Journal of Global Optimization **15:** 315–342, 1999.

An Outcome Space Branch and Bound-Outer Approximation Algorithm for Convex Multiplicative Programming

HAROLD P. BENSON
University of Florida, College of Business Administration, Department of Decision and Information Sciences, Gainesville, FL 32611-7169, USA

(Received 20 February 1998; accepted in revised form 1 June 1999)

Abstract. This article presents a new global solution algorithm for Convex Multiplicative Programming called the Outcome Space Algorithm. To solve a given convex multiplicative program (P_D), the algorithm solves instead an equivalent quasiconcave minimization problem in the outcome space of the original problem. To help accomplish this, the algorithm uses branching, bounding and outer approximation by polytopes, all in the outcome space of problem (P_D). The algorithm economizes the computations that it requires by working in the outcome space, by avoiding the need to compute new vertices in the outer approximation process, and, except for one convex program per iteration, by requiring for its execution only linear programming techniques and simple algebra.

Key words: Multiplicative programming, Convex multiplicative programming, Global optimization, Outer approximation, Branch and bound, Nonconvex programming

1. Introduction

Consider the Convex Multiplicative Programming problem

$$(P_D) \quad \phi = \min \prod_{j=1}^{p} f_j(x), \quad \text{s.t. } x \in D,$$

where $p \geqslant 2$, $f_j : \Re^n \to \Re$ is a finite, convex function for each $j = 1, 2, \ldots, p$, D is a nonempty, compact convex set in $\Re^n$, and, for each $j = 1, 2, \ldots, p$, $f_j(x) > 0$ for all $x \in D$. Problem (P_D) has a number of important applications in various areas, including, for example, economic analysis [5], bond portfolio optimization [9], VLSI chip design [19], and multiple objective optimization [4]. It is well known that the objective function of problem (P_D) need not be convex on D, and that problem (P_D) generally possesses many local minima that are not global; i.e., problem (P_D) is a global optimization problem [10,28]. Furthermore, problem (P_D) is known to be NP-hard, even in special cases such as when $p = 2$, D is a polyhedron, and f_j is linear for each $j = 1, 2$ [13,22].

To solve cases of problem (P_D) when $p = 2$, quite a large number of exact global solution algorithms can be used. For instance, when D is a polyhedral set and f_1 and f_2 are linear functions, the various parametric simplex-based methods of Konno and Kuno [11,12], Konno et al. [15] and Schaible and Sodini [27] can be used to solve the problem. In addition, for this case two branch and bound algorithms ([24] and [16]) are available, as are the enumeration, discrete approximation, outer approximation and polyhedral annexation algorithms of Pardalos [25], Konno and Kuno [10], Aneja et al. [1] and Tuy and Tam [30], respectively. For the case where D is compact and convex and f_1 and f_2 are convex functions, at least three exact global solution algorithms are available for solving problem (P_D). One of these, an outer approximation method due to Konno et al. [14], applies to the problem of minimizing the sum of k products of two convex functions each, which, when $p = 2$, includes problem (P_D) as a special case. The other two of these algorithms, by Kuno and Konno [17] and Thoai [28], use underestimation and outer approximation, respectively.

Globally solving problem (P_D) for cases where $p > 2$ has been shown empirically to generally require significantly more computational effort than the effort needed for the case where $p = 2$ [8,18,26]. To solve problem (P_D) when $p > 2$, D is polyhedral and f_j, $j = 1, 2, \ldots, p$, are linear, at least three exact gobal solution algorithms and one heuristic algorithm are available. The three exact algorithms are a polyhedral annexation method with dimension reduction due to Tuy [29], a branch and bound, image-space algorithm by Falk and Palocsay [3] and a branch and bound algorithm with range reduction developed by Ryoo and Sahinidis [26]. The heuristic algorithm is an efficient point search algorithm due to Benson and Boger [2].

Three algorithms have been proposed that can solve the Convex Multiplicative Programming problem (P_D) when $p > 2$. Since p is generally much smaller than n, to obtain computational efficiency, these algorithms all essentially work in the outcome space $\Re^p$ rather than in the decision space $\Re^n$ of problem (P_D). The first of these algorithms to be proposed, due to Thoai [28], reduces problem (P_D) to a minimization of a quasiconcave function in $\Re^p$ over a convex set. The resulting problem is then solved by outer approximation. The algorithm of Kuno et al. [18] uses a different transformation to rewrite problem (P_D) as a concave minimization problem in the outcome space $\Re^p$. This concave minimization problem is then solved by an outer approximation technique that is specially adapted to its structure. A third algorithm suitable for problem (P_D) has recently been proposed by Jaumard et al. [8]. Using an extension of the transformation in [18], this algorithm rewrites problem (P_D) as a special quasiconcave minimization problem in outcome space $\Re^p$. A conical branch and bound algorithm involving the solution of $(p + 1)$ nonlinear convex programming problems per iteration is then used to solve this quasiconcave minimization problem.

A brief review of algorithms for solving problem (P_D) can be found in Benson and Boger [2]. For a more comprehensive review, see Konno and Kuno [12].

The purpose of this article is to describe and validate a new exact global solution algorithm that we have developed for solving problem (P_D). Like its predecessors, to enhance its efficiency, the algorithm works essentially in the outcome space $\Re^p$ of the problem. In addition, however, it combines branch and bound with outer approximation in such a way that the new vertices of the polyhedra used in the outer approximation process need not be explicitly calculated. Furthermore, only one nonlinear convex program is solved per iteration of the algorithm.

Section 2 shows how problem (P_D) is converted to a quasiconcave minimization problem in outcome space suitable for solution by the new algorithm. The branching, bounding, and outer approximation operations of the new algorithm are described in Section 3. Section 4 gives a statement of the algorithm and describes its convergence properties. Some key computational issues are discussed in Section 5. In Section 6, an example problem is solved and, in the last section, some concluding remarks are given.

2. Conversion to quasiconcave minimization

In this section, we show how to convert problem (P_D) to a quasiconcave minimization problem (P_Y) in outcome space. The new branch and bound-outer approximation algorithm can be applied to problem (P_Y) in order to globally solve problem (P_D). For a similiar transformation, see [28].

For each $j = 1, 2, \ldots, p$, let $\hat{y}_j \in \Re$ satisfy

$$\hat{y}_j > \max f_j(x), \quad \text{s.t. } x \in D,$$

where $\hat{y}_j < +\infty$, and let $\hat{y}^T = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_p]$. For each $x \in \Re^n$, let $[f(x)]^T = [f_1(x), f_2(x), \ldots, f_p(x)]$, and define the set Y by

$$Y = \{y \in \Re^p | f(x) \leqslant y \leqslant \hat{y} \text{ for some } x \in D\}.$$

It is easy to show that Y is a nonempty, compact convex set in the outcome space $\Re^p$ of problem (P_D). Notice also that the interior of Y, denoted int Y, is nonempty, due to the choice of $\hat{y}$.

Let $g : \Re^p \to \Re$ be defined for each $y \in \Re^p$ by

$$g(y) = \prod_{j=1}^{p} y_j, \tag{1}$$

and consider the outcome space problem

$$(P_Y) \quad \min\ g(y), \quad \text{s.t. } y \in Y.$$

The following results give some properties of problem (P_Y). Let Y_{ex} and ∂Y denote the set of all extreme points of Y and the boundary of Y, respectively.

LEMMA 2.1. *Problem (P_Y) consists of the minimization of a function g that is continuous on $\Re^p$ and quasiconcave on the nonempty, compact convex set Y.*

Proof. The continuity of g on $\Re^p$ follows from (1) and elementary limit results. As observed earlier, it is easy to show that Y is nonempty, compact and convex. From Corollary 2.1 in Benson and Boger [2], since Y is a nonempty convex set and, for each $j = 1, 2, \ldots, p$, the function $h_j(y) = y_j$ is positive and concave on Y, it follows that g is quasiconcave on Y. □

Using Lemma 2.1 and the definition of Y, we obtain the following theorem.

THEOREM 2.1. *Problem (P_Y) has a global optimal solution in Y_{ex}. Any global optimal solution to problem (P_Y) is in ∂Y.*

Proof. From [6], when g is continuous on $\Re^p$ and quasiconcave on the nonempty, compact convex set Y, the global minimum of g over Y is attained at some extreme point of Y. Together with Lemma 2.1, this proves the first statement of the theorem.

To prove the second statement of the theorem, let y^* be a global optimal solution for problem (P_Y), and suppose, to the contrary, that $y^* \notin \partial Y$. Then $y^* \in (\text{int}Y)$, so that we may choose a point $\bar{x} \in D$ such that $f(\bar{x}) < y^* < \hat{y}$. Since $\bar{x} \in D$, $0 < f(\bar{x})$. As a result, if we set $\bar{y} = f(\bar{x})$, it follows that $\bar{y} \in Y$ and

$$\prod_{j=1}^{p} \bar{y}_j < \prod_{j=1}^{p} y_j^*.$$

By the definition (1) of g, this contradicts the fact that y^* is a global optimal solution for problem (P_Y). □

From Theorem 2.1, any global optimal solution to problem (P_Y) must belong to ∂Y. Although, as we shall see below, problem (P_D) is equivalent to problem (P_Y), problem (P_D) need *not* have a global optimal solution on the boundary of D. For instance, let $p = 2$, $D = \{(x_1, x_2) | 0 \leqslant x_j \leqslant 6, \ j = 1, 2\}$,

$$f_1(x_1, x_2) = (x_1 - 2)^2 + 1$$

and

$$f_2(x_1, x_2) = (x_2 - 4)^2 + 1$$

in problem (P_D). Then the unique global optimal solution x^* to problem (P_D) is given by $x^{*T} = (2, 4)$ which does not lie on the boundary of D.

THEOREM 2.2. *Problem (P_D) is equivalent to problem (P_Y) in the following sense: If y^* is a global optimal solution for problem (P_Y), then any $x^* \in D$ such that $f(x^*) \leqslant y^*$ is a global optimal solution for problem (P_D), and $\phi = g(y^*) = \prod_{j=1}^{p} f_j(x^*)$. Conversely, if x^* is a global optimal solution for problem (P_D), then $y^* = f(x^*)$ is a global optimal solution for problem (P_Y), and $\phi = g(y^*) = \prod_{j=1}^{p} f_j(x^*)$.*

Proof. Let y^* be a global optimal solution for problem (P_Y), and let $x^* \in D$ satisfy $f(x^*) \leqslant y^*$. Assume that for some $x \in D, \prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*)$. Then, since $0 < f(x^*) \leqslant y^*$, this implies that

$$\prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*) \leqslant \prod_{j=1}^{p} y_j^*. \tag{2}$$

Let $y = f(x)$. Since $x \in D$, by the choice of $\hat{y}$, it follows that $y \in Y$. From (2), $\prod_{j=1}^{p} y_j < \prod_{j=1}^{p} y_j^*$ is also true. By (1), the latter two statements together contradict the fact that y^* is a global optimal solution to problem (P_Y). Therefore, the assumption that for some $x \in D, \prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*)$ is false, so that x^* is a global optimal solution for problem (P_D) and $\phi = \prod_{j=1}^{p} f_j(x^*)$.

Let $\bar{y} = f(x^*)$. Then, since $\hat{y} > f(x^*)$ and $x^* \in D$, this implies that $\bar{y} \in Y$. If $\bar{y} \leqslant y^*$ and $\bar{y} \neq y^*$ were to hold, then, since $\bar{y} > 0$, by (1), $g(\bar{y}) < g(y^*)$ would hold, and y^* would not be a global optimal solution for problem (P_Y). Since $\bar{y} \leqslant y^*$, this implies that $\bar{y} = y^*$ must hold. By (1), since $\bar{y} = f(x^*)$ and $\phi = \prod_{j=1}^{p} f_j(x^*)$, it follows that $g(y^*) = \phi$.

To show the converse statement, let x^* be a global optimal solution for problem (P_D), and let $y^* = f(x^*)$. Assume that for some $y \in Y, g(y) < g(y^*)$. Since $y \in Y$, we may choose an $x \in D$ such that $f(x) \leqslant y$. Because $x \in D, 0 < f(x)$. As a result, $\prod_{j=1}^{p} f_j(x) \leqslant \prod_{j=1}^{p} y_j$. By (1), since $g(y) < g(y^*)$, this implies that

$$\prod_{j=1}^{p} f_j(x) \leqslant \prod_{j=1}^{p} y_j < \prod_{j=1}^{p} y_j^*.$$

Since $y^* = f(x^*)$, this implies that $\prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*)$. Because $x \in D$, this contradicts that x^* is a global optimal solution for problem (P_D). Therefore, the assumption that $g(y) < g(y^*)$ for some $y \in Y$ is false.

By the definitions of Y and $\hat{y}$, since $x^* \in D$ and $y^* = f(x^*)$, $y^* \in Y$. Combined with the fact that $g(y) \geqslant g(y^*)$ for all $y \in Y$, this implies that y^* is a global optimal solution for problem (P_Y). Moreover, by (1), since $y^* = f(x^*)$ and x^* is a global optimal solution for problem (P_D), $\phi = g(y^*) = \prod_{j=1}^{p} f_j(x^*)$. □

3. Outer approximation, branching, and bounding operations

In addition to the assumptions for problem (P_D) given in the Introduction, we will assume henceforth that

$$D = \{x \in \Re^n | g_i(x) \leqslant 0,\ i = 1, 2, \dots, m\},$$

where, for each $i = 1, 2, \dots, m$, $g_i : \Re^n \to \Re$ is a finite, convex, differentiable function. We will also assume that there exists a point $\alpha \in \Re^n$ that satisfies

$g_i(\alpha) < 0, i = 1, 2, \ldots, m$, and that for each $j = 1, 2, \ldots, p$, the function f_j is differentiable.

To solve problem (P_Y), the branch and bound-outer approximation algorithm performs several key operations in outcome space. This section explains these operations.

3.1. OUTER APPROXIMATION

The algorithm constructs a set of one or more nonempty, decreasing polytopes containing Y that serve as outer approximations to Y. The first such polytope Z^0 is given by the rectangle

$$Z^0 = \{y \in \Re^p | 0 \leqslant y \leqslant \hat{y}\}.$$

At the beginning of a typical iteration $k \geqslant 1$ of the algorithm, we have available from the previous iteration a nonempty polytope $Z^{k-1} \subseteq \Re^p$ that contains Y and a point $y^{k-1} \in Z^t$ for some $t \leqslant k-1$. The point y^{k-1} may or may not belong to Y. It is found as part of the lower bounding operation in iteration $k-1$, as we shall see later.

Assume that $k \geqslant 1$. In iteration k, to construct a polytope $Z^k \subseteq \Re^p$ that satisfies $Y \subseteq Z^k \subseteq Z^{k-1}$, first the convex nonlinear program $(T(y))$ given by

$$\begin{array}{lll} (T(y)) & \min \lambda, & \\ & \text{s.t.} & \\ & f(x) - \lambda(y^I - y) - y \leqslant 0, & \\ & g_i(x) \qquad\qquad\qquad\quad \leqslant 0, & i = 1, 2, \ldots, m, \\ & 0 \qquad\qquad\qquad\qquad\; \leqslant \lambda \leqslant 1, & \end{array}$$

is solved, with y set equal to y^{k-1}, for an optimal solution (x^{k^T}, λ_k), where $y^I \in \Re^p$ is a vector in Y prechosen in the algorithm to satisfy $f(x) < y^I < \hat{y}$ for some $x \in D$. This yields a point $w^k = \lambda_k(y^I - y^{k-1}) + y^{k-1}$. It is easy to show that when $\lambda_k = 0$, $w_k = y^{k-1} \in Y$, and, when $\lambda_k > 0$, $w^k \in \partial Y$ and $y^{k-1} \notin Y$.

When $\lambda_k = 0$, the outer approximating polytope Z^k is given by $Z^k = Z^{k-1}$, i.e., a new outer approximating polytope to Y is *not* constructed in iteration k.

When $\lambda_k > 0$, the algorithm finds a supporting hyperplane E to Y at $w^k \in \partial Y$ of the form

$$E = \{y \in \Re^p | \langle u, y - w^k \rangle = 0\}.$$

Furthermore, the algorithm chooses the value $\hat{u}$ of u in E in such a way that for all $y \in Y$,

$$\langle \hat{u}, y - w^k \rangle \geqslant 0 \tag{3}$$

and

$$\langle \hat{u}, y^{k-1} - w^k \rangle < 0. \tag{4}$$

Subsequently, the new containing polytope Z^k is constructed via the formula

$$Z^k = Z^{k-1} \cap \{y \in \Re^p | \langle \hat{u}, y - w^k \rangle \geqslant 0\}.$$

In this way, Z^k satisfies $Y \subseteq Z^k \subseteq Z^{k-1}$. Notice that $y^{k-1} \notin Z^k$ and that Z^k is obtained from Z^{k-1} by seeking to "cut off" a portion of Z^{k-1} via the inequality (3). For a construction of Z^k of this type for the case where Y is polyhedral, see [28].

To find a vector $\hat{u} \in \Re^p$ that satisfies (3) and (4), the algorithm finds any values for the vectors $u \in \Re^p$ and $v \in \Re^m$ that satisfy the system

$$u, v \geqslant 0, \tag{5}$$

$$\sum_{j \in B} u_j = 1, \tag{6}$$

$$\sum_{j=1}^{p} u_j [\nabla f_j(x^k)] + \sum_{i=1}^{m} v_i [\nabla g_i(x^k)] = 0, \tag{7}$$

$$u_j [f_j(x^k) - w_j^k] = 0, \quad j = 1, 2, \ldots, p, \tag{8}$$

and

$$v_i g_i(x^k) = 0, i = 1, 2, \ldots, m, \tag{9}$$

and sets $\hat{u} = u$, where

$$B = \{j \in \{1, 2, \ldots, p\} | f_j(x^k) = w_j^k\},$$

and, for any function h, ∇h denotes the gradient of h. The next result validates this portion of the outer approximation operation.

THEOREM 3.1. *Assume that $\lambda_k > 0$. Then there exist vectors $u \in \Re^p$ and $v \in \Re^m$, where $u \neq 0$, that satisfy (5)–(9). Furthermore, if $(\hat{u}, \hat{v})$ satisfies (5)–(9), then*
(a) $\hat{u} \neq 0$;
(b) $\hat{u}$ *satisfies (3) for all* $y \in Y$;
(c) $\hat{u}$ *satisfies (4).*

Proof. Consider the convex nonlinear program (TH) given by

$$
\begin{aligned}
(TH) \quad & \min \theta, \\
& \text{s.t.} \\
& f(x) - e\theta - w^k \leqslant 0, \\
& g_i(x) \leqslant 0, \quad i = 1, 2, \ldots, m,
\end{aligned}
$$

where $x \in \Re^n$ and $\theta \in \Re$ are the variables and $e \in \Re^p$ is the vector of ones. By definition of α, we may choose $\bar{\theta} \in \Re$ so that

$$f(\alpha) - e\bar{\theta} - w^k < 0$$

and

$$g_i(\alpha) < 0, \quad i = 1, 2, \dots, m.$$

Since $f_j : \Re^n \to \Re$, $j = 1, 2, \dots, p$, and $g_i : \Re^n \to \Re$, $i = 1, 2, \dots, m$, are finite, convex, differentiable functions, this implies that Slater's Constraint Qualification holds for problem (TH) (e.g., see Mangasarian [20]). Notice also that since (x^{k^T}, λ_k) is an optimal solution to problem $T(y^{k-1})$, where $\lambda_k > 0$, $(x^T, \theta) = (x^{k^T}, 0)$ is an optimal solution for problem (TH).

Using the Kuhn–Tucker–Karush necessary conditions for problem (TH) at $(x^{k^T}, 0)$ [20], it is easy to show that there exist $u \in \Re^p$ and $v \in \Re^m$ such that

$$\sum_{j=1}^{p} u_j = 1 \tag{10}$$

and which together satisfy (5) and (7)–(9). By (5) and (10), for any such vectors, $u \neq 0$ must hold. Furthermore, by (8), (10), the definition of B, and the feasibility of $(x^{k^T}, 0)$ in problem (TH), (6) will also hold. As a result, there exist vectors $u \in \Re^p$ and $v \in \Re^m$, where $u \neq 0$, that together satisfy (5)–(9).

Now suppose that $(\hat{u}, \hat{v})$ satisfies (5)–(9). Notice that since (x^{k^T}, λ_k) is an optimal solution to problem $T(y^{k-1})$ with $\lambda_k > 0$, B is nonempty. By (5) and (6), this implies that $\hat{u} \neq 0$.

For any $x \in \Re^n$, let $g^m(x)$ denote $[g_1(x), g_2(x), \dots, g_m(x)]$ and let $d : \Re^n \to \Re$ be defined for each $x \in \Re^n$ by

$$d(x) = \langle \hat{u}, f(x) \rangle + \langle \hat{v}, g^m(x) \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Then, by (5), since each entry in $f(x)$ and $g^m(x)$ is a finite, convex, differentiable function, d is also finite, convex, and differentiable. From (7), this implies that

$$d(x) \geqslant d(x^k)$$

for any $x \in \Re^n$. From (8) and (9) and the definition of d, this inequality may be equivalently written

$$\langle \hat{u}, f(x) \rangle + \langle \hat{v}, g^m(x) \rangle \geqslant \langle \hat{u}, w^k \rangle. \tag{11}$$

When $x \in D$, $g^m(x) \leqslant 0$. By (5) and (11), this implies that for any $x \in D$,

$$\langle \hat{u}, f(x) \rangle \geqslant \langle \hat{u}, w^k \rangle. \tag{12}$$

Let $\bar{y} \in Y$. Then we may choose a point $\bar{x} \in D$ such that $f(\bar{x}) \leqslant \bar{y}$. By (5), this implies that $\langle \hat{u}, \bar{y} - f(\bar{x})\rangle \geqslant 0$. From (12), since $\bar{x} \in D$, $\langle \hat{u}, f(\bar{x}) - w^k\rangle \geqslant 0$. Adding the inequalities in the previous two sentences, we obtain that $\langle \hat{u}, \bar{y} - w^k\rangle \geqslant 0$. By the choice of $\bar{y}$, this implies that (3) holds for all $y \in Y$.

By definition of y^I, we may choose a point $x^I \in D$ such that $f(x^I) < y^I$. Thus, for $\delta \in \{t \in \Re | 0 < t < 1\}$ sufficiently small, $[f(x^I) - y^I] + \delta[y^I - y^{k-1}] \leqslant 0$. This implies that

$$f(x^I) - (1-\delta)(y^I - y^{k-1}) - y^{k-1} \leqslant 0,$$

so that $(x, \lambda) = (x^I, 1-\delta)$ is a feasible solution for problem $(T(y^{k-1}))$. Therefore, $\lambda_k \leqslant (1-\delta) < 1$.

Define a linear function $\beta : Y \to \Re$ for each $y \in Y$ by

$$\beta(y) = \langle \hat{u}, y - w^k\rangle.$$

By definition of y^I, we may choose a point $x^I \in D$ such that $f(x^I) < y^I < \hat{y}$. As a result, for $\varepsilon > 0$ sufficiently small,

$$f(x^I) \leqslant y^I - \varepsilon\hat{u} < \hat{y},$$

so that $(y^I - \varepsilon\hat{u}) \in Y$. Therefore, since (3) holds for all $y \in Y$,

$$\begin{aligned} 0 &\leqslant \beta(y^I - \varepsilon\hat{u}) \\ &= \langle \hat{u}, y^I - \varepsilon\hat{u} - w^k\rangle \\ &= \langle \hat{u}, y^I - w^k\rangle - \varepsilon\langle \hat{u}, \hat{u}\rangle \\ &= \beta(y^I) - \varepsilon\langle \hat{u}, \hat{u}\rangle. \end{aligned}$$

Since ε and $\langle \hat{u}, \hat{u}\rangle$ are strictly positive, this implies that $\beta(y^I) > 0$.

Because $w^k = \lambda_k y^I + (1-\lambda_k)y^{k-1}$ belongs to Y and β is linear on Y, it follows that

$$\beta(w^k) = \lambda_k\beta(y^I) + (1-\lambda_k)\beta(y^{k-1}). \tag{13}$$

By definition of β, $\beta(w^k) = 0$. Since $0 < \lambda_k < 1$ and $\beta(y^I) > 0$, this, together with (13), implies that

$$\beta(y^{k-1}) = -[\lambda_k\beta(y^I)/(1-\lambda_k)]$$

is negative. As a result, from the definition of β, it follows that (4) holds. □

An efficient linear programming procedure for calculating values $u = \hat{u}$ and $v = \hat{v}$ that together satisfy (5)–(9) will be explained in Section 5.

3.2. BRANCHING

The algorithm performs a branching process in outcome space that iteratively partitions a rectangle H^0 containing Y into subrectangles. This process helps the algorithm identify the location in Y of an optimal solution to problem (P_Y).

To help explain the branching process, we will need the following definition.

DEFINITION 3.1 [7]. Let Q be a subset of $\Re^p$ and let J be a finite set of indices. A set $\{M_j,\ j \in J\}$ of subsets of Q is said to be a *partition* of Q when

$$(a)\ Q = \bigcup_{j \in J} M_j$$

and

$$(b)\ M_i \cap M_j = \partial_r M_i \cap \partial_r M_j \text{ for all } i, j \in J, i \neq j,$$

where $\partial_r M_i$ denotes the (relative) boundary of M_i.

During each iteration of the algorithm, a more refined partition is constructed of a portion of H^0 that contains an optimal solution to problem (P_Y).

The initial rectangle H^0 containing Y in the branching operation is given by

$$H^0 = Z^0 = \{y \in \Re^p | 0 \leqslant y \leqslant \hat{y}\}.$$

Notice that H^0 is p-dimensional, since $0 < \hat{y}$. The initial partition P^0 consists of simply H^0, since at the beginning of the algorithm, we cannot yet exclude any portion of H^0 from consideration. This partition P^0 is given by

$$P^0 = \{H^0\}.$$

Let $k \geqslant 1$. At the beginning of step k of the algorithm, we have available a partition P^{k-1}, consisting of p-dimensional rectangles, of a portion of H^0 that cannot yet be excluded on the basis of it not containing an optimal solution for problem (P_Y). Also available is a member H^{k-1} of P^{k-1} chosen in step $k-1$ for further examination.

During step k, the branching process subdivides H^{k-1} into two p-dimensional subrectangles of equal volume. This subdivision is accomplished by a process called rectangular bisection, which is defined as follows.

DEFINITION 3.2. Let W be a p-dimensional rectangle in $\Re^p$ given by

$$W = \{y \in \Re^p | a \leqslant y \leqslant b\},$$

where $a, b \in \Re^p$ and $a < b$. Suppose that

$$(b_i - a_i) = \max\{(b_j - a_j) | j \in J\},$$

where $J = \{1, 2, \dots, p\}$. Let $m \in \Re^p$ be defined by

$$m_j = \begin{cases} a_j, & \text{if } j \neq i \\ (a_i + b_i)/2, & \text{if } j = i. \end{cases}$$

Then $\{W_1, W_2\}$ is called a *rectangular bisection* of W, where W_1 and W_2 are the p-dimensional rectangles given by

$$W_1 = \{y \in \Re^p | a_j \leqslant y_j \leqslant b_j, \ j \in J \backslash \{i\}, a_i \leqslant y_i \leqslant m_i\}$$

and

$$W_2 = \{y \in \Re^p | a_j \leqslant y_j \leqslant b_j, \ j \in J \backslash \{i\}, m_i \leqslant y_i \leqslant b_i\}.$$

Notice in Definition 3.2 that m is the midpoint of a longest edge of W. From Horst and Tuy [7], the rectangular bisection $\{W_1, W_2\}$ in Definition 3.2 forms a partition of W in the sense of Definition 3.1.

Let $\{\bar{H}, \bar{\bar{H}}\}$ denote the rectangular bisection of H^{k-1} formed by the branching process in step k. Then the partition P^k of the portion $H^0 \backslash F$ of H^0 not yet excluded from consideration is

$$P^k = \{H \in (P^{k-1} \backslash H^{k-1}) | H \notin F\} \bigcup \{H \in \bar{H}, \bar{\bar{H}}\} | H \notin F\}, \tag{14}$$

where F denotes the current set of rectangles that have been eliminated from consideration as possible regions for containing a global optimal solution to problem (P_Y). We will see how F is derived and maintained in the next subsection.

For purposes of analyzing convergence, we need the following definition and theorem (cf. Horst and Tuy [7]).

DEFINITION 3.3. Let W be a subset of $\Re^p$, and let $\{P^k\}$ denote a sequence of partitions of W such that for each k, each element of P^{k+1} is formed by partitioning some element of P^k. Let $\{M^k\}$ denote an arbitrary sequence of partition elements such that for each k, $M^k \in P^k$ and M^{k+1} is a strict subset of M^k. The set of partitions $\{P^1, P^2, \dots\}$ is called *exhaustive* when $\lim_k M^k = \cap_k M^k = \{q\}$ for some single point $q \in \Re^p$.

THEOREM 3.2. *Let W be a p-dimensional rectangle in $\Re^p$. Let $\{P^k\}$ denote a sequence of partitions of W such that, for each $k = 1, 2, \dots$, each element of P^{k+1} is formed by creating a rectangular bisection of some element of P^k. Then the set $\{P^1, P^2, \dots\}$ is exhaustive.*

3.3. BOUNDING

Three types of bounds are computed by the algorithm. The first is a local lower bound for the objective function g of problem (P_Y) over $(Y \cap H)$, where H is a given p-dimensional rectangle in $\Re^p$. This local lower bound is a number $LB(H)$ that satisfies

$$LB(H) \leqslant g(y) \quad \text{for all } y \in (Y \cap H).$$

In the algorithm, $LB(H)$ is computed for $H = H^0$ and for each rectangle $\hat{H}$ formed via rectangular bisection of H^{k-1} for each $k = 1, 2, \ldots$ (cf. Section 3.2).

Recall from Subsection 3.2 that

$$H^0 = \{y \in \Re^p | 0 \leqslant y \leqslant \hat{y}\},$$

and $Y \subseteq H^0$. Therefore, a simple lower bound for g over $(Y \cap H^0)$ is given by 0. In the algorithm, $LB(H^0)$ is set equal to 0.

Let k satisfy $k \geqslant 1$. In step k of the algorithm, H^{k-1} is subdivided into two rectangles via rectangular bisection. We may assume that each of these rectangles is of the form

$$\hat{H} = \{h \in \Re^p | h^1 \leqslant h \leqslant h^2\},$$

where $0 \leqslant h^1 < h^2$. The lower bound $LB(\hat{H})$ for g over $(Y \cap \hat{H})$ computed by the algorithm is given by

$$LB(\hat{H}) = \max\{\widehat{LB}(\hat{H}),\ LB(H^{k-1})\}, \tag{15}$$

where the rule for finding $\widehat{LB}(\hat{H})$ depends upon whether or not $h^1 \in (Z^k \cap \hat{H})$.

If $h^1 \in (Z^k \cap \hat{H})$, then $\widehat{LB}(\hat{H})$ is given by

$$\widehat{LB}(\hat{H}) = \prod_{j=1}^{p} h_j^1. \tag{16}$$

Notice that if $\widehat{LB}(\hat{H})$ is given by (16), then

$$\widehat{LB}(\hat{H}) \leqslant g(y) \quad \text{for all } y \in (Y \cap \hat{H}). \tag{17}$$

To see this, let $h^1 \in (Z^k \cap \hat{H})$. Since $Y \subseteq Z^k$, $(Y \cap \hat{H}) \subseteq (Z^k \cap \hat{H})$. Therefore, the minimum value of g over $(Z^k \cap \hat{H})$ is a lower bound for g over $(Y \cap \hat{H})$. Furthermore, since $h^1 \in (Z^k \cap \hat{H})$, it is easy to see that the minimum value of g over $(Z^k \cap \hat{H})$ is equal to $g(h^1)$. The previous two statements together imply that

$$g(h^1) = \prod_{j=1}^{p} h_j^1 \leqslant g(y) \quad \text{for all } y \in (Y \cap \hat{H}). \tag{18}$$

Taken together, (16) and (18) show that (17) holds.

When $h^1 \notin (Z^k \cap \hat{H})$, $LB(\hat{H})$ is again given by (15), but a different method is used to calculate the value of $\widehat{LB}(\hat{H})$. This method relies upon an underestimating function $q : \Re^p \to \Re$ for g on $\hat{H}$. For any $y \in \Re^p$, the value of this underestimating function is given by

$$q(y) = \max\{q_1(y),\ q_2(y)\}, \tag{19}$$

where $q_1(y)$ and $q_2(y)$ are linear functions of y given by the formulas

$$q_1(y) = \sum_{j=1}^{p} \left[\prod_{\substack{i=1\\ i\neq j}}^{p} h_i^1 \right] y_j - \left[(p-1) \prod_{i=1}^{p} h_i^1 \right] \tag{20}$$

and

$$q_2(y) = \sum_{j=1}^{p} \left[\prod_{\substack{i=1\\ i\neq j}}^{p} h_i^2 \right] y_j - \left[(p-1) \prod_{i=1}^{p} h_i^2 \right], \tag{21}$$

respectively. Notice that for $p = 2$, (19)–(21) gives the *convex envelope* of g over $\hat{H}$ [23].

THEOREM 3.3. *If $y \in \hat{H}$ then $q(y) \leqslant g(y)$.*

Proof. To prove the theorem, we will show that if $y \in \hat{H}$, then $q_1(y) \leqslant g(y)$ and $q_2(y) \leqslant g(y)$.

Suppose that $y \in \hat{H}$. To show that $q_1(y) \leqslant g(y)$, we will use induction on p. For $p = 2$, $q_1(y) \leqslant g(y)$ follows from McCormick [23].

Let us hypothesize, then, that for an arbitrary value k of p, $q_1(y) \leqslant g(y)$. Set $p = k + 1$ in (20). Then, since $y \in \hat{H}$,

$$(y_{k+1} - h_{k+1}^1) \geqslant 0$$

and

$$\prod_{t=1}^{k} y_t \geqslant \prod_{t=1}^{k} h_t^1.$$

It follows that

$$(y_{k+1} - h_{k+1}^1) \left(\prod_{t=1}^{k} y_t - \prod_{t=1}^{k} h_t^1 \right) \geqslant 0. \tag{22}$$

Therefore,

$$\begin{aligned}\prod_{t=1}^{k+1} y_t &\geqslant h_{k+1}^1\left(\prod_{t=1}^{k} y_t\right) + \left(\prod_{t=1}^{k} h_t^1\right) y_{k+1} - \prod_{t=1}^{k+1} h_t^1 \\ &\geqslant h_{k+1}^1\left[\sum_{t=1}^{k}\left(\prod_{\substack{i=1\\ i\neq t}}^{k} h_i^1\right) y_t - (k-1)\prod_{i=1}^{k} h_i^1\right] + \left(\prod_{t=1}^{k} h_t^1\right) y_{k+1} - \prod_{t=1}^{k+1} h_t^1 \\ &= \left[\sum_{t=1}^{k}\left(\prod_{\substack{i=1\\ i\neq t}}^{k+1} h_i^1\right) y_t\right] + \left(\prod_{\substack{t=1\\ t\neq k+1}}^{k+1} h_t^1\right) y_{k+1} - k\prod_{t=1}^{k+1} h_t^1 \\ &= \sum_{t=1}^{k+1}\left(\prod_{\substack{i=1\\ i\neq t}}^{k+1} h_i^1\right) y_t - k\prod_{t=1}^{k+1} h_t^1, \end{aligned} \tag{23}$$

where the first inequality follows from (22), and the second inequality follows from $h_{k+1}^1 \geqslant 0$ and the induction hypothesis. Using (1) and (20), from (23) it follows that for $p = k+1$, $g(y) \geqslant q_1(y)$.

By induction on p, we conclude that for all values of p, if $y \in \hat{H}$, then $q_1(y) \leqslant g(y)$. The fact that $q_2(y) \leqslant g(y)$ for all $y \in \hat{H}$ can be similarly shown by induction, so that the theorem is proved. □

When $h^1 \notin (Z^k \cap \hat{H})$, $\widehat{LB}(\hat{H})$ is set equal to the optimal value of the nonlinear programming problem

$$\min q(y), \quad \text{s.t. } y \in (Z^k \cap \hat{H}). \tag{24}$$

Since $(Z^k \cap \hat{H}) \supseteq (Y \cap \hat{H})$, from Theorem 3.3, with this value for $\widehat{LB}(\hat{H})$, the inequality (17) is satisfied. Notice that $\widehat{LB}(\hat{H}) = +\infty$ when $(Z^k \cap \hat{H}) = \emptyset$.

We have seen that whether $h^1 \in (Z^k \cap \hat{H})$ or $h^1 \notin (Z^k \cap \hat{H})$, the algorithm computes $\widehat{LB}(\hat{H})$ so that (17) holds. From (15), it will follow that

$$LB(\hat{H}) \leqslant g(y) \quad \text{for all } y \in (Y \cap \hat{H}), \tag{25}$$

as required by the algorithm, if for each $k \geqslant 1$,

$$LB(H^{k-1}) \leqslant g(y) \quad \text{for all } y \in (Y \cap \hat{H}). \tag{26}$$

To see that (26) holds for all $k \geqslant 1$ and all $y \in (Y \cap \hat{H})$, let y be an arbitrary element of $(Y \cap \hat{H})$. Then, for $k = 1$, we have already shown that (26) holds. Choose any $s > 1$, and suppose that (26) holds for all $k = 1, 2, \ldots, s-1$. Notice that H^{s-1} is the immediate offspring of some rectangle H^t, where $0 \leqslant t < (s-1)$. From (15), this implies that

$$LB(H^{s-1}) = \max\{\widehat{LB}(H^{s-1}),\ LB(H^t)\}.$$

Therefore, $LB(H^{s-1})$ equals either $\widehat{LB}(H^{s-1})$ or $LB(H^t)$. If $LB(H^{s-1}) = \widehat{LB}(H^{s-1})$, then, by (17), we may apply (25) with $\hat{H} = H^{s-1}$. Upon doing so, we see that (26) holds with $k = s$. Otherwise, $LB(H^{s-1}) = LB(H^t)$ will hold. By hypothesis, $LB(H^t) \leqslant g(y)$. Since $LB(H^{s-1}) = LB(H^t)$, this implies that (26) again holds with $k = s$. Therefore, by induction on k and the choice of y, (26) holds for all $k \geqslant 1$ and all $y \in (Y \cap \hat{H})$.

The property given in the next result shows that q and g are equal at certain points of $\hat{H}$.

COROLLARY 3.1. *Let $\hat{H} = \{h \in \Re^p | h^1 \leqslant h \leqslant h^2\}$, and let $q : \Re^p \to \Re$ be defined by (19)–(21). Then $q(y) = g(y)$ for all $y \in \{h^1\} \cup N_1 \cup \{h^2\} \cup N_2$, where, for each $i = 1, 2$, N_i denotes the set of all extreme points of $\hat{H}$ adjacent to h^i.*

Proof. We will prove the corollary for $i = 1$. The proof for $i = 2$ is similar and will not be given.

Let $y = h^1$. Then, from (20),

$$\begin{aligned} q_1(y) &= q_1(h^1) \\ &= \sum_{j=1}^{p} \left[\prod_{\substack{i=1 \\ i \neq j}}^{p} h_i^1 \right] h_j^1 - \left[(p-1) \prod_{i=1}^{p} h_i^1 \right] \\ &= pg(h^1) - (p-1)g(h^1) \\ &= g(h^1). \end{aligned} \tag{27}$$

Now let $y^T = (h_1^1,\ h_2^1, \ldots,\ h_{t-1}^1,\ h_t^2,\ h_{t+1}^1, \ldots,\ h_p^1)$ where $t \in \{1, 2, \ldots,\ p\}$. Then, from (20),

$$\begin{aligned} q_1(y) &= \sum_{\substack{j=1 \\ j \neq t}}^{p} \left[\prod_{\substack{i=1 \\ i \neq j}}^{p} h_i^1 \right] h_j^1 + \left[\prod_{\substack{i=1 \\ i \neq t}}^{p} h_i^1 \right] h_t^2 - \left[(p-1) \prod_{i=1}^{p} h_i^1 \right] \\ &= (p-1)g(h^1) + g(y) - [(p-1)g(h^1)] \\ &= g(y). \end{aligned} \tag{28}$$

From (27) and (28), $q_1(y) = g(y)$ for all $y \in \{h^1\} \cup N_1$. For all $z \in \hat{H}$, by (19) and Theorem 3.3, $q(z) = \max\{q_1(z),\ q_2(z)\} \leqslant g(z)$. The previous two statements imply that if $y \in \{h^1\} \cup N_1$, then

$$q(y) = \max\{g(y),\ q_2(y)\} \leqslant g(y). \tag{29}$$

Choose any $y \in \{h^1\} \cup N_1$. Then there are three possibilities, Case (1): $q_2(y) < g(y)$, Case (2): $q_2(y) = g(y)$, and Case 3: $q_2(y) > g(y)$.

Case (1): $q_2(y) < g(y)$. From (29), this implies that $q(y) = \max\{g(y), q_2(y)\} = g(y)$.

Case (2): $q_2(y) = g(y)$. Then from (29), $q_2(y) = g(y) = \max\{g(y),\ q_2(y)\} = q(y)$.

Case (3): $q_2(y) > g(y)$. Then, using (29), we obtain

$$g(y) < q_2(y) \leqslant q(y) \leqslant g(y),$$

which is impossible. Therefore, this case cannot hold.

Summarizing, the arguments so far imply that for all $y \in \{h^1\} \cup N_1,\ q_2(y) \leqslant g(y) = q_1(y)$. From (19), this implies that for all $y \in \{h^1\} \cup N_1,\ q(y) = g(y)$. □

The second bound computed by the algorithm is a global lower bound for the optimal objective function value of problem (P_Y). In step 0 of the algorithm, this lower bound is denoted $LB(H^0)$ and is given by $LB(H^0) = 0$. For each $k \geqslant 1$, in step k, this lower bound LB is computed via the equation

$$LB = \min\{LB(\tilde{H})|\tilde{H} \in P^k\}, \tag{30}$$

where P^k is given by (14). Subsequently, any $\hat{H} \in P^k$ such that $LB = LB(\hat{H})$ is chosen, and H^k is set equal to $\hat{H}$.

The third bound computed by the algorithm at step $k \geqslant 0$ is a global upper bound UB_k for the optimal objective function value of problem (P_Y). For each $k \geqslant 0$, $UB_k = g(y^c)$, where y^c is the *incumbent* feasible solution for problem (P_Y), i.e., among all feasible solutions for problem (P_Y) found through any point in the algorithm, y^c achieves the smallest value of g.

The set F is the set of *fathomed* subrectangles $\hat{H}$ of H^0. When, in some step $k \geqslant 1$, the algorithm detects that

$$LB(\hat{H}) > UB_k$$

for some subrectangle $\hat{H} \subseteq H^0$, $\hat{H}$ is added to F. Subrectangles in F cannot contain global optimal solutions to problem (P_Y).

The validity of (30) as a global lower bound for the optimal value of problem (P_Y) and of the rules for calculating UB_k, $k \geqslant 0$, and maintaining F can be shown quite easily by using standard branch and bound arguments from global optimization [7].

4. The algorithm

Based upon the results and operations given in Sections 2 and 3, the new outcome space algorithm for problem (P_D) may be stated as follows.

OUTCOME SPACE ALGORITHM

Step 0. Choose $\varepsilon \geqslant 0$. Find points $x^I \in D$ and $y^I \in \Re^p$ such that $\hat{y} > y^I > f(x^I)$. Set $UB_0 = g(y^I)$, $y^c = y^I$ and $x^c = x^I$. Set $F = \emptyset$, and set $H^0 = Z^0 = \{y \in \Re^p | 0 \leqslant y \leqslant \hat{y}\}$. Set $P^0 = \{H^0\}$, $LB = LB(H^0) = 0$, and $y^0 = 0 \in \Re^p$. Set $k = 1$ and go to step k.

Step k. $k \geqslant 1$.

Step k.1. Set UB_k=UB_{k-1}. Find an optimal solution $(x^{k^T},\ \lambda_k)$ to the convex nonlinear programming problem $(T(y))$ with $y = y^{k-1}$.

Step k.2. Set $w^k = \lambda_k(y^I - y^{k-1}) + y^{k-1}$. If $g(w^k) < UB_k$, set $UB_k = g(w^k)$, $y^c = w^k$, $x^c = x^k$, and $F = F \cup \{\hat{H} \in P^{k-1} | LB(\hat{H}) > UB_k\}$, and continue. If $g(w^k) \geqslant UB_k$, continue.

Step k.3. If $UB_k - LB \leqslant \varepsilon$, STOP: x^c is a global ε-optimal solution for problem (P_D), and y^c is a global ε-optimal solution for problem (P_Y). Otherwise, if $\lambda_k > 0$, go to step k.4, and, if $\lambda_k = 0$, set $Z^k = Z^{k-1}$ and go to step k.6.

Step k.4. Find any values for the vectors $u \in \Re^p$ and $v \in \Re^m$ that satisfy (5)–(9).

Step k.5. Set

$$Z^k = Z^{k-1} \cap \{y \in \Re^p | \langle u,\ y - w^k \rangle \geqslant 0\}.$$

Step k.6. Subdivide H^{k-1} into two p-dimensional rectangles $\bar{H}, \bar{\bar{H}} \subseteq \Re^p$ via the rectangular bisection process.

Step k.7. For each of $\hat{H} = \bar{H}$ and $\hat{H} = \bar{\bar{H}}$, compute $LB(\hat{H})$ and a point $y^{\hat{H}} \in \Re^p$ as follows, where we assume without loss of generality that $\hat{H} = \{h \in \Re^p | h^1 \leqslant h \leqslant h^2\}$ for some $h^1, h^2 \in \Re^p$ such that $0 \leqslant h^1 < h^2$.

Case 1: $h^1 \in (Z^k \cap \hat{H})$. Set $y^{\hat{H}} = h^1$ and, with $\widehat{LB}(\hat{H}) = g(h^1)$, set $LB(\hat{H}) = \max\{\widehat{LB}(\hat{H}), LB(H^{k-1})\}$.

Case 2: $h^1 \notin (Z^k \cap \hat{H})$. Set $\widehat{LB}(\hat{H})$ equal to the optimal value of problem (24), and, if $\widehat{LB}(\hat{H}) \neq +\infty$, set $y^{\hat{H}}$ equal to any optimal solution to (24). Then, set $LB(\hat{H}) = \max\{\widehat{LB}(\hat{H}), LB(H^{k-1})\}$.

Step k.8. For each of $\hat{H} = \bar{H}$ and $\hat{H} = \bar{\bar{H}}$, if $LB(\hat{H}) > UB_k$, set $F = F \cup \{\hat{H}\}$. Set

$$P^k = \left\{ H \in [\ (P^{k-1} \backslash \{H^{k-1}\}) \bigcup \{\bar{H}, \bar{\bar{H}}\}\ ]\ |\ H \notin F \right\}.$$

Step k.9. Set

$$LB = \min\{LB(\tilde{H}) | \tilde{H} \in P^k\}.$$

Choose any $\hat{H} \in P^k$ that satisfies $LB = LB(\hat{H})$. Set $H^k = \hat{H}$ and $y^k = y^{\hat{H}}$. Set $k = k + 1$ and go to step k.

We shall soon see that except for step k.1, the steps of the Outcome Space Algorithm can be implemented by simple algebraic calculation and linear programming. First, we will give the convergence properties of the algorithm.

The main convergence property of the algorithm is given in the following result.

THEOREM 4.1. *Suppose that the algorithm is infinite. Then it generates a sequence* $\{w^k\}$ *of feasible solutions for problem* (P_Y)*, every accumulation point of which is a global optimal solution for problem* (P_Y)*, and*

$$\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB(H^k) = \phi.$$

Proof. See Appendix.

A solution $\bar{x}$ is said to be a *global* ε*-optimal* solution for problem (P_D), where $\varepsilon > 0$ is a given number, when $\bar{x} \in D$ and $\phi \leqslant \prod_{j=1}^{p} f_j(\bar{x}) \leqslant \phi + \varepsilon$. A *global* ε*-optimal* solution for problem (P_Y) is defined similarly.

By the following result, when ε is chosen in step 0 to be positive, the algorithm will be finite and will yield global ε-optimal solutions for both problems (P_D) and (P_Y).

COROLLARY 4.1. *If* $\varepsilon > 0$*, then the algorithm is finite and terminates in some step* $\hat{k}$*, where* $\hat{k} \geqslant 1$*. In this case, upon termination* x^c *is a global* ε*-optimal solution for problem* (P_D)*, and* y^c *is a global* ε*-optimal solution for problem* (P_Y)*.*

Proof. Let $\varepsilon > 0$, and suppose, to the contrary, that the algorithm is infinite. Then, by Theorem 4.1,

$$\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB(H^k) = \phi. \tag{31}$$

Notice that for each $k \geqslant 1$, $LB(H^k)$ is equal to the value of LB in step (k + 1.3) of the algorithm. Therefore, it follows from (31), since $\varepsilon > 0$, that there will exist an integer $\bar{k}$ sufficiently large such that $UB_{\bar{k}} - LB < \varepsilon$ is satisfied in step $\bar{k}$.3 of the algorithm. Since the algorithm will terminate in this case, we have a contradiction to the assumption that the algorithm is infinite. Thus, it must be finite. Let $\hat{k} \geqslant 1$ denote the terminal step of the algorithm.

At the termination of the algorithm, $UB_{\hat{k}} = g(y^c)$. From step $\hat{k}$.3, this implies that $g(y^c) - LB(H^{\hat{k}-1}) \leqslant \varepsilon$. From Section 3.3, $LB(H^{\hat{k}-1})$ is a lower bound for the optimal objective function value of problem (P_Y). By Theorem 2.2, the optimal objective function value of problem (P_Y) is ϕ. Taken together, the previous three observations imply that $g(y^c) \leqslant \phi + \varepsilon$. Since $y^c \in Y, \phi \leqslant g(y^c)$. Thus, we have shown that $\phi \leqslant g(y^c) \leqslant \phi + \varepsilon$ and $y^c \in Y$, which together imply that y^c is a global ε-optimal solution for problem (P_Y).

For each $k \geqslant 1$, by step k.1, $x^k \in D$. This implies by step 0 and by step k.2 that $x^c \in D$ at the termination of the algorithm. By assumption, since $x^c \in D$, $f(x^c) > 0$. For each $k \geqslant 1$, from step k.1, we know that $f(x^c) \leqslant w^k$. From step k.2, this implies that $f(x^c) \leqslant y^c$ at the algorithm's termination. From the previous paragraph, $g(y^c) \leqslant \phi + \varepsilon$. Since $0 < f(x^c) \leqslant y^c$, this implies that $\prod_{j=1}^{p} f_j(x^c) \leqslant g(y^c) \leqslant \phi + \varepsilon$. Furthermore, since $x^c \in D, \phi \leqslant \prod_{j=1}^{p} f_j(x^c)$. We have thus shown that $\phi \leqslant \prod_{j=1}^{p} f_j(x^c) \leqslant \phi + \varepsilon$ and $x^c \in D$, which together imply that x^c is a global ε-optimal solution for problem (P_D). □

5. Computational issues

To initiate the Outcome Space Algorithm, a finite vector $\hat{y} \in \Re^p$ must be available that, for each $j = 1, 2, \ldots, p$, satisfies

$$\hat{y}_j > \max f_j(x), \quad \text{s.t. } x \in D. \tag{32}$$

In some cases, $\hat{y}$ can be immediately estimated by simply examining the particular functions f_j, $j = 1, 2, \ldots, p$, and feasible region D involved in the instance of problem (P_D) in question. When this is not possible, an efficient computational procedure is available for determining $\hat{y}$. This procedure avoids the need to solve the p convex maximization problems in (32). It is based upon the following two results.

THEOREM 5.1. *Let $M \in \Re$ and $v^0 \in \Re^n$ satisfy*

$$M > \max\langle \bar{e}, x\rangle, \quad \textit{s.t. } x \in D$$

and

$$v_t^0 = \min x_t, \quad \textit{s.t. } x \in D,$$

for each $t = 1, 2, \ldots, n$, where $\bar{e} \in \Re^n$ denotes the vector of ones. Let $\gamma = M - \langle \bar{e}, v^0\rangle$, and define $v^i \in \Re^n$, $i = 1, 2, \ldots, n$, by

$$v_j^i = \begin{cases} v_j^0, & \textit{if } j \neq i, \\ v_j^0 + \gamma & \textit{if } j = i. \end{cases}$$

Then the convex hull S of $v^0, v^1, \ldots, v^n$ is an n-dimensional simplex with vertex set $\{v^0, v^1, \ldots, v^n\}$, and $S \supseteq D$.

Proof. Since $\gamma \neq 0$, $\{(v^t - v^0) | t = 1, 2, \ldots, n\}$ is a linearly independent set. Therefore, $\{v^0, v^1, \ldots, v^n\}$ is an affinely independent set. By definition, S is thus an n-dimensional simplex with vertex set $\{v^0, v^1, \ldots, v^n\}$.

Suppose that $\bar{x} \in D$. Then, by definition of v^0, $(\bar{x} - v^0) \geqslant 0$. Furthermore, from the definition of M, since $\bar{x} \in D$,

$$\langle \bar{e}, \bar{x} - v^0\rangle < M - \langle \bar{e}, v^0\rangle. \tag{33}$$

By definition of γ, since $(\bar{x} - v^0) \geqslant 0$, this implies that for each $t = 1, 2, \ldots, n$,

$$0 \leqslant (\bar{x} - v^0)_t < \gamma. \tag{34}$$

Therefore, we may choose $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that for each $t = 1, 2, \ldots, n$, $0 \leqslant \alpha_t < 1$ and

$$(\bar{x} - v^0)_t = \alpha_t \gamma. \tag{35}$$

From the definitions of $v^t, t = 1, 2, \ldots, n$, this implies that

$$\begin{aligned}\bar{x} &= v^0 + \sum_{t=1}^{n} \alpha_t (v^t - v^0) \\ &= \left(1 - \sum_{t=1}^{n} \alpha_t\right) v^0 + \sum_{t=1}^{n} \alpha_t v^t. \end{aligned} \tag{36}$$

From (33) and (35) and the definition of γ,

$$\gamma \sum_{t=1}^{n} \alpha_t \leqslant M - \langle \bar{e}, v^0 \rangle = \gamma.$$

Since, by (34), $\gamma > 0$, this implies that $(1 - \sum_{t=1}^{n} \alpha_t) \geqslant 0$. Therefore, from (36), since $\alpha_t \geqslant 0$, $t = 1, 2, \ldots, n$, $\bar{x}$ is in the convex hull of $v^t, t = 0, 1, \ldots, n$, i.e., $\bar{x} \in S$. Thus, $D \subseteq S$, and the proof is complete. □

COROLLARY 5.1. *Any finite vector $\hat{y} \in \Re^p$ that satisfies*

$$\hat{y}_j > \max\{f_j(v^i) | i = 0, 1, \ldots, n\}$$

for each $j = 1, 2 \ldots, p$, also satisfies (32), where $v^i, i = 0, 1, \ldots, n$, are defined in Theorem 5.1.

Proof. From Theorem 5.1 and Martos [21], for each $j = 1, 2, \ldots, p$, since f_j is a convex function over S,

$$\max\{f_j(x) | x \in S\} = \max\{f_j(v^i) | i = 0, 1, \ldots, n\}.$$

Since, by Theorem 5.1, $S \supseteq D$, for each $j = 1, 2, \ldots, p$, the left-hand side of the equation above equals or exceeds

$$\max\{f_j(x) | x \in D\}.$$

By the choice of $\hat{y}$, this implies that (32) holds. □

From Theorem 5.1 and Corollary 5.1, a point $\hat{y}$ that satisfies (32) can be obtained by computing the vertices $v^i, i = 0, 1, \ldots, n$, given in the theorem for the containing simplex S of D and then using the corollary to find $\hat{y}$. Notice that this process involves solving $(n+1)$ convex programming problems, each of which has a simple linear objective function and the same feasible region D.

Step 0 of the algorithm suggests setting both H^0 and Z^0 equal to

$$\{y \in \Re^p | 0 \leqslant y \leqslant \hat{y}\}. \tag{37}$$

Since $f(x) > 0$ for all $x \in D$, this choice and the choice of $\hat{y}$ guarantee that H^0 and Z^0 contain Y. To speed convergence of the algorithm, however, it may be useful to initialize H^0 and Z^0 with a smaller set containing Y than (37). One option for accomplishing this is to calculate, for each $j = 1, 2, \ldots, p$,

$$m_j = \min f_j(x), \quad \text{s.t. } x \in D,$$

and to set both H^0 and Z^0 equal to

$$\{y \in \Re^p | m \leqslant y \leqslant \hat{y}\},$$

where $m = (m_1, m_2, \ldots, m_p)$. While this option involves solving p additional convex programming problems, the savings that could be obtained thereby in speeding convergence may in some cases make the option worthwhile.

For each $k \geqslant 1$, step k.4 calls for solving the feasible system (5)–(9) of equations (see step k.4 and Theorem 3.1) involving the nonnegative variables $u \in \Re^p$ and $v \in \Re^m$. This step can be implemented by linear programming methods as follows. First, for each $j \in \{1, 2, \ldots, p\}$ such that $f_j(x^k) \neq w_j^k$ in (8), set $u_j = 0$. Next, for each $i \in \{1, 2, \ldots, m\}$ such that $g_i(x^k) \neq 0$ in (9), set $v_i = 0$. Let B (as in Section 3) denote the indices of u that are *not* preset to 0, and let C denote the indices of v that are *not* preset to 0. Now initiate the simplex method on the linear program

$$\max \quad \sum_{j \in B} u_j,$$

s.t.

$$\begin{aligned}
&\sum_{j \in B} u_j \leqslant 1,\\
&\sum_{j \in B} u_j [\nabla f_j(x^k)] + \sum_{i \in C} v_i [\nabla g_i(x^k)] = 0,\\
&u_j \geqslant 0, \quad j \in B,\\
&v_i \geqslant 0, \quad i \in C.
\end{aligned}$$

When the simplex method finds a feasible solution $(\hat{u}, \hat{v})$ to this linear program with

$$\sum_{j \in B} \hat{u}_j = 1,$$

it is terminated. Then (u, v) defined by

$$u_j = \begin{cases} 0 & \text{if } j \notin B \\ \hat{u}_j & \text{if } j \in B \end{cases}$$

$$v_i = \begin{cases} 0 & \text{if } i \notin C \\ \hat{v}_i & \text{if } i \in C \end{cases}$$

satisfies (5)–(9).

Case 2 of step $k.7$ of the algorithm calls for solving the nonlinear programming problem (24). In particular, problem (24) calls for minimizing the nonlinear function $q(y)$, given by (19), over a compact polyhedron $(Z^k \cap \hat{H})$. From (19), to solve problem (24), the problem (LP)

$$\begin{aligned} &\min t, \\ &\text{s.t. } t \geqslant q_1(y), \\ &\quad\;\; t \geqslant q_2(y), \\ &\quad\;\; y \in (Z^k \cap \hat{H}) \end{aligned}$$

can be solved instead. Since q_1 and q_2 are linear functions of y, and since $(Z^k \cap \hat{H})$ is a polyhedron, problem (LP) is a linear programming problem. Thus, Case 2 of step $k.7$ can be implemented by linear programming methods.

The discussions in this section and the algorithm statement show that for each $k \geqslant 1$, step k of the Outcome Space Algorithm can be implemented by simple algebraic calculations and linear programming, and by solving one convex nonlinear program (cf. step $k.1$).

6. Example

To illustrate the Outcome Space Algorithm, consider the problem (P_D) with $p = n = 2$, where

$$\begin{aligned} f_1(x_1,\ x_2) &= (x_1 - 2)^2 + 1, \\ f_2(x_1,\ x_2) &= (x_2 - 4)^2 + 1, \end{aligned}$$

and D is defined by the constraints

$$\begin{aligned} g_1(x) &= 25x_1^2 + 4x_2^2 - 100 \leqslant 0, \\ g_2(x) &= x_1 + 2x_2 - 4 \leqslant 0. \end{aligned}$$

Prior to initiating the algorithm, $\hat{y} = (18.0, 38.0)$ was determined by the computational procedure suggested in Section 5.

Initialization: We choose $\varepsilon = 0.025$, $x^i = (0.585, 0.837) \in D$, and $y^I = (3.0, 11.0)$. Then $UB_0 = g(y^I) = 33.0$, $y^c = (3.0, 11.0)$ and $x^c = (0.585, 0.837)$. We set $F = \emptyset$ and, using the suggestion in Section 5, we set $H^0 = Z^0 = \{y \in \Re^2 | (1, 1) \leqslant y \leqslant (18, 38)\}$. We set $P^0 = \{H^0\}$, $LB = LB(H^0) = 1$, and $y^0 = (1, 1)$.

Step 1: We set UB_1=33.0. Then, solving the convex program $(T(y))$ with $y = (1, 1)$ yields $(x^1, \lambda_1) = (0.904, 1.548, 0.600)$. As a result, we determine that $w^1 = (2.200, 7.000)$. Since $g(w^1) = 15.40 < 33.0$, UB_1 is set equal to 15.40, y^c is set equal to (2.200, 7.000), and x^c is set equal to (0.904, 1.548). Since $LB(H^0) =$

$1 \not\geq 15.40$, $F = F \cup \emptyset = \emptyset$. Because $UB_1 - LB = 14.40 \not< 0.025$, we do not stop the algorithm. Instead, we determine that $\lambda_1 = 0.600 > 0$. Using the linear programming procedure given in Section 5, we find that $(u, v) = (0.528, 0.472, 0.000, 1.157)$. Therefore, $Z^1 = \{(y_1, y_2) \in \Re^2 | 1 \leqslant y_1 \leqslant 18, 1 \leqslant y_2 \leqslant 38, 0.528y_1 + 0.472y_2 \geqslant 4.466\}$. Next, H^0 is subdivided by bisection into $\bar{H} = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 19.5 \leqslant h_2 \leqslant 38\}$ and $\bar{\bar{H}} = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1 \leqslant h_2 \leqslant 19.5\}$. We find that $LB(\bar{H}) = 19.5$, $y^{\bar{H}}$=(1.00,19.50), and, via the linear programming method given in Section 5, that $LB(\bar{\bar{H}})$=7.564 and $y^{\bar{\bar{H}}}$= (7.564, 1.000). Since $LB(\bar{H}) > UB_1$=15.40, but $LB(\bar{\bar{H}}) \not> UB_1$, F is set equal to $\{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 19.5 \leqslant h_2 \leqslant 38\}$ and P^1 is set equal to $\{\{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1 \leqslant h_2 \leqslant 19.5\}\}$. As a result, $LB = 7.564$, $H^1 = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1 \leqslant h_2 \leqslant 19.5\}$, and $y^1 = (7.564, 1.000)$.

Step 2: We set $UB_2 = 15.40$. With $y = (7.564, 1.000)$, the convex program $(T(y))$ has optimal solution $(x^2, \lambda_2) = (-0.2050, 2.1025, 0.36)$. The point w^2 is found to be (5.921, 4.600). Since $g(w^2) = 27.2366 \geqslant UB_2$, we next test for termination. Upon doing so, we find that $UB_2 - LB = 15.400 - 7.564 = 7.836 \not< \epsilon = 0.025$. Continuing, we see that $\lambda_2 = 0.36 > 0$. We therefore proceed to computing $(u, v) = (0.1771, 0.8229, 0.000, 0.7810)$. As a result, $Z^2 = \{(y_1, y_2) \in \Re^2 | 1 \leqslant y_1 \leqslant 18, 1 \leqslant y_2 \leqslant 38, 0.528y_1 + 0.472y_2 \geqslant 4.466, 0.1771y_1 + 0.8229y_2 \geqslant 4.833\}$. Subdividing H^1 via bisection yields $\bar{H} = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 10.25 \leqslant h_2 \leqslant 19.50\}$ and $\bar{\bar{H}} = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1.00 \leqslant h_2 \leqslant 10.25\}$. We find that $LB(\bar{H}) = 10.25$, $y^{\bar{H}} = (1.00, 10.25)$, $LB(\bar{\bar{H}}) = 8.0$, and $y^{\bar{\bar{H}}} = (3.9690, 5.0215)$. We then set $P^2 = \{\{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 10.25 \leqslant h_2 \leqslant 19.50\}, \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1.00 \leqslant h_2 \leqslant 10.25\}\}$, $LB = \min\{10.25, 8.00\} = 8.00$, $H^2 = \{h \in \Re^2 | 1 \leqslant h_1 \leqslant 18, 1.00 \leqslant h_2 \leqslant 10.25\}$, and $y^2 = (3.9690, 5.0215)$.

The algorithm terminates in step 8 with global ε-optimal solutions for (P_D) and for (P_Y) given by

$$x^* = (1.8881967, 1.0553269)$$

and

$$y^* = (1.0125, 9.6711),$$

respectively, where $\varepsilon = 0.025$. In addition, the terminal step shows that $9.7669887 \leqslant \phi \leqslant 9.7919887$.

7. Concluding remarks

We have presented a new algorithm for solving the Convex Multiplicative Programming problem (P_D). The algorithm, called the Outcome Space Algorithm, has the following key advantages.

(1) By working in the outcome space $\Re^p$ of problem (P_D) instead of in the decision space $\Re^n$, the algorithm economizes the computations required to solve

the problem. This is mainly due to the fact that p is typically much smaller than n. The algorithm works in $\Re^p$ instead of $\Re^n$ by solving the problem (P_Y), which is equivalent to problem (P_D), in order to solve problem (P_D).

(2) In contrast to typical outer approximation algorithms, by combining branch and bound with outer approximation (both in the outcome space), the algorithm avoids computing the vertices of the successive polyhedra containing the set Y that are created by the outer approximation process. Avoiding such calculations is known to yield considerable computational savings.

(3) Only one nonlinear convex programming problem need be solved per iteration of the algorithm. The remaining operations in each iteration can be implemented by simple algebraic means and linear programming techniques.

For these reasons, we conclude that the Outcome Space Algorithm offers a potentially very attractive option for solving convex multiplicative programming problems.

Appendix

Proof of Theorem 4.1. Since the algorithm is infinite, it generates a sequence $\{y^k\}$ of points that satisfies $y^k \in H^k$ for each k. We may assume that for each k, $H^{k+1} \subset H^k$ i.e., that the sequence $\{H^k\}$ is *nested*. By Theorem 3.2, $\lim_k H^k = \cap_k H^k = \{\bar{y}\}$ for some point $\bar{y} \in \Re^p$. Therefore, $\lim_k y^k = \bar{y}$. We will first show that $\bar{y} \in Y$.

Consider the sequence $\{\lambda_k\}$. If $\{\lambda_k\}$ contains a subsequence consisting entirely of zeros, then, from step k.1 of the algorithm and the fact that $y^k \in H^k \subseteq H^0$ for all k, a corresponding subsequence of points from $\{y^k\}$ will lie entirely in Y, so that $\bar{y} \in Y$ will hold.

Now assume that $\{\lambda_k\}$ contains no subsequence of zeros. We may then assume for notational simplicity that $\lambda_k > 0$ for all k. From steps k.1, k.2 and k.4, for each k, w^k, λ_k and $u = u^k$ belong to the compact sets ∂Y, $\{t \in \Re | 0 \leqslant t \leqslant 1\}$, and $\{u \in \Re^p | u \geqslant 0, u_1 + u_2 + \cdots + u_p = 1\}$, respectively. Therefore, we may assume that as $k \to \infty$, $w_k \to \bar{w} \in \partial Y$, $\lambda_k \to \bar{\lambda}$, where $0 \leqslant \bar{\lambda} \leqslant 1$, and $u^k \to \bar{u}$, where $\bar{u} \geqslant 0$ and $\bar{u}_1 + \bar{u}_2 + \cdots + \bar{u}_p = 1$. By step k.2, $\bar{w} = \bar{\lambda}(y^I - \bar{y}) + \bar{y}$. Since $\bar{w} \in \partial Y$ and $y^I \notin \partial Y$, this implies that $\bar{\lambda} \neq 1$. In addition, this equation implies that

$$(\bar{y} - \bar{w}) = [\bar{\lambda}/(1 - \bar{\lambda})](\bar{w} - y^I). \tag{38}$$

From Theorem 3.1(b) it is easy to see that $\langle \bar{u},\ y - \bar{w} \rangle \geqslant 0$ for all $y \in Y$. Furthermore, it is a simple exercise to show that, as a result,

$$\langle \bar{u},\ y^I - \bar{w} \rangle > 0. \tag{39}$$

From Theorem 3.1(c), we see that

$$\langle \bar{u},\ \bar{y} - \bar{w} \rangle \leqslant 0. \tag{40}$$

By using steps k.5 and k.7 of the algorithm and the fact that $\{H^k\}$ is nested, where $y^k \in H^k$ for all k, it is easy to show that $\langle \bar{u},\ \bar{y} - \bar{w} \rangle \geqslant 0$. Combined with

(40), this shows that $\langle \bar{u},\ \bar{y} - \bar{w} \rangle = 0$. Using (38), we obtain that

$$\langle \bar{u},\ \bar{y} - \bar{w} \rangle = [\bar{\lambda}/(1-\bar{\lambda})]\langle \bar{u},\ \bar{w} - y^I \rangle.$$

By (39), since $\langle \bar{u},\ \bar{y} - \bar{w} \rangle = 0$, this implies that $\bar{\lambda} = 0$. As a result, since $\bar{w} = \bar{\lambda}(y^I - \bar{y}) + \bar{y}$, we see that $\bar{w} = \bar{y}$. Since $\bar{w} \in Y$, this shows that $\bar{y} \in Y$.

Having shown that $\bar{y} \in Y$, we now show that $\bar{y}$ is an optimal solution to problem (P_Y). There are two cases to consider. Either (1) for some subsequence of $\{H^k\}$, Case 1 holds in step $k.7$ or (2) for some subsequence of $\{H^k\}$, Case 2 holds in step $k.7$.

Case (1): Assume that Case 1 in step $k.7$ holds for some subsequence of $\{H^k\}$. Then we may assume, without loss of generality, that this case holds for each element of $\{H^k\}$. Suppose that $k \geqslant 1$. Let $H^k = \{h \in \Re^p | h_k^1 \leqslant h \leqslant h_k^2\}$. Then, from steps $k.7$ and $k.9$, $h_k^1 = y^k$, $\widehat{LB}(H^k) = g(h_k^1)$, and $LB(H^k) \geqslant g(h_k^1)$. Since the sequence $\{LB(H^k)\}$ can be shown to be nondecreasing and bounded above by ϕ, and since, by Theorem 3.2, $\lim_k H^k = \lim_k y^k = \bar{y}$, this implies that

$$\phi \geqslant \lim_k LB(H^k) \geqslant \lim_k g(y^k) = g(\bar{y}).$$

Therefore, $g(\bar{y}) \leqslant \phi$. From Theorem 2.2, since $\bar{y} \in Y$, it follows that $g(\bar{y}) = \phi$ and $\bar{y}$ is an optimal solution for problem (P_Y).

Case (2): Assume that Case 2 in step $k.7$ holds for some subsequence of $\{H^k\}$. Then we may assume, without loss of generality, that this case holds for each element of $\{H^k\}$. Assume $k \geqslant 1$, and let $H^k = \{h \in \Re^p | h^{1,k} \leqslant h \leqslant h^{2,k}\}$. From step $k.9$ and Case 2 of step $k.7$,

$$LB(H^k) \geqslant \widehat{LB}(H^k) = q(y^k),$$

where, by (19)–(21),

$$q(y^k) = \max \left\{ \sum_{j=1}^{p} \left[\prod_{\substack{i=1 \\ i \neq j}}^{p} (h^{1,k})_i \right] y_j^k - (p-1) \prod_{i=1}^{p} (h^{1,k})_i, \right.$$
$$\left. \sum_{j=1}^{p} \left[\prod_{\substack{i=1 \\ i \neq j}}^{p} (h^{2,k})_i \right] y_j^k - (p-1) \prod_{i=1}^{p} (h^{2,k})_i \right\}.$$

Since $\lim_k H^k = \lim_k y^k = \bar{y}$ and $\{LB(H^k)\}$ is a nondecreasing sequence bounded above by ϕ, this implies that

$$\phi \geqslant \lim_k LB(H^k) \geqslant \lim_k q(y^k),$$

where

$$\begin{aligned}\lim_k q(y^k) &= \max\left\{\sum_{j=1}^{p}\left(\prod_{i=1}^{p}\bar{y}_i\right)-(p-1)\left(\prod_{i=1}^{p}\bar{y}_i\right),\right.\\ &\qquad \left.\sum_{j=1}^{p}\left(\prod_{i=1}^{p}\bar{y}_i\right)-(p-1)\left(\prod_{i=1}^{p}\bar{y}_i\right)\right\}\\ &= \prod_{i=1}^{p}\bar{y}_i\\ &= g(\bar{y}).\end{aligned}$$

Therefore, $g(\bar{y}) \leqslant \phi$. From Theorem 2.2, since $\bar{y} \in Y$, it follows that $g(\bar{y}) = \phi$ and $\bar{y}$ is an optimal solution for problem (P_Y).

Notice that in both Case (1) and Case (2),

$$\phi \geqslant \lim_k\ LB(H^k) \geqslant g(\bar{y}) = \phi.$$

Therefore,

$$\lim_k\ LB(H^k) = \phi.$$

Now let $\hat{w}$ be an accumulation point of w^k. Then, since $\lim_k\ w^k = \bar{w}$, $\hat{w} = \bar{w}$. Furthermore, since $\bar{w} = \bar{y}$, $\bar{w} = \hat{w}$ is an optimal solution for problem (P_Y).

From step k.2, for each $k \geqslant 1$, $g(w^k) \geqslant UB_k$. From the same step, since $w^k \in Y$ for all k, $\{UB_k\}$ is a nonincreasing sequence bounded below by ϕ. The latter two statements together imply that

$$\lim_k\ g(w^k) \geqslant \lim_k\ UB_k \geqslant \phi.$$

Since $\lim_k w^k = \bar{w}$ and $\bar{w}$ is an optimal solution for problem (P_Y), this implies that

$$\phi = g(\bar{w}) \geqslant \lim_k\ UB_k \geqslant \phi.$$

Therefore, $\phi = \lim_k\ UB_k$, and the proof is complete. □

References

1. Aneja, Y.P., Aggarwal, V. and Nair, K.P.K. (1984), On a Class of Quadratic Programs, *European Journal of Operational Research* 18: 62–70.
2. Benson, H.P. and Boger, G.M. (1997), Multiplicative Programming Problems: Analysis and Efficient Point Search Heuristic, Journal of Optimization Theory and Applications 94, 487–510.
3. Falk, J.E. and Palocsay, S.W. (1994), Image Space Analysis of Generalized Fractional Programs, *Journal of Global Optimization* 4: 63–88.

4. Geoffrion, A.M. (1967), Solving Bicriterion Mathematical Programs, *Operations Research* 15: 39–54.
5. Henderson, J.M. and Quandt, R.E. (1971), *Microeconomic Theory*, McGraw-Hill, New York.
6. Horst, R., Pardalos, P.M. and Thoai, N.V. (1995), *Introduction to Global Optimization*, Kluwer, Dordrecht, The Netherlands.
7. Horst, R. and Tuy, H. (1993), *Global Optimization: Deterministic Approaches*, 2nd Edition, Springer, Berlin.
8. Jaumard, B., Meyer, C. and Tuy, H. (1997), Generalized Convex Multiplicative Programming via Quasiconcave Minimization, *Journal of Global Optimization* 10: 229–256.
9. Konno, H. and Inori, M. (1988), Bond Portfolio Optimization by Bilinear Fractional Programming, *Journal of the Operations Research Society of Japan* 32: 143–158.
10. Konno, H. and Kuno, T. (1990), Generalized Linear Multiplicative and Fractional Programming, *Annals of Operations Research* 25: 147–161.
11. Konno, H. and Kuno, T. (1992), Linear Multiplicative Programming, *Mathematical Programming* 56: 51–64.
12. Konno, H. and Kuno, T. (1995), Multiplicative Programming Problems, In: *Handbook of Global Optimization*, R. Horst and P.M. Pardalos, Eds., Kluwer, Dordrecht, The Netherlands, pp. 369–405.
13. Konno, H., Kuno, T. and Yajima, Y. (1992), Parametric Simplex Algorithms for a Class of NP-Complete Problems Whose Average Number of Steps is Polynomial, *Computational Optimization and Applications* 1, 227–239.
14. Konno, H., Kuno, T. and Yajima, Y. (1994), Global Minimization of a Generalized Convex Multiplicative Function, *Journal of Global Optimization* 4: 47–62.
15. Konno, H., Yajima, Y. and Matsui, T. (1991), Parametric Simplex Algorithms for Solving a Special Class of Nonconvex Minimization Problems, *Journal of Global Optimization* 1: 65–81.
16. Kuno, T. (1996), A Practical Algorithm for Minimizing a Rank-Two Saddle Function on a Polytope, *Journal of the Operations Research Society of Japan* 39: 63–76.
17. Kuno, T. and Konno, H. (1991), A Parametric Successive Underestimation Method for Convex Multiplicative Programming Problems, *Journal of Global Optimization* 1: 267–285.
18. 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.
19. Maling, K., Mueller, S.H. and Heller, W.R. (1982), On Finding Most Optimal Rectangular Package Plans, *Proceedings of the 19th Design Automation Conference*, pp. 663–670.
20. Mangasarian, O.L. (1969), *Nonlinear Programming*, McGraw-Hill, New York.
21. Martos, B. (1965), The Direct Power of Adjacent Vertex Programming Methods, *Management Science* 12: 241–252.
22. Matsui, T. (1996), NP-Hardness of Linear Multiplicative Programming and Related Problems, *Journal of Global Optimization* 9: 113–119.
23. McCormick, G.P. (1976), Computability of Global Solutions to Factorable Nonconvex Programs: Part I – Convex Underestimating Problems, *Mathematical Programming* 10: 147–175.
24. Muu, L.D. and Tam, B.T. (1992), Minimizing the Sum of a Convex Function and the Product of Two Affine Functions over a Convex Set, *Optimization* 24: 57–62.
25. Pardalos, P.M. (1990), Polynomial Time Algorithms for Some Classes of Constrained Nonconvex Quadratic Problems, *Optimization* 21, 843–853.
26. Ryoo, H.S. and Sahinidis, N.V. (1996), A Branch-and-Reduce Approach to Global Optimization, *Journal of Global Optimization* 8: 107–138.
27. Schaible, S. and Sodini, C. (1995), Finite Algorithm for Generalized Linear Multiplicative Programming, *Journal of Optimization Theory and Applications* 87: 441–455.
28. Thoai, N.V. (1991), A Global Optimization Approach for Solving the Convex Multiplicative Programming Problem, *Journal of Global Optimization* 1: 341–357.

29. Tuy, H. (1991), Polyhedral Annexation, Dualization, and Dimension Reduction Technique in Global Optimization, *Journal of Global Optimization* 1: 229–244.
30. Tuy, H. and Tam, B.T. (1992), An Efficient Solution Method for Rank-Two Quasiconcave Minimization Problems, *Optimization* 24: 43–56.