ACTA MATHEMATICA VIETNAMICA
Volume 22, Number 1, 1997, pp. 29-51

GENERATING THE EFFICIENT OUTCOME SET IN MULTIPLE OBJECTIVE LINEAR PROGRAMS: THE BICRITERIA CASE

HAROLD P. BENSON

Dedicated to Hoang Tuy on the occasion of his seventieth birthday

ABSTRACT. This article presents an algorithm called OUTSET for generating the efficient outcome set of a multiple objective linear program (BX) with two objective functions. Unlike standard vector maximization approaches to multiple objective linear programming, OUTSET does not attempt to generate the efficient decision set for problem (BX), which can be large and quite complicated. Instead, it directly generates the efficient faces of the smaller, simpler efficient outcome set. As a result, it is expected that in practice, the OUTSET algorithm will have the potential to allow decision makers to solve bicriteria linear programming problems relatively easily and accurately, including those large-scale problems that heretofore were too large to be amenable to standard vector maximization methods for multiple objective linear programming.

1. INTRODUCTION

A multiple objective mathematical programming problem (P) involves the simultaneous maximization of $p \geq 2$, noncomparable criteria functions over a nonempty set in $\mathbf{R}^n$. Since the criteria functions are, in general, conflicting, the "maximization" aspect of this problem is not a priori well defined. As a result, usually, the decision maker (DM), together with an analyst, seeks a *most preferred* solution, if one exists, to the problem, i.e., a solution that maximizes the DM's value function for the problem over all feasible alternatives. Generally, this value function is not explicitly available or computable. Mathematically, however, even in the absence of this function, in most real-world situations it can be shown that a most preferred solution can be found somewhere in the subset of feasible solutions called the *efficient (*or *nondominated) decision set.* Therefore,

Received October 3, 1996
Key words. Multiple objective linear programming, vector maximization, efficient set, bicriteria linear programming, global optimization.

many of the approaches for solving multiple objective mathematical programs generate all, or at least some, of the efficient decision set for the DM to examine. In this way, inherent tradeoffs among the criteria are revealed, and most-preferred solutions can be sought. Included among these approaches are the vector maximization approach, interactive approaches and several others; see, for instance Cohon [10], Evans [16], Goicoechea et al. [19], Kuhn and Tucker [21], Luc [22], Sawaragi et al. [28], Steuer [32], Yu [34], Zeleny [36] and references therein.

The vector maximization approach for solving problem (P) is one of the oldest and most familiar approaches for solving the problem. In typical procedures that use this approach, either all of the efficient decision set of problem (P) or, substantial portions thereof, are first generated with the aid of a computerized routine. Subsequently, the DM chooses a most preferred solution from the generated set.

The efficient decision set of problem (P), however, 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 in only certain special cases; see, e.g., Benson [6], Isermann [20], Ecker et al. [15], Bitran [9], Yu and Zeleny [35], Villarreal and Karwan [33]. 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, e.g., Evans and Steuer [17]. Marcotte and Soland [24] and Steuer [32]. Furthermore, the sheer size of the efficient decision set becomes so huge that it becomes difficult to describe to the DM and can overwhelm the DM to the extent that he or she is not able to choose a most preferred solution from it [31].

In response to this, in recent years, a handful of researchers has begun to turn their attention to the mathematics and tools for generating all or portions of the *efficient outcome set*, rather than the efficient decision set, for problem (P); see Benson [5], Benson and Sayin [8], Dauer [11, 12], Dauer and Liu [13], Dauer and Saleh [14], Gallagher and Saleh [18]. Although it is also nonconvex, the efficient outcome set for problem (P) has a size and structure that are invariably much smaller and simpler than the size and structure of the efficient decision set. This is largely due to the fact that the efficient outcome set lies the space $\mathbf{R}^p$ of the objective function value, rather than in the decision space $\mathbf{R}^n$, and p is typically smaller than n by factors of 10, 100, or even more; see e.g., Benson [5], Dauer and Liu [13] and Steuer [32].

In this article, we present and validate an algorithm called OUTSET

for generating the entire efficient outcome set of a multiple objective linear program (BX) with two objective functions. To our knowledge, this algorithm is the first of its type. The OUTSET algorithm exploits the fact that the dimension of the outcome set of a bicriteria linear programming problem is at most two.

Unlike standard multiple objective linear programming algorithms (e.g., Armand and Malivert [1], Ecker et al. [15], Evans and Steuer [17], Isermann [20], Philip [26], Steuer [31, 32], Yu and Zeleny [35], Zeleny [36]), the OUTSET algorithm for problem (BX) generates the efficient outcome set, rather than the efficient decision set, for the problem. There are three important advantages that result from this approach.

First, it has been shown that in practice, decision makers base their decisions on outcome set values rather than on values of points in the decision set [8, 13, 14]. Hence, the output of the new algorithm is expected to be more beneficial to the DM than the output given by typical decision set-based methods.

Second, it is well known that frequently many points in the efficient decision set of problem (BX) are mapped by the criteria of the problem onto a single outcome in the efficient outcome set [2, 5, 11, 13, 14]. As a result, by generating points directly from the efficient outcome set, OUTSET avoids risking redundant calculations from the decision set that would be of little or no use to the DM.

Third, since the efficient outcome set of problem (BX) invariably has a much simpler structure and smaller size than the efficient decision set, the output of the OUTSET algorithm can typically be expected to be generated much more easily and quickly than the output of decision set-baset methods, and with a smaller probability of overwhelming the DM.

Furthermore, as we shall see, OUTSET can be implemented with well-known linear programming techniques. In addition, it avoids the need for the complicated bookkeeping or backtracking required by many of the standard multiple objective linear programming methods (cf., e.g., Ecker et al. [15], Evans and Steuer [17], Steuer [32]).

As a result, it is expected that the new OUTSET algorithm will have the potential to allow a DM and an analyst to solve large-scale bicriteria linear programming problems that heretofore were too large to be amenable to the vector maximization approach.

This article is organized as follows. In the next section, notation and preliminary results are presented. The OUTSET algorithm is presented

in Section 3. Section 4 gives a geometrically-motivated proof of the convergence of OUTSET. In Section 5 some practical benefits to be expected from the algorithm are illustrated via an example. Some concluding remarks are given in the last section.

2. NOTATION AND PRELIMINARIES

Let $X \subseteq \mathbf{R}^n$ be a nonempty, compact polyhedron. Assume without loss of generality that

$$X = \{x \in \mathbf{R}^n \mid Ax = d,\ x \geq 0\},$$

where A is an $m \times n$ matrix of real numbers and $d \in \mathbf{R}^m$, Let C denote a $2 \times n$ matrix with rows $c_1^T, c_2^T \in \mathbf{R}^n$. Then the *bicriteria linear programming problem* may be written

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

The function $\langle c_1, x\rangle$, $\langle c_2, x\rangle$ are called the *objective functions or criteria* of problem (BX), and X is called the *feasible decision set* of the problem.

Since the criteria of problem (BX) are generally conflicting, the "vector maximization" operator VMAX of problem (BX) is not a priori well defined. Usually, a DM, with the aid of an analyst, seeks a most preferred solution x^*, if one exists, to problem (BX); i.e., a solution $x^* \in X$ that maximum $v[\langle c_1, x\rangle, \langle c_2, x\rangle]$ over X, where $v : \mathbf{R}^2 \to \mathbf{R}$ is the preference (or value) function of the DM for problem (BX). Unfortunately, the preference functions v of the DM usually is not explicitly available or computable [28, 29, 32]. In the absence of v, to help the DM find a most preferred solution, the vector maximization approach for problem (BX) generates the entire efficient decision set of the problem, where the efficient decision set is defined as follows.

Definition 2.1. The *efficient* (or *nondenominated) decision set* X_E for problem (BX) is the set of all points $x^0 \in X$ for which there exists no point $x \in X$ such that $Cx \geq Cx^0$ and $Cx \neq Cx^0$.

The rationale for generating X_E for the DM to search for a most preferred solution in the absence of v stems from the fact that X_E will contain a most preferred solution whenever v is known to be coordinatewise nondecreasing (cf., e.g., Benson and Aksoy [7], Soland [30] and Steuer [32]). In practical situations, v may not be known explicitly, but, invariably, the DM's preferences dictate that v must be nondecreasing in its arguments.

Direct generation of X_E, however, can be a daunting task. This is because X_E is generally a large, complicated nonconvex set consisting of the union of large numbers of faces of X. These faces form a connected set, but their dimensions and locations within X can be widely [1, 5, 32, 34].

This is one of the main reasons that the algorithm OUTSET avoids direct generation of X_E. Instead of working in the decision space $\mathbf{R}^n$, OUTSET works in the outcome space $\mathbf{R}^2$.

Let $Y = \{Cx | x \in X\}$. From Rockafellar [27], Y is a nonempty, compact polyhedron in the outcome space $\mathbf{R}^2$ of problem (BX). We refer to Y as the *feasible outcome set* (or simply, the *outcome set*) of problem (BX). Notice that Y is the image of X under the linear mappings C. Furthermore, it is easy to show that the image CX_E of the efficient decision set X_E under C is identical to the set of efficient points Y_E of the bicriteria linear programming problem

$$\text{(BY)} \qquad \text{VMAX}: \ I_2 y, \ \text{s.t.} \ y \in Y,$$

where I_2 denotes the 2×2 identity matrix (cf. Proposition 2.1 below). The set Y_E is thus also called the *efficient outcome set* of problem (BX).

The algorithm OUTSET indirectly, rather than directly, generates the efficient decision set X_E by finding the smaller, simpler efficient outcome set $Y_E = CX_E$ instead. In particular, OUTSET directly generates the efficient faces of Y_E one at a time, without complicated bookkeeping or backtracking. In this way, as we shall see, generally far fewer efficient faces need to be generated than if X_E were directly generated. Furthermore, OUTSET needs to only generate faces of Y_E of dimensions zero and one, i.e., efficient extreme points and edges of Y. In contrast, direct face generation of X_E could require generating large numbers of faces of much higher dimension (cf., e.g. [5, 11, 13, 14]).

In the remainder of this section, we review some preliminary results that will be needed to develop and validate the new algorithm OUTSET. Recall that a face of a convex set V is a convex subset F of V such that any closed line segment in V with at least one relative interior point in F must have both of its endpoints in F. The zero-dimensional faces of V are called *extreme points* of V.

The following result will be used frequently in the sequel. It follows directly from the definitions.

Proposition 2.1. (a) *For any* $y^0 \in Y_E$, *if* $x^0 \in X$ *satisfies* $Cx^0 = y^0$, *then* $x^0 \in X_E$.

(b) *For any* $x^0 \in X_E$, *if* $y^0 = Cx^0$, *then* $y^0 \in Y_E$.

It is well known that X_E and Y_E consist of unions of efficient faces of X and Y, respectively. However, the image under C of an efficient face of X need not be an efficient face of Y. In particular, large numbers of efficient faces X_F of X may have images CX_F under C that are *strict subsets* of the relative interiors of efficient faces of Y [5, 11]. However, the inverse mapping of C successfully maps efficient faces of Y onto efficient faces of X, as shown by the following result.

Theorem 2.1. *Let* Y_F *be an arbitrary efficient face of* Y. *Then* $X_F = \{x \in X \mid Cx \in Y_F\}$ *is an efficient face of* X, *and* $\dim X_F \geq \dim Y_F$.

Theorem 2.1 follows immediately from Theorem 3.4 in [5]. Notice that it is readily apparent from Theorem 2.1 and the discussion preceding it that Y_E must always contain the same number or a smaller number of faces than X_E, and that the dimension of the facial pre-image X_F of any efficient face Y_F of Y will always be greater than or equal to the dimension of Y_F. In practice, in fact, both the numbers and the dimensions of the faces in X_E can far exceed those of the faces in Y_E [5, 11, 13].

For each $i = 1, 2$, let

$$M_i = \max\langle c_i, x\rangle, \text{ s.t. } x \in X, \tag{1}$$

let m_2 equal the optimal value of the linear program

$$\begin{aligned} &\max\langle c_2, x\rangle, \\ &\text{s.t. } \langle c_1, x\rangle = M_1, \\ &\qquad x \in X, \end{aligned} \tag{Q}$$

and let $I = \{b \in \mathbf{R} \mid m_2 \leq b \leq M_2\}$. For each $b \in I$ consider the linear programming problem

$$\begin{aligned} &\max\langle c_1, x\rangle, \\ &\text{s.t. } \langle c_2, x\rangle \geq b, \\ &\qquad x \in X, \end{aligned} \tag{P_b}$$

and let $w(b)$ denote the optimal value of this problem. The following results will be used by the OUTSET algorithm to help to systematically generate faces of Y_E.

Theorem 2.2. *A point $x^0 \in \mathbf{R}^n$ satisfies $x^0 \in X_E$ if and only if x^0 is an optimal solution to problem (P_b) for some $b \in I$.*

Theorem 2.3. (a) *The function w is continuous, concave and piecewise linear on I.*

(b) *For each $b \in I$, $w(b)$ is equal to the optimal value of the dual linear program to problem (P_b), where this dual linear program may be written*

$$(Q_b) \qquad \begin{aligned} &\min -bu + \langle d, q\rangle, \\ &\text{s.t. } -c_2 u + A^T q \geq c_1, \\ &\qquad u \geq 0. \end{aligned}$$

Theorem 2.2 is fairly well known it follows, for instance, directly from the main result in [6]. Theorem 2.3 is a standard linear programming result.

In some cases, $X_E = X$ may occur, i.e., problem (BX) may be *completely efficient*. Notice that $X_E = X$ if and only if $Y_E = Y$. To detect whether or not problem (BX) is completely efficient, the linear programming test in the following result can be used. Let $e = [1, 1]^T \in \mathbf{R}^2$.

Theorem 2.4. *Problem* (BX) *is completely efficient if and only if the optimal t of the linear program*

$$(T) \qquad \begin{aligned} &\min \langle d, q\rangle, \\ &s.t. \; -C^T u + A^T r - z = C^T e, \\ &\qquad A^T q - z \geq 0, \\ &\qquad u, z \geq 0 \end{aligned}$$

is equal to 0.

Theorem 2.4 follows from [4]. From [4], since X is nonempty and compact, the value of t in problem (T) is always nonnegative and finite.

3. THE ALGORITHM OUTSET

From [27] and [32], since the outcome set is a nonempty, compact polyhedron in $\mathbf{R}^2$, either Y is completely efficient, Y_E is a single extreme point of Y, or Y_E consists of one or a set of connected one-dimensional faces (edges) of Y. The algorithm OUTSET first uses simple linear programming to detect if Y is completely efficient or if Y_E is a singleton. If either case is found to be true, the algorithm indicates this, provides Y_E

as output and terminates. Otherwise, the algorithm systematically generates and provides the edges of Y_E, one at a time, until all of Y_E has been found. Each edge of Y_E is described by providing the coordinates of the endpoints of the line segment in $\mathbf{R}^2$ that the edge forms. For each additional edge of Y_E that is found, its first endpoint is identical to the most-recently generated endpoint of the most-recently found edge of Y_E. The second endpoint of this edge is found by solving two simple linear programming problems. Since Y_E consists of a finite number of faces, the algorithm will always be finite.

The algorithm may be stated as follows.

Algorithm OUTSET

Step 1. Compute the optimal objective function value t of the linear program (T). If $t = 0$, stop: $Y_E = \{Cx \mid x \in X\}$. Otherwise, continue.

Step 2. Compute M_1 and M_2 by finding the optimal objective function values for the linear program (1) with $i = 1$ and $i = 2$, respectively. Find any optimal solution x^1 and the optimal objective function value m_2 to the linear program (Q).

Step 3. If $m_2 = M_2$, stop : $Y_E = \{(M_1, M_2)^T\}$. Otherwise, let $b_1 = m_2$, $\alpha_1 = M_1$ and $Y_E = \phi$, set $k = 1$, and continue.

Step 4. Set $b = b_k$ and $\alpha = \alpha_k$ and find any optimal solution (u_k, q^k) to the linear program

$$(\mathrm{P}_{b,\alpha}) \qquad \begin{aligned} &\max u, \\ \text{s.t. } &-bu + \langle d, q\rangle = \alpha, \\ &-c_2 u + A^T q \geq c_1, \\ &u \geq 0, \end{aligned}$$

Step 5. Find any optimal solution x^{k+1} to the linear program

$$\begin{aligned} &\max \langle c_2, x\rangle, \\ \text{s.t. } &\langle c_1, x\rangle + u_k \langle c_2, x\rangle = \langle d, q^k\rangle, \\ &x \in X, \end{aligned}$$

and set $b_{k+1} = \langle c_2, x^{k+1}\rangle$, $\alpha_{k+1} = \langle c_1, x^{k+1}\rangle$.

Step 6. Set $Y_E = Y_E \cup L_k$, where L_k is the line segment in $\mathbf{R}^2$ with endpoints $(\alpha_i, b_i)^T$, $i = k, k+1$. If $b_{k+1} = M_2$, stop. Otherwise, set $k = k + 1$ and go to Step 4.

Step 1 of the algorithm tests problem (BX) for complete efficiency. If $t = 0$ in Step 1, then, by Theorem 2.4, complete efficiency holds and the algorithm terminates. Otherwise, by Theorem 2.4, complete efficiency does not hold. In this case, the algorithm proceeds to Step 2 where the interval $I = \{b \in \mathbf{R} | m_2 \leq b \leq M_2\}$ and an initial point $y^1 = (\langle c_1, x^1\rangle, \langle c_2, x^1\rangle)^T = (M_1, m_2)^T \in Y_E$ are found. If $m_2 = M_2$, then from Theorem 2.2 and the definitions of M_1 and M_2, Y_E consists of the single extreme point y^1 of Y. When this occurs, the algorithm stops in Step 3 with the indication that $Y_E = \{(M_1, M_2)^T\}$.

The iterative Steps 4 through 6 are executed when Y_E consists of one or more edges $L_1, L_2, \ldots, L_{\overline{k}}$ of Y. As we shall see later, when Step 4 is executed for the first time, the first endpoint $(\alpha_1, b_1)^T = (w(b_1), b_1)^T$ of L_1 is available from Step 3. Subsequently, as we shall see, at the beginning of the kth execution of Step 4, the first endpoint $(\alpha_k, b_k)^T$ of $L_k \in Y_E$ is available from the $(k-1)$st execution of Step 6.

For each $k \geq 1$, given the first endpoint $(\alpha_k, b_k)^T$ of L_k, the goal of Steps 4-6 is to generate the second endpoint $(\alpha_{k+1}, b_{k+1})^T$ of L_k and to add L_k to the description of Y_E. To accomplish this, first in Step 4 a linear program is solved to find the values of $u_k \in \mathbf{R}$ and $q^k \in \mathbf{R}^m$. As we shall see later, the values for u_k and q^k calculated in Step 4 provide the implicit description

$$L_k = \{y \in Y \mid y_1 + u_k y_2 = \langle d, q^k\rangle\} \tag{2}$$

of the efficient edge L_k of Y.

Next, given u_k and q^k, a second linear program is solved in Step 5 for an optimal solution x^{k+1}. As we shall see later, x^{k+1} provides the second endpoint $(\alpha_{k+1}, b_{k+1})^T$ of L_k via the equations

$$\alpha_{k+1} = \langle c_1, x^{k+1}\rangle, \quad b_{k+1} = \langle c_2, x^{k+1}\rangle.$$

Furthermore, we shall see that these values of α_{k+1} and b_{k+1} satisfy $\alpha_{k+1} = w(b_{k+1})$, where $b_{k+1} > b_k$.

In Step 6, the edge $L_k \in \mathbf{R}^2$ is added to Y_E. Notice that L_k is explicitly described in Y_E as the line segment in $\mathbf{R}^2$ with endpoints $(\alpha_i, b_i)^T$, $i = k, k+1$. If $b_{k+1} = M_2$, then, since $b_{k+1} = \langle c_2, x^{k+1}\rangle$ and $b_{i+1} > b_i$ for all i, Theorem 2.2 and the definition of M_2 imply that all of Y_E has been identified, and the algorithm terminates. Otherwise, the generation of the efficient edges of Y_E continues with another iteration of Steps 4-6.

Remark 3.1. From Theorem 2.2, it is possible to generate Y_E by solving linear program (P_b) parametrically for all $b \in I$. The algorithm OUTSET, however, does not take this approach because it is not as efficient computationally as the approach used in OUTSET (cf. Section 5).

4. Convergence

In this section, we will prove via geometrically-motivated arguments that the algorithm OUTSET is finite and always generates all of the efficient outcome set for problem (BX). To accomplish this, we must first prove some preliminary lemmas.

Lemma 4.1. *In the algorithm OUTSET, for each $k \geq 0$, x^{k+1} is an optimal solution to the linear program (P_b) at $b = b_{k+1}$. Furthermore, for each $k \geq 1$, and for each $\bar{b} \in \{b \in \mathbf{R} \mid b_k \leq b \leq b_{k+1}\}$, (u_k, q^k) is an optimal solution to the dual linear program (Q_b) to problem (P_b) at $b = \bar{b}$.*

Proof. Notice from Step 2 of the OUTSET algorithm that x^1 is an optimal solution to problem (Q), and from Steps 2 and 3 that $b_1 = m_2$, where m_2 is the optimal objective function value of problem (Q). It follows that with $b = b_1$, x^1 is a feasible solution to problem (P_b) and $\langle c_1, x^1 \rangle = M_1$. From the definition of M_1, we conclude that x^1 is an optimal solution to problem (P_b) at $b = b_1$.

Now assume that $k \geq 1$. From the definitions of b_{k+1} and x^{k+1} in Step 5 of OUTSET, it follows that

$$\langle c_2, x^{k+1} \rangle = b_{k+1}, \tag{3}$$

$$\langle c_1, x^{k+1} \rangle + u_k \langle c_2, x^{k+1} \rangle = \langle d, q^k \rangle, \tag{4}$$

$$x^{k+1} \in X. \tag{5}$$

From (3) and (5), x^{k+1} is a feasible solution to problem (P_b) at $b = b_{k+1}$. From (3) and (4), the objective function value $\langle c_1, x^{k+1} \rangle$ of x^{k+1} in this problem satisfies

$$\langle c_1, x^{k+1} \rangle = -b_{k+1} u_k + \langle d, q^k \rangle. \tag{6}$$

From Step 4 of OUTSET, (u_k, q^k) is a feasible solution to problem (Q_b) at $b = b_{k+1}$. From (6), since x^{k+1} is a feasible solution to the dual linear (P_b) to problem (Q_b) at $b = b_{k+1}$, this and duality theory of linear programming [25] imply that x^{k+1} and (u_k, q^k) are optimal solutions to problems (P_b) and (Q_b), respectively, at $b = b_{k+1}$.

From Step 4 of OUTSET, (u_k, q^k) is a feasible solution to problem $(P_{b,\alpha})$ when $b = b_k$ and $\alpha = \alpha_k$. From Step 2 and 3 of OUTSET, x^1 is a feasible solution to problem (P_b) at $b = b_1$ with objective function value $\langle c_1, x^1 \rangle = M_1$. By definition of M_1, this implies that x^1 is an optimal solution to problem (P_b) at $b = b_1$, so that $w(b_1) = M_1$. By Step 3 of OUTSET, it follows that $\alpha_1 = w(b_1)$. Furthermore, from Step 5 of OUTSET, since x^{k+1} is an optimal solution to problem (P_b) at $b = b_{k+1}$, $w(b_{k+1}) = \langle c_1, x^{k+1} \rangle = \alpha_{k+1}$. It follows that $b = b_k$ and $\alpha = \alpha_k = w(b_k)$ in problem $(P_{b,\alpha})$ in Step 4 of the algorithm.

Since (u_k, q^k) is a feasible solution to problem $(P_{b,\alpha})$ when $b = b_k$ and $\alpha = \alpha_k$, and since $\alpha_k = w(b_k)$, it follows that (u_k, q^k) is a feasible solution to problem (Q_b) at $b = b_k$ that satisfies

$$-b_k u_k + \langle d, q^k \rangle = w(b_k).$$

By Theorem 2.3 (b), this implies that (u_k, q^k) is an optimal solution to problem (Q_b) at $b = b_k$.

Assume now that $\bar{b} \in \mathbf{R}$ satisfies $b_k \leq \bar{b} \leq b_{k+1}$. Then, for some $\theta \in \mathbf{R}$, $\bar{b} = \theta b_k + (1 - \theta) b_{k+1}$. From Theorem 2.3(a), this implies that

$$w(\bar{b}) \geq \theta w(b_k) + (1 - \theta) w(b_{k+1}).$$

Since (u_k, q^k) is an optimal solution to problem (Q_b) for both $b = b_k$ and $b = b_{k+1}$, this inequality may be written

$$\begin{aligned} w(\bar{b}) &\geq \theta(-b_k u_k + \langle d, q^k \rangle) + (1 - \theta)(-b_{k+1} u_k + \langle d, q^k \rangle) \\ &= -\bar{b} u_k + \langle d, q^k \rangle. \end{aligned}$$

Because $\langle u_k, q^k \rangle$ is a feasible solution to problem (Q_b) at $b = \bar{b}$, together with Theorem 2.3(b), this implies that (u_k, q^k) is an optimal solution to this problem. □

Remark 4.1. Notice from the proof of Lemma 4.1 that in OUTSET, for each $k \geq 1$, $\alpha_k = w(b_k)$. From linear programming theory [3] and Step 4 of the algorithm, this implies that for each $k \geq 1$, $-u_k$ is the right-hand derivative of w at b_k. By Theorem 2.3(a), this implies that for each $k \geq 1$, a $\delta_k > 0$ exists such that $w(\bar{b}) = -\bar{b} u_k + \langle d, q^k \rangle$ for all $\bar{b}$ such that $b_k \leq \bar{b} \leq b_k + \delta_k$.

Remark 4.2. For each $k \geq 1$, from Lemma 4.1 it follows that $\langle c_1, x^k \rangle = -b_k u_k + \langle d, q^k \rangle$. Furthermore, from the proof of Lemma 4.1, $\langle c_2, x^k \rangle = b_k$

for each $k \geq 1$. It follows that for each $k \geq 1$, $\langle c_1, x^k \rangle + u_k \langle c_2, x^k \rangle = \langle d, q^k \rangle$, so that x^k is a feasible solution to the linear program solved in Step 5 of the OUTSET algorithm. From Step 5 of the algorithm, this implies that for each $k \geq 1$, $b_k = \langle c_2, x^k \rangle \leq \langle c_2, x^{k+1} \rangle = b_{k+1}$.

Lemma 4.2. *For each $k \geq 1$, the scalar u_k computed in Step 4 of the OUTSET algorithm is positive.*

Proof. Assume that $k \geq 1$ and choose $\bar{b} \in \mathbf{R}$ such that $b_k < \bar{b} \leq b_k + \delta_k$, where δ_k is chosen as in Remark 4.1. Let $x^{\bar{b}}$ denote an optimal solution to the linear program (P_b) at $b = \bar{b}$, and suppose that $\hat{x}$ maximizes $\langle c_1, x \rangle$ over X. Then from Step 4 of the algorithm, Remark 4.1 and the definition of w,

$$\langle c_1, x^{\bar{b}} \rangle = -\bar{b} u_k + \langle d, q^k \rangle, \tag{7}$$

where

$$-c_2 u_k + A^T q^k \geq c_1. \tag{8}$$

From (8), q^k is a feasible solution to the linear program

$$\begin{aligned} &\min \langle d, q \rangle, \\ &\text{s.t. } A^T q \geq c_1 + u_k c_2. \end{aligned}$$

Since $\hat{x} \in X$, $\hat{x}$ is a feasible solution to the dual linear program to the latter problem, which may be written

$$\begin{aligned} &\max \langle c_1, x \rangle + u_k \langle c_2, x \rangle, \\ &\text{s.t. } x \in X. \end{aligned}$$

Therefore, by duality theory of linear programming [25],

$$\langle c_1, \hat{x} \rangle + u_k \langle c_2, \hat{x} \rangle \leq \langle d, q^k \rangle. \tag{9}$$

From (7) and (9)

$$\left[\langle c_1, x^{\bar{b}} \rangle - \langle c_1, \hat{x} \rangle \right] + u_k \left[\bar{b} - \langle c_2, \hat{x} \rangle \right] \geq 0. \tag{10}$$

Since $\bar{b} > b_k$, and $k \geq 1$, Remark 4.2 implies that $\bar{b} > b_1$. Therefore, $\langle c_2, x^{\bar{b}} \rangle > b_1$. From the definitions of M_1, m_2 and $\hat{x}$, since $b_1 = m_2$, this implies that

$$\langle c_1, x^{\bar{b}} \rangle - \langle c_1, \hat{x} \rangle < 0. \tag{11}$$

Furthermore, since $\hat{x} \in X$, $\langle c_1, \hat{x} \rangle = M_1$ and $\bar{b} > b_1 = m_2$, the definition of m_2 implies that

$$\bar{b} - \langle c_2, \hat{x} \rangle > 0. \tag{12}$$

Together with (10), the inequalities (11)-(12) imply that $u_k > 0$. □

Remark 4.3. Using Lemma 4.2, it is easy to show that the optimal value function $w(b)$ of problem (P_b) is strictly decreasing on I.

Lemma 4.3. *For each $k \geq 1$, the values of b_k and b_{k+1} computed in the OUTSET algorithm satisfy $b_k < b_{k+1}$. Furthermore, for each $k \geq 1$ and $\bar{b} \in \mathbf{R}$ such that $b_k \leq \bar{b} \leq b_{k+1}$, $w(\bar{b}) = -\bar{b}u_k + \langle d, q^k \rangle$.*

Proof. Assume that $k \geq 1$. From Remark 4.1, we may choose a number $\hat{b} > b_k$ such that $w(\hat{b}) = -\hat{b}u_k + \langle d, q^k \rangle$. Then, if we let $x^{\hat{b}}$ denote an optimal solution to problem (P_b) at $b = \hat{b}$, it follows that

$$\langle c_1, x^{\hat{b}} \rangle = -\hat{b}u_k + \langle d, q^k \rangle. \tag{13}$$

From Step 4 of OUTSET, (u_k, q^k) is a feasible solution to problem (Q_b) at $b = \hat{b}$. By (13), since problems (P_b) and (Q_b) at $b = \hat{b}$ are linear programming duals of one another, this implies that (u_k, q^k) is an optimal solution to problem (Q_b) at $b = \hat{b}$ [25]. Therefore, by the complementary slackness property of linear programming [25], $u_k(\langle c_2, x^{\hat{b}} \rangle - \hat{b}) = 0$. From Lemma 4.2, this implies that $\langle c_2, x^{\hat{b}} \rangle = \hat{b}$. By substituting $\langle c_2, x^{\hat{b}} \rangle$ for $\hat{b}$ in (13), we obtain that

$$\langle c_1, x^{\hat{b}} \rangle + u_k \langle c_2, x^{\hat{b}} \rangle = \langle d, q^k \rangle.$$

Since $x^{\hat{b}} \in X$, this equality implies that $x^{\hat{b}}$ is a feasible solution for the linear program solved in Step 5 of the algorithm. From the definitions of x^{k+1} and b_{k+1} in Step 5, it follows that $\langle c_2, x^{\hat{b}} \rangle \leq b_{k+1}$. Since $\langle c_2, x^{\hat{b}} \rangle =$

$\hat{b} > b_k$, this implies that $b_k < b_{k+1}$. The second statement in the lemma is an immediate consequence of Theorem 2.3(b) and Lemma 4.1. □

Remark 4.4. From Lemmas 4.1-4.3 and Remark 4.1, we see the following. First, each execution of Steps 4-6 of the OUTSET algorithm yields the additional linear piece of the graph of the function w given by the line segment in $\mathbf{R}^2$ with endpoints $[b_k, w(b_k)]$ and $[b_{k+1}, w(b_{k+1})]$. Second, this line segment has slope $-u_k < 0$ and lies on the whose equation is given by

$$w(b) = -u_k b + \langle d, q^k \rangle.$$

The next result shows that this line segment corresponds to the efficient edge L_k given in Step 6 of the algorithm and implicitly described by (2).

Theorem 4.1. *For each $k \geq 1$, the line segment L_k given in Step 6 of the OUTSET algorithm is an efficient face of Y, and L_k can be implicitly described by (2).*

Proof. Assume that $k \geq 1$. From Remark 4.1 and Step 6 of the algorithm, L_k is the line segment in $\mathbf{R}^2$ with endpoints $[w(b_k), b_k]$ and $[w(b_{k+1}, b_{k+1}]$. Let F_k denote the optimal solution set of the linear program (G_k) given by

$$(\mathrm{G}_k) \qquad \begin{aligned} &\max \langle y_1 + u_k y_2 \rangle, \\ &\text{s.t. } y \in Y, \end{aligned}$$

where u_k is generated in Step 4 of the algorithm. By Lemma 4.2, $u_k > 0$. From Yu [34] and linear programming theory, this implies that F_k is an efficient face of Y. To prove the theorem, we will show that $F_k = L_k = \{y \in Y \,|\, y_1 + u_k y_2 = \langle d, q^k \rangle\}$.

Towards this end, assume that $[w(\hat{b}), \hat{b}] \in L_k$. Then $b_k \leq \hat{b} \leq b_{k+1}$ and, from Lemma 4.3, $w(\hat{b}) + u_k \hat{b} = \langle d, q^k \rangle$. Choose an optimal solution $x^{\hat{b}}$ to problem (P_b) at $b = \hat{b}$. Then, by definition of w, $w(\hat{b}) = \langle c_1, x^{\hat{b}} \rangle$. Furthermore, from the proof of Lemma 4.3, $\langle c_2, x^{\hat{b}} \rangle = \hat{b}$. As a result, we see that

$$\langle c_1, x^{\hat{b}} \rangle + u_k \langle c_2, x^{\hat{b}} \rangle = \langle d, q^k \rangle. \tag{14}$$

In addition, notice from the derivation of (9) in the proof of Lemma 4.2 that for all $x \in X$,

$$\langle c_1, x \rangle + u_k \langle c_2, x \rangle \leq \langle d, q^k \rangle. \tag{15}$$

Since $x^{\hat{b}} \in X$, (14) and (15) imply that $\langle d, q^k \rangle$ is the optimal value of problem (G_k) and $F_k = \{y \in Y | \ y_1 + u_k y_2 = \langle d, q^k \rangle\}$. Furthermore, since $w(\hat{b}) + u_k \hat{b} = \langle d, q^k \rangle$, where $w(\hat{b}) = \langle c_1, x^{\hat{b}} \rangle$ and $\hat{b} = \langle c_2, x^{\hat{b}} \rangle$, we also see that $[w(\hat{b}), \hat{b}] \in F_k$, so that $L_k \subseteq \{y \in Y \mid y_1 + u_k y_2 = \langle d, q^k \rangle\}$.

To conclude the proof, we must show that $\{y \in Y \mid y_1 + u_k y_2 = \langle d, q^k \rangle\} \subseteq L_k$. To show this, suppose that $y \in Y$ and $y_1 + u_k y_2 = \langle d, q^k \rangle$.

First, we will show that $b_k \leq y_2 \leq b_{k+1}$. Towards this end, assume, to the contrary, that y_2 fails to satisfy $b_k \leq y_2 \leq b_{k+1}$. Then, either $y_2 \geq m_2$ or $y_2 < m_2$.

Case 1. $y_2 \geq m_2$. Then, since $y \in Y$, $y_2 \in I$ must hold. By Lemma 4.3, we may choose $a\, \bar{b} \in \mathbf{R}$ such that $b_k < \bar{b} < b_{k+1}$, and $\bar{b}$ will satisfy $w(\bar{b}) = -\bar{b} u_k + \langle d, q^k \rangle$. Therefore, $w'(\bar{b}) = -u_k$ will hold. From Theorem 6.1.2 in Mangasarian [23] and Theorem 2.3(a), this implies that

$$w(y_2) \leq w(\bar{b}) - u_k(y_2 - \bar{b}),$$

or, equivalently,

$$w(y_2) + u_k y_2 \leq w(\bar{b}) + u_k \bar{b}. \tag{16}$$

Suppose that (16) holds as an equality. Since $w(\bar{b}) = -\bar{b} u_k + \langle d, q^k \rangle$, this implies that

$$w(y_2) + u_k y_2 = \langle d, q^k \rangle.$$

By Remarks 4.3 and 4.4, this implies that y_2 must satisfy $b_k \leq y_2 \leq b_{k+1}$, which is a contradiction. Therefore, (16) must hold as a strict inequality.

Since $w(\bar{b}) = -\bar{b} u_k + \langle d, q^k \rangle$, the fact that (16) must hold as a strict inequality implies that

$$w(y_2) + u_k y_2 < \langle d, q^k \rangle. \tag{17}$$

Notice by definition of $w(y_2)$ that since $y \in Y$, we know that $y_1 \leq w(y_2)$ must hold. Combining this with (17), we obtain

$$y_1 + u_k y_2 < \langle d, q^k \rangle,$$

which is a contradiction. Therefore, this case cannot hold.

Case 2. $y_2 < m_2$. From Step 3 of the algorithm, $b_1 = m_2 \in I$. By applying the argument used in Case 1 with $[w(b_1), b_1] \in Y$ playing the role of y, we obtain

(18) $$w(b_1) + u_k b_1 < \langle d, q^k \rangle.$$

By Lemma 4.2, $u_k > 0$. Since $y_2 < m_2 = b_1$, this implies that $u_k y_2 <$ $u_k b_1$. Furthermore, from the definitions of w, M_1 and m_2, we obtain tha $w(b_1) = M_1$. Since $y \in Y$, by the definition of M_1, this implies tha $y_1 \leq w(b_1)$. Combining this with (18) and the fact that $u_k y_2 < u_k b$ yields the conclusion that

$$y_1 + u_k y_2 < \langle d, q^k \rangle,$$

which is a contradiction. Therefore, this case cannot hold.

The assumption that y_2 fails to satisfy $b_k \leq y_2 \leq b_{k+1}$ has led to the false conclusion that neither Case 1 nor Case 2 holds. Therefore, it follows that $b_k \leq y_2 \leq b_{k+1}$.

From Lemma 4.3, since $b_k \leq y_2 \leq b_{k+1}$, it follows that

$$w(y_2) = -u_k y_2 + \langle d, q^k \rangle.$$

From this and the equation $y_1 + u_k y_2 = \langle d, q^k \rangle$, we conclude that $y_1 = w(y_2)$. Therefore,

$$y_1 = w(y_2) = -u_k y_2 + \langle d, q^k \rangle.$$

Since $b_k \leq y_2 \leq b_{k+1}$ and L_k is the line segment in $\mathbb{R}^2$ with endpoints $[w(b_i), b_i]$, $i = k, k+1$, by Remarks 4.3 and 4.4, this implies that $y \in L_k$. Therefore, $\{y \in Y \mid y_1 + u_k y_2 = \langle d, q^k \rangle\} \subseteq L_k$ and the proof is complete. □

Theorem 4.2. *The algorithm OUTSET is finite and always generates the entire efficient outcome set Y_E of problem* (BX).

Proof. As noted at the beginning of Section 3.1, since Y is a nonempty, compact polyhedron in $\mathbb{R}^2$, Y_E consists of either all of Y, a single extreme point of Y, or one or a set of connected one-dimensional faces (edges) of Y. When $Y_E = Y$, as explained in Section 3.1, Step 1 of the algorithm detects this and the algorithm terminates. Otherwise, from Section 3.1, the algorithm continues by executing next Steps 2 and 3. From the definition of M_1 and Theorem 2.2, $(\langle c_1, x^1 \rangle, \langle c_2, x^1 \rangle)^T = (M_1, m_2)^T \in Y_E$, where

x^1 is calculated in Step 2. If $m_2 = M_2$, Step 3 detects this and, from Theorem 2.2 and the definitions of M_1 and M_2, Y_E consists of the single extreme point $(M_1, M_2)^T$. When this occurs, the algorithm terminates in Step 3 with the indication that $Y_E = \{(M_1, M_2)^T\}$.

When $Y_E \neq Y$ and Y_E consists of more than a single point, the algorithm does not terminate in Step 1 or in Step 3. Instead, from Lemma 4.3 and Theorem 4.1, at the end of each iteration $k \geq 1$ of Steps 4-6, the algorithm detects an additional distinct efficient edge $L_k \subseteq \mathbf{R}^2$ of Y_E. Furthermore, by Theorem 2.2 and Step 6 of the algorithm, every efficient edge of Y_E will be detected if Steps 4-6 are repeated a sufficient number of times. Since Y is polyhedral, it has a finite number of efficient faces [34]. Combining the latter four observations, it follows that when $Y_E \neq Y$ and Y_E is not a singleton, the OUTSET algorithm is finite and generates all of Y_E. □

5. Illustrative problem

To illustrate the workings and benefits of the OUTSET algorithm, we will apply to a sample bicriteria linear programming problem (BX) with $m = 10$ and $n = 20$ which is specified as follows. Let A be given by

$$A = [I_{10} \vdots I_{10}],$$

where I_{10} denotes the 10×10 identity matrix, and let $d \in \mathbf{R}^{10}$ be the vector whose entries are each equal to 1.0. For notational convenience, to describe the 2×20 matrix C, first let $c^j \in \mathbf{R}^2$, $j = 1, 2, 3$, be column vectors defined by

$$c^1 = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}, \quad c_2 = \begin{bmatrix} 0.667 \\ -0.333 \end{bmatrix}, \quad c^2 = \begin{bmatrix} -0.75 \\ 0.25 \end{bmatrix},$$

and let $c^4 \in \mathbf{R}^2$ be the column vector of zeroes. Then C is the 2×20 matrix with columns one through four each equals to c^1, columns five through eight equal to c^2, columns nine and ten equal to c^3 and columns 11 through 20 equal to c^4.

Notice that if we view x_j, $j = 11, 12, \ldots, 20$, as slack variables, then the feasible decision set X of this sample problem (BX) is a hypercube in $\mathbf{R}^{10}$. It is not difficult to show that in this example, this hypercube has 34 efficient extreme points, 68 efficient edges, one two-dimensional efficient face, and two four-dimensional efficient faces. In contrast, the outcome set $Y = CX$ is a compact, two-dimensional polyhedron with only four

efficient extreme points and three efficient edges. What follows is a brief synopsis and discussion of the computations that result from using the OUTSET algorithm to generate the efficient outcome set Y_E in this example. All points in X will be given with slack variables omited.

Step 1. The algorithm solves linear program (T) and finds that $t > 0$. Thus problem (BX) is not completely efficient and the algorithm continues.

Step 2. Two linear programs of the form (1) are solved to yield $M_1 = 2.667$ and $M_2 = 4.500$. Given this value for M_1, the algorithm next solves linear program (Q) and finds the optimal solution x^1 and value m_2 for it given by

$$(x^1)^T = (0,0,0,0,1,1,1,1,0,0)$$

and $m_2 = -1.333$, respectively.

Step 3. Since $m_2 < M_2$, Y_E is not a singleton. The algorithm sets $b_1 = -1.333$, $\alpha_1 = 2.667$ and $Y_E = \phi$ and proceeds to Step 4 with $k = 1$.

Steps 4-6. During this initial iteration of these steps, the algorithm finds a first efficient edge L_1 of Y. To accomplish this, first, with $b = -1.333$ and $\alpha = 2.667$, the algorithm solves linear program $(P_{b,\alpha})$ and finds the optimal solution (u_1, q^1), where $u_1 = 1.0$ and

$$(q^1)^T = (0,0,0,0,0.333,0.333,0.333,0.333,0,0).$$

Next, the linear program in Step 5 is solved with $k = 1$ to yield the optimal solution x^2 given by

$$(x^2)^T = (1,1,1,1,1,1,1,1,0,0).$$

From x^2, the values $b_2 = \langle c_2, x^2 \rangle = 2.667$ and $\alpha_2 = \langle c_1, x^2 \rangle = -1.333$ are calculated. Then, in Step 6, the line segment $L_1 \subseteq \mathbf{R}^2$ is added to Y_E, where the endpoints of L_1 are

$$\begin{bmatrix} 2.667 \\ -1.333 \end{bmatrix}, \quad \begin{bmatrix} -1.333 \\ 2.667 \end{bmatrix}.$$

Notice that as required by Theorem 4.1, L_1 lies on the line $\{y \in \mathbf{R}^2 | y_1 + y_2 = 1.333\}$. Since $b_2 = 2.667 \neq M_2 = 4.500$, k is set equal to 2 and a second iteration of Steps 4-6 will be performed next.

Steps 4-6. With $k = 2$, the second iteration of these steps is executed to find a second efficient edge L_2 of Y. To accomplish this, first, with $b = b_2 = 2.667$ and $\alpha = \alpha_2 = -1.333$, the algorithm finds the optimal solution (u_2, q^2) to the linear program $(P_{b,\alpha})$ where $u_2 = 2.0$ and

$$(q^2)^T = (1,1,1,1,0,0,0,0,0,0,0).$$

Next, the linear program in Step 5 is solved with $k = 2$ to yield the optimal solution x^3 given by

$$(x^3)^T = (1,1,1,1,0,0,0,0,0,0,0).$$

From x^3, the values $b_3 = \langle c_2, x^3 \rangle = 4.0$ and $\alpha_3 = \langle c_1, x^3 \rangle = -4.0$ are calculated. Using these results, the line segment $L_2 \subseteq \mathbf{R}^2$ is then added to Y_E, where the endpoints of L_2 are

$$\begin{bmatrix} -1,333 \\ 2.667 \end{bmatrix}, \quad \begin{bmatrix} -4.00 \\ 4.00 \end{bmatrix}.$$

Notice that the line segment L_2 lies on the line $\{y \in \mathbf{R}^2 \mid y_1 + 2.0y_2 = 4.00\}$, as required by Theorem 4.1. Since $b_3 = 4.0 \neq M_2 = 4.500$, the algorithm sets $k = 3$ and a third iteration of Steps 4-6 will be executed next.

Steps 4-6. With $b = b_3 = 4.0$ and $\alpha = \alpha_3 = -4.0$, the algorithm finds the optimal solution (u_3, q^3) to the linear program $(P_{b,\alpha})$, where $u_3 = 3$ and

$$(q^3)^T = (2,2,2,2,0,0,0,0,0,0,0).$$

With $k = 3$, the linear program in Step 5 is next solved to yield

$$(x^4)^T = (1,1,1,1,0,0,0,0,1,1).$$

Then, the algorithm calculates $b_4 = \langle c_2, x^4 \rangle = 4.50$ and $\alpha_4 = \langle c_1, x^4 \rangle = -5.50$. Using these results, the line segment $L_3 \subseteq \mathbf{R}^2$ with endpoints

$$\begin{bmatrix} -4.00 \\ 4.00 \end{bmatrix}, \quad \begin{bmatrix} -5.50 \\ 4.50 \end{bmatrix}$$

is added to Y_E. Since $b_4 = 4.50 = M_2$, at this point the algorithm terminates with $Y_E = \{L_1, L_2, L_3\}$.

Notice from this example that the computational effort required to execute the steps of OUTSET comes primarily from solving linear programming problems. In this case, to find the entire efficient outcome set, 14 linear programming problems were solved. Since the feasible decision set X in this example contains 34 extreme points, 68 edges, and faces with dimensions as large as four, to generate X_E in its entirety instead of Y_E would be expected to involve significantly more computational effort.

Also notice in this example that the output Y_E consists of three simple, nonredudant line segments in $\mathbf{R}^2$. A decision maker can easily examine and graph this output to gauge the efficient tradeoffs between $\langle c_1, x\rangle$ and $\langle c_2, x\rangle$ that are available in problem (BX). He or she can then potentially pick a most preferred outcome y^* from Y_E and, with the aid of an analyst, recover a most preferred solution $x^* \in X_E$ for which $Cx^* = y^*$.

In contrast, since X_E in this example contains 34 extreme points, 68 edges, and faces of dimension as large as four, solving the example by a traditional decision-based method that generates the entire efficient decision set X_E would result in as much larger, more complicated output. Many of the points in X_E would map into the same points in Y_E. Furthermore, presenting X_E to the DM would be more difficult, and the DM would not be expected to be able to choose a most preferred decision x^* directly from X_E as easily as he or she could by first examining Y_E instead.

From Theorem 2.2, to generate Y_E, an alternate approach in this example is to parametrically solve the linear program (P_b) by the parametric simplex method for all $b \in [m_2, M_2] = [-1.333, 4.500]$ (cf. [25]). This approach, however, is expected to be less efficient computationally than the approach used in OUTSET, because, in general, it necessitates finding many more optimal bases for problem (P_b) than there are efficient faces for Y_E. For instance, in this example, although Y_E contains only three efficient edges, solving (P_b) parametrically necessitates calculating 10 different optimal bases for (P_b), $b \in I$.

6. Concluding remarks

Notice that outset does not involve complicated bookkeeping or backtracking. Also, notice that the OUTSET algorithm will always generate all of Y_E by solving at most $(4+2E)$ linear programming problems, where E is the number of efficient edges in Y. In addition, as Steps 4-6 are repeated during the execution of OUTSET, computational savings can be created by using the optimal solutions (u_k, q^k) and x^{k+1} found in iteration

k to the linear programs in Steps 4 and 5 as starting points for solving these linear programs in the next iteration $k+1$ [25]. Furthermore, various powerful algorithms and computer codes now exist for efficiently solving large-scale linear programming problems [3]. These observations lead us to expect that the new OUTSET algorithm will have the potential to solve large bicriteria linear programming problems that heretofore, because of their size, were not amenable to decision set-based multiple objective linear programming methods. Combined with the fact that, in practice, a DM can more easily comprehend and work with the efficient outcome set Y_E than the efficient decision set X_E of problem (BX), this indicates that the new OUTSET algorithm offers significant promise for both analysts and decision makers involved in multiple objective decision making.

REFERENCES

1. P. Armand and C. Malivert, *Determination of the efficient decision set in multiobjective linear programming*, J. Optimization Theory and Appl. **70** (1991), 467-489.
2. R. Armann, *Solving multiobjective programming problems by discrete representation*, Optimization **20** (1989), 483-492.
3. M. S. Bazaraa, J. J. Javis and H. D. Sherali, *Linear Programming and Network Flows*, 2nd Edition John Wiley and Sons, New York, 1990.
4. H. P. Benson, *Complete efficiency and the initialization of algorithms for multiple objective programming*, Operations Research Letters **10** (1991), 481-487.
5. H. P. Benson, *A geometrical analysis of the efficient outcome set in multiple-objective convex programs with linear criterion functions*, J. of Global Optimization **6** (1995), 231-251.
6. H. P. Benson, *Vector maximization with two objective functions*, J. of Optimization Theory and Applications **28** (1979), 253-257.
7. H. P. Benson and Y. Aksoy, *Using efficient feasible direction in interactive multiple objective linear programming*, Operations Research Letters **10** (1991), 203-209.
8. H. P. Benson and S. Sayin, *Towards finding global representations of the efficient set in multiple objective mathematical programming*, Naval Research Logistics (to appear).
9. G. R. Bitran, *Theory and algorithms for linear multiple objective programs with zero-one variables*, Mathematical Programming **17** (1979), 362-390.
10. J. L. Cohon, *Multiobjective Programming and Planning*, Academic Press, New York, 1978.
11. J. P. Dauer, *Analysis of the objective space in multiple objective linear programming*, J. of Mathematical Analysis and Appl. **126** (1987), 579-593.
12. J. P. Dauer, *On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program*, Annals of Operations Research **47** (1993), 279-292.
13. J. P. Dauer and Y. H. Liu, *Solving multiple objective linear programs in objective space*, European J. of Operational Research **46** (1990), 350-357.

14. J. P. Dauer and O. A. Saleh, *Constructing the set of efficient objective values in multiple objective linear programs*, European J. of Operational Research **46** (1990), 358-365.

15. J. G. Ecker, N. S. Hegner and I. A. Kouada, *Generating all maximal efficient faces for multiple objective linear programs*, J. of Optimization Theory and Applications **30** (1980), 353-381.

16. G. W. Evans, *An overview of techniques for solving multiobjective mathematical programs*, Management Science **30** (1984), 1268-1282.

17. J. P. Evans and R. E. Steuer, *Generating Efficient Extreme Points in Linear Multiple Objective Programming: Two Algorithms and Computing Experience*, Multiple Criteria Decision Making, Edited by J. L. Cochrane and M. Zeleny, University of South Carolina Press, Columbia, South Carolina, (1973), 349-365.

18. H. J. Gallagher and O. A. Saleh, *A representation of on efficiency equivalent polyhedron for the objective set of a multiple objective linear program*, European J. of Operational Research **80** (1995), 204-212.

19. A. Goicoechea, D. R. Hansen and L. Duckstein, *Multiobjective Decision Analysis with Engineering and Business Applications*, John Wiley and Sons, New York 1982.

20. H. Isermann, *The enumeration of the set of all efficient solutions for a linear multiple objective program*, Operational Research Quarterly **28** (1977), 711-725.

21. H. W. Kuhn and A. W. Tucker, *Nonlinear Programming*, Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, Edited by J. Neyman, University of California Press, Berkeley, California (1950), 481-492.

22. D. T. Luc, *Theory of Vector Optimization*, Springer Verlag, Berlin, Germany, 1989.

23. O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill Book Company, New York, 1969.

24. O. Marcotte and R. M. Soland, *An interactive branch-and-bound algorithm for multiple criteria optimization*, Management Science **32** (1986), 61-75.

25. K. G. Murty, *Linear Programming*, John Wiley and Sons, New York, 1983.

26. J. Philip, *Algorithms for the vector maximization problem*, Mathematical Programming **2** (1972), 207-279.

27. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.

28. Y. Sawaragi, H. Nakayama and T. Tanino, *Theory of Multiobjective Optimization*, Academic Press, Orlando, Florida, 1985.

29. W. S. Shin and A. Raviudran, *Interactive multiple objective optimization: Survey I-continuous case*, Computers and Operations Research **18** (1991), 97-114.

30. R. M. Soland, *Multicriteria optimization: A general characterization of efficient solutions*, Decision Sciences **10** (1979), 26-38.

31. R. E. Steuer, *A Five Phase Procedure for Implementing a Vector-Maximum Algorithm for Multiple Objective Linear Programming Problems*, Multiple Criteria Decision Making: Jouy-en-Josas, Edited by H. Thiriez and S. Zionts, Springer Verlag, New York (1975), 159-168.

32. R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley and Sons, New York, 1986.

33. B. Villarreal and M. H. Karwan, *Multicriteria integer programming: A (hybrid) dynamic programming recusive approach*, Mathematical Programming **21** (1981), 204-223.

34. P. L. Yu, *Multiple Criteria Decision Making*, Plenum, New York, 1985.

35. P. L. Yu and M. Zeleny, *The set of all nondominated solutions in linear cases and a multicriteria simplex method*, J. of Mathematical Analysis and Appl. **49** (1975), 430-468.

36. M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.

DEPARTMENT OF DECISION AND INFORMATION SCIENCES,
351 BUSINESS BUILDING, UNIVERSITY OF FLORIDA,
GAINESVILLE, FL 32611, USA