Vietnam Journal of Mathematics
Volume 23, Number 1, 1995

COMPUTATIONAL ASPECTS OF OPTIMIZATION PROBLEMS OVER THE EFFICIENT SET [1]

LE DUNG MUU

Abstract. *We give an introduction to computational methods for optimizing a real valued function over the efficient set of a vector optimization problem. This problem has many applications in multiple criteria decision making. Some recently developed algorithms for solving this problem are described in a unified manner. Questions for further development of this problem are discussed.*

1. INTRODUCTION AND THE PROBLEM STATEMENT

The purpose of this paper is to give some computational aspects of optimization problems over the efficient set of a vector maximization linear problem. This problem is shown to be equivalent to a linear program with a special additional reverse convex constraint. Some recently developed methods for solving this problem are described in a unified manner.

Throughout the paper for two vectors $a := (a_1, \ldots, a_k)$, $b := (b_1, \ldots, b_k)$ we write $a \geqq b$ (respect. $a \geq b$) if and only if $a_i \geq b_i$ (respect. $a_i \geq b_i$ and $a \neq b$) for all i. Also, $a > b$ if and only if $a_i > b_i$ for all i. For simplicity of notations we will omit the notation of transpose in writing the inner product of two vectors.

Let $X \subset R^n$ be a polyhedral convex set given by linear inequalities and/or equalities. Let C be a $(p \times n)$ real matrix. Then the vector (or multiple) objective linear programming problem written as

$$\max\{Cx : x \in X\} \tag{VP}$$

[1] This paper is supported in part by the National Basic Research Program in Natural Sciences

can be viewed as the problem of finding all efficient solutions to Problem (VP). We recall that a point $x^0 \in X$ is said to be an *efficient solution* of Problem (VP) if whenever $Cx \geqq Cx^0$ for some $x \in X$, then $Cx = Cx^0$. An efficient solution is also often called a *nondominated* or *Pareto* point. Each component of the vector Cx is called the *objective function* or the *criterion function*. We shall denote by X_E the set of all efficient solutions of Problem (VP).

The problem of main concern in this paper then can be written as

$$\max\{dx : x \in X_E\}, \tag{P}$$

where $d \in R^n$ is given.

Problem (P) can be classified as a global optimization problem, since its feasible region X_E is, in general, nonconvex. In [28] Philip first proposed Problem (P) and outlined a cutting plane algorithm for finding a global optimum of this problem. In 1979 Dessouky et al [8] considered a special case of this problem where the objective function is one of the criterion functions of the vector maximization linear problem. They used Philip's algorithm for solving this case. In [13] Isermann and Steuer developed an algorithm for solving (P). Like Philip's method, this algorithm used simplex-type pivots to move along paths of adjacent efficient extreme points which increase the value of the linear objective function. A cutting plane is added each time and a locally optimal efficient extreme point is thereby found. These methods require finding all efficient extreme points that lie on this cutting plane in the newly created polyhedron. Since neither of these algorithms explains how to perform this search mathematically, it is not clear how to implement them.

Benson [2] established necessary and sufficient conditions for efficiency to a vector convex optimization problem, and for an efficient point to be a maximal point of a linear function over the efficient set of the underlying vector optimization problem. Using these results, recently Benson [3], [4], [5] proposed three algorithms for solving Problem (P). The methods in [3], [4] are based on the fact that one can find a simplex $\Lambda \subset R^p$ such that (P) is equivalent to the following infinitely constrained optimization problem given by

$$\max dx \tag{IP}$$

subject to

$$\lambda Cx \geqq \lambda Cy \quad \forall y \in X, \quad x \in X, \quad \lambda \in \Lambda.$$

At each iteration k of the methods in [3], [4] the infinitely constrained problem (IP) is relaxed by the following finite constrained optimization problem:

$$\max dx \tag{Pk}$$

subject to

$$\lambda Cx \geqq \lambda Cx^i \quad (i = 0, 1, \ldots, k), \quad x \in X, \quad \lambda \in \Lambda.$$

For solving this nonconvex Subproblem, in the first algorithm [3], called all-linear programming method, the term λCx is replaced by its convex envelope, and then this subproblem is solved by an infinitely convergent procedure. In the second method [4], called nonadjacent extreme point search, the request of solving Subproblem (Pk) at iteration k is weakened by finding an efficient extreme point x^{k+1} such that dx^{k+1} is greater than the currently known lower bound of the optimal value of (P). The third algorithm [5], called bisection extreme point search method, is proposed for solving a special case of (P) where the vector d is linearly dependent upon the rows of the matrix C. In this algorithm, Problem (P) is shown to be equivalent to a finite number of concave minimization problems in x-space. A heuristic algorithm is also proposed recently in [6].

In Muu [20] Problem (IP) is replaced by a convex-concave programming problem given by

$$\max dx \tag{CP}$$

subject to

$$h(\lambda, x) \leq 0, \quad x \in X, \quad \lambda \in \Lambda,$$

where $h(\lambda, x)$ is a convex - concave function defined by

$$h(\lambda, x) := \max\{\lambda Cy : y \in X\} - \lambda Cx.$$

This problem is a special case of problem considered in [17], and therefore it could be solved by a decomposition branch-and-bound algorithm described there. This algorithm however would be efficient only if the dimension of the "concave-space" where subdivisions were performed is relatively small. However, in many applications the dimension of x-space in Problem (P) is much greater than that of λ-space which just equals the number of the criteria. In [20], some methods are developed for solving Problem (CP) by linearization of the convex function g given by

$$g(\lambda) := \max\{\lambda C y : y \in X\}.$$

The subdivisions in these methods are performed only in λ - space. Therefore they are efficient if the number of the criteria is relatively small while the number of the variables may be fairly large. In [21] an efficient parametric simplex method is described for solving Problem (P) with $p = 2$. This case frequently arises in many applications, especially in bimatrix game theory.

The paper is organized as follows. In the next section we present some theoretical results on optimality conditions for Problem (P) that will be used for deriving solution methods described in the following sections. Section 3 is devoted to describe three methods based on relaxation approach for solving (P). In Section 4 we outline algorithms which use efficient extreme point search for finding a global solution to (P) among the vertices of the polytope X. In Section 5 we consider the bicriteria problem which can be solved very efficiently by a parametric simplex method given there. Some other results for special cases of the bicriteria case are also presented in this section. We close the paper by conclusions in Section 6.

2. CONDITIONS FOR OPTIMALITY

In this section we shall be concerned with optimality conditions which will be used for deriving solution methods to Problem (P). The proofs of theorems in this section can be found in [2], [28]. We assume additionally that X is bounded. Then both the vertex set of X, which we denote henceforth by $V(X)$, and the efficient set X_E are nonempty, (see e.g. [29] and [2], [28]). Furthermore Problem (P) has an optimal solution. Let $P(X_E)$ denote the optimal value of (P).

Theorem 2.1. *Problem* (P) *attains its maximum at a vertex of* X.

Let C' be the $(p+1) \times n$ matrix whose first p rows are identical to those of matrix C and $(p+1)$th row is the vector d. Consider the vector maximization problem (VP') given by

$$\max\{C'x : x \in X\} \qquad \text{(VP')}$$

The next theorem says that the solution set of (P) lies in the efficient set of Problem (VP').

Theorem 2.2 ([2]). *If a point x^0 is an optimal solution for Problem* (P), *then x^0 is efficient for Problem* (VP').

Consider the function $r : G \to R$ given by [2]

$$r(x) := \max\{eCy - eCx : Cy \geqq Cx,\ y \in X\}$$

where $G := \{x | Cy \geqq Cx \text{ for some } y \in X\}$, and e stands for the vector whose every entry is unit.

Now let t be a real fixed number, and consider the Problem (Ut) given by

$$\min\{r(x),\ dx \geqq t,\ x \in X\}. \tag{Ut}$$

Let r_t denote the optimal value of this problem. As usual for the convention we take $r_t = +\infty$ if the feasible set of this problem is empty. The following result uses Problem (Ut) to characterize optimal solution for Problem (P).

Theorem 2.3 ([2]). *The optimal value of Problem* (P) *is equal to t^*, where t^* is the largest value of t in Problem* (Ut) *such that $r_t = 0$. Furthermore x^* is an optimal solution for* (P) *if and only if it is an optimal solution for* (Ut*).

By considering the dual linear programming problem of the linear program defining r, we immediately obtain the following corollary of Theorem 2.3.

Corollary 2.1 ([2]). *Let A be a $m \times n$ matrix and let $b \in R^m$. Assume that $X := \{x_m \in R^n : Ax \leqq b,\ x \geqq 0\}$. Then $P(X_E) = t^*$, where t^* is the largest value of t in the Problem* (Vt) *given by*

$$v_t := \min\{-yCx + bu - eCx\} \tag{Vt}$$

subject to

$$dx \geqq t, \quad Ax \leqq b, \quad -yC + uA \geqq eC, \quad (x, y, u) \geqq 0$$

such that $v_t = 0$. Furthermore, x^0 is an optimal solution for Problem (P) *if and only if for some $y^0 \in R^p$ and $u^0 \in R^m$, (x^0, y^0, u^0) is an optimal solution for Problem* (Vt*).

Theorem 2.4. *Problem* (P) *is equivalent to the linear program with an additional reverse convex constraint given as*

$$\max\{dx : x \in X,\ r(x) \leqq 0\}. \tag{RP}$$

Theorem 2.4 implies that solving Problem (P) amounts to solve a linear program with an additional reverse convex constraint. The latter can be solved by some available methods (see e.g. [9], [10], [14], [31]). It is not difficult to show that the function r is constant on any affine space which is parallel to the subspace $H := \{x \in R^n : Cx = 0\}$. If rank $C = k$ then the dimension of H is equal to $(n-k)$. Thus the function r actually depends only on k-variables. This suggests studying methods which employ this property for solving Problem (P) more efficiently.

3. RELAXATION APPROACH TO SOLVING PROBLEM (P)

As mentioned in the previous section, Problem (P) is equivalent to the infinitely constrained Problem (IP). The main difficulty in the latter problem arises from the bilinear constraints. This fact suggests using relaxation approach to solving (P) by using the branch-and-bound technique.

The branch-and-bound is a fundamental technique in nonconvex optimization. Branch-and-bound methods differ on the way they define rules for branching and the methods used for calculating bounds. In this section we shall present two branch-and-bound methods for solving Problem (P).

Note that the infinitely constrained nonconvex optimization Problem (IP) can be written as

$$\max\{dx : x \in X,\ \lambda \in \Lambda,\ g(\lambda) - \lambda Cx \leqq 0\} \tag{IP'}$$

where we recall that

$$g(\lambda) = \max\{\lambda Cy : y \in X\},$$

and Λ is the simplex as in Problem (IP), which can be found by a procedure given in [28]. In branch-and-bound algorithms that we are going to describe the bounds are obtained by certain relaxations of Problem (IP) whereas the branching operations are performed by subdividing rectangles (Algorithm 1) and subsimplices of Λ (Algorithm 2).

Algorithm 1 ([3]).

Initialization. Choose any point $x^0 \in X_E \cap V(X)$. Find λ^0 such that x^0 is an optimal solution to Problem (P_{λ^0}) given by $\max\{\lambda^0 Cx : x \in X\}$.

Set $k = 0$ and go to iteration k.

Iteration k $(k \geqq 0)$.

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

$$\max dx \tag{Pk}$$

subject to

$$x \in X, \quad \lambda \in \Lambda, \quad \lambda Cx \geqq \lambda Cx^i \quad (i = 0, 1, \ldots, k).$$

Let ξ_{k+1} denote the optimal value of this problem.

Step k.2. If $\xi_{k+1} = dx^j$ for some $j \in \{0, \ldots, k\}$, then terminate: x^j is an optimal solution to (P). Otherwise continue.

Step k.3. Verify the efficiency of $\overline{x}^{k+1}$ by solving the linear program given by

$$\max\{eCx : x \in X,\ Cx \geqq C\overline{x}^{k+1}\} \tag{Lk}$$

If this program has an optimal solution x^* such that $eCx^* = eC\overline{x}^{k+1}$, then terminate; $\overline{x}^{k+1}$ solves Problem (P). Otherwise continue.

Step k.4. Solve the linear program

$$\max\{\lambda Cx : x \in X\} \tag{Pλ}$$

with $\lambda = \overline{\lambda}^{k+1}$. Let x^{k+1} be any basic optimal solution of this program. Increase k by 1 and go to iteration k.

Convergence theorem 1.

(i) *The algorithm terminates only when a global optimal solution for Problem* (P) *has been found.*

(ii) *The algorithm terminates after a finite number of iterations.*

Proof. (i) If algorithm terminates at Step k.2., then $\xi_{k+1} = dx^j$ for some $0 \leq j \leq k$. Since ξ_{k+1} is an upper bound for the optimal value of (P), and $x^j \in X_E$, x^j is a global optimal solution to (P).

If the algorithm terminates at Step k.3., then $\overline{x}^{k+1} \in X_E$. Since $\xi_{k+1} = d\overline{x}^{k+1}$, it follows that $\overline{x}^{k+1}$ solves (P).

(ii) Since x^j for every $j \geq 0$ is an extreme point of the polytope X, it is sufficient to show that all these points are distinct. Indeed, since x^{j+1} is generated at Step j.4., $eCx^* > eC\overline{x}^{j+1}$. Thus $\overline{x}^{j+1} \notin X_E$, and therefore $\overline{\lambda}^{j+1}Cx^{j+1} > \overline{\lambda}^{j+1}C\overline{x}^{j+1}$. But from Step j.1, we have $\overline{\lambda}^{j+1}C\overline{x}^{j+1} \geq \overline{\lambda}^{j+1}Cx^j$ for all $0 \leq j \leq k$. By transitivity we see that $x^{k+1} \notin \{x^0, \ldots, x^k\}$. Hence the algorithm terminates after a finite number of iterations, since the vertex set of X is finite.

Remark. The crucial question in the above algorithm is of solving the relaxed Problem (Pk). This problem is a difficult global optimization problem because of the bilinear constraints

$$\lambda Cx \geq \lambda Cx^i \quad (i = 0, \ldots, k).$$

These constraints can be replaced by only one constraint defined by

$$g_k(\lambda, x) := \max\{\lambda Cx^i : i = 1, \ldots, k\} - \lambda Cx \leq 0.$$

Since $g_k(\lambda, x)$ is a convex-concave function, Problem (Pk) could be solved by several existing methods ([17], [18], [19]). These methods however are infinitely convergent, and efficient only if the dimension of "concave-space" is small.

In [3] Benson proposed an infinitely convergent branch-and-bound algorithm for solving Problem (Pk) by using the convex envelope of the bilinear terms λCx over a rectangle which can be easily calculated by a formula given in [1].

Now we describe another method which solves Problem (IP') directly. As mentioned this problem is equivalent to (P).

Note that since the function $h(\lambda, x)$ is convex-linear on $\Lambda \times X$, Problem (IP') is of the form of the problem considered in our paper [17]. Thus one could use the method proposed there for finding a global optimal solution of (IP'). However to apply that algorithm to Problem (IP') the subdivision process would be performed in R^n, and therefore [17] would be a good choice only if n is small. In many applications however the number n is much larger than p. The algorithm that we are going to describe is also a branch-and-bound procedure in which the branching operation is a simplicial bisection which is not necessarily exhaustive and takes place in R^p only.

As usual we adopt the convention that the maximum of a function over an empty set equals $-\infty$.

Algorithm 2.

Initialization. Set $S_0 = \Lambda$, $\Gamma_0 = \{S_0\}$, $v^0 \in S_0$ and solve the linear program

$$\max_{z \in X} v^0 C z$$

to obtain an optimal solution z^0 (hence $z^0 \in X_E$).

Take $t^0 \in \partial g(v^0)$ (a subgradient of the convex function g at v^0) and solve the Program (RPS_0) given by

$$\max\ dx$$

subject to

$$x \in X, \quad \lambda \in S_0, \quad (t^0, \lambda - v^0) + v^0 C z^0 - \lambda C x \leqq 0$$

to obtain an optimal solution $(\lambda^0, z^0) \in S_0 \times X$ and the optimal value $\alpha_0 := \alpha(S_0) = dz^0$.

Let $\overline{x}^0 = z^0$, $\beta_0 = d\overline{x}^0$. Set $k = 0$ and go to iteration k. Delete all $S \in \Gamma_k$ such that $\alpha(S) \leq \beta_k$. Let $\mathcal{R}_k$ be the remaining set.

1) If $\mathcal{R}_k = \emptyset$, then terminate: $\overline{x}^k$ is an optimal solution of (P).

2) If $\mathcal{R}_k \neq \emptyset$, then select $S_k \in \mathcal{R}_k$ such that $\alpha_k := \alpha(S_k) = \max\{\alpha(S) | S \in \mathcal{R}_k\}$.

Let (λ^k, x^k) be the obtained solution of Problem (RPS_k). Divide the simplex S_k into two simplices S_{k_1} and S_{k_2} by a simplicial bisection will be presented below.

Choose $u^k \in V(S_{k_1}) \cap V(S_{k_2})$ and solve the linear program:

$$\max\{u^k C z : z \in X\}.$$

Let w^k be the obtained solution of this linear Program ($w^k \in X_E$).

Take $d^k \in \partial g(u^k)$ and solve the relaxed Pprograms (RPS_{k_i}) ($i = 1,2$):

$$\max \; dx$$

subject to

$$x \in X, \quad \lambda \in S_{k_i}, \quad (d^k, \lambda^k - u^k) + u^k C w^k - \lambda C x \leqq 0$$

to obtain the optimal value $\alpha(S_{k_i})$ and optimal solutions (λ^{k_i}, x^{k_i}) ($i = 1,2$).

If $\alpha(S_{k_i}) > \alpha(S_k)$ let $\alpha(S_{k_i}) = \alpha(S_k)$. As these programs are solved we may obtain new efficient points. Let $\overline{x}^{k+1}$ be the best efficient point known so far. Let $\beta_{k+1} = d\overline{x}^{k+1}$. Set

$$\Gamma_{k+1} := (\Gamma_k \setminus \{S_k\}) \cup \{S_{k_1}, S_{k_2}\}.$$

Increase k by 1 and go to iteration k.

In order to implement the above algorithm, the relaxed Problem (RPS) must be solved for any simplex $S \subset \Lambda$. This problem can be solved via linear programs due to the following lemma.

Lemma 3.1. *For each simple $S_k \subseteq \Lambda$ the relaxed Problem (RPS_k)*

$$\max \; dx$$

subject to

$$x \in X, \quad \lambda \in S_k, \quad (t^k, \lambda - v^k) + v^k C z^k - \lambda C x \leqq 0$$

attains its global maximum, if any, at a point (λ^k, x^k) such that $\lambda^k \in V(S_k)$.

Proof. Let

$$L(\lambda, x) = \langle t^k, \lambda - v^k \rangle + v^k C z^k - \lambda C z.$$

Then L is a bilinear function. Problem (RPS_k) then is equivalent to the problem

$$\max \{dx : x \in X. \; \min_{\lambda \in S_k} L(\lambda, x) \leqq 0\}.$$

Since $L(.,x)$ is linear for each fixed $x \in X$ we have

$$\max\left\{dx : x \in X, \min_{\lambda\in S_k} L(\lambda,x) \leqq 0\right\}$$
$$= \max\left\{dx : x \in X, \min_{\lambda\in V(S_k)} L(\lambda,x) \leq 0\right\},$$

which implies the lemma.

Thus solving Problem (RPS_k) amounts to solving the p linear programs:

$$\max\{dx : x \in X, L(\lambda^i,x) \leqq 0\} \quad (\lambda^i \in V(S_k)).$$

If $\alpha(\lambda^i)$ denotes the optimal value of the linear program corresponding to $\lambda^i \in V(S_k)$ then $\alpha(S_k) := \max\{\alpha(\lambda^i) : \lambda^i \in V(S_k)\}$ is the optimal value of Problem (RPS_k).

We turn now to the simplicial bisections which can be used in the above algorithm. Here we give two simplicial bisections which seem to be suitable for Algorithm 2.

The following simplicial subdivision introduced first by Horst [11] can be described as follows.

Let S be a fully dimensional simplex in R^p. Let v^i and v^j be two vertices of S such that the edge of S defined by these two vertices is longest. Let u be the midpoint of this edge, i.e., $u = (v^i + v^j)/2$. We bisect S into the two simplices S_1 and S_2 by replacing v^i and v^j by u respectively.

Clearly $S = S_1 \cup S_2$. Moreover this bisection is exhaustive in the sense that any infinite nested sequence generated by it tends to a single point (see e.g. [11], [12]).

It is well recognized that this bisection is not expected to be a best way to bisect a simplex, since it does not take into account the data of problem being solved as well as the iteration points obtained in the previous iterations. Below we give another simplicial bisection which uses the points obtained through a bounding operation. The idea of this bisection is taken from the branching operation in our ealier paper [15]. The main difference is that the sets generated by this bisection are simplices whereas in [15] they may be polytopes.

Let S be a subsimplex of Λ. Assume that we want to divide S into two simplices. Consider Problem (RPS):

$$\max\ dx$$

subject to

$$x \in X, \quad \lambda \in S, \quad \langle t, \lambda - u\rangle + uCw - \lambda Cx \leq 0$$

where $u \in V(S)$, $t \in \partial g(u)$ and $w \in \arg\max_{x \in X} uCx$ are given.

Let $(\lambda, x) \in S \times X$ be an optimal solution of Problem (RPS). It is clear that if $\lambda = u$, then (λ, x) is a feasible point of Problem (IP'). Thus the simplex S may be deleted from further consideration. In particular, if the optimal solution (λ^k, x^k) of Problem (RPS_k) is feasible for (IP'), then the algorithm terminates since the optimal value $\alpha(S_k)$ of Problem (RPS_k) is an upper bound for the optimal value of (IP'). Hence if $\lambda^k \neq u^k$ it suggests bisecting the simples S_k via the edge determined by the vertices λ^k and u^k of S_k. Precisely, we bisect S_k as follows:

Let (λ^k, x^k) be an optimal solution of Problem (RPS_k). Assume that $\lambda^k \neq u^k$. Let $u^{k+1} = (\lambda^k + u^k)/2$ be the midpoint of the edge $[\lambda^k, u^k]$ determined by λ^k and u^k. We bisect S_k into the two simplices S_{k_1} and S_{k_2} by replacing λ^k and u^k by u^{k+1} respectively. We shall refer to this bisection as adaptive simplicial bisection (or subdivision). The points λ^k, u^k will be called *bisection points* of S_k. Since the edge $[\lambda^k, u^k]$ is not necessarily longest, this bisection may not be exhaustive. However under an appropriate condition this subdivition is sufficient to guarantee the convergence.

Definition 3.1. An infinite sequence $\{S_q\}$ of simplices generated by the above bisection process is said to be locally exhaustive if the sequences of their bisection points converge to the same limit. A simplicial subdivision process is said to be locally exhaustive if its every infinite nested sequence is locally exhaustive.

Locally exhaustive subdivisions have been used in our recent papers [15], [20], [22], [23], [24].

From the definition it is clear that the simplicial bisection process via a longest edge is locally exhaustive. Another locally exhaustive bisection process is the simplicial bisection via a shortest edge of the simplex.

Convergence theorem 2. (1) *If the algorithm terminates at iteration k, then $\overline{x}^k$ is a global optimal solution of Problem* (P).

(2) *If the algorithm is infinite and the simplicial bisection process*

used in the algorithm is locally exhaustive, then $\alpha_k \to P(X_E)$, *and the sequence* $\{x^k\}$ *has a limit point which solves* (P).

Proof. (1) If the algorithm terminates at iteration k, then $\mathcal{R}_k = \emptyset$. Hence $\alpha(S) \leq \beta_k$ for all $S \in \Gamma_k$. In particular $\alpha_k \leq \beta_k = d\bar{x}^k$. Since $\alpha_k \geq P(X_E)$ and $\bar{x}^k$ is feasible for (P) we have that $\alpha_k = \beta_k = d\bar{x}^k = \alpha_*$ and $\bar{x}^k$ is a global optimal solution of (P).

(2) Assume now that the algorithm is infinite. Then there must exist an infinite nested subsequence $\{S_q\}$ of the sequence $\{S_k\}$. Let λ^q and u^q be the bisection points of S_q. From the locally exhaustive assumption, the sequences $\{\lambda^q\}$ and $\{u^q\}$ converge to the same limit point, say, $\bar{\lambda}$.

On the other hand, since (λ^q, x^q) is feasible for Problem (RPS_q) and $g(u^q) = u^q C w^q$, it follows that the inequality $\langle t^q, \lambda^q - u^q \rangle + g(u^q) - \lambda^q C x^q \leq 0$ is satisfied.

Note that $\{t^q\}$ is bounded and g is continuous [7] we obtain in the limit that $g(\bar{\lambda}) - \bar{\lambda} C \bar{x} \leq 0$. From the definition of g follows $\bar{\lambda} C x - \bar{\lambda} C \bar{x} \leq 0$ for all $x \in X$ which together with $\bar{\lambda} \in \Lambda$, $\bar{x} \in X$ implies that $(\bar{\lambda}, \bar{x})$ is feasible for (IP'). Hence $\bar{x} \in X_E$. Furthermore, from $\alpha_q = dx^q \geq P(X_E)$ follows $\lim \alpha_q = d\bar{x} \geq P(X_E)$. Hence $\bar{x}$ is an optimal solution of (P). This and the monotonicity of the sequence $\{\alpha_k\}$ imply the theorem.

Remark. Consider the following parametric problem

$$\max\{tdx + h(\lambda, x),\ \lambda \in \Lambda,\ x \in X\}, \qquad \text{(Pt)}$$

where $h(\lambda, x)$ is defined as in Problem (CP). In [25], [26] it is shown that under a certain nondegenerate assumption there exists $t^* > 0$ such that any solution of (Pt*) is also an optimal solution to (P). Note that for each $t > 0$ (Pt) is a convex-concave programming problem.

4. SOLUTION METHODS BY EXTREME POINT SEARCH

Another approach for solving problem (P) is extreme point search technique. The main idea of this approach is based on the fact that (P) has a global optimal solution among the vertices of X. To accomplish this idea, the proposed methods iteratively identify efficient faces

X_λ, $(\lambda \in \Lambda)$ of X. For each such face X_λ that is found, an efficient extreme point of X which maximizes dx over X_λ is generated. Then procedures for searching all efficient faces are executed, and a global optimal solution of (P) thereby is found.

Below is an informal description of such an extreme point search algorithm. For more detail see [4].

Algorithm 3 (An informal description).

At the start an arbitrary efficient extreme point x^0 is found. Thereafter in each iteration, an additional efficient extreme point is found which is distinct from all previous generated efficient points. In some iterations, the new point is guaranteed to have a larger objective function value than all previously generated efficient points. In other iterations, it may not.

Assume $k \geqq 0$. To find an efficient extreme point which distinct from each of x^j, $0 \leqq j \leqq k$, iteration k executes the following step. First it seeks a point x which, together with some $\lambda \in \Lambda$, is a feasible point of Problem (Pk) and which satisfies $dx > LB_k$, where LB_k is the largest lower bound for the optimal value of (P). If no such point x exists, then a global optimal solution has been found. The algorithm then terminates. If such a feasible point of (Pk) exists, then the algorithm finds any such a feasible point and denotes it by $(\overline{x}^{k+1}, \overline{\lambda}^{k+1})$.

Next the algorithm uses this point to find a new efficient extreme point x^{k+1}. The method used to find x^{k+1} depends upon the nature of the point $\overline{x}^{k+1}$. Since $\overline{x}^{k+1} \in X$, either

(i) $\overline{x}^{k+1} \notin X_E$, or

(ii) $\overline{x}^{k+1} \in X_E \cap V(X)$, or

(iii) $\overline{x}^{k+1} \in X_E$ but $\overline{x}^{k+1} \notin V(X)$.

In all but the first case, the new efficient extreme point found by the algorithm is guaranted to satisfy $dx^{k+1} > LB_k$.

When $\overline{x}^{k+1}$ is not efficient the algorithm chooses x^{k+1} to be any basic solution of Problem (Pλ) with $\lambda = \overline{\lambda}^{k+1}$. It can be seen that in this cases, x^{k+1} is an efficient extreme point and distinct from each of the previously found efficient extreme points. However $dx^{k+1} > LB_k$ may not hold in this case.

When $\overline{x}^{k+1}$ is efficient point the algorithm set x^{k+1} equal to

$\overline{x}^{k+1}$. This clearly guarentees that x^{k+1} is distinct from each point x^j, $0 \leqq j \leqq k$ and that $dx^{k+1} > LB_k$.

Finally when $\overline{x}^{k+1}$ is efficient but not extreme point of X, the point x^{k+1} is found by maximizing dx over the efficient face which contains $\overline{x}^{k+1}$. In this case x^{k+1} is also a new efficient extreme point for which $dx^{k+1} > LB_k$ holds. The algorithm thus is finite, since in each iteration a new extreme point of X is generated.

Recently Benson in [5] developed another algorithm for solving Problem (P) in the case where d linearly depends upon the rows of the matrix C. An important special case of this problem is the problem where $d = -c^i$ and c^i is the ith row of C. The latter problem arises in decision making theory when it requires us to minimize a criterion function over the efficient set. The proposed algorithm is based on the following proposition.

Proposition 4.1 [5]. *Let X be as in Corollary 2.3 and d linearly depends upon the rows of C. Then there exists a positive number M^* such that for any $M \geq M^*$, $P(X_E) = t^*$, where t^* is the smallest value of the parameter t in the Problem* (Wt) *given by*

$$w_t := \max \lambda Cx - bu - tv \tag{Wt}$$

subject to

$$Ax \leq b, \quad uA + vd - \lambda C \geqq 0.$$
$$e\lambda = M, \quad \lambda \geqq e, \ x, \ u, \ v \geqq 0$$

such that $w_t = 0$.

The proof of this proposition can be done by a similar argument as that in the proof of Corollary 2.3. Note that for each fixed t Problem (Wt) is a bilinear program.

Let

$$t_m := \min\{dx : x \in X\}, \quad t^m := \max\{dx : x \in X\}.$$

It is clear that t_m and t^m are lower and upper bounds respectively for $P(X_E) = t^*$.

Proposition 4.1 leads to the following method, called bisection extreme point search algorithm for solving Problem (P) when d linearly depends upon the rows of C.

Algorithm 4 (Outline of the bisection point search algorithm).

Let an interval $[L, U]$ containing t^* be given (at the start set $L = t_m$ and $U = t^m$). Set $t := (L+U)/2$ and checking whether $w_t = 0$ or $w_t > 0$.

If $w_t = 0$, set $U = t$, (L is unchanget); if $w_t > 0$, set $L = t$. (U is unchanged).

Repeat the procedure with the new interval. The algorithm terminates when $U - L$ is small enough.

For checking the value w_t one must solve the bilinear programming Problem (Wt). The following corollary says that determining w_t leads to maximization of a convex function over X.

Corollary 4.1. ([5]) *For each $t \geq t_m$ we have*

$$w_t = \max\{h_t(x) : Ax \leq b,\ x \geq 0\},$$

where for each $x \in R^m$, h_t is the continuous picewise linear convex function defined by

$$h_t(x) := \max \lambda Cx - bu - tv$$

subject to

$$uA + vd - \lambda C \geq 0, \quad e\lambda = M, \quad \lambda \geq e,\ u,\ v \geq 0$$

(M as in Proposition 4.1).

5. BICRITERIA CASE

An important special case of Problem (P) that frequently arises in many applications, especially in the bimatrix game theory, is the bicriteria case, i.e., the matrix C has exactly two rows. In this case

Problem (P) could be solved more efficiently. As we have seen Algorithm 2 for this case involves a simplex bisection in R^2, and therefore it is quite efficient even with n is fairly large.

In this section we show that solving (P) for bicriteria case amounts to performing a parametric simplex tableau with one parameter in the objective function. In an important special case when $d = \alpha c^1 + \beta c^2$ with $\alpha \leq 0$, $\beta \geq 0$, in particular $d = -c^1$, we show that a globally optimal solution of Problem (P) can be obtained by solving at most two linear programs.

The algorithm we are going to describe is based upon the following proposition whose proof can be found, for example in [28].

Proposition 5.1. *Let x^0 be an extreme point of X, then $x^0 \in X_E$ if and only if there exists a $\lambda \in R^p$, $\lambda > 0$ such that x^0 is an optimal solution for the scalarized problem*

$$\max \lambda Cx, \textit{ subject to } x \in X. \qquad (\mathrm{P}\lambda)$$

By dividing to $\sum \lambda_i$ one can always assume that $\sum \lambda_i = 1$. Thus in the case $p = 2$, Problem (Pλ) can be written as

$$\max\langle tc^1 + (1-t)c^2, x\rangle, \quad \text{subjected to} \quad x \in X, \qquad (\mathrm{Pt})$$

where c^1 and c^2 are two rows of the matrix C.

From Proposition 5.1 and the fact that the set of optimal solutions of a linear program is a face of its feasible domain, it follows that there exists a finite set I of real numbers such that $X_E = \bigcup_{t \in I} X_t$ where X_t denotes the solution set of the linear Program (Pt).

Let $\xi(t)$ denotes the optimal value of (Pt). Then solving Problem (P) amounts to solving the following linear Program (Lt), one for each t in I:

$$\max \ dx \qquad (\mathrm{Lt})$$

subject to

$$x \in X, \quad \langle tc^1 + (1-t)c^2, x\rangle = \xi(t)$$

Let x^t be an optimal solution and $\eta(t)$ be the optimal value of this problem. Then $\eta(t^*) := \max\{\eta(t) : t \in I\}$ is the globally optimal value and x^{t^*} is an optimal solution of Problem (P). Using these results we have the following algorithm

Algorithm 5 (A parametric simplex algorithm for a bicriteria linear problem).

Assume that we are given the critical values $t_1, \ldots, t_K$ of the parametric linear Program (Pt). Let α_0 be a lower bound for $P(X_E)$ and $x^0 \in X_E$ such that $dx^0 = \alpha_0$. Set $i = 1$.

Iteration i $(i = 1, \ldots, K)$.

Step 1. Solve the linear program

$$\max\{\langle t_i c^1 + (1 - t_i)c^2, x\rangle,\ x \in X\}.$$

Let w^i be the obtained optimal basic solution, $B_i = (z_{jk})$ the corresponding basic matrix, and J_i the set of the basic indices.

Step 2. (The case when w^i is also an optimal solution of (Lti)).

If either

$$\Delta_{ik} := c_k^{t_i} - \sum_{j \in J_i} z_{jk} c_j^{t_i} < 0\ \forall k \notin J_j \text{ or } \quad d_k = \sum_{j \in J_i} z_{jk} d_j \le 0\ \forall k \notin J_i.$$

then set

$$x^i := \begin{cases} w^i \text{ if } dw^i > \alpha_{i-1} \\ x^{i-1} \text{ otherwise} \end{cases}$$

and $\eta_i = dx^i$.

If $i = K$, then terminate: $x^* := x^i$ is an optimal solution of Problem (P).

If $i < K$, then increase i by 1 and go to iteration i.

Step 3 (The case when w^i is not an optimal solution of (Lti)).

If

$$d_k - \sum_{j \in J_i} z_{jk} d_j > 0 \quad \text{for some} \quad k \in J_i,$$

then let

$$J_i^+ := \left\{ k \notin J_i : c_k^{t_i} - \sum_{j \in J_i} z_{jk} c_j^{t_i} < 0 \right\}$$

and solve the linear Program (Mti) given by

$$\max\{dx : x \in X,\ x_k = 0, k \in J_i^+\} \qquad \text{(Mti)}$$

Let y^i denote the obtained optimal solution of this linear program, and set

$$x^i = \begin{cases} y^i \text{ if } dy^i > \alpha_{i-1} \\ x^{i-1} \text{ otherwise} \end{cases}$$

and $\eta_i = dx^i$. Increase i by 1 and go to iteration i.

An important special case of Problem (P) occurs when

$$d = -c^i, \quad \text{for some} \quad i \in \{1, \ldots, p\}.$$

This problem is considered in some papers [13], [21]. For the cases when $p = 2$ and d is a linear combination of the two rows of C, Benson has shown that the maximal value of dx over X_E attains at a vertex of X which is also an optimal solution of at least one of the following three linear problems:

$$\max c^i x, \quad \text{subject to} \quad x \in X, \quad (i = 1, 2), \tag{Li}$$

$$\max dx, \quad \text{subject to} \quad x \in X. \tag{L}$$

Using this result Benson proposed an algorithm for maximizing dx over X_E by generating all basic solutions of these linear programs.

In particular, it is easily shown that an optimal solution of Problem (P) with $d = \alpha c^1 + \beta c^2$ and $\alpha \leq 0$, $\beta \geq 0$ can be obtained by solving the linear program

$$\max C^1 x, \text{ subjected to } x \in X_2$$

with X_2 being the solution set of (L2).

6. CONCLUSIONS

The Problem (P) of maximizing a real valued linear function over the efficient set of a multiple objective linear program has important uses in multiple criteria decision making. This is a difficult global optimization problem due to the fact that its feasible region is in general a nonconvex set. In this paper we have given a brief survey of computational methods for this problem. The existing methods have been classified into two approaches: branch-and-bound technique and efficient extreme point search scheme. The bicriteria case has been considered, and an efficient parametric simplex algorithm has been described for this case.

Due to the inherent difficulty of the considered problem, the proposed methods are efficient only in particular cases where either the number of the criteria or the number of the underlying variables are somewhat small. For other cases, to our knowledges there does not exist an efficient solution method in the literature. General cases of Problem (P) where either the objective function or the concerning multiple objective program are no longer linear and/or X is a unbounded convex set would be an interesting subject to further researches. Decomposition methods which employ the fact that the constancy space of r is of $n-k$ dimension, where k is the rank of the matrix C, are now being prepared [27] in order to develop more efficient methods for solving Problem (P).

REFERENCES

1. F. A. Al-Khayyal and J. Falk, *Jointly constrained biconvex programming*, Mathematics of Operations Research **8** (1983), 273-386.
2. H. P. Benson, *Optimization over the efficient set*, J. of Mathematical Analysis and Applications **98** (1984), 562-590.
3. H. P. Benson, *An all-linear programming relaxation algorithm for optimizing over the efficient set*, J. of Global Optimization **1** (1991), 83-104.
4. H. P. Benson, *A finite nonadjacent extreme point search algorithm for optimization over the efficient set*, J. of Optimization Theory and Applications **73** (1992), 47-63.
5. H. P. Benson, *A bisection - extreme point search algorithm for optimizing over the efficient set in the linear dependence case*, J. of Global Optimization **3** (1993), 95-111.
6. H. P. Benson and S. Sayin, *A face search heuristic algorithm for optimizing over the efficient set*, Naval Research Logistics **40** (1993), 103-116.
7. C. Berge, *Topological spaces*, Mac Millam, New York, 1968.
8. M. I. Dessouky, M. Ghiassi and W. J. Davis, *Estimates of the minimum nondominated criterion values in multiple-criteria decision-making*, Engineering Costs and Production Econometrics **10** (1979), 95-104.
9. R. J. Hillestad and S. E. Jacobsen, *Reverse convex programming*, Applied Mathematics and Optimization **6** (1980), 63-78.
10. R. J. Hillestad and S. E. Jacobsen, *Linear programs with an additional reverse convex constraint*, Applied Mathematics and Optimization **6** (1980), 257-269.

11. R. Horst, *An algorithm for nonconvex programming problems*, Mathematical Programming **10** (1976), 312-321.

12. R. Horst and H. Tuy, *Global optimization*, Springer-Verlag, Berlin, 1993.

13. H. Isermann and R. E. Steuer, *Computational experience concerning payoff table and minimum criterion values over the efficient set*, European J. of Operational Research **33** (1987), 91-97.

14. Le Dung Muu, *A convergent algorithm for solving linear programs with an additional reverse convex constraint*, Kybernetika **91** (1986), 438-425.

15. Le Dung Muu and W. Oettli, *An algorithm for indefinite quadratic programming with convex constraints*, Operations Research Letters **10** (1991), 323-327.

16. Le Dung Muu and W. Oettli, *Method for minimizing a convex-concave function over a convex set*. Journal of Optimization Theory and Applications, **70** (1991), 337-384.

17. Le Dung Muu, *An algorithm for solving convex programs with an additional convex-concave constraints*, Mathematical Programming **61** (1993), 75-87.

18. Le Dung Muu and W. Oettli, *Combined branch-and-bound and cutting plane methods for solving a class of nonlinear programming problems*, J. of Global Optimization **3** (1993), 377-391.

19. Le Dung Muu, *Convex-concave programming as a decomposition approach to global optimization*, Acta Mathematica Vietnamica **18** (1993), 61-67.

20. Le Dung Muu, *Methods for optimization of a linear function over the efficient set*, J. of Global Optimization (to appear).

21. Le Dung Muu and Ng. D. Dan, *Simplex method for optimizing a linear function over the efficient set of a bicriteria linear problem*, submitted.

22. Le Dung Muu and B. T. Tam, *Efficient methods for solving certain bilinear programming problems*, Acta Mathematica Vietnamica **19** (1994), 97-110.

23. Le Dung Muu and B. T. Tam, *Minimizing the sum of a convex function and the product of two affine functions over a convex set*, Optimization **24** (1992), 57-62.

24. Le Dung Muu, B. T. Tam and L. Schaible, *Efficient algorithms for solving certain nonconvex programs dealing with the product of two affine fractional functions*, J. of Global Optimization, **6** (1995), 179-191.

25. Le Dung Muu and W. Oettli, *Convergence of an adaptive penalty method for monotone variational inequalities and convex optimization*, Nonlinear Analysis: Theory, Methods and Applications **18** (1992), 1-10.

26. Le Dung Muu, *On a Lagrangian penalty function method for convex programs*, Applied Methematics and Optimization **25** (1992), 1-9.

27. Le Dung Muu, *Decomposition method for maximizing a convex function over the efficient set*, (submitted).

28. J. Philip, *Algorithm for the vector maximization problem*, Mathematical Programming **2** (1972), 207-229.

29. R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, 1970.

Institute of Mathematics
P.O. Box 631, Bo Ho
Hanoi, Vietnam

Received November 29, 1993