Mathematical Programming 5 (1973) 54–72. North-Holland Publishing Company

A REVISED SIMPLEX METHOD FOR LINEAR MULTIPLE OBJECTIVE PROGRAMS *

J.P. EVANS

University of North Carolina, Chapel Hill, North Carolina, U.S.A.

R.E. STEUER

University of Kentucky, Lexington, Kentucky, U.S.A.

Received 11 October 1972
Revised manuscript received 16 March 1973

For linear multiple-objective problems, a necessary and sufficient condition for a point to be efficient is employed in the development of a revised simplex algorithm for the enumeration of the set of efficient extreme points. Five options within this algorithm were tested on a variety of problems. Results of these tests provide indications for effective use of the algorithm.

1. Introduction

Let A and C be $m \times n$ and $k \times n$ matrices, respectively, and let $x \in \mathbf{R}^n$, $b \in \mathbf{R}^m$. Define[1] $S = \{x: Ax = b, x \geqq 0\}$.

Definition 1.1. A point $x^0 \in \mathbf{R}^n$ is an *efficient point of S* if
(i) $x^0 \in S$;
(ii) there is no $x \in S \ni Cx \geq Cx^0$.

Define the set $E = \{x \in \mathbf{R}^n\colon x \text{ is an efficient point of } S\}$. The linear multiple-objective program is to determine E, i.e., the set of all efficient points for given C, A and b. This problem and its generalization to non-

* This work was partially supported by the Office of Naval Research, Contract No. N00014-67-A-0321-0003 (NR047-095).

[1] Throughout this paper we use the following convention for vector inequalities: $x \geq 0$ if $x_j \geqq 0$, $j = 1, ..., n$, $x \neq 0$; and $x \geqq 0$ if $x_j \geqq 0, j = 1, ..., n$.

linear criteria and constraints have been studied by many authors [1–7, 9, 10]. Our goal in this paper is the application of a necessary and sufficient condition for the efficiency of a feasible point (in the linear problem) to the development of an algorithm for the computation of the points in E. We will point out relationships to previous research throughout the development. In particular, we provide computational experience on a point raised by Philip (see [10, p. 222]).

First we present a lemma which characterizes efficient points in a form that is particularly useful for our purposes. Even though this result may not previously have been stated in the specific manner selected herein, it is certainly available as a consequence of early work on linear multiple-objective programs.[2]

Lemma 1.2. *Let $x^0 \in S$ and let D be an $n \times n$ diagonal matrix with*

$$d_{jj} = \begin{cases} 1 & \text{if } x_j^0 = 0\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Then $x^0 \in E$ if and only if the system

$$Cu \geq 0,\quad Du \geqq 0\,,\quad Au = 0\,, \tag{1.1}$$

has no solution $u \in \mathbf{R}^n$.

This lemma has three corollaries whose role will be central in the subsequent development.

Corollary 1.3. *Let $x^0 \in S$ and D be as defined above. Then x^0 is efficient if and only if there exist $p \in \mathbf{R}^k$, $y \in \mathbf{R}^n$, $w \in \mathbf{R}^m$ such that*

$$C^{\mathrm{T}} p + D^{\mathrm{T}} y + A^{\mathrm{T}} w = 0\,,\quad p > 0\,,\quad y \geqq 0\,. \tag{1.2}$$

Proof. Tucker's theorem of the alternative (see [8]) states that either (1.1) has a solution or (1.2) has a solution, but never both.

Corollary 1.4. *A point $x^0 \in S$ is efficient if and only if there is a $p \in \mathbf{R}^k$, $p > 0$ such that*[3] $(p^{\mathrm{T}} C)x^0 = \max\{(p^{\mathrm{T}} C)x \colon x \in S\}$.

[2] See for example [2, 10].
[3] Note that $\max\{p^{\mathrm{T}} Cx \colon x \in S\}$ is an ordinary linear program for which the objective function is obtained by assigning some fixed positive weight to each of the k original criteria and summing to obtain a single criterion. We shall call this problem (L_p).

Proof. Can be proved by application of Corollary 1.3.

Note that Corollary 1.4 implies:

Corollary 1.5. *If S has an efficient point, at least one extreme point of S is efficient.*

In at least two previous developments on multiple-objective problems, the idea of assigning positive weight to each objective and solving the resulting single-objective program has been used to obtain a first efficient point (see [2, 3]). Subsequent efficient points were then obtained by parametric variation of the weighting vector. This parametrization takes a simple form in the case of two criteria [3]. A typical choice for the initial weighting is $p_i = k^{-1}$, $i = 1, ..., k$; however, this has the feature that one or more efficient points may be by-passed before the algorithm detects one.

It is Corollary 1.5 that provides the primary motivation for the current work. Specifically suppose computation of the efficient extreme points is approached by first choosing arbitrary positive weights for the criteria and then maximizing the weighted objective function to obtain the first efficient extreme point. Then in the sequence of basic feasible solutions generated prior to the optimum point there may be one or more efficient points. However, this would not be detected unless a test for efficiency is applied to each basic feasible solution. As a consequence, the computation of all efficient extreme points could well require substantial back-tracking; in problems with large bases this can amount to a substantial amount of computation time.

In Section 2, a test for the efficiency of an arbitrary extreme point is developed. In Section 3, we provide a classification of multiple-objective programs. Section 4 outlines and justifies our algorithm and in Section 5 we present the results of extensive numerical tests of several possible ways to implement computation for multiple-objective programs.

2. Testing for efficiency at each simplex iteration

Corollary 1.4 provides the primary tool for the efficiency test we seek. Namely, if x^0 is a basic feasible solution (b.f.s.), it is efficient if and only if x^0 solves (L_p) for some positive k-vector p. In this section, a computationally useful form of this result will be first developed under the assumption that each basic feasible solution is nondegenerate. Modification of this result for a degenerate b.f.s. is treated at the end of this section.

We will use the following common notation for linear problems. Suppose x is a b.f.s. to the constraints with associated basis B. By renumbering variables if necessary and partitioning A and C, we have

$$x_B = B^{-1}b - B^{-1}Nx_N , \quad z = C_B B^{-1}b - (C_B B^{-1}N - C_N)x_N , \tag{2.1}$$

where the subscript B denotes basic and the subscript N denotes nonbasic. Observe that C_B is $k \times m$, C_N is $k \times (n - m)$, and z is the k-vector of criterion values associated with the b.f.s. x.

Lemma 2.1. *Let x^0 be a nondegenerate b.f.s. in S with corresponding basis B. Then x^0 is efficient if and only if*

$$-(C_B B^{-1}N - C_N)\, u_N \geq 0 , \quad u_N \geqq 0 \tag{2.2}$$

is inconsistent, where u_N is an $(n - m)$ vector.

Proof. Follows from (1.1).

Corollary 2.2. *Define $R = C_B B^{-1} N - C_N$ and consider the following auxiliary problem:*

(P) $\quad \max\{e^T v:\ Ru + Iv = 0 , \ u \geqq 0 , \ v \geqq 0\}$,

where $e^T = (1, ..., 1)$, $u \in \mathbf{R}^{n-m}$, $v \in \mathbf{R}^k$. If (2.2) *is consistent, problem* (P) *is consistent unbounded. If* (2.1) *is inconsistent, problem* (P) *is consistent, bounded, with optimal objective value equal to zero.*

Proof. Straightforward.

The latter result provides a computational test of the efficiency of an extreme point without requiring identification of the weighting vector (see Corollary 1.4) for which the point is optimal. Note that the subproblem (P) involves one constraint for each criterion, thus one would generally expect (P) to have a small basis. Moreover, a convenient initial feasible basis for (P) is available, namely the variables, $v_1, ..., v_k$. In addition, we shall see below that this same sub-problem plays a central role in the process of enumerating the efficient extreme points once the first one has been identified. The sub-problem employed herein is a variation of a formulation suggested by Philip [10] for a similar purpose.

Now we turn to the problem of enumerating the efficient extreme points once the first efficient point has been obtained. In the approach suggested by Charnes and Cooper [2] the original weighting vector was changed parametrically to test whether an adjacent extreme point is an alternative optimum to the first efficient point computed. Our approach is motivated by the sub-problem (P) and is designed to detect adjacent efficient points without explicitly displaying a weighting vector p for which two adjacent extreme points are alternative optima.

The parametric approach of Charnes and Cooper is based on the following result.

Lemma 2.3. *Let x^0 be an efficient extreme point of S and let x_j be a non-basic variable in the basic feasible solution associated with x^0. Then the adjacent extreme point with x_j a basic variable (with some currently basic variable converted to non-basic status) is efficient if and only if the following version of problem* (P) *is consistent and bounded:*

$$(\mathrm{P}^j) \qquad \max\{e^{\mathrm{T}} v\colon\ Ru - r_j w + Iv = 0\,,\quad u \geqq 0\,,\quad v \geqq 0\}\,,$$

where $w \in \mathbf{R}$ and r_j is the column of R associated with x_j.

Proof. Can be proved by the dual theorem of linear programming.

The impact of Lemma 2.3 is that, regardless of the number of criteria, an extreme point adjacent to a given efficient extreme point x^0 can be identified as efficient or non-efficient by a straightforward form of post-optimality analysis applied to the sub-problem (P) associated with x^0. Specifically we simply add to the original data for sub-problem (P) the negative of the column r_j. If this modification results in a consistent bounded sub-problem, the corresponding adjacent point is also efficient. Otherwise it is not. This test can be conducted for each non-basic x_j and thus each adjacent extreme point can be classified as efficient or rejected as non-efficient, from a consideration of data in sub-problem (P). This test will form a crucial part of the algorithm to be described subsequently.

In the case of a degenerate basic feasible solution, a slight modification of Lemma 2.1 and the sub-problem in Corollary 2.2 is required in order to retain the sharpness of the efficiency test. This change is provided in the following result.

Lemma 2.4. *Let x^0 be a b.f.s. in S with corresponding basis B. Let $Q = \{i: x^0_{B_i} = 0\}$. Then x^0 is efficient if and only if*

$$(C_B B^{-1} N - C_N) u_N \geq 0, \quad Y_i u_N \geqq 0, \quad i \in Q, \quad u_N \geqq 0 \tag{2.3}$$

is inconsistent, where $Y = -B^{-1} N$ and Y_i is the i^{th} row of Y.

The proof is a straightforward application of Lemma 1.2 and will be omitted. The new form of the auxiliary program (P) is then

$$(\mathrm{P}') \qquad \max\{e^{\mathrm{T}} v\colon\ Ru + Iv = 0,\ Y_i u - s_i = 0, \quad i \in Q,$$
$$u \geqq 0, \quad v \geqq 0, \quad s \geqq 0\} .$$

Corollary 2.5.
(2.3) *inconsistent* $\Leftrightarrow$ (P′) *consistent and bounded;*
(2.3) *consistent* $\Leftrightarrow$ (P′) *consistent and unbounded;*

3. Classification of multiple-objective programs

In this section, we wish to indicate a classification of linear multiple-objective problems which is a natural extension of the corresponding description of ordinary linear programs as either inconsistent, consistent-bounded, or consistent-unbounded.[4] For this purpose we need the following definition.

Definition 3.1. Let $x^0 \in S$ and $d \in \mathbf{R}^n$, $d \neq 0$. Then the direction d is an *efficient direction at* x^0 if there is an $\bar{\alpha} > 0$ such that $x_\alpha = x^0 + \alpha d$ is efficient for each $\alpha \in [0, \bar{\alpha}]$. If x_α is efficient for each $\alpha \geqq 0$, then S is said to have an *unbounded efficient path* (with respect to C).

[4] The bounded–unbounded terminology refers to objective function values and not to the feasible set S.

Theorem 3.2. *Let C, A, b be as defined in Section* 1, *with $S = \{x \in \mathbf{R}^n : Ax = b, x \geqq 0\}$. Then one and only one of the following holds:*

(i) $S = \emptyset$;

(ii) $S \neq \emptyset, E = \emptyset$, *each criterion unbounded*;

(iii) $S \neq \emptyset, E = \emptyset$, *at least one criterion bounded*;

(iv) $S \neq \emptyset$, $E \neq \emptyset$, *and S contains one or more unbounded efficient paths*;

(v) $S \neq \emptyset, E \neq \emptyset$ *but S contains no unbounded efficient paths.*

Proof. The five cases are clearly mutually exclusive. Since case (i) can obviously occur, it remains to exhibit an example of each of the other cases.

Case (ii). $C = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right], A = (1, -1), b = 0$.
Case (iii). $C = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right], A = (0, \ \ 1), b = 1$.
Case (iv). $C = \left[\begin{smallmatrix}2 & 0\\ -1 & 1\end{smallmatrix}\right], A = (0, \ \ 1), b = 1$.
Case (v). $C = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right], A = (1, \ \ 1), b = 1$.

Corollary 3.3. *If S has an unbounded efficient path, then S has an efficient extreme point.*

Proof. Since each point on an unbounded efficient path is efficient by definition, the conclusion follows from Corollary 1.5.

Next we indicate the manner in which each of cases (ii)–(v) can be detected in the process of finding and enumerating efficient extreme points.

Lemma 3.4. *Let $x^0 = (x_B^0, x_N^0)$ be a b.f.s. in S. Let x_j be a non-basic variable and r_j, y_j be the corresponding columns of R and $Y = B^{-1} N$, respectively. If $r_j = C_B\, y_j - C_N < 0$ and $y_j \leqq 0$, then all criteria are unbounded, i.e., Case* (ii) *of Theorem* 3.2 *obtains.*

Proof. Obvious.

Lemma 3.5. *Let x^0, x_j be as in Lemma* 3.4. *If $r_j \leq 0, y_j \leqq 0$, and* (*say*) *the i^{th} row of $R = C_B\, B^{-1}\, N - C_N$ is non-negative, then $E = \emptyset$ and at least one criterion* (*the i^{th}*) *is bounded; Case* (iii) *obtains.*

Proof. Straightforward.

Lemma 3.6. *Let x^0, x_j be as in Lemma* 3.4. *Suppose the sub-problem associated with x^0 is bounded; assume also that making x_j positive identifies an efficient path from x^0. Then if $y_j \leq 0$, Case* (iv) *of Theorem* 3.2 *holds.*

Proof. Straightforward.

The classification provided in Theorem 3.2 helps to highlight an aspect of multiple-objective programs which is relevant in the selection of a computational procedure. Consider again the example of Case (iv) above. Suppose one approaches this problem by assigning equal weight to the two objectives. That is, one attempts to find an efficient point by solving the linear program

$$\max\{0.5\, x_1 + 0.5\, x_2 \colon\; x_2 = 1,\;\; x_1 \geqq 0,\;\; x_2 \geqq 0\}\;.$$

Observe that the *equal weight* objective function of the linear problem is unbounded. However, note that this example was used to illustrate Case (iv) of Theorem 3.3. The problem has an efficient point at $(x_1, x_2) = (0, 1)$, but this cannot be detected by assigning equal weight to each criterion. Moreover, this same procedure would not detect the unbounded efficient path in this example.

In connection with Theorem 3.2, we can state a result in the spirit of a duality theorem. Let us define a family of problems

$$(\mathrm{D}_p) \qquad \min\{b^{\mathrm{T}} w \colon A^{\mathrm{T}} w \geqq p^{\mathrm{T}} C\}\;,$$

where $w \in \mathbf{R}^k$ and $p \in P = \{p \in \mathbf{R}^k \colon\; p > 0\}$,

Theorem 3.7. *Let C, A, b be as defined in Section* 1. *Then:*

Case (i) $\Rightarrow$ *Either* (D_p) *is inconsistent for each $p \in P$, or for some $p \in P$ (D_p) is consistent unbounded.*

Case (ii) *or* (iii) $\Rightarrow$ (D_p) *is inconsistent for each $p \in P$.*

Case (iv) $\Rightarrow$ (D_p) *is inconsistent for at least one $p \in P$, and consistent, bounded for at least one $p \in P$.*

Case (v) $\Rightarrow$ (D_p) *is consistent and bounded for each $p \in P$.*

Proof. Follows from application of the dual theorem of linear programming.

4. Algorithm for computation of efficient extreme points

In describing the algorithm that has been developed, it will be assumed that the vector x contains any artificial variables which were added to aid formation of an initial basis. The general outline of the algorithm is as follows:

Phase 1. Starting from an initial basis which contains vectors associated with artificial variables, proceed to a feasible basis if one exists or terminate if the problem is inconsistent.

Phase 2. From a feasible basis proceed to an efficient basis if one exists or detect one of the other cases of Theorem 3.2. The sub-problem (P) is used as described in Corollary 2.2 to test for efficiency of a b.f.s.

Phase 3. From an efficient basis proceed to enumerate the list of efficient basic feasible solutions. The sub-problem (P) is used as described in Lemma 2.3 to test for efficiency of an adjacent b.f.s.

A number of computational options are available in Phase 2 and will be discussed later in this section.

For implementation of the algorithm in a revised simplex mode it is necessary to store the matrices A and C, the right-hand side vector b, and the artificial (Phase 1) objective function coefficient vector $\hat{c}$. For a general iteration in Phase 1, the revised simplex basis matrix will be

$$\bar{B} = \begin{bmatrix} B & 0 & 0 \\ -C_{\mathrm{B}} & I_k & 0 \\ -\hat{c}_{\mathrm{B}} & 0 & 1 \end{bmatrix}_{m+k+1}$$

where C_{B} and $\hat{c}_{\mathrm{B}}$ have the meaning used in Section 2. The revised simplex basis inverse is then

$$\bar{B}^{-1} = \begin{bmatrix} B^{-1} & 0 & 0 \\ C_{\mathrm{B}}\,B^{-1} & I_k & 0 \\ \hat{c}_{\mathrm{B}}\,B^{-1} & 0 & 1 \end{bmatrix}.$$

At the transition from Phase 1 to Phase 2 the $(m + k + 1)^{\text{st}}$ row of $\bar{B}$ is modified in the usual way to keep the artificial objective function value equal to zero. This row is retained because it performs a useful function in Phase 3.

It is readily seen that if a_j is a non-basic vector, the corresponding column of the matrix R used in sub-problem (P) is obtained by the computation

$$[C_B B^{-1} \;\; I_k \;\; 0] \; [a_j \;\; -c_j \;\; 0]^T = C_B B^{-1} a_j - c_j \,.$$

Computations in sub-problem (P) are also executed by a revised simplex algorithm. It is readily seen by reference to Corollary 2.2 that the variables υ provide a convenient initial feasible sub-problem basis with associated matrix $\begin{bmatrix} I_k & 0 \\ -e^T & 1 \end{bmatrix}$ which has inverse $\begin{bmatrix} I_k & 0 \\ e^T & 1 \end{bmatrix}$ in which the $(k+1)^{\text{st}}$ row is for the sub-problem objective function. If at some Phase 2 iteration the current b.f.s. is degenerate, the sub-problem can be expanded to the form (P′) if desired. The original sub-problem (P) can be used and if it has a bounded objective value, the associated b.f.s. is an efficient extreme point.

During Phase 2, each time the sub-problem test for efficiency is not met, a non-basic variable must be chosen to become basic. It is possible to employ a scheme in making this selection which insures that Phase 2 of the algorithm will terminate finitely in one of the cases listed in Theorem 3.2. [5] However, the example which follows Lemma 3.6 shows that choosing the new basic variable to maximize some weighted sum of the objectives may fail to detect one of the cases which can occur. The investigator using the algorithm must decide whether the computational cost of using a fail-safe strategy is justified. We turn now to a discussion of such a strategy for which computational experience is reported in Section 4.

The key to the strategy referred to above comes directly from the classification provided in Theorem 3.2. The procedure developed below, which we call *sequential maximization*, constructs a sequence of nested subsets of the feasible region which permits us to distinguish the possible cases. We shall need the following preliminary results.

Lemma 4.1. *Suppose $E \neq \emptyset$, and let $x^0 \in S \setminus E$. Define $T = \{x \in S\colon Cx \geqq Cx^0\}$ and let E_0 be the set of efficient points of T for criterion matrix C. Then there is $\bar{x} \in E_0 \ni C\bar{x} \geq Cx^0$.*

Proof. We note that $x^0 \in T$. Suppose $E_0 = \emptyset$. Then for each k-vector $\lambda > 0$, the problem $\max\{(\lambda^T C)x\colon x \in T\}$ has no solution, i.e., is unbounded. But since $T \subseteq S$, it follows that $(\lambda^T C)x$ is unbounded over S,

[5] In [10], a pivot selection rule is suggested which reduces but does not eliminate the risk of cycling. The scheme described herein maximizes one of k criteria at a time, holding previously treated criteria at fixed levels.

which by Corollary 1.4 implies $E = \emptyset$, a contradiction. Thus $E_0 \neq \emptyset$ and $E_0 \subseteq T$ yields the desired conclusion.

Lemma 4.2. *Let x_0, T, E_0 be as in Lemma* 4.1. *Then $E_0 \subseteq E$.*

Proof. Let $\bar{x} \in E_0$; suppose $\bar{x} \notin E$. But then there is $\tilde{x} \in S \ni C\tilde{x} \geq C\bar{x} \geqq Cx_0$. This contradicts $\bar{x} \in E_0$, hence $E_0 \subseteq E$.

Lemma 4.3. *Let $x^0 \in S \setminus E$, $\bar{x} \in E$. If $C\bar{x} \geqq Cx^0$, then $C\bar{x} \geq Cx^0$.*

Proof. Assume $C\bar{x} = Cx^0$. Since $\bar{x} \in E$, there is a positive k-vector λ such that $\lambda^{\mathrm{T}} C\bar{x} = \max\{\lambda^{\mathrm{T}} Cx : x \in S\}$. But then $\lambda^{\mathrm{T}} C\bar{x} = \lambda^{\mathrm{T}} Cx^0$, which contradicts $x^0 \in S \setminus E$.

The preceding results yield the conclusion that if $E \neq \emptyset$, then each inefficient point is dominated by an efficient point. In the remainder of this discussion, we will employ the following recursively defined sets:

$$S_0 = S, \quad S_l = \{y : c_{j_l} y = \max\{c_{j_l} x : x \in S_{l-1}\}\}, \quad l = 1, \ldots, k,$$

in which c_j is the j^{th} row of the matrix C.[6] We will now employ this process of sequential maximization either to yield an efficient extreme point if one exists, or terminate with the indication that $E = \emptyset$. Note that $S_0 \supset S_1 \supset \ldots \supset S_k$.

Lemma 4.4. *Suppose $E \neq \emptyset$. If $S_l \neq \emptyset$ for some l, $1 \leqq l \leqq k$, then $S_l \cap E \neq \emptyset$.*

Proof. Let $x^0 \in S_l$ for some l, $1 \leqq l \leqq k$. If $x^0 \in E$, we are done. Hence suppose $x^0 \notin E$. Then by Lemma 4.3 there is a point $\bar{x} \in E$ such that $C\bar{x} \geq Cx^0$. Hence

$$c_{j_q} \bar{x} \geqq c_{j_q} x^0, \qquad q = 1, \ldots, l,$$

which implies $\bar{x} \in S_l$ and hence $S_l \cap E \neq \emptyset$.

[6] To construct the set $S_0, S_1, \ldots, S_k$, we first identify a criterion $c_j x$ which is bounded over S_0. Then we maximize a criterion $c_i x$ holding the first criterion at its maximum value. The process continues until S_k is obtained or for some value l, $S_l \neq \emptyset$, but each remaining criterion is unbounded over S_l. In the latter case $S_{l+1} = \ldots = S_k = \emptyset$.

From Lemma 4.4 we see that if there is at least one efficient point, *each nonempty set* S_l, $0 \leqq l \leqq k$, contains an efficient point.

Suppose without loss of generality that the sequential maximization process employed $c_1, ..., c_{l-1}$ to define the sets $S_1, ..., S_l - 1$, respectively. In the result below we will examine the expression $\Sigma_{i=l}^{k} p_i c_i x$ on S_{l-1}.

Lemma 4.5. *Suppose* $S_0, S_1, ..., S_{l-1}$ *are non-empty, but* $S_l = \emptyset$.

(i) *If* $\Sigma_{i=l}^{k} p_i c_i x$ *is bounded on* S_{l-1} *for some set of positive weights* $p_l, ..., p_k$, *then* $E \neq \emptyset$.

(ii) *If* $\Sigma_{i=l}^{k} p_i c_i x$ *is unbounded on* S_{l-1} *for each set of positive weights* $p_l, ..., p_k$, *then* $E = \emptyset$.

Proof. (i) Suppose for some fixed positive scalars, $p_l, ..., p_k$, the expression $\Sigma_{i=l}^{k} p_i c_i x$ is bounded on S_{l-1} with the maximum value occurring at $\bar{x} \in S_{l-1}$. Assume $\bar{x} \notin E$; then there is an $x^0 \in S_0$ such that $C x^0 \geq C \bar{x}$. Hence $x^0 \in S_{l-1}$, and moreover

$$\sum_{i=l}^{k} p_i c_i x^0 > \sum_{i=l}^{k} p_i c_i \bar{x},$$

which is a contradiction. Thus $\bar{x} \in E$.

(ii) Now suppose that $\Sigma_{i=l}^{k} p_i c_i x$ is unbounded on S_{l-1} for each choice of positive weights $p_l, ..., p_k$. Assume $\bar{x} \in S_{l-1} \cap E$; then by Corollary 1.4 there exists $p \in \mathbf{R}^k, p > 0$, such that $\bar{x}$ solves $\max\{(p^{\mathrm{T}} C) x: x \in S_0\}$. Since $S_{l-1} \subset S_0$, this is a contradiction. Thus $E = \emptyset$.

In the context of Lemma 4.5, let x be a basic feasible solution in S_{l-1}, and consider the following associated subproblem, a variation of (P) which reflects the additional constraints required to define S_{l-1}:

$$(\mathrm{P}_{l-1}) \qquad \max\{e^{\mathrm{T}} v:\ (B^{-1}N) u \leqq 0, \quad R u + I v = 0, \quad u, v \geq 0\}$$

in which the basis B contains $m + l - 1$ rows and R is the reduced cost matrix for the remaining $k - l + 1$ unmaximized criteria.

Corollary 4.6. *Assume* $S_0, ..., S_{l-1}$ *non-empty, but* $S_l = \emptyset$ *and let* x *be a basic feasible solution in* S_{l-1}. *Then* $E \neq \emptyset \Leftrightarrow$ *subproblem* (P_{l-1}) *is consistent, bounded.*

Lemma 4.7. *If* $S_k \neq \emptyset$, *then* $S_k \subseteq E$.

Proof. Assume $x^0 \in S_k/E$. Then there is an $\bar{x} \in E \ni C\bar{x} \geq Cx^0$, by Lemmas 4.1–4.3. But then $\bar{x} \in S_k$, hence $c_j\bar{x} = c_j x^0, j = 1, ..., k$, which contradicts $C\bar{x} \geq Cx^0$. Thus we conclude $S_k \subseteq E$.

To summarize these results, we have the following list of termination possibilities for the sequential maximization process:

(a) If $S_k \neq \emptyset$, then $E \neq \emptyset$, and furthermore each point in S_k is efficient.

(b) If $S_l = \emptyset$ for some l, $1 \leqq l \leqq k$, and subproblem (P_{l-1}) is consistent, bounded, then $E \neq \emptyset$. In addition, any optimal solution to the dual of (P_{l-1}) yields a set of weights which can be used to identify an efficient point in S_{l-1}.

(c) If $S_l = \emptyset$ for some l, $1 \leqq l \leqq k$, but the associated subproblem (P_{l-1}) has unbounded objective value, then $E = \emptyset$.

The process employed in distinguishing among these possibilities requires at most a finite number of simplex pivots.

5. Computational experience

The sequential maximization procedure described in the preceding section is guaranteed to terminate at an efficient point if one exists. However, since the sub-problem can be used to test for the efficiency of any extreme point, a number of computational variations is available for Phase 2. These options together with computational experience obtained to date are described below.

Option 1. Assign equal weights to each of the k criteria and maximize the resulting scalar objective. Employ no subproblem testing for efficiency.[7]

Option 2. Maximize equally weighted criteria with sub-problem testing for efficiency after each pivot.

Option 3. Employ the sequential maximization described in the preceding section with no sub-problem testing.

Option 4. Employ sequential maximization with sub-problem testing

[7] Recall that this procedure is not fail-safe in that it may fail to find an efficient point when one exists.

only at the completion of each new maximization.

Option 5. Employ sequential maximization with sub-problem testing after each pivot.

Each of these options was investigated on a moderate number of randomly generated problems for the purpose of providing guidelines for the identification of methods which are most effective in Phase 2, obtaining the first efficient point. In these computational results, the right-hand side vector b is an m-vector with each component set to 100. Zero density in the A-matrix was explicitly controlled; each non-zero element of A was an observation from the uniform distribution over the integers 1 to 20. The C-matrices were also randomly generated. In runs 1–18, each element of C was an observation from the uniform distribution over the integers 0–20. Sample sizes are indicated in Tables 1–6.

Each sub-problem call employed version (P′) of the sub-problem so as to retain the sharpness of the Lemma 2.4 in identifying efficient points.[8]

The data tabulated are identified by the following key:

T = mean time per problem in tenths of seconds;
MP = mean number of master problem pivots;
MS = mean number of sub-problem pivots;
σ_T = standard deviation of computation time.
σ_M = standard deviation of the number of master problem pivots;
σ_S = standard deviation of the number of sub-rpoblem pivots;

All computation was performed on an IBM 370/165.

In the following tables, the size designation is $k \times m \times n$, where k is the number of objectives, m the number of constraints, and n the number of variables exclusive of slacks.

It is apparent from the data that Options 2 and 5, which call the sub-problem after each master problem pivot, are uniformly poor as measured by the time required to obtain the first efficient extreme point. These results clearly suggest that such frequent calls of the sub-problem involve a very significant setup cost which is not sufficiently offset by reductions in sub-problem pivots to yield net time savings.

[8] In [10], the efficiency test proposed in method II.2, p. 218, corresponds to the use of sub-problem (P) which may fail to detect the efficiency of a degenerate b.f.s.

Table 1
Size 3 × 10 × 20; sample size 50

Option	1	2	3	4	5
Run 1; Zero density 0%					
T	2.32	5.16	3.54	2.38	4.56
MP	9.64	7.26	11.14	7.34	6.72
MS	0.0	7.60	0.0	2.02	7.10
σ_T	0.13	0.32	0.15	0.11	0.23
σ_M	0.50	0.45	0.53	0.39	0.35
σ_S	0.0	0.78	0.0	0.10	0.81
Run 2; Zero density 50%					
T	2.56	6.42	3.50	2.94	5.48
MP	11.14	9.56	11.78	8.90	8.34
MS	0.0	5.10	0.0	1.66	3.46
σ_T	0.13	0.31	0.14	0.13	0.31
σ_M	0.53	0.46	0.56	0.49	0.43
σ_S	0.0	0.58	0.0	0.09	0.39
Run 3; Zero density 75%					
T	1.60	4.30	3.38	3.16	6.44
MP	7.22	6.74	10.96	9.62	9.48
MS	0.0	1.80	0.0	0.52	1.76
σ_T	0.16	0.40	0.14	0.15	0.28
σ_M	0.71	0.64	0.61	0.53	0.47
σ_S	0.0	0.49	0.0	0.10	0.33

Table 2
Size 5 × 10 × 20; sample size 50

Option	1	2	3	4	5
Run 4; Zero density 0%					
T	2.96	7.06	6.20	3.18	5.80
MP	10.70	6.52	12.90	6.88	5.52
MS	0.0	10.70	0.0	1.88	7.32
σ_T	0.17	0.47	0.22	0.11	0.32
σ_M	0.55	0.44	0.74	0.41	0.33
σ_S	0.0	0.11	0.0	0.10	0.63
Run 5; Zero density 50%					
T	3.00	9.04	6.60	4.08	9.22
MP	11.44	9.06	14.92	10.28	9.44
MS	0.0	8.40	0.0	1.74	6.78
σ_T	0.14	0.50	0.16	0.16	0.41
σ_M	0.52	0.48	0.60	0.49	0.42
σ_S	0.0	0.13	0.0	0.13	0.54
Run 6; Zero density 75%					
T	1.62	5.92	5.92	5.36	11.58
MP	6.04	5.56	11.58	10.56	10.36
MS	0.0	1.96	0.0	0.44	2.28
σ_T	0.15	0.52	0.18	0.22	0.48
σ_M	0.64	0.55	0.48	0.40	0.41
σ_S	0.0	0.54	0.0	0.11	0.38

Table 3
Size 3 × 25 × 50; sample size 25

Option	1	2	3	4	5
Run 7; Zero density 0%					
T	21.56	54.44	33.04	26.44	63.04
MP	22.76	20.08	30.16	24.80	23.36
MS	0.0	26.84	0.0	2.44	25.16
σ_T	1.68	4.53	1.69	1.52	4.58
σ_M	1.67	1.56	1.74	1.58	1.65
σ_S	0.0	3.49	0.0	0.17	2.72
Run 8; Zero density 50%					
T	32.76	39.68	38.68	33.32	84.80
MP	36.44	34.68	38.36	33.88	33.28
MS	0.0	34.28	0.0	2.56	25.36
σ_T	1.69	4.99	1.58	1.54	4.59
σ_M	1.84	1.88	1.74	1.68	1.69
σ_S	0.0	3.04	0.0	0.17	2.87
Run 9; Zero density 80%					
T	32.16	88.76	34.72	30.28	75.68
MP	36.80	35.00	34.60	30.80	30.00
MS	0.0	25.68	0.0	2.24	13.80
σ_T	2.29	6.45	2.51	2.27	5.81
σ_M	2.55	2.53	2.76	2.52	2.38
σ_S	0.0	3.47	0.0	0.23	1.64

Table 4
Size 5 × 25 × 50; sample size 25

Option	1	2	3	4	5
Run 10; Zero density 0%					
T	31.84	108.56	44.60	28.16	78.76
MP	30.12	24.88	30.76	22.00	19.28
MS	0.0	82.28	0.0	2.52	38.96
σ_T	2.91	12.07	1.90	1.59	6.11
σ_M	2.58	2.56	1.71	1.51	1.38
σ_S	0.0	12.58	0.0	0.18	6.06
Run 11; Zero density 50%					
T	44.28	159.76	54.08	37.44	117.28
MP	43.64	38.52	41.44	32.20	29.64
MS	0.0	98.56	0.0	2.52	39.44
σ_T	2.48	10.99	2.28	1.93	7.55
σ_M	2.39	2.59	2.16	1.86	1.79
σ_S	0.0	10.67	0.0	1.93	5.11
Run 12; Zero density 80%					
T	33.96	123.72	56.04	44.76	141.68
MP	34.80	31.68	44.12	38.04	36.80
MS	0.0	43.60	0.0	2.00	29.72
σ_T	2.42	9.02	2.98	3.71	12.08
σ_M	2.40	2.24	2.58	2.88	2.99
σ_S	0.0	6.50	0.0	0.28	3.81

Table 5
Size 3 × 50 × 100; sample size 5

Option	1	2	3	4	5
Run 13; Zero density 0%					
T	157.8	426.4	339.8	316.4	840.4
MP	51.4	49.8	102.4	97.2	96.8
MS	0.0	76.8	0.0	2.6	149.8
σ_T	38.3	108.5	12.9	14.4	38.2
σ_M	12.0	12.4	3.4	4.2	4.3
σ_S	0.0	22.2	0.0	0.4	11.4
Run 14; Zero density 50%					
T	386.2	1065.2	301.0	280.2	750.4
MP	127.4	124.4	93.8	90.2	89.6
MS	0.0	197.6	0.0	3.0	80.8
σ_T	55.4	142.5	36.4	34.7	97.4
σ_M	17.8	16.8	11.2	11.2	11.5
σ_S	0.0	30.7	0.0	0.3	17.3
Run 15; Zero density 90%					
T	274.2	748.4	340.6	327.4	872.0
MP	94.0	91.8	111.8	108.2	106.6
MS	0.0	86.4	0.0	1.6	64.8
σ_T	100.2	270.8	79.3	79.7	219.5
σ_M	34.4	33.4	27.2	27.1	27.1
σ_S	0.0	37.4	0.0	0.7	33.3

Table 6
Size 5 × 50 × 100; sample size 5

Option	1	2	3	4	5
Run 16; Zero density 0%					
T	317.6	1215.8	317.2	255.4	898.0
MP	95.6	88.6	82.4	67.0	68.8
MS	0.0	379.8	0.0	3.2	201.8
σ_T	45.8	160.3	22.2	30.6	123.1
σ_M	13.3	11.5	6.2	8.5	8.4
σ_S	0.0	55.7	0.0	0.4	48.6
Run 17; Zero density 50%					
T	383.0	1474.0	353.2	301.8	1096.8
MP	119.6	111.4	98.2	89.8	87.4
MS	0.0	383.0	0.0	3.0	150.8
σ_T	43.5	160.9	81.3	77.0	305.9
σ_M	12.7	11.3	24.3	23.5	23.7
σ_S	0.0	57.4	0.0	0.3	59.4
Run 18; Zero density 90%					
T	262.6	996.4	398.0	377.4	1326.4
MP	85.6	82.6	116.4	112.4	109.6
MS	0.0	82.4	0.0	1.2	86.6
σ_T	82.6	307.8	48.2	60.2	216.4
σ_M	26.3	24.9	14.8	17.3	17.3
σ_S	0.0	47.9	0.0	0.6	19.0

It is also apparent that Option 3, which invariably follows the sequential maximization scheme to its final conclusion without reference to any sub-problem, is generally poorer in terms of computation time than either Options 1 or 4. The clear implication is that the first efficient extreme point is typically obtained before all k objectives have been treated in Option 3.

For the small test problems, Option 1 yielded better mean computation times than did Option 4. This advantage was relatively small for low to moderate zero densities but more pronounced for high zero density (see runs 1–6). However, in the other runs neither Option 1 nor Option 4 dominated the other. This suggests the tentative conclusion that the insurance provided by Option 4 (i.e., the guarantee that it reaches an efficient extreme point if one exists) is not achieved at the cost of greatly increased computation time as compared to Option 1, which does not have this fail-safe property. Indeed the computation times for Option 4 were in some instances better than those for Option 1 (see runs 9, 10, 11, 14, 16, 17).

The effect of zero density is not clear. In runs 3 and 6, the best computation times were sharply less than those for runs 1, 2, 4, 5. By contrast, the best times in runs 7 and 10 were lower than those for runs 8, 9, 11, 12. Runs 13–18 also present a mixed picture.

The general influence of an increase in the number of criteria, other things being equal, appears to be to increase the average time required in Phase 2. However, there are exceptions as can be seen by comparing runs 13 and 16 (Option 4), runs 14 and 17 (Option 1), and runs 15 and 18 (Option 1).

As a summary of the results reported in this section we re-iterate the apparent indication that the sharpness of Option 4 in identifying the various cases of Theorem 3.1 is obtained with occasional increases in computation times over Option 1. However, with impressive frequency Option 4 is actually better than Option 1.

6. Conclusion

Phase 3 of the algorithm has been implemented, and testing of computational options is now in progress. This phase levies a significant bookkeeping task on the algorithm in order to avoid revisiting a previously computed efficient extreme point. Moreover, degeneracy presents special problems because the number of extreme points adjacent to a given extreme point exceeds the number of nonbasic variables. In these circumstances, special procedures are employed to insure examination of each adjacent extreme point. These topics will be the subject of future reports.

Acknowledgment

Constructive discussions with D.S. Rubin are gratefully acknowledged.

References

[1] K.J. Arrow, E.W. Barankin and D. Blackwell, "Admissible points of convex sets", in: *Contributions to the theory of games*, Eds. H.W. Kuhn and A.W. Tucker (Princeton Univ. Press, Princeton, N.J., 1953) 87–91.
[2] A. Charnes and W.W. Cooper, *Management models and industrial applications of linear programming*, Vol. 1 (Wiley, New York, 1961) 299–310.
[3] A.M. Geoffrion, "Strictly concave parametric programming, Part I", *Management Science* 13 (3) (1966) 244–253.
[4] A.M. Geoffrion, "Proper efficiency and the theory of vector maximization", *Journal of Mathematical Analysis and Applications* 22 (1968) 618–630.
[5] S. Karlin, *Mathematical methods and theory in games, programming and economics*, Vol. I (Addison-Wesley, Reading, Mass., 1959) 216–218.
[6] T. Koopmans, "Analysis of production as an efficient combination of activities", in: *Activity analysis of production and allocation*, Cowles Commission Monograph Vol. 13 , (Wiley, New York, 1951) 33–97.
[7] H.W. Kuhn and A.W. Tucker, "Nonlinear programming", in: *Proceedings of the* 2nd *Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Berkeley, Calif., 1950) 481–492.
[8] O. Mangasarian, *Nonlinear programming* (McGraw-Hill, New York, 1969) 27–35.
[9] M. Markowitz, "The optimization of a quadratic function subject to linear constraints", *Naval Research Logistics Quarterly* 3 (1, 2) (1956) 111–133.
[10] J. Philip, "Algorithms for the vector maximization problem", *Mathematical Programming* 2 (1972) 207–229.