NBER WORKING PAPER SERIES

A CONIC ALGORITHM FOR THE GROUP MINIMIZATION PROBLEM

Bruno Simeone*

Working Paper No. 159

COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE
National Bureau of Economic Research, Inc.
575 Technology Square
Cambridge, Massachusetts 02139

December 1976

*NBER Computer Research Center and Instituto Nazionale per le Applicazioni del Calcolo (Rome). Research supported in part by National Science Foundation Grant MCS75-20988 to the National Bureau of Economic Research, Inc.

ABSTRACT

A new algorithm for the group minimization problem (GP) is proposed. The algorithm can be broadly described as follows. A suitable relaxation of (GP) is defined, in which any feasible point satisfies the group equation but may have negative components. The feasible points of the relaxation are then generated in order of ascending costs by a variant of a well-known algorithm of Glover, and checked for non-negativity. The first non-negative point is an optimal solution of (GP).

Advantages and disadvantages of the algorithm are discussed; in particular the implementation of the algorithm (which can be easily extended so as to solve integer linear programming problems) does not require group arithmetics.

TABLE OF CONTENTS

1. Notations

R^n	module (i.e., additive abelian group) of all real n-vectors.
Z^n	the submodule of R^n constituted by all integer n-vectors.
R^n_+	the set $\{x: x \in R^n, x \geq 0\}$.
R^n_-	the set $\{x: x \in R^n, x \leq 0\}$.
Z^n_+	the set $\{x: x \in Z^n, x \geq 0\}$.
Z^n_-	the set $\{x: x \in Z^n, x \leq 0\}$.
$p+V$	$(p \in R^n, V \subseteq R^n)$: the set $\{p+v: v \in V\}$.
AV	(A n×n real matrix, $V \subseteq R^n$): the set $\{Av: v \in V\}$.
u_k	the k-th unit vector $(0,0,\ldots,1,\ldots,0)$, with 1 at the k-th place.
e	the vector (1,1.,,,.1)
$a \wedge b$	$(a,b \in R^n)$: the vector which is the minimum componentwise of a and b.
$[a]$	$(a \in R^n)$: the vector whose k-th component is the integer part of a_k.
$]a[$	$(a \in R^n)$: the fractional part of a, i.e. $a-[a]$.

2. Introduction

The group minimization problem is

$$(2.1)\qquad \begin{aligned} \min\quad & c_1x_1 +\dots+ c_nx_n \\ \text{s.t.}\quad & g_1x_1 +\dots+ g_nx_n = h \\ & x_k \text{ non-negative integer } (k=1.,,,.n). \end{aligned}$$

or, in short-hand notation,

$$(2.2)\qquad \begin{aligned} \min\quad & cx \\ \text{s.t.}\quad & gx=h \\ & x\varepsilon Z^n_+, \end{aligned}$$

where c is a positive vector of R^n, h is an element of a finite abelian group G (with operation +), and g is an element of G^n.

Let L be the set $\{x: x\varepsilon Z^n, gx=h\}$ and K the set $\{x: x\varepsilon Z^n, gx=0\}$. It is well known that K is a subgroup of Z^n and that L is a coset of Z^n modulo K; in other words L=p+K, where p is any particular element of L.

Let us rewrite (2.2) as

$$(2.3)\qquad \begin{aligned} & \min cx \\ & x\varepsilon Z^n_+\cap L \end{aligned}$$

and denote the feasible set $Z^n_+\cap L$ by X.

Several methods for solving (2.3) are known [4], [6], [7], [8], [9], [10], [13]. A simple solution strategy is the following: generate all vectors $x^0, x^1, \ldots \varepsilon Z^n_+$ in order of ascending costs $cx^0 \leq cx^1 \leq \ldots$, each time checking whether the last vector produced x^k belongs to L or not. The first x^k which is found to belong to L is an optimal solution of (2.3).

Actually, this is the essential idea behind a method for solving the group minimization problem, proposed by Glover [7]. This author suggests an elegant and simple algorithm for ranking the vectors in Z_+^n in order of ascending costs, provided that the cost vector c is positive. Such an algorithm will be reviewed in the Appendix. Glover also describes a more sophisticated version of the algorithm, in which only the vectors of a smaller set V, $Z_+^n \subset V \subset X$, need to be generated and ranked. Such version has been seen to be computationally quite effective [7], [12].

However, the above approach has the inherited inconvenience that the density p of the feasible set X in Z_+^n is usually low. Actually, p equals the density of L in Z^n, which in turn is equal to $\frac{1}{|G/K|}$, where G/K is the factor group of G modulo K. If $h_1,\ldots,h_n$ is any basis for the subgroup K (assumed to be of full rank), the order of G/K is equal to $|\det H|$, where H is the matrix whose columns are $h_1,\ldots,h_n$. Hence one has $p = \frac{1}{|\det H|}$.

These considerations suggest that a complementary approach to the solution of (2.3) might be undertaken, in which the vectors in some set S, $L \subset S \subset X$, are generated in order of ascending costs and checked for non-negativity until a vector in Z_+^n is found. The first such vector is then an optimal solution of (2.3).

This alternative stragegy is implemented in the algorithm which will be described in Section 4.

3. An Introductory Example

In order to introduce the basic geometric ideas underlying the method to be described in the next section, let us discuss a simple example.

Consider the group minimization problem over the cyclic group $Z_{12} = \{\bar{0}, \bar{1}, \ldots, \overline{11}\}$, given by

$$\begin{aligned} \min \quad & x_1 + 4\, x_2 \\ \text{s.t.} \quad & \overline{10}\, x_1 + \bar{9}\, x_2 = \bar{2} \\ & x_1, x_2 \text{ integers} \geq 0 \end{aligned}$$

The assignment $(x_1,x_2) \to \overline{10}\, x_1 + \bar{9}\, x_2$ defines a map from Z^2 to Z_{12}. Such a map is visualized in fig. 1, where the circle in the point (x_1,x_2) contains the corresponding element $\overline{10}\, x_1 + \bar{9}\, x_2$ of Z_{12}. The feasible set of (3.1) is the set of all points of Z^2 which are marked ⓶ and are contained in the non-negative quadrant.

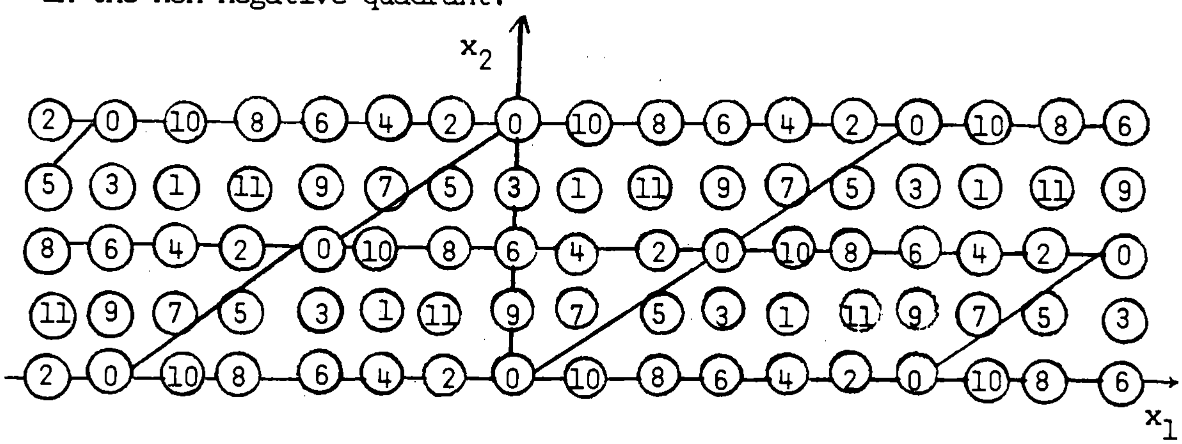

fig 1.

Let us now consider the set K of all points in Z^2 which are marked ⓪. The two points (3,2) and (6,0) belong to K; moreover, every point in K can be expressed as an integer linear combination of such two points, i.e. K is the lattice spanned by (3,2) and (6,0). Likewise, let us consider the set L of all points in Z^2 which are marked ⓶, and pick up an arbitrary element of L, say (-1,0). Then every element in L can be expressed as

$(-1,0)+v_1(3,2)+v_2(6,0)$, with v_1 and v_2 integers (fig. 1). In other words, L can be obtained through a translation of the lattice K.

Next let us introduce, for any $w = (w_1,w_2)$ in R^2, the two sets

$$\check{Z}(w) \equiv \{(w_1,w_2)+v_1(3,2)+v_2(6,0): v_1,v_2 \text{ integers} \geq 0\}$$
$$\check{R}(w) \equiv \{(w_1,w_2)+v_1(3,2)+v_2(6,0): v_1,v_2 \text{ reals} \geq 0\}$$

Taking, for example, $w = (-1,0)$, $\check{Z}(w)$ is the set of all "square" points and $\check{R}(w)$ is the shaded region in fig. 2.

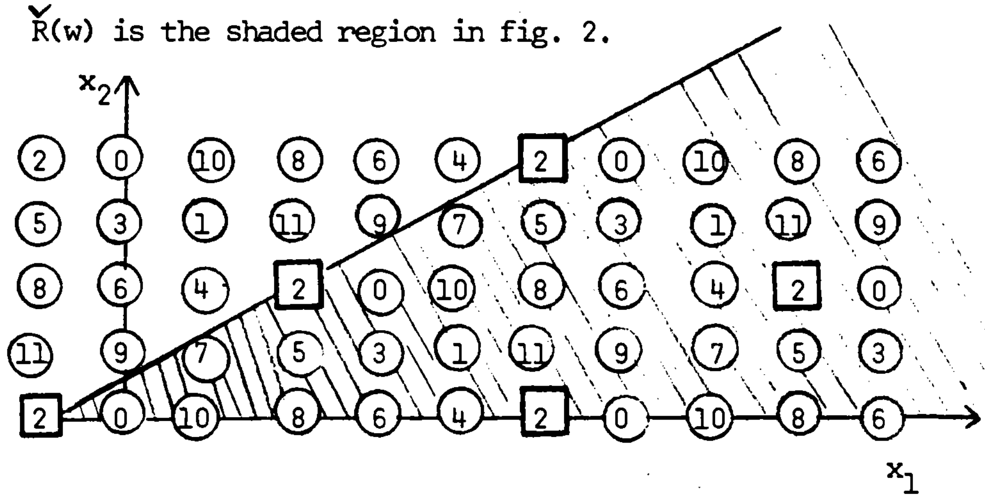

fig. 2

Suppose now we know a particular feasible solution for our problem, say (8,2) with cost 16. Let q be a point of L such that $\check{R}(q)$ contains the triangle $T \equiv \{(x_1,x_2): x_1,x_2 \geq 0 \text{ real}, x_1+4\, x_2 \leq 16\}$. It is intuitively clear that we can always find such a point q, provided that we go "sufficiently far" from T: in our case, it suffices to take $q=(-7,0)$. (fig. 3).

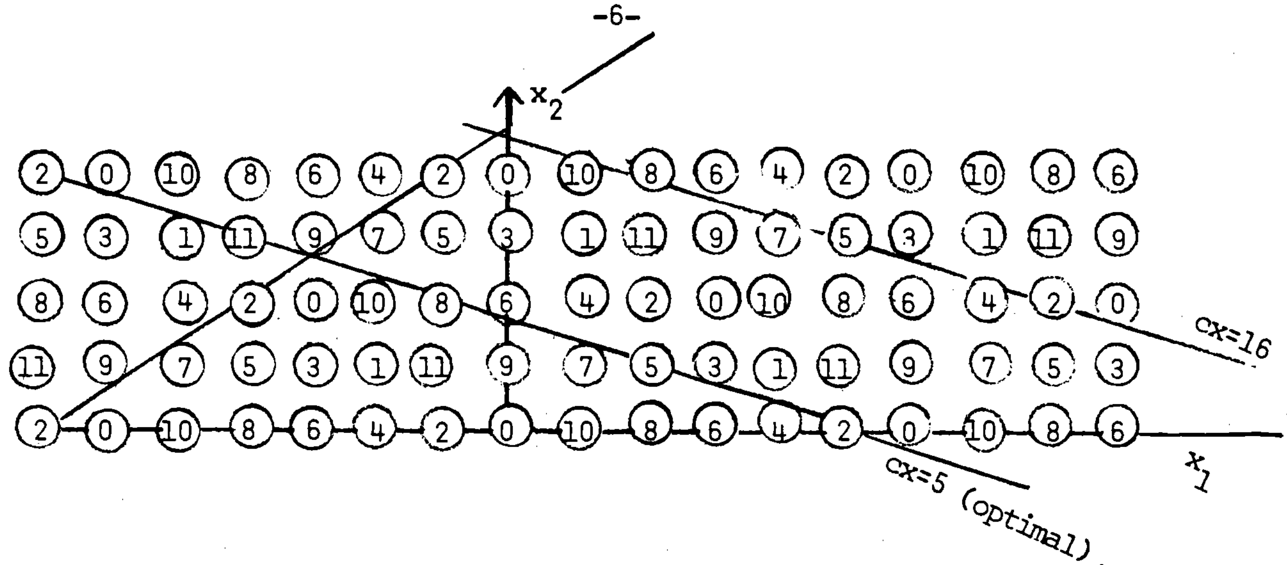

fig. 3

Any such point q has the property that all feasible solutions of (3.1) with cost $\leq$ 16 are contained in $\check{Z}(q)$. Let us rank the elements of $\check{Z}(q)$ in order of ascending costs; we get:

Costs	-7	-1	4	5	
Elements	(-7,0)	(-1,0)	(-4,2)	(5,0)	. . .

The point (5,0) is the first non-negative point of the sequence: hence it is the optimal solution of (3.1).

If we had ranked the points of the non-negative quadrant in order of ascending costs, we would have needed to generate eight points, rather than four, before discovering the optimal solution of (3.1).

4. The Conic Algorithm

In the present section we outline an algorithm for the group minimization problem (2.2), which implements the general solution strategy discussed at the end of Section 2. We maintain here the notations of that section.

It is well known (see e.g., Lekkerkerker [11] p. 20) that any full-rank subgroup of Z^n admits a non-negative basis(*). Thus, under some regularity assumptions which ensure that K has full rank, we can always find a non-negative non-singular matrix H such that $K = HZ^n$. In the next section we shall give a constructive method for deriving such a matrix H in the case - the only really interesting one for applications - that (2.2) arises as a group relaxation of a linear integer programming problem (IP).

We also assume that some particular feasible solution $\underline{P}$ of (2.2) is at hand, so that $z = cp$ is an upper bound for the optimum value of the group minimization problem. If (2.2) is a group relaxation of an IP, any feasible solution of the IP will do the job. Clearly, we are interested only in those points $x \in X$ such that $cx \leq z$. Let us denote X_z the set of all such points; X_z is contained in the simplex $T_z \equiv \{x: \ x \in R^n_+, \ cx \leq z\}$. If x is any vector in R^n, we define <u>the integer cone of vertex x spanned by H</u> as the set

$$\check{Z}(x) = x + HZ^n_+ .$$

Likewise, we define <u>the real cone of vertex x spanned by H</u> as the set

$$\check{R}(x) = x + HR^n_+$$

Clearly, $\check{R}(x) \supset \check{Z}(x)$ for all x.

(*) For the group-theoretic terminology, see e.g. Fuchs [5].

Let q be any point of L such that $\check{R}(q) \supseteq T_z$. Such a point does always exist, as we shall see in Section 6, and has the following fundamental property:

<u>Theorem 4.1</u> $\check{Z}(q) \supseteq X_z$.

<u>Proof</u> Let us first prove that, for any $q \in L$, $\check{Z}(q) = \check{R}(q) \cap L$. The inclusion $\subseteq$ is obvious; on the other hand, let $x \in \check{R}(q) \cap L$. Then $x = q + Hu = q + Ht$ for some $u \in R^n_+, t \in Z^n$. The last equality implies $u = t$, since H is non-singular. Hence $u \in Z^n_+$, $x \in \check{Z}(q)$ and $\check{Z}(q) = \check{R}(q) \cap L$.

But then, since $\check{R}(q) \supseteq T_z$, we have $\check{Z}(q) = \check{R}(q) \cap L \supseteq T_z \cap L = X_z$ ||.

Any element in $\check{Z}(q)$ has the form $x = q + Hy$, where $y \in Z^n_+$. Consequently, its cost is given by $cx = cq + cHy$. The vector $c^* = cH$ is positive, since c is positive and H is non-negative and non-singular. Hence, using the Glover's ranking algorithm reviewed in the Appendix, we can rank the elements of Z^n_+ in order of ascending c^*-costs $c^*y^0 \leq c^*y^1 \leq \ldots$. If we put $x^k = q + Hy^k$, the elements of $\check{Z}(q)$ will then be automatically ranked in order of ascending c-costs $cx^0 \leq cx^1 \leq \ldots$.

Each time a new x^k is produced, it is tested for non-negativity. The first non-negative x^k is an optimal solution of the group minimization problem (2.2). Earlier or later, a non-negative x^k must be encountered because the method exhaustes all the elements in $\check{Z}(q)$ and $\check{Z}(q) \supseteq X_z$ by theorem 4.1.

To summarize:

<u>Conic algorithm</u> (C)

0) We assume to have at hand some $p \in X$, with cost $z = cp$.

1) We build a non-negative non-singular matrix H such that $K = HZ^n_+$ (see Section 5).

2) We pick up a point $q \in L$ such that $\check{R}(q) \supseteq T_z \equiv \{x: \; x \in R^n_+, \; cx \leq z\}$. The vector q is called the <u>starting point</u> for the algorithm (see Section 6).

3) Setting $c^* = cH$, we rank the elements of Z_+^n according to ascending costs $c^*y^0 \leq c^*y^1 \leq$ .. using the ranking algorithm described in the Appendix. This induces a ranking of the elements of $\check{Z}(q)$ in order of ascending costs $cx^0 \leq cx^1 \leq$ ... . Each time we test the last x^k produced for non-negativity. The first non-negative x^k discovered is an optimal solution of our problem.

The method just described presents a basic advantage and a basic disadvantage with respect to the naïve version of Glover's algorithm (NG), discussed in Section 2. The basic advantage is that (C) generates only vectors in L, thereby getting closer to the optimal solution of (2.2) at a speed which is $|\det H|$ times higher than the speed of (NG). Put in other words, while (NG) generates the current $\bar{x}^k$ from a previous $\bar{x}^p$ (as we shall better see in the appendix) taking "small steps":

$$\bar{x}^k = \bar{x}^p + u_r ,$$

(C) generates x^k from a previous x^p taking "large steps"

$$x^k = x^p + h_r ,$$

where h_r is the r-th column of H.

The basic disadvantage is that (C) is subjected to the initial handicap to get started at a point q which may be considerably far from the feasible set X. It is clear that, the closer is q to the non-negative orthant, the smaller is the initial handicap and the more efficient is (C). In Section 6 we shall be able to derive an explicit formula for the point q which belongs to L, is such that $\check{R}(q) \supseteq T_z$ and has least distance from R_+^n. It is also apparent that (C) is sensitive to the tightness of the upper bound z to the true optimum z^*. In Section 8 we shall describe a technique for improving at low cost the available upper bound, once the non-negative matrix H has been built.

An interesting feature of (C) is that it solves the group minimization problem without employing group arithmetics at all.

Finally, we notice that, in the case that (2.2) arises as a group relaxation of an IP, the method can be easily extended in order to find an optimal solution of the IP. All that is needed is to check the non-negativity not only of the variables x (which, as it is well known, are the non-basic variables in the optimal solution of the continuous relaxation of the IP), but also of the corresponding basic variables. The method appears then to be well in line with the strategy, emphasized by Wolsey in [14], of solving the IP by finding the k-th best solution of its group relaxation.

5. Finding a Non-Negative Basis

In the present section, we show how to construct a non-negative integer matrix H such that $K=HZ^n$, in the particular case that (2.2) is the group relaxation of an IP. Let the IP be

$$\text{(5.1)} \qquad \begin{array}{l} \min dt \\ At \leq b, \quad t \in Z_+^n \end{array}$$

or, adding slacks s,

$$\text{(5.2)} \qquad \begin{array}{l} \min dt \\ At+I_m s = b \\ t \in Z_+^n, \; s \in Z_+^m, \end{array}$$

where I_m is the identity matrix of order m. All data are assumed to be integer.

Dropping the integrality requirements on t and s, one obtains a linear programming problem (LP). Let us partition its matrix $[A \mid I_m]$ as $[B \mid N]$, where B and N are formed, respectively, by the basic and by the non-basic columns of $[A \mid I_m]$ in an optimal basic solution of the LP. Accordingly, we partition the vector $[t \mid s]$ as $[y \mid x]$, where y and x are the basic and the non-basic variables, respectively. Thus (5.2) becomes

$$
\begin{aligned}
&\bar{z} + \min cx \\
(5.3)\qquad &By + Nx = b \\
&y \in Z_+^m,\ x \in Z_+^n
\end{aligned}
$$

where $\bar{z}$ is the optimal value of the LP and c is the vector of reduced costs. One has $c \geq 0$: we assume that c is strictly positive.

Let π be the natural homomorphism of Z^m onto the factor group $G = Z^m/_{<B>}$, where $<B>$ is the subgroup of Z^m spanned by the columns of B, i.e., $<B> = BZ^m$. Applying π to both sides of the equation in (5.3), we obtain the group relaxation of (5.1):

$$
\begin{aligned}
&\bar{z} + \min cx \\
&\pi(Nx) = \pi b\ , \qquad x \in Z_+^n
\end{aligned}
$$

which, aside from the constant $\bar{z}$, can be written as (2.1) by putting $h = \pi b$, $g_j = \pi a_j$, where a_j is the j-th column of N.

Next, let us define an n×n matrix F as follows. Each row f^h of F corresponds to the non-basic variable x_h. If x_h is a structural variable t_j, we define f^h to be the row unit vector u_j^T; if x_h is a slack variable s_i, we define f^h to be minus the i-th row of A : $f^h = -u_i^T A$. Notice that the matrix F so defined is all-integer.

Example If the matrix $[A \mid I_m]$ is

$$
\begin{array}{ccc|cc}
t_1 & t_2 & t_3 & s_1 & s_2 \\
\end{array}
$$

$$
\left[\begin{array}{ccc|cc}
1 & 0 & 2 & 1 & 0 \\
-1 & 2 & 1 & 0 & 1
\end{array}\right]
$$

and the basic variables are t_1, s_1, then F is given by

$$\begin{matrix} t_2 \\ t_3 \\ s_2 \end{matrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -2 & -1 \end{bmatrix} .$$

If K, as in Section 2, is the set $\{x: x \in Z^n, gx=0\}$, F has the following fundamental property (which is closely related to a result of Bowman [2])

Theorem 5.1 $K=FZ^n$.

Proof Let $x \in Z^n$; x belongs to K if and only if $\pi(Nx)=0$, i.e., if and only if

$$(5.4) \qquad By+Nx=0 \ , \qquad \exists y \in Z^m,$$

which can be rewritten as

$$At+s=0 \qquad t \in Z^n, \ s \in Z^m$$

or as

$$\begin{aligned} t &= I_n v \\ s &= -Av \end{aligned} \qquad \exists v \in Z^n,$$

and finally, after a suitable partition of the rows of the matrix $\begin{bmatrix} I_n \\ \hline -A \end{bmatrix}$, as

$$(5.5) \qquad \begin{aligned} y &= D_v \\ x &= F'v \end{aligned} \qquad \exists v \in Z^n.$$

But, by construction, F' is exactly equal to the matrix F above introduced, so we can drop the "prime" in (5.5). Moreover, since $[A \vdots I_m] = [B \vdots N]$, $\begin{bmatrix} I_n \\ \hline -A \end{bmatrix} = \begin{bmatrix} D \\ \hline F \end{bmatrix}$ and $[A \vdots I_m] \begin{bmatrix} I_n \\ \hline -A \end{bmatrix} = 0$, one must have

$$(5.6) \qquad BD+NF = 0 \ .$$

Now, let x be any element in K. By the above reasoning, there exists a $v \varepsilon Z^n_+$ such that $x=Fv$: thus $K \subseteq FZ^n$. Conversely, let x be such that $x=Fv$ for some $v \varepsilon Z^n$; if se put $y=Dv$, y belongs to Z^m, and $By+Nx = BDv+NFv = (BD+NF)v = 0$ by (5.6).

Since (5.4) is equivalent to "$x \varepsilon K$", one has $FZ^n=K$ and the theorem is proved||.

We make the assumption that F is non-singular. Then the columns of F constitute a basis for K, which has therefore full rank n.

A more intrinsic method for finding a basis of K, which only requires the knowledge of the group elements $g_1, g_2, \ldots, g_n$, is described in Bell [1], p. 106.

In any case, with suitable elementary column operations, we can reduce F to the Hermite canonical form, i.e., we can find an Hermite matrix M and an unimodular matrix P such that $F=MP$ (see Bradley [3]).

The matrix M is lower triangular; its diagonal elements are positive and its off-diagonal elements are non-positive. By adding a suitable multiple of the last column to the second last one, we can replace the second last column with a non-negative one (actually, with a column whose lower diagonal elements are positive); by adding a suitable multiple of the new second last column to the third last one, we can replace the third last column with a non-negative one - again, with positive lower diagonal elements - and so on, until all the lower diagonal elements have become positive. The process, which is schematized in figure 4, amounts to post-multiply M by an unimodular matrix Q. The matrix $H=MQ$ is

$$\begin{bmatrix} + & 0 & 0 & 0 \\ - & + & 0 & 0 \\ - & - & + & 0 \\ - & - & - & + \end{bmatrix} \rightarrow \begin{bmatrix} + & 0 & 0 & 0 \\ - & + & 0 & 0 \\ - & - & + & 0 \\ - & - & + & + \end{bmatrix} \rightarrow \begin{bmatrix} + & 0 & 0 & 0 \\ - & + & 0 & 0 \\ - & + & + & 0 \\ - & + & + & + \end{bmatrix} \rightarrow \begin{bmatrix} + & 0 & 0 & 0 \\ + & + & 0 & 0 \\ + & + & + & 0 \\ + & + & + & + \end{bmatrix}$$

Figure 4

non-negative and is related to F by the identity H=FPQ. Since both P and Q are unimodular, their product PQ is unimodular as well. It follows that, since F is a basis for K, also H is a basis, and is therefore the matrix we were looking for.

The above procedure has been presented mainly for expository purposes, and should not be regarded as the most efficient one for converting F into a non-negative equivalent matrix H: thus it is not really necessary to go through the intermediate stage of an hermitian matrix.

6. Finding a Starting Point

In this section, we are faced with the problem of finding a starting point for the conic algorithm. Namely, we want to solve the following:

Problem 1: Find a $q\varepsilon L$ such that $\check{R}(q)\supseteq T_z$ and q has least distance from R^n_+. First of all, we observe that, in view of the convexity of $\check{R}(q)$, one has $\check{R}(q)\supseteq T_z$ if and only if $\check{R}(q)$ contains the extreme points of T_z, i.e., 0, $\frac{z}{c_1} u_1, \ldots, \frac{z}{c_n} u_n$. Hence we are led to consider the following

Problem 2: Given any finite subset $A=\{a_s,\ldots,a_s\}$ of R^n, find a $w\varepsilon L$ such that $\check{R}(w)\supseteq A$.

It is convenient to introduce, in parallel with $\check{R}(x)$, the set

$$\hat{R}(x) = x+HR^n_-$$

where x is an arbitrary element of R^n.

The following relationships will be needed later on:

Prop 6.1 a) $x\varepsilon\check{R}(y)\Leftrightarrow y\varepsilon\hat{R}(x)$

b) $x\varepsilon\check{R}(y)\Leftrightarrow\check{R}(x)\subseteq\check{R}(y)$

b') $x\varepsilon\hat{R}(y)\Leftrightarrow\hat{R}(x)\subseteq\hat{R}(y)$

Proof: a): It suffices to observe that $x = y+Hv \Longleftrightarrow y = x+H(-v)$

b): The implication $\Longleftarrow$ is obvious. The reverse implication $\Longrightarrow$ follows from the fact that, if $x = y+Hu$ and $t = x+Hv$, with $u, v \geq 0$, then $t = y+H(u+v)$, with $u+v \geq 0$.

b'): as b). ||

We solve Problem 2 in two steps. First, we show that we can always find an x in R^n such that $\check{R}(x) \supseteq A$. Indeed, if such an x does exist, by Prop 6.1 a) one must have $x \in \hat{R}(a_i)$, $i=1,\ldots,s$, and hence $x \in \hat{R}(a_1) \cap \ldots \cap \hat{R}(a_s)$. We show that such an intersection is not empty, and actually is an $\hat{R}(a)$ for a suitable $a \in R^n$.

Theorem 6.2 $\hat{R}(a_1) \cap \ldots \cap \hat{R}(a_s) = \hat{R}(a)$, where

$$(6.1) \qquad a = H(H^{-1}a_1 \wedge \ldots \wedge H^{-1}a_s).$$

Proof Let a be defined by (6.1). Then, for each $i=1,\ldots,s$, there exists some $v_i \in R^n_-$ such that $a = H(H^{-1}a_i+v_i)$; one then has $a = a_i+Hv_i$, i.e., $a \in \hat{R}(a_i)$ for each i. Thus, by Prop 6.1, $\hat{R}(a) \subseteq \hat{R}(a_1) \cap \ldots \cap \hat{R}(a_s)$. Conversely, let $x \in \hat{R}(a_1) \cap \ldots \cap \hat{R}(a_s)$. Then there exist $z_1,\ldots,z_s \in R^n_-$ such that $x = a_1+Hz_1=\ldots=a_s+Hz_s$ or, equivalently, $H^{-1}x = H^{-1}a_1+a_1=\ldots=H^{-1}a_s+z_s$, which implies $H^{-1}x \leq H^{-1}a_i$ for each i, and hence $H^{-1}x \leq H^{-1}a_1 \wedge \ldots \wedge H^{-1}a_s$, i.e., $H^{-1}x=H^{-1}a_1 \wedge \ldots \wedge H^{-1}a_s+v$ for some $v \in R^n_-$. Thus, taking into account (6.1), one has $x=a+Hv$, which proves that $\hat{R}(a_s) \cap \ldots \cap \hat{R}(a_s) \subseteq \hat{R}(a)$ ||.

Next, let us define the cell of vertex x spanned by H as the set

$$C(x) = \{x+Hr : r \in R^n, \ 0 \leq r < e\} ,$$

where x is any point of R^n.

It is intuitively clear that the collection of all cells C(y), when y varies within L, constitutes a partition of the whole space R^n. The next theorem formally proves such assertion and gives also, for each $x \in R^n$, an explicit formula for the vertex x_L of the unique cell $C(x_L)$ which contains x.

Theorem 6.3 Given any $x \in R^n$, there exists a unique $x_L \in L$ such that $x \in C(x_L)$. Such an x_L is given by

$$(6.2) \qquad x_L = p + H[H^{-1}(x-p)].$$

where p is any element of L.(*)

Proof If x_L is the point defined by (6.2), one has (see Section 1 for the notations): $X_L+H]\,H^{-1}(x-p)[\, = p+HH^{-1}(x-p)=x$, which proves that $x \in C(x_L)$.

Let now y be any element of L such that $x \in C(y)$. One has $y=p+Ht$ for some $t \in Z^n$ and $x=y+Hr$ for some $0 \leq r < e$. Combining the above equalities one gets $H^{-1}(x-p)=t+r$, which implies $t=[H^{-1}(x-p)]$, $r=]H^{-1}(x-p)[$ and hence $y=x_L$. Thus the uniqueness of x_L is established ||.

Combining theorems 6.2 and 6.3, we obtain a constructive answer to Problem 2:

Corollary 6.4 The point

$$(6.3) \qquad w = p+H[H^{-1}a_1 \wedge \ldots \wedge H^{-1}a_s - H^{-1}p]$$

belongs to L and is such that $\check{R}(w) \supseteq A$.

Proof By theorem 6.2, the point a defined by (6.1) is such that $\check{R}(a) \supseteq A$. Thus, if we put $w=a_L=p+H[H^{-1}(a-p)]$, w belongs to L and, by theorem 6.3, $a \in C(w) \subseteq \check{R}(w)$. Hence, by Prop 6.1 b), $\check{R}(a) \subseteq \check{R}(w)$, so that $\check{R}(w) \supseteq A$. Taking into account (6.1), the expression of w can be simplified to (6.3) ||.

(*) It is easy to see that the r.h.s. of (6.2) does not change if we replace p by any other $p' \in L$.

It is important to notice the following extremality property of the point w given by (6.3):

Theorem 6.5 If y is any element of L such that $\check{R}(y)\supseteq A$, then $y\in\hat{R}(w)$.

Proof Since $y\in L$, one must have

(6.4) $$y = p+Ht, \qquad \exists t\in Z^n .$$

Combining (6.4) with the identity $w = p+H[H^{-1}(a-p)]$ which defines w, one obtains

(6.5) $$H^{-1}(y-w) = t -[H^{-1}(a-p)] = [t-H^{-1}(a-p)]$$

On the other hand, since $\check{R}(y)\supseteq A$, it must be $y\in\hat{R}(a)$ by theorem 6.2, i.e.,

(6.6) $$y = a+Hv, \qquad \exists v\in R^n_-$$

From (6.4) and (6.6) we obtain

$$t-H^{-1}(a-p) = v\leq 0.$$

Comparing this inequality and (6.5), one has $H^{-1}(y-w)\leq 0$ or, equivalently, $y\in\hat{R}(w)$ ||.

Remark In deriving the above results, use has never been made of the non-negativity of H. Thus such results hold for an arbitrary non-singular matrix H.

We have now at hand an explicit answer to our Problem 1. Let us state it as the conclusive theorem of this section.

Theorem 6.6 The point

(6.7) $$q = p+H[0\wedge\frac{z}{c_1} H^{-1}u_1\wedge\ldots\wedge\frac{z}{c_m} H^{-1}u_n - H^{-1}p]$$

is, among all the points $y\in L$ such that $\check{R}(y)\supseteq T_z$, the one having least distance from R^n_+.

Proof The first two assertions of the theorem, i.e., $q\varepsilon L$ and $\check{R}(q)\supseteq T_z$, follow directly from Cor. 6.4, specialized to the set $A=\{0, \frac{z}{c_1}u_1 \ldots, \frac{z}{c_n} u_n\}$. If y is any point in L such that $\check{R}(y)\supseteq T_z$, then $y\varepsilon\hat{R}(q)$ by theorem 6.5. Since H is non-negative, one must then have $y\leq q$, which implies $\mathrm{dist}(y,R^n_+)\geq \mathrm{dist}(q,R^n_+)$ ||.

As expected, the point q depends on the upper bound z : q=f(z), and the (vector) function f is non-decreasing. This confirms the goemetric intuition that, the tigher is the upper bound z, the closer is the point q = f(z) to the non-negative orthant.

7. Improving the Initial Upper Bound

As we have seen in Section 4, the effectiveness of the conic algorithm is enhanced by the knowledge of a good initial upper bound z. Fortunately, once a non-negative basis H for k is available, it is usually possible to improve at low cost the upper bound by a simple steepest descent procedure. If x, $y\varepsilon X$, define y to be a predecessor of x whenever

$$x = y+h_j \quad \text{for some } j, \quad 1\leq j\leq n,$$

where h_j is the j-th column of H. Since the vector $c^* = cH$ is positive, we always have $cx>cy$. The above-mentioned steepest descent procedure can be described as follows. Starting from the known feasible solution p, we look, among the feasible predecessors of p, for the one, x, with minimal cost cx. Then we replace p with x and go on, until we find an y without feasible predecessors. Alternatively, the procedure can be described in the following way. Re-number the variables $x_1,\ldots,x_n$ in such a way that $c_1\geq c_2\geq\ldots\geq c_n$. Then set

$$
\begin{aligned}
p_0 &= p \\
k_1 &= \max\{k : k\in Z_+,\ p_0 - kh_1 \geq 0\} \\
p_1 &= p_0 - k_1h_1 \\
k_2 &= \max\{k : k\in Z_+,\ p_1 - kh_2 \geq 0\} \\
&\ldots \\
k_n &= \max\{k : k\in Z_+,\ p_{n-1} - kh_n \geq 0\} \\
p_n &= p_{n-1} - k_nh_n
\end{aligned}
$$

It is easily seen that $p_j \in X$ for all j; moreover, $cp_0 > cp_1 > \ldots > cp_n$ and p_n has no feasible predecessors. Thus the upper bound is improved from $z = cp_0$ to $z' = cp_n$.

8. Computational Complexity of the Conic Algorithm: A Theoretical Estimate

In this section we shall give a theoretical estimate for the number of points which need to be generated by the conic algorithm (C) before finding an optimal solution for (2.2). The number of such points is approximately equal to the density of L in Z^n multiplied for the volume of the simplex $V \equiv \{x : x \in \check{R}(q),\ cx \leq z^*\}$, where q is the starting point of the algorithm and z^* is the optimum value of (2.2).

Since $V = q + HY$, where $Y \equiv \{y : y \in R^n_+,\ cHy \leq z^* - cq\}$, one has (see [11], p. 23):

$$\text{vol } V = |\det H| \text{ vol } Y = |\det H| \frac{(z^* - cq)^n}{2(ch_1 ch_2 \ldots ch_n)} .$$

On the other hand, we have seen in Section 2 that the density of L in Z^n is equal to $\frac{1}{|\det H|}$. Hence the number of points generated by (C) is approximately equal to

$$\nu_C \simeq \frac{1}{2} \frac{(z^* - cq)^n}{ch_1 . ch_2 \ldots ch_n}$$

For the sake of comparison, the naive Glover's algorithm, mentioned in Section 2, generates on the whole a number of points ν_{NG} which is approximately equal to the volume of the simplex $T_{z*} \equiv \{x: x \in R^n_+,\ cx \leq z^*\}$, i.e .

$$\nu_{NG} \simeq \text{vol } T_{z*} = \frac{1}{2} \frac{z^{*n}}{c_1 \cdot c_2 \cdots c_n}$$

The ratio between ν_C and ν_{NG} is therefore approximately equal to

$$\frac{\nu_C}{\nu_{NG}} \simeq \left(1 - \frac{cq}{z^*}\right)^n \frac{c_1 \cdot c_2 \cdots c_n}{ch_1 \cdot ch_2 \cdots ch_n}$$

where q is given by formula (6.7): $q = p + H[0 \wedge \frac{z}{c_1} H^{-1} u_1 \wedge \cdots \wedge \frac{z}{c_n} H^{-1} u_n - H^{-1} p]$

The above formula further supports the qualitative remarks which were made in Section 4 about the relative advantages and disadvantages of the two methods.

APPENDIX

Review of Glover's Ranking Algorithm

The method described in Section 4 requires a procedure which, for a given positive cost vector c, ranks the vectors of Z_+^n in order of ascending costs $0 = cx^0 \leq cx^1 \leq \ldots$.

In this appendix we shall review an algorithm due to Glover [7], which accomplishes such a task.

The first element x^0 of the sequence is obviously the 0 vector. Each x^k in the sequence is obtained by increasing some component r of a previous vector x^p by one unit:

$$x^k = x^p + u_r \qquad 0 \leq p < k, \qquad 1 \leq r \leq n.$$

Of course, p and r depend on k, $p=p^k$ and $r=r^k$. During the k-th iteration, p^k and r^k are generated by means of two auxiliary vectors, t^k and c^k. The vector t^k, which may be called the transition vector, is built in such a way that in its r^k-th component $t^k_{r^k}$ we are always certain to find the correct index p^k.

Accordingly, the vector c^k, which may be called the next-cost vector, is built in such a way that $c^k_{r^k} = cx^k$. The vectors t^{k+1} and c^{k+1} are obtained from t^k and c^k, respectively, by modifying only the r^k-th component of such vectors.

Here are the details of the algorithm.

Step 0. (Initialization) Set $k=0$, $x^0=0$, $t^1=0, c^1=c$

Step 1. Increment k by one. Let r^k be such that $c^k_{r^k} = \min_{1 \leq j \leq n} c^k_j$ (if there is more than one such r^k, take the smallest one).

Step 2. Set $p^k = t_{r^k}^k$

Step 3. Set $x^k = x^{p^k} + u_{r^k}$ and $z^k = cx^k$

Step 4. Set $t_j^{k+1} = \begin{cases} t_j^k & \text{if } j \neq r^k \\ \min\{h: p^k < h \leq k,\ r^h \geq r^k\} & \text{if } j = r^k \end{cases}$

Step 5. Set $c_j^{k+1} = \begin{cases} c_j^k & \text{if } j \neq r^k \\ z^{t_j^{k+1}} + c_j & \text{if } j = r^k \end{cases}$. Go to Step 1.

The sequence $x^0, x^1, \ldots$ generated by the algorithm has the following properties:

1) It is exhaustive, i.e. $\{x^0, x^1, \ldots\} = Z_+^n$

2) $h < k \Longrightarrow cx^h \leq cx^k$

3) $h < k$ and $cx^h = cx^k \Longrightarrow x^h$ is lex-greater than x^k

The last property implies that there are no repetitions in the sequence, i.e., $h \neq k \Longrightarrow x^h \neq x^k$. For the proofs of the above properties, one can consult [7]; we merely observe here that the vector t^k has the crucial property:

$$cx^{t_j^k} = \min\{cx^h : h < k,\ cx^h \geq cx^k - c_j\} \quad \text{for all } j = 1, \ldots, n.$$

REFERENCES

[1] Bell, D.E. "The resolution of duality gaps in discrete optimization" ORC, MIT, Tech. Rep. 81, Aug 1973.

[2] Bowman, V.J. "The structure of integer programs under the Hermitian normal form". Oper. Res. 22 (5), pp. 1066-1080, Sept 1974.

[3] Bradley, G. "Equivalent integer programs", Proc. 5th Intern. Conf. of Oper. Res., Venice 1969 (Lawrence, J. edt), Tavestock Pub., London.

[4] Burdet, C.A. and Johnson, E.L. "A subadditive approach to the group problem of integer programming", Math. Progr. Studies 2, pp. 51-71, 1974.

[5] Fuchs, L. Abelian groups, Pergamon Press, New York, 1960.

[6] Gallo, G., Martino, E. and Simeone, B. "Group optimization algorithms and some numerical results via a branch and bound approach", Atti Convegno AICA su "Tecniche di Simulazione ed Algoritmi", Milan 1972, Tamburini Pub.

[7] Glover, F. "Integer programming over a finite additive group", SIAM J. on Control, 7, pp. 213-231, 1969.

[8] Gomory, R.E. "Some polyhedra related to combinatorial problems", Lin. Alg. and Appl. 2, pp. 451-558, 1969.

[9] Hu, T.C. "On the asymptotic integer algorithm", Lin. Alg. and Appl. 3, pp. 279-294, 1970.

[10] Johnson, E.L. "Cyclic groups, cutting planes and shortest paths" Mathematical Programming, (Hu, T.C. and Robinson, S. eds.) Academic Press, New York, 1973.

[11] Lekkerkerker, C.G. Geometry of Numbers, North Holland, Amsterdam, 1969.

[12] Northup, W.D. Personal communication, Feb 1976.

[13] Shapiro, J.F. "Group theoretic algorithms for the integer programming problem - II: Extension to a general algorithm" Oper. Res 16, pp. 928-947, 1968.

[14] Wolsey, L.A. "Generalized dynamic programming methods in integer programming" Math. Progr. 4, pp. 222-232, 1973.