Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2011, Article ID 464832, 21 pages
doi:10.1155/2011/464832

Research Article

Generating Efficient Outcome Points for Convex Multiobjective Programming Problems and Its Application to Convex Multiplicative Programming

Le Quang Thuy, Nguyen Thi Bach Kim, and Nguyen Tuan Thien

Faculty of Applied Mathematics and Informatics, Hanoi University of Technology, Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam

Correspondence should be addressed to Le Quang Thuy, thuylq-fami@mail.hut.edu.vn

Received 30 November 2010; Revised 7 April 2011; Accepted 28 May 2011

Academic Editor: Ya Ping Fang

Convex multiobjective programming problems and multiplicative programming problems have important applications in areas such as finance, economics, bond portfolio optimization, engineering, and other fields. This paper presents a quite easy algorithm for generating a number of efficient outcome solutions for convex multiobjective programming problems. As an application, we propose an outer approximation algorithm in the outcome space for solving the multiplicative convex program. The computational results are provided on several test problems.

1. Introduction

The convex multiobjective programming problem involves the simultaneously minimize $p \geq 2$ noncomparable convex objective functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, over nonempty convex feasible region X in $\mathbb{R}^n$ and may be written by

$$\text{Min} f(x), \quad \text{s.t. } x \in \mathrm{X}, \tag{VP_X}$$

where $f(x) = (f_1(x), f_2(x), \ldots, f_p(x))^T$. When $\mathrm{X} \subset \mathbb{R}^n$ is a polyhedral convex set and f_j, $j = 1, \ldots, p$ are linear functions, problem (VP_X) is said to be a *linear multiobjective programming problem* (LP_X).

For a nonempty set $Q \subset \mathbb{R}^p$, we denote by Q_E and Q_{WE} the sets of all *efficient points* and *weakly efficient points* of Q, respectively, that are

$$\begin{aligned} Q_E &:= \{q^* \in Q \mid \nexists q \in Q \ \text{ such that } \ q^* \geq q \ \text{ and } \ q^* \neq q\} \\ &= \left\{q^* \in Q \mid \left(q^* - \mathbb{R}^p_+\right) \cap Q = \{q^*\}\right\}, \\ Q_{\text{WE}} &:= \{q^* \in Q \mid \nexists q \in Q \ \text{ such that } \ q^* \gg q\} \\ &= \left\{q^* \in Q \mid \left(q^* - \operatorname{int} \mathbb{R}^p_+\right) \cap Q = \emptyset\right\}. \end{aligned} \tag{1.1}$$

Here for any two vectors $a, b \in \mathbb{R}^p$, the notations $a \geq b$ and $a \gg b$ mean $a - b \in \mathbb{R}^p_+$ and $a - b \in \operatorname{int} \mathbb{R}^p_+$, respectively, where $\mathbb{R}^p_+$ is the nonnegative orthant of $\mathbb{R}^p$, and $\operatorname{int} \mathbb{R}^p_+$ is the interior of $\mathbb{R}^p_+$. By definition, $Q_E \subseteq Q_{\text{WE}}$.

The set

$$Y^{\text{image}} = f(X) = \{f(x) \mid x \in X\} \tag{1.2}$$

is called the *outcome set* (or *image*) *of X under f*. A point $x^0 \in X$ is said to be an *efficient solution* for problem (VP_X) if $f(x^0) \in Y^{\text{image}}_E$. For simplicity of notation, let X_E denote the set of all efficient solutions for problem (VP_X). When $f(x^0) \in Y_{\text{WE}}$, x^0 is called *weakly efficient solution* for problem (VP_X) and the set of all weakly efficient solutions is denoted by X_{WE}, it is clear that X_E and X_{WE} are the preimage of Y^{image}_E and $Y^{\text{image}}_{\text{WE}}$ under f, respectively. We will refer to Y^{image}_E and $Y^{\text{image}}_{\text{WE}}$ as the *efficient outcome set* and *weakly efficient outcome set* for problem (VP_X), respectively.

The goal of problem (VP_X) is to generate the sets X_E and X_{WE} or at least their subsets. However, it has been shown that, in practice, the decision maker prefers basing his or her choice of the most preferred solution primarily on Y^{image}_E and $Y^{\text{image}}_{\text{WE}}$ rather than X_E and X_{WE}. Arguments to this effect are given in [1].

It is well known that the task of generating X_E, X_{WE}, Y^{image}_E, $Y^{\text{image}}_{\text{WE}}$, or significant portions of these sets for problem (VP_X) is a difficult problem. This is because they are, in general, nonconvex sets, even in the case of linear multiobjective programming problem (LP_X).

Problem (VP_X) arises in a wide variety of applications in engineering, economics, network planning, production planning, operations research, especially in multicriteria design and in multicriteria decision making (see, for instance, [2, 3]). Many of the approaches for analyzing convex multiobjective programming problems involve generating either the sets X_E, X_{WE}, Y^{image}_E, and $Y^{\text{image}}_{\text{WE}}$ or a subset thereof, without any input from decision maker (see, e.g., [1, 2, 4–13] and references therein). For a survey of recent developments see [6].

This paper has two purposes:

(i) the first is to propose an algorithm for generating a number of efficient outcome points for convex multiobjective programming problem (VP_X) depending on the requirements of decision makers (Algorithm 1 in Section 2). Computational experiments show that this algorithm is quite efficient;

(ii) as an application, we present an outer approximation algorithm for solving the convex multiplicative programming problem (CP_X) associated with the problem

(VP_X) in outcome space $\mathbb{R}^p$ (Algorithm 2 in Section 3), where the problem (CP_X) can be formulated as

$$\min \prod_{j=1}^{p} f_j(x), \quad \text{s.t.}\ x \in X. \qquad (CP_X)$$

It is well known that problem (CP_X) is a global optimization problem and is known to be NP-hard, even special cases such as when $p = 2$, X is a polyhedron, and f_j is linear for each $j = 1, 2$ (see [14]). Because of the wide range of its applications, this problem attracts a lot of attention from both researchers and practitioners. Many algorithms have been proposed for globally solving the problem (CP_X), (see, e.g., [10, 14–20] and references therein).

The paper is organized as follows. In Section 2, we present Algorithm 1 for generating efficient outcome points for convex multiobjective programming problem (VP_X) and its theoretical basis. To illustrate the performance of Algorithm 1, we use it to generate efficient points for a sample problem. The Algorithm 2 for solving the convex multiplicative programming (CP_X) associated with the problem (VP_X), and numerical examples are described in Section 3.

2. Generating Efficient Outcome Points for Problem (VP_X)

2.1. Theoretical Basis

Assume henceforth that $X \subset \mathbb{R}^n$ is a nonempty, compact convex set given by

$$X := \{x \in \mathbb{R}^n \mid g_i(x) \le 0, \quad i = 1, \dots, m\}, \qquad (2.1)$$

where all the $g_1, g_2, \dots, g_m$ are convex functions on $\mathbb{R}^n$. Then the set

$$G^0 = Y^{\text{image}} + \mathbb{R}^p_+ \qquad (2.2)$$

is nonempty full-dimensional convex set in $\mathbb{R}^p$. Furthermore, from Yu [21, page 22, Theorem 3.2], we know that $Y_E^{\text{image}} = G_E^0$.

For each $k = 1, \dots, p$, we consider the following problem:

$$\min y_k, \quad \text{s.t.}\ y \in G^0. \qquad (P^0(k))$$

We denote the optimal value for the problem $(P^0(k))$ by y_k^{opt} and the optimal solution for this problem by $y^k = (y_1^k, \dots, y_p^k) \in \mathbb{R}^p$.

Let $y^m = (y_1^m, y_2^m, \dots, y_p^m)$, where

$$y_k^m = y_k^{\text{opt}} \quad \forall k = 1, 2, \dots, p. \qquad (2.3)$$

As usual, the point y^m is said to be an *ideal efficient point* of G^0. It is clear that if $y^m \in G^0$ then $Y_E^{\text{image}} = G_E^0 = \{y^m\}$. Therefore, we suppose that $y^m \notin G^0$. Obviously, by definition, if $(x^*, y^*) \in \mathbb{R}^{n+p}$ is an optimal solution for the problem $(P(k))$ given by

$$\begin{aligned} &\min y_k \\ &\text{s.t. } f_j(x) - y_j \leq 0, \quad j = 1, \ldots, p, \\ &g_i(x) \leq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{$P(k)$}$$

then $y^k = y^*$ is an optimal solution for the problem $(P^0(k))$, and the optimal values of these two problems are equal.

To generate various efficient outcome points in Y_E^{image}, the algorithm will rely upon the point $y^M = (y_1^M, y_2^M, \ldots, y_p^M)$, where for each $j = 1, 2, \ldots, p$, $y_j^M = \alpha$, and α is any real number satisfying

$$\alpha > \max\left\{y_j^k,\ k = 1, \ldots, p,\ j = 1, \ldots, p\right\}. \tag{2.4}$$

We consider the set G given by

$$G = G^0 \cap \left(y^M - \mathbb{R}_+^p\right). \tag{2.5}$$

It is obvious that G is a nonempty, full-dimensional compact convex set in $\mathbb{R}^p$. The set G is instrumental in Algorithm 1 to be presented in Section 2.2 for generating efficient outcome points for problem (VP_X).

Remark 2.1. In 1998, Benson [1] presented the outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiobjective linear programming problem. Here, the G seems to be analogous with the set Y considered by Benson [1]. However, note that $Y \supset Y^{\text{image}}$ and Y^{image} is not necessarily a subset of G. The Figure 1 illustrates the set G in the case of $p = 2$.

Proposition 2.2. $Y_E^{\text{image}} = G_E$.

Proof. This result is easy to show by using Yu [21, page 22, Theorem 3.2] and the definition of the point y^M. Therefore, the proof is omitted. □

Let

$$B^0 = \left(y^m + \mathbb{R}_+^p\right) \cap \left(y^M - \mathbb{R}_+^p\right). \tag{2.6}$$

It is clear that $G \subset B^0$. The following fact will play an important role in establishing the validity of our algorithm.

Proposition 2.3. *For each point $\overline{y} \in B^0 \setminus G$, let y^w denote the unique point on the boundary of G that belongs to the line segment connecting $\overline{y}$ and y^M. Then $y^w \in G_E$.*

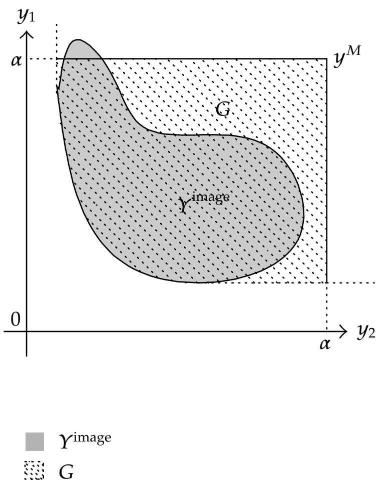

Figure 1

Proof. Let $D = G - y^w$. Since G is the compact convex set and y^w belongs to the boundary of G, the set D is also the compact convex set containing the origin 0 of the outcome space $\mathbb{R}^p$ and 0 belongs to the boundary of D. According to the separation theorem [22], there is a nonzero vector $q \in \mathbb{R}^p$ such that

$$\langle q, u \rangle \geq 0 \quad \forall u \in D. \tag{2.7}$$

Let $S \subset \mathbb{R}^p$ be a p-simplex with vertices $0, e^1, e^2, \ldots, e^p$, where $e^1, e^2, \ldots, e^p$ are the unit vectors of $\mathbb{R}^p$. By definition, we can choose the point y^M such that $(y^w + S) \subset G$. This implies that $S \subset D$. From (2.7), by taking u to be $e^1, e^2, \ldots, e^p$ to see that

$$q \geq 0. \tag{2.8}$$

Furthermore, the expression (2.7) can be written by

$$\langle q, y - y^w \rangle \geq 0 \quad \forall y \in G, \tag{2.9}$$

that is,

$$\langle q, y \rangle \geq \langle q, y^w \rangle \quad \forall y \in G. \tag{2.10}$$

According to [23, Chapter 4, Theorem 2.10], a point $y^* \in G$ is a weakly efficient point of G if and only if there is a nonzero vector $v \in \mathbb{R}^p$ and $v \geq 0$ such that y^* is an optimal solution to the convex programming problem

$$\min\{\langle v, y \rangle \mid y \in G\}. \tag{2.11}$$

Combining this fact, (2.8) and (2.10) give $y^w \in G_{WE}$. To complete the proof, it remains to show that $y^w \in G_E$. Assume the contrary that

$$y^w \in G_{WE} \setminus G_E. \tag{2.12}$$

By the definition, we have

$$G_{WE} = G_E \cup \left(\bigcup_{k=1}^{p} F_k \right), \tag{2.13}$$

where for each $k = 1, \dots, p$, F_k is the optimal solution set for the following problem:

$$\min y_k, \quad \text{s.t.}\, y \in G. \tag{$P^G(k)$}$$

It is easy to see that the optimal values of two problems $(P^G(k))$ and $(P^0(k))$ are the same. From this fact and the definition of the point y^m, it follows that

$$F_k = \left\{ y \in G \mid y_k = y_k^m \right\}. \tag{2.14}$$

Therefore, if $y^w \in G_{WE} \setminus G_E$ then there is $i_0 \in \{1, 2, \dots, p\}$ such that $y_{i_0}^w = y_{i_0}^m$. Since $\overline{y} \in B^0 \setminus G$, we always have $\overline{y} \geq y^m$ and $y^w = \lambda y^M + (1 - \lambda)\overline{y}$ with $0 < \lambda < 1$. By the choice of the point y^M, we have $y^M \gg y^m$. Hence,

$$y_{i_0}^m = y_{i_0}^w = \lambda y_{i_0}^M + (1 - \lambda)\overline{y}_{i_0} > y_{i_0}^m. \tag{2.15}$$

This contradiction proves that y^w must belong to the efficient outcome set G_E. □

Remark 2.4. This Proposition 2.3 can be viewed as an Benson's extension in [1, Proposition 2.4]. Benson showed that the unique point on the boundary of Y (which is considered in [1]) and corresponding segment belongs to the weakly efficient outcome set Y_{WE}, here we prove that y^w belongs to the efficient outcome set G_E.

Remark 2.5. Let $\overline{y}$ be a given point in $B^0 \setminus G$. From Propositions 2.3 and 2.2, the line segment connecting $\overline{y}$ and y^M contains a unique point $y^w \in Y_E^{\text{image}}$ which lies on the boundary of G. To determine this efficient outcome point y^w, we have to find the unique value λ^* of λ, $0 < \lambda < 1$, such that

$$y^w = \lambda^* y^M + (1 - \lambda^*)\overline{y} = \overline{y} + \lambda^* \left(y^M - \overline{y} \right) \tag{2.16}$$

belongs to the boundary of G (see Figure 2). It means that λ^* is the optimal value for the problem

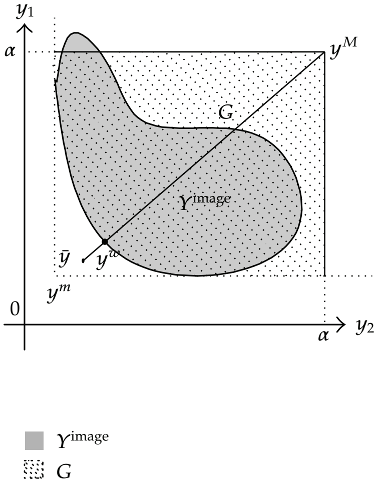

Figure 2

$$\min \lambda$$

$$\text{s.t.}\, \overline{y} + \lambda\left(y^M - \overline{y}\right) \in G, \tag{2.17}$$

$$0 \leq \lambda \leq 1.$$

By definition, it is easy to see that λ^* is also the optimal value for the following convex programming problem with linear objective function

$$\begin{aligned} &\min \lambda \\ &\text{s.t.}\, f(x) - \lambda\left(y^M - \overline{y}\right) - \overline{y} \;\leq 0, \\ &g_i(x) \leq 0, \quad i = 1, \dots, m, \\ &0 \leq \lambda \leq 1. \end{aligned} \tag{$T(\overline{y})$}$$

Note that λ^* exists because the feasible region of problem $(T(\overline{y}))$ is a nonempty, compact convex set. Furthermore, by the definition, it is easy to show that if $(x^*, \lambda^*) \in \mathbb{R}^{n+1}$ is an optimal solution for Problem $(T(\overline{y}))$, then x^* is an efficient solution for the convex multiobjective programming problem (VP_X), that is, $x^* \in X_E$. For the sake of convenience, x^* is said to be *an efficient solution associated with* y^w, and y^w is said to be *an efficient outcome point generated by* $\overline{y}$. It is easily seen that by varying the choices of points $\overline{y} \in B^0 \setminus G$, the decision maker can generate multiple points in Y_E^{image}. In theory, if $\overline{y}$ could vary over all of $B^0 \setminus G$, we could generate all of the properly efficient points of Y (see [24]) in this way.

The following Proposition 2.6 shows that for each efficient outcome point y^w generated by a given point $\overline{y}$, we can determine p new points which belong to $B^0 \setminus G$ and differ $\overline{y}$. This work is accomplished by a technique called cutting reverse polyblock.

A set of the form $B = \bigcup_{y \in V}[y, y^M] \subset \mathbb{R}^p$, where $[y, y^M] := \{\overline{y} \mid y \leq \overline{y} \leq y^M\}$, $V \subset B^0 = [y^m, y^M]$, and $|V| < +\infty$, is called a *reverse polyblock* in hyperrectangle B^0 with *vertex set* V. A vertex $y \in V$ is said to be *proper* if there is no $y' \in V \setminus \{y\}$ such that $[y, y^M] \subset [y', y^M]$. It is clear that a reverse polyblock is completely determined by its proper vertices.

Proposition 2.6 (see, e.g., [20]). *Let $G = (Y^{\text{image}} + \mathbb{R}^p_+) \cap (y^M - \mathbb{R}^p_+)$ be a nonempty compact convex set contained in a reverse polyblock $B = \bigcup_{y \in V}[y, y^M]$ with vertex set $V \subset B^0$. Let $v \in V \setminus G$ and y^w is the unique point on the boundary of G that belongs to the line segment connecting v and y^M. Then, $\overline{B} = B \setminus [v, y^w]$ is a reverse polyblock with vertex set $\overline{V} = (V \setminus \{v\}) \cup \{v^1, v^2, \ldots, v^p\}$, where*

$$v^i = y^w - (y^w_i - v_i)e^i, \quad i = 1, 2, \ldots, p, \tag{2.18}$$

where, as usual, e^i denotes the ith unit vector of $\mathbb{R}^p$, and

$$G \subset \overline{B} \subset B, v \in B \setminus \overline{B}. \tag{2.19}$$

Remark 2.7. By Proposition 2.3, the point y^w as described in Proposition 2.6 belongs to G_E. From (2.18), it is easy to see that $v^i \neq v$ for all $i = 1, \ldots, p$ and for each $i = 1, \ldots, p$, the vertex $v^i \in B^0 \setminus G$ because $y^w \geq v^i$ and $y^w \neq v^i$. The points $v^1, \ldots, v^p$ are called *new vertices corresponding to y^w*.

2.2. The Algorithm

After the initialization step, the algorithm can execute the iteration step many times to generate a number of efficient outcome points for problem (CMP) depending on the user's requirements. Note that the construction of the box B^0 in Substep 1.1 of Algorithm 1 involves solving p convex programming problems, each of which has a simple linear objective function and the same feasible region. Let $NumExpect$ be a positive integer number. The algorithm for generating $NumExpect$ efficient outcome points to the problem (VP_X) and $NumExpect$ efficient solutions associated with them are described as follows.

Algorithm 1.

Step 1 (Initialization).

(1.1) Construct $B^0 = [y^m, y^M]$, where y^m and y^M are described in Section 2.1. Set $B = B^0$.

(1.2) Set $Y^{\text{out}}_E = \emptyset$ (the set of efficient outcome points),
$X^{\text{out}}_E = \emptyset$ (the set of efficient solutions),
Nef := $NumExpect$,
$k := 0$ (the number of elements of the set Y^{out}_E),
$S = \{y^m\}$.

Step 2 (Iteration). See Steps 2.1 through 2.3 below.

(2.1) Set $\overline{S} = \emptyset$.

(2.2) **For** each $\overline{y} \in S$ **do**

```
        begin
        $k := k + 1$.
        Find an optimal solution $(x^*, \lambda^*) \in \mathbb{R}^{n+1}$ and the optimal value $\lambda^*$ to the problem $(T(\overline{y}))$.
          Set $w^k = \overline{y} + \lambda^*(y^M - \overline{y})$,
          $Y_E^{\text{out}} = Y_E^{\text{out}} \cup \{w^k\}$ and $X_E^{\text{out}} = X_E^{\text{out}} \cup \{x^*\}$ (see Remark 2.5),
          $\overline{B} = B \setminus [v, y^w]$ with $v = \overline{y}$ and $y^w = w^k$.
        Determine $p$ vertices $v^1, \ldots, v^p$ corresponding to $y^w$ via formula (2.18).
        Set $\overline{S} = \overline{S} \cup \{v^1, \ldots, v^p\}$ and $B = \overline{B}$.
        end
  (2.3) If $k \geq$ Nef Then Terminate the algorithm
        Else  Set $S = \overline{S}$ and return the Step 2.
```

2.3. Example

To illustrate the Algorithm 1, we consider the convex multiobjective programming problem (VP_{X}) (see, Benson in [14]) with $p = n = 2$, where

$$f_1(x) = (x_1 - 2)^2 + 1, \qquad f_2(x) = (x_2 - 4)^2 + 1, \tag{2.20}$$

and $\text{X} \subset \mathbb{R}^2$ that satisfies the constraints

$$g_1(x) = 25x_1^2 + 4x_2^2 - 100 \leq 0, \qquad g_2(x) = x_1 + 2x_2 - 4 \leq 0. \tag{2.21}$$

What follows is a brief summary of the results of executing Algorithm 1 for determining $NumExpect = 7$ different efficient outcome points for this sample problem, and 7 efficient solutions associated with them.

Step 1. By solving the problem $(P(1))$, we obtain the optimal solution $(1.0, 0.0, 1.0, 100.0)$ and the optimal value $y_1^{\text{opt}} = 1.0$. Then, $y^1 = (1.0, 100.0)$ is the optimal solution for the problem $(P^0(1))$.

By solving problem $(P(2))$, we obtain the optimal value $y_2^{\text{opt}} = 2.380437$ and the optimal solution $(-1.6501599, 2.8250799, 100.0, 2.380437)$. Hence, $y^2 = (100.0, 2.380437)$ is the optimal solution for the problem $(P^0(2))$. From (2.4), we choose

$$\alpha = 110.0 > \max\left\{y_1^1, y_2^1, y_1^2, y_2^2\right\} = \max\{1.0, 100.0, 100.0, 2.380437\}. \tag{2.22}$$

Then $y^m = (y_1^{\text{opt}}, y_2^{\text{opt}}) = (1.0, 2.380437)$, and $y^M = (\alpha, \alpha) = (110.0, 110.0)$. Set $B = B^0 = [y^m, y^M]$, $Y_E^{\text{out}} := \emptyset$, $X_E^{\text{out}} := \emptyset$, Nef $:= NumExpect = 7$; $k := 0$ and $S := \{y^m\}$.

Step 2 (The first time).

Substep 2.1. Set $\overline{S} = \emptyset$.

Substep 2.2 (The set S has only one element (1.0, 2.380437)). Let $\overline{y} := (1.0, 2.380437)$ and $k = 1$. By solving the problem $(T(\overline{y}))$, we obtain the optimal solution $(x^*, \lambda^*) = (0.230078, 1.884961, 0.028740)$ and the optimal value $\lambda^* = 0.028740$. Then,

$$w^1 = \overline{y} + \lambda^* \left(y^M - \overline{y} \right) = (4.132625, 5.473389) \tag{2.23}$$

(the first efficient outcome point),

$$x^1 = x^* = (0.230078, 1.884961) \tag{2.24}$$

(the first efficient solution).

Set

$$\begin{gathered} Y_E^{\text{out}} := Y_E^{\text{out}} \cup \left\{ w^1 \right\} = \left\{ w^1 \right\}, \\ X_E^{\text{out}} := X_E^{\text{out}} \cup \left\{ x^1 \right\} = \left\{ x^1 \right\}, \\ \overline{B} = B \setminus \left[v, y^w \right] \quad \text{with } v = \overline{y}, y^w = w^1. \end{gathered} \tag{2.25}$$

Using formula (2.18), we obtain $p = 2$ new vertices $v^1 = (1.0, 5.473389)$ and $v^2 = (4.132625, 2.380437)$ corresponding to y^w.

Set $\overline{S} = \overline{S} \cup \{v^1, v^2\} = \{(1.0, 5.473389), (4.132625, 2.380437)\}$ and $B = \overline{B}$.

Substep 2.3. Since $k = 1 < \text{Nef} = 7$, set $S = \overline{S}$ and return to the Step 2.

Step 2 (The second time).

Substep 2.1. $\overline{S} := \emptyset$.

Substep 2.2. (The set S has 2 elements: $s^1 = (1.0, 5.473389)$, $s^2 = (4.132625, 2.380437)$)

(i) Let $\overline{y} = s^1 = (1.0, 5.473389)$ and $k = 2$. The problem $(T(\overline{y}))$ has the optimal solution $(x^*, \lambda^*) = (0.820117, 1.589942, 0.012772)$, and the optimal value $\lambda^* = 0.012772$. We have

$$\begin{gathered} w^2 = (2.392125, 6.808381), \quad \text{(the second efficient outcome point)}, \\ x^2 = (0.820117, 1.589942), \quad \text{(the second efficient solution)}. \end{gathered} \tag{2.26}$$

Set

$$\begin{aligned} Y_E^{\text{out}} &= Y_E^{\text{out}} \cup \left\{w^2\right\} = \left\{w^1, w^2\right\}, \\ X_E^{\text{out}} &= X_E^{\text{out}} \cup \left\{x^2\right\} = \left\{x^1, x^2\right\}, \\ \overline{B} = B \setminus &\left[v, y^w\right] \quad \text{with } v = \overline{y},\ y^w = w^2. \end{aligned} \tag{2.27}$$

Two new vertices corresponding to y^w are $v^1 = (1.0, 6.808381)$, and $v^2 = (2.392125, 5.473389)$.

Set $\overline{S} = \overline{S} \cup \{v^1, v^2\} \ = \{(1.0, 6.808381), (2.392125, 5.473389)$ and $B = \overline{B}$.

(ii) Let $\overline{y} = s^2 = (4.132625, 2.380437)$ and $k = 3$. Then, $(x^*, \lambda^*) = (-0.276796, 2.138398, 0.019375)$, and $\lambda^* = 0.019375$ are the optimal solution and the optimal value for the problem $(T(\overline{y}))$, respectively. Then,

$$\begin{aligned} w^3 &= (6.183801, 4.465562), \quad (\text{the third efficient outcome point}), \\ x^3 &= (-0.276796, 2.138398), \quad (\text{the third efficient solution}). \end{aligned} \tag{2.28}$$

Set

$$\begin{aligned} Y_E^{\text{out}} &= Y_E^{\text{out}} \cup \left\{w^3\right\} = \left\{w^1, w^2, w^3\right\}, \\ X_E^{\text{out}} &= X_E^{\text{out}} \cup \left\{x^3\right\} = \left\{x^1, x^2, x^3\right\}, \\ \overline{B} = B \setminus &\left[v, y^w\right] \quad \text{with } v = \overline{y}, \quad y^w = w^3. \end{aligned} \tag{2.29}$$

Two new vertices corresponding to y^w are $v^1 = (4.132625, 4.465562)$, and $v^2 = (6.1838011, 2.380437)$.

Set $\overline{S} = \overline{S} \cup \{v^1, v^2\} = \{(1.0, 6.808381), (2.392125, 5.473389), (4.132625, 4.465562), (6.1838011, 2.380437)\}$, and $B = \overline{B}$.

Substep 2.3. Since $k = 3 < \text{Nef} = 7$, set $S := \overline{S}$ and return to the Step 2.

Step 2 (The third time).

Substep 2.1. $\overline{S} := \emptyset$.

Substep 2.2. (The set S has 4 elements: $s^1 = (1.0, 6.808381)$, $s^2 = (2.392125, 5.473389)$, $s^3 = (4.132625, 4.465562)$, $s^4 = (6.1838011, 2.380437)$).

By a calculation analogous to above, we yield four next efficient outcome points w^4, w^5, w^6, and w^7 generated by $\overline{y} = s^1$, $\overline{y} = s^2$, $\overline{y} = s^3$, $\overline{y} = s^4$, respectively, and four next efficient solution x^4, x^5, x^6, x^7 associated with w^4, w^5, w^6, w^7, respectively. Namely,

$$\begin{aligned} w^4 &= (1.779439, 7.546285); \quad x^4 = (1.117142, 1.441430);\\ w^5 &= (3.109453, 6.170177); \quad x^5 = (0.547605, 1.726198);\\ w^6 &= (4.780944, 5.111841); \quad x^6 = (0.055535, 1.972232);\\ \mathrm{w}^7 &= (7.668913, 3.919957); \quad x^7 = (-0.582424, 2.291212). \end{aligned} \tag{2.30}$$

Since $k = \mathrm{Nef} = 7$, the algorithm is terminated. Thus, after 3 iterations, we obtain $Y_E^{\text{out}} = \{w^1, w^2, \ldots, w^7\}$, and $\mathrm{X}_E^{\text{out}} = \{x^1, x^2, \ldots, x^7\}$.

3. Application to Problem (CP_X)

Consider convex multiplicative programming problem (CP_X) associated with the convex multiobjective programming problem (VP_X)

$$\min \prod_{j=1}^{p} f_j(x) \quad \text{s.t. } x \in X, \tag{CP_X}$$

where $X \subset \mathbb{R}^n$ is a nonempty compact convex set defined by (2.1) and $f_j : \mathbb{R}^n \to \mathbb{R}$ is convex on $\mathbb{R}^n$ and positive on X, $j = 1, 2, \ldots, p$.

One of the direct reformulations of the problem (CP_X) as an outcome-space problem is given by

$$\min h(y) = \prod_{j=1}^{p} y_j \quad \text{s. t. } y \in Y^{\text{image}}, \tag{OP_Y}$$

where $Y^{\text{image}} = \{f(x) = (f_1(x), \ldots, f_p(x))^T \mid x \in X\}$ is the outcome set of X under f. By assumption, we have $Y^{\text{image}} \subset \operatorname{int} \mathbb{R}_+^p$.

The following proposition tells us a link between the global solution to the problem (OP_Y) and the efficient outcome set Y_E^{image} for the problem (VP_X).

Proposition 3.1. *Any global optimal solution to problem (OP_Y) must belong to the efficient outcome set Y_E^{image}.*

Proof. The proposition follows directly from the definition. □

Remark 3.2. We invoke Propositions 3.1 and 2.2 to deduce that problem (OP_Y) is equivalent to the following problem:

$$\min h(y) = \prod_{j=1}^{p} y_j \quad \text{s.t. } y \in G_E. \tag{OP_{GE}}$$

The relationship between two problems (CP_X) and (OP_Y) is described by the following theorem and was given in [14, Theorem 2.2]. However, we give here a full proof for the reader's convenience.

Theorem 3.3. *If y^* is a global optimal solution to problem (OP_Y), then any $x^* \in X$ such that $f(x^*) \leq y^*$ is a global optimal solution to problem (CP_X). Furthermore, the global optimal values of two problems (CP_X) and (OP_Y) are equal, that is,*

$$h(y^*) = \prod_{j=1}^{p} y_j^* = \prod_{j=1}^{p} f_j(x^*). \tag{3.1}$$

Proof. Suppose that y^* is a global optimal solution to problem (OP_Y) and $x^* \in X$ satisfies

$$f(x^*) \leq y^*. \tag{3.2}$$

By Proposition 3.1, $y^* \in Y_E^{\text{image}}$. Since $f(x^*) \in Y^{\text{image}}$ and $y^* \in Y_E^{\text{image}}$, from (3.2), we have $f(x^*) = y^*$. Thus, $h(y^*) = \prod_{j=1}^{p} y_j^* = \prod_{j=1}^{p} f_j(x^*)$. Now, we show that x^* is a global optimal solution of the problem (CP_X). Indeed, assume the contrary that there is a point $x \in X$ such that $\prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*)$. Combining this fact and (3.2) gives

$$\prod_{j=1}^{p} f_j(x) < \prod_{j=1}^{p} f_j(x^*) \leq \prod_{j=1}^{p} y_j^*. \tag{3.3}$$

Since $x \in X$, we have $y = f(x) \in Y^{\text{image}}$. Therefore, $\prod_{j=1}^{p} y_j < \prod_{j=1}^{p} y_j^*$. This contradicts the hypothesis that y^* is a global optimal solution to problem (OP_Y) and proves that x^* is a global optimal solution to problem (CP_X). This completes the proof. □

By Theorem 3.3 and Proposition 3.1, solving the problem (CP_X) can be carried out in two stages:

(1) finding a global optimal solution y^* to the problem (OP_{G_E}). Then y^* is also a global optimal solution to the problem (OP_Y),

(2) finding a global optimal solution $x^* \in X$ to the problem (CP_X) which satisfies $f(x^*) \leq y^*$.

In the next section, the outer approximation algorithm is developed for solving the problem (OP_{G_E}).

3.1. Outer Approximation Algorithm for Solving Problem (OP_{G_E})

Starting with the polyblock $B^0 = [y^m, y^M]$ (see Section 2.1), the algorithm will iteratively generate a sequence of reverse polyblocks B^k, $k = 1, 2, \ldots$, such that

$$B^0 \supset B^1 \supset B^2 \supset \cdots \supset G \supset G_E. \tag{3.4}$$

For each $k = 0, 1, 2, \ldots$, the new reverse polyblock B_{k+1} is constructed via the formula

$$B^{k+1} := B^k \setminus [v, y^w], \tag{3.5}$$

where $v = y^k$, y^k is a global optimal solution to the problem $\min\{h(y) \mid y \in B^k\}$, and y^w is the efficient outcome point generated by $\overline{y} = y^k$.

For each k, let V^k denote the vertex set of the reverse polyblock B^k. The following Proposition 3.4 shows that the function $h(y)$ achieves a minimum over the reverse polyblock B^k at a proper vertex.

Proposition 3.4. *Let $h(y) = \prod_{j=1}^{p} y_j$, and let B^k be a reverse polyblock. Consider the problem to minimize $h(y)$ subject to $y \in B^k$. An optimal solution y^k to the problem then exists, where y^k is a proper vertex of B^k.*

Proof. By definition, note that the objective function $h(y)$ is a continuous function on $\mathbb{R}^p$, and B^k is a compact, and the problem $\min\{h(y) \mid y \in B^k\}$ has an optimal solution $y^k \in B^k$. For each $y \in B^k$, there is a proper vertex v of B^k such that $y \in [v, y^M]$. That means $v \leq y$. By definition of the function $h(y)$, we have $h(v) \leq h(y)$. This shows that $h(y^k) = \min\{h(y) \mid y \in V^k\}$, where V^k is the vertex set of B^k, and the proof is completed. □

Remark 3.5. By Proposition 3.4, instead of solving problem $\min\{h(y) \mid y \in B^k\}$, we solve the simple problem $\min\{h(y) \mid y \in V^k\}$. From (3.4), it is clear that for each $k = 0, 1, 2 \ldots$, the optimal value

$$\beta_k = \min\left\{ h(y) = \prod_{j=1}^{p} y_j \;\middle|\; y \in B^k \right\} \tag{3.6}$$

is a lower bound for the problem (OP_{GE}), and $\{\beta_k\}$ is an increasing sequence, that is, $\beta_{k+1} > \beta_k$ for all $k = 0, 1, 2, \ldots$.

Let ε be a given sufficient small real number. Let $y^* \in G_E$. Then $h(y^*)$ is an upper bound for the problem (OP_{GE}). A point y^* is said to be an *ε-optimal solution* to problem (OP_{GE}) if there is a lower bound β_* for this problem such that $h(y^*) - \beta_* < \varepsilon$.

Below, we will present an algorithm for finding ε-optimal solution to problem (OP_{GE}).

At the beginning of a typical iteration $k \geq 0$ of the algorithm, we have from the previous iteration an available nonempty reverse polyblock $B^k \subset \mathbb{R}^p$ that contains G and an upper bound θ_k for the problem (OP_{GE}). In iteration k, firstly problem $\min\{h(y) \mid y \in V_k\}$ is solved to obtain the optimal solution set T_k^{opt}. By the construction, $T_k^{\text{opt}} \subset (B^0 \setminus G)$. Take any $y^k \in T_k^{\text{opt}}$. The optimal value $\beta_k = h(y^k)$ is *the best currently lower bound*. Then, we solve the convex programming problem $(T(\overline{y}))$ with $\overline{y} := y^k$ to receive the optimal value λ^*. By Proposition 2.3, the feasible solution $\omega^k = \overline{y} + \lambda^*(y^M - \overline{y}) \in G_E$ is an outcome efficient point generated by $\overline{y} = y^k$ (see Remark 2.5). Now, *the best currently upper bound* is $\theta_k = \min\{\theta_k, h(\omega^k)\}$ and the feasible solution y^{best} satisfying $h(y^{\text{best}}) = \theta_k$ is said to be a *currently best feasible solution.* If $\theta_k - \beta_k < \varepsilon$, then the algorithm terminates, and y^{best} is

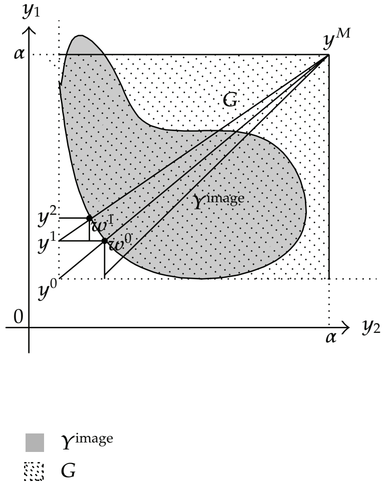

Figure 3

an ε*-optimal solution for the problem* (OP_{GE}). Otherwise, set $B^{k+1} := B^k \setminus [v, y^w]$, where $v := y^k$ and $y^\omega = \omega^k$. According to Proposition 2.6, the vertex set V^{k+1} of the reverse polyblock B^{k+1} is

$$V^{k+1} = \left(V^k \setminus \{v\}\right) \cup \left\{v^1, v^2, \dots, v^p\right\}, \tag{3.7}$$

where $v^1, \dots, v^p$ are determined by formula (2.18). Figure 3 illustrates two beginning steps of the algorithm in the case of $p = 2$.

By the construction, it is easy to see that $\{\theta_k\}$ (the sequence of upper bounds for the problem (OP_{GE}) is the nonincreasing sequence, that is, $\theta_{k+1} \leq \theta_k$ for all $k = 0, 1, 2, \dots$

Now, the outer approximation algorithm for solving (OP_{GE}) is stated as follows.

Algorithm 2 (Outer approximation algorithm).

Initialization Step

Construct $B^0 = [y^m, y^M]$, where y^m and y^M are described in Section 2.1. Choose $\varepsilon > 0$ ($\varepsilon > 0$ is a sufficient small number). Set $V^0 = \{y^m\}$ and $\theta_0 := Numlarge$ ($Numlarge$ is a sufficient large number. This number can be viewed as an initialization *the upper bound*). Set $k = 0$ and go to Iteration step k.

Iteration Step k $[k = 0, 1, 2, \dots]$. See Steps *k.1* through *k.5* below.

k.1 Determine the optimal solution set $T_k^{\text{opt}} := \operatorname{Argmin}\{h(y) \mid y \in V^k\}$. Choose an arbitrary $y^k \in T_k^{\text{opt}}$ and set $\beta_k := h(y^k)$ (*currently best lower bound*).

k.2 Let $\overline{y} = y^k$. Find the optimal value λ^* to the problem $(T(\overline{y}))$. And set $\omega^k = \overline{y} + \lambda^*(y^M - \overline{y}) \in G_E$.

k.3 (Update the upper bound)

If $h(\omega^k) < \theta_k$ **Then** $\theta_k = h(\omega^k)$ (*currently best upper bound*)

and $y^{\text{best}} = \omega^k$ (*currently best feasible point*).

```
$k.4$ If $\theta_k - \beta_k \leq \varepsilon$
    Then Terminate the algorithm: $y^{\text{best}}$ is an $\varepsilon$-optimal solution
    Else Set $B^{k+1} := B^k \setminus [v, y^w]$ where $v := y^k$ and $y^\omega = \omega^k$
        and determine the set $V^{k+1}$ by formula (2.18).
$k.5$ Set $\theta_{k+1} := \theta_k$; $k := k + 1$ and go to iteration step $k$.
```

Theorem 3.6. *The algorithm terminates after finitely many steps and yields an ε-optimal solution to problem (OP_{GE}).*

Proof. Let ε be a given positive number. Since the function $h(y)$ is uniformly continuous on the compact set B^0, we can choose an enough small number $\delta > 0$ such that if $y, y' \in B_0$ and $\|y - y'\| < \delta$, then $|h(y) - h(y')| < \varepsilon$. Then, to prove the termination of the algorithm, we need to show only that

$$\lim_{k \to \infty} \left\| w^k - y^k \right\| = 0. \tag{3.8}$$

Observe first that the positive series $\sum_{k=1}^{\infty} \operatorname{Vol}[y^k, w^k]$ is convergent, since the open boxes int $[y^k, w^k]$ are disjoint, and all of them are contained in the closure of $B^0 \setminus G$. It follows that

$$\lim_{k \to \infty} \operatorname{Vol}\left[y^k, w^k\right] = 0. \tag{3.9}$$

Now, by the construction of w^k and y^k, we have

$$w^k - y^k = \lambda_k \left(y^M - y^k\right), \tag{3.10}$$

$$\operatorname{Vol}\left[y^k, w^k\right] = \lambda_k^p \operatorname{Vol}\left[y^k, y^M\right], \tag{3.11}$$

for $\lambda_k \in (0, 1)$. Note that all of the points y^k are contained in the set $B^0 \setminus G$. Furthermore, by the choice of y^M (see (2.4)), the closure of $B^0 \setminus G$ is a compact subset of the interior of the cone $y^M - \mathbb{R}^p_+$. This observation implies that $\operatorname{Vol}[y^k, y^M]$ has a lower bound far from zero. Combining this fact, (3.9) and (3.11) imply that $\lim_{k \to \infty} \lambda_k = 0$. Also, the observation implies that $\|y^M - y^k\|$ is bounded. Finally, by (3.10), we have $\lim_{k \to \infty} \|w^k - y^k\| = 0$. □

3.2. Computational Results

First, the Algorithm 2 has been applied for solving the test example given by Benson in [14] (see the example in Section 2.3)

$$\min\{f_1(x) f_2(x) \mid g_1(x) \leq 0, g_2(x) \leq 0\}, \tag{3.12}$$

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

The calculation process for solving this example is described as follows.

Initialization.

Similarly, the example in Section 2.3, we have $B^0 = [y^m, y^M]$ with $y^m = (1.00000, 2.380437)$, and $y^M = (110.000000, 110.000000)$. We choose $\varepsilon = 0.025$ and set $V^0 = \{y^m = (1.00000, 2.380437)\}$, $\theta_0 = 10000$ (the initialization upper bound), $k = 0$ and go to iteration step $k = 0$.

Iteration $(k = 0)$.

Step 1. We have $T_0^{\text{opt}} := \text{argmin}\{h(y) \mid y \in V^0\} = \{y^m\}$. Choose $y^0 := y^m \in T_0^{\text{opt}}$, and set $\beta_0 := h(y^0) = 2.380437$ (currently best lower bound).

Step 2. Let $\overline{y} = y^0$. Solving the problem $(T(\overline{y}))$, we obtain the optimal solution $(x^*, \lambda^*) = (0.230078, 1.884961, 0.028740))$ and the optimal value $\lambda^* = 0.028740$. Then, $w^0 = \overline{y} + \lambda^*(y^M - \overline{y}) = (4.132625, 5.473389) \in G_E$.

Step 3. Since $h(w^0) = 22.619464 < \theta_0$, we have $\theta_0 = h(w^0) = 22.619464$ (currently best upper bound), and $y^{\text{best}} = w^0$ (currently best feasible point).

Step 4. Since $\theta_0 - \beta_0 = 20.239027 > \varepsilon$, we set $B^1 := B^0 \setminus [v, y^w]$, where $v := y^0$, $y^w = w^0$ and determine the set

$$V^1 = \{(1.000000, 5.473389), (4.132625, 2.380437)\}. \tag{3.13}$$

Step 5. Set $\theta_1 := \theta_0 = 22.619464$; $k := 0 + 1 = 1$, and go to iteration step $k = 1$.

Iteration $(k = 1)$.

Step 1. We have $T_1^{\text{opt}} := \text{argmin}\{h(y) \mid y \in V^1\} = \{y^1 = (1.000000, 5.473389)\}$. Choose $y^1 \in T_1^{\text{opt}}$ and set $\beta_1 := h(y^1) = 5.473389$ (the currently lower bound).

Step 2. Let $\overline{y} = y^1$. The convex programming $(T(\overline{y}))$ has the optimal solution $(x^*, \lambda^*) = (0.82011, 1.589942, 0.012772))$, and the optimal value $\lambda^* = 0.012772$. Then, we have $w^1 = (2.392125, 6.808381) \in G_E$.

Step 3. Since $h(w^1) = 16.286498 < \theta_1$, we set $\theta_1 = h(w^1)$ (the currently best upper bound), and $y^{\text{best}} = w^1$ (the currently best feasible point).

Step 4. Now, $\theta_1 - \beta_1 = 10.813109 > \varepsilon$. Therefore, $B^2 := B^1 \setminus [v, y^w]$, where $v := y^1$, $y^w = w^1$, and $V^2 = \{(4.132625, 2.380437), (1.000000, 6.808381), (2.392125, 5.473389)\}$.

Step 5. Set $\theta_2 = \theta_1 = 16.286498$; $k := 1 + 1 = 2$, and go to iteration step $k = 2$.

Iteration $(k = 2)$.

Step 1. We have $T_2^{\text{opt}} = \text{argmin}\{h(y) \mid y \in V^2\} = \{y^2 = (1.000000, 6.808381)\}$, choose $y^2 := (1.000000, 6.808381) \in T_2^{\text{opt}}$, and set $\beta_2 := h(y^2) = 6.808381$.

Step 2. Let $\overline{y} = y^2$. The problem $(T(\overline{y}))$ has the optimal solution $(x^*, \lambda^*) = (1.117142, 1.441429, 0.007151))$, and the optimal value $\lambda^* = 0.007151$. Then $w^2 = (1.779439, 7.546285) \in G_E$.

Step 3. Since $h(w^2) = 13.428154 < \theta_2$, the currently best upper bound $\theta_2 := h(w^2)$ and the currently best feasible point $y^{\text{best}} = w^2 = (1.779439, 7.546285)$.

Step 4. Since $\theta_2 - \beta_2 = 6.619773 > \varepsilon$, we set $B^3 := B^2 \setminus [v, y^w]$, where $v := y^2$, $y^w = w^2$, and determine the set $V^3 = \{(4.132625, 2.380437), (2.392125, 5.473389), (1.000000, 7.546285), (1.779439, 6.808381)\}$.

Step 5. Set $\theta_3 := \theta_2 = 13.428154$, $\quad k := 2 + 1 = 3$, and go to iteration step $k = 3$.

After 42 iterations, the algorithm terminates with $\theta_{42} = 9.770252094$ and $\beta_{42} = 9.745596873$, where $\theta_{42} = h(w^{20})$. Then, the ε-optimal solution for the problem (OP_Y) and for the problem (CP_X) are given by

$$y^* = y^{\text{best}} = w^{20} = (1.023846379, 9.542693410), \tag{3.14}$$

and $x^* = (1.845577271, 1.077211364)$. The approximate optimal value of problem (3.12) is 9.770252094.

Below, we present the results of computational experiment for two types of problem. We take $\varepsilon = 0.025$ in the numerical experiments. We tested our algorithm for each problem with random data for several times. Then, we took results in average. Numerical results are summarized in Tables 1 and 2.

Type 1.

$$\begin{gathered} \min \prod_{j=1}^{p} f_j(x), \\ \text{s.t. } Ax \leq b, \\ x \geq 0, \end{gathered} \tag{3.15}$$

where $f_j(x) = \langle \alpha^j, x \rangle + \mu_j$, $\alpha^j \in \mathbb{R}^n$, $\mu_{\mathrm{j}} \in \mathbb{R}$, $j = 1, \ldots, p$, A is an $(m \times n)$-matrix, and $b \in \mathbb{R}^m$.

Type 2.

$$\begin{gathered} \min \prod_{j=1}^{p} f_j(x), \\ \text{s.t. } Ax \leq b, \\ x \geq 0, \end{gathered} \tag{3.16}$$

where $f_j(x) = (\alpha_1^j x_1 + \beta_1^j)^2 + \cdots + (\alpha_n^j x_n + \beta_n^j)^2 + \nu_j$, with $\alpha^j, \beta^j \in \mathbb{R}^n$, $\nu_j \in \mathbb{R}$, $j = 1, \ldots, p$, A is an $(m \times n)$-matrix, and $b \in \mathbb{R}^m$.

Table 1: Computational results on the problem of Type 1.

Problem no.	p	m	n	Average number of iterations	Average CPU time (sec)
1	3	10	10	11	0.83
2	3	20	20	9.9	2.08
3	3	30	50	19.2	19.4
4	3	40	60	17.5	24.42
5	3	60	100	26.2	109.6
6	4	20	20	23.1	4.3
7	4	20	30	14.1	4.87
8	4	30	50	26.2	20
9	4	40	60	59.6	61.64
10	4	60	100	36.6	142.32

Table 2: Computational results on the problem of Type 2.

Problem no.	p	m	n	Average number of iterations	Average CPU time (sec)
1	3	20	20	14	5.18
2	3	20	30	26.9	14.64
3	3	30	50	40	51.35
4	3	40	60	42.6	77.38
5	3	60	100	59.5	316.79
6	4	20	20	34.8	7.57
7	4	20	30	89.8	35.27
8	4	30	50	80.7	86.83
9	4	40	60	78.4	112.23
10	4	60	100	85.5	372.59

Our program was coded in Matlab R2007a and was executed on our own PC with the configuration, CPU Intel Core 2 T5300 1.73 GHz, 1 G Ram. On the two above problems, the parameters are defined as follows:

(i) $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ is a randomly generated matrix with elements in $[-1, 1]$;

(ii) $b = (b_1, \ldots, b_m)^T$ is a random vector satisfying the formula

$$b_i = \sum_{j=1}^{n} a_{ij} + b_{0i} \tag{3.17}$$

with b_{0i} being a random number in $[0, 2]$ for $i = 1, \ldots, m$;

(iii) $\alpha^j = (\alpha_1^j, \ldots, \alpha_n^j)$ and $\beta^j = (\beta_1^j, \ldots, \beta_n^j)$, $j = 1, \ldots, p$ are vectors with elements randomly distributed on $[0, 1]$;

(iv) the coefficients μ_j, ν_j, $j = 1, \ldots, p$ are uniformly distributed on $[0, 1]$.

4. Conclusion

In this paper, we have presented Algorithm 1 for generating a finite set Y_E^{out} of efficient outcome points for the convex multiobjective programming problem (VP_X). The number of such points depends on the requirement of the decision makers. When the selected number is large enough, the convex set $\mathrm{conv}(Y_E^{\mathrm{out}} + \mathbb{R}_+^p)$, where $\mathrm{conv}(Y_E^{\mathrm{out}} + \mathbb{R}_+^p)$ is convex hull of the set $(Y_E^{\mathrm{out}} + \mathbb{R}_+^p)$, may be viewed as an inner approximation of $G^0 = Y + \mathbb{R}_+^p$ and the efficient set $(\mathrm{conv}(Y_E^{\mathrm{out}} + \mathbb{R}_+^p))_E$ may be viewed as an inner approximation of the efficient outcome set Y_E^{image} for the problem (VP_X). For each efficient outcome point $y^w \in Y_E^{\mathrm{image}}$ to be generated by a given $\overline{y} \in B^0 \setminus G$, the algorithm calls for solving a convex programming problem with linear objective function $(T(\overline{y}))$. Note that by solving problem $(T(\overline{y}))$, we also obtain the efficient solution x^* associated with y^w, where (x^*, λ^*) is the optimal solution for the problem $(T(\overline{y}))$.

In [7], Ehrgott et al. have proposed an outer approximation algorithm for representing an inner approximation of G_0. This algorithm is the combination of an extension of Benson's outer approximation algorithm [1] for multiobjective linear programming problems and linearization technique. In each iteration step of these algorithms, a polyhedron is obtained from the previous one by adding a new hyperplane to it. The vertices of the new polyhedron can be calculated from those of the previous polyhedron by some available global optimization methods. Unlike those algorithms, our algorithm constructs a reverse polyblock at each iteration step from the previous one by cutting out a box, and its vertices can be easily determined by the formula (2.18).

As an application, we have proposed the outer approximation algorithm (Algorithm 2) for solving the convex multiplicative programming problem (CP_X) associated with the problem (VP_X) in outcome space. Since the number of terms p in the objective function of problem (VP_X) is, in practice, often much smaller than the number of variables n, we hope that the algorithms help to reduce considerably the size of the problems.

Acknowledgments

The authors would like to express their sincere gratitude to the referees for their valuable comments and suggestions which enable them to improve substantially this paper. This paper is supported by NAFOSTED, Vietnam.

References

[1] H. P. Benson, "An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem," *Journal of Global Optimization*, vol. 13, no. 1, pp. 1–24, 1998.

[2] J. Jahn, *Vector Optimization: Theory, Applications and Extensions*, Springer, Berlin, Germany, 2004.

[3] M. T. Tabucanon, "Multiobjective programming for industrial engineers," in *Mathematical Programming for Industrial Engineers*, M. Avriel et al., Ed., vol. 20 of *Indust. Engrg.*, pp. 487–542, Marcel Dekker, New York, NY, USA, 1996.

[4] P. Armand, "Finding all maximal efficient faces in multiobjective linear programming," *Mathematical Programming*, vol. 61, no. 3, pp. 357–375, 1993.

[5] H. P. Benson and E. Sun, "Outcome space partition of the weight set in multiobjective linear programming," *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 17–36, 2000.

[6] A. Chinchuluun and P. M. Pardalos, "A survey of recent developments in multiobjective optimization," *Annals of Operations Research*, vol. 154, pp. 29–50, 2007.

[7] M. Ehrgott, L. Shao, and A. Schöbel, "An approximation algorithm for convex multi-objective programming problems," *Journal of Global Optimization*, vol. 50, no. 3, pp. 397–416, 2011.

[8] N. T. B. Kim and D. T. Luc, "Normal cones to a polyhedral convex set and generating efficient faces in linear multiobjective programming," *Acta Mathematica Vietnamica*, vol. 25, no. 1, pp. 101–124, 2000.

[9] N. T. B. Kim and D. T. Luc, "Normal cone method in solving linear multiobjective problems," *Journal of Statistics & Management Systems*, vol. 5, no. 1-3, pp. 341–358, 2002.

[10] N. T. B. Kim, N. T. Thien, and L. Q. Thuy, "Generating all efficient extreme solutions in multiple objective linear programming problem and its application to multiplicative programming," *East-West Journal of Mathematics*, vol. 10, no. 1, pp. 1–14, 2008.

[11] Z. H. Lin, D. L. Zhu, and Z. P. Sheng, "Finding a minimal efficient solution of a convex multiobjective program," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 587–600, 2003.

[12] D. T. Luc, T. Q. Phong, and M. Volle, "Scalarizing functions for generating the weakly efficient solution set in convex multiobjective problems," *SIAM Journal on Optimization*, vol. 15, no. 4, pp. 987–1001, 2005.

[13] W. Song and G. M. Yao, "Homotopy method for a general multiobjective programming problem," *Journal of Optimization Theory and Applications*, vol. 138, no. 1, pp. 139–153, 2008.

[14] H. P. Benson, "An outcome space branch and bound-outer approximation algorithm for convex multiplicative programming," *Journal of Global Optimization*, vol. 15, no. 4, pp. 315–342, 1999.

[15] B. Jaumard, C. Meyer, and H. Tuy, "Generalized convex multiplicative programming via quasiconcave minimization," *Journal of Global Optimization*, vol. 10, no. 3, pp. 229–256, 1997.

[16] N. T. B. Kim, "Finite algorithm for minimizing the product of two linear functions over a polyhedron," *Journal of Industrial and Management Optimization*, vol. 3, no. 3, pp. 481–487, 2007.

[17] N. T. B. Kim, N. T. L. Trang, and T. T. H. Yen, "Outcome-space outer approximation algorithm for linear multiplicative programming," *East-West Journal of Mathematics*, vol. 9, no. 1, pp. 81–98, 2007.

[18] L. D. Muu and B. T. Tam, "Minimizing the sum of a convex function and the product of two affine functions over a convex set," *Optimization*, vol. 24, no. 1-2, pp. 57–62, 1992.

[19] N. V. Thoai, "A global optimization approach for solving the convex multiplicative programming problem," *Journal of Global Optimization*, vol. 1, no. 4, pp. 341–357, 1991.

[20] H. Tuy and N. D. Nghia, "Reverse polyblock approximation for generalized multiplicative/fractional programming," *Vietnam Journal of Mathematics*, vol. 31, no. 4, pp. 391–402, 2003.

[21] P. L. Yu, *Multiple-Criteria Decision Making*, vol. 30 of *Mathematical Concepts and Methods in Science and Engineering*, Plenum Press, New York, NY, USA, 1985.

[22] R. T. Rockafellar, *Convex Analysis*, Princeton Mathematical Series, no. 28, Princeton University Press, Princeton, NJ, USA, 1970.

[23] D. T. Luc, *Theory of Vector Optimization*, vol. 319 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Berlin, Germany, 1989.

[24] A. M. Geoffrion, "Proper efficiency and the theory of vector maximization," *Journal of Mathematical Analysis and Applications*, vol. 22, pp. 618–630, 1968.