

This article gives a global optimization algorithm for minimization of a product of two positive convex functions. The algorithm has the following advantages.

(1) The algorithm economizes the computations required to solve problem (SP_D) by working in the outcome space R^2 instead of the decision space R^n.

(2) Due to the simplicial partition, the algorithm fast reduces the feasible region Y of problem (SP_Y).

(3) In each iteration the algorithm uses 2-dimensional tighter simplexes to approximate to the remanent portion of the feasible region, which at the same time makes the bounding technique of enumeration both easily available and comparatively tight.

The algorithm combines simplicial reduce with branch and bound, and can solve higher dimensional problems with more constraints. Theoretical and numerical results show that it is a global convergent and more effective algorithm.

References

[1] D.R. Anatoliy, A Convex Envelope Formula for Multilinear Functions, Journal of Global Optimization, 10 (1997), 425-437.

[2] E. Balas, and J.B. Mazzola, Nonlinear 0-1 Programming: I Liberalization Techniques, Mathematical Programming, 30 (1984), 1-21.

[3] H.P. Benson and G.M. Boger, Multiplicative Programming Problems: Analysis and Efficient Point Search Heuristic, Journal of Optimization Theory and Applications, 94 (1997), 487-510.

[4] H.P. Benson, An Outcome Space Branch and Bound-Out Approximation Algorithm for Convex Multiplicative Programming, Journal of Global Optimization, 15 (1999), 315-342.

[5] J. Brigitte, M. Christophe and T. Hoang, Generalized Convex Multiplicative Programming via Quasiconcave Minimization, Journal of Global Optimization, 4 (1994), 47-62.

[6] B. Jaumard, C. Meyer and H. Tuy, Generalized Convex Multiplicative Programming via Quasiconcave Minimization, Journal of Global Optimization, 10 (1997), 229-256.

[7] H. Konno and T. Kuno, Generalized Linear Multiplicative and Fractional Programming, Annals of Optimizations Research, 25 (1990), 147-161.

[8] H. Konno and T. Kuno, Linear Multiplicative Programming, Mathematical Programming, 56 (1992), 51-64.

[9] H. Konno, Y. Yajima and T. Matssui, Parametric Simplex Algorithms for Solving a Special Class of Nonconvex Minimization Problems, Journal of Global Optimization 1 (1991), 65-82.

[10] T. Kuno and H. Konno, A Parametric Successive Underestimation Method for Convex Multiplicative Programming Problems, Journal of Global Optimization, 1 (1992), 267-286.

[11] T. Kuno and H. Konno, A Parametric Successive Underestimation Method for Convex Multiplicative Programming Problems with an Additional Convex Multiplicative Constraint, Journal of the Optimations Research Society of Japan, 35 (1992), 290-299.

[12] T. Kuno, Y. Yajima and H. Konno, An Outer Approximation Method for Minimizing the Product of Several Convex Function on a Convex Set, Journal of Global Optimization, 3 (1993), 325-335.

[13] T. Kuno, Y. Yajima and Y. Yamamoto, Convex Programs with an Additional Constraint on the Product of Several Convex Functions, European Journal of Operational Research, 7 (1994), 314-324.

[14] P.M. Pardalos, Polynomial Time Algorithms for Some Classes of Constrained Nonconvex Quadratic Problems, Optimization, 21 (1990), 843-853.

[15] S. Schaible and C. Sodini, Finite Algorithm for Generalized Linear Multiplicative Programming, Journal of Optimization Theory and Application, 87 (1995), 441-455.

[16] N.V. Thoai, A Global Optimization Approach for Solving the Convex Multiplicative Programming Problem. Journal of Global Optimization, 1 (1991), 341-357.

[17] H. Tuy and B.T. Tam, An Efficient Solution Method for Rank-Two Quasiconcave Minimization Problems, Optimization, 24 (1992), 43-56.

Received: November 20, 2007