

# An Algorithmic Proof of the Polyhedral Decomposition Theorem

Mustafa Akgül

Department of Industrial Engineering, Bilkent University, P.O. Box 8, 06572  
Maltepe, Ankara, Turkey

It is well-known that any point in a convex polyhedron  $P$  can be written as the sum of a convex combination of extreme points of  $P$  and a non-negative linear combination of extreme rays of  $P$ . Grötschel, Lovász, and Schrijver gave a polynomial algorithm based on the ellipsoidal method to find such a representation for any  $x$  in  $P$  when  $P$  is bounded. Here we show that their algorithm can be modified and implemented in polynomial time using the projection method or a simplex-type algorithm : in  $n(2n + 1)$  simplex pivots, where  $n$  is the dimension of  $x$ . Extension to the unbounded case is immediate.

The decomposition theorem of Minkowskii, Motzkin, and Goldman [6] states that a polyhedron  $P = \{x \in \mathbb{R}^n \mathcal{R}: Ax \leq b\}$  can be written as  $P = K + R$ , where  $K$  is the convex hull of the extreme points (vertices) of  $P$  and  $R$  is the cone generated by the extreme rays of  $P$  (recession cone). In other words,  $x \in P$  if and only if

$$x = \sum_{i=1}^{i=p} \alpha_i x^i + \sum_{j=1}^{j=s} \beta_j q^j, \quad (1)$$

with

$$\sum \alpha_i = 1, \alpha \geq 0, \quad \beta \geq 0,$$

where the  $x^i$ 's are (among) the extreme points of  $P$  and the  $q^j$ 's are (among) the extreme rays of  $P$ . Often, one is interested in determining an explicit representation of a given  $x$  in  $P$ . By the Carathéodory theorem there is at least one such representation with  $p \leq n + 1$  and  $s \leq n$ . In their seminal work Grötschel, Lovász, and Schrijver [7,8], among other things, give a polynomial algorithm to find such a representation for a given  $x$  in a polyhedron  $P$  which is bounded, i.e., has  $R = \{0\}$ . They work with a full dimensional polytope and use a separation algorithm as a subroutine to find facets of the polytope. They solve  $n$  LP's with the ellipsoidal algorithm. Thus their algorithm needs  $n$  calls to the ellipsoidal algorithm, and  $n$  calls to separation subroutine assuming an explicit representation of  $P$  as the solution set of a system of linear inequalities. The ellipsoidal algorithm performs  $O(n^4 T)$  elementary operations, where  $T$  is the size of the input. They do not need an explicit description of  $P$  as a system of inequalities. However, then they need to call the separation routine  $O(n \log T)$  times, to replace simple ratio tests.

In this article we give a modification of the Grötschel-Lovász-Schrijver algorithm for a polyhedron explicitly represented as the solution set of a linear inequality system.

In particular, we avoid the separation algorithm and solve each LP with  $n$  simplex pivots by an algorithm which we call VERTEX. We also extend the algorithm to the case when  $P$  is unbounded and to the case when  $P$  has no vertex. In other words, we prove the following theorem.

**THEOREM:** For any  $x \in P$ , one can find a representation (1) with  $s \leq n, p \leq n + 1$  in at most  $n(2n + 1)$  simplex pivots.

Clearly, the decomposition theorem follows from the above theorem.

The article is organized as follows. We first review the necessary concepts from polyhedral theory and present the algorithm VERTEX. We then successively prove the theorem for a polytope, for an unbounded polyhedron with a vertex, and for a polyhedron with no vertex. We use the following notation. Greek symbols denote scalars. Lower-case Roman letters (except  $t$  and  $t_j$ , which are scalars) represent vectors. All vectors are column vectors, except  $a_i, c, c^j, w$ , and  $e$ , which are row vectors.  $z_j, w_j$ , and  $b_j$  denote the  $j$ th component of the corresponding vector.  $U_n$  denotes the  $n \times n$  unit matrix.

## 1. PRELIMINARIES

Given an  $m \times n$  matrix  $A$  and an  $m$ -vector  $b$ , we obtain a polyhedron  $P$  in  $\mathcal{R}^n$  as the solution set of  $Ax \leq b$  or, equivalently,

$$a_i x \leq b_i, \quad i \in I = \{1, 2, \dots, m\}. \quad (2)$$

We assume that  $A$  does not contain any zero rows.

For any subset  $X$  of  $P$ , let

$$I(X) = \{i \in I: a_i x = b_i, \forall x \in X\}. \quad (3)$$

be its *equality set*. By definition  $I(X)$  is maximal. Hence for  $i \notin I(X)$ , there exists an  $x \in X$  such that  $a_i x < b_i$ . When  $X = \{x\}$ , we have  $a_i x < b_i \forall i \notin I(X)$ .

The *rank* of a subset  $X \subset P$  is defined by

$$r(X) = r\{a_i: i \in I(X)\}, \quad (4)$$

where the right side denotes the linear rank. [When  $I(X) = \emptyset$ , we set  $r(X)$  to zero.]  $x \in P$  is a *vertex* (*extreme point*) of  $P$  if and only if

$$r(x) = n. \quad (5)$$

(Our terminology is a bit unconventional. For details see [1,2].) A polyhedron  $P$  is *full dimensional* if and only if  $r(P) = 0$ .

The *faces* of  $P$  are the empty set  $\emptyset$ ,  $P$  itself, and all solution sets of systems obtained from (2) by replacing some of the inequalities with equalities.  $\emptyset$  and  $P$  are improper, while all others are *proper faces* of  $P$ . Thus each face  $\mathcal{F}$  of  $P$ , with the possible exception of  $\emptyset$ , can be specified as

$$\begin{aligned} \mathcal{F} &= \mathcal{F}(J) = \{x: a_i x = b_i, i \in J, a_i x \leq b_i, i \notin J\} \\ &= \{x \in P: a_i x = b_i, i \in J\}, \end{aligned} \quad (6)$$

for some subset  $J$  of  $I$ . The mapping  $J \rightarrow \mathcal{F}(J)$  may in general be many to one. Clearly,  $\mathcal{F}(\emptyset) = P$  and  $I(\emptyset) = I$ .  $P$  is full dimensional if and only if  $I(P) = \emptyset$ .  $I(\mathcal{F})$

is the maximal  $J$  satisfying (6). By restricting our attention to  $I(\mathcal{F})$ 's we have a one-to-one correspondence between  $\mathcal{F}$  and  $I(\mathcal{F})$ . One can define the *dimension* of a face  $\mathcal{F}$  as

$$\dim \mathcal{F} = n - r(\mathcal{F}). \tag{7}$$

A maximal proper face, i.e., a *facet*, has dimension  $\dim P - 1$ . Clearly  $P$  is full dimensional if and only if  $\dim P = n$ .

The following is an elementary but very useful result.

LEMMA: Let  $\mathcal{F}$  be a face of  $P$  and let

$$c = \sum_{i \in I(\mathcal{F})} \lambda_i a_i, \quad \lambda_i > 0, \quad \text{for } i \in I(\mathcal{F}). \tag{8}$$

Then  $\bar{x}$  solves

$$\max \{cx : x \in P\} \tag{9}$$

if and only if

$$\bar{x} \in \mathcal{F}. \tag{10}$$

PROOF. (1) for  $x \in P$ ,  $cx = \sum_{i \in I(\mathcal{F})} \lambda_i a_i x \leq \sum_{i \in I(\mathcal{F})} \lambda_i b_i \equiv \alpha$ . Hence  $\alpha$  is an upper bound for  $cx$ . (2) For  $\bar{x} \in \mathcal{F}$ ,  $a_i \bar{x} = b_i, \forall i \in I(\mathcal{F}) \Rightarrow c\bar{x} = \alpha$ . Hence each  $\bar{x}$  solves (9). (3) For  $y \in P - \mathcal{F}$ , by definition of  $I(\mathcal{F})$  and  $\mathcal{F}$ , i.e., (3) and (6), there exists  $i \in I(\mathcal{F})$  such that  $a_i y < b_i$ . This together with (8) implies that  $c y < \alpha$ . Thus the maximizers of (9) are precisely the members of  $\mathcal{F}$ . ■

REMARK: If  $I(\mathcal{F}) = \emptyset$  then by convention  $c = 0$  in (8).

The following corollary is widely used in combinatorial optimization to aid in characterizing the extreme points of certain polyhedra, such as those that arise in matching theory and matroid theory. Since each extreme point is a face, we have the following corollary.

COROLLARY: For every extreme point  $\bar{x}$  of  $P$ , there exists at least one objective function  $cx$  (in fact, infinitely many) such that  $\bar{x}$  is the unique maximizer of (9).

## 2. THE ALGORITHM VERTEX

The key to our approach is the following algorithm.

- Input:  $P = \{x : Ax \leq b\}$ ,  $\bar{x} \in P$ ,  $c^T \in \mathcal{R}^n$ .
- Assumptions:  $P$  has a vertex and  $\max \{cx : x \in P\} < \infty$ .
- Output: a vertex  $x^*$  of  $P$  with  $c x^* \geq c \bar{x}$ .
- Complexity:  $n$  simplex iterations.

### Procedure and Justification

We will modify the algorithm in [1, p.33] to accommodate the fact that the  $x$ 's are *free* variables. Let us rewrite  $P = \{x : Ax \leq b\}$ , as  $P' = \{(x,s) : Ax + s = b, s \geq 0\}$ , and let  $z = (x^T, s^T)^T \in \mathcal{R}^{n+m}$ . Let  $\mathbf{S}(\mathbf{F})$  be the index set for the slack (original) variables, and  $\mathbf{B}(\mathbf{N})$  be the set of basic (nonbasic) variables. Our tableau for  $P'$  will consist of an update of  $A' = [A|U_m]$ , a row of reduced costs  $\bar{w}$  giving the corre-

sponding update of  $(c,0)$ , and extra row of  $\bar{z}^T$  (we will omit the transposition symbols), representing the current values of  $z = (x,s)$ . We will not transform the right-hand side vector, but instead determine step size and new values of the vector  $z$  explicitly by viewing the simplex method as a feasible direction algorithm. The algorithm will maintain the invariants:  $k \in \mathbf{S} \cap \mathbf{N}$  implies  $\bar{z}_k = 0$ ,  $k \in \mathbf{S} \cap \mathbf{B}$ ,  $z_k \geq 0$ .

Given  $\bar{x} \in P$ , clearly  $\bar{s} = b - A\bar{x} \geq 0$  is such that  $\bar{z} = (\bar{x}, \bar{s})$  is feasible for  $P'$ . Let  $A'$  together with  $\bar{w} = (c, 0)$  and  $\bar{z}$  be the initial tableau, with  $\mathbf{B} = \mathbf{S}$ . It is easy to see that once a basis  $\mathbf{B}$  of  $A'$  with  $|\mathbf{S} \cap \mathbf{N}| = n$  ( $\mathbf{F} \cap \mathbf{N} = \emptyset$ ) is obtained, then the current  $\bar{z} = (\bar{x}, \bar{s})$  is a vertex of  $P'$  and  $\bar{x}$  is a vertex of  $P$ . (Simply, the variables in  $\mathbf{S} \cap \mathbf{N}$  define  $n$  linearly independent hyperplanes passing through  $\bar{z}$ .) So long as  $|\mathbf{S} \cap \mathbf{N}| < n$ , we can choose a  $k \in \mathbf{F} \cap \mathbf{N}$ . If  $\bar{w}_k > 0$  ( $\bar{w}_k < 0$ ), we increase (decrease)  $z_k$  as much as possible while keeping the variables in  $\mathbf{S} \cap \mathbf{B}$  non-negative; the one of these variables thereby reduced to zero (ties broken arbitrarily) replaces variable  $z_k$  in  $\mathbf{N}$ . To determine this variable we need to perform a ratio test involving only the variables in  $\mathbf{S} \cap \mathbf{B}$ . Note that the result of this ratio test must be finite, otherwise the LP would be unbounded. If  $\bar{w}_k = 0$ , we can either increase or decrease  $z_k$ . At least one of these two feasible directions must have a finite step size, since we are assuming that  $P$  has a vertex (and thus contains no full line). Since we are restricting incoming variables to  $\mathbf{F} \cap \mathbf{N}$ , and the ratio test to  $\mathbf{S} \cap \mathbf{B}$ , every pivot will interchange a nonbasic free variable with a basic slack variable. Thus the algorithm will either deliver a vertex  $x^*$  of  $P$  with  $cx^* \geq c\bar{x}$  in exactly  $n$  pivots, or else discover a direction vector  $d$  such that the line  $\bar{z} + td$ ,  $t \in \mathcal{R}$ , lies in  $P'$ , thus demonstrating that  $P'$  (and hence  $P$ ) has no vertex. The direction vector  $d$  for variable  $k \in \mathbf{F} \cap \mathbf{N}$  is a rearrangement of  $\sigma(-\bar{a}^T, u^T)^T$ , where  $\bar{a}$  is the  $k$ 'th column of the current update of  $A'$ ,  $u$  is the  $k$ 'th column of the unit matrix  $U_n$ , and  $\sigma$  is set to  $+1$  ( $-1$ ) to increase (decrease)  $z_k$ . Then the ratio test is

$$t = \min \left\{ \frac{-\bar{z}_j}{d^T u^j}; j \in \mathbf{S} \cap \mathbf{B}, d^T u^j < 0 \right\},$$

where  $u^j$  is the  $j$ 'th column of the unit matrix  $U_{m+n}$ .

### Remarks

1. The existence of such an  $x^*$  is guaranteed by the fundamental theorem of linear programming under the given assumptions.
2. The idea of finding an extreme point solution from a feasible solution is well known, and treated in textbooks, e.g., [5,11].
3. The idea of using the simplex method for this purpose goes back to Dantzig [5, p.113, problems 22,23], and Charnes, Cooper, and Raik [4].
4. The technique of going to an extreme point from a near-optimal or interior point has been used recently in the ellipsoidal algorithm [1], the projective algorithm [9], and others [10,12].
5. The algorithm VERTEX can be replaced with the projection method. The projection method [13,3] generates a finite sequence of points  $x^j \in P$ , which determine expanding subsets  $J_j = I(x^j)$  of  $I$  and contracting faces  $\mathcal{F}(J_j)$  of  $P$ . The method starts with some  $x^0 \in P$ , possibly an interior point. At the  $j$ 'th step, if  $x^j$  is not already an extreme point, the method determines the direction  $d_j$  of the orthogonal projection of

$c$  upon  $\mathcal{F}(J_j)$ , and determines  $x^{j+1}$  as the point on the boundary of  $\mathcal{F}(J_j)$  obtained by moving from  $x_j$  in the direction of  $d^j$ ; i.e.,

$$x^{j+1} = x^j + t_j d^j, \tag{11}$$

where  $t_j \geq 0$  is determined by a simple ratio test. Thus at each iteration the cardinality of  $J_j$ , more importantly  $r(x^j)$ , increases at least by 1. Hence the algorithm reaches an extreme point in at most  $n$  iterations. It may happen that some  $d^j$  is the zero vector, due to the fact that the current nonextreme  $x^j$  might be optimal (which can happen almost every time in our main algorithm in the next section). Let  $Q$  be the projection matrix at  $x^j$ . Then  $d^j = Q c^T = 0$  implies that the whole face  $\mathcal{F}(J_j)$  is optimal for the LP. In particular, for any  $u \in \mathbb{R}^n$ ,  $Qu$ , if nonzero, is a feasible direction of  $\mathcal{F}(J_j)$  on which  $cx$  is constant, since  $cQu = (Qc^T)^T(Qu) = 0$ . We can choose  $u$  as any  $u^k$ , the usual unit vector such that  $Qu^k$ , the  $k$ th column of  $Q$ , is nonzero. Note that  $J_j$  and  $r(x^j)$  may increase by more than 1 per iteration; however, the computational cost of updating  $Q$  in this case is the same as with the equivalent application of rank-one updates.

6. The algorithm VERTEX can be used to check whether  $P$  has any extreme point. Given  $\bar{x} \in P$ , just take  $c = 0$  and apply the algorithm VERTEX.  $P$  has no extreme point if and only if the recession cone  $R = \{x: Ax \leq 0\}$  of  $P$  contains a line. Thus the algorithm will either deliver a vertex  $x^*$  of  $P$  or produce a direction vector  $d$  for which the ratio test fails. Note that for each such vector  $d$ , the components of  $d$  on  $\mathbf{S}$  are zero: They are zero on  $\mathbf{S} \cap \mathbf{B}$  because the ratio test fails and they are zero on  $\mathbf{S} \cap \mathbf{N}$  by definition. Hence the descriptions with respect to  $x$  and  $z$  are equivalent. Clearly  $d$  will be in the recession cone  $R$  [in fact, in the lineality space  $L = R \cap (-R)$ ].

### 3. THE MAIN ALGORITHM

We first study the case of bounded  $P$ . We construct sequences  $x^0, x^1, x^2, \dots, x^k$  of vertices,  $y^0, y^1, y^2, \dots, y^k$ , of points, and  $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_k$ , of faces of  $P$  so that  $\mathcal{F}_j \supset \mathcal{F}_{j+1}$ , and  $x^j, y^j \in \mathcal{F}_j, \mathcal{F}_j = \mathcal{F}(I(y^j))$ .

Given  $\bar{x}$ , the point to be represented as in (1), let  $y^0 = \bar{x}, \mathcal{F}_0 = \mathcal{F}(I(y^0))$ , and let  $x^0$  be the outcome of the algorithm VERTEX with input  $\bar{x}$  and  $c = \sum\{a_k; k \in I(y^0)\}$ . We claim that  $x^0 \in \mathcal{F}_0$ . This is certainly true for any  $c$  if the projection method is used to find the vertex  $x^0$  as explained in Remark 5 above. For VERTEX, it follows from the proposition presented below.

Assume that  $x^i, y^i, \mathcal{F}_i, I_i = I(\mathcal{F}_i)$  are defined for  $i \leq j$ . If  $y^j = x^j$  then stop; otherwise determine  $y^{j+1}$  by extending the line segment  $[x^j, y^j]$  through  $y^j$  until the boundary of  $P$  is met; i.e.,

$$y^{j+1} = x^j + t_{j+1} (y^j - x^j). \tag{12}$$

Clearly  $t_{j+1} > 0$  and its value can be determined by a simple ratio test, namely,

$$t_{j+1} = \min \left\{ \frac{b_k - a_k x^j}{a_k (y^j - x^j)}; a_k (y^j - x^j) > 0, k \in I_j \right\}. \tag{13}$$

Note that, because of the property  $\mathcal{F}_j = \mathcal{F}(I(y^j)), t_{j+1} > 1$  and it is finite, since by assumption  $P$  is bounded. Let

$$J_{j+1} = \{k \in I - I_j; a_k y^{j+1} = b_k\}, \tag{14}$$

i.e., the (nonempty) set of additional inequalities defining  $P$  which are satisfied as equality at  $y^{j+1}$ . Let

$$I_{j+1} = I_j \cup J_{j+1} = I(y^{j+1}), \tag{15}$$

and

$$\mathcal{F}_{j+1} = \mathcal{F}(I_{j+1}). \tag{16}$$

$x^{j+1}$  is the outcome of the algorithm VERTEX with inputs  $y^{j+1}$  and

$$c^{j+1} = \sum \{a_k: k \in I_{j+1}\}. \tag{17}$$

The following is now easy to prove

**PROPOSITION:**  $x^{j+1}$  solves  $\max \{c^{j+1} x: x \in P\}$ , and  $x^{j+1} \in \mathcal{F}_{j+1}$ .

**PROOF:** Notice that  $c^{j+1} y^{j+1} = \sum \{b_k: k \in I_{j+1}\}$  is an upper bound for the above LP. Since VERTEX assures  $c^{j+1} x^{j+1} \geq c^{j+1} y^{j+1}$ , the assertion follows, and then the second one is implied by the Lemma in Section 1. ■

Since  $r(\mathcal{F}_j)$  increases at least by 1 at each iteration, the process stops in at most  $n$  iterations. Hence for some  $p \leq n$ ,  $y^p = x^p$ , and we are done.

The required representation (1) of  $\bar{x}$  can now be obtained as follows. Let us rewrite (12) as

$$y_j = \frac{1}{t_{j+1}} y^{j+1} + \frac{t_{j+1} - 1}{t_{j+1}} x^j, \quad 0 \leq j < p. \tag{18}$$

Defining

$$\delta_j = \frac{1}{t_{j+1}}, \quad \gamma_j = \frac{t_{j+1} - 1}{t_{j+1}}, \tag{19}$$

we have

$$y^j = \delta_j y^{j+1} + \gamma_j x^j, \quad \delta_j + \gamma_j = 1, \quad \delta_j, \gamma_j \geq 0; \tag{20}$$

i.e.,

$$y^j \in \text{co}(y^{j+1}, x^j), \quad 0 \leq j < p. \tag{21}$$

Since  $y^p = x^p$ , by induction we have

$$y^j \in \text{co}(x^j, x^{j+1}, \dots, x^p), \quad 0 \leq j < p. \tag{22}$$

In particular,

$$\bar{x} = y^0 \in \text{co}(x^0, x^1, \dots, x^p). \tag{23}$$

It is easy to show by induction that a specific set of convex “weights” for (23), i.e., coefficients  $\alpha_j$  in the desired representation (1),

$$y^0 = \bar{x} = \sum_{j=0}^p \alpha_j x^j, \tag{24}$$

are given by

$$\begin{aligned} \alpha_0 &= \gamma_0, \\ \alpha_1 &= \delta_0 \gamma_1, \\ \alpha_j &= \delta_0, \delta_1, \dots, \delta_{j-1} \gamma_j, \quad j \leq p, \\ \alpha_p &= 1. \end{aligned} \tag{25}$$

This completes the proof of the theorem for the bounded case.

#### 4. THE UNBOUNDED CASE

The recession cone  $R$  of  $P$  can be written as  $R = \{x: Ax \leq 0\}$ . Clearly  $0 \in R$ . For ease in presentation we first study the case where  $0$  is the only solution of  $Ax = 0$ , i.e., where the *lineality space*  $L$  of  $P$  consists of  $0$ . This condition is satisfied if and only if  $P$  has a vertex. We want to find a hyperplane  $H$  such that  $H \cap R \equiv \bar{P}$  is a polytope whose extreme points are in 1-1 correspondence with the extreme rays of  $R$ . Under the hypothesis that  $P$  has a vertex, every nonzero element of  $R$  will satisfy at least one inequality as a strict inequality. Let  $e$  be the (row) vector of ones. Then for any  $x \neq 0$  in  $R$ , we have

$$eAx < 0. \tag{26}$$

By normalizing (26), we may take  $H$  as

$$eAx = -1. \tag{27}$$

Thus,  $R$  is generated by

$$\bar{P} = \{x: Ax \leq 0, eAx = -1\}; \tag{28}$$

i.e.,  $R = \text{cone}(\bar{P})$ , and extreme points of  $\bar{P}$  are in 1-1 correspondence with extreme rays of  $R$  (see, e.g., [11]). Geometrically, it is clear that  $\bar{P}$  is bounded. For completeness, let us show that  $\bar{P}$  is bounded. Let  $x \in \bar{P}$  and  $\pi = Ax$ . Then  $\pi \leq 0, e\pi = -1$ , implying that  $-1 \leq \pi_i \leq 0, \forall i$ ; hence  $\pi$  is bounded. The assumption that  $P$  has a vertex implies that  $r(A) = n$ . Since the system  $\pi = Ax$  has a unique solution for a given  $\pi$ , for any full-rank square submatrix  $B$  of  $A$  we have  $\pi_B = Bx$ . Then  $x = B^{-1}\pi_B$  and it follows that  $x$ , hence  $\bar{P}$  is bounded.

If  $P$  is unbounded, then during the determination of  $t_{j+1}$  in (12), for some first  $j$ , we may end up with

$$t_{j+1} = \infty.$$

In that case we have determined a recession direction  $d \in R$ ,

$$d \equiv y^j - x^j. \tag{29}$$

$d \neq 0$  makes a nontrivial contribution to the representation of  $\bar{x}$  in (1). By induction and by (21) we have

$$y^0 = \bar{x} \in \text{co}(x^0, x^1, \dots, x^{j-1}, y^j), \tag{30}$$

which implies that, for suitable convex "weights"  $\alpha_i$ , explicitly determinable as in (25) above,

$$y^0 = \bar{x} = \sum_{i=0}^{j-1} \alpha_i x^i + \alpha_j y^j. \quad (31)$$

Rewriting (31) by using (29), we have

$$y^0 = \bar{x} = \sum_{i=0}^j \alpha_i x^i + \bar{d}, \quad (32)$$

with

$$\begin{aligned} \bar{d} &= \alpha_j d, \\ \bar{x} &= \bar{x} + \bar{d}, \end{aligned} \quad (33)$$

and

$$\bar{x} \in \text{co}(x^0, x^1, \dots, x^j) \in K, \quad \bar{d} \in R, \quad (34)$$

where, as before,  $K$  is the convex hull of the extreme points of  $P$ . Since  $\bar{d} \in R$ ,  $d^- = \beta \bar{d} \in P$ , with

$$\beta = -1/(eA\bar{d}) \geq 0. \quad (35)$$

Since  $\bar{P}$  is a polytope, the main algorithm can be applied to  $d^-$  and  $\bar{P}$  to obtain

$$d^- = \sum_{j=0}^s \bar{\beta}_j q^j, \quad (36)$$

with  $s \leq n - 1$  and

$$\sum \bar{\beta}_j = 1, \quad \bar{\beta} \geq 0,$$

where the  $q^j$ 's are extreme points of  $\bar{P}$ . Then (32) - (36) give a representation of  $\bar{x}$ . Note that  $\dim \bar{P} = n - 1$  implying  $s \leq n - 1$ .

When the lineality space  $L$  of  $P$  has positive dimension,  $P$  has no vertex. Then one can decompose  $P$  as

$$P = L + K + R, \quad (37)$$

where  $K + R$  is the decomposition of the polyhedron  $P^+ = P \cap L^\perp$ , and  $L^\perp$  is the orthogonal complement of  $L$ . We can rewrite (1) as

$$x = \sum \gamma_k v^k + \sum \alpha_i x^i + \sum \beta_j q^j, \quad (38)$$

with

$$\sum \alpha_i = 1, \quad \alpha \geq 0, \quad \beta \geq 0, \quad \gamma \geq 0,$$

and each  $v^k \in L$ . The first call to VERTEX will detect that  $L \neq 0$ , and produce a nonzero  $d \in L$ . Suppose  $\bar{x}^0$  is the input to VERTEX for this call, and at some stage the algorithm discovers a direction vector  $d \in L$ . Then we can rewrite  $\bar{x}^0$  as

$$\bar{x}^0 = \varepsilon d + x^+, \quad \text{with } d^T x^+ = 0, \quad (39)$$



where  $\varepsilon = d^T \bar{x}^0 / d^T d$  and  $x^+ \in P$ . If  $\varepsilon > 0$  ( $\varepsilon < 0$ ) then let  $(\gamma, v) \leftarrow (\varepsilon, d) [(-\varepsilon, -d)]$  and store  $(\gamma, v)$  as one term entering into the summation in (38). Now we can add  $d^T x = 0$  to the linear inequalities of  $P$ , defining a polyhedron  $P'$ , and then apply VERTEX to  $P'$  with  $x^+$  as input. Equivalently, we can continue with the current tableau by adding one row corresponding to  $d^T x = 0$ . If  $d$  is encountered during an attempt to pivot on the  $k$ th variable, then we can just declare  $k$  as a basic variable and perform one pivot on the  $k$ th element of  $d$  to obtain a tableau for the new  $P'$ . Recall that for such a vector  $d$ , the components of  $d$  on  $S$  are zero. Note also the cardinality of  $F \cap N$  decreases by one. Clearly we can encounter such  $d$  vectors  $\dim L$  times. At the conclusion of the first call to the (modified) algorithm VERTEX, we will have  $P' = P^+$ , a vertex  $x^0$  of  $P^+$ , and a decomposition of  $\bar{x}$  as

$$\bar{x} = v + x^+, \tag{40}$$

with  $v \in L$ ,  $v = \sum \gamma_k v^k$ , all  $v^k \in L$ ,  $\gamma \geq 0$ , and  $x^+ \in P^+$ . By letting  $y^0 \leftarrow x^+$ , we can continue with the main algorithm. Since (the new)  $y^0$  and  $x^0 \in P^+ \subset L^\perp$  all successive  $y^j$  and  $x^j$  will be in  $P^+$ . Then by previous arguments we obtain a representation of  $x^+$ , which together with (40) gives a representation of  $\bar{x}$ .

Note that identification of vectors  $d \in L$  is a part of the simplex method: When the ratio test fails we have a  $d \in L$ . Update of a tableau by adding a row corresponding to  $d$  is roughly equivalent to one simplex pivot. Then one might think that we have additional  $\dim L$  pivots to count. On the contrary, when we apply VERTEX to  $P^+$ , it requires  $\dim L$  less pivots. Hence the given upper bound is a very pessimistic one.

At first sight one might think that (38) is quite different than (1). Letting  $R(X)$  be the recession cone of the polyhedron  $X$  we have

$$R(P) = L + R(P^+) = L + R. \tag{41}$$

Hence

$$\sum \gamma_j v^k + \sum \beta_j q^j \tag{42}$$

of (38) is actually  $\sum \beta_j q^j$  of (1), and the number of terms entering into (42) is  $\leq n$ , for  $\dim P$  of (28) is  $\leq n - 1$ .

Let us summarize the algorithm for the most general case. Given  $\bar{x} \in P$ , a point to be represented as in (1), call VERTEX with inputs  $\bar{x}$  and  $c = \sum \{a_k; k \in I(\bar{x})\}$ . If  $P$  has a vertex then the algorithm VERTEX will produce a vertex  $x^0$ . If  $P$  has no vertex then the algorithm will recognize that fact, and by adding  $\dim L$  equations of the form  $d^T x = 0$ ,  $d \in L$ , will decompose  $\bar{x}$  as in (40); i.e., it will produce  $v = \sum \gamma_k v^k$ ,  $x^+$ , and a vertex  $x^0$  of  $P^+$ . In either case, the total work is essentially equivalent to  $n$  simplex pivots. Then the main algorithm takes over with inputs  $y^0$  ( $\bar{x}$  or  $x^+$ ), and (already computed)  $x^0$ , and tries to find a representation of  $y^0$  in the current polyhedron ( $P$  or  $P^+$ ). If it does not encounter a recession direction  $d \in R$ , then it will produce a decomposition of  $y^0$  as in (24). This will require at most  $n$  calls to VERTEX. If, on the other hand, the algorithm encounters a recession direction  $d$ , then one has the representation given in (31)–(34). Then the main algorithm is called with input  $d^-$  for the polytope  $P$ , and will produce the representation of (36) in at most  $(n - \dim L)$  calls to VERTEX. Clearly, one can combine the above representations to yield the representation required in (1) or (38). Thus, one needs at most  $2n + 1$  calls to VERTEX, so that, in total, VERTEX performs at most  $n(2n + 1)$  simplex pivots.

## ACKNOWLEDGMENT

We would like to thank the editor and referees for helpful suggestions which improved the presentation of the article.

## REFERENCES

- [1] Akgül, M., *Topics in Relaxation and Ellipsoidal Methods*, Research Notes in Mathematics No. 97, Pitman, London, 1984.
- [2] Bachem, A. and Grötschel, M., "New aspects of Polyhedral Theory," in B. Korte, Ed., *Modern Applied Mathematics*, North-Holland, New York, 1982.
- [3] Best, M. J. and Ritter, K. *Linear Programming: An Active Set Approach*, Prentice-Hall, Englewood Cliffs, 1985.
- [4] Charnes, A., Kortanek, K., and Raïke, W., "An Implementation of the Opposite Sign Purification Algorithm," System Research Memo No. 104, Northwestern University, 1964.
- [5] Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, 1963.
- [6] Goldman, A. J., "Resolution and Separation Theorems for Polyhedral Convex Sets," in H. W. Kuhn and A. W. Tucker, Eds., *Linear Inequalities and Related Systems*, Annals of Mathematics Studies No. 38 Princeton University Press, Princeton, 1956, pp. 41–56.
- [7] Grötschel, M., Lovász, L., and Schrijver, A., "The Ellipsoid Method and its Consequences in Combinatorial Optimization," *Combinatorica*, **1**, 169–197 (1981).
- [8] Grötschel, M., Lovász, L., and Schrijver, A., *Geometric Algorithms and Combinatorial Optimization*, Springer, New York, 1988.
- [9] Karmarkar, N., "A New Polynomial Algorithm for Linear Programming," *Combinatorica*, **4**, 373–395 (1984).
- [10] Maurras, J. F., Truemper, K., and Akgül, M., "Polynomial Algorithms for a Class of Linear Programs," *Mathematical Programming*, **21**, 121–136 (1981).
- [11] Murty, K. G., *Linear Programming*, Wiley, New York, 1983.
- [12] Murty, K. G. and Fathi, Y., "A Feasible Direction Method for Linear Programming," *Operations Research* **3**, 121–127 (1984).
- [13] Rosen, J., "The Gradient Projection Method for Nonlinear Programming: I.: Linear Constraints," *SIAM Journal of Applied Mathematics*, **8**, 181–217 (1960).

Manuscript received February 9, 1984

First revised manuscript April 1986

Second revised manuscript May 1986

Third revised manuscript May 1987

Fourth revised manuscript December 1987

Accepted December 31, 1987