A BRANCH–AND–CUT ALGORITHM FOR CONVEX MULTIPLICATIVE PROGRAMMING

Alireza Mohebi Ashtiani
Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Matemática – DAMAT
Estrada dos Pioneiros, 3131, Jd. Morumbi
86036–370, Londrina – PR
ashtiani@utfpr.edu.br

Paulo Augusto Valente Ferreira
Universidade Estadual de Campinas – UNICAMP
Faculdade de Engenharia Elétrica e de Computação – FEEC
Av. Albert Einstein, 400, Barão Geraldo
13083–852, Campinas – SP
valente@dt.fee.unicamp.br

ABSTRACT

This paper addresses the problem of minimizing the product of $p \geq 2$ convex positive functions over a convex set. Linear multiplicative programming problems and linear–multiplicative–fractional programming problems fall into this category as two special cases, with important applications in various areas. As the objective function is not convex (or quasi–convex), in general, the problem may have local optimal solutions that are not global optimal solutions. A global optimization algorithm based on a suitable reformulation of the problem in the outcome space is proposed. Global minimizers are obtained as the optimal solutions of a sequence of special convex programs, coordinated a rectangular branch–and–bound procedure. Computational experiences demonstrate the efficiency of the proposed algorithm.

KEYWORDS. Global Optimization, Multiplicative Programming, Branch–and–Cut.

Paper topics (Mathematical Programming)

RESUMO

Este artigo aborda o problema de minimizar o produto de $p \geq 2$ funções convexas positivas em um conjunto convexo. Problemas de programação multiplicativa linear e problemas de programação multiplicativa–fracionária se enquadram nesta categoria como dois casos especiais com várias aplicações importantes em diversas áreas. Como a função objetivo não é mais convexa (ou quase–convexa), em geral, o problema pode ter soluções locais que não são globais. Um algoritmo de otimização global baseado em uma reformulação adequada do problema no espaço das funções é proposto. Minimizadores globais são encontrados como soluções ótimas de uma sequência de problemas convexos especiais, coordenada por um algoritmo *branch–and–bound* retangular. Experiências computacionais demonstram a eficiência do algoritmo proposto.

PALAVRAS CHAVE. Otimização Global, Programação Multiplicativa, Branch–and–Cut.

Tópicos (Programação Matemática)

1. Introduction

Many practical problems in Engineering, Economics and Planning are modeled in a convenient way as global optimization problems. The main purpose of this paper is to introduce a new global optimization technique for globally solving a special class of optimization problems, namely, minimization of a product of convex functions on a convex set. This class includes the Linear–Multiplicative (LMP) and Multiplicative–Fractional Programs (MFP), as special cases.

Problems of the following forms are considered:

$$(P_M) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\prod_{i=1}^{p} f_i(x) \\ \text{subject to} & x \in \Omega \end{array} \right.$$

where f_i $(i = 1, 2, \ldots, p)$ are convex functions defined on $\mathbb{R}^n$. It is also assumed that $\Omega \subset \mathbb{R}^n$ is a nonempty compact convex set and that each f_i is positive over Ω. The product of two or more convex positive functions $(p \geq 2)$ need not be convex or quasi–convex, and, therefore, problem (P_M) may have local optimal solutions that are not global optimal solutions. In nonconvex global optimization, problem (P_M) has been referred as the convex multiplicative problem.

Microeconomics, financial optimization, VLSI chip design, decision tree optimization, bond portfolio optimization, layout design, multicriteria optimization problems, robust optimization, data mining–pattern recognition and geometric design are some of the areas where this convex multiplicative programming finds interesting applications (see [Ryoo and Sahindis, 2003]).

Linear–Multiplicative problems assume the more specific form

$$(LMP) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\prod_{i=1}^{p} (c_i^T x + d_i) \\ \text{subject to} & x \in X := \Big\{ x \in \mathbb{R}^n \ : \ -\infty < l \leq x \leq u < +\infty, \ \ Ax \leq b \Big\}, \end{array} \right.$$

where $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$ for $i = 1, 2, \ldots, p$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c_i^T x + d_i > 0$ for $i = 1, 2, \ldots, p$ and all $x \in X$. The case $p = 2$ was originally proposed and studied in [Swarup, 1966]. Problem (LMP) is known to be NP–hard [Matsui, 1996], even when $p = 2$, despite its minimum be achieved at an extreme point of the polytope X. When $p = 2$, problem (LMP) can be solved efficiently in polynomial time [Goyal et al., 2009].

Many global optimization approaches have been proposed for globally solving (LMP) for $p \geq 3$, such as outer approximations methods, vertex enumeration methods, primal–dual simplex methods, cutting plane methods in outcome space, parametrization based methods, branch–and–bound methods, heuristic methods, among others (see [Ryoo and Sahindis, 2003]). All these methods are computationally very expensive, even when $p \ll n$ and the objective function is structurally simple. Very few computational results have been reported for problems with $p > 5$. Computational results reported in the literature of the (linear) multiplicative programming show that the proposed algorithms are all very sensitive to p, even when $p < 5$.

The paper is organized in six sections, as follows. In Section 2, the problem is reformulated in the outcome space, and an outer approximation approach for solving convex multiplicative problems is outlined. In Sections 3 and 4, a branch–and–cut algorithm is derived. Computational experiences with the proposed branch–and–cut algorithm are reported in Section 5. Conclusions are presented in Section 6.

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 $x_i \geq y_i, i = 1, 2, ..., n$.

Accordingly, $x > y$ ($x - y \in \mathbb{R}^n_{++}$) implies $x_i > y_i, i = 1, 2, ..., n$. The standard inner product in $\mathbb{R}^n$ is denoted as $\langle x, y\rangle$. 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

The outcome space approach for solving problem (P_M) is inspired in a combination of approaches introduced in [Oliveira and Ferreira, 2008] and [Ashtiani and Ferreira, 2015] for solving the convex multiplicative and the generalized convex–concave fractional problems, respectively. The objective function in (P_M) can be written as the composition $u(f(x))$, where $u : \mathbb{R}^p \to \mathbb{R}$, defined by

$$u(y) \; := \prod_{i=1}^{p} y_i.$$

The function u can be viewed as a particular aggregating function [Yu, 1985] for the following multiobjective problem:

$$\left|\begin{array}{ll} \text{minimize} & f(x) := \big(f_1(x), f_2(x), \ldots, f_p(x)\big), \\ \text{subject to} & x \in \Omega, \end{array}\right. \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^p$. The image of Ω under f,

$$\mathcal{Y} \; := f(\Omega), \tag{2.2}$$

is the outcome space associated with problem (P_M). Since f is positive over Ω, it follows that u is strictly increasing (and also quasi–concave) over $\mathcal{Y}$ and any optimal solution of (P_M) is Pareto–optimal or efficient. In other words, if $x^\star \in \Omega$ is an optimal solution of the convex multiobjective (multiplicative) problem (P_M), then $x^\star$ is efficient for (2.1) [Geoffrion, 1967]. Since any optimal solution $x^\star \in \Omega$ of problem (P_M) is an efficient solution of problem (2.1), then there exits some p–dimensional vector $w \in \mathcal{W}$, dependent on $x^\star$, such that $x^\star$ also solves the convex minimization problem:

$$(P_{\mathcal{W}}) \quad \left|\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} w_i f_i(x) \\ \text{subject to} & x \in \Omega, \end{array}\right.$$

where $\mathcal{W}$ is the $(p-1)$–dimensional simplex $\mathcal{W} := \left\{w \in \mathbb{R}^p_+ \; : \; \sum_{i=1}^{p} w_i = 1\right\}$. The following theorem (see [Katoh and Ibaraki, 1987] for a proof) characterizes the optimal solution of (P_M) in terms of problem $(P_{\mathcal{W}})$.

Theorem 2.1 *Let $x^\star \in \Omega$ be an optimal solution of (P_M). Then any optimal solution of $(P_{\mathcal{W}})$ for $w = w^\star$, where*

$$w_i^\star = \prod_{j \neq i} f_j(x^\star), \qquad i = 1, 2, \ldots, p,$$

is also optimal to (P_M).

The previous discussion ensures the existence but does not suggests a procedure for determining $w \in \mathcal{W}$ that would lead to an optimal solution of (P_M) through the solution of the convex minimization problem $(P_{\mathcal{W}})$. The global optimization algorithm developed in this paper iteratively finds $w \in \mathcal{W}$ with this property.

The outcome space formulation of problem (P_M) is simply

$$(P_{\mathcal{Y}}) \quad \left| \begin{array}{ll} \text{minimize} & u(y) := \prod_{i=1}^{p} y_i \\ \text{subject to} & y \in \mathcal{Y}. \end{array} \right.$$

where the outcome space $\mathcal{Y}$ is compact, but generally nonconvex. Let effi(Ω) be the set of efficient solutions of (2.1). It can be proved that effi$(\mathcal{Y}) \subset \partial\mathcal{Y}$, where effi$(\mathcal{Y}) = f(\text{effi}(\Omega))$ and $\partial\mathcal{Y}$ is the boundary of $\mathcal{Y}$. Defining

$$\mathcal{F} := \mathcal{Y} + \mathbb{R}^p_+,$$

it can be also proved that $\mathcal{F}$ is convex and effi$(\mathcal{Y}) =$ effi$(\mathcal{F})$. The convexity of $\mathcal{F}$ and the fact that effi$(\mathcal{F}) \subset \partial\mathcal{F}$ imply that $\mathcal{F}$ admits a supporting half–space at each efficient solution of $(P_{\mathcal{Y}})$. A suitable representation for $\mathcal{F}$ is $\mathcal{F} = \left\{ y \in \mathbb{R}^p \ : \ f(x) \leq y \quad \text{for some } x \in \Omega \right\}$, which leads to an equivalent outcome space formulation with a convex and closed feasible region:

$$(P_{\mathcal{F}}) \quad \left| \begin{array}{ll} \text{minimize} & u(y) := \prod_{i=1}^{p} y_i \\ \text{subject to} & y \in \mathcal{F}. \end{array} \right.$$

Theorem 2.2 (Equivalence Theorem) *Let $(x^\star, y^\star)$ be an optimal solution of problem $(P_{\mathcal{F}})$. Then $y^\star \in$ effi$(\mathcal{Y})$, $y^\star = f(x^\star)$ and $x^\star$ and $y^\star$ solve problems (P_M) and $(P_{\mathcal{Y}})$, respectively. Conversely, if $x^\star$ solves (P_M), then $(x^\star, y^\star)$ is an optimal solution for problem $(P_{\mathcal{F}})$, where $y^\star := f(x^\star)$ and $y^\star$ solves also problem $(P_{\mathcal{Y}})$.*

Proof. See [Oliveira and Ferreira, 2008].

In [Oliveira and Ferrira, 2008], the authors explore the fact that $x \in \mathcal{F}$ if and only if y satisfies the semi–infinite inequality system

$$\sum_{i=1}^{p} w_i y_i \geq \min_{x \in \Omega} \sum_{i=1}^{p} w_i f_i(x), \qquad \text{for all } \ w \in \mathcal{W}, \tag{2.3}$$

and any subset of the inequalities (2.3) determines an outer approximation of $\mathcal{F}$. Let $y^L := (y_1^L, y_2^L, \ldots, y_p^L)$ and $y^U := (y_1^U, y_2^U, \ldots, y_p^U)$ be p–dimensional vectors defined as

$$y_i^L = \min_{x \in \Omega} \ f_i(x), \qquad y_i^U = \max_{x \in \Omega} \ f_i(x),$$

for $i = 1, 2, \ldots, p$, such that $0 < y^L \leq y^U$. Define also the rectangle (polytope) $\mathcal{R}^0 := \{ y \in \mathbb{R}^p \ : \ y^L \leq y \leq y^U \}$. Now problem $(P_{\mathcal{F}})$ assumes the following equivalent form

$$(P_{\mathcal{S}}) \quad \left| \begin{array}{ll} \underset{y}{\text{minimize}} & \prod_{i=1}^{p} y_i \\ \text{subject to} & \sum_{i=1}^{p} w_i y_i \geq \min_{x \in \Omega} \sum_{i=1}^{p} w_i f_i(x), \qquad \text{for all } \ w \in \mathcal{W}, \\ & y \in \mathcal{R}^0, \end{array} \right.$$

which consists in minimization a product of elementary functions subject to a linear semi–infinite inequality constraint and to the original bounding constraints. It is worth noting that y satisfies the semi–infinite inequality system (2.3) if and only if $\theta(y) \leq 0$ where

$$\theta(y) \ := \max_{w \in \mathcal{W}} \ \phi_y(w), \tag{2.4}$$

and

$$\phi_y(w) := \min_{x\in\Omega} \sum_{i=1}^{p} w_i(f_i(x) - y_i). \tag{2.5}$$

In fact, the most violated constraint is found by computing $\theta(y)$. Let $(\overline{x}, \overline{w})$ be an optimal solution of the max–min problem (2.4)–(2.5) that defines the value of $\theta(y)$. If $\theta(y) > 0$, then at least the following inequality constraint is violated by $y \in \mathcal{R}^0$:

$$\sum_{i=1}^{p} \overline{w}_i y_i \geq \sum_{i=1}^{p} \overline{w}_i f_i(\overline{x}).$$

3. Cutting Plane Algorithm

Problem $(P_{\mathcal{S}})$ has a small number of variables, but infinitely many linear inequality constraints. An adequate approach for solving $(P_{\mathcal{S}})$ is relaxation: determine a global minimizer y^k over an outer approximation $\mathcal{F}^k$ of $\mathcal{F}$ as a subset of the inequality constraints on the type of (2.3) and then append to $\mathcal{F}^k$ only the inequality constraint most violated by y^k.

Algorithm 1 (Cutting Plane Algorithm)

- **Step 0:** Find $\mathcal{R}^0$ and set $k := 0$;
- **Step 1:** Solve the approximate linear–multiplicative problem

$$(P_{\mathcal{R}}) \quad \left| \begin{array}{lll} \underset{y}{\text{minimize}} & \displaystyle\prod_{i=1}^{p} y_i & \\ \text{subject to} & \displaystyle\sum_{i=1}^{p} w_i^l y_i \geq \min_{x\in\Omega} \sum_{i=1}^{p} w_i^l f_i(x), & l = 0, 1, \ldots, k-1, \\ & y \in \mathcal{R}^0, & \end{array} \right.$$

  obtaining y^k. The l–th most violated inequality constraint is associated to w^l. (Initially, $l = -1$, there is no inequality appended to $(P_{\mathcal{R}})$, and $y^0 = y^L$).
- **Step 2:** Find $\theta(y^k)$, obtaining (x^k, w^k). If $\theta(y^k) \leq 0$, then stop: y^k is an optimal solution of $(P_{\mathcal{S}})$. Otherwise, the most violated constraint is defined by

$$\sum_{i=1}^{p} w_i^k y_i \geq \min_{x\in\Omega} \sum_{i=1}^{p} w_i^k f_i(x).$$

  Set $k = k + 1$ and return to **Step 1**.

Given $y \in \mathbb{R}^p$, it can be proved [Oliveira and Ferreira, 2008] that the function ϕ_y is concave on $\mathcal{W}$ and the function θ is convex (hence continuous) on $\mathbb{R}^p$. The following theorem is very useful for determining $\theta(y^k)$ and w^k.

Theorem 3.1 *For any $y \in \mathbb{R}^p$, $\theta(y)$ is the optimal value of the following convex programming problem*

$$\left| \begin{array}{ll} \underset{x,\sigma}{\text{minimize}} & \sigma \\ \text{subject to} & f(x) \leq \sigma e + y \\ & x \in \Omega, \ \sigma \in \mathbb{R}. \end{array} \right. \tag{3.1}$$

where $\sigma \in \mathbb{R}$ and $e \in \mathbb{R}^p$ is the vector of ones.

Proof. See [Oliveira and Ferreira, 2010].

As a consequence of this theorem, all the information required at Step 2 of Algorithm 1 can be obtained by solving the primal problem (3.1). For y^k, $\theta(y^k)$ corresponds to the optimal value of the convex problem (3.1), and the multipliers associated to the inequality constraints $f(x) \leq \sigma e + y^k$ (which produce the deepest cuts) are the $w^k \in \mathcal{W}$.

Theorem 3.2 *Any limit point $y^\star$ of the sequence $\{y^k\}$ generated by Algorithm 1 is an optimal solution of the convex multiplicative problem (P_M).*

Proof. See [Oliveira and Ferreira, 2008].

It follows that $\theta(y^k) \leq \epsilon_r$ for a sufficient large k, where $\epsilon_r > 0$ is a small tolerance for the finite convergence of Algorithm 1.

4. `rAI` Branch–and–Cut Algorithm

In [Adjiman et al., 1995], the authors discuss a convex lower bound for the bilinear term $y_{i1}y_{i2}$ inside a rectangular region $[y_{i1}^L, y_{i1}^U] \times [y_{i2}^L, y_{i2}^U]$, where $y_{i1}^L, y_{i1}^U, y_{i2}^L$ and y_{i2}^U are the lower and upper bounds on y_{i1} and y_{i2}, respectively. Bilinear terms of the form $y_{i1}y_{i2}$ are underestimated by introducing a new variable y_i and two inequalities,

$$y_i = y_{i1}y_{i2} \geq \max \left\{ \begin{array}{c} y_{i1}y_{i2}^L + y_{i1}^L y_{i2} - y_{i1}^L y_{i2}^L \\ y_{i1}y_{i2}^U + y_{i1}^U y_{i2} - y_{i1}^U y_{i2}^U \end{array} \right\}. \tag{4.1}$$

which depend on the bounds on y_{i1} and y_{i2}. Now, by recursively replacing each bilinear term in the objective function of ($P_{\mathcal{R}}$) with a new variable until it is replaced by a single variable

$$\prod_{i=1}^{p} y_i = \underbrace{\underbrace{\underbrace{\underbrace{\underbrace{y_1 y_2}_{=:y_{p+1}} \, y_3}_{=:y_{p+2}} \, y_4 \cdots}_{\vdots} \, y_{p-1}}_{=:y_{2p-2}} \, y_p}_{=:y_{2p-1}}$$

and using the convex envelope (4.1) for all $i = p+1, p+2, \ldots, 2p-1$ with the bounds on the new variables given by

$$y_i^L := y_{i1}^L y_{i2}^L, \qquad y_i^U := y_{i1}^U y_{i2}^U,$$

for $i = p+1, p+2, \ldots, 2p-1$, a convex envelope for the objective function of ($P_{\mathcal{R}}$) will be available. In [Ryoo, 2001], the authors show that this `rAI` (Recursive Arithmetic Interval) scheme provides tight bounds.

Replacing the elementary linear multiplicative objective function of ($P_{\mathcal{R}}$) by its convex envelope, we obtain the following linear semi–infinite optimization problem:

$$(P_{\mathcal{L}}) \quad \left| \begin{array}{lll} \underset{y}{\text{minimize}} & y_{2p-1} & \\ \text{subject to} & \displaystyle\sum_{i=1}^{p} w_i y_i \geq \min_{x\in\Omega} \sum_{i=1}^{p} w_i f_i(x), & \text{for all } \; w \in \mathcal{W}, \\ & y_i \geq y_{i1}y_{i2}^L + y_{i1}^L y_{i2} - y_{i1}^L y_{i2}^L, & i = p+1, p+2, \ldots, 2p-1, \\ & y_i \geq y_{i1}y_{i2}^U + y_{i1}^U y_{i2} - y_{i1}^U y_{i2}^U, & i = p+1, p+2, \ldots, 2p-1, \\ & y \in \mathcal{R}^0, & \end{array} \right.$$

Let $u^\star(P_M)$, $u^\star(P_\mathcal{S})$ and $u^\star(P_\mathcal{L})$ be the optimal values of problems (P_M), $(P_\mathcal{S})$ and $(P_\mathcal{L})$, respectively. Any feasible point of $(P_\mathcal{L})$ provides an upper bound for the optimal value of $(P_\mathcal{S})$. Furthermore, $u^\star(P_\mathcal{L}) \leq u^\star(P_\mathcal{S}) = u^\star(P_M)$. Hence, problem $(P_\mathcal{S})$ can be solved through a branch–and–bound algorithm, in particular by a rectangular branch–and–bound algorithm. The proposed rectangular branch–and–bound algorithm for globally solving problem $(P_\mathcal{S})$ assumes the structure below.

In rectangular branching–and–bounding algorithms the feasible region of the problem is partitioned into subrectangles. Let $\mathcal{R}$ be a subrectangle of $\mathcal{R}^0$ with bounds $y^L(\mathcal{R})$ e $y^U(\mathcal{R})$ satisfying $y^L(\mathcal{R}) \geq y^L$, $y^U(\mathcal{R}) \leq y^U$, with the understanding that $y^L(\mathcal{R}^0) = y^L$ and $y^U(\mathcal{R}^0) = y^U$.

Let $(P_\mathcal{L}(\mathcal{R}))$ and $(P_\mathcal{S}(\mathcal{R}))$ be problems of the forms $(P_\mathcal{L})$ and $(P_\mathcal{S})$ when $\mathcal{R}^0$ is replaced by $\mathcal{R}$; let $y(\mathcal{R})$ be any optimal solution of $(P_\mathcal{L}(\mathcal{R}))$. The optimal value of $(P_\mathcal{S}(\mathcal{R}))$ is lower than or equal to the upper bound $\mu(\mathcal{R}) := \prod_{i=1}^{p} y_i(\mathcal{R})$, because $y(\mathcal{R})$ is feasible for $P_\mathcal{S}(\mathcal{R})$, and greater than or equal to the lower bound $\gamma(\mathcal{R}) := y_{2p-1}(\mathcal{R})$, because $(P_\mathcal{L}(\mathcal{R}))$ is a understimation of $(P_\mathcal{S}(\mathcal{R}))$.

Algorithm 2 (Branch–and–Bound Algorithm)

- **Step 0:** Solve $(P_\mathcal{L}(\mathcal{R}^0))$, obtaining an optimal solution $y(\mathcal{R}^0)$. Set $\gamma^0 := \gamma(\mathcal{R}^0)$, $\mu^0 := \mu(\mathcal{R}^0)$, $\mathcal{L}^0 := \{\mathcal{R}^0\}$ and $q = 0$;
- **Step 1:** If $\mu^q = \gamma^q$, then stop. The incumbent solution y^q is an optimal solution of $(P_\mathcal{S})$;
- **Step 2:** Find $\mathcal{R} \in \mathcal{L}^q$ such that $\gamma(\mathcal{R}) = \gamma^q$. Bisect $\mathcal{R}$ into subrectangles $\mathcal{R}^I$ and $\mathcal{R}^{II}$ and set

$$\mathcal{L}^{q+1} := (\mathcal{L}^q \backslash \{\mathcal{R}\}) \cup \{\mathcal{R}^I, \mathcal{R}^{II}\}.$$

  Compute $\gamma(\mathcal{R}^I), \mu(\mathcal{R}^I), \gamma(\mathcal{R}^{II})$ and $\mu(\mathcal{R}^{II})$. Eliminate all the subrectangles $\mathcal{R} \in \mathcal{L}^{q+1}$ such that $(P_\mathcal{L}(\mathcal{R}))$ is infeasible or $\mu(\mathcal{R}) < \gamma^q$;
- **Step 3:** Find $\mathcal{R}^\star \in \arg \max_{\mathcal{R} \in \mathcal{L}^{q+1}} \mu(\mathcal{R})$ and set $y^{q+1} := y(\mathcal{R}^\star)$, $\mu^{q+1} := \mu(\mathcal{R}^\star)$, $\gamma^{q+1} := \min_{\mathcal{R} \in \mathcal{L}^{q+1}} \gamma(\mathcal{R})$, $q := q + 1$ and return to **Step 1**.

The branching, bounding and pruning rules of Algorithm 2 fulfill all the conditions required for the convergence of rectangular branch–and–bound algorithms [Horst et al., 1995]. Thus, any accumulation point $y^\star$ of the sequence $\{y^q\}$ generated by Algorithm 2 solves $(P_\mathcal{S})$. Convergence results also guarantee that for q sufficiently large,

$$\mu^q - \gamma^q \leq \epsilon_{bb},$$

where $\epsilon_{bb} > 0$ is a small tolerance for the finite convergence of Algorithm 2.

A decisive feature of the proposed algorithm is the propagation of deepest cuts to subrectangles. Only the additional deepest cuts needed to solve problem $P_\mathcal{S}(\mathcal{R})$ for a given rectangle are generated, which confers a branch–and–cut characteristic of the proposed algorithm. This strategy was responsible for speeding-up the convergence of the algorithm in all computational experiments carried out.

5. Computational Experiments

The computational performance of the global optimization algorithm proposed has been investigated with basis in multiplicative problems selected from the literature. Some numerical experiences are reported below. The number of bisections and the number of deepest cuts generated by the algorithm are the most important parameters analyzed.

Algorithms 1 and 2 were coded in MATLAB (V. 7.1)/Optimization Toolbox (V. 4) and run on a Intel(R) Core(TM)2 Duo system, 2.00 GHz, 2GB RAM, 32 bits.

5.1. Illustrative Examples

In order to illustrate the convergence of the global optimization algorithms proposed, the following examples have been employed.

Example 5.1 *Consider the illustrative problem discussed in* [Gao et al., 2010] *where the problem is globally solved by an alternative algorithm:*

$$\left|\begin{array}{ll} \text{minimize} & (x_1+x_2)(x_1-x_2+7) \\ \text{subject to} & 2x_1+x_2 \leq 14, \quad x_1+x_2 \leq 10, \quad -4x_1+x_2 \leq 0 \\ & 2x_1+x_2 \geq 6, \quad x_1+2x_2 \geq 6, \quad x_1-x_2 \leq 3 \\ & x_1 \leq 5, \quad x_1+x_2 \geq 0, \quad x_1-x_2+7 \geq 0. \end{array}\right.$$

Functions f_1 and f_2 are convex and positive over the feasible convex set, which is compact and nonempty. The lower and upper bounds on $y = (y_1, y_2)$ are $\underline{y} = (4, 1)$ and $\overline{y} = (10, 10)$, respectively. With a convergence criterion equivalent to $\epsilon_{bb} = 0.001$, the algorithm proposed in [Gao et al., 2010] converged after 27 iterations to the ϵ_{bb}–global solution $x^\star = (2.0003, 7.9999)$ with the optimal value equal to $u^\star(P_M) = 10.0042$; the reported CPU time was $10.83s$.

With the same ϵ_{bb}, at Step 0 of the proposed algorithm, Algorithm 2, we obtained $\mu^0 - \gamma^0 = 0.0000$, meaning that the algorithm converged at iteration $q = 0$ without performing a single branching. Our algorithm provided ϵ_{bb}–global solution is $x^\star = (2.0000, 8.0000)$ with the optimal value equal to $u^\star(P_M) = 10.0000$. Only a single deepest cut (introduced in the initial rectangle $\mathcal{R}^0$) was needed; the CPU time was $0.2496s$.

Example 5.2 *As a second illustrative example, consider the linear multiplicative problem, also obtained from* [Gao et al., 2010].*The problem is*

$$\left|\begin{array}{ll} \text{minimize} & \displaystyle\prod_{i=1}^{3}(c_i^T x + d_i) \\ \text{subject to} & x \in X := \left\{x \in \mathbb{R}^{11}_+ \ : \ Ax \leq b\right\}, \end{array}\right.$$

where

$$c_1^T = \begin{pmatrix} 1 & 0 & \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad d_1 = 0,$$
$$c_2^T = \begin{pmatrix} 0 & 1 & \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad d_2 = 0,$$
$$c_3^T = \begin{pmatrix} 0 & 1 & 1 & \frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad d_3 = 0,$$
$$A = \begin{pmatrix} 9 & 9 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 8 & 8 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 7 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 7 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 7 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \ b = \begin{pmatrix} 81 \\ 72 \\ 72 \\ 9 \\ 9 \\ 9 \\ 8 \\ 8 \end{pmatrix}.$$

Functions f_1, f_2 and f_3 are positive over the feasible region Ω. The lower and upper bounds on y are $\underline{y} = (1.1111, 1.1111, 2.1111)$ and $\overline{y} = (7.1944, 7.1944, 13.1667)$, respectively. With $\epsilon_{bb} = 0.001$, after 36 iterations, [Gao et al., 2010] reported the ϵ_{bb}–global solution $x^\star = (0.0002, 0.0001, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)$ with the optimal value equals to $u^\star(P_M) = 2.0e-012$; the reported CPU time was $16.03s$.

With the same ϵ_{bb}, at Step 0 of the proposed algorithm, Algorithm 2, we obtained $\mu^0 - \gamma^0 = 1.3323e-015$, again, meaning that the proposed algorithm converged at iteration $q = 0$ without performing a single branching. At the convergence, the obtained ϵ_{bb}–global solution was $x^\star =$

$(0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)$ with the optimal value equal to $u^\star(P_M) = 0.0000$. As in the previous example, only a single deepest cut (introduced in the initial rectangle $\mathcal{R}^0$) was needed; the CPU time was $0.1872s$.

5.2. Comparative Computational Results

In order to compare the computational performance of the proposed branch–and–cut algorithm with alternative algorithms from the literature ([Kuno et al., 1993], [Ryoo and Sahinidis, 2003] and [Ferreira and Oliveira, 2008]), the following test problem (used by all the authors) was considered:

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

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c^i \in \mathbb{R}^n$ are constant matrices with entries pseudo–randomly generated in the interval $[0,\ 100]$. The tolerances for convergence of the proposed algorithm were fixed at $\epsilon_r = 10^{-5}$ (Algorithm 1) and $\epsilon_{bb} = 10^{-3}$ (Algorithm 2). The following parameters are adopted: W, number of problems ($P_{\mathcal{W}}$) solved, C, number of cutting planes needed for convergence. Ten problems for selected combinations of n (number of variables) and m (number of constraints) were solved. Average and standard deviation values (in parenthesis) of C and W are presented. The symbol $\star$ in Tables 1 , 2, 3, 4 and 5 means that the required information is not provided in [Kuno et al., 1993], [Ferreira and Oliveira, 2008] or [Ryoo and Sahinidis, 2003].

Table 1 reports the results obtained with the algorithms proposed in [Kuno et al., 1993] indicated by [KYK:1993], [Oliveira and Ferreira, 2008] indicated by [OF:2008], and the branch–and–cut algorithm proposed in this paper for products of four ($p = 4$) linear functions and selected values of n and m.

Table 1: Average (standard deviation) values of W, C for $p = 4$.

(n, m)	[KYK:1993]	[OF:2008]		Proposed	
	W	W	C	W	C
(30,20)	62.8 (12.66)	42.7 (5.56)	8.7 (1.25)	21.00 (18.81)	7.90 (3.31)
(40,50)	77.9 (21.60)	49.3 (5.56)	9.1 (2.07)	20.30 (19.14)	8.10 (3.38)
(60,50)	81.9 (11.41)	54.6 (7.24)	9.5 (1.51)	20.80 (21.56)	7.80 (3.05)
(80,60)	86.8 (15.09)	52.9 (6.45)	8.6 (0.96)	40.70 (40.37)	7.90 (2.77)
(100,80)	100.1 (17.84)	56.4 (7.47)	8.9 (1.44)	32.80 (20.81)	9.40 (1.65)
(100,100)	101.5(24.62)	56.7 (8.56)	8.8 (1.62)	45.40 (35.44)	9.50 (3.75)
(120,100)	98.5 (13.68)	63.3 (8.99)	10.0 (2.66)	24.70 (21.66)	10.80 (3.36)
(120,120)	99.8 (18.65)	62.7 (7.87)	10.4 (2.91)	30.90 (27.25)	9.70 (1.89)
(200,200)	$\star$	70.5 (5.36)	10.4 (2.71)	22.90 (8.12)	9.40 (2.37)
(250,200)	$\star$	$\star$	$\star$	21.50 (19.97)	13.50 (2.72)
(250,250)	$\star$	$\star$	$\star$	25.10 (20.66)	12.40 (2.59)

Table 2 reports the average and standard deviation CPU times (in sec) obtained with the algorithms proposed in [Kuno et al., 1993], [Ryoo and Sahinidis, 2003], indicated by [RS2003], [Oliveira and Ferreira, 2008] and in the present paper. Since the results of Table 2 were obtained by using different computational resources, the following relative performance measure suggested in [Ryoo and Sahinidis, 2003] (also in [Oliveira and Ferreira, 2008]) was adopted:

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

Table 3 shows the growth of the computing times requirements of the algorithms as measured by $r_{i,j}$, for $(i, j) = (40, 50)$, $(60, 50)$, $(80, 60)$, $(100, 80)$, $(100, 100)$, $(120, 100)$, $(120, 120)$, $(200, 200)$, $(250, 200)$ and $(250, 250)$. Observe that the growth of computational requirements of the proposed algorithm is much slower than those presented by the algorithms of [Kuno et al., 1993], [Ryoo and Sahinidis, 2003] and [Ferreira and Oliveira, 2008].

Table 2: Average (standard deviation) CPU times for $p = 4$.

n	30	40	60	80	100	100	120	120	200	250	250
m	20	50	50	60	80	100	100	120	200	200	250
[KYK:1993]	14.21	49.05	95.05	155.10	330.55	524.49	617.51	1154.83	$\star$	$\star$	$\star$
	(10.46)	(46.44)	(32.49)	(66.54)	(101.87)	(210.27)	(141.65)	(381.51)	$\star$	$\star$	$\star$
[RS:2003]	2.6	10.4	13.6	28.1	56.1	61.0	86.1	94.2	396.3	$\star$	$\star$
	(0.8)	(4.0)	(5.1)	(6.3)	(17.2)	(21.1)	(35.9)	(23.3)	(189.4)	$\star$	$\star$
[OF:2008]	1.55	4.95	11.33	20.57	35.95	38.54	61.29	63.86	257.39	$\star$	$\star$
	(0.25)	(0.84)	(1.69)	(2.95)	(4.70)	(7.83)	(8.51)	(8.42)	(57.46)	$\star$	$\star$
Proposed	2.73	5.49	8.37	11.91	18.73	20.30	21.31	24.41	61.67	93.03	125.07
	(2.38)	(1.03)	(4.97)	(6.05)	(12.16)	(14.74)	(17.49)	(22.99)	(20.00)	(74.18)	(81.01)

Table 3: Growths of CPU times for $p = 4$.

	$r_{40,50}$	$r_{60,50}$	$r_{80,60}$	$r_{100,80}$	$r_{100,100}$	$r_{120,100}$	$r_{120,120}$	$r_{200,200}$	$r_{250,200}$	$r_{250,250}$
[KYK:1993]	3.5	6.7	10.9	23.3	36.9	43.5	81.3	$\star$	$\star$	$\star$
[RS:2003]	4.0	5.2	10.8	21.6	23.5	33.1	36.2	152.4	$\star$	$\star$
[OF:2008]	3.2	7.3	13.3	23.2	24.9	39.5	41.2	166.1	$\star$	$\star$
Proposed	2.01	3.06	4.36	6.86	7.43	7.80	8.94	22.59	34.08	48.81

Table 4 reports the average and standard deviation CPU times (in sec) of the algorithms proposed in [Kuno et al., 1993], [Ryoo and Sahinidis, 2003], [Ferreira and Oliveira, 2008] and in the present paper, as a function of p and $(n, m) = (30, 20)$. Results for products of more than five linear functions are only reported in [Oliveira and Ferreira, 2008].

Table 4: Average (standard deviation) CPU times ($n = 30$, $m = 20$).

p	[KYK:1993]	[RS:2003]	[OF:2008]	Proposed
2	0.46 (0.05)	0.3 (0.1)	0.56 (0.12)	0.81 (0.53)
3	1.27 (0.25)	0.8 (0.3)	1.57 (0.85)	2.02 (1.67)
4	14.21 (10.26)	2.6 (0.8)	2.97 (1.73)	3.13 (2.92)
5	1170.36 (950.53)	6.0 (2.0)	3.41 (1.30)	7.96 (5.72)
6	$\star$	$\star$	9.81 (8.29)	14.01 (9.67)
7	$\star$	$\star$	28.81 (19.49)	30.94 (18.37)
8	$\star$	$\star$	84.39 (25.82)	68.99 (25.38)

Table 5 reports the growths of the computing times requirements of the algorithms as measured by r_i, where

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

Observe that as p increases, the growth of the computational requirements of the proposed algorithms in [Kuno et al., 1993], [Ryoo and Sahinidis, 2003] and [Ferreira and Oliveira, 2008] grow faster than that of the proposed algorithm in this paper.

Table 5: Growths of CPU times ($n = 30$, $m = 20$).

	r_3	r_4	r_5	r_6	r_7	r_8
[KYK:1993]	2.8	30.9	2544.3	$\star$	$\star$	$\star$
[RS:2003]	2.7	8.7	20.0	$\star$	$\star$	$\star$
[OF:2008]	2.8	5.3	6.1	17.5	51.4	150.7
Proposed	2.49	3.86	9.83	17.30	38.20	85.17

6. Conclusions

In this work we proposed a global optimization approach for convex multiplicative programs. By using convex analysis results, the problem was reformulated in the outcome space as an optimization problem with infinitely many linear inequality constraints, and then solved through a relaxation branch–and–bound algorithm. Experimental results have attested the viability and efficiency of the proposed global optimization algorithm, which is, in addition, easily programmed through standard optimization packages. Extensions of the proposed algorithm to other class of global optimization problems are under current investigation by the authors.

Acknowledgment

This work was partially sponsored by grants from the “Conselho Nacional de Pesquisa e Desenvolvimento” (Universal MCTI/CNPq, Grant No. 459259/2014–8), Brazil.

References

Adjiman, C. S., Dallwig, S., Floudas, C. A. and Neumaier, A. (1998), A global optimization method, αBB, for general twice–differentiable constrained NLPs – I. Theoretical Advances, *Computers and Chemical Engineering*, 22:1137–1158.

Ashtiani, A. M. and Ferreira, P. A. V. (2015), A branch–and–cut algorithm for a class of sum–of–ratios problems, *Applied Mathematics and Computation*, 268:596–608.

Ashtiani, A. M. and Ferreira, P. A. V. (2011), On the solution of generalized multiplicative extremum problems, *Journal of Optimization Theory and Applications*, 49:411–419.

Gao, Y., Wu, G and Ma, W. (2010), A new global optimization approach for convex multiplicative programming, *Applied Mathematics and Computation*, 216:1206–1218.

Geoffrion, A. M. (1967), Solving bicriterion mathematical programs, *Operational Research*, 15:39–54.

Gotal, V., Genc–Kaya, L. and Ravi, R. (2009), An FPTAS for minimizing the product of two non–negative linear cost functions, *Mathematical Programming*, 126:401–405.

Katoh, N., Ibaraki, T. (1987), Aparametric characterization and an ϵ–approximation scheme for the minimization of a quasiconcave program, *Discrete Applied Mathematics*, 17:39–66.

Kuno, T., Yajima, Y. and Konno, H. (1993), An outer approximation method for minimizing the product of several convex functions on a convex set, *Journal of Global Optimization*, 3:325–335.

Matsui, T. (1996), NP–hardness of linear multiplicative programming and related problems, *Journal of Global Optimization*, 9:113–119.

Oliveira, R. M., Ferreira, P. A. V. (2010), An outcome space approach for generalized convex multiplicative programs, *Journal of Global Optimization*, 47:107–118.

Oliveira, R. M. and Ferreira, P. A. V. (2008), A convex analysis approach for convex multiplicative programming, *Journal of Global Optimization*, 41:579–592.

Ryoo, H–S. and Sahinidis, N. V. (2003), Global optimization of multiplicative programs, *Journal of Global Optimization*, 26:387–418.

Swarup, K. (1966), Quadratic Programming, *Cahiers du Centre d'Études de Recherche Opérationnelle*, 8:223–234.

Yu, P. L. (1985), *Multiple-Criteria Decision Marketing*, Plenum Press, New York.