J Glob Optim (2008) 41:579–592
DOI 10.1007/s10898-007-9267-5

A convex analysis approach for convex multiplicative programming

Rúbia M. Oliveira · Paulo A. V. Ferreira

Received: 2 November 2006 / Accepted: 12 October 2007 / Published online: 21 December 2007

Abstract Global optimization problems involving the minimization of a product of convex functions on a convex set are addressed in this paper. Elements of convex analysis are used to obtain a suitable representation of the convex multiplicative problem in the outcome space, where its global solution is reduced to the solution of a sequence of quasiconcave minimizations on polytopes. Computational experiments illustrate the performance of the global optimization algorithm proposed.

Keywords Global optimization · Multiplicative programming · Convex analysis · Numerical methods

JEL Classification A12

1 Introduction

This paper is concerned with multiplicative problems having a product of convex functions as the objective function to be minimized on a convex set. Microeconomics and geometric design are some of the areas where convex multiplicative programming finds interesting applications [13]. Another important source of multiplicative problems are certain convex multiobjective problems in which the product of the individual objectives plays the role of a surrogate objective function [27]. Correspondencies between multiplicative and multiobjective programming have been pointed out in the literature [15,2].

Multiplicative programming is reviewed in [13]. A traditional manipulation is to project the convex multiplicative problem in the *outcome space*, that is, in the real space where the vector of convex functions which constitute the multiplicative objective has its image.

R. M. Oliveira · P. A. V. Ferreira (✉)
Faculty of Electrical & Computer Engineering, University of Campinas, 13084-970 Campinas, SP, Brazil
e-mail: valente@dt.fee.unicamp.br

R. M. Oliveira
e-mail: rubia@dt.fee.unicamp.br

A number of multiplicative programming approaches for solving the problem in the outcome space have been proposed. In [25] the multiplicative problem is reduced to the minimization of a quasiconcave objective over a convex set. The resulting problem is solved by outer approximation. Outer approximation is also employed in [17], where an equivalent concave minimization problem is obtained through a suitable transformation, and then solved by a cutting plane algorithm. The method is generalized in [14] to problems in which the objective function consists of a sum of products of two convex functions. The transformation proposed in [17] is extended in [11], where the multiplicative problem is rewritten as a special quasiconcave minimization problem, whose global optimum is obtained by a conical branch-and-bound algorithm. Another equivalent quasiconcave minimization problem is derived in [1]. Branching, bounding and outer approximation by polytopes are combined to solve the multiplicative problem in the outcome space. In [3] a cutting plane algorithm developed for concave minimization [10] is adapted to the formulation of linear multiplicative problems in the outcome space.

More recently, a number of branch-and-bound techniques have been proposed for affine and generalized affine multiplicative programming. In [16], the affine multiplicative problem is reduced to a separable concave minimization problem, and then solved by a branch-and-bound algorithm specially designed to reduce the number of branching operations. Affine and generalized affine multiplicative problems are treated in [23] by using a combination of a lower bounding procedure proposed by the authors in [22] and a new branching scheme.

Despite its inherent computational complexity, relatively few heuristic algorithms for multiplicative programming have been reported in the literature [2,19]. Special algorithms for some classes of multiplicative problems have been proposed [4]. In a more theoretical framework, in [24] a complete duality theory for a class of quasiconcave multiplicative programs is derived by using conjugate function theory and generalized geometric programming.

The method proposed in this paper is inspired in elements of Convex Analysis [21], employed in projection techniques as the Generalized Benders Decomposition introduced in [9], and whose principles have been progressively extended to global nonconvex optimization problems [7,26]. By combining results of convex analysis and convex multiobjective programming we obtain a representation of the feasible region of the problem in the outcome space as a semi-infinite inequality system. The problem is then solved by relaxation: the solution of a master problem is sent to a maximin subproblem, which tests it with respect to its ϵ-feasibility. If not ϵ-feasible, the solution of the maximin subproblem generates an improved outer approximation of the problem. The procedure eventually converges to an ϵ-optimum solution after finitely many iterations. Solving the master problem involves a quasiconcave minimization over a polytope, carried out by an adequate vertex enumeration procedure. The maximin subproblem involves only the coordinated solution of convex programming problems.

The resulting global optimization algorithm is proved to be substantially more efficient than other global algorithms for convex multiplicative programming based on outer approximations. Its efficiency derives mainly from the use of deepest cuts, which control the generation of cutting planes and, as a result, the growth of vertices to be considered while solving relaxed multiplicative problems in the outcome space. Numerical experiences show, for example, that the algorithm can solve multiplicative problems involving products of, at least, ten linear functions. Applications of exact methods to problems involving products of more than five linear functions have not been reported in the literature.

The paper is organized as follows. In Sect. 2 we formulate the convex multiplicative problem and analyse its connection with convex multiobjective programming. In Sect. 3 an outcome space approach based on elements of convex analysis is proposed. Convergence and implementation issues are discussed. Computational experiments, analyses and comparisons with competing algorithms available in the literature are discussed in Sect. 4. Conclusions are presented in Sect. 5.

Notation. The set of all n-dimensional real vectors is represented as $\mathbb{R}^n$. The sets of all nonnegative and positive real vectors are denoted as $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$, respectively. Inequalities are meant to be componentwise: given $x, y \in \mathbb{R}^n$, then $x \geq y$ ($x - y \in \mathbb{R}^n_+$) implies $x_i \geq y_i,\ i = 1, 2, \dots, n$. Accordingly, $x > y$ ($x - y \in \mathbb{R}^n_{++}$) implies $x_i > y_i,\ i = 1, 2, \dots, n$. The standard inner product and the Euclidean norm in $\mathbb{R}^n$ are denoted as $\langle x, y\rangle$ and $\|x\|$, respectively. The subset of boundary points of $\Omega \subset \mathbb{R}^n$ is denoted as $\partial\Omega$. If $f : \mathbb{R}^n \to \mathbb{R}^m$ is defined on Ω, then $f(\Omega) := \{f(x) :\ x \in \Omega\}$. The symbol $:=$ means *equal by definition*.

2 Preliminary results

Consider the convex multiplicative problem

$$(P_M)\ \left|\ \begin{array}{l} \text{minimize } v(x) = \prod_{i=1}^{m} f_i(x) \\ \text{subject to } \ g_j(x) \leq 0,\ j = 1, 2, \dots, p, \end{array}\right.$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ ($i = 1, 2, \dots, m$) and $g_j : \mathbb{R}^n \to \mathbb{R}$ ($j = 1, 2, \dots, p$) are convex functions. As usual we assume that

$$\Omega := \{x \in \mathbb{R}^n \ :\ g_j(x) \leq 0,\ j = 1, 2, \dots, p\} \tag{1}$$

is a nonempty, compact (convex) set, and that each f_i is positive over Ω. The objective function in (P_M) can be written as the composition $v(x) = u(f(x))$, where $u : \mathbb{R}^m \to \mathbb{R}$, defined by

$$u(y) := \prod_{i=1}^{m} y_i, \tag{2}$$

may be viewed as a particular aggregating function for the problem of minimizing the vector valued objective $f := (f_1, f_2, \dots, f_m)$, $f : \mathbb{R}^n \to \mathbb{R}^m$, on Ω [27]. While v is generally nonconvex on Ω, u is quasiconcave on $\mathbb{R}^m_{++}$ [10]. The positiveness of f on Ω implies that u is quasiconcave on

$$\mathcal{Y} := f(\Omega), \tag{3}$$

the *outcome space* associated with problem (P_M).

A solution $x^\star \in \Omega$ is an *efficient solution* of the multiplicative (multiobjective) problem (P_M) if there exists no other $x \in \Omega$ such that $f(x) \leq f(x^\star)$ and $f(x) \neq f(x^\star)$. We denote the set of all efficient solutions of (P_M) as effi(Ω). Since u is increasing in each y_i ($i = 1, 2, \dots, m$) on the outcome space $\mathcal{Y}$, the following fundamental property holds.

Theorem 1 *Let $x^\star \in \Omega$ be an optimal solution of the convex multiobjective (multiplicative) problem (P_M). Then $x^\star \in$ effi(Ω).*

Proof See [8]. □

It is known from the multiobjective programming literature [27] that if $x \in \Omega$ is an efficient solution of (P_M) then there exists $w \in \mathbb{R}^m_+$ such that x is also an optimal solution of the convex programming problem

$$(P_W)\left|\begin{array}{l}\text{minimize } \langle w, f(x)\rangle\\ \text{subject to } \ x \in \Omega.\end{array}\right.$$

Conversely, let $x(w)$ be any optimal solution of (P_W). Then $x(w)$ is efficient if $w \in \mathbb{R}^m_{++}$. Defining

$$\mathcal{W} := \left\{ w \in \mathbb{R}^m_+ \ : \ \sum_{i=1}^{m} w_i = 1 \right\}, \tag{4}$$

the whole set effi(Ω) can be generated by solving (P_W) over $\mathcal{W}$, being also possible to characterize the optimal solution of (P_M) in terms of problem (P_W).

Theorem 2 *Let $x^\star$ be an optimal solution of (P_M). Then any optimal solution of (P_W) for $w = w^\star$, where*

$$w_i^\star = \prod_{j \neq i} f_j(x^\star) > 0, \quad i = 1, 2, \ldots, m, \tag{5}$$

is also optimal to (P_M).

Proof See [15]. □

When $m = 2$ the optimal weight w^* can be efficiently located by the parametric algorithm proposed in [12].

Theorem 2 is an existence theorem: the optimal weighting vector $w^\star$ depends on the (unknown) optimal solution of (P_M). A method for obtaining $w^\star$ as the limit of the sequence generated by an algorithm derived with basis on convex analysis arguments is proposed in this paper.

3 The outcome space approach

The *outcome space* formulation of (P_M) is

$$(P_{\mathcal{Y}})\left|\begin{array}{l}\text{minimize } u(y) = \prod_{i=1}^{m} y_i\\ \text{subject to } \ y \in \mathcal{Y},\end{array}\right.$$

where $\mathcal{Y}$ is defined by (3). Outcome space formulations have been successfully employed in convex multiplicative programming. A different manipulation of $(P_{\mathcal{Y}})$ which combines elements of convex analysis and multiobjective optimization in the outcome space [6] is proposed in this section. Preliminarily, it is worth noting that the continuity of f and the compactness of Ω imply the compactness of $\mathcal{Y}$. The convexity of Ω also entails the connectedness by arcs of $\mathcal{Y}$, but not its convexity.

The set of all efficient solutions in the outcome space is given by effi$(\mathcal{Y}) = f(\text{effi}(\Omega))$. It is readily seen that if $y \in \text{effi}(\mathcal{Y})$ then $y \in \partial\mathcal{Y}$. Central to the global algorithm proposed in this paper is the following result.

Theorem 3 *Define $\mathcal{F} := \mathcal{Y} + \mathbb{R}^m_+$. Then*

i) *$\mathcal{F}$ is a convex set;*
ii) *effi$(\mathcal{F})$ = effi$(\mathcal{Y})$.*

Proof See [27]. □

Since $\mathcal{F}$ is a convex set, effi$(\mathcal{F})$ = effi$(\mathcal{Y})$ and effi$(\mathcal{F}) \subset \partial\mathcal{F}$, it follows that $\mathcal{F}$ admits a supporting half-space at each efficient solution of $(P_\mathcal{Y})$.

We observe that $\mathcal{F}$ can be explicitly represented as

$$\mathcal{F} := \{y \in \mathbb{R}^m \ : \ f(x) \le y \quad \text{for some } x \in \Omega\}, \tag{6}$$

as any $y \in \mathcal{F}$ is a sum of elements of $\mathcal{Y}$ and $\mathbb{R}^m_+$. (We redefine the result of the sum as y, for convenience.) The importance of Theorem 3 relies on the fact that now it is possible to obtain an equivalent outcome space formulation with a convex closed (but unbounded) feasible region:

$$(P_\mathcal{F}) \left| \begin{array}{l} \text{minimize } u(y) \\ \text{subject to} \quad y \in \mathcal{F}. \end{array} \right.$$

Theorem 4 *Let $y^\star \in \mathcal{F}$ be an optimal solution of $(P_\mathcal{F})$. Then*

i) *$y^\star \in$ effi$(\mathcal{Y})$;*
ii) *$y^\star$ is an optimal solution of $(P_\mathcal{Y})$.*

Proof If $y^\star \in \mathcal{F}$ solves $(P_\mathcal{F})$, there exists a $x^\star \in \Omega$ such that $y^\star = f(x^\star) \in \mathcal{Y}$. Otherwise, if $y^\star \ge f(x^\star)$ and $y^\star \ne f(x^\star)$, then $y^0 = f(x^\star)$ would contradict the optimality of $y^\star$, because $y^0 \in \mathcal{F}$ and $u(y^0) < u(y^\star)$. Clearly, $y^\star \in$ effi$(\mathcal{Y})$, and since $\mathcal{Y} \subset \mathcal{F}$, we conclude that $y^\star$ is also an optimal solution of $(P_\mathcal{Y})$. □

3.1 Feasibility problem

Differently from $(P_\mathcal{Y})$, the feasible region of $(P_\mathcal{F})$ is amenable to representation through convex analysis results. Although Theorem 5 below holds in a more general context [18], an alternative proof is provided in order to illustrate the role played by convex analysis in this paper.

Theorem 5 *$y \in \mathcal{F}$ if and only if y satisfies the semi-infinite inequality system*

$$\min_{x\in\Omega} \langle w, f(x) - y\rangle \le 0 \quad \textit{for all} \quad w \in \mathcal{W}. \tag{7}$$

In addition, every linear inequality in (7) defines a supporting half-space for $\mathcal{F}$.

Proof Suppose that $\bar{y} \in \mathcal{F}$. Then there exists $\bar{x} \in \Omega$ such that $f(\bar{x}) \le \bar{y}$. It follows that

$$\langle w, f(\bar{x}) - \bar{y}\rangle \le 0 \quad \text{for all} \quad w \in \mathcal{W},$$

and (7) holds for $y = \bar{y}$. Now suppose that $\bar{y} \notin \mathcal{F}$. By the Separation Theorem [21] there exists $w \in \mathbb{R}^m\backslash\{0\}$ and $t \in \mathbb{R}$ such that $\langle w, \bar{y}\rangle < t$ and $\langle w, y\rangle \ge t$ for all $y \in \mathcal{F}$, which implies $w \in \mathbb{R}^m_+$, because, if $w_i < 0$ for some i, the inequality $\langle w, y\rangle \ge t$ could be violated by selecting a sufficiently large $y \in \mathcal{F} \subset \mathbb{R}^m_{++}$. Then we can assume $w \in \mathcal{W}$, and knowing that $\langle w, f(x)\rangle \ge t$ for all $x \in \Omega$, it follows that $\langle w, f(x) - \bar{y}\rangle > 0$ for all $x \in \Omega$. Since Ω is compact, we conclude that $\min_{x\in\Omega}\langle w, f(x) - \bar{y}\rangle > 0$ for some $w \in \mathcal{W}$.

Finally, for any $w \in \mathcal{W}$,

$$\mathcal{H} := \{y \in \mathbb{R}^m \ : \ \langle w, y \rangle \geq \langle w, f(x(w)) \rangle\}$$

is a supporting half-space for $\mathcal{F}$, because $y \in \mathcal{F}$ implies $y \in \mathcal{H}$ and $\mathcal{H}$ contacts $\mathcal{F}$ at $y = f(x(w))$. $\square$

In practice, we implement the following corollary of Theorem 5: $y \in \mathcal{F}$ if and only if $\theta(y) \leq 0$, where

$$\theta(y) := \max_{w \in \mathcal{W}} \phi_y(w) \tag{8}$$

and

$$\phi_y(w) := \min_{x \in \Omega} \langle w, f(x) - y \rangle. \tag{9}$$

Theorem 6 *Let ϕ_y and θ be defined by (9) and (8), respectively. Then*

i) Given $y \in \mathbb{R}^m$, the function ϕ_y is concave on $\mathcal{W}$;
ii) $f(x(w)) - y$ is a supergradient of ϕ_y at $w \in \mathcal{W}$;
iii) The function θ is convex on $\mathbb{R}^m$.

Proof

i) For all $w^1, w^2 \in \mathcal{W}$ and all $\alpha \in [0, 1]$,

$$\begin{aligned}
\phi_y(\alpha w^1 + (1-\alpha) w^2) &= \min_{x \in \Omega} \langle \alpha w^1 + (1-\alpha) w^2, f(x) - y \rangle \\
&= \min_{x \in \Omega} \left\{ \langle \alpha w^1, f(x) - y \rangle + \langle (1-\alpha) w^2, f(x) - y \rangle \right\} \\
&\geq \alpha \min_{x \in \Omega} \langle w^1, f(x) - y \rangle + (1-\alpha) \min_{x \in \Omega} \langle w^2, f(x) - y \rangle \\
&= \alpha \phi_y(w^1) + (1-\alpha) \phi_y(w^2).
\end{aligned}$$

ii) From the definition of ϕ_y,

$$\phi_y(w) \leq \langle w, f(x(w^0)) - y \rangle \quad \text{for all } w, w^0 \in \mathcal{W}$$

and

$$\phi_y(w^0) = \langle w^0, f(x(w^0)) - y \rangle \quad \text{for all } w^0 \in \mathcal{W}.$$

Subtracting the last two expressions, we obtain

$$\phi_y(w) \leq \phi_y(w^0) + \langle f(x(w^0)) - y, w - w^0 \rangle \tag{10}$$

for all $w \in \mathcal{W}$, meaning that $f(x(w^0)) - y$ is a supergradient of ϕ_y at $w^0 \in \mathcal{W}$.

iii) Since the minimum on the right-hand side of (9) does not depend on y, the following property holds: For all $y^1, y^2 \in \mathbb{R}^m$ and all $\alpha \in [0, 1]$,

$$\phi_{\alpha y^1 + (1-\alpha) y^2}(w) = \alpha \phi_{y^1}(w) + (1-\alpha) \phi_{y^2}(w).$$

Therefore, for all $y^1, y^2 \in \mathbb{R}^m$ and all $\alpha \in [0, 1]$,

$$\begin{aligned}\theta(\alpha y^1 + (1-\alpha)y^2) &= \max_{w \in \mathcal{W}} \phi_{\alpha y^1 + (1-\alpha)y^2}(w) \\ &= \max_{w \in \mathcal{W}} \left\{\alpha\phi_{y^1}(w) + (1-\alpha)\phi_{y^2}(w)\right\} \\ &\leq \alpha \max_{w \in \mathcal{W}} \phi_{y^1}(w) + (1-\alpha) \max_{w \in \mathcal{W}} \phi_{y^2}(w) \\ &= \alpha\theta(y^1) + (1-\alpha)\theta(y^2),\end{aligned}$$

and θ is convex on $\mathbb{R}^m$. The convexity of θ implies its continuity on $\mathbb{R}^m$, a property to be used in Sect. 3.3, where the convergence of the global optimization algorithm proposed is demonstrated. □

At each point $w \in \mathcal{W}$, a supergradient $f(x(w)) - y$ determines a supporting hyperplane to the hypograph of ϕ_y, which enables us to build piecewise linear approximations to ϕ_y. A l-th approximation to ϕ_y would be

$$\phi_y^l(w) = \min_{1 \leq i \leq l} \{\langle w, f(x(w^i)) - y\rangle\}. \tag{11}$$

The maximization of ϕ_y over $\mathcal{W}$ can be carried out by the outer approximation algorithm discussed in [18] in the context of the solution of dual programming problems. Instead of maximizing ϕ_y on $\mathcal{W}$, as defined in (8), the maximization on $\mathcal{W}$ of progressively better piecewise linear approximations ϕ_y^l of ϕ_y, given by (11), is considered. The problem of maximizing ϕ_y^l on $\mathcal{W}$ is formulated as a linear programming problem in the algorithm below.

Algorithm A_1

Step 0: Choose $w^1 \in \mathcal{W}$ and set $l := 1$;

Step 1: Solve the convex programming problem

$$(P_W)\left|\begin{array}{l}\text{minimize } \langle w^l, f(x)\rangle \\ \text{subject to } \quad x \in \Omega,\end{array}\right.$$

obtaining $x(w^l)$;

Step 2: Solve the linear programming problem in the variables w and σ

$$(P_L)\left|\begin{array}{ll}\text{maximize} & \sigma \\ \text{subject to} & \sigma \leq \langle w, f(x(w^i)) - y\rangle, \quad i = 1, 2, \ldots, l, \\ & w \in \mathcal{W},\ \sigma \in \mathbb{R},\end{array}\right.$$

obtaining σ^{l+1}, w^{l+1} and $\phi_y(w^{l+1})$. If $\sigma^{l+1} - \phi_y(w^{l+1}) < \epsilon_1$, where $\epsilon_1 > 0$ is a small tolerance, declare $\theta(y) = \sigma^{l+1}$ and stop. Otherwise, set $l := l + 1$ and return to Step 1.

3.2 Master problem

Define an initial polytope $\mathcal{P}^0 := \{y \in \mathbb{R}^m \ : \ 0 < \underline{y} \leq y \leq \overline{y}\}$ containing the optimal set of problem $(P_{\mathcal{F}})$. (The problem of finding suitable vectors $\underline{y}$ and $\overline{y}$ is discussed in Sect. 3.3.) The previous assumptions about the multiplicative problem imply that the minimum of u on $\mathcal{P}^0$ is attained at $y^0 = \underline{y}$. By using Algorithm A_1 we will generally conclude that $\theta(y^0) > 0$, that is, $y^0 \notin \mathcal{F}$. At convergence, Algorithm A_1 provides the positive half-space

$$\mathcal{H}_+^0 := \{y \in \mathbb{R}^m \ : \ \langle w^0, y\rangle \geq \langle w^0, f(x(w^0))\rangle\}, \tag{12}$$

which supports $\mathcal{F}$ at $y = f(x(w^0))$ (Theorem 5). Viewing $\mathcal{F}$ as the semi-infinite inequality system (7), the vector y^0 violates mostly $\mathcal{H}^0_+$, since w^0 maximizes the left-hand side of (7). We then say that the hyperplane associated with $\mathcal{H}^0_+$ produces a *deepest cut* in $\mathcal{P}^0$, in the sense that the intersection $\mathcal{H}^0_+ \cap \mathcal{P}^0$ removes as much as possible of $\mathcal{P}^0$. This procedure recalls the cutting plane approach in linear semi-infinite programming.

Proceeding similarly, the problem to be solved at an arbitrary iteration k would be

$$(P_{\mathcal{P}^k}) \left| \begin{array}{l} \text{minimize } u(y) \\ \text{subject to} \quad y \in \mathcal{P}^k. \end{array} \right.$$

Given that u is a continuous quasiconcave function on $\mathbb{R}^m_{++}$ and $\mathcal{P}^k$ is a compact polytopic subset of $\mathbb{R}^m_{++}$, the global minimum of $(P_{\mathcal{P}^k})$ is attained at a vertex of $\mathcal{P}^k$ [10]. The number of vertices of $\mathcal{P}^k$ is linked to m. When m is small, which is often true in practice [17], problem $(P_{\mathcal{P}^k})$ can be solved by using vertex enumeration procedures. The procedure adopted in this paper is the following. Initially, $\mathcal{P}^0$ has 2^m vertices and the solution of $(P_{\mathcal{P}^0})$ is obviously the vertex $y^0 = \underline{y}$. At an arbitrary iteration k, the vertices of

$$\mathcal{P}^k := \mathcal{P}^k \cap \mathcal{H}^k_+, \tag{13}$$

denoted as $\mathcal{V}(\mathcal{P}^k)$, are determined by using the Adjacency List Algorithm [5]. Our implementation of this algorithm can overcome degeneracy problems, discussed in details in [10]. Then any optimal solution of

$$\left| \begin{array}{l} \text{minimize } u(y) \\ \text{subject to} \quad y \in \mathcal{V}(\mathcal{P}^k) \end{array} \right. \tag{14}$$

globally solves $(P_{\mathcal{P}^k})$.

3.3 Global algorithm

In this section we formalize a global optimization algorithm for solving convex multiplicative programming problems and discuss some of its characteristics.

Algorithm A_2

Step 0: Find $\mathcal{P}^0$ and set $k := 0$;

Step 1: Solve the multiplicative problem

$$(P_{\mathcal{P}^k}) \left| \begin{array}{l} \text{minimize } u(y) \\ \text{subject to} \quad y \in \mathcal{P}^k, \end{array} \right.$$

obtaining y^k;

Step 2: Find $\theta(y^k) = \langle w^k, f(x(w^k)) - y^k \rangle$ by using algorithm A_1. If $\theta(y^k) < \epsilon_2$, where $\epsilon_2 > 0$ is a small tolerance, stop: y^k and $x(w^k)$ are ϵ_2-optimal solutions of $(P_{\mathcal{F}})$ and (P_M), respectively. Otherwise, define

$$\mathcal{P}^{k+1} := \{y \in \mathcal{P}^k \ : \ \langle w^k, y\rangle \geq \langle w^k, f(x(w^k))\rangle\},$$

set $k := k + 1$ and return to Step 1.

Assuming that a global minimum is always determined at Step 1 of Algorithm A_2, the infinite convergence of Algorithm A_2 to a global minimum of $(P_{\mathcal{F}})$ (and hence, of (P_M)) can be established as follows.

Theorem 7 *Any limit point $y^\star$ of the sequence $\{y^k\}$ generated by algorithm A_2 is an optimal solution of the convex multiplicative problem $(P_{\mathcal{F}})$.*

Proof Note that problem $(P_{\mathcal{P}^k})$ always has an optimal solution. At any iteration k, the last linear inequality incorporated into $\mathcal{P}^k$ is

$$\langle w^k, y - f(x^k)\rangle \geq 0, \tag{15}$$

and can be rewritten as

$$\begin{aligned}\langle w^k, y - y^k\rangle &\geq \langle w^k, f(x^k) - y^k\rangle \\ &= \theta(y^k).\end{aligned}$$

At any subsequent iteration $p > k$ of algorithm A_2, we must have

$$\begin{aligned}\theta(y^k) &\leq \langle w^k, y^p - y^k\rangle \\ &\leq \|w^k\|\,\|y^p - y^k\| \\ &\leq \|y^p - y^k\|,\end{aligned}$$

because $\|w^k\| \leq 1$ for all $w^k \in \mathcal{W}$. As $k \to \infty$, we obtain $y^k \to y^\star$, $y^p \to y^\star$ and the continuity of θ at $y^\star$ yields $\theta(y^\star) \leq 0$. Therefore, $y^\star \in \mathcal{F}$, that is, $y^\star$ is a feasible solution of $(P_{\mathcal{F}})$. Denoting by $u^\star$ the optimal value of $(P_{\mathcal{F}})$, and knowing that $\mathcal{P}^k$ contains the optimal set of $(P_{\mathcal{F}})$ for all $k = 0, 1, 2, \ldots$, we conclude that $u^\star \geq u(y^\star)$. Consequently, $y^\star$ is an optimal solution of $(P_{\mathcal{F}})$. $\square$

The algorithm initiates with a polytope $\mathcal{P}^0 = \{y \in \mathbb{R}^m \ : \ 0 < \underline{y} \leq y \leq \overline{y}\}$ containing the optimal set of $P_{\mathcal{F}}$. Defining $\underline{y}$ as

$$\underline{y}_i := \min_{x \in \Omega} f_i(x), \ i = 1, 2, \ldots, m, \tag{16}$$

at the cost of solving m additional convex programming problems, has contributed to speeding up the convergence of the algorithm. The upper bound $\overline{y}$ can be also defined as the individual maxima of the convex functions $f_1, f_2, \ldots, f_m$ on Ω, which demands the solution of m convex maximization problems. In practice, the characteristics of the particular multiplicative problem at hand usually suggest a vector $\overline{y}$ large enough to guarantee that all efficient solutions of (P_M) will be contained in $\mathcal{P}^0$. Analytical approaches for confining the optimal set of convex multiplicative programming problems to hyper-rectangles are discussed in [1] and [17].

Most of the computational effort required by algorithm A_2 is concentrated at Step 2, where $\theta(y^k)$ is computed by algorithm A_1. While the linear programming minimizations (Step 2 of A_1) are relatively inexpensive, the nonlinear ones (Step 1 of A_1) demand some effort, although their convexity enable the use of very efficient convex programming algorithms. The codification and preparation efforts related to the approach proposed (Algorithms A_1 and A_2) seem to be small compared with other approaches for convex multiplicative programming available in the literature.

4 Computational experiments

Consider the illustrative example discussed in [1], where an alternative algorithm for convex multiplicative problems combining branch and bound and outer approximation techniques is proposed. The data involved are: $n = m = p = 2$,

$$f_1(x) = (x_1 - 2)^2 + 1, \quad f_2(x) = (x_2 - 4)^2 + 1,$$
$$g_1(x) = 25x_1^2 + 4x_2^2 - 100, \quad g_2(x) = x_1 + 2x_2 - 4.$$

Letting $\underline{y} = (1, 1)$, $\overline{y} = (18, 38)$ (as in [1]), we obtained the results reported in Table 1.

With a convergence criterion equivalent to $\epsilon_2 = 0.025$, the algorithm proposed in [1] converged after 8 iterations. Algorithm A_2 converged after only 5 iterations to the ϵ_2-global solution $x^4 = (1.9009, 1.0495)$ satisfying $\epsilon_2 < 0.01$. The optimal multiplicative function value was $f_1(x^4) f_2(x^4) = 9.8008$. As expected, x^4 is an efficient solution for the associated convex bi-objective problem, as both components of w^4 are positive. Indeed, all the intermediate solutions generated by algorithm A_2 are efficient.

The main objective of this section is to compare the computational performance of the proposed algorithm with those exhibited by the alternative convex multiplivative programming algorithms proposed in [17], where an equivalent concave minimization problem is obtained through a suitable transformation and solved by a cutting plane algorithm, and in [23], where global optimization algorithms for affine and generalized affine multiplicative problems based on branch and bound schemes are developed and extensively compared with other global optimization algorithms. All the algorithms were evaluated with basis on linear multiplicative programming problems of the following form:

$$(P_{ML}) \left| \begin{array}{l} \text{minimize} \displaystyle\prod_{i=1}^{m} \langle c^i, x \rangle \\ \text{subject to} \quad Ax \geq b, \quad x \in \mathbb{R}^n_+, \end{array} \right.$$

where $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ and $c^i \in \mathbb{R}^n$ are constant matrices with entries pseudo-randomly generated in the interval [0, 100].

The proposed algorithm was coded in MATLAB (V. 6.1)/Optimization Toolbox (V. 2.1.1) [20]; a personal computer—Pentium IV, 2.4 GHz, 512 MB RAM—was used for obtaining numerical results. The tolerances for convergence of the proposed algorithm were fixed at $\epsilon_1 = 10^{-4}$ (Algorithm A_1) and $\epsilon_2 = 10^{-5}$ (Algorithm A_2), the latter being the same adopted by the alternative algorithms proposed in [17] and [23].

The performance of the proposed algorithm was quantified in terms of the following parameters (total values): W, number of problems (P_W) solved, C, number of cutting planes

Table 1 Convergence of algorithm A_2

k	y^k	w^k	$x(w^k)$	$\Theta(y^k)$
0	(1.0000,1.0000)	(0.4074,0.5926)	(0.0000,2.0000)	4.0000
1	(1.0000,7.7500)	(0.6585,0.3415)	(1.3547,1.3226)	0.4170
2	(1.0000,8.9711)	(0.8129,0.1871)	(1.7014,1.1493)	0.1016
3	(1.0000,9.5139)	(0.8907,0.1093)	(1.8509,1.0745)	0.0247
4	(1.0000,9.7394)	(0.9451,0.0549)	(1.9009,1.0495)	0.0074

needed for convergence, and V, number of vertices generated in the process. The parameter W was introduced in order to establish approximate comparisons with results of [17]. Ten problems for selected combinations of n (number of variables) and p (number of constraints) were solved. Average and standard deviation values (in parenthesis) of C, W and V are presented. The symbol $\star$ in Tables 2, 3, 4, 5 and 6 means that the required information is not provided in [17] or [23].

Table 2 reports the results obtained with the algorithms proposed in [17]) and in this paper for produts of four linear functions and selected values of n and p. The average values of W obtained with the proposed algorithm are smaller and exhibit slower growth. More importantly, the average values of V obtained with the proposed algorithm are much smaller than

Table 2 Average (standard deviation) values of W, C and V for $m = 4$

(n, p)	Algorithm of [17]		Proposed		
	W	V	W	C	V
(30,20)	62.8 (12.66)	733.2 (207.12)	42.7 (5.56)	8.7 (1.25)	39.2 (8.90)
(40,50)	77.9 (21.60)	983.7 (365.13)	49.3 (5.56)	9.1 (2.07)	38.3 (8.92)
(60,50)	81.9 (11.41)	1060.6 (199.37)	54.6 (7.24)	9.5 (1.51)	39.6 (8.25)
(80,60)	86.8 (15.09)	1153.8 (258.10)	52.9 (6.45)	8.6 (0.96)	39.5 (7.55)
(100,80)	100.1 (17.84)	1386.0 (311.34)	56.4 (7.47)	8.9 (1.44)	42.1 (8.55)
(100,100)	101.5(24.62)	1414.7 (422.30)	56.7 (8.56)	8.8 (1.62)	43.8 (12.53)
(120,100)	98.5 (13.68)	1370.6 (251.71)	63.3 (8.99)	10.0 (2.66)	45.7 (6.79)
(120,120)	99.8 (18.65)	1385.3 (327.60)	62.7 (7.87)	10.4 (2.91)	47.5 (12.74)
(200,200)	$\star$	$\star$	70.5 (5.36)	10.4 (2.71)	51.7 (8.69)

Table 3 Average (standard deviation) computing times for $m = 4$

n	30	40	60	80	100	100	120	120	200
p	20	50	50	60	80	100	100	120	200
Algorithm of [17]	14.21	49.05	95.05	155.10	330.55	524.49	617.51	1154.83	$\star$
	(10.46)	(46.44)	(32.49)	(66.54)	(101.87)	(210.27)	(141.65)	(381.51)	$\star$
Algorithm of [23]	2.6	10.4	13.6	28.1	56.1	61.0	86.1	94.2	396.3
	(0.8)	(4.0)	(5.1)	(6.3)	(17.2)	(21.1)	(35.9)	(23.3)	(189.4)
Proposed	1.55	4.95	11.33	20.57	35.95	38.54	61.29	63.86	257.39
	(0.25)	(0.84)	(1.69)	(2.95)	(4.70)	(7.83)	(8.51)	(8.42)	(57.46)

Table 4 Growths of computing times requirements for $m = 4$

	$r_{40,50}$	$r_{60,50}$	$r_{80,60}$	$r_{100,80}$	$r_{100,100}$	$r_{120,100}$	$r_{120,120}$	$r_{200,200}$
Algorithm of [17]	3.5	6.7	10.9	23.3	36.9	43.5	81.3	$\star$
Algorithm of [23]	4.0	5.2	10.8	21.6	23.5	33.1	36.2	152.4
Proposed	3.2	7.3	13.3	23.2	24.9	39.5	41.2	166.1

Table 5 Average (standard deviation) computing times ($n = 30$, $p = 20$)

m	Algorithm of [17]	Algorithm of [23]	Proposed
2	0.46 (0.05)	0.3 (0.1)	0.56 (0.12)
3	1.27 (0.25)	0.8 (0.3)	1.57 (0.85)
4	14.21 (10.26)	2.6 (0.8)	2.97 (1.73)
5	1170.36 (950.53)	6.0 (2.0)	3.41 (1.30)
6	⋆	⋆	9.81 (8.29)
7	⋆	⋆	28.81 (19.49)
8	⋆	⋆	84.39 (25.82)
9	⋆	⋆	488.58 (186.15)
10	⋆	⋆	2370.15 (1057.05)

Table 6 Growths of computing times requirements ($n = 30$, $p = 20$)

	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
Algorithm of [17]	2.8	30.9	2544.3	⋆	⋆	⋆	⋆	⋆
Algorithm of [23]	2.7	8.7	20.0	⋆	⋆	⋆	⋆	⋆
Proposed	2.8	5.3	6.1	17.5	51.4	150.7	872.5	4232.4

those obtained with the algorithm of [17]. The main reason is that, while in [17] V is an increasing function of W, according to the method proposed in this paper, V increases with C, the number of (deepest) cuts generated by Algorithm A_2, which is substantially smaller than W. Another reason is the use of the Adjacency List Algorithm for vertex enumeration.

The instances of problem (P_{ML}) reported in Table 2 are also solved in [23]. Table 3 furnishes the average and standard deviation times (in sec) obtained with the algorithms proposed in [17,23] and in the present paper. Since the results of Table 3 were obtained by using different computational resources, the following relative performance measure suggested in [23] is adopted:

$$r_{i,j} := \frac{\text{average time for } n = i \text{ and } p = j}{\text{average time for } n = 30 \text{ and } p = 20}.$$

The growths of the computing times requirements of the algorithms as measured by $r_{i,j}$, $(i, j) = (40, 50), (60, 50), (80, 60), (100, 80), (100, 100), (120, 100), (120, 120), (200, 200)$, are presented in Table 4. The growth of computational requirements of the proposed algorithm is slightly faster than that exhibited by the algorithm of [23], which in turn is much slower than that presented by the algorithm of [17]. Additional computational experiences indicate that similar conclusions to those drawn from Tables 2, 3 and 4 can be reached when any other fixed value of m is considered.

The critical parameter for evaluating the performance of multiplicative programming algorithms is the number of functions (m) that compose the overall objective. Table 5 furnishes the average and standard deviation times (in sec) of the algorithms proposed in [17,23] and in the present paper as a function of m and $(n, p) = (30, 20)$. Results for products of more than five linear functions are not reported in [17] and [23].

Table 6 reports the growths of the computing times requirements of the algorithms as measured by r_i, where

$$r_i := \frac{\text{average time for } m = i}{\text{average time for } m = 2}, \quad i = 3, 4, \ldots, 10.$$

As m increases, the growth of the computational requirements of the proposed algorithm is significantly slower than that exhibited by the algorithm of [23], which in turn is much slower than that presented by the algorithm of [17].

As a final remark, it is worth mentioning that as long as the multiplicative function is a product of convex functions, problem (P_W) (Algorithm A_1) will be a convex programming problem, for which very efficient solvers are available. This characteristic of problem (P_W) has been taken into account while implementing Algorithm A_1.

5 Conclusions

A global optimization approach for convex multiplicative programming inspired in elements of convex analysis has been proposed in this paper. Some properties related to the concept of efficient solution in multiobjective programming have been used to derive progressively better outer approximations of the problem in the outcome space. Convex analysis results have been employed to decompose the convex multiplicative problem into a master, quasiconcave problem in the outcome space, globally solved by vertex enumeration, and a maximin subproblem, which tests the global solution generated by the master problem with respect to its feasibility.

Numerical experiments have shown that the computational effort invested in generating deepest cuts in the outcome space through the solution of maximin subproblems is compensated by a faster convergence of the algorithm. The use of deepest cuts also limits the growth of vertices in the master problem and enables its effective solution by vertex enumeration. An Adjacency List Algorithm has been implemented to accomplish this task.

The algorithms derived are easily programmed by using standard optimization packages. Further properties of the approach proposed as well as its extension to more general multiplicative and fractional global optimization problems are under current investigation.

Acknowledgements This work was partially sponsored by grants from CNPq and FAPESP, Brazil.

References

1. Benson, H.P.: An outcome space branch and bound-outer approximation algorithm for convex multiplicative programming. J. Global Optim. **15**, 315–342 (1999)
2. Benson, H.P., Boger, G.M.: Multiplicative programming problems: analysis and efficient point search heuristic. J. Optim. Theory Appl. **94**, 487–510 (1997)
3. Benson, H.P., Boger, G.M.: Outcome-space cutting-plane algorithm for linear multiplicative programming. J. Optim. Theory Appl. **104**, 301–322 (2000)
4. Cambini, R., Sodini, C.: A finite algorithm for a class of nonlinear multiplicative programs. J. Global Optim. **26**, 279–296 (2001)
5. Chen, P.C., Hansen, P., Jaumard, B.: On-line and off-line vertex enumeration by adjacency lists. Opers. Res. Lett. **10**, 403–409 (1991)
6. Ferreira, P.A.V., Machado, M.E.S.: Solving multiple objective problems in the objective space. J. Optim. Theory Appl. **89**, 659–680 (1996)
7. Floudas, C.A., Visweswaram, V.: A primal-relaxed dual global optimization approach. J. Optim. Theory Appl. **78**, 187–225 (1993)

8. Geoffrion, A.M.: Solving bicriterion mathematical programs. Oper. Res. **15**, 39–54 (1967)
9. Geoffrion, A.M.: Generalized benders decomposition. J. Optim. Theory Appl. **10**, 237–260 (1972)
10. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization. Kluwer Academic Publishers, Netherlands (1995)
11. Jaumard, B., Meyer, C., Tuy, H.: Generalized convex multiplicative programming via quasiconcave minimization. J. Global Optim. **10**, 229–256 (1997)
12. Konno, H., Kuno, T.: Linear multiplicative programming. Math. Program. 56, 51–84 (1992)
13. Konno, H., Kuno, T.: Multiplicative programming problems. In: Horst, R., Pardalos, P.M. (eds.), Handbook of Global Optimization, pp. 369–405. Kluwer Academic Publishers, Netherlands (1995)
14. Konno, H., Kuno, T., Yajima, Y.: Global minimization of a generalized convex multiplicative function. J. Global Optim. **4**, 47–62 (1994)
15. Katoh, N., Ibaraki, T.: A parametric characterization and an ϵ-approximation scheme for the minimization of a quasiconcave program. Discrete Appl. Math. **17**, 39–66 (1987)
16. Kuno, T.: A finite branch-and-bound algorithm for linear multiplicative programming. Comput. Optim. Appl. **20**, 119–135 (2001)
17. Kuno, T., Yajima, Y., Konno, H.: An outer approximation method for minimizing the product of several convex functions on a convex set. J. Global Optim. **3**, 325–335 (1993)
18. Lasdon, L.S.: Optimization Theory for Large Systems. MacMillan Publishing Co., New York (1970)
19. Liu, X.J., Umegaki, T., Yamamoto, Y.: Heuristic methods for linear multiplicative programming. J. Global Optim. **15**, 433–447 (1999)
20. MATLAB, User's Guide, The MathWorks Inc., http://www.mathworks.com/
21. Rockafellar, R.T.: Convex Analysis. Princeton University Press, New Jersey (1970)
22. Ryoo, H.S., Sahinidis, N.V.: Analysis of bounds for multilinear functions. J Global Optim. **19**, 403–424 (2001)
23. Ryoo, H.S., Sahinidis, N.V.: Global optimization of multiplicative problems. J. Global Optim. **26**, 387–418 (2003)
24. Scott, C.H., Jefferson, T.R.: On duality for a class of quasiconcave multiplicative problems. J. Optim. Theory Appl. **117**, 575–583 (2003)
25. Thoai, N.V.: A global optimization approach for solving the convex multiplicative programming problem. J. Optim. Theory Appl. **1**, 341–357 (1991)
26. Thoai, N.V.: Convergence and application of a decomposition method using duality bounds for nonconvex global optimization. J. Optim. Theory Appl. **133**, 165–193 (2002)
27. Yu, P.-L.: Multiple-Criteria Decision Making. Plenum Press, New York (1985)