An All-Linear Programming Relaxation Algorithm for Optimizing over the Efficient Set*

HAROLD P. BENSON
University of Florida, College of Business Administration, Gainesville, Florida 32611-2017, U.S.A.

(Received: 3 April 1990; accepted: 24 October 1990)

Abstract. The problem (P) of optimizing a linear function over the efficient set of a multiple objective linear program has many important applications in multiple criteria decision making. Since the efficient set is in general a nonconvex set, problem (P) can be classified as a global optimization problem. Perhaps due to its inherent difficulty, it appears that no precisely-delineated implementable algorithm exists for solving problem (P) globally. In this paper a relaxation algorithm is presented for finding a globally optimal solution for problem (P). The algorithm finds an exact optimal solution to the problem after a finite number of iterations. A detailed discussion is included of how to implement the algorithm using only linear programming methods. Convergence of the algorithm is proven, and a sample problem is solved.

Key words. Global optimization, multiple criteria decision making, relaxation algorithms, efficient set, multiple objective linear programming.

1. Introduction

Assume that $p \geqq 2$ is an integer, and that $c_1, c_2, \ldots, c_p \in R^n$ are row vectors. Let C be the $p \times n$ matrix whose ith row is given by c_i, $i = 1, 2, \ldots, p$, and let X be a nonempty, compact polyhedron in R^n. Then the *multiple objective linear programming* problem (MOLP), given by

$$\text{VMAX: } Cx, \text{ subject to } x \in X,$$

can be viewed as the problem of finding all solutions that are efficient in the sense of the following definition.

DEFINITION 1.1. A point x^0 is said to be an *efficient* solution of problem (MOLP) when $x^0 \in X$, and whenever $Cx \geqq Cx^0$ for some $x \in X$, then $Cx = Cx^0$.

An efficient solution is also called a *nondominated* or *Pareto-optimal* solution. The functions $f_i(x) = \langle c_i, x \rangle$, $i = 1, 2, \ldots, p$, are called the *objective* or *criterion functions* for problem (MOLP). Let X_E denote the set of all efficient solutions of problem (MOLP).

The central problem of interest in this paper is the problem of optimizing a linear function over the efficient set X_E of problem (MOLP). This problem,

* Research supported by a grant from the College of Business Administration, University of Florida, Gainesville, Florida, U.S.A.

Journal of Global Optimization **1**: 83–104, 1991
© 1991 *Kluwer Academic Publishers. Printed in the Netherlands.*

denoted henceforth as problem (P), is given by

$$\max\langle d, x\rangle, \text{ subject to } x \in X_E,$$

where $d \in R^n$. Let θ denote the optimal objective function value of problem (P).

Let c be any element of $\{c_1, c_2, \ldots, c_p\}$. An important special case of problem (P) is to find the value of ϕ such that

$$\phi = \min\langle c, x\rangle, \text{ subject to } x \in X_E.$$

The latter problem, which we will denote as problem (Q), seeks the minimum value ϕ of criterion function $f(x) = \langle c, x\rangle$ over the efficient set X_E of problem (MOLP). When $d = -c$, any optimal solution for problem (P) is also an optimal solution for problem (Q), and vice versa. Therefore, any algorithm for solving problem (P) can also be used to solve problem (Q).

Problem (MOLP) has been extensively studied and increasingly used as a decision aid during the past twenty years (see for instance, books and reviews by Evans [11], Hansen [14], Hemming [15], Rosenthal [29], Steuer [30], Yu [33], and Zeleny [34]). It has become more and more popular as researchers and practitioners have come to realize that many decision making situations involve multiple criteria. In most cases, these criteria are in conflict.

Many of the approaches for analyzing problem (MOLP) involve the generation of points in X_E [11, 30, 33]. Two of these, in particular, are quite commonly used. The first, called the *vector maximization* approach, is a two-stage process. First, either the entire set of efficient solutions or the subset of efficient solutions which are also extreme points of X is mathematically generated. Second, the generated set of efficient solutions is presented to the decision maker. He examines this set and the attainable objective function values and tradeoffs that it reveals. He thereby chooses a most preferred efficient solution. The other common class of approaches which generates points in X_E is the *interactive* approach. In this approach the decision maker, by interacting with a computerized routine, searches points in the efficient set until he finds one that he most prefers.

Problem (P) arose at least partially in response to some of the difficulties involved in using problem (MOLP) as a decision aid. Although not nearly as extensively studied as problem (MOLP), problem (P) has received increasing attention in recent years. It can serve several useful purposes in multiple criteria decision making. Philip [27], who in 1972 first proposed the problem, suggested that it can be useful because unlike problem (MOLP), which only allows the consideration of the relative importance of the different activities x_j, $j = 1, 2, \ldots, n$, problem (P) measures their absolute importance. Benson, who later studied properties of problem (P) and of more general versions of it [4, 7], pointed out that by solving problem (P) rather than problem (MOLP), the computational burden of generating the entire efficient set is avoided. He noted that, in addition, such an approach does not require the decision maker to choose a solution from a potentially overwhelmingly-large set of efficient solutions. To help describe situations in which problem (P) may arise, he also presented an

illustrative application of the problem to a multiple objective production-employment profit maximization problem.

Problem (Q), although mathematically a special case of problem (P), has its own unique uses in multiple criteria decision making. Several of these, discussed in recent papers by Weistroffer [32], Dessouky, Ghiassi, and Davis [10], and Isermann and Steuer [22], are based upon the fact that the optimal value ϕ of problem (Q), together with the maximum value of $\langle c, x \rangle$ attained over X_E, defines the range of values that the objective function $\langle c, x \rangle$ achieves over the efficient set. Knowledge of this range of values can serve several practical purposes. First, it can aid the decision maker in setting goals. Second, it can help the decision maker to evaluate the utility of the values of $\langle c, x \rangle$ achieved by individual efficient solutions (see Ghiassi *et al.* [13] for an application). Third, if the range is relatively narrow, the decision maker may decide that in order to find a most preferred solution, it is not necessary to rank the objective function very high in importance relative to other objective functions with broader ranges. Fourth, if the range is extremely narrow, the decision maker may decide that he can eliminate the objective function $\langle c, x \rangle$ entirely from problem (MOLP).

Other possible benefits of problem (Q) involve the use of its optimal objective function value ϕ in various algorithms for solving problem (MOLP). For instance, as pointed out in [10, 22 and 28], several interactive algorithms for finding a most preferred solution to problem (MOLP) could benefit if the precise optimal values of problem (Q) for the objective functions $\langle c, x \rangle$ of problem (MOLP) were used rather than estimates of these values. Among such interactive algorithms are STEM [3], and the algorithms of Belenson and Kapur [2] and Kok and Lootsma [23].

Mathematically, problem (P) can be classified as a *global optimization* problem (also called a *nonconvex programming* problem), since its feasible region X_E is, in general, a nonconvex set [6, 16–19, 21, 24–26]. Such problems possess local optima which need not be globally optimal. Furthermore, the number of these local optima frequently is very large. Therefore, global optimization problems are much more difficult to solve than convex programming problems. In fact, except for certain special cases, most global optimization problems of realistic sizes cannot yet be solved (see, for instance [20, 21, 25, 26]).

Perhaps due to its inherent difficulty, it appears that no major efforts have been made to precisely delineate algorithms for solving problem (P). In a small section of his 1972 paper, Philip [27] schematically described a cutting plane procedure for solving the problem. Several years later, Isermann and Steuer [22] independently suggested using the same approach for problem (Q). In both algorithms, each time a cutting plane constraint is added, it is required to search along the intersection of the cutting plane and the current feasible polyhedron for all newly-created efficient extreme points. Since neither algorithm explains how to mathematicaly accomplish this search, it is not clear how to implement these algorithms.

In this paper a relaxation algorithm is presented for finding a globally optimal

solution for problem (P). The algorithm always terminates with an exact optimal solution to the problem after a finite number of iterations. Depending upon in which step of an iteration the algorithm terminates, either an extreme point optimal solution or a non-extreme point optimal solution for problem (P) is found. A detailed discussion is included of how to implement the algorithm using only linear programming methods. Convergence of the algorithm is also proven.

The next section gives the relaxation algorithm and proves that it finds an optimal solution for problem (P) after a finite number of iterations. Section 3 shows in detail how the algorithm can be implemented using only linear programming methods. In Section 4 a small example problem is solved to illustrate the relaxation algorithm and its implementation via linear programming. Concluding remarks are given in Section 5. In the Appendix, we prove the validity of the implementation given in Section 3 for one of the steps of the algorithm.

2. The Relaxation Algorithm

Let X_{ex} denote the set of extreme points of the polyhedron X. Assume that X_{ex} is nonempty. Then, from Theorem 4.5 in [7], problem (P) has an optimal solution which belongs to X_{ex}. We will assume henceforth in the paper, for each optimal solution $x^d \in X_{ex}$ to the linear program

$$\max\langle d, x\rangle, \text{ subject to } x \in X,$$

that $x^d \notin X_E$. This assumption merely states, from the statement before it, that the requirement $x \in X_E$ in problem (P) is *essential* in the sense that if the relaxed requirement that x belong to X were used instead, a larger optimal value could be achieved.

Let $\Lambda = \{\lambda \in R^p \mid \langle e, \lambda\rangle \leqq M, \lambda \geqq e\}$, where $e \in R^p$ is a vector whose entries each equal to one, and M is a positive real number. For sufficiently large M, from [27], x^0 belongs to X_E if, and only if, there exists at least one $\lambda^0 \in \Lambda$ such that x^0 is an optimal solution to the "weighted sum" problem (P_λ) given by

$$\max \lambda^T C x, \text{ subject to } x \in X,$$

with $\lambda = \lambda^0$. Let us assume henceforth that M is chosen to be large enough to guarantee this property. Then problem (P) can be equivalently written as the infinitely-constrained problem (PI) given by

$$\begin{aligned} &\theta = \max\langle d, x\rangle, \\ &\text{s.t. } \lambda^T C x \geqq \lambda^T C \hat{x} \text{ for all } \hat{x} \in X, \\ &\qquad x \in X, \\ &\qquad \lambda \in \Lambda, \end{aligned} \tag{1}$$

where both x and λ are variables. Notice that each of the infinite number of constraints in (1) involves the bilinear term $\lambda^T C x$.

The algorithm solves problem (P) by solving the equivalent problem (PI) with a relaxation procedure. The relaxation procedure is similar to one used by Blanken-

ship and Falk [9] to solve certain classes of infinitely-constrained optimization problems. In a general iteration k of the algorithm, a finitely-constrained relaxation (PI_k) of problem (PI) is solved. Problem (PI_k) is identical to problem (PI), except that it substitutes the constraints

$$\lambda^T Cx \geqq \lambda^T Cx^j,\ j = 0, 1, 2, \ldots, k, \tag{2}$$

for (1), where $x^0, x^1, \ldots, x^k$ are certain elements of $X_{\mathrm{ex}} \cap X_E$ obtained from previous iterations of the algorithm. Thus, problem (PI_k) has a finite, rather than an infinite, number of constraints involving the bilinear term $\lambda^T Cx$. If an optimal solution is found by the algorithm to the relaxation (PI_k) which lies in X_E, then the algorithm terminates. Otherwise, a new element x^{k+1} of $X_{\mathrm{ex}} \cap X_E$ is found. In the next iteration $k+1$, a "tighter" relaxation (PI_{k+1}) of problem (PI) is solved which is identical to the previous relaxation, except for including the additional "cutting" constraint

$$\lambda^T Cx \geqq \lambda^T Cx^{k+1}$$

in (2). The algorithm may be stated as follows.

RELAXATION ALGORITHM FOR PROBLEM (P)

Initialization Step. Choose any point $x^0 \in X_E \cap X_{\mathrm{ex}}$. Find any vector $\bar{\lambda}^0 \in \Lambda$ such that x^0 is an optimal solution to problem (P_λ) with $\lambda = \bar{\lambda}^0$. Set $k = 0$ and go to Iteration k.

Iteration k, $k \geqq 0$.

Step k.1. Find an optimal solution $(\bar{x}^{k+1}, \bar{\lambda}^{k+1})$ to the relaxed problem (PI_k) given by

$$\begin{aligned} &\max \langle d, x \rangle, \\ &\text{s.t. } \lambda^T Cx \geqq \lambda^T Cx^j,\ j = 0, 1, \ldots, k, \\ &\qquad x \in X, \\ &\qquad \lambda \in \Lambda, \end{aligned}$$

where both x and λ are variables. Let θ_{k+1} denote the optimal value of problem (PI_k).

Step k.2. If $\theta_{k+1} = \langle d, x^j \rangle$ for some $j \in \{0, 1, \ldots, k\}$, then STOP: x^j is an optimal solution for problem (P) for any $\hat{j} \in \{0, 1, \ldots, k\}$ such that $\theta_{k+1} = \langle d, x^{\hat{j}} \rangle$. If $\theta_{k+1} \neq \langle d, x^j \rangle$ for all $j \in \{0, 1, \ldots, k\}$, continue.

Step k.3. Solve the linear programming problem (T_k) given by

$$\begin{aligned} &\max e^T Cx, \\ &\text{s.t. } Cx \geqq C\bar{x}^{k+1}, \\ &\qquad x \in X, \end{aligned}$$

for any optimal solution $x^* \in X$. If $e^T Cx^* = e^T C\bar{x}^{k+1}$, then STOP: $\bar{x}^{k+1}$ is an optimal solution for problem (P). If $e^T Cx^* \neq e^T C\bar{x}^{k+1}$, continue.

Step k.4. Solve the linear programming problem $(P_{\bar{\lambda}^{k+1}})$ given by

$$\max(\bar{\lambda}^{k+1})^T Cx, \text{ subject to } x \in X$$

for any optimal solution $x^{**} \in X_{\text{ex}}$. Set $x^{k+1} = x^{**}$ and $k = k+1$, and go to Iteration k.

REMARK 2.1. Notice for each $k \geqq 0$ that the vector $\bar{\lambda}^{k+1}$ generated by the algorithm lies in Λ, and that x^{k+1} is an optimal solution in X_{ex} for the linear program $(P_{\bar{\lambda}^{k+1}})$. From this and the choice of x^0 in the Initialization Step, it follows that the points $x^0, x^1, x^2, \ldots$ generated by the algorithm are efficient extreme points for problem (MOLP). Thus, for each $t \geqq 0$, $\langle d, x^t \rangle$ is a *lower bound* for the optimal value θ of problem (P). The lower bounds $\langle d, x^t \rangle$, $t = 0, 1, 2, \ldots$, however, need *not* satisfy $\langle d, x^t \rangle \leqq \langle d, x^{t+1} \rangle$ for each $t \in \{0, 1, 2, \ldots\}$.

By the following result, the points $\bar{x}^1, \bar{x}^2, \ldots$ provide a nonincreasing sequence of *upper bounds* for θ.

THEOREM 2.1. *Let $t \in \{1, 2, \ldots\}$ and let $\bar{x}^t$ and $\bar{x}^{t+1}$ be two points generated by the algorithm in Iterations $t-1$ and t, respectively. Then $\theta \leqq \langle d, \bar{x}^{t+1} \rangle \leqq \langle d, \bar{x}^t \rangle$.*

Proof. Let $y \in X_E$ be any optimal solution for problem (P). Then, since $y \in X_E$, for some $\lambda^y \in \Lambda$, $(\lambda^y)^T Cy \geqq (\lambda^y)^T C\hat{x}$ for all $\hat{x} \in X$ [27]. Therefore, $(\lambda^y)^T Cy \geqq (\lambda^y)^T Cx^j$, $j = 0, 1, 2, \ldots, w$, where w is an arbitrary nonnegative integer and x^j, $j = 0, 1, 2, \ldots, w$, are points generated by the algorithm by the start of Iteration w. Since $y \in X$ and $\lambda^y \in \Lambda$, this implies that (y, λ^y) is a feasible solution for problem (PI_w) solved in Step $w.1$ of the algorithm. From Step $w.1$, $\langle d, \bar{x}^{w+1} \rangle \geqq \langle d, y \rangle = \theta$ follows. By the choice of w, this implies that $\langle d, \bar{x}^t \rangle \geqq \theta$ for each $t \in \{1, 2, \ldots\}$. The inequality $\langle d, \bar{x}^{t+1} \rangle \leqq \langle d, \bar{x}^t \rangle$ follows for any $t \in \{1, 2, \ldots\}$ from the fact that for each such t, the feasible region of problem (PI_t) is a subset of the feasible region of problem (PI_{t-1}). □

Let $t \in \{0, 1, 2, \ldots\}$, and let $x^0, x^1, \ldots, x^t$ and $\bar{x}^{t+1}$ be points generated by the algorithm after Step $t.1$ of Iteration t has been executed. Let $\text{LB}_t = \max\{\langle d, x^w \rangle \mid w = 0, 1, \ldots, t\}$ and $\text{UB}_t = \theta_{t+1}$. Then, from Step $t.1$ of the algorithm, from Theorem 2.1, and from Remark 2.1,

$$\text{LB}_t \leqq \theta \leqq \text{UB}_t . \tag{3}$$

From (3), the algorithm may be terminated after Step $t.1$ of some Iteration t if $(\text{UB}_t - \text{LB}_t)$ is sufficiently small. If this were done, then any efficient extreme point x^w, $w \in \{0, 1, \ldots, t\}$ found by the algorithm for which $\text{LB}_t = \langle d, x^w \rangle$ could be considered to be an optimal or near-optimal solution for problem (P). However, such a premature termination may not be necessary because, as we shall now show, the algorithm always terminates in a finite number of iterations with a valid optimal solution to problem (P).

To prove that the algorithm is finite and valid, we must first present two theorems which describe two additional properties of the algorithm. The first of these two properties is given in the next theorem. For any subset Y of X, let Y_E denote the set of efficient solutions for the multiple objective problem obtained from (MOLP) by substituting Y for X in (MOLP).

THEOREM 2.2. *Let* $t \in \{0, 1, 2, \ldots\}$, *and let* H *be the convex hull of* $\{x^0, x^1, \ldots, x^t, \bar{x}^{t+1}\}$. *Then* $\bar{x}^{t+1} \in H_E$.

Proof. In this proof, for any vectors $w, v \in R^p$, let $w \geq v$ denote that $w \geqq v$ and $w \neq v$. To prove the theorem, assume, to the contrary, that $\bar{x}^{t+1} \notin H_E$. Then, by Definition 1.1., there exists a point $x \in H$ such that $Cx \geqq C\bar{x}^{t+1}$. Since $x \in H$, there exist nonnegative constants $\alpha_0, \alpha_1, \ldots, \alpha_{t+1}$ which sum to unity such that

$$x = \sum_{j=0}^{t} \alpha_j x^j + \alpha_{t+1}\bar{x}^{t+1} .$$

Since $\alpha_0, \alpha_1, \ldots, \alpha_t, \alpha_{t+1}$ sum to unity and $Cx \geqq C\bar{x}^{t+1}$, this equation implies that

$$\sum_{j=0}^{t} \alpha_j Cx^j + \alpha_{t+1} C\bar{x}^{t+1} \geqq \sum_{j=0}^{t} \alpha_j C\bar{x}^{t+1} + \alpha_{t+1} C\bar{x}^{t+1} . \tag{4}$$

From Step $t.1$ of the algorithm,

$$(\bar{\lambda}^{t+1})^T C\bar{x}^{t+1} \geqq (\bar{\lambda}^{t+1})^T Cx^j, \; j = 0, 1, \ldots, t ,$$

so that by nonnegativity of $\alpha_0, \alpha_1, \ldots, \alpha_t$,

$$\left[\sum_{j=0}^{t} \alpha_j (\bar{\lambda}^{t+1})^T C\bar{x}^{t+1}\right] \geqq \left[\sum_{j=0}^{t} \alpha_j (\bar{\lambda}^{t+1})^T Cx^j\right] . \tag{5}$$

Subtracting $\alpha_{t+1} C\bar{x}^{t+1}$ from both sides of inequality (4) and taking the inner product of the remaining term on each side with $\bar{\lambda}^{t+1} > 0$ yields

$$\sum_{j=0}^{t} \alpha_j (\bar{\lambda}^{t+1})^T Cx^j > \sum_{j=0}^{t} \alpha_j (\bar{\lambda}^{t+1})^T C\bar{x}^{t+1} .$$

But this contradicts inequality (5). Therefore, our assumption that $\bar{x}^{t+1} \notin H_E$ must be false, so that the theorem is proved. □

Next, we show that the efficient extreme points x^t, $t = 0, 1, \ldots$ generated by the algorithm are distinct form one another.

THEOREM 2.3. *Let* $t \in \{0, 1, 2, \ldots\}$. *Then* $x^{t+1} \notin \{x^0, x^1, \ldots, x^t\}$.

Proof. Since x^{t+1} is generated in Step $t.4$ of the algorithm, $e^T Cx^* \neq e^T C\bar{x}^{t+1}$ occurs in Step $t.3$. Therefore, from Benson [5], $\bar{x}^{t+1} \notin X_E$. This implies that for each $\lambda \in \Lambda$, $\bar{x}^{t+1}$ is a suboptimal solution for problem (P_λ), by [27]. Therefore, from Step $t.4$, $(\bar{\lambda}^{t+1})^T Cx^{t+1} > (\bar{\lambda}^{t+1})^T C\bar{x}^{t+1}$. From Step $t.1$, $(\bar{\lambda}^{t+1})^T C\bar{x}^{t+1} \geqq (\bar{\lambda}^{t+1})^T Cx^j$, $j = 0, 1, \ldots, t$. By transitivity, $(\bar{\lambda}^{t+1})^T Cx^{t+1} > (\bar{\lambda}^{t+1})^T Cx^j$, $j = 0, 1, \ldots, t$, which implies the desired result. □

We are now able to show that the algorithm is finite and valid.

THEOREM 2.4. (1) *The algorithm terminates after a finite number of iterations.*
(2) *The algorithm terminates only when an optimal solution for problem (P) has been found.*

Proof. (1). From Remark 2.1, $x^t \in X_{\text{ex}} \cap X_E$ for each $t \in \{0, 1, 2, \ldots\}$. By Theorem 2.3, the points x^t, $t = 0, 1, 2, \ldots$ are distinct from one another. During each iteration t of the algorithm, the execution of Step $t.4$ thus generates an element of $X_{\text{ex}} \cap X_E$ not previously found. The number of elements q in $X_{\text{ex}} \cap X_E$ is finite. Therefore, after at most $q-1$ iterations, the set of points $\{x^0, x^1, \ldots, x^k\}$ used in the constraints of problem (PI_k) will equal $X_{\text{ex}} \cap X_E$. From Theorem 2.2, this implies that after a finite number of iterations, the point $\bar{x}^{k+1}$ derived in Step $k.1$ will be an efficient point of the convex hull of $[X_{\text{ex}} \cap X_E] \cup \{\bar{x}^{k+1}\}$. Since X_E is a subset of this convex hull [33], it is easily shown that this implies that $\bar{x}^{k+1} \in X_E$. Therefore, from [5], $\bar{x}^{k+1}$ will be an optimal solution to problem (T_k) in Step $k.3$. Since this implies that the algorithm terminates, and $\bar{x}^{k+1}$ is derived after a finite number of iterations, part (1) is proven.

(2) Either (a) the algorithm terminates in some Step $t.2$ or (b) it terminates in some Step $t.3$, where t is a finite, nonnegative integer.

Case 1. The algorithm terminates in Step $t.2$ for some finite, nonnegative integer t. Then $\theta_{t+1} = \langle d, x^j \rangle$ for some $j \in \{0, 1, 2, \ldots, t\}$. Let $x^{\hat{j}}$ satisfy $\theta_{t+1} = \langle d, x^{\hat{j}} \rangle$. From Theorem 2.1 and Step $t.1$, $\theta_{t+1} \geqq \theta$. By Remark 2.1, $\theta \geqq \langle d, x^{\hat{j}} \rangle$. The three previous statements together imply that $\langle d, x^{\hat{j}} \rangle = \theta$. From Remark 2.1, $x^{\hat{j}} \in X_E$. Therefore, $x^{\hat{j}}$ is an optimal solution for problem (P).

Case 2. The algorithm terminates in Step $t.3$ for some finite, nonnegative integer t. Then, from Step $t.3$, since $\bar{x}^{t+1} \in X$, $\bar{x}^{t+1}$ is an optimal solution for the linear program (T_t) given in this step. From [5], this implies that $\bar{x}^{t+1} \in X_E$. From Theorem 2.1 and Step $t.1$, $\langle d, \bar{x}^{t+1} \rangle = \theta_{t+1} \geqq \theta$. Additionally, since $\bar{x}^{t+1} \in X_E$, $\theta \geqq \langle d, \bar{x}^{t+1} \rangle$. The latter two statements imply that $\theta = \langle d, \bar{x}^{t+1} \rangle$ and that $\bar{x}^{t+1}$ is an optimal solution for problem (P). □

REMARK 2.2. Notice from Theorem 2.4(2) and Remark 2.1 that if the algorithm terminates in Step $t.2$ for some $t \geqq 0$, then $x^{\hat{j}}$ is an optimal *extreme* point solution for problem (P). On the other hand, if termination occurs in Step $t.3$ for some $t \geqq 0$, then, although $\bar{x}^{t+1}$ is an optimal solution for problem (P), it need not be an extreme point of X. This is because $\bar{x}^{t+1}$, for each $t \geqq 0$, is part of the optimal solution found in Step $t.1$ to the relaxed problem (PI_t), and problem (PI_t) does not necessarily have an optimal solution $(\bar{x}, \bar{\lambda})$ for which $\bar{x} \in X_{\text{ex}}$.

REMARK 2.3. Each of the constraints

$$\lambda^T C x \geqq \lambda^T C x^j, \; j = 0, 1, \ldots, k$$

in the relaxed problem (PI_k) in Step $k.1$ of the algorithm involves the bilinear term $\lambda^T Cx$. This causes the feasible region of problem (PI_k) to be a nonconvex set, so that problem (PI_k) is a global optimization problem. Such problems, as pointed out earlier, are much more difficult to solve than convex programs. However considerable progress has been made in the past 25 years in developing practical algorithms for solving certain special classes of global optimization problems (see, for example, [1, 4, 6, 8, 9, 12, 16–21, 24–26, 31] and references therein). In the next section, a branch and bound procedure for solving problem (PI_k) will be presented which involves solving a sequence of linear programming problems.

3. Implementation Issues

Except for the Initialization Step and the Iteration Step $k.1$, it is immediately apparent that the relaxation algorithm is implementable via linear programming methods. As we shall now show, the Initialization Step and the Iteration Step $k.1$ can also be accomplished via linear programming. Of course, other means of implementing these steps, especially Iteration Step $k.1$, could be formulated. However, since efficient methods such as the simplex method and Karamarkar's method exist for solving linear programs, the implementations to be shown here are expected to represent potentially valuable practical approaches.

3.1. THE INITIALIZATION STEP

The Initialization Step calls for finding an element x^0 of $X_E \cap X_{ex}$ and an associated vector $\bar{\lambda}^0 \in \Lambda$ such that x^0 is an optimal solution to problem (P_λ) with $\lambda = \bar{\lambda}^0$. This is readily accomplished by choosing any $\bar{\lambda}^0 \in \Lambda$ (e.g. $\bar{\lambda}^0 = e$) and solving the linear program (P_λ) with $\lambda = \bar{\lambda}^0$ for an optimal extreme point solution x^0 [27]. Such a solution is guaranteed to exist, since X is a nonempty, compact polyhedron.

3.2. THE ITERATION STEP $K.1$

In order to implement the Iteration Step $k.1$, the relaxed global optimization problem (PI_k) must be solved. No algorithm in the global optimization literature seems to exist which is specifically designed to solve problems such as problem (PI_k) [18, 21, 25, 26]. Fortunately, however, due to its special structure, we are able to offer an all-linear programming procedure for solving this problem. Preliminary computational research has shown that of the approaches used thus far for global optimization, branch and bound is most frequently the most efficient [8, 18, 20, 21]. Indeed, several general prototype branch and bound algorithms for solving various classes of global optimization problems have been proposed [6, 12, 17, 19]. Therefore, the procedure that we have developed for problem

(PI_k) uses branch and bound. While it does not specifically follow the format of any of the prototype algorithms, it is motivated by one recently given by Horst [17].

The branch and bound procedure for Iteration Step $k.1$ solves problem (PI_k) by solving an equilvalent problem (QI_k). To explain how problem (QI_k) is derived from problem (PI_k), we introduce some additional notation. In problem (PI_k) let a^j denote Cx^j for each $j = 0, 1, \ldots, k$. For each $i \in \{1, 2, \ldots, p\}$, let

$$\underline{v}_i = \min - \langle c_i, x \rangle, \text{ subject to } x \in X,$$

and

$$\bar{v}_i = \max - \langle c_i, x \rangle, \text{ subject to } x \in X.$$

Let $V = \{v \in R^p \mid \underline{v} \leqq v \leqq \bar{v}\}$, where the ith components of $\underline{v}$ and $\bar{v}$ are $\underline{v}_i$ and $\bar{v}_i$, respectively, $i = 1, 2, \ldots, p$. Let $L = \{\lambda \in R^p \mid 1 \leqq \lambda_i \leqq M - p + 1, \quad i = 1, 2, \ldots, p\}$. Finally, let $Q = \{(x, v, \lambda) \in R^{n+2p} \mid x \in X, \ v + Cx = 0, \ \langle e, \lambda \rangle \leqq M\}$. Then by adding the variables $v = -Cx$ to problem (PI_k), we obtain the equivalent problem (QI_k) given by

$$\begin{aligned} &\min - \langle d, x \rangle, \\ &\text{s.t. } \langle v, \lambda \rangle + \langle a^j, \lambda \rangle \leqq 0, \ j = 0, 1, \ldots, k, \\ &\quad (x, v, \lambda) \in Q, \\ &\quad v \in V, \\ &\quad \lambda \in L. \end{aligned} \tag{6}$$

Notice that (x^*, λ^*) is an optimal solution for problem (PI_k) if and only if (x^*, v^*, λ^*) is an optimal solution for problem (QI_k), where $v^* = -Cx^*$.

To aid in the presentation of the procedure for solving problem (QI_k), we will use the following definitions.

DEFINITION 3.1. The *convex envelope* of a function h taken over a convex subset H of its domain is that function h_H over H whose epigraph is the convex hull of the epigraph of h taken over H.

DEFINITION 3.2. A set $H = \{x \in R^n \mid c \leqq x \leqq d\}$, where $c, d \in R^n$, is called an *n-rectangle* when $\dim H = n$.

DEFINITION 3.3. Let $H \subseteq R^n$ be an n-rectangle. Suppose that H is divided into two n-rectangles with equal volumes such that the midpoint of one of the longest edges of H is an extreme point of both new rectangles. Then H is said to be divided into two sub-n-rectangles by *bisection*. (see [16] and [31], for example).

DEFINITION 3.4. Let T be a compact set in R^n. A set $\boldsymbol{M} = \{T_1, T_2, \ldots, T_u\}$ of finitely many compact sets T_i, $i = 1, 2, \ldots, u$, of T is a *partition* of T when T

equals the union of T_i, $i = 1, 2, \ldots, u$, and each pair of sets T_i and T_j, $i \neq j$, intersect only on their boundaries relative to T.

It can be shown that the convex envelope h_H of a function h is the pointwise supremum of all convex functions which underestimate h over H.

The procedure for solving problem (QI_k) will repeatedly use an underestimate of the convex envelope g_G of the indefinite quadratic function $g(x_1, x_2) = x_1 x_2$ taken over the two-rectangle $G = \{(x_1, x_2) \in R^2 \mid \underline{r} \leqq x_1 \leqq \bar{r}, \underline{s} \leqq x_2 \leqq \bar{s}\}$, where $\underline{r}$, $\bar{r}$, $\underline{s}$, $\bar{s} \in R$. From McCormick [24], the formula for this convex envelope is given by

$$g_G(x_1, x_2) = \max[\underline{s}x_1 + \underline{r}x_2 - \underline{rs}, \bar{s}x_1 + \bar{r}x_2 - \bar{r}\bar{s}]\,. \tag{7}$$

Assume without loss of generality that $V \times L$ is a $2p$-rectangle. The procedure for solving problem (QI_k) consists of a branch and bound search. In each step w of the procedure, branching bisects a sub-$2p$-rectangle of $V \times L$ into two smaller sub-$2p$-rectangles. A new partition $\boldsymbol{M}_w$ of $V \times L$ is thereby created. Problem (QI_k) is thus separated into a finite number of problems, each of the form $(\mathrm{QI}_k^{w,t})$ given by

$$\begin{aligned}
&\min -\langle d, x\rangle\,,\\
&\text{s.t. } \langle v, \lambda\rangle + \langle a_j, \lambda\rangle \leqq 0\,, \quad j = 0, 1, \ldots, k\,,\\
&\qquad (x, v, \lambda) \in Q\,,\\
&\qquad v \in V^{w,t}\\
&\qquad \lambda \in L^{w,t}\,,
\end{aligned}$$

where $V^{w,t} \times L^{w,t}$ is an element of $\boldsymbol{M}_w$. The bounding process finds lower bounds for the optimal values of these problems. To accomplish this, certain linear programming relaxations of these problems are solved. It then calculates the minimum lower bound found for these problems. The value LB_w of this minimum is a lower bound for $-\theta_{k+1}$.

As the linear programming relaxations are solved for their optimal solutions, checks are performed to determine whether or not these solutions are also feasible solutions for problem (QI_k). As feasible solutions are found, updates are performed, if necessary, to the minimum value that $-\langle d, x\rangle$ achieves over the set of feasible solutions for problem (QI_k) thus far encountered in the procedure. This value UB is an upper bound for $-\theta_{k+1}$. Any feasible solution for problem (QI_k) achieving this value is called an *incumbent solution*. If, at the end of Step w, $\mathrm{LB}_w = \mathrm{UB}$, then the incumbent solution is optimal in problem (QI_k) and the procedure terminates. If not, then in Step $w + 1$ the branching process is again invoked to create a more refined partition of $V \times L$, and the process of obtaining lower and upper bounds for $-\theta_{k+1}$ is repeated.

To complete this informal description of the procedure, we must give the form of the linear programming relaxation of problem $(\mathrm{QI}_k^{w,t})$ that it uses. Let

$H = V^{w,t} \times L^{w,t}$ and let $h(v, \lambda) = \langle v, \lambda \rangle$. Notice that a lower bound for the optimal value of problem $(\mathrm{QI}_k^{w,t})$ is given by the optimal value of the convex programming relaxation $(C_k^{w,t})$ given by

$$\begin{aligned} &\min - \langle d, x \rangle \\ &\text{s.t. } h_H(v, \lambda) + \langle a^j, \lambda \rangle \leqq 0\,, \quad j = 0, 1, \ldots, k\,, \\ &\qquad (x, v, \lambda) \in Q\,, \\ &\qquad v \in V^{w,t}\,, \\ &\qquad \lambda \in L^{w,t}\,, \end{aligned} \tag{8}$$

where h_H denotes the convex envelope of h taken over H. The linear programming relaxation $(P_k^{w,t})$ of problem $(\mathrm{QI}_k^{w,t})$ used in the procedure is a linear programming relaxation of this convex problem. If we assume that

$$V^{w,t} = \bigtimes_{i=1}^{p} V_i^{w,t} = \bigtimes_{i=1}^{p} \{v_i \in R \mid \underline{v}_i^{w,t} \leqq v_i \leqq \bar{v}^{w,t}\}\,,$$

and

$$L^{w,t} = \bigtimes_{i=1}^{p} L_i^{w,t} = \bigtimes_{i=1}^{p} \{\lambda_i \in R \mid \underline{\lambda}_i^{w,t} \leqq \lambda_i \leqq \bar{\lambda}_i^{w,t}\}\,,$$

then, by reordering coordinates, H may be written as the cross product

$$H = \bigtimes_{i=1}^{p} (V_i^{w,t} \times L_i^{w,t})$$

of the two-rectangles $V_i^{w,t} \times L_i^{w,t}$, $i = 1, 2, \ldots, p$. Using this representation of H, equation (7), and a result of Al-Khayyal and Falk [1], the constraints (8) can be written

$$\begin{aligned} &\sum_{i=1}^{p} \max(\underline{\lambda}_i^{w,t} v_i + \underline{v}_i^{w,t} \lambda_i - \underline{v}_i^{w,t} \underline{\lambda}_i^{w,t},\ \bar{\lambda}_i^{w,t} v_i + \bar{v}_i^{w,t} \lambda_i - \bar{v}_i^{w,t} \bar{\lambda}_i^{w,t}) \\ &\qquad + \langle a^j, \lambda \rangle \leqq 0\,, \quad j = 0, 1, \ldots, k\,. \end{aligned}$$

For each $i = 1, 2, \ldots, p$, by using the subgradient $\frac{1}{2}(\underline{\lambda}_i^{w,t} + \bar{\lambda}_i^{w,t},\ \underline{v}_i^{w,t} + \bar{v}_i^{w,t})$ of the function $f_i(v_i, \lambda_i) = \max(\underline{\lambda}_i^{w,t} v_i + \underline{v}_i^{w,t} \lambda_i - \underline{v}_i^{w,t} \underline{\lambda}_i^{w,t},\ \bar{\lambda}_i^{w,t} v_i + \bar{v}_i^{w,t} \lambda_i - \bar{v}_i^{w,t} \bar{\lambda}_i^{w,t})$ at the point $\frac{1}{2}(\underline{v}_i^{w,t} + \bar{v}_i^{w,t},\ \underline{\lambda}_i^{w,t} + \bar{\lambda}_i^{w,t})$ to construct a linear underestimator of $f_i(v_i, \lambda_i)$, it is a simple exercise to show that $\frac{1}{2}(\underline{\lambda}_i^{w,t} + \bar{\lambda}_i^{w,t}) v_i + \frac{1}{2}(\underline{v}_i^{w,t} + \bar{v}_i^{w,t}) \lambda_i - \frac{1}{2}(\underline{v}_i^{w,t} \underline{\lambda}_i^{w,t} + \bar{v}_i^{w,t} \bar{\lambda}_i^{w,t})$ underestimates $f_i(v_i, \lambda_i)$ throughout $V_i^{w,t} \times L_i^{w,t}$. Using this fact, the procedure constructs and solves the linear programming relaxation $(P_k^{w,t})$ of problem $(C_k^{w,t})$ given by

$$\begin{aligned} &\min - \langle d, x \rangle \\ &\text{s.t. } \sum_{i=1}^{p} [\tfrac{1}{2}(\underline{\lambda}_i^{w,t} + \bar{\lambda}_i^{w,t}) v_i + \tfrac{1}{2}(\underline{v}_i^{w,t} + \bar{v}_i^{w,t}) \lambda_i] \\ &\qquad + \langle a^j, \lambda \rangle \leqq \tfrac{1}{2} \sum_{i=1}^{p} (\underline{v}_i^{w,t} \underline{\lambda}_i^{w,t} + \bar{v}_i^{w,t} \bar{\lambda}_i^{w,t})\,, \quad j = 0, 1, \ldots, k\,, \end{aligned} \tag{9}$$

$$(x, v, \lambda) \in Q\,, \tag{10}$$

$$v \in V^{w,t} \tag{11}$$

$$\lambda \in L^{w,t}\,. \tag{12}$$

Then for any optimal solution $(x^{w,t}, v^{w,t}, \lambda^{w,t})$ to linear program $(P_k^{w,t})$, $-\langle d, x^{w,t}\rangle$ is a lower bound for the optimal value of problem $(\mathrm{QI}_k^{w,t})$.

We may now give a formal statement of the branch and bound procedure for implementing Iteration Step k.1.

BRANCH AND BOUND PROCEDURE FOR ITERATION STEP K.1

Step 0.

Step 0.1. Write the relaxed problem (PI_k) in the form of problem (QI_k). Choose a value for PCNT between 0 and 100, inclusive. Set $V^{01} \times L^{01}$ equal to $V \times L$ and set $\boldsymbol{M}_0 = \{(V^{01} \times L^{01})\}$.

Step 0.2. Let $(x^c, v^c, \lambda^c) = (x^j, -Cx^j, \bar{\lambda}^j)$, where x^j is any element of argmin $\{-\langle d, x^j\rangle \mid j \in \{0, 1, \ldots, k\}\}$ and $x^j, \bar{\lambda}^j$, $j = 0, 1, \ldots, k$ are iterates obtained from the Relaxation Algorithm for Problem (P). Let $\mathrm{UB} = -\langle d, x^j\rangle$.

Step 0.3. Find an optimal solution $(x^{01}, v^{01}, \lambda^{01})$ for the linear program $(P_k^{0,1})$. Set $\beta(V^{01} \times L^{01}) = -\langle d, x^{01}\rangle$ and $\mathrm{LB}_0 = -\langle d, x^{01}\rangle$.

Step 0.4. If $(x^{01}, v^{01}, \lambda^{01})$ is not a feasible solution for problem (QI_k), go to Step 0.5. Otherwise, set $\mathrm{UB} = \min\{\mathrm{UB}, -\langle d, x^{01}\rangle\}$ and if $\mathrm{UB} = -\langle d, x^{01}\rangle$, set $(x^c, v^c, \lambda^c) = (x^{01}, v^{01}, \lambda^{01})$.

Step 0.5. If $|\mathrm{UB} - \mathrm{LB}_0| \leqq |\mathrm{UB}|(\mathrm{PCNT}/100)$, then STOP: (x^c, v^c, λ^c) is an optimal solution for problem (QI_k), (x^c, λ^c) is an optimal solution for problem (PI_k), and $\theta_{k+1} = -\mathrm{UB}$. Otherwise, set $x^0 = x^{01}$, $v^0 = v^{01}$, $\lambda^0 = \lambda^{01}$, $(V^0 \times L^0) = (V^{01} \times L^{01})$, and $w = 1$, and go to Iteration w.

Iteration w, $w \geqq 1$. At the beginning of Iteration w, the following data is available:

(i) A partition $\boldsymbol{M}_{w-1}$ of a subset of $V \times L$ still of interest;

(ii) For each $(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}) \in \boldsymbol{M}_{w-1}$, $0 \leqq \hat{s} \leqq w-1$, $\hat{t} = 1$ or 2, a lower bound $\beta(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}})$ for the optimal value of problem $(\mathrm{QI}_k^{\hat{s},\hat{t}})$, calculated by constructing and solving a linear program $(P_k^{s,t})$ for some s and t such that $0 \leqq s \leqq \hat{s}$, $t = 1$ or 2. Either $\beta(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}) = -\langle d, x^{s,t}\rangle$, where $(x^{s,t}, v^{s,t}, \lambda^{s,t})$ is an optimal solution for linear program $(P_k^{s,t})$, or $\beta(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}) = +\infty$, indicating that both problems $(P_k^{\hat{s},\hat{t}})$ and $(\mathrm{QI}_k^{\hat{s},\hat{t}})$ are infeasible;

(iii) Current lower and upper bounds LB_{w-1} and UB satisfying $\mathrm{LB}_{w-1} \leqq -\theta_{k+1} \leqq \mathrm{UB}$;

(iv) A feasible solution (x^c, v^c, λ^c) for problem (QI_k) satisfying $-\langle d, x^c\rangle = \mathrm{UB}$;

(v) An element $V^{w-1} \times L^{w-1}$ of $\boldsymbol{M}_{w-1}$ for which $\mathrm{LB}_{w-1} = \beta(V^{w-1} \times L^{w-1}) = -\langle d, x^{w-1}\rangle$, where, for some $\hat{s}$ and $\hat{t}$ satisfying $0 \leqq \hat{s} \leqq w-1$, $\hat{t} = 1$ or 2, $V^{w-1} \times L^{w-1} = V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}$ and $(x^{st}, v^{st}, \lambda^{st}) = (x^{w-1}, v^{w-1}, \lambda^{w-1})$ is an

optimal solution found for some linear program $(P_k^{s,t})$, where $0 \leqq s \leqq \hat{s}$ and $t = 1$ or 2.

Step w.1 (*Deletion*). Delete from $\boldsymbol{M}_{w-1}$ all elements $(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}})$ for which $\beta(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}) \geqq \mathrm{UB}$. Let $\boldsymbol{R}_w$ denote the collection of remaining members of $\boldsymbol{M}_{w-1}$, and let $(V^w \times L^w)$ denote $(V^{w-1} \times L^{w-1})$.

Step w.2 (*Branching*). Find an edge of maximum length of the $2p$-rectangle $V^w \times L^w$. Let $\hat{t}$ be the index of any sub-two-rectangle $V_i^w \times L_i^w$, $i = 1, 2, \ldots, p$ of $V^w \times L^w$ which contains this edge. Subdivide $V_{\hat{t}}^w \times L_{\hat{t}}^w$ into two two-rectangles $(V_{\hat{t}}^{w,\hat{t}} \times L_{\hat{t}}^{w,\hat{t}})$, $\hat{t} = 1, 2$, by bisection. Let $V^{w,\hat{t}} \times L^{w,\hat{t}}$, $\hat{t} = 1, 2$, denote the two sub-$2p$-rectangles of $V^w \times L^w$ thereby created. Set $\boldsymbol{M}_w = [\boldsymbol{R}_w \backslash (V^w \times L^w)] \cup [\cup_{\hat{t}=1}^{2} (V^{w,\hat{t}} \times L^{w,\hat{t}})]$.

Step w.3 (*Local Bounding*). For each $\hat{t} = 1, 2$, if the linear program $(P_k^{w,\hat{t}})$ is infeasible, set $\beta(V^{w,\hat{t}} \times L^{w,\hat{t}}) = +\infty$. For each $\hat{t} = 1, 2$ such that the linear program $(P_k^{w,\hat{t}})$ is feasible, solve the linear program for an optimal solution $(x^{w,\hat{t}}, v^{w,\hat{t}}, \lambda^{w,\hat{t}})$ and *set* $\beta(V^{w,\hat{t}} \times L^{w,\hat{t}}) = \max\{-\langle d, x^{w,\hat{t}} \rangle, \beta(V^w \times L^w)\}$.

Step w.4 (*Update Incumbent*). For each $\hat{t} = 1, 2$, determine whether or not the constraints (6) of problem (QI_k) are satisfied with $(v, \lambda) = (v^{w,\hat{t}}, \lambda^{w,\hat{t}})$. Let $\mathrm{TF} \subseteq \{1, 2\}$ be the set of values of $\hat{t}$ for which $(v^{w,\hat{t}}, \lambda^{w,\hat{t}})$ satisfies (6). If TF is empty, go to Step *w*.5. If TF is nonempty, set $\mathrm{UB} = \min\{\mathrm{UB}, \mathrm{UB}'\}$, where $\mathrm{UB}' = \min\{-\langle d, x^{w,\hat{t}} \rangle \mid \hat{t} \in \mathrm{TF}\}$. If $\mathrm{UB} = -\langle d, x^{w,\bar{t}} \rangle$ for some $\bar{t} \in \mathrm{TF}$, choose any such $\bar{t} \in \mathrm{TF}$ and set $(x^c, v^c, \lambda^c) = (x^{w\bar{t}}, v^{w,\bar{t}}, \lambda^{w,\bar{t}})$.

Step w.5 (*Global Bounding*). Let $\mathrm{LB}_w = \min\{\beta(V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}) \mid (V^{\hat{s},\hat{t}} \times V^{\hat{s},\hat{t}}) \in \boldsymbol{M}_w\}$. Let $(V^w \times L^w)$ be any element of $\boldsymbol{M}_w$ for which $\beta(V^w \times L^w) = \mathrm{LB}_w$, and let (x^w, v^w, λ^w) satisfy $-\langle d, x^w \rangle = \beta(V^w \times L^w)$, where, for some $\hat{s}$ and $\hat{t}$ such that $0 \leqq \hat{s} \leqq w$, $\hat{t} = 1$ or 2, $V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}} = V^w \times L^w$ and $(x^{\hat{s},\hat{t}}, v^{\hat{s},\hat{t}}, \lambda^{\hat{s},\hat{t}}) = (x^w, v^w, \lambda^w)$ is an optimal solution found for a linear program $(P_k^{s,t})$, where $0 \leqq s \leqq \hat{s}$, $t = 1$ or 2.

Step w.6 (*Termination Test*). If $|\mathrm{UB} - \mathrm{LB}_w| \leqq |\mathrm{UB}|(\mathrm{PCNT}/100)$, then STOP: (x^c, v^c, λ^c) is an optimal solution for problem(QI_k), (x^c, λ^c) is an optimal solution for problem (PI_k), and $\theta_{k+1} = -\mathrm{UB}$. Otherwise, set $w = w + 1$ and go to Iteration *w*.

In Steps 0.5 and *w*.6, $w \geqq 1$, (x^c, v^c, λ^c) and (x^c, λ^c) are actually approximately-optimal solutions in the sense that they are feasible solutions for which $(\theta_{k+1} - \langle d, x^c \rangle) \leqq (\mathrm{PCNT}/100)|\langle d, x^c \rangle|$. This inequality follows from Steps 0.5 and *w*.6, $w \geqq 1$. Since PCNT is a parameter chosen from [0, 100], the approximation can be as precise as desired.

To reduce computational bookkeeping requirements, the Deletion Step *w*.1 is included in the procedure. In this step, elements of the partition $\boldsymbol{M}_{w-1}$ are removed from further consideration that cannot contain vectors $(v, \lambda) \in V \times L$ which constitute a portion of an optimal solution to problem (QI_k).

Notice that the procedure can be executed by using linear programming methods. Therefore, all of the steps, including Step *k*.1, of the Relaxation

Algorithm for Problem (P) given in Section 2, can be implemented by solving linear programming problems.

It must be added that the procedure for executing Step $k.1$ may or may not require a finite number of iterations. The following theorem guarantees the validity of the procedure in the infinite case. The interested reader is referred to the Appendix for a proof of this theorem.

THEOREM 3.1. *Suppose that the Branch and Bound Procedure for Iteration Step k.1 does not terminate after a finite number of iterations. Then*

(1) $\lim_{w\to\infty} LB_w = -\theta_{k+1}$;

(2) *Any accumulation point of the sequence* $\{(x^w, v^w, \lambda^w)\}$ *is an optimal solution for problem* (QI_k).

4. A Small Example

To illustrate the suggested implementation of the Relaxation Algorithm and some of its properties, consider a realization of problem (P) in which

(a) $X = \{(x_1, x_2, x_3) \in R^3 \mid 2x_1 + x_2 \leqq 16,\ 8x_1 + 5x_2 \leqq 66,\ 2x_1 + 3x_2 \leqq 27,\ x_1 \geqq 0,\ 0 \leqq x_2 \leqq 7,\ 0 \leqq x_3 \leqq 2\}$;

(b) $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$;

(c) $d^T = (-1, 0, 1)$.

The sets X and X_E are shown in Figure 1, where X_E consists of the three two-dimensional faces of X which are shaded. Table I lists the extreme points of X. The maximum value of $\langle d, x\rangle$ over X equals two and is achieved at the extreme points A and B, neither of which belong to X_E. Therefore, the requirement $x \in X_E$ in problem (P) is *essential* in this example in the sense explained in Section 2. Let $\Lambda = \{(\lambda_1, \lambda_2) \in R^2 \mid \lambda_1 + \lambda_2 \leqq 20;\ \lambda_1, \lambda_2 \geqq 1\}$.

Initialization Step. We choose $\bar{\lambda}^0 = (9.0, 5.0)^T$. Solving the linear program (P_λ) with $\lambda = \bar{\lambda}^0$ for an optimal extreme point solution, we obtain the efficient extreme point $x^0 = (7.0, 2.0, 0.0)$. Set $k = 0$.

Iteration 0.

Step 0.1. Using the Branch and Bound Procedure given in Section 3, we find via a sequence of linear programs that an (exact) optimal solution to the relaxed problem (PI_0) is given by $(\bar{x}^1, \bar{\lambda}^1) = [(0.00, 4.385967, 2.00)^T, (1.0, 9.5)^T]$ with $\theta_1 = 2.0$. (Notice by Theorem 2.1 and Remark 2.1 that $\langle d, x^0\rangle = -7.0$ and $\theta_1 = 2.0$ are lower and upper bounds, respectively, for the optimal value θ of problem (P). Also notice that, as guaranteed by Theorem 2.2, $\bar{x}^1 \in H_E$, where H is the convex hull of $\{x^0, \bar{x}^1\}$.)

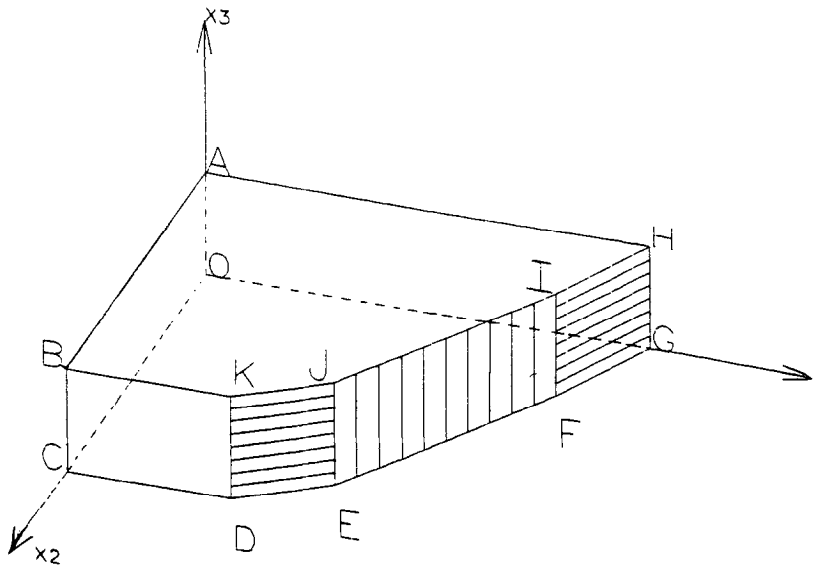

Fig. 1. The sets X and X_E.

Step 0.2. Since $\theta_1 = 2.0$ and $\langle d, x^0 \rangle = -7.0 \neq 2.0$, we continue.

Step 0.3. Solving the linear program (T_0), we find that an optimal solution is given by $x^* = (4.5, 6.0, 0.0)^T$ with $e^T C x^* = 10.50$. Since $e^T C \bar{x}^1 \neq 10.50$, we continue. (The point $\bar{x}^1$ is not an element of X_E.)

Step 0.4. Solving the linear program (P_λ) with $\lambda = \bar{\lambda}^1 = (1.0, 9.5)^T$ for an extreme point optimal solution, we obtain the optimal solution $x^{**} = (4.5, 6.0, 0.0)^T$. We set $x^1 = (4.5, 6.0, 0.0)^T$. (Then $x^1 \in X_E \cap X_{ex}$. Notice that $x^1 \neq x^0$, as guaranteed by Theorem 2.3.). We then set $k = 1$ and proceed to Iteration 1.

Table I. Extreme Points of X

Point	Coordinates
A	(0, 0, 2)
B	(0, 7, 2)
C	(0, 7, 0)
D	(3, 7, 0)
E	(4½, 6, 0)
F	(7, 2, 0)
G	(8, 0, 0)
H	(8, 0, 2)
I	(7, 2, 2)
J	(4½, 6, 2)
K	(3, 7, 2)
O	(0, 0, 0)

Iteration 1.

Step 1.1. Using the Branch and Bound Procedure given in Section 3, we find by solving only linear programs that the (exact) optimal solution for the relaxed problem (PI_1) is $(\bar{x}^2, \bar{\lambda}^2) = [(0.00, 6.2368421, 2.00)^T, (1.0, 19.0)^T]$, and we find that $\theta_2 = 2.0$. (Then by Theorem 2.1 and Remark 2.1, $\max\{\langle d, x^0\rangle, \langle d, x^1\rangle\} = -4.50 \leqq \theta \leqq 2.0$. Notice that $\bar{x}^2 \in H_E$, where H is the convex hull of $\{x^0, x^1, \bar{x}^2\}$, as guaranteed by Theorem 2.2.)

Step 1.2. Since $\theta_2 = 2.0$ and neither $\langle d, x^0\rangle$ nor $\langle d, x^1\rangle$ equals 2.0, we continue.

Step 1.3. Solving the linear program (T_1), we obtain an optimal solution $x^* = (4.1447368, 6.2368421, 0.00)^T$ with $e^T C x^* = 10.38158$. Since $e^T C \bar{x}^2 \neq 10.38158$, we continue. (The point $\bar{x}^2$ does not lie in X_E).

Step 1.4. We solve the linear program (P_λ) with $\lambda = \bar{\lambda}^2 = (1.0, 19.0)^T$ for any extreme point optimal solution and obtain $x^{**} = (3.0, 7.0, 0.0)^T$. We set $x^2 = (3.0, 7.0, 0.0)^T$, and $k = 2$ and proceed to Iteration 2. (Notice that $x^2 \in X_E \cap X_{ex}$ and, as guaranteed by Theorem 2.3, $x^2 \neq x^0$ and $x^2 \neq x^1$.)

Iteration 2.

Step 2.1. From the Branch and Bound Procedure given in Section 3, we find the (exact) optimal solution $(\bar{x}^3, \bar{\lambda}^3) = [(3.0, 7.0, 2.0)^T, (1.0, 1.5)^T]$ for the relaxed problem (PI_2) by solving a sequence of linear programs. The optimal value for problem (PI_2) is $\theta_3 = -1.0$. (Then $\max\{\langle d, x^0\rangle, \langle d, x^1\rangle, \langle d, x^2\rangle\} = -3 \leqq \theta \leqq -1.0$. Notice that $\bar{x}_3 \in H_E$, where H is the convex hull of $\{x^0, x^1, x^2, \bar{x}^3\}$, as guaranteed by Theorem 2.2.)

Step 2.2. Since $\theta_3 = -1.0$ and none of the numbers $\langle d, x^j\rangle$, $j = 0, 1, 2$, equals -1.0, we continue.

Step 2.3. Solving the linear program (T_2), we obtain an optimal solution $x^* = (3.0, 7.0, 0.0)^T$ with $e^T C x^* = 10.0$. Since $e^T C \bar{x}^3 = 10.0$, we STOP: The point $\bar{x}^3 = (3.0, 7.0, 2.0)^T$ is an optimal solution for problem (P), and $\theta = -3.0 + 2.0 = -1.0$.

Notice that termination occurs in Step 2.3 of this example by the following logic. For the optimal solution $(\bar{x}^3, \bar{\lambda}^3)$ computed in Step 2.1 for the relaxation (PI_2) of Problem (PI), we found in Step 2.3 that $\bar{x}^3 \in X_E$. Therefore, for some $\bar{\lambda} \in \Lambda$, $(\bar{x}^3, \bar{\lambda})$ is a feasible solution for Problem (PI). Since $(\bar{x}^3, \bar{\lambda})$ is an optimal solution for the relaxation (PI_2) of problem (PI) and is feasible in problem (PI), it must, in fact, be optimal in problem (PI) as well. This implies, by the equivalence of problems (PI) and (P), that $\bar{x}^3$ is an optimal solution for problem (P).

5. Concluding Remarks

This paper has presented the first readily-implementable algorithm for finding a globally optimal solution to the problem (P) of optimizing a linear function over the efficient set X_E of a multiple objective linear program. As a special case, the

algorithm can be applied to the problem (Q) of minimizing any individual criterion of the multiple objective linear program over X_E. We have shown several properties of the algorithm, including the following:

(1) It finds an exact, globally optimal solution for problem (P) after a finite number of iterations.
(2) Depending upon at which step of an iteration the algorithm terminates, either an extreme point or a non-extreme point (globally) optimal solution is found.
(3) The algorithm can be validly implemented using only linear programming methods.
(4) Each successive iteration of the algorithm yields lower and upper bounds for the optimal value of problem (P) which are at least as good as the lower and upper bounds available from the previous iteration. Thus, for instance, the algorithm can be prematurely terminated when an incumbent solution is found which, although possibly suboptimal, achieves an objective function value suitably close to the optimal value.

Since problems (P) and (Q) have many important applications in multiple criteria decision making, these properties imply that the algorithm represents a potentially valuable, practical tool for aiding decision makers faced with multiple objective problems.

6. Appendix: Proof of Theorem 3.1

To prove Theorem 3.1, three preliminary results must first be shown. The first states, in the sense of Horst [17, 19], that the use of bisection in the branch and bound procedure for Step $k.1$ is *exhaustive*.

LEMMA 6.1. *Let* $\{(V^u \times L^u)\}_{u=1}^{\infty}$ *be an infinite nested sequence of partition elements created by the algorithm. Then, for some point* $(\tilde{v}, \tilde{\lambda}) \in R^{2p}$, $\lim_{u\to\infty}(V^u \times L^u) = \cap_{u=1}^{\infty}(V^u \times L^u) = \{(\tilde{v}, \tilde{\lambda})\}$.

Proof. Let $\{(V^u \times L^u)\}_{u=1}^{\infty}$ be an infinite nested sequence of partition elements created by the algorithm. Then, from Definition 3.3 and the Branching Step the algorithm, $\{(V^u \times L^u)\}_{u=1}^{\infty}$ is a subsequence of some sequence of $2p$-rectangles $\{(V^w \times L^w)\}_{u=1}^{\infty}$ for which, for each w, $(V^{w+1} \times L^{w+1})$ is a sub-$2p$-rectangle of $(V^w \times L^w)$ created by bisection. From Horst [19], $\lim_{w\to\infty}(V^w \times L^w) = \cap_{w=1}^{\infty}(V^w \times L^w) = \{(\tilde{v}, \tilde{\lambda})\}$ for some $(\tilde{v}, \tilde{\lambda}) \in R^{2p}$. Therefore, $\lim_{u\to\infty}(V^u \times L^u) = \cap_{u=1}^{\infty}(V^u \times L^u) = \{(\tilde{v}, \tilde{\lambda})\}$ as well. □

LEMMA 6.2. *Let* $\{(V^u \times L^u)\}_{u=1}^{\infty}$ *be an infinite nested sequence of partition elements created by the algorithm. Let* $\tilde{x}$ *be any accumulation point of* $\{x^u\}_{u=1}^{\infty}$, *where, for each* u, $-\langle d, x^u\rangle = \beta(V^u \times L^u)$. *Then* $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ *is a feasible solution for problem* (QI_k), *where* $\{(\tilde{v}, \tilde{\lambda})\} = \lim_{u\to\infty}(V^u \times L^u)$.

Proof. From Lemma 6.1 $\{(\tilde{v}, \tilde{\lambda})\} = \lim_{u\to\infty}(V^u \times L^u)$ exists. Furthermore,

from the Branching and Global Bounding Steps of the algorithm, for each u, since $\{(V^u \times L^u)\}_{u=1}^{\infty}$ is an infinite sequence, $\beta(V^u \times L^u) = -\langle d, x^u \rangle$, where for some $\hat{s}$ and $\hat{t}$ satisfying $0 \leqq \hat{s} \leqq u$, $\hat{t} = 1$ or 2, $V^u \times L^u = V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}$ and (x^u, v^u, λ^u) is an optimal solution found by the algorithm for the linear program $(P_k^{s,t})$ for some s and t satisfying $0 \leqq s \leqq \hat{s}$, $t = 1$ or 2.

For each u, let (P_k^u) denote the linear program $(P_k^{s,t})$ with V^u and L^u replacing $V^{s,t}$ and $L^{s,t}$, respectively, in (11) and (12), and with $\underline{\lambda}_i^u$, $\bar{\lambda}_i^u$, $\underline{v}_i^u$ and $\bar{v}_i^u$ replacing $\underline{\lambda}_i^{s,t}$, $\bar{\lambda}_i^{s,t}$, $\underline{v}_i^{s,t}$ and $\bar{v}_i^{s,t}$, respectively, in (9). Assume for each u that

$$V^u = \{v \in R^p \mid \underline{v}^u \leqq v \leqq \bar{v}^u\}$$

and

$$L^u = \{\lambda \in R^p \mid \underline{\lambda}^u \leqq \lambda \leqq \bar{\lambda}^u\},$$

where $\underline{v}^u$, $\bar{v}^u$, $\underline{\lambda}^u$, $\bar{\lambda}^u \in R^p$.

Since $\lim_{u\to\infty}(V^u \times L^u) = \{(\tilde{v}, \tilde{\lambda})\}$, $\lim_{u\to\infty} \underline{v}^u = \lim_{u\to\infty} \bar{v}^u = \tilde{v}$ and $\lim_{u\to\infty} \underline{\lambda}^u = \lim_{u\to\infty} \bar{\lambda}^u = \tilde{\lambda}$. This implies, since $\underline{v}^u \leqq v^u \leqq \bar{v}^u$ and $\underline{\lambda}^u \leqq \lambda^u \leqq \bar{\lambda}^u$ for each u, that $\lim_{u\to\infty} v^u = \tilde{v}$ and $\lim_{u\to\infty} \lambda^u = \tilde{\lambda}$. For each u, since (x^u, v^u, λ^u) is an optimal solution for problem (P_k^u), (x^u, v^u, λ^u) satisfies the constraints (9) and (10) of this problem. Setting $v_i = v_i^u$ and $\lambda_i = \lambda_i^u$, $i = 1, 2, \ldots, p$, in the constraints (9) and taking limits as u approaches infinity on each side of each constraint in (9), we obtain, using the observations in the previous two sentences,

$$\langle \tilde{v}, \tilde{\lambda} \rangle + \langle a^j, \tilde{\lambda} \rangle \leqq 0, \quad j = 0, 1, \ldots, k. \tag{13}$$

Since X is a compact set and, from (10), $x^u \in X$ for all u, if we choose any accumulation point $\tilde{x}$ of $\{x^u\}_{u=1}^{\infty}$, then $\tilde{x} \in X$. For simplicity of notation, assume without loss of generality that $\lim_{u\to\infty} x^u = \tilde{x}$. Since $\lim_{u\to\infty}(v^u, \lambda^u) = (\tilde{v}, \tilde{\lambda})$, and, from (10), $(x^u, v^u, \lambda^u) \in Q$ for each u, it is easily seen from the definition of Q that $(\tilde{x}, \tilde{v}, \tilde{\lambda}) \in Q$. Finally, since $(\tilde{v}, \tilde{\lambda}) \in (V^u \times L^u) \subseteq (V \times L)$ for each u, $(\tilde{v}, \tilde{\lambda}) \in V \times L$. From (13) and the previous two statements, $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ is a feasible solution for problem (QI_k). □

LEMMA 6.3. *Let $\{(V^u \times L^u)\}_{u=1}^{\infty}$ be an infinite nested sequence of partition elements created by the algorithm with $\lim_{u\to\infty}(V^u \times L^u) = \cap_{u=1}^{\infty}(V^u \times L^u) = \{(\tilde{v}, \tilde{\lambda})\}$. Let $\tilde{x}$ be any accumulation point of $\{x^u\}_{u=1}^{\infty}$, where, for each u, $-\langle d, x^u \rangle = \beta(V^u \times L^u)$. Then there exists a subsequence $\{(V^{u'} \times L^{u'})\}$ of $\{(V^u \times L^u)\}_{u=1}^{\infty}$ such that*

$$\lim_{u'\to\infty} \beta(V^{u'} \times L^{u'}) = -\langle d, \tilde{x} \rangle.$$

Proof. Let $\{(V^u \times L^u)\}_{u=1}^{\infty}$ be an infinite nested sequence of partition elements created by the algorithm with $\lim_{u\to\infty}(V^u \times L^u) = \cap_{u=1}^{\infty}(V^u \times L^u) = \{(\tilde{v}, \tilde{\lambda})\}$. From the proof of Lemma 6.2, for each u, $\beta(V^u \times L^u) = -\langle d, x^u \rangle$ is the (finite) optimal value of the linear program (P_k^u) defined in that proof. From the Local and Global Bounding Steps of the algorithm, $\beta(V^u \times L^u) \leqq \beta(V^{u+1} \times$

$L^{u+1}) < +\infty$ for all u. From elementary calculus this implies that $\lim_{u\to\infty} \beta(V^u \times L^u)$ exists. Let $\bar{\beta}$ denote this limit.

Now let $\tilde{x}$ be any accumulation point of $\{x^u\}_{u=1}^{\infty}$, where, for each u, $-\langle d, x^u\rangle = \beta(V^u \times L^u)$. Assume that $\lim_{u'\to\infty} x^{u'} = \tilde{x}$. Then $\bar{\beta} = \lim_{u'\to\infty} \beta(V^{u'} \times L^{u'}) = \lim_{u'\to\infty} -\langle d, x^{u'}\rangle = -\langle d, \tilde{x}\rangle$. □

Proof of Theorem 3.1. Suppose that the Branch and Bound Procedure for Iteration Step $k.1$ does not terminate after a finite number of iterations. Then it generates a sequence of lower bounds $\{\mathrm{LB}_w\}_{w=1}^{\infty}$ for which, from the Global Bounding Step, for each w, $\mathrm{LB}_w = \beta(V^w \times L^w) = -\langle d, x^w\rangle$, where for some $\hat{s}$ and $\hat{t}$ satisfying $0 \leqq \hat{s} \leqq w$, $\hat{t} = 1$ or 2, $V^w \times L^w = V^{\hat{s},\hat{t}} \times L^{\hat{s},\hat{t}}$ and (x^w, v^w, λ^w) is an optimal solution found by the algorithm for some linear program $(P_k^{s,t})$, where $0 \leqq s \leqq \hat{s}$, $t = 1$ or 2. From the construction of the algorithm, $\mathrm{LB}_w \leqq \mathrm{LB}_{w+1} \leqq -\theta_{k+1}$ for all w, so that $\lim_{w\to\infty} \mathrm{LB}_w$ exists and

$$\lim_{w\to\infty} \mathrm{LB}_w \leqq -\theta_{k+1}\,. \tag{14}$$

Let $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ denote any accumulation point of the sequence $\{(x^w, v^w, \lambda^w)\}_{w=1}^{\infty}$, and let $\{(x^{w''}, v^{w''}, \lambda^{w''})\}$ denote the corresponding subsequence of $\{(x^w, v^w, \lambda^w)\}_{w=1}^{\infty}$ such that $\lim_{w''} (x^{w''}, v^{w''}, \lambda^{w''}) = (\tilde{x}, \tilde{v}, \tilde{\lambda})$. Using a standard argument on the finiteness of the number of partition elements in each step of the algorithm (see, for instance, [16, 17]) and Lemma 6.1, there exists a decreasing (nested) subsequence $\{(V^{w''} \times L^{w''})\} \subseteq \{(V^{w'} \times L^{w'})\}$ of successively-refined partition elements such that $\lim_{w'\to\infty} (V^{w'} \times L^{w'}) = \{(\tilde{v}, \tilde{\lambda})\}$. Therefore, since $\tilde{x} = \lim_{w'\to\infty} x^{w'}$ and, for each w', $-\langle d, x^{w'}\rangle = \beta(V^{w'} \times L^{w'})$, by Lemma 6.3 there exists a subsequence $\{(V^{\hat{w}} \times L^{\hat{w}})\}$ of $\{(V^{w'} \times L^{w'})\}$ such that

$$\lim_{\hat{w}\to\infty} \beta(V^{\hat{w}} \times L^{\hat{w}}) = -\langle d, \tilde{x}\rangle\,, \tag{15}$$

and, by Lemma 6.2, $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ is a feasible solution for problem (QI_k). Since $\mathrm{LB}_{\hat{w}} = \beta(V^{\hat{w}} \times L^{\hat{w}})$ for each $\hat{w}$, (14) and (15) imply that

$$\lim_{\hat{w}\to\infty} \mathrm{LB}_{\hat{w}} = -\langle d, \tilde{x}\rangle \leqq -\theta_{k+1}\,. \tag{16}$$

But since $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ is a feasible solution for problem (QI_k), the inequality in (16) implies that $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ must be an optimal solution for problem (QI_k), thus proving part (2) of the theorem. From (16) and the optimality of $(\tilde{x}, \tilde{v}, \tilde{\lambda})$ in problem (QI_k),

$$\lim_{\hat{w}\to\infty} \mathrm{LB}_{\hat{w}} = -\langle d, \tilde{x}\rangle = -\theta_{k+1}\,. \tag{17}$$

Since $\lim_{w\to\infty} \mathrm{LB}_w = \lim_{\hat{w}\to\infty} \mathrm{LB}_{\hat{w}}$, (17) implies part (1) of the theorem. □

References

1. Al-Khayyal, F. A. and Falk, J. E. (1983), Jointly Constrained Biconvex Programming, *Mathematics of Operations Research* **8**, 273–286.

2. Belenson, S. and Kapur, K. C. (1973), An Algorithm for Solving Multicriterion Linear Programming Problems with Examples, *Operational Research Quarterly* **24**, 65–77.
3. Benayoun, R., de Montgolfier, J., Tergny, J., and Laritchev, O. (1971), Linear Programming with Multiple Objective Functions: Step Method (STEM), *Mathematical Programming* **1**, 366–375.
4. Benson, H. P. (1986), An Algorithm for Optimizing over the Weakly-Efficient Set, *European J. of Operational Research* **25**, 192–199.
5. Benson, H. P. (1978), Existence of Efficient Solutions for Vector Maximization Problems, *J. of Optimization Theory and Applications* **26**, 569–580.
6. Benson, H. P. (1982), On the Convergence of Two Branch-and-Bound Algorithms for Nonconvex Programming Problems, *J. of Optimization Theory and Applications* **36**, 129–134.
7. Benson, H. P. (1984), Optimization over the Efficient Set, *J. of Mathematical Analysis and Applications* **98**, 562–580.
8. Benson, H. P. and Horst, R. (forthcoming), A Branch and Bound-Outer Approximation Algorithm for Concave Minimization over a Convex Set, *Computers and Mathematics with Applications*.
9. Blankenship, J. W. and Falk, J. E. (1976), Infinitely Constrained Optimization Problems, *J. of Optimization Theory and Applications* **19**, 261–281.
10. Dessouky, M. I., Ghiassi, M., and Davis, W. J. (1986), Estimates of the Minimum Nondominated Criterion Values in Multiple-Criteria Decision-Making, *Engineering Costs and Production Economics* **10**, 95–104.
11. Evans, G. W. (1984), An Overview of Techniques for Solving Multiobjective Mathematical Programs, *Management Science* **30**, 1268–1282.
12. Falk, J. E. and Soland, R. M. (1969), An Algorithm for Separable Nonconvex Programming Problems, *Management Science* **15**, 550–569.
13. Ghiassi, M., DeVor, R. E., Dessouky, M. I. and Kijowski, B. A. (1984), An Application of Multiple Criteria Decision Making Principles for Planning Machining Operations, *IEEE Transactions* **16**, 106–114.
14. Hansen, P. (ed.) (1983), *Essays and Surveys on Multiple Criteria Decision Making*, Springer-Verlag, Berlin.
15. Hemming, T. (1978), *Multiobjective Decision Making Under Certainty*, Economic Research Institute of the Stockholm School of Economics, Stockholm.
16. Horst, R. (1976), An Algorithm for Nonconvex Programming Problems, *Mathematical Programming* **10**, 312–321.
17. Horst, R. (1988), Deterministic Global Optimization with Partition Sets Whose Feasibility is Not Known. Appliction to Concave Minimization, Reverse Convex Constraints, D.C.-Programming and Lipschitzian Optimization, *J. of Optimization Theory and Applications* **58**, 11–37.
18. Horst, R. (1990), Deterministic Global Optimization: Recent Advances and New Fields of Application, *Naval Research Logistics* **37**, 433–471.
19. Horst, R. (1986), A General Class of Branch and Bound Methods in Global Optimization with Some New Approaches for Concave Minimization, *J. of Optimization Theory and Applications* **51**, 271–291.
20. Horst, R., Thoai, N. V. and Benson, H. P. (forthcoming), Concave Minimization via Conical Partitions and Polyhedral Outer Approximation, *Mathematical Programming*.
21. Horst, R. and Tuy, H. (1990), *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin.
22. Isermann, H. and Steuer, R. E. (1987), Computational Experience Concerning Payoff Tables and Minimum Criterion Values over the Efficient Set, *European J. of Operational Research* **33**, 91–97.
23. Kok, M. and Lootsma, F. A. (1985), Pairwise-Comparison Methods in Multiple Objective Programming, with Applications in a Long-Term Energy-Planning Model, *European J. of Operational Research* **22**, 44–55.
24. McCormick, G. P. (1976), Computability of Global Solutions to Factorable Nonconvex Programs: Part I – Convex Underestimating Problems, *Mathematical Programming* **10**, 147–175.
25. Pardalos, P. M. and Rosen, J. B. (1982), *Constrained Global Optimization: Algorithms and Applications*, Springer-Verlag, Berlin.
26. Pardalos, P. M. and Rosen, J. B. (1986), Methods for Global Concave Minimization: A Bibliographic Survey, *SLAM Review* **28**, 367–379.

27. Philip, J. (1972), Algorithms for the Vector Maximization Problem, *Mathematical Programming* **2**, 207–229.
28. Reeves, G. R. and Reid, R. C. (1988), Minimum Values over the Efficient Set in Multiple Objective Decision Making, *European J. of Operational Research* **36**, 334–338.
29. Rosenthal, R. E. (1985), Principles of Multiobjective Optimization, *Decision Sciences* **16**, 133–152.
30. Steuer, R. E. (1986), *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York.
31. Thoai, N. V. and Tuy, H. (1980), Convergent Algorithms for Minimizing a Concave Function, *Mathematics of Operations Research* **5**, 556–566.
32. Weistroffer, H. R. (1985), Careful Usage of Pessimistic Values is Needed in Multiple Objectives Optimization, *Operations Research Letters* **4**, 23–25.
33. Yu, P. L. (1985), *Multiple-Criteria Decision Making*, Plenum, New York.
34. Zeleny, M. (1982), *Multiple Criteria Decision Making*, McGraw-Hill, New York.