*Journal of Global Optimization* **13:** 1–24, 1998.



# An Outer Approximation Algorithm for Generating All Efficient Extreme Points in the Outcome Set of a Multiple Objective Linear Programming Problem

HAROLD P. BENSON
*College of Business Administration, Department of Decision and Information Sciences, University of Florida, P.O. Box 117169, Gainesville, FL 32611–7169, USA*
*E-mail: cyeonwc@nervm.nerdc.ufl.edu*

(Received 21 March 1997; accepted 25 November 1997)

**Abstract.** Various difficulties have been encountered in using decision set-based vector maximization methods to solve a multiple objective linear programming problem (MOLP). Motivated by these difficulties, some researchers in recent years have suggested that outcome set-based approaches should instead be developed and used to solve problem (MOLP). In this article, we present a finite algorithm, called the Outer Approximation Algorithm, for generating the set of all efficient extreme points in the outcome set of problem (MOLP). To our knowledge, the Outer Approximation Algorithm is the first algorithm capable of generating this set. As a by-product, the algorithm also generates the weakly efficient outcome set of problem (MOLP). Because it works in the outcome set rather than in the decision set of problem (MOLP), the Outer Approximation Algorithm has several advantages over decision set-based algorithms. It is also relatively easy to implement. Preliminary computational results for a set of randomly-generated problems are reported. These results tangibly demonstrate the usefulness of using the outcome set approach of the Outer Approximation Algorithm instead of a decision set-based approach.

**Key words:** Efficient set, Global optimization, Multiple objective linear programming, Outer approximation, Vector maximization

## 1. Introduction and motivation

In a multiple objective mathematical programming problem (P), the goal is to simultaneously maximize $p \geqslant 2$ noncomparable objective functions over a nonempty feasible region $X$ in $\Re^n$. To help the decision maker (DM) find a most preferred solution to problem (P), researchers have shown that one can generally restrict one's attention to the subset of feasible solutions called the *efficient* (or *nondominated*) *decision set*. Motivated by this result, researchers have developed a variety of methods for generating all, or at least some, of the efficient decision set for the DM to examine. The hope is that the DM will thereby be able to detect inherent tradeoffs among the objective functions and choose a most preferred solution. Included among these approaches are the *vector maximization* approach, *interactive* approaches, and several others; see, e.g., the books and survey papers by Cohon [1], Evans [2], Goicoechea et al. [3], Luc [4], Sawaragi et al. [5], Steuer [6], Yu [7], and Zeleny [8] and references therein.

The vector maximization concept dates from the 1950s [9], but the approach was not explored in earnest until almost 20 years later [10–13]. The goal in the vector maximization approach is to generate either all of the efficient decision set, or a representative portion thereof, without any input from the DM. Subsequently, the entire set generated is presented to the DM who, without further aid from the analyst, seeks a most preferred solution from it.

In practice, the vector maximization approach has had some success in aiding the DM to solve problem (P), but this success has been relatively limited. The primary reason for this is that the efficient decision set of problem (P) is generally a complicated, nonconvex set that grows rapidly as the size of the problem increases. Consequently, generating this set in its entirety is possible only in certain special cases; see, e.g., Yu and Zeleny [13], Benson [14], Isermann [15], Bitran [16], Villarreal and Karwan [17], and Kostreva and Wiecek [18]. Even in these special cases, the computational effort required to generate all of the efficient decision set becomes rapidly unmanageable and seems to grow exponentially with problem size; see, for instance, Steuer [6], Evans and Steuer [12], and Marcotte and Soland [19]. Furthermore, the sheer size of the efficient decision set often becomes so huge that it either becomes too difficult to describe to the DM in a meaningful way or it can overwhelm the DM to the extent that he or she is not able to choose a most preferred solution from it [20].

When $X$ is a polyhedral set and the $p$ objective functions of problem (P) are linear functions $\langle c_i, x\rangle$, $i = 1, 2, \ldots, p$, where $c_i \in \Re^n$ for each $i$, then problem (P) is called a *multiple objective linear programming problem*. The problem may then be written

$$\text{(MOLP)} \quad \text{VMAX}: \ Cx, \quad \text{s.t. } x \in X,$$

where $C$ is the $p \times n$ matrix whose $i$th row contains the vector $c_i$ for each $i = 1, 2, \ldots, p$. Problem (MOLP) is one of the simpler and more common cases of the multiple objective mathematical programming problem (P). It has been studied in literally hundreds of articles, chapters in books, and books; see, e.g., Armand [23], Armand and Malivert [24], Benson [21, 25, 26], Benson and Aksoy [27], Ecker and Kouada [28], Ecker et al. [29], Gal [30], Zeleny [31], also [1–8, 11–13, 15, 20] and references therein. Nevertheless, vector maximization approaches for problem (MOLP) have also had only limited success.

In the case of problem (MOLP), the efficient decision set $X_E$ consists of a union of faces of $X$. While $X_E$ is also always a connected set, generally, it is a complicated nonconvex subset of the boundary of $X$ [6, 7, 21, 22]. The vector maximization methods for problem (MOLP) fall into two classes. One class consists of methods that generate the entire efficient decision set $X_E$ of problem (MOLP), while the second class consists of methods for generating only the set of all extreme points of $X$ that belong to $X_E$.

Some of the most well-known algorithms for generating all of $X_E$ can be found in [6, 7, 13, 15, 22–24, 29–31]. These algorithms employ various search schemes to iteratively identify and test faces of $X$ for efficiency. Regardless of the schemes used, these methods have met with only limited success in practice. There are at least two reasons for this.

First, the computational demands of finding all of $X_E$ grow rapidly with problem size, so that mathematically only relatively-small problems can be analyzed; see, e.g., Benson and Sayin [32] and [6, 22]. Second, the sheer size and nature of $X_E$ have so far precluded the possibility of finding a concrete, useful way of presenting it in its entirety to the DM without overwhelming him or her; see, e.g., [20, 22, 32].

Let $X_{\text{ex}}$ denote the set of all extreme points of $X$. The second class of vector maximization methods for problem (MOLP) consists of algorithms for generating all of $X_E \cap X_{\text{ex}}$, that is, all of the efficient extreme points in the decision set $X$. Representative algorithms of this type can be found in Steuer [33] and in [2, 3, 6–8, 11–13, 28, 31]. The rationale for this approach is that since $X_E \cap X_{\text{ex}}$ is a finite, discrete set that is smaller than all of $X_E$, it ought to be more computationally practical to generate it and to present it to the DM without overwhelming him or her than $X_E$.

Unfortunately, in practice, methods for problem (MOLP) that seek to generate $X_E \cap X_{\text{ex}}$ have also achieved only limited success. There are at least two major reasons for this.

First, although $X_E \cap X_{\text{ex}}$ is smaller than $X_E$, it was soon found that the number of elements in $X_E \cap X_{\text{ex}}$ generally grows exponentially with problem size. As a result, as the size of problem (MOLP) grows, $X_E \cap X_{\text{ex}}$, like $X_E$, can quickly become computationally burdensome to generate, and its sheer size can easily overwhelm the DM [6, 34–38]. For example, we used the ADBASE algorithm of Steuer [33] on randomly-generated problems with four objective functions. We found that when $n = 30$ and $X$ is described by 25 linear inequalities, the average number of efficient extreme points in $X$ in a set of 10 randomly-generated problems was 7245.90. When we increased $n$ to 50 and $X$ was described by 50 linear inequalities, this average jumped to $83,780.60$ points. With $n = 60$ and with 50 linear inequalities describing $X$, each of the 10 random problems that we generated exceeded the solution capacity of ADBASE, indicating that the number of efficient extreme points in each of these problems exceeded $200,000$.

Second, most algorithms for generating $X_E \cap X_{\text{ex}}$ require some sort of extra bookkeeping or backtracking schemes that are not necessarily required to generate all of $X_E$. These schemes make implementations of these methods more involved and slower [6, 12, 21, 22].

Motivated, in part, by these difficulties, a handful of researchers in recent years has begun to turn their attention to the mathematics and tools for generating all or portions of the *efficient outcome set* $Y_E^=$, rather than the efficient decision set, for problem (MOLP), where

$$Y_E^{=} = \{Cx \mid x \in X_E\}; \quad (1)$$

see, for instance, [32, 34, 35, 39–44]. There are at least three reasons for this.

First, $Y_E^{=}$ invariably has a much simpler structure and smaller size than $X_E$; see, e.g., [34, 35, 39–44]. This is, in part, because $Y_E^{=} \subseteq \Re^p$ and $X_E \subseteq \Re^n$, where $p$ is typically much smaller than $n$, often by factors of 10, 100 or more. As a result, generating all or portions of $Y_E^{=}$ is expected, in general, to be less demanding computationally than generating all or portions of $X_E$. In addition, the probability of overwhelming the DM by generating all or portions of $Y_E^{=}$ is expected to be less than if $X_E$ or portions of $X_E$ were generated.

Second, it has been shown that, in practice, the DM prefers basing his or her choice of a most preferred solution primarily on $Y_E^{=}$, rather than $X_E$. For instance, arguments to this effect are given in [32, 34, 35].

Third, it is well known that frequently many points in $X_E$ are mapped by $C$ onto either a single point in $Y_E^{=}$ or onto essentially-equivalent outcomes in $Y_E^{=}$; see, for instance, [39, 40, 45]. Thus, generating points directly from $Y_E^{=}$ avoids risking redundant calculations of points from $X_E$ that would be of little or no use to the DM.

Recently some researchers have suggested that to more efficiently generate parts or all of $Y_E^{=}$ (or $X_E$), tools from the *global optimization* literature might be useful; see, e.g., [32, 46–48]. This suggestion is motivated by the fact that these tools are suited, among other things, to exploring complicated nonconvex sets.

In this article, we present and validate a new algorithm, called the Outer Approximation Algorithm, for generating the set of all extreme points of $Y_E^{=}$; that is, the set of all efficient extreme points in the outcome set for problem (MOLP). The algorithm, to our knowledge, is the first algorithm capable of generating this set. It uses a technique called outer approximation. This technique has been used successfully to help solve various single-objective optimization problems, including global optimization problems. The Outer Approximation Algorithm is shown to be finite. As a by-product, the algorithm also generates the entire weakly efficient outcome set of problem (MOLP).

The article is organized as follows. In the next section, theoretical prerequisites for presenting and analyzing the Outer Approximation Algorithm are given. In Section 3, the algorithm is presented and shown to generate the set of all efficient extreme points in the outcome set of problem (MOLP) after a finite number of iterations. The results of some preliminary computational experiments with a prototype VS-FORTRAN code that we have written that implements the algorithm are given and briefly analyzed in Section 4. In Section 5 we conclude that the Outer Approximation Algorithm offers significant promise of allowing decision makers to more easily solve practical applications of problem (MOLP) and to solve larger instances of problem (MOLP) than can presently be solved by decision set-based vector maximization algorithms.

## 2. Theoretical prerequisites

We will assume henceforth that in problem (MOLP), $X$ is a nonempty, compact polyhedron given by

$$X = \{x \in \Re^n \mid Ax = b,\ x \geqslant 0\},$$

where $A$ is an $m \times n$ matrix and $b \in \Re^m$. We will also assume in problem (MOLP) that the rank of $C$ equals $q$, where $q \geqslant 1$. Let $Y^=$ be defined by

$$Y^= = \{Cx \mid x \in X\}.$$

The set $Y^=$ is called the *outcome set* for problem (MOLP); see, for instance, [32, 34, 35, 39, 40]. Notice that $y \in Y^=$ iff $y = Cx$ for some $x \in X$. Other studies [7, 32, 35, 39] have used the set $Y^{\leqslant} = \{y \in \Re^p \mid y \leqslant Cx \text{ for some } x \in X\}$ to good effect as well. We shall not focus, however, on $Y^{\leqslant}$ here. A point $x^0 \in \Re^n$ is called an *efficient* (or *nondominated*) *solution* for problem (MOLP) when $x^0 \in X$ and there exists no point $x \in X$ such that $Cx \geqslant Cx^0$ and $Cx \neq Cx^0$. Similarly, a point $y^0 \in \Re^p$ is called an *efficient* (or *nondominated*) *outcome* for problem (MOLP) when $y^0 \in Y^=$ and there exists no $y \in Y^=$ such that $y \geqslant y^0$ and $y \neq y^0$ [7, 32, 39]. The set of all efficient solutions and the set of all efficient outcomes for problem (MOLP) are called the *efficient decision set* and the *efficient outcome set*, respectively, for problem (MOLP) and are denoted $X_E$ and $Y_E^=$, respectively (cf. Section 1). It is an easy exercise to show that $Y_E^=$ may be equivalently defined by Equation (1) in Section 1; see, for instance, [50].

A point $\overline{x} \in \Re^n$ is called a *weakly efficient* (or *weakly nondominated*) *solution* for problem (MOLP) when $\overline{x} \in X$ and there exists no $x \in X$ such that $Cx > C\overline{x}$. Similarly, a point $\overline{y} \in \Re^p$ is called a *weakly efficient* (or *weakly nondominated*) *outcome* for problem (MOLP) when $\overline{y} \in Y^=$ and there exists no $y \in Y^=$ such that $y > \overline{y}$. The *weakly efficient decision set* $X_{\text{WE}}$ and the *weakly efficient outcome set* $Y_{\text{WE}}^=$ are defined similarly to $X_E$ and $Y_E^=$. It is easy to show that $Y_{\text{WE}}^=$ may also be defined by the equation

$$Y_{\text{WE}}^= = \{Cx \mid x \in X_{\text{WE}}\}.$$

It can be shown that the outcome set $Y^=$ is a nonempty, compact polyhedron in $\Re^p$; see, e.g., [49]. We will have some interest in the dimension of $Y^=$ in this article. In this regard, we will find the following result useful. For any convex set $Z$, let $\dim Z$ denote the dimension of $Z$.

PROPOSITION 2.1. *The dimension of* $Y^=$ *satisfies* $\dim Y^= \leqslant q$.

*Proof.* Let R(C) = $\{Cx | x \in \Re^n\}$ denote the range of $C$, and let N($C^T$) = $\{z \in \Re^p | C^T z = 0\}$ denote the null space of $C^T$. Then, from elementary linear algebra, the sum of dimR(C) and dimN($C^T$) equals $p$. Furthermore, since the rank of $C$ equals $q$, dimN($C^T$) = $p$-$q$. The latter two statements imply the dimR(C) = $q$. Because $Y^=$ is a subset of R(C), this implies that $\dim Y^= \leqslant q$. □

For each $i = 1, 2, \dots, p$, let

$$y_i^{AI} = \min y_i, \text{ s.t. } y \in Y^{=}.$$

The vector $y^{AI} \in \Re^p$ is called the *anti-ideal outcome* for problem (MOLP). Notice that since $X$ is nonempty and compact, the components of $y^{AI}$ are all finite.

Let $\hat{y} \in \Re^p$ satisfy $\hat{y} < y^{AI}$, and consider the set $Y$ given by

$$Y = \{y \in \Re^p \mid \hat{y} \leqslant y \leqslant Cx \text{ for some } x \in X\}.$$

The set $Y$ is instrumental in the algorithm to be presented in Section 3 for reasons that will become clear shortly.

PROPOSITION 2.2. *The set $Y$ is a nonempty, bounded polyhedron in $\Re^p$ of dimension $p$.*

*Proof.* Since $\hat{y} < y^{AI} \leqslant Cx$ for all $x \in X$, where $X$ is nonempty and bounded, the definition of $Y$ implies that $Y$ is a nonempty, bounded set in $\Re^p$. Notice that $Y$ may be written

$$Y = P_1 \cap (Y^{=} + P_2), \tag{2}$$

where

$$P_1 = \{y \in \Re^p \mid y \geqslant \hat{y}\}$$

and

$$P_2 = \{z \in \Re^p \mid z \leqslant 0\}.$$

By definition, $P_1$ and $P_2$, are polyhedral sets, and, as noted previously, the outcome set $Y^{=}$ is a polyhedron. From (2), Corollary 19.3.2 in [49], and the definition of a polyhedron, this implies that $Y$ is a polyhedral set. Since $\hat{y} < Cx$ for all $x \in X$, the interior of $Y$ is nonempty. Therefore, $\dim Y = p$, and the proof is complete. □

A point $y^0 \in Y$ is called an *efficient* (or *admissible*) *point* of $Y$ when no $y \in Y$ exists such that $y \geqslant y^0$ and $y \neq y^0$. When $y^0 \in Y$ and no $y \in Y$ exists such that $y > y^0$, then $y^0$ is called a *weakly efficient* (or *weakly admissible*) *point* of $Y$. Let $Y_E$ and $Y_{\text{WE}}$ denote the set of all efficient and weakly efficient points, respectively, of $Y$.

THEOREM 2.1. $Y_E^{=} = Y_E$.

*Proof.* Suppose that $y \in Y_E^{=}$. Then from (1), $y = Cx$ for some $x \in X_E$. By definition of $\hat{y}$, this implies that $\hat{y} < y \leqslant Cx$, so that $y \in Y$.

Assume that $y \notin Y_E$. Then we may choose a point $y^1 \in Y$ such that $y^1 \geqslant y$ and $y^1 \neq y$. Since $y^1 \in Y$, there exists a point $x^1 \in X$ such that $y^1 \leqslant Cx^1$. The latter two statements imply that $Cx^1 \geqslant y$ and $Cx^1 \neq y$. Substituting for $y$, we obtain that $Cx^1 \geqslant Cx$ and $Cx^1 \neq Cx$. Since $x^1 \in X$, this contradicts that $x \in X_E$. Therefore, the assumption that $y \notin Y_E$ must be false, so that $Y_E^{=} \subseteq Y_E$.

Suppose that $y \in Y_E$. To show that $y \in Y_E^{=}$, we will show that $y = Cx$ for some $x \in X_E$. Towards this end, notice that $y \in Y$, so that $y \leqslant Cx$ for some $x \in X$.

Choose such an $x$, and assume that, in particular, $y \leqslant Cx$ and $y \neq Cx$. Then, since $y \in Y$ and $x \in X$, $y \notin Y_E$. But by assumption, $y \in Y_E$. This contradiction implies that whenever $y \leqslant Cx$ for some $x \in X$, $y = Cx$ must hold.

Let $x^0 \in X$ satisfy $y \leqslant Cx^0$. Then from the previous paragraph, $y = Cx^0$. If $x^0 \notin X_E$ were true, then for some $x^1 \in X$, $Cx^1 \geqslant Cx^0 = y$ with $Cx^1 \neq y$ would hold, which, from the previous paragraph, is impossible. Therefore, $x^0 \in X_E$. Since $y = Cx^0$, this implies by (1) that $y \in Y_E^=$. Therefore, $Y_E \subseteq Y_E^=$ and the theorem is established. □

Notice from Proposition 2.2. and Theorem 2.1. that $Y$ is a nonempty, full-dimensional compact polyhedron in $\Re^p$ whose efficient point set is precisely equal to the set of all efficient points of the outcome set $Y^=$ for problem (MOLP). We will therefore refer to $Y$ as an *efficiency-equivalent polyhedron* for $Y^=$. The outer approximation algorithm to be presented will generate the entire efficiency-equivalent polyhedron $Y$ for $Y^=$. This will allow the user to immediately identify the set of all efficient extreme points of the outcome set $Y^=$ for problem (MOLP).

REMARK 2.1. A slightly-different form for an efficiency-equivalent polyhedron from the one that we are using here can be found in [42, 43].

REMARK 2.2. Notice from Propositions 2.1. and 2.2. and from Theorem 2.1. that even though $Y_E^= = Y_E$, the dimension of $Y^=$ may be strictly less than $\dim Y = p$.

Let

$$\beta = \max\langle e, y\rangle, \quad \text{s.t. } y \in Y,$$

where $e \in \Re^p$ is the vector in which each entry is equal to 1.0. By Proposition 2.2., $\beta$ is a finite number. Let $v^0 = \hat{y}$ and, for each $j = 1, 2, \ldots, p$, define $v^j \in \Re^p$ by

$$v_i^j = \begin{cases} \hat{y}_i & \text{if } i \neq j, \\ \beta + \hat{y}_j - \langle e, \hat{y}\rangle & \text{if } i = j, \end{cases}$$

$i = 1, 2, \ldots, p$. In the outer approximation algorithm for problem (MOLP), an initial simplex containing $Y$ is constructed. This construction is based upon the following result.

THEOREM 2.2. *The convex hull $S$ of $V(S) := \{v^j \mid j = 0, 1, \ldots, p\}$ is a $p$-dimensional simplex with vertex set $V(S)$, and $S$ contains $Y$.*

*Proof.* Since $\hat{y} < y^{AI} \leqslant y$ for all $y \in Y$, $\beta - \langle e, \hat{y}\rangle > 0$. For each $j = 1, 2, \ldots, p$,

$$\langle v^j - v^0\rangle_i = \begin{cases} 0 & \text{if } i \neq j, \\ \beta - \langle e, \hat{y}\rangle & \text{if } i = j, \end{cases} \tag{3}$$

$i = 1, 2, \ldots p$. The latter two statements imply that $\{(v^j - v^0) \mid j = 1, 2, \ldots, p\}$ is a linearly independent set. Hence, $\{v^0, v^1, \ldots, v^p\}$ is an affinely independent set, so that, by definition, $S$ is a $p$-dimensional simplex with vertex set $V(S)$.

To show that $S$ contains $Y$, suppose first that $\overline{y} \in Y$. Then $\hat{y} \leqslant \overline{y}$, so that $(\overline{y} - \hat{y}) = (\overline{y} - v^0) \geqslant 0$. Furthermore,

$$\begin{aligned} \langle e, \overline{y} - v^0\rangle &\leqslant \max_{y \in Y}\langle e, y - v^0\rangle \\ &= \max_{y \in Y}\langle e, y\rangle - \langle e, v^0\rangle \\ &= \beta - \langle e, v^0\rangle \\ &= \beta - \langle e, \hat{y}\rangle, \end{aligned} \tag{4}$$

where the latter two equations follow from the definitions of $\beta$ and of $v^0$, respectively. Since $(\overline{y} - v^0) \geqslant 0$, (4) implies that for each $j = 1, 2, \ldots, p$,

$$0 \leqslant (\overline{y} - v^0)_j \leqslant \beta - \langle e, \hat{y}\rangle.$$

Therefore, we may choose scalars $\alpha_j$, $j = 1, 2, \ldots, p$, such that for each $j = 1, 2, \ldots, p$, $0 \leqslant \alpha_j \leqslant 1$ and

$$(\overline{y} - v^0)_j = \alpha_j(\beta - \langle e, \hat{y}\rangle).$$

From (3), this implies that

$$(\overline{y} - v^0) = \sum_{j=1}^{p} \alpha_j(v^j - v^0). \tag{5}$$

Notice that since $\alpha_j \geqslant 0$, $j = 1, 2, \ldots, p$,

$$\sum_{j=1}^{p} \alpha_j \geqslant 0.$$

If

$$\sum_{j=1}^{p} \alpha_j > 1$$

were true, then, using (3), (5), and this assumption, it would follow that

$$\begin{aligned} \langle e, \overline{y} - v^0\rangle &= \sum_{j=1}^{p} \alpha_j\langle e, v^j - v^0\rangle \\ &= \sum_{j=1}^{p} \alpha_j(\beta - \langle e, \hat{y}\rangle) \\ &> \beta - \langle e, \hat{y}\rangle, \end{aligned}$$

which contradicts (4). Therefore, $0 \leqslant \sum_{j=1}^{p} \alpha_j \leqslant 1$ must hold. Furthermore, from (5),

$$\overline{y} = \left(1 - \sum_{j=1}^{p} \alpha_j\right) v^0 + \sum_{j=1}^{p} \alpha_j \, v^j .$$

Since, for each $j = 1, 2, \ldots, p$, $0 \leqslant \alpha_j \leqslant 1$, and $0 \leqslant \left(1 - \sum_{j=1}^{p} \alpha_j\right) \leqslant 1$, this means that $\overline{y}$ is a convex combination of $\{v^j \mid j = 0, 1, \ldots, p\}$. Therefore, $\overline{y} \in S$, and we have shown that $Y \subseteq S$. □

The outer approximation algorithm for problem (MOLP) will also make use of the alternate representation of the simplex $S$ given in the following theorem.

THEOREM 2.3. *The simplex $S$ defined in Theorem 2.2. may also be written*

$$S = \{y \in \Re^p \mid \hat{y} \leqslant y, \ \langle e, y \rangle \leqslant \beta\}.$$

*Proof.* Suppose that $\overline{y} \in \Re^p$ is contained in the convex hull of $\{v^j \mid j = 0, 1, \ldots, p\}$. Then we may choose $p + 1$ scalars $f_j$, $j = 0, 1, 2, \ldots, p$, that sum to 1.0 and satisfy $0 \leqslant f_j \leqslant 1$ for each $j = 0, 1, \ldots, p$, in such a way that

$$\overline{y} = \sum_{j=0}^{p} f_j \, v^j .$$

As a result,

$$\begin{aligned}
\overline{y} &= \left(1 - \sum_{j=1}^{p} f_j\right) v^0 + \sum_{j=1}^{p} f_j \, v^j \\
&= v^0 + \sum_{j=1}^{p} f_j \left(v^j - v^0\right) \\
&= \hat{y} + \left(\beta - \langle e, \hat{y} \rangle\right) f, \qquad (6)
\end{aligned}$$

where $f \in \Re^p$ has components $f_j$, $j = 1, 2, \ldots, p$, the first equality holds because $f_0, f_1, \ldots, f_p$ sum to 1.0 and the third equality holds by the definitions of $v^0, v^1, \ldots, v^p$. From the proof of Theorem 2.2., $\beta - \langle e, \hat{y} \rangle > 0$. This, together with (6) and the fact that $f \geqslant 0$, implies that $\hat{y} \leqslant \overline{y}$. Furthermore,

$$\begin{aligned}\langle e, \overline{y}\rangle &= \langle e, \hat{y}\rangle + \sum_{j=1}^{p} f_j \left(\beta - \langle e, \hat{y}\rangle\right) \\ &= \left(1 - \sum_{j=1}^{p} f_j\right) \langle e, \hat{y}\rangle + \beta \sum_{j=1}^{p} f_j \\ &= f_0\langle e, \hat{y}\rangle + \beta \sum_{j=1}^{p} f_j,\end{aligned}$$

where the first equality follows from (6) and the third equality holds because the sum of $f_0, f_1, \ldots, f_p$ is 1.0. Since $\langle e, \hat{y}\rangle < \beta$ holds from the proof of Theorem 2.2., this implies that

$$\langle e, \overline{y}\rangle \leqslant f_0\beta + \beta \sum_{j=1}^{p} f_j = \beta.$$

Therefore, $\overline{y} \in \{y \in \Re^p \mid \hat{y} \leqslant y, \langle e, y\rangle \leqslant \beta\}$.

Now suppose that $\overline{y} \in \{y \in \Re^p \mid \hat{y} \leqslant y, \langle e, y\rangle \leqslant \beta\}$. For each $j = 1, 2, \ldots, p$, let

$$\alpha_j = \frac{(\overline{y} - v^0)_j}{(\beta - \langle e, \hat{y}\rangle)},$$

where, as shown previously, $(\beta - \langle e, \hat{y}\rangle) > 0$. Since $\hat{y} = v^0$ and $\langle e, \overline{y}\rangle \leqslant \beta$,

$$\begin{aligned}\beta - \langle e, \hat{y}\rangle &= \beta - \langle e, v^0\rangle \\ &\geqslant \langle e, \overline{y}\rangle - \langle e, v^0\rangle \\ &= \langle e, \overline{y} - v^0\rangle,\end{aligned} \tag{7}$$

so that for each $j = 1, 2, \ldots, p$, $(\overline{y} - v^0)_j \leqslant \beta - \langle e, \hat{y}\rangle$. Together with the facts that $\overline{y} \geqslant \hat{y} = v^0$ and $(\beta - \langle e, \hat{y}\rangle)$ is positive, this implies that for each $j = 1, 2, \ldots, p$, $0 \leqslant \alpha_j \leqslant 1$. Furthermore, by the definition of $\alpha_j$, $j = 1, 2, \ldots, p$,

$$\begin{aligned}\sum_{j=1}^{p} \alpha_j &= \frac{\sum_{j=1}^{p} (\overline{y} - v^0)_j}{\beta - \langle e, \hat{y}\rangle} \\ &= \frac{\langle e, \overline{y}\rangle - \langle e, v^0\rangle}{\beta - \langle e, \hat{y}\rangle} \\ &\leqslant 1,\end{aligned} \tag{8}$$

where the inequality follows from (7) and the fact that $(\beta - \langle e, \hat{y}\rangle)$ is positive.

By the definitions of $v^j$, $j = 0, 1, \dots, p$, and $\alpha_j$, $j = 1, 2, \dots, p$,

$$\begin{aligned} v^0 + \sum_{j=1}^{p} \alpha_j (v^j - v^0) &= v^0 + \sum_{j=1}^{p} \frac{(\overline{y} - v^0)_j}{\beta - \langle e, \hat{y} \rangle} (v^j - v^0) \\ &= v^0 + (\overline{y} - v^0) \\ &= \overline{y}. \end{aligned} \tag{9}$$

Rearranging the left-hand side of (9), we obtain

$$\left(1 - \sum_{j=1}^{p} \alpha_j\right) v^0 + \sum_{j=1}^{p} \alpha_j v^j = \overline{y}.$$

From (8), since $0 \leqslant \alpha_j \leqslant 1$, this implies that $\overline{y}$ belongs to the convex hull of $\{v^j \mid j = 0, 1, \dots, p\}$. □

From Proposition 2.2., we may choose a point $\overline{p} \in \operatorname{int} Y$, where $\operatorname{int} Y$ denotes the interior of $Y$. Starting with the simplex $S$ defined in Theorem 2.2., the outer approximation algorithm will iteratively generate a finite number of nonempty, compact, polyhedra $S^k$, $k = 0, 1, 2, \dots, K$, such that $S = S^0 \supset S^1 \supset \cdots \supset S^{K-1} \supset S^K = Y$. In a typical iteration $k$, a vertex $y^k$ of $S^k$ will be identified such that $y^k \notin Y$. Subsequently, the unique point $w^k$ on the boundary of $Y$ that lies on the line segment connecting $\overline{p}$ and $y^k$ will be identified. The next result implies that $w^k$ belongs to $Y_{\mathrm{WE}}$. As we shall see, this fact will play an important role in establishing the validity of the algorithm.

THEOREM 2.4. *Let $\overline{p} \in \operatorname{int} Y$ and suppose that $y \geqslant \hat{y}$ and $y \notin Y$. Let $w$ denote the unique point on the boundary of $Y$ that belongs to the line segment connecting $y$ and $\overline{p}$. Then $w \in Y_{WE}$.*

*Proof.* Suppose, to the contrary, that $w \notin Y_{\mathrm{WE}}$. Then we may choose a point $y^0 \in Y$ such that $y^0 > w$. Since $y^0 \in Y$, we may also choose a point $x^0 \in X$ such that $Cx^0 \geqslant y^0$. Therefore, $Cx^0 > w$.

By assumption, $y \notin Y$ and $\overline{p} \in \operatorname{int} Y$. Since $Y$ is closed, and since $w$ belongs to the boundary of $Y$ and to the line segment connecting $y$ and $\overline{p}$, this implies that $w = \lambda \overline{p} + (1 - \lambda) y$ for some $\lambda$ that satisfies $0 < \lambda < 1$. By assumption, $y \geqslant \hat{y}$ and, since $\overline{p} \in \operatorname{int} Y$, $\overline{p} > \hat{y}$. From the previous two observations, it follows that $w > \hat{y}$.

Choose $\epsilon > 0$ such that $\epsilon < d_1$ and $\epsilon < d_2$, where

$$\begin{aligned} d_1 &= \min\{(Cx^0 - w)_j \mid j = 1, 2, \dots, p\}, \\ d_2 &= \min\{(w - \hat{y})_j \mid j = 1, 2, \dots, p\}. \end{aligned}$$

For any $v \in \Re^p$, let $\|v\|$ denote the Euclidean norm of $v$. Suppose that $z \in N_\epsilon(w)$, where $N_\epsilon(w) = \{q \in \Re^p \mid \|q - w\| < \epsilon\}$ is the open ball of radius $\epsilon$ centered at $w$.

Then, for each $j = 1, 2, \ldots, p$, $-\epsilon < (z_j - w_j) < \epsilon$, which, upon rearrangement, may be written $w_j - \epsilon < z_j < w_j + \epsilon$. Since $\epsilon < d_1$, $\epsilon < (Cx^0)_j - w_j$ for each $j = 1, 2, \ldots, p$, and, since $\epsilon < d_2$, $-\epsilon > \hat{y}_j - w_j$ for each $j = 1, 2, \ldots, p$. Combining the latter two statements, we conclude that for each $j = 1, 2, \ldots, p$, $\hat{y}_j < z_j < (Cx^0)_j$. Since $x^0 \in X$, this implies that $z \in Y$. It follows that the open ball $N_\epsilon(w)$ is a subset of $Y$. Since $\epsilon > 0$, this contradicts the fact that $w$ belongs to the boundary of $Y$, so that the proof is complete. □

From Proposition 2.2. and [49], $Y$ is a $p$-dimensional polyhedron with a finite number of faces, and a set $F \subseteq \Re^p$ is a face of $Y$ if and only if $F$ equals the optimal solution set $Y^*(\alpha)$ to the problem

$$(P_\alpha) \quad \max\langle \alpha, y\rangle, \quad \text{s.t. } y \in Y$$

for some $\alpha \in \Re^p$. Since $\hat{y} < Cx$ for all $x \in X$, by the definition of $Y$, this implies that $p$ of the $(p-1)$-dimensional faces of $Y$ are given by the sets

$$F_j = \{y \in Y \mid y_j = \hat{y}_j\},$$

$j = 1, 2, \ldots, p$. It is well known that for each $j = 1, 2, \ldots, p$, either $F_j \subseteq Y_{\text{WE}}$ or ri $F_j \cap Y_{\text{WE}} = \emptyset$, where ri $F_j$ denotes the relative interior of $F_j$; see, for instance, [7]. For each $j = 1, 2, \ldots, p$, since $\hat{y} \in F_j$ and $\hat{y} \notin Y_{\text{WE}}$, this implies that ri $F_j \cap Y_{\text{WE}} = \emptyset$. As a result, the point $w$ in Theorem 2.4. lies in some face $F \subseteq Y_{\text{WE}}$ of $Y$ that satisfies $F \neq F_j$ for each $j = 1, 2, \ldots, p$. From [7], any such face $F$ is precisely equal to the optimal solution set $Y^*(\alpha)$ of problem $(P_\alpha)$ for some $\alpha \in \Re^p$ such that $\alpha \geqslant 0$, $\alpha \neq 0$. The following result provides a means for finding such a face $F$ and representing it in this way.

THEOREM 2.5. *Assume that $w \in Y_{WE}$, and let $(u^{*^T}, v^{*^T})$ denote any optimal solution for the dual linear program to the problem*

$$\begin{aligned} (Q_w) \quad \max \quad & t, \\ \textit{s.t.} \quad & Cz - et \geqslant w, && (10) \\ & Az = b, && (11) \\ & z, t \geqslant 0, \end{aligned}$$

*where $u^* \in \Re^p$ and $v^* \in \Re^m$ correspond to constraints (10) and (11), respectively, of problem $(Q_w)$. Then $u^* \geqslant 0$, $u^* \neq 0$, and $w$ belongs to the weakly efficient face $Y^*(u^*)$ of $Y$. Furthermore, $Y^*(u^*)$ is given by*

$$Y^*(u^*) = \{y \in Y \mid \langle u^*, y\rangle = \langle b, v^*\rangle\}.$$

*Proof.* Let $Y^{\leqslant} = \{y \in \Re^p \mid y \leqslant Cx \text{ for some } x \in X\}$. For each $y \in Y^{\leqslant}$, let $g(y)$ denote the optimal value of problem $(Q_w)$ with $w = y$. Then, since $w \in Y$, $g(w) \geqslant 0$. In fact, since $w \in Y_{\text{WE}}$, it is easy to see that $g(w) = 0$. Therefore, by duality theory of linear programming, the dual linear program to problem $(Q_w)$, which is given by

$$
\begin{aligned}
(QD_w) \quad \min \quad & -\langle w, u\rangle + \langle b, v\rangle, \\
\text{s.t.} \quad & -u^T C + v^T A \geqslant 0, \\
& \langle e, u\rangle \geqslant 1, \\
& u \geqslant 0,
\end{aligned}
$$

also has an optimal value of $g(w) = 0$.

Let $(u^{*^T}, v^{*^T})$ denote any optimal solution to problem $(QD_w)$. Then, from the constraints of problem $(QD_w)$, it follows that $u^* \geqslant 0$, $u^* \neq 0$. Furthermore, since the optimal value of problem $(QD_w)$ equals 0,

$$
\langle w, u^*\rangle = \langle b, v^*\rangle. \tag{12}
$$

Since $u^* \geqslant 0$ and $u^* \neq 0$, from [7] we know that the optimal solution set $Y^*(u^*)$ for problem $(P_{u^*})$ corresponds to a weakly efficient face of $Y$. From this and (12), it follows that if we show that $w$ is an optimal solution to problem $(P_{u^*})$, the theorem will be established.

To show that $w$ is an optimal solution to problem $(P_{u^*})$, notice first that by the definition of $Y$, this problem can also be written

$$
\begin{aligned}
(P_{u^*}) \quad \max \quad & \langle u^*, y\rangle \\
\text{s.t.} \quad & -y \leqslant -\hat{y}, \\
& y - Cz \leqslant 0, \\
& Az = b, \\
& z \geqslant 0.
\end{aligned}
$$

Until we indicate otherwise in this proof, we will use the latter representation of problem $(P_{u^*})$. The dual linear program to problem $(P_{u^*})$ is given by

$$
\begin{aligned}
(D_{u^*}) \quad \min \quad & -\langle \hat{y}, s\rangle + \langle b, q\rangle, \\
\text{s.t.} \quad & -s^T + r^T = u^{*T}, \\
& -r^T C + q^T A \geqslant 0, \\
& s, r \geqslant 0.
\end{aligned}
$$

Notice that since $(u^{*^T}, v^{*^T})$ is an optimal solution to problem $(QD_w)$, the vector $(s^T, r^T, q^T) = (0^T, u^{*T}, v^{*T})$ is feasible in problem $(D_{u^*})$ and has objective function value $\langle b, v^*\rangle$ there.

Let $(z^{*^T}, t^*)$ be an optimal solution for problem $(Q_w)$. Since $g(w) = 0$, this implies that $t^* = 0$, $Cz^* \geqslant w$, $Az^* = b$, and $z^* \geqslant 0$. Together with the fact that $w \geqslant \hat{y}$, this implies that $(y^T, z^T) = (w^T, z^{*^T})$ is a feasible solution for problem $(P_{u^*})$ with an objective function value equal to $\langle u^*, w\rangle$. From (12), $\langle u^*, w\rangle =$

$\langle b, v^* \rangle$. Since $(0^T, u^{*^T}, v^{*^T})$ is a feasible solution for problem $(D_{u^*})$ with objective function value $\langle b, v^* \rangle$, this implies by duality theory of linear programming that $(y^T, z^T) = (w^T, z^{*T})$ is an optimal solution to problem $(P_{u^*})$. Therefore, $w$ is an optimal solution to the representation of problem $(P_{u^*})$ given by

$$(P_{u^*}) \quad \max \langle u^*, y \rangle, \quad \text{s.t. } y \in Y. \qquad \square$$

Theorem 2.5. will provide the basis for constructing certain linear inequality cuts needed in the Outer Approximation Algorithm for problem (MOLP) to be presented in the next section.

## 3. The outer approximation algorithm

The Outer Approximation Algorithm presented below uses results from Section 2 and the idea of outer approximation to generate the entire efficiency-equivalent polyhedron $Y$ for the outcome set $Y^=$ of problem (MOLP). As we shall soon see, this will allow the set of all efficient extreme points in $Y^=$ to be immediately identified.

*Outer approximation algorithm*

*Initialization step*. Compute a point $\overline{p} \in \operatorname{int} Y$ and construct the $p$-dimensional simplex $S^0 = S$ containing $Y$ described in Theorems 2.2. and 2.3.. Store both the vertex set $V(S^0)$ of $S^0 = S$ given in Theorem 2.2. and the inequality representation of $S^0 = S$ given in Theorem 2.3.. Set $k = 0$ and go to iteration $k$.

*Iteration* $k,\ k \geqslant 0$. See Steps $k1$ through $k4$ below.

*Step* $k1$. If, for each $y \in V(S^k)$, $y \in Y$ is satisfied, then stop: $Y = S^k$. Otherwise, choose any $y^k \in V(S^k)$ such that $y^k \notin Y$ and continue.

*Step* $k2$. Find the unique value $\lambda_k$ of $\lambda,\ 0 < \lambda < 1$, such that $\lambda y^k + (1 - \lambda)\overline{p}$ belongs to the boundary of $Y$, and set $w^k = \lambda_k y^k + (1 - \lambda_k)\overline{p}$.

*Step* $k3$. Set $S^{k+1} = S^k \cap \{y \in \Re^p \mid \langle u^k, y \rangle \leqslant \langle b, v^k \rangle\}$, where $(u^{k^T}, v^{k^T})$ is any dual optimal solution to the linear program $(Q_w)$ with $w = w^k$ (cf. Theorem 2.5.).

*Step* $k4$. Using $V(S^k)$ and the definition of $S^{k+1}$ given in Step $k3$, determine $V(S^{k+1})$. Set $k = k + 1$ and go to iteration $k$.

For each $k \geqslant 0$, the inequality

$$\langle u^k, y \rangle \leqslant \langle b, v^k \rangle$$

appended to $S^k$ in Step $k3$ is called an *inequality cut* [47, 48]. As we shall see, each such inequality is constructed so that $S^{k+1}$ "cuts off" a portion of $S^k$ containing $y^k$ in such a way that $S^k \supset S^{k+1} \supset Y$.

Notice that except for Steps $k2$ and $k4$, the Outer Approximation Algorithm can be implemented by using linear programming techniques. For instance, to compute $\overline{p} \in \operatorname{int} Y$ in the Initialization Step, $\overline{p}$ may be set equal to any strict convex combination of $y^{AI}$ and $Cz^*$, where $(z^{*^T}, t^*)$ is any optimal solution to the linear program $(Q_w)$ given in Section 2 with $w = y^{AI}$. In Step $k1$, to test a given point $y \in V(S^k)$ for membership in $Y$, one may apply the Phase 1 procedure of the simplex method to problem $(Q_w)$ with $w = y$. It is easy to show that the other steps, apart from Steps $k2$ and $k4$, can be implemented by using linear programming as well.

Step $k2$ can be executed by applying linear programming and standard univariate search techniques. In particular, $\lambda_k$ in Step $k2$ can be found by using univariate search techniques to determine the largest value of $\lambda$, $0 < \lambda < 1$, such that the linear program $(Q_w)$ is feasible with $w = \lambda y^k + (1-\lambda)\overline{p}$. The feasibility of this linear program can be determined, for instance, via the Phase 1 procedure of the simplex method.

The execution of Step $k4$ can be accomplished via one of several special techniques from the global optimization literature; see, for instance, [51–56] and a review by Horst [57]. We expect from evidence in the global optimization literature that as the dimension $p$ of the polyhedra $S^k$, $k = 0, 1, 2, \ldots$, increases, executing Step $k4$ will, in general, become a computationally-demanding task. However, for cases where $p \leqslant 20$, any of the methods given in [51–56] can be expected to be relatively efficient; see, for instance, [46, 58]. Fortunately, in practice, the number of objective functions $p$ in problem (MOLP) is almost invariably less than 20; see, e.g. [3, 6, 59].

Notice also that, in contrast to many decision set-based approaches for problem (MOLP), the Outer Approximation Algorithm avoids the need for complicated bookkeeping or backtracking [6, 12, 29].

THEOREM 3.1. *The Outer Approximation Algorithm is finite and, when it terminates, $S^K = Y$, where $K \geqslant 0$ is the final iteration number.*

*Proof.* From Theorem 2.2., the initial simplex $S = S^0$ of the algorithm contains $Y$. Suppose that $k \geqslant 0$ and that $S^k \supseteq Y$ but $S^k \neq Y$. Then in Step $k1$ of the algorithm, an element $y^k$ of $V(S^k)$ will be found such that $y^k \notin Y$. By Theorem 2.3. and Step $k3$ of the algorithm, for any point $y \in S^k$, $y \geqslant \hat{y}$ must hold. Since $y^k \in S^k$, this implies that $y^k \geqslant \hat{y}$. By Theorem 2.4. and Step $k2$, since $\overline{p} \in \operatorname{int} Y$ and $y^k \notin Y$, $w^k \in Y_{\mathrm{WE}}$. From Theorem 2.5. and Step $k3$, $\{y \in Y \mid \langle u^k, y\rangle = \langle b, v^k\rangle\}$ is a weakly efficient face of $Y$ containing $w^k$. Since $\overline{p} \in \operatorname{int} Y$, this implies that $\langle u^k, \overline{p}\rangle \neq \langle b, v^k\rangle$. In addition, from Theorem 2.5, $\langle u^k, y\rangle \leqslant \langle b, v^k\rangle$ for all $y \in Y$. Therefore $\langle u^k, \overline{p}\rangle < \langle b, v^k\rangle$. Since $\langle u^k, w^k\rangle = \langle b, v^k\rangle$, by Step $k2$ of the algorithm, this implies that $\langle u^k, y^k\rangle > \langle b, v^k\rangle$. Therefore, by the definition of $S^{k+1}$ in Step $k3$, $y^k \notin S^{k+1}$. On the other hand, since $S^k \supseteq Y$ and $\langle u^k, y\rangle \leqslant \langle b, v^k\rangle$ for all $y \in Y$, $S^{k+1} \supseteq Y$. By the choice of $k$, we conclude that the algorithm generates *distinct* polyhedra $S^k$, $k = 0, 1, 2, \ldots$, such that

$$S^0 \supset S^1 \supset \cdots \supset S^k \supset Y.$$

Furthermore, for each $k \geqslant 0$, from Step $k3$,

$$S^{k+1} = S^k \cap \{y \in \Re^p \mid \langle u^k, y\rangle \leqslant \langle b, v^k\rangle\},$$

where $\{y \in Y \mid \langle u^k, y\rangle = \langle b, v^k\rangle\}$ is a face of $Y$. By Proposition 2.2., this implies that the algorithm must be finite and it must terminate in some iteration $K \geqslant 0$ with $S^K = Y$. □

When the algorithm terminates, the set of all efficient extreme points in the outcome set $Y^=$ for problem (MOLP) can be easily found by using the next result.

THEOREM 3.2. *Let $K \geqslant 0$ denote the iteration number in which $S^K = Y$ and the Outer Approximation Algorithm terminates. Let*

$$E = \{y \in V(S^K) \mid y > \hat{y}\}.$$

*Then $E$ is identical to the set of all efficient extreme points of $Y^=$.*

*Proof.* From Theorems 2.3.–2.5. and 3.1., and from Step $k3$ of the Outer Approximation Algorithm,

$$\begin{aligned} S^K &= Y \\ &= \{y \in \Re^p \mid \hat{y} \leqslant y,\ \langle e, y\rangle \leqslant \beta\} \cap \left(\bigcap_{i=0}^{K-1} H_i\right), \end{aligned}$$

where, for each $i = 0, 1, 2, \ldots, K-1$,

$$H_i = \{y \in \Re^p \mid \langle u^i, y\rangle \leqslant \langle b, v^i\rangle\},$$

and $\{y \in Y \mid \langle u^i, y\rangle = \langle b, v^i\rangle\} \subseteq Y_{\mathrm{WE}}$. Notice also that since $e > 0$, the definition of $\beta$ implies that $\{y \in Y \mid \langle e, y\rangle = \beta\} \subseteq Y_{\mathrm{WE}}$.

Suppose that $v \in E$. Then $v \in V(S^K) = V(Y)$ and $v > \hat{y}$. Therefore, at least $p$ of the inequalities

$$\begin{aligned} &\langle e, y\rangle \leqslant \beta \\ &\langle u^i, y\rangle \leqslant \langle b, v^i\rangle, \quad i = 0, 1, \ldots, K-1, \end{aligned}$$

must hold as equations at $y = v$. This implies that $v \in Y_{\mathrm{WE}}$.

To show that $v \in Y_E$, we will use a proof by contradiction. Towards this end, suppose that $v \notin Y_E$. Then we may choose $y \in Y$ such that $y \geqslant v$ and $y \neq v$. Since $v \in Y_{\mathrm{WE}}$, $y \not> v$. Let

$$I_1 = \{i \in \{1, 2, \ldots, p\} \mid y_i = v_i\}$$

and

$$I_2 = \{i \in \{1, 2, \ldots, p\} \mid y_i > v_i\}.$$

Then $I_1, I_2 \neq \emptyset$ and $I_1 \cup I_2 = \{1, 2, \ldots, p\}$. For each $i \in I_2$, let $n_i = y_i - v_i > 0$. Choose $M > 0$ sufficiently large so that for each $i \in I_2$, $v_i - n_i/M > \hat{y}_i$, and define $\overline{v} \in \Re^p$ by

$$\overline{v}_i = \begin{cases} v_i, & i \in I_1 \\ v_i - \dfrac{n_i}{M}, & i \in I_2. \end{cases}$$

Then, by the choice of $M$, since $v \in Y$, $\overline{v} \in Y$. Also, since $I_2 \neq \emptyset$, $\overline{v} \neq v$.

Notice that

$$y_i = \begin{cases} v_i, & i \in I_1 \\ v_i + n_i, & i \in I_2, \end{cases}$$

so that

$$v = \frac{1}{M+1}\, y + \frac{M}{M+1}\, \overline{v}.$$

Since $0 < [1/(M+1)] < 1$, this implies that $v$ is a strict convex combination of $y$ and $\overline{v}$. On the other hand, $v$ is an extreme point of $Y$, and $y, \overline{v} \in Y$, where $y \neq v$ and $\overline{v} \neq v$, which is a contradiction. Therefore, $v \in Y_E$ must hold.

From Theorem 2.1., since $v \in Y_E$, $v \in Y_E^=$. Therefore $v \in Y^=$. Since $v \in V(S^K) = V(Y)$ and $Y^= \subseteq Y$, this implies that $v$ is an extreme point of $Y^=$. Thus, $v$ is an efficient extreme point of $Y^=$. By the choice of $v$, we have shown that $E$ is a subset of the set of efficient extreme points of $Y^=$.

Now suppose that $v$ is an efficient extreme point of $Y^=$. Then, since $v \in Y^=$, $v \geqslant y^{AI}$. Since $y^{AI} > \hat{y}$, this implies that $v > \hat{y}$. Thus, to complete the proof, we need only to show that $v \in V(S^K)$.

Towards this end, assume, to the contrary, that $v \notin V(S^K)$. Then, since $S^K = Y$, $v \notin V(Y)$. Therefore, we may choose $z^1, z^2 \in Y$ and $\theta \in \Re$ such that $z^1 \neq v$, $z^2 \neq v$, $0 < \theta < 1$, and

$$v = \theta z^1 + (1-\theta) z^2. \tag{13}$$

From Theorem 3.1. in [60], since $v$ is an efficient extreme point of the polyhedron $Y^=$, we may select a point $u \in \Re^p$, $u > 0$, such that $v$ is the unique optimal solution to the problem

$$\max\langle u, y\rangle, \quad \text{s.t. } y \in Y^=.$$

From the definition of $Y$, this implies that $v$ is also the unique optimal solution to the problem $(P_\alpha)$, defined in Section 2, with $\alpha = u$. Therefore, since $z^1, z^2 \in Y$,

$$\langle u, z^1\rangle < \langle u, v\rangle$$

and

$$\langle u, z^2\rangle < \langle u, v\rangle$$

must hold. Since $0 < \theta < 1$, these inequalities imply that

$$\theta\langle u, z^1\rangle + (1-\theta)\langle u, z^2\rangle < \langle u, v\rangle.$$

From (13), the left-hand-side of the previous inequality equals $\langle u, v\rangle$, yielding a contradiction. Therefore, $v \in V(S^K)$ must be true. □

Notice that in the Outer Approximation Algorithm, each of the sets $V(S^k)$, $k = 0, 1, 2, \ldots, K$, is explicitly computed. From Theorem 3.2., this implies that the set $E$ of all efficient extreme points in the outcome set $Y^=$ is essentially immediately available upon termination of the algorithm.

REMARK 3.1. It can also be shown that when the Outer Approximation Algorithm terminates, the weakly efficient outcome set $Y^=_{\mathrm{WE}}$ is given by

$$Y^=_{\mathrm{WE}} = \bigcup_{i=0}^{K-1} \{y \in \Re^p \mid y = Cx \text{ for some } x \in X \text{ and } \langle u^i, y\rangle = \langle b, v^i\rangle\},$$

where $K$ is the iteration in which the algorithm terminates. As a result, $Y^=_{\mathrm{WE}}$ can be recovered from the optimal solution sets $W_i^*$ of the linear programming problems

$$\begin{aligned} \max\ & \langle u^i, y\rangle, \\ \text{s.t.}\ & y - Cx = 0, \\ & Ax = b, \\ & x \geqslant 0, \end{aligned}$$

$i = 0, 1, 2, \ldots, K-1$.

## 4. Computational experiments

In order to perform some preliminary computational experiments, we have written a prototype VS-FORTRAN code capable of executing the Outer Approximation Algorithm on moderately-small instances of problem (MOLP). We then applied this code to 30 randomly-generated problems.

The code implements Step $k4$ of the Outer Approximation Algorithm by the Horst–Thoai–DeVries method [54]. The univariate searches in Step $k2$ are accomplished by simple bisection search, and linear programming tasks are performed using the simplex method as implemented by the subroutines of IMSL [61].

To construct the data for the $p \times n$ matrix $C$, the $m \times n$ matrix $A$, and the vector $b \in \Re^m$ required for each of the 30 instances of problem (MOLP), pseudorandom numbers from uniform distributions were used. The experimental runs were performed on an ES/9000 model 831 computer.

Some statistics summarizing the results of these computational experiments are displayed in Tables 1 and 2. In Table 1, for each problem, the column $K$ contains

*Table 1.* Computational results for individual problems.

| Problem | m | n | p | K | $\|E\|$ | CPU time (seconds) |
|---|---|---|---|---|---|---|
| 1 | 4 | 11 | 3 | 10 | 12 | 1.65 |
| 2 | 4 | 11 | 3 | 16 | 16 | 2.10 |
| 3 | 4 | 11 | 3 | 19 | 28 | 2.84 |
| 4 | 4 | 11 | 3 | 13 | 11 | 2.14 |
| 5 | 4 | 11 | 3 | 3 | 1 | 0.93 |
| 6 | 5 | 10 | 3 | 8 | 5 | 1.41 |
| 7 | 5 | 10 | 3 | 7 | 3 | 1.52 |
| 8 | 5 | 10 | 3 | 11 | 8 | 2.36 |
| 9 | 5 | 10 | 3 | 12 | 6 | 2.03 |
| 10 | 5 | 10 | 3 | 16 | 17 | 2.91 |
| 11 | 5 | 10 | 2 | 4 | 3 | 1.02 |
| 12 | 5 | 10 | 2 | 4 | 3 | 1.04 |
| 13 | 5 | 10 | 2 | 5 | 4 | 1.12 |
| 14 | 5 | 10 | 2 | 5 | 4 | 1.05 |
| 15 | 5 | 10 | 2 | 8 | 7 | 1.31 |
| 16 | 10 | 15 | 2 | 6 | 5 | 1.88 |
| 17 | 10 | 15 | 2 | 3 | 2 | 1.36 |
| 18 | 10 | 15 | 2 | 3 | 2 | 1.29 |
| 19 | 10 | 15 | 2 | 7 | 6 | 1.99 |
| 20 | 10 | 15 | 2 | 4 | 3 | 1.60 |
| 21 | 10 | 15 | 3 | 3 | 1 | 1.41 |
| 22 | 10 | 15 | 3 | 25 | 48 | 7.53 |
| 23 | 10 | 15 | 3 | 17 | 22 | 5.83 |
| 24 | 10 | 15 | 3 | 22 | 30 | 7.57 |
| 25 | 10 | 15 | 3 | 5 | 5 | 1.75 |
| 26 | 15 | 20 | 3 | 23 | 23 | 13.90 |
| 27 | 15 | 20 | 3 | 35 | 54 | 24.40 |
| 28 | 15 | 20 | 3 | 6 | 5 | 3.48 |
| 29 | 15 | 20 | 3 | 16 | 14 | 8.39 |
| 30 | 15 | 20 | 3 | 29 | 38 | 16.17 |

the iteration number in which the Outer Approximation Algorithm terminated, and the column $|E|$ contains the total number of extreme points in the efficient outcome set. Recall from Section 3 that $K - 1$ equals the total number of inequality cuts added by the algorithm to the initial simplex $S = S^0$ to create $Y$. In Table 2, the 30 problems are grouped by problem size into six groups of five problems each. For each group, by using Table 1, we computed the average number of efficient extreme points contained in the outcome set. These numbers are given in the last column of

*Table 2.* Comparative statistics for domain and outcome sets.

| Problem size | | | Average no. of efficient points | |
|---|---|---|---|---|
| $m$ | $n$ | $p$ | In domain set | In outcome set |
| 4 | 11 | 3 | 17.8 | 13.6 |
| 5 | 10 | 3 | 20.0 | 7.8 |
| 5 | 10 | 2 | 9.0 | 4.2 |
| 10 | 15 | 2 | 13.6 | 3.6 |
| 10 | 15 | 3 | 108.4 | 21.2 |
| 15 | 20 | 3 | 261.4 | 26.8 |

Table 2. For comparison purposes, for each problem size, the column immediately to the left of the last column contains the average number of efficient extreme points contained in the domain set for a group of five similarly-generated problems. We computed these averages with the aid of the domain set-based algorithm ADBASE [33].

Notice from Table 1 that for all but one of the 30 problems, the Outer Approximation Algorithm required fewer than 30 iterations. In addition, the number $|E|$ of efficient extreme points in the outcome sets of these problems is always less than 55. As expected, as $p, m$ or $n$ increases, both $|E|$ and the number of iterations $K$ also increase, although Table 1 does not give sufficient evidence to draw conclusions as to the relative influences of the sizes of $p, m$ and $n$ on $|E|$ or $K$.

Table 2 clearly shows for these 30 problems that over all problem sizes considered, the average number of efficient extreme points in the outcome set is less than the average number of efficient extreme points in the decision set, often considerably so. In addition, from this table, we see that in these 30 problems, as the problem size increases, the ratio of the number of efficient extreme points in the outcome set to the number in the decision set decreases quite rapidly. As a result, while the average number of efficient extreme points in the domain sets of the problems with $m = 15$, $n = 20$ and $p = 3$ grows to 261.4, which is large enough to possibly overwhelm the decision maker, the corresponding average in the outcome set reaches only 26.8.

## 5. Conclusions

We have seen that decision set-based vector maximization approaches for problem (MOLP) are limited by problem size. In particular, as the size of problem (MOLP) grows, the sizes of both the efficient decision set $X_E$ of this problem and the set of extreme points in $X_E$ grow quite rapidly. As a result, generating all of $X_E$ or the set of all extreme points in $X_E$ and presenting the results to the DM can quickly become unworkable as the size of problem (MOLP) grows. This is because either the computations required to generate these sets become impractical, or the results are not useful to the decision maker because he or she is overwhelmed by them.

To attempt to mitigate the effects of problem size, we have presented a finite algorithm, called the Outer Approximation Algorithm, for generating the set of all efficient extreme points in the outcome set, rather than in the decision set, of problem (MOLP). We have seen that solid motivations in theory and in practice have been given in the literature for believing that outcome set-based approaches, if carefully developed, should be superior to decision set-based approaches.

The Outer Approximation Algorithm can be implemented relatively easily by using univariate search methods, linear programming techniques, and any one of several special methods from the global optimization literature for generating new vertex sets as linear inequality cuts are added to the containing polyhedra generated by the algorithm. Furthermore, the Outer Approximation Algorithm does not call for the use of any of the special accounting or backtracking procedures required by many decision set-based approaches. Finally, since the number of objective functions in problem (MOLP) is almost always less than 20, the global optimization literature gives reason to believe that the outer approximation portion of the algorithm should work efficiently.

Preliminary computational experiments that we have performed using a prototype computer code that executes the Outer Approximation Algorithm demonstrate, for 30 problems, the practicality of the algorithm. Those experiments also tangibly demonstrate the usefulness in these 30 problems of using the outcome set approach of the Outer Approximation Algorithm instead of a decision set-based approach. In particular, in the larger problems, considerably fewer efficient extreme points were found, on the average, in the outcome sets than in the decision sets.

As a result, we conclude that the Outer Approximation Algorithm offers significant promise for solving applications of problem (MOLP) more easily than decision set-based methods, and without overwhelming the decision maker. In addition, these results indicate that the Outer Approximation Algorithm offers promise for solving larger instances of problem (MOLP) than can presently be solved by decision set-based vector maximization algorithms.

## References

1. Cohon, J.L. (1978), *Multiobjective Programming and Planning*, Academic Press, New York.
2. Evans, G.W. (1984), An Overview of Techniques for Solving Multiobjective Mathematical Programs, *Management Science* 30, 1268–1282.
3. Goicoechea, A., Hansen, D.R. and Duckstein, L. (1982), *Multiobjective Decision Analysis with Engineering and Business Applications*, John Wiley and Sons, New York.
4. Luc, D.T. (1989), *Theory of Vector Optimization*, Springer Verlag, Berlin/New York.
5. Sawaragi, Y., Nakayama, H. and Tanino, T. (1985), *Theory of Multiobjective Optimization*, Academic Press, Orlando, Florida.
6. Steuer, R.E. (1986), *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley and Sons, New York.

7. Yu, P.L. (1985), *Multiple Criteria Decision Making*, Plenum, New York.
8. Zeleny, M. (1982), *Multiple Criteria Decision Making*, McGraw-Hill, New York.
9. Kuhn, H.W. and Tucker, A.W. (1950), Nonlinear Programming, in J. Neyman (ed.), *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, California, pp. 481–492.
10. Geoffrion, A.M. (1968), Proper Efficiency and the Theory of Vector Maximization, *Journal of Mathematical Analysis and Applications* 22, 618–630.
11. Philip, J. (1972), Algorithms for the Vector Maximization Problem, *Mathematical Programming* 2, 207–229.
12. Evans, J.P. and Steuer, R.E. (1973), Generating Efficient Extreme Points in Linear Multiple Objective Programming: Two Algorithms and Computing Experience, in J.L. Cochrane and M. Zeleny (eds.), *Multiple-Criteria Decision Making*, University of South Carolina Press, Columbia, South Carolina, pp. 349–365.
13. Yu, P.L. and Zeleny, M. (1975), The Set of All Nondominated Solutions in Linear Cases and a Multicriteria Simplex Method, *Journal of Mathematical Analysis and Applications* 49, 430–468.
14. Benson, H.P. (1979), Vector Maximization with Two Objective Functions, *Journal of Optimization Theory and Applications* 28, 253–257.
15. Isermann, H. (1977), The Enumeration of the Set of All Efficient Solutions for a Linear Multiple Objective Program, *Operational Research Quarterly* 28, 711–725.
16. Bitran, G.R. (1979), Theory and Algorithms for Linear Multiple Objective Programs with Zero-One Variables, *Mathematical Programming* 17, 362–390.
17. Villarreal, B. and Karwan, M.H. (1981), Multicriteria Integer Programming: A (Hybrid) Dynamic Programming Recursive Approach, *Mathematical Programming* 21, 204–223.
18. Kostreva, M.M. and Wiecek, M.M. (1993), Time Dependency in Multiple Objective Dynamic Programming, *Journal of Mathematical Analysis and Applications* 173, 289–307.
19. Marcotte, O. and Soland, R.M. (1986), An Interactive Branch-and-Bound Algorithm for Multiple Criteria Optimization, *Management Science* 32, 61–75.
20. Steuer, R.E. (1976), A Five-Phase Procedure for Implementing a Vector-Maximum Algorithm for Multiple Objective Linear Programming Problems, in H. Thiriez and S. Zionts (eds.), *Multiple-Criteria Decision Making: Jouy-en-Josas, France*, Springer Verlag, Berlin/New York.
21. Benson, H.P. (1997), Generating the Efficient Outcome Set in Multiple Objective Linear Programs: The Bicriteria Case, *Acta Mathematica Vietnamica* 22, 29–51.
22. Sayin, S. (1996), An Algorithm Based on Facial Decomposition for Finding the Efficient Set in Multiple Objective Linear Programming, *Operations Research Letters* 19, 87–94.
23. Armand, P. (1993), Finding All Maximal Efficient Faces in Multiobjective Linear Programming, *Mathematical Programming* 61, 357–375.
24. Armand, P. and Malivert, C. (1991), Determination of the Efficient Decision Set in Multiobjective Linear Programming, *Journal of Optimization Theory and Applications* 70, 467–489.
25. Benson, H.P. (1981), Finding an Initial Efficient Extreme Point for a Linear Multiple Objective Program, *The Journal of the Operational Research Society* 32, 495–498.
26. Benson, H.P. (1985), Multiple Objective Linear Programming with Parametric Criteria Coefficients, *Management Science* 31, 461–474.
27. Benson, H.P. and Aksoy, Y. (1991), Using Efficient Feasible Directions in Interactive Multiple Objective Linear Programming, *Operations Research Letters* 10, 203–209.
28. Ecker, J.G. and Kouada, I.A. (1978), Finding All Efficient Extreme Points for Multiple Objective Linear Programs, *Mathematical Programming* 14, 249–261.
29. Ecker, J.G., Hegner, N.S. and Kouada, I.A. (1980), Generating All Maximal Efficient Faces for Multiple Objective Linear Programs, *Journal of Optimization Theory and Applications* 30, 353–381.

30. Gal, T. (1977), A General Method for Determining the Set of All Efficient Solutions to a Linear Vector Maximum Problem, *European Journal of Operational Research* 1, 307–322.
31. Zeleny, M. (1974), *Linear Multiobjective Programming*, Springer Verlag, Berlin/New York.
32. Benson, H.P. and Sayin, S. (1997), Towards Finding Global Representations of the Efficient Set in Multiple Objective Mathematical Programming, *Naval Research Logistics* 44, 47–67.
33. Steuer, R.E. (1989), *ADBASE Multiple Objective Linear Programming Package*, University of Georgia, Athens, Georgia, 1989.
34. Dauer, J.P. and Liu, Y.H. (1990), Solving Multiple Objective Linear Programs in Objective Space, *European Journal of Operational Research* 46, 350–357.
35. Dauer, J.P. and Saleh, O.A. (1990), Constructing the Set of Efficient Objective Values in Multiple Objective Linear Programs, *European Journal of Operational Research* 46, 358–365.
36. Morse, J.N. (1980), Reducing the Size of the Nondominated Set: Pruning by Clustering, *Computers and Operations Research* 7, 55–66.
37. Steuer, R.E. (1976), Multiple Objective Linear Programming with Interval Criterion Weights, *Management Science* 23, 305–316.
38. Steuer, R.E. and Harris, F.W. (1980), Intra-Set Point Generation and Filtering in Decision and Criterion Space, *Computers and Operations Research* 7, 41–53.
39. Benson, H.P. (1995), A Geometrical Analysis of the Efficient Outcome Set in Multiple-Objective Convex Programs with Linear Criterion Functions, *Journal of Global Optimization* 6, 231–251.
40. Dauer, J.P. (1987), Analysis of the Objective Space in Multiple Objective Linear Programming, *Journal of Mathematical Analysis and Applications* 126, 579–593.
41. Dauer, J.P. (1993), On Degeneracy and Collapsing in the Construction of the Set of Objective Values in a Multiple Objective Linear Program, *Annals of Operations Research* 47, 279–292.
42. Gallagher, R.J. and Saleh, O.A. (1995), A Representation of an Efficiency Equivalent Polyhedron for the Objective Set of a Multiple Objective Linear Program, *European Journal of Operational Research* 80, 204–212.
43. Dauer, J.P. and Gallagher, R.J. (1996), A Combined Constraint-Space, Objective-Space Approach for Determining High-Dimensional Maximal Efficient Faces of Multiple Objective Linear Programs, *European Journal of Operational Research* 88, 368–381.
44. Jahn, J., On the Determination of Minimal Facets and Edges of a Polyhedral Set, *Journal of Multi-Criteria Decision Analysis*, to appear.
45. Armann, R. (1989), Solving Multiobjective Programming Problems by Discrete Representation, *Optimization* 20, 483–492.
46. Horst, R. and Tuy, H. (1993), *Global Optimization: Deterministic Approaches*, 2nd edition, Springer Verlag, Berlin/New York.
47. Horst, R. and Pardalos, P., eds. (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht/Boston/London.
48. Benson, H.P. (1996), Deterministic Algorithms for Constrained Concave Minimization: A Unified Critical Survey, *Naval Research Logistics* 43, 765–795.
49. Rockafellar, R.T. (1970), *Convex Analysis*, Princeton University Press, Princeton, New Jersey.
50. Benson, H.P. and Lee, D. (1996), Outcome-Based Algorithm for Optimizing over the Efficient Set of a Bicriteria Linear Programming Problem, *Journal of Optimization Theory and Applications* 88, 77–105.
51. Falk, J.E. and Hoffman, K.L. (1976), A Successive Underestimating Method for Concave Minimization Problems, *Mathematics of Operations Research* 1, 251–259.
52. Tuy, H. (1983), On Outer Approximation Methods for Solving Concave Minimization Problems, *Acta Mathematica Vietnamica* 8, 3–34.
53. Thieu, T.V., Tam, B.T. and Ban, T.V. (1983), An Outer Approximation Method for Globally Minimizing a Concave Function over a Compact Convex Set, *Acta Mathematica Vietnamica* 8, 21–40.

54. Horst, R., Thoai, N.V. and Devries, J. (1988), On Finding the New Vertices and Redundant Constraints in Cutting Plane Algorithms for Global Optimization, *Operations Research Letters* 7, 85–90.
55. Thieu, N.V. (1988), A Finite Method for Globally Minimizing a Concave Function over an Unbounded Polyhedral Convex Set and Its Applications, *Acta Mathematica Hungarica* 52, 21–36.
56. Chen, P.C., Hansen, P. and Jaumard, B. (1991), On-Line and Off-Line Vertex Enumeration by Adjacency Lists, *Operations Research Letters* 10, 403–409.
57. Horst, R. (1991), On the Vertex Enumeration Problem in Cutting Plane Algorithms of Global Optimization, in G. Fandel and H. Gehring (eds.), *Operations Research*, Springer Verlag, Berlin/New York, pp. 13–22.
58. Horst, R. and Thoai, N.V. (1989), Modification, Implementation and Comparison of Three Algorithms for Globally Solving Linearly Constrained Concave Minimization Problems, *Computing* 42, 271–289.
59. Stadler, W., ed. (1988), *Multicriteria Optimization in Engineering and the Sciences*, Plenum Press, New York.
60. Benson, H.P. (1982), Admissible Points of a Convex Polyhedron, *Journal of Optimization Theory and Applications* 38, 341–361.
61. International Mathematical and Statistical Libraries, Inc. (1991), *The IMSL Library Reference Manual*, IMSL, Houston, Texas.