

A CONVEX-CONCAVE PROGRAMMING METHOD FOR OPTIMIZING OVER THE EFFICIENT SET

LE DUNG MUU

ABSTRACT. The problem of optimizing a real valued function over the efficient set of a multiple objective linear program has some applications in multiple objective decision making. The main difficulty of this problem arises from the fact that its feasible domain, in general, is nonconvex and not given explicitly. In this paper we formulate this problem as a linearly constrained convex-concave program where the number of “non-convex variables” is just equal to the number of independent criteria. We propose inner and outer procedures to constructing an initial set allowing convex-concave programming decomposition methods to be applied.

1. INTRODUCTION

Let X be a closed convex set in the Euclidean space R^n and C a $(p \times n)$ -matrix whose i th row is denoted by c^i . We recall that a point $x^0 \in X$ is said to be *efficient* of C on X if whenever $Cx \geq Cx^0$, $x \in X$, then $Cx = Cx^0$. By $E(C, X)$ we shall denote the set of all efficient points of C on X . The efficient-set even in the case when X is polyhedral may not necessarily be convex. Generating this set in its entirety is thus possible only in certain special cases. Even in these special cases the computational efforts required to generate all of the efficient points become rapidly unmanageable and seem to growth exponentially with problem size. In many situations however a real valued function, say f , is available which acts as a criterion function for measuring the importance of or for discriminating among the efficient alternatives. The problem of finding a most preferred (with respect to f) efficient point can be written as a mathematical programming problem

$$(1.1) \quad \max \{f(x) : x \in E(C, X)\}.$$

Received June 23, 1998

1991 Mathematics Subject Classification. 90C29

Key words and phrases. Optimization over the efficient set, initial polyhedron, linearly constrained, convex-concave programming, decomposition algorithm.

This paper is supported in part by the Basic Program in Natural Sciences.

Note that when X is polyhedral and f is quasiconvex, Problem (1.1) attains its global optimal solution at a vertex of X . This is an immediate consequence of the well known fact that X_E is the union of faces of X . Employing this property Philip [15] outlined a cutting plane procedure for solving (1.1) with f linear. This procedure was implemented recently in [5] with f quasiconvex.

A main difficulty of this problem is raised from the fact that the constrained set $E(C, X)$ is not convex and not given explicitly. To overcome this difficulty some equivalent representations of $E(C, X)$ have been presented.

In [3] Beson used Philip's simplex to describe the efficient set $E(C, X)$ as a system of infinite inequalities defined by bilinear forms. Using this formulation Beson developed a branch-and-bound algorithm for Problem (1.1) with f linear. This algorithm consists of finite iterations but each iteration requires solving a bilinear program. In [4] the computational effort for solving the encountered bilinear programs in [3] is weakened to finding a feasible point better than the current one.

In [12] (see also [1]) $E(C, X)$ is defined by adding to X a convex-concave constraint and branch-and-bound algorithms using an adaptive simplicial subdivision were proposed for solving the resulting convex-concave constrained problem with f concave. In the bicriteria case this method results in a parametric simplex procedure [6, 12] for optimizing a linear function over $E(C, X)$. In [8] the convex-concave constraint is replaced by a reverse convex one, and a cutting plane method using convexity and disjunctive cuts is developed there for maximizing a quasiconvex function over $E(C, X)$. The cutting planes in this algorithm, as in those of Philip and Bolintineanu, are created in x -space.

In [17] Thach et al formed $E(C, X)$ as a d.c. set (difference of two convex sets) and used its dual form to develop an outer approximation algorithm whose vertex-searching takes place in the criteria space R^p .

Recently an inner approximation algorithm using a vertex-searching operation performed in a k -dimensional Euclidean space with $k = \text{rank } C$ was described in [11] for solving the reverse convex constrained form of Problem (1.1) with f quasiconvex.

From an algorithmic viewpoint a new difficulty raised when applying basic techniques of global optimization such as branch-and-bound and outer approximation is that the effective domain of function defining the efficient set is not given explicitly. In this case the available methods in [9, 19] for constructing an initial polyhedral convex set (simplex, cone, box) to a branch-and-bound or an outer approximation algorithm cannot be used.

In this paper we first continue our works in [11] by presenting linearly constrained convex-concave programming formulations to optimizing a function (not necessarily convex) over the efficient set. In this case the optimal value may not attain at a vertex of X , therefore solution methods using inner approximation techniques fail to apply. Next we show how to reduce “nonconvexity size” of the resulting problem. Finally we propose the use of inner and outer approximation techniques for constructing an initial polyhedral convex set which allows convex-concave programming decomposition methods to be applied for maximizing a concave function over $E(C, X)$.

2. LINEARLY CONSTRAINED MATHEMATICAL PROGRAMMING FORMULATIONS

As usual, for two vector $a = (a_1, \dots, a_p)$ and $b = (b_1, \dots, b_p)$ the inequality $a \geq b$ means that $a_i \geq b_i$ for all i . We shall use the following definition which is well known in vector optimization.

Definition 2.1. Let $Q : R^n \rightarrow R^p$ and K be a convex set in R^n . A real valued function q defined on K is said to be *nondecreasing with respect to* Q (or briefly *Q -nondecreasing*) on K if $q(x) \leq q(x')$ for every $x, x' \in K$ satisfying $Q(x) \leq Q(x')$.

The function q is said to be *increasing with respect to* Q (or briefly *Q -increasing*) on K if $q(x) < q(x')$ whenever $x, x' \in K$, $Q(x) \leq Q(x')$ and $Q(x) \neq Q(x')$.

Following [2, 8] we define

$$G(X) := \{x \in R^n : Cy \geq Cx \text{ for some } y \in X\}$$

and

$$r(x) := \max \{e^T(Cy - Cx) : Cy \geq Cx, y \in X\}.$$

It is well known [8] that if the efficient set $E(C, X)$ is not empty then r is finite on $G(X)$. As usual we take $r(x) = -\infty$ if $x \notin G(X)$. So the effective domain of r is $G(X)$. Clearly, $G(X)$ is a polyhedral convex set if so is X .

The function r has the following properties [1, 2, 7, 8, 11] which will be useful in the sequel.

Lemma 2.1. *Assume that $E(C, X) \neq \emptyset$, then*

- (i) $r(x) \geq 0$ for every $x \in X$,
- (ii) $r(x) = 0$, $x \in X$ if and only if $x \in E(C, X)$,
- (iii) $-r$ is C -increasing on its effective domain,

(iv) If X is polyhedral, then $-r$ is piecewise linear convex and subdifferentiable at every point in $G(X)$.

Since $-r$ is increasing on $G(X)$, it follows that if f is C -nondecreasing on $G(x)$ then the function $f - Nr$ is C -increasing on $G(X)$ for every positive number N . In this case Problem (1.1) is equivalent to the linearly constrained program

$$(2.1) \quad \max \{ f(x) - Nr(x) : x \in X \}$$

by the following lemma.

Lemma 2.2. *If f is C -nondecreasing on $G(X)$ then for any $N > 0$ Problem (2.1) is equivalent to (1.1).*

Proof. Let x^N be a global optimal solution of (2.1). Then $x^N \in E(C, X)$. Indeed, otherwise there would exist $x' \in X$ such that $Cx' \geq Cx^N$, $Cx^N \neq Cx'$. Since $f - Nr$ is C -increasing we would have $f(x^N) - Nr(x^N) < f(x') - Nr(x')$. This would contradict the optimality of x^N . Hence $r(x^N) = 0$. Then

$$f(x^N) = f(x^N) - Nr(x^N) \geq f(x) - Nr(x) = f(x) \quad \forall x \in E(C, X) \subset X$$

which means that x^N solves Problem (1.1) globally.

Conversely, if x^* is a global optimal solution of (1.1), then

$$f(x^*) - Nr(x^*) = f(x^*) \geq f(x^N) = f(x^N) - Nr(x^N).$$

Thus, x^* is a global optimal solution of (2.1), because so x^N is. \square

Let define, for each $u = (u_1, \dots, u_p) \in R^p$, the number

$$|u|_- := \begin{cases} \max\{-u_j, u_j \leq 0\} & \text{if } u \not\geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

and denote by $C(A)$ the image of a set A under C , i.e.,

$$C(A) := \{y : y = Cx, x \in A\}.$$

Proposition 2.1. *Let $A \subseteq G(X)$ and $f(x) = \varphi(Cx)$ for every $x \in A$ with φ being a differentiable function on A . Then for every*

$$N > N_A := \sup \{ |\varphi'(\xi)|_- : \xi \in C(A) \}$$

the function $F_N(x) := f(x) - Nr(x)$ is C -increasing on A .

Proof. Let $x, x' \in A$ such that $Cx \leq Cx'$ and $Cx \neq Cx'$. Since $Cx \leq Cx'$ we have

$$\begin{aligned} & \max\{e^T(Cy - Cx') : Cy \geq Cx', y \in X\} \\ & \leq \max\{e^T(Cy - Cx) : Cy \geq Cx, y \in X\}. \end{aligned}$$

This implies

$$\begin{aligned} F_N(x') &= f(x') - Nr(x') \\ &= f(x') - N \max\{e^T(Cy - Cx') : Cy \geq Cx', y \in X\} \\ &= f(x') - N \max\{e^T(Cy - Cx + Cx - Cx') : Cy \geq Cx', y \in X\} \\ &\geq f(x') - N \max\{e^T(Cy - Cx) : Cy \geq Cx', y \in X\} \\ &\quad - N \max\{e^T(Cx - Cx') : Cy \geq Cx', y \in X\} \\ &\geq f(x) - N \max\{e^T(Cy - Cx) : Cy \geq Cx, y \in X\} \\ &\quad + f(x') - f(x) - N \max\{e^T(Cx - Cx') : Cy \geq Cx', y \in X\} \\ &= F_N(x) + f(x') - f(x) + Ne^T C(x' - x). \end{aligned}$$

Since $f(x) = \varphi(Cx)$, it follows that

$$F_N(x') \geq F_N(x) + \varphi(Cx') - \varphi(Cx) + Ne^T C(x' - x).$$

By the well known mean value theorem we have

$$\begin{aligned} F_N(x') &\geq F_N(x) + \varphi'(\xi)C(x' - x) + Ne^T C(x' - x) \\ &= F_N(x) + (\varphi'(\xi) + Ne^T)(Cx' - Cx). \end{aligned}$$

Remembering that $Cx' \geq Cx$, $Cx' \neq Cx$ we deduce $F_N(x') > F_N(x)$ whenever $\varphi'(\xi) + Ne^T > 0$. The latter is fulfilled for every $N > |\varphi'(\xi)|_-$, $\xi \in C(A)$. \square

In an important special case [7, 10, 11] where $f(x) = d^T x$ with d being a linear combination of the criteria, the number N_A can be easily calculated. Namely we have the following result.

Corollary. Let $f(x) = d^T x$ with $d = \sum_{j=1}^p w_j c^j$. Then for every $N > |w|_-$ the function $F_N(x) := f(x) - Nr(x)$ is C -increasing on any set $A \subseteq G(X)$.

Alternatively, another linearly constrained formulation to Problem (1.1) can be determined as follows. Let

$$\Lambda := \left\{ \lambda = (\lambda_1, \dots, \lambda_m) : \lambda_j \geq 1, \sum_{j=1}^p \lambda_j \leq M \right\}.$$

From Philip [15] we know that for every $M > 0$ sufficiently large, $x \in E(C, X)$ if and only if there exists $\lambda \in \Lambda$ such that x is an optimal solution of the linear program

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

Let

$$h(\lambda, x) := g(\lambda) - \lambda^T C x.$$

Then h is a nonnegative convex-linear function on $\Lambda \times R^n$ and $h(\lambda, x) = 0$, $(\lambda, x) \in \Lambda \times X$ if and only if $x \in E(C, X)$ and $g(\lambda) = \lambda^T C x$.

Proposition 2.2. *For any $N > 0$, if (λ^*, x^*) is an optimal solution of problem*

$$(2.2) \quad \max \{ f(x) - N h(\lambda, x) : (\lambda, x) \in \Lambda \times X \}$$

with x^ is efficient, then x^* solves (1.1).*

Proof. Let y_x be a solution of the program defining $r(x)$, i.e., $r(x) = \langle e, C y_x - C x \rangle$. Since $\lambda \geq e$, $y_x \in X$, $C y_x \geq C x$ we have $r(x) \leq h(\lambda, x)$. Let x be any efficient point. Then $r(x) = 0$ and there exists $\lambda \in \Lambda$ such that $h(\lambda, x) = 0$. Since (λ^*, x^*) is a global optimal solution to (2.2), it follows that

$$f(x) - N r(x) = f(x) - N h(\lambda, x) \leq f(x^*) - N h(\lambda^*, x^*) \leq f(x^*) - N r(x^*).$$

Since $r(x) = r(x^*) = 0$, this implies $f(x) \leq f(x^*)$ which means that x^* solves (1.1). \square

3. VARIABLE REDUCTION FORM

From the above results it follows that optimizing a function over the efficient set of a multiple objective linear program amounts to solving linearly constrained problems of forms (2.1) or (2.2). Note that even with f linear these problems are multiextremal.

It is well known that computational costs (time, memory) required for convergence of an algorithm for solving a multiextremal optimization problem increase very rapidly as the number of nonconvex variables gets larger. Therefore from a computational viewpoint an important question is that how to reduce the number of nonconvex variables. Fortunately, in problems (2.1) and (2.2) the number of nonconvex variables can be reduced to just the number of independent criteria which in many applications is much less than that of total variables.

In fact, noting that $\text{rank } C = k$, by using linear transformations the bilinear term $\lambda^T C x$ can be cast into the form $\sum_{j=1}^k \xi_j \lambda_j x_j$. Since the convexity is preserved under linear transformations, Problem (2.2) takes the form

$$(3.1) \quad \max \left\{ f_1(x) - N g_1(\lambda) + N \sum_{j=1}^k \xi_j \lambda_j x_j : (\lambda, x) \in \Lambda_1 \times X_1 \right\}$$

where Λ_1, X_1 are polyhedra, g_1 is convex, while f_1 is convex or concave if so is f . In this form the bilinear term $\sum_{j=1}^k \xi_j \lambda_j x_j$, which makes the problem difficult, depends only on x_j and λ_j ($j = 1, \dots, k$).

To reduce nonconvex variables in Problem (2.1) we first recall [16 Section 8] that for a convex set K the set of all y satisfying $x + ty \in K$ for every $t \geq 0, x \in K$ is called the *recession cone* of K and denoted by O^+K . The largest subspace contained in O^+K is called the *lineality space* of K . This subspace consists of the zero vector and all the non-zero vectors y such that, for every $x \in K$, the line through x in the direction of y is contained in K . The dimension of the lineality space of K is called the *lineality* of K . If F is a closed proper convex function, then all non-empty level sets of the form $\{x : F(x) \leq \alpha\}, \alpha \in R$, have the same lineality space [16 Theorem 8.7]. This lineality space is often called the *constancy space* of F .

Let C_0 be the polyhedral cone vertexed at the origin and generated by the matrix C , i.e.,

$$C_0 := \{y \in R^n : Cy \leq 0\}.$$

Then we have the following result.

- Lemma 3.1.** (i) $G(X) = C_0 + X$,
(ii) $C_0 \subset O^+G(X)$,
(iii) *The constancy space of $-r$ is $L(C) := \{y : Cy = 0\}$.*

Proof. (i) Let $a \in G(X)$. By the definition of $G(X)$ there is $b \in X$ such that $C(a - b) \leq 0$. Then $z := a - b \in C_0$. Hence $a \in C_0 + X$. Conversely, if $a := d + b$ with $d \in C_0$, $b \in X$. Then $C(d + b) = Cd + Cb \leq Cb$ which means that $d + b \in G(X)$.

(ii) Let x be any point of $G(X)$. By definition of $G(X)$ there exists $z \in X$ such that $Cx \leq Cz$. Thus, for any positive number t and any y satisfying $Cy \leq 0$ we have

$$Cx + tCy \leq Cx \leq Cz$$

which means that $x + ty \in G(X)$. Hence $C_0 \subset O^+G(X)$.

(iii) It is easy to verify that $-r$ is closed proper convex and that $\text{dom}(-r) = G(X)$. Since $G(X) = \{x : -r(x) \leq 0\}$, by Theorem 8.7 in [16] the constancy space of $-r$ is the lineality space of $G(X)$ which is equal to the set $-O^+G(X) \cap O^+G(X)$. From (i) we have $L(C) \subset O^+G(X)$. Now we need to show $L(C) \subset -O^+G(X)$. Indeed, let $x \in G(X)$. By the definition of $G(X)$, there exists $z \in X$ such that $Cz \geq Cx$. Then for every $t \geq 0$, $y \in L(C)$ one has

$$C(x - ty) = Cx - tCy = Cx \leq Cz,$$

which means that $x - ty \in G(X)$. This is true for all $t \geq 0$ and $y \in L(C)$. Hence $L(C) \subset -O^+G(X)$. \square

From this proposition it follows that r is constant on the subspace

$$L(C) := \{x : Cx = 0\}.$$

Thus, if for some $x \in L(C)$, $x \in X$ we have

$$r(x) := \max \{e^T C(y - x) : Cy \geq Cx, y \in X\} = 0,$$

then every point of X lying in $L(C)$ belongs to $E(C, X)$. Otherwise, $L(C)$ contains no efficient points. Note that $\dim L(C) = n - k$ because $\text{rank } C = k$.

Without loss of generality we may assume that the first k -rows c^1, \dots, c^k of the matrix C are independent. Let L denote the linear space generated by these rows. Then the algebraic direct sum of L and $L(C)$ is R^n . So every $x \in R^n$ can be uniquely written as $x = x^1 + x^2$ with $x^1 \in L$, $x^2 \in L(C)$.

Lemma 3.2. *Let $x = x^1 + x^2$, $x^1 \in L$, $x^2 \in L(C)$. Then*

$$r(x) = \max \{e^T C(y - x^1) : Cy \geq Cx^1\} := r(x^1).$$

Proof. Let

$$\begin{aligned} X(x) &:= \{y \in X : Cy \geq Cx\}, \\ X(x^1) &:= \{y \in X : Cy \geq Cx^1\}. \end{aligned}$$

Since $x = x^1 + x^2$ and $Cx^2 = 0$, we have $X(x) = X(x^1)$ and $e^T C(y - x) = e^T C(y - x^1)$. Thus $r(x) = r(x^1)$ by the definitions of $r(x)$ and $r(x^1)$. \square

Let

$$c^1, \dots, c^k, b^{k+1}, \dots, b^n$$

be a basis of R^n , where b^{k+1}, \dots, b^n is a basis of $L(C)$. Since convexity is preserved under linear transformations, we may assume that all data are given in this basis. Let $x = x^1 + x^2$ with $x^1 \in L$, $x^2 \in L(C)$. Then x^1 and x^2 is uniquely written as

$$x^1 = \sum_{i=1}^k u_i c^i, \quad x^2 = \sum_{i=k+1}^n u_i b^i.$$

Thus one can identify x with the vector (u, v) where u and v are defined by

$$u := (u_1, \dots, u_k), \quad v := (v_{k+1}, \dots, v_n).$$

As usual suppose that the polyhedron X is given by

$$(3.2) \quad X = \{x \in R^n : Ax + b \leq 0\},$$

where A is an $(m \times n)$ -matrix, $b \in R^m$. Denote by A_1 and A_2 the matrices obtained from A by taking the first k - and the last $(n - k)$ -columns of A respectively. Then X can be expressed as

$$X = \{(u, v) : A_1 u + A_2 v + b \leq 0\}.$$

Let

$$s(u) := r\left(\sum_{j=1}^k u_j c^j\right).$$

Then Problem (2.1) takes the form

$$(3.3) \quad \max \{F_N(u, v) := f(u, v) - Ns(u) : A_1 u + A_2 v + b \leq 0\}.$$

4. A SOLUTION METHOD BY BRANCH-AND-BOUND

In this section we shall describe a branch-and-bound method for solving problem (3.3) with f being a concave function on R^n (hence f is continuous). Since s is concave, this is a multiextremal optimization problem. Actually it is a linearly constrained d.c. optimization problem. Two special features of this problem which differ from a general d.c. optimization problem are the following:

- The concave function s that makes the problem difficult depends upon only the variable u .
- The effective domain of s is not given explicitly.

The first point suggests that a branch-and-bound procedure for solving this problem should take its branching operation in the u -space, since it is well recognized that reducing the dimension of search space may dramatically reduce computational cost required for convergence.

The second point requires special treatments for constructing a polyhedral convex set from which a branch-and-bound procedure can start. This initial set must contain a global optimal solution (or its projection on u -space) but it is not beyond the domain where the involved functions are properly defined. Moreover the vertices and extreme directions of this polyhedron can be computed with a reasonable effort. In the sequel we shall refer to a polyhedron satisfying these conditions as a *valid polyhedron*. In view of the first point such a valid polyhedron should be constructed in the u -space.

We propose two algorithms using inner and outer approximations for constructing a valid polyhedron. By $V(S)$ and $R(S)$ we will denote the sets of extreme points and directions respectively.

Inner Approximation

The procedure starts with a (simple structured) polyhedral convex set S contained in $G(X)$. If $X \subset S$ we are done. Otherwise we seek a point $x \in V(X) \setminus S$ and repeat the procedure with S replaced by the convex hull of S and x .

As usual for a set A we shall denote by A^* the polar of A , i.e.,

$$A^* = \{z : \langle z, x \rangle \leq 1 \ \forall x \in A\},$$

and by $P(A)$ the projection of A on the u -space. Then the algorithm can be described as follows.

IV ALGORITHM (Inner approximation for constructing an initial polyhedral set)

Initialization step. Set a polyheron $S_0 \subset G(X)$; compute $V(S_0^*)$ and $R(S_0^*)$. Let $j \leftarrow 0$.

Iteration j ($j = 0, 1, \dots, K$). For each $u \in V(S_j^*) \cup R(S_j^*)$, $u \neq 0$ solve the linear program

$$\max \{ \langle u, x \rangle : x \in X \}$$

to obtain a basic optimal solution $x(u)$ (hence $x(u) \in V(X)$).

Step 1.

a) If

$$\langle u, x(u) \rangle \leq 1 \quad \forall u \in V(S_j^*)$$

and

$$\langle u, x(u) \rangle \leq 0 \quad \forall u \in R(S_j^*),$$

then set $S = S_j$ and terminate the algorithm.

b) Otherwise, choose $u^{j+1} \in V(S_j^*) \cup R(S_j^*)$ and

$$x^{j+1} \in \operatorname{argmax} \{ \langle u^{j+1}, x \rangle : x \in X \}$$

such that

$$\langle u^{j+1}, x^{j+1} \rangle > 1 \quad \text{if } u^{j+1} \in V(S_j^*)$$

or

$$\langle u^{j+1}, x^{j+1} \rangle > 0 \quad \text{if } u^{j+1} \in R(S_j^*).$$

Step 2. Take

$$S_{j+1} = \operatorname{conv}(S_j, \{x^{j+1}\})$$

and

$$S_{j+1}^* := S_j^* \cap \left\{ u = (u_1, \dots, u_k) : \sum_{i=1}^k u_i \langle x^{j+1}, c^i \rangle \leq 1 \right\}.$$

Compute $V(S_{j+1}^*)$, $R(S_{j+1}^*)$. Set $j \leftarrow j + 1$ and go to iteration j .

Noting that $S_0 \subset G(X)$, $x^j \in G(X)$ for every $j > 1$ we have $S_j \subset G(X)$ for all j because $G(X)$ is convex and $S_{j+1} = \operatorname{conv}(S_j, \{x^{j+1}\})$. Since at each iteration $j \geq 0$, the polyhedron S_{j+1} is obtained by taking the convex hull of S_j and a new vertex x^{j+1} of X , this algorithm is finite yielding a polyhedral convex set S satisfying $X \subset S \subset G(X)$.

A valid polyhedron in the u -space can be obtained from S by taking, for example, its projection on u -space.

Remark. Without loss of generality we may assume that $0 \in X$. Then the cone $C_0 \subset G(X)$. So at the start we can set $S_0 = C_0$. Since $C_0 = \{x : Cx \leq 0\}$, its polar $C_0^* = \text{cone}(c^1, \dots, c^k)$. Note that, since $S_{j+1} = \text{conv}(S_j, \{x^{j+1}\})$, from convex analysis (see e.g. [16, Theorem 3.3]) it follows that

$$S_{j+1}^* = S_j^* \cap \{u : \langle x^{j+1}, u \rangle \leq 1\}.$$

Thus the vertices and extreme directions of S_{j+1}^* can be calculated from those of S_j^* by available methods [9, 18]. These methods work well in small dimensional space, but become computationally expensive in higher space. Note that since

$$\text{lineality } S_j = \text{lineality } S_0 = n - k,$$

we have

$$\dim S_j^* = n - \text{lineality } S_j = k.$$

So the vertices and extreme directions are computed in a k -dimensional linear space.

Outer Approximation

Alternatively one can use an outer approximation, which can be regarded as a dual version of the above inner approximation, for constructing a valid polyhedron.

Namely, as usual we start the outer approximation procedure by setting a bounded polyhedral convex set (polytope) S_0 in the u -space such that $P(X) \subset S_0$ and its vertices are easy to calculate. If $S_0 \subset P(G(X))$ we are done. Otherwise we cut off parts of $S_0 \setminus P(G(X))$ iteratively until we obtain a polytope that is contained in $P(G(X))$. Since we do not know explicitly constraints defining $G(X)$, we cut off parts of $S_0 \setminus P(X)$ rather than $S_0 \setminus P(G(X))$.

Two questions raised in this outer approximation are of how to construct a polyhedron S_0 in the u -space such that $P(X) \subset S_0$ and how to cut off a part of $S_0 \setminus P(X)$ by a hyperplane.

Recall that the polytope X is given by

$$X = \{(u, v) : A_1 u + A_2 v + b \leq 0\}.$$

The following well known lemma is useful in the sequel.

Lemma 3.3 [9, 18]. *Let $P(X)$ denote the projection of X in the u -space. Then a point u^0 belongs to $P(X)$ if and only if $h = 0$ is the optimal solution of the linear program defined by*

$$(L(u^0)) \quad \max \{\langle A_1 u^0 + b, h \rangle, A_2^T h = 0, h \geq 0\}.$$

Let h^0 be an optimal solution of $(L(u^0))$. If $\langle A_1 u^0 + b, h^0 \rangle > 0$, then $u^0 \notin P(X)$. The constraint

$$\langle A_1 u + b, h^0 \rangle \leq 0$$

is satisfied for every $u \in P(X)$ but is violated by u^0 .

From this lemma it follows that

$$P(X) = \{u : \langle A_1 u + b, h \rangle \leq 0, h \in V(W)\}$$

where

$$(3.4) \quad W := \left\{ h = (h_1, \dots, h_m) : \sum_{j=1}^m h_j \leq 1, A_2^T h = 0 \right\}.$$

This gives an explicit form to the polyhedron $P(X)$. However if m is large enough, the number of the vertices of $P(X)$ may be large, and generating all of them is very costly. In this case we choose a certain set $V_0 \subset W$ and take

$$S_0 := \{u : \langle A_1 u + b, h \rangle \leq 0, h \in V_0\}.$$

Obviously, $P(X) \subset S_0$.

We can also construct the polytope S_0 by setting first a simplex X_0 in the x -space and then take $S_0 = P(X_0)$.

To check whether $S_0 \subset P(G(X))$ or not it is sufficient to verify

$$(3.5) \quad V(S_0) \subset P(G(X)).$$

Note that since $X \subset G(X)$, if $u^0 \notin P(G(X))$ then $u^0 \notin P(X)$ as well. So we can apply Lemma 3.3 to construct a cutting plane that cuts off u^0 from S_0 but does not cut off any point from $P(X)$.

OV ALGORITHM (outer approximation for computing an initial polyhedron)

Step 0. Find a polytope S_0 such that $S_0 \subset P(X)$. Calculate $V(S_0)$.

Step 1. If $v \in P(G(X))$ for all $v \in V(S_0)$, then $S_0 \subset P(G(X))$. The algorithm terminates.

Otherwise, find $u^0 \in V(S_0)$ such that $u^0 \notin P(G(X))$ (hence $u^0 \notin P(X)$).

Step 2. Solve the linear program

$$\max \{ \langle A_1 u^0 + b, h \rangle, A_2^T h = 0, h \geq 0 \}.$$

to obtain a basic solution h^0 .

Set

$$S_0 \leftarrow \{u \in S_0 : \langle A_1 u + b, h^0 \rangle \leq 0\}.$$

Step 3. Compute $V(S_0)$ and go to step 1.

The valid polyhedron constructed by the above methods is not a simple set (simplex, cone, rectangle...) in general. Therefore, branch-and-bound methods using simplicial, conical or rectangular subdivisions starting from this set can not be applied in general. For this initial polyhedron suitable subdivisions should be used. We propose to use the adaptive polyhedral bisection operations developed in convex-concave programming [13, 14] for solving Problem (3.3) with $f(u, v)$ is concave. Note that since f is concave, an optimal solution of (3.3) in general is not attained at a vertex of its feasible domain.

Suppose that a polytope S_0 in the u -space satisfying $P(X) \subset S_0 \subset P(G(X))$ has been constructed. Let S be a subpolyhedron of S_0 . Consider problem (3.3) with respect to S , i.e.,

($P(S)$)

$$\beta(S) := \max \{F_N(u, v) := f(u, v) - Ns(u) : u \in S, A_1 u + A_2 v + b \leq 0\}.$$

We decouple the variables u and v in this problem to obtain a relaxed problem

($R(S)$)

$$\alpha(S) := \max \{F_N(u, v) := f(u, v) - Ns(w) : u, w \in S, A_1 u + A_2 v + b \leq 0\}.$$

Then $\alpha(S) \geq \beta(S)$. Let (u^S, v^S, w^S) be an optimal solution of ($R(S)$). Clearly, if $s(u^S) = s(w^S)$, in particular if $u^S = w^S$, then $\alpha(S) = \beta(S)$.

Note that solving $R(S)$ amounts to solving the linearly constrained convex program (concave maximization)

$$\max \{f(u, v) : u \in S, A_1 u + A_2 v + b \leq 0\}$$

and to minimizing the concave function s on S which, since s is finite on S , can be solved by evaluating s at the vertices of S . Thus we may suppose that $0 = s(w^S)$ because otherwise if $s(w^S) > 0$ then $s(u) > 0$ for every $u \in S$, which means that S does not contain the projection of any efficient point. In these cases S can be eliminated from further consideration.

Suppose that $s(w^S) < s(u^S)$ which implies $u^S \neq w^S$. Then it suggests subdividing S by a branching operation that makes the distance between these two points tends to 0 as rapidly as possible.

We call u^S and w^S the subdivision (bisection) points and propose the following two subdivision strategies:

Rule 1 (Euclidean rule). Take $l^S = (w^S - u^S)/(\|w^S - u^S\|)$. Then we bisect S by setting

$$(3.6) \quad S^- := \{v : \langle l^S, v - (u^S + w^S)/2 \rangle \leq 0\},$$

$$(3.7) \quad S^+ := \{v : \langle l^S, v - (u^S + w^S)/2 \rangle \geq 0\}.$$

Clearly $w^S \in S^-$ and $u^S \in S^+$.

Rule 2 (Subgradient rule). Take $l^S \in \partial(s(w^S))$ and divide S as

$$(3.8) \quad S^- := \{v : \langle l^S, v - w^S \rangle \leq 0\},$$

$$(3.9) \quad S^+ := \{v : \langle l^S, v - w^S \rangle \geq 0\}$$

Then $w^S \in S^-$ and, since

$$\langle l^S, u^S - w^S \rangle \geq s(u^S) - s(w^S) = s(u^S) > 0,$$

$u^S \in S^+$. For these subdivision we have the following result:

Lemma 3.4. *Let $\{S_k\}$ be any infinite nested sequence of partition sets generated by subdivision Rule 1 or Rule 2, and let $\{w^k\}$, $\{u^k\}$ be the corresponding bisection points. Suppose that the sequences $\{w^k\}$, $\{u^k\}$ are bounded. Then they have a common cluster point.*

Proof. Since $\{S_k\}$ is nested and $S_k \subset S_0$ for all k , by taking subsequences if necessary, we may assume that

$$(3.10) \quad S_{k+1} \subset S_k^- \quad \forall k$$

or

$$(3.11) \quad S_{k+1} \subset S_k^+ \quad \forall k,$$

and that the sequences $\{u^k\}$, $\{w^k\}$ are convergent. Note that for both Rule 1 and Rule 2 the sequence $\{l^k\}$ is bounded, since in Rule 1 $\|l^k\| = 1$

for all k , whereas in Rule 2 $l^k \in \partial s(u^k)$. Thus we may also assume that the sequence $\{l^k\}$ is convergent.

Consider first the Rule 1. If (3.10) holds then $u^{k+1} \in S_k^-$ which means

$$\langle l^k, u^{k+1} - (u^k + w^k)/2 \rangle \leq 0.$$

Since

$$\langle l^k, u^k - (u^k + w^k)/2 \rangle \geq 0,$$

it follows that

$$\begin{aligned} 0 &\leq \langle l^k, u^k - (u^k + w^k)/2 \rangle \\ &\leq \langle l^k, u^k - (u^k + w^k)/2 \rangle - \langle l^k, u^{k+1} - (u^k + w^k)/2 \rangle \\ &= \langle l^k, u^k - u^{k+1} \rangle \leq \|u^k - u^{k+1}\| \rightarrow 0. \end{aligned}$$

Hence

$$\langle l^k, u^k - (u^k + w^k)/2 \rangle \rightarrow 0.$$

Since $\|l^k\| = 1$ we obtain in the limit that

$$u^k - (u^k + w^k)/2 = (u^k - w^k)/2 \rightarrow 0 \text{ as } k \rightarrow \infty.$$

If (3.11) holds, we use $w^{k+1} \in S_k^+$ to obtain, by a similar way, that

$$\begin{aligned} 0 &\geq \langle l^k, w^k - (u^k + w^k)/2 \rangle \\ &\geq \langle l^k, w^k - (u^k + w^k)/2 \rangle - \langle l^k, w^{k+1} - (u^k + w^k)/2 \rangle \\ &= \langle l^k, w^k - w^{k+1} \rangle \leq \|w^k - w^{k+1}\| \rightarrow 0. \end{aligned}$$

Thus

$$\langle l^k, w^k - (u^k + w^k)/2 \rangle \rightarrow 0,$$

which as before implies that $w^k - u^k \rightarrow 0$ as $k \rightarrow \infty$.

For the subdivision defined by Rule 2 the proof can be done by the same argument. \square

Now we are in a position to describe an algorithm for solving problem (3.3). The algorithm is a branch-and-bound procedure consisting of two phases. In the first phase the algorithm constructs a valid polyhedron that serves as an initial set to a branch-and-bound procedure using the above decoupling technique for bounding, and Rule 1 or Rule 2 for branching. For simplicity in what follows we shall use the notations $u^k, v^k, w^k \dots$ for

$u^{S_k}, v^{S_k}, w^{S_k} \dots$. As usual we agree that maximum over on the empty set is $-\infty$. Let $\varepsilon \geq 0$ be a tolerance. A feasible point x is said to be ε -optimal solution to Problem (3.3) if $f^* - f(x) \leq \varepsilon(|f(x)| + 1)$. The algorithm can then be described as follows.

ALGORITHM

Phase 1. Use *IV* or *OV* algorithm to construct a valid polyhedron S_0 .

Phase 2. At the beginning of this phase we have a polytope S_0 satisfying $P(X) \subseteq S_0 \subseteq P(G(X))$ whose vertices have been computed.

Step 0. Solve Program $R(S_0)$ to obtain $\alpha(S_0), u^0, w^0, v^0$. Set $\Delta_0 := \{S_0\}$, $\alpha_0 := \alpha(S_0)$ (current best upper bound), $\beta_0 := \beta(S_0)$ (current best lower bound), $x^0 := (u^0, v^0)$ (currently best feasible point). Set $k \leftarrow 0$.

Iteration k ($k = 0, 1, \dots$)

Step 1 (stop criterion). If $\alpha_k - \beta_k \leq \varepsilon(|\beta_k| + 1)$ terminate: x^k is an ε -optimal solution. Otherwise do the following:

Step 2 (selecting). Find $S_k \in \Delta_k$ such that

$$\alpha(S_k) := \max \{ \alpha(S') : S' \in \Delta_k \}.$$

Step 3 (branching). Use Rule 1 or 2 to divide S_k into S_k^- and S_k^+ .

Step 4 (bounding). Solve $R(S_k^-)$ and $R(S_k^+)$.

Step 5 (updating). Update α_k, β_k and x^k to obtain $\alpha_{k+1}, \beta_{k+1}$ and x^{k+1} . Set

$$\Delta_{k+1} := \{ S' \in (\Delta_k \setminus S) \cup \{S^-, S^+\} : \alpha(S') - \beta_{k+1} > \varepsilon(|\beta_{k+1}| + 1) \},$$

$k \leftarrow k + 1$ and go back to Step 1.

Convergence Theorem (i) *If the algorithm terminates at iteration k then $x^k = (u^k, v^k)$ is an ε -optimal solution.*

(ii) *If the algorithm never terminates then $\beta_k \nearrow f^*$, $\alpha_k \searrow f^*$, and any cluster point of $x^k = (u^k, v^k)$ is a global optimal solution.*

Proof. (i) is straightforward from the stop criterion of algorithm and the definition of Δ_k .

(ii) If algorithm runs with infinitely many iterations, it generates an infinite sequence $\{S_k\}$ of nested partition sets for which the corresponding sequences of bisection points $\{w^k\}, \{u^k\}$, by Lemma 3. 4, have a common cluster point, say u^* . By taking subsequences if necessary we may assume that $u^k \rightarrow u^*, w^k \rightarrow u^*, v^k \rightarrow v^*$.

Since the sequences $\{\alpha_k\}$, $\{\beta_k\}$ are monotone and

$$\alpha_k = f(u^k, v^k) - Ns(w^k), \quad \beta_k = f(u^k, v^k) - Ns(u^k),$$

we obtain in the limit that

$$\alpha^* = f(u^*, v^*) - Ns(u^*), \quad \beta^* = f(u^*, v^*) - Ns(u^*).$$

which implies $\alpha^* = \beta^* = f^*$.

Let $x = (u, v)$ be any cluster point of sequence $\{x^k = (u^k, v^k)\}$. Then (u, v) is feasible. Then there is a subsequence $\{x^j = (u^j, v^j)\}$ such that $u^j \rightarrow u$, $v^j \rightarrow v$. Using again $\beta_j = f(u^j, v^j) - Ns(u^j)$ and letting $j \rightarrow +\infty$ we have $\beta^* = f(u, v)$. Hence (u, v) is global optimal. \square

Corollary. *If $\varepsilon > 0$ the algorithm is finite.*

REFERENCES

1. Le T. H. An, Le D. Muu and Pham D. Tao, *D. C. Optimization approach for optimizing over the efficient set*, Oper. Res. Lett. **19** (1996), 117-128.
2. H. P. Benson, *Optimization over the efficient set*, J. of Math. Anal. and Appl. **98** (1984), 562-580.
3. H. P. Benson, *An all-linear programming relaxation algorithm for optimizing over the efficient set*, J. of Global Optim. **1** (1991), 83-104.
4. H. P. Benson, *A finite, nonadjacent extreme-point search algorithm for optimization over the efficient set*, J. of Optim. Theory and Appl. **73** (1993), 47-64.
5. S. Bolintineanu, *Minimization of a quasi-concave function over an efficient set*, Math. Prog. **61** (1993), 89-110.
6. Ng. D. Dan and Le D. Muu, *Parametric simplex method for optimizing a linear function over the efficient set of a bicriteria linear problem*, Acta Math. Vietnam. **21** (1996), 59-67.
7. J. Dauer and T. Fosnaugh, *Optimization over the efficient set*, J. of Global Optim. **7** (1995), 261-277.
8. J. Fülöp, *A Cutting Plane Algorithm for Linear Optimization over the Efficient Set*, In: Generalized Convexity, Edited by S. Komlosi, T Rapcsaj and S. Schaible, Springer, Berlin, 374-385.
9. R. Horst and H. Tuy, *Global Optimization (Deterministic approaches)*, Springer-Verlag, Berlin, 1993.
10. H. Isermann and R. E. Steuer, *Computational experience concerning Payoff tables and minimum criterion values over the efficient set*, Euro. J. of Oper. Res. **33** (1987), 91-97.
11. Le T. Luc and Le D. Muu, *Global optimization approach to optimizing over the efficient set*, In: Lect. Notes Eco. Math. Syst. **452** (1997), 183-196.
12. Le D. Muu, *Computational aspects of optimization over the efficient set*, Vietnam J. of Math. **23** (1995), 85-106.
13. Le D. Muu and W. Oettli, *A method for minimizing a convex-concave function over a convex set*, J. Optim. Theory and Appl. **70** (1990), 377-384.

14. Le D. Muu, *An algorithm for solving convex programs with an additional convex-concave constraint*, Math. Prog. **61** (1993), 75-87.
15. J. Philip, *Algorithms for the vector maximization problem*, Math. Prog. **2** (1972), 207-229.
16. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
17. P. T. Thach, H. Konno and D. Yojota, *Dual Approach to optimization on the set of Pareto-optimal solutions*, J. of Optim. Theo. and Appl. **88** (1996), 689-707.
18. T. V. Thieu, B. T. Tam and V. T. Ban, *An outer approximation method for globally minimizing a concave function over a compact set*, Acta Math. Vietnam. **8** (1983), 21-40.
19. H. Tuy, *Convex Analysis and Global Optimization*, Kluwer Academic Publishers, 1998.

INSTITUTE OF MATHEMATICS
BOX 631, BO HO, HANOI, VIETNAM