AN ABSTRACT OF THE THESIS OF

GEORGE EDWARD SEYMORE for the DOCTOR OF PHILOSOPHY
(Name) (Degree)

in STATISTICS presented on March 30, 1971
(Major) (Date)

Title: INTERVAL INTEGER PROGRAMMING

Abstract approved: Redacted for Privacy
Dr. Donald Guthrie

This paper is concerned with the integer programming problem

$$\text{maximize} \quad cx$$
$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x_j \ \text{integer} \ \forall j,$$

where A is a matrix of full row rank and all elements of c, A, b^- and b^+ are integers. Notice that the elements of the vector x are unrestricted in sign. The research is motivated by the fact that continuous linear programs of the form

$$\text{maximize} \quad cx$$
$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x_j \ \text{real} \ \forall j$$

have explicit solutions when A has full row rank and c is in the row space of A; since many existing integer programming algorithms require the solution to the linear program associated with the integer program of interest (some requiring the solutions to a

sequence of linear programs) it seems natural to inquire whether it might be possible to solve the above integer programming problem in a relatively efficient manner by capitalizing on our ability to solve the associated linear program so easily.

Two algorithms for solving the above integer programming problem are derived. The first is a condensed form of Gomory's cutting plane method, and the second is a partial enumeration scheme.

Interval Integer Programming

by

George Edward Seymore

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

June 1971

APPROVED:

Redacted for Privacy

Professor of Statistics

in charge of major

Redacted for Privacy

Chairman of Department of Statistics

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 30, 1971

Typed by Clover Redfern for George Edward Seymore

ACKNOWLEDGMENT

The author wishes to express his thanks to his wife, Marcia, for her remarkable patience, to his major professor, Dr. Donald Guthrie, for his encouragement and confidence, and to the United States Department of Health, Education, and Welfare for support of this research under Public Health Service grant 2 P10 ES00210-04.

TABLE OF CONTENTS

LIST OF FIGURES

INTERVAL INTEGER PROGRAMMING

I. INTRODUCTION

The standard linear programming problem, written

$$\begin{aligned} &\text{maximize} \quad cx \\ &\text{subject to} \quad Ax \leq b, \quad x \geq 0, \end{aligned}$$

is well understood; the theory is extensively developed and a number of algorithms, all related to the original simplex method of Dantzig, are known to solve problems of this form when they are properly formulated. All these algorithms are iterative in nature, however, and in cases of real interest may be tedious to solve. Ben-Israel and Charnes (1968) discuss a particular class of linear programming problems for which explicit solutions are readily available; these problems, of the form

$$\begin{aligned} &\text{maximize} \quad cx \\ &\text{subject to} \quad b^- \leq Ax \leq b^+, \quad A \text{ of full row rank}, \end{aligned}$$

are called <u>interval linear programming problems</u> in the later terminology of Ben-Israel and Robers (1970). Notice that x is unrestricted in sign. It is immediately clear that the interval linear programming problem can be cast as a standard linear program, so that it could be solved iteratively by conventional methods. An appreciation of the economy of Ben-Israel and Charnes' approach, though, can be gained by considering the case when A is nonsingular; then

$$cx = c(A^{-1}A)x$$
$$= (cA^{-1})(Ax),$$

so that letting $z = Ax$ the original problem becomes

$$\text{maximize} \quad (cA^{-1})z$$
$$\text{subject to} \quad b^- \leq z \leq b^+.$$

The solution to the transormed problem is immediate: if the jth element of the vector cA^{-1} is positive (negative) then the optimal z_j^* is $b_j^+ (b_j^-)$. If the jth element of cA^{-1} is zero than z_j^* may take on any arbitrary value between b_j^- and b_j^+. Thus we see that this particular interval program is solved by inverting A and noting the signs of the elements of the vector cA^{-1}. Setting $x^* = A^{-1}z^*$ yields an optimal solution to the original interval program. Solving the same problem (cast as a standard linear program) with a general purpose linear programming code would at the very least involve the inversion of a matrix with twice as many rows and columns as A, so the computational advantage of Ben-Israel and Charnes' method is clear. When A is singular, a similar transformation of the problem may be made using a simple generalized inverse of A; this is discussed in Appendix 2 and is seen to require approximately the same computational effort.

The optimal solution to any linear program is a vector of real numbers and in general it is not a vector of integers; in many problems of applied interest, though, noninteger solutions are meaningless. For example, any problem dealing with an optimal vector of indivisible commodities must have an integer solution; an optimal solution calling for the purchase of 3 1/2 locomotives is simply unusable. As one might reasonably expect, the addition of the restriction that the optimal solution to a linear program be a vector of integers generally makes the problem appreciably more difficult than the original continuous problem. These discrete problems are called integer linear programming problems, or simply integer programs.

This paper is concerned with finding solutions to the discrete problem

$$\begin{aligned}&\text{maximize} \quad cx\\ &\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x_j \text{ integer } \forall j,\end{aligned}$$

where all elements of A, b^-, b^+ and c are integers; we may call this integer program the <u>interval integer program</u> (IIP). Several algorithms are known for the solution of the standard integer programming problem

$$\text{maximize} \quad cx$$
$$\text{subject to} \quad Ax \le b, \quad x \ge 0, \quad x_j \text{ integer } \forall j,$$

and of course any of these would solve the IIP if it were cast in standard form. However, these algorithms are as a general rule very cumbersome and slow to converge, and almost all involve solving the corresponding linear continuous linear program

$$\text{maximize} \quad cx$$
$$\text{subject to} \quad Ax \le b, \quad x \ge 0;$$

some require that a potentially large number of linear programs be derived and solved sequentially. Since a large amount of computational effort in solving the integer programming problem is expended on solving related continuous linear programs, it seems natural to inquire whether it might be possible to solve the IIP in a relatively efficient manner by capitalizing on our ability to solve the continuous interval linear program so easily. The explicit solution to the interval linear program will be an important part of the two methods derived here.

In Chapter II we derive a very much condensed form of Gomory's cutting plane method for solving the IIP when A is nonsingular, and a rather less condensed version for the case when A is singular. In both cases the derived scheme requires a smaller tableau and less computational effort than the same interval integer program expressed in standard form and solved with the usual cutting plane algorithm. We show that one formulation of the IIP as a standard integer program allows several portions of the optimal tableau for the continuous linear program to be dropped, and that the remaining portion of the tableau required to generate the Gomory cuts can be determined directly from a generalized inverse of A (or the inverse if A is nonsingular).

In Chapter III we derive an enumerative scheme for solving the IIP for the case when A is nonsingular. This method is unrelated to the cutting plane algorithm and has the desirable property that an upper bound can be computed on the number of points that must be investigated before an optimal solution is obtained. This upper bound depends only on the dimensions and the determinant of A, so that the bound is not derived from the fact that each constraint is two-sided. We derive the computational algorithm and show that if A is singular but of full row rank with c in the row space of A then the IIP can be expressed as an IIP with a nonsingular constraint matrix.

The following notation will be used in the remainder of this paper:

$\mathbb{R}$ = the real numbers

$\mathbb{R}^+$ = the nonnegative real numbers

$\mathbb{Z}$	= the ring of integers
$\mathbb{Z}^+$	= the nonnegative integers
S^n	= the n-fold cartesian product of the set S
max	= maximize
min	= minimize or minimum
$\mathbb{R}^{m,n}$	= the vector space of m x n matrices over $\mathbb{R}$
$\mathbb{Z}^{m,n}$	= the module of m x n matrices over $\mathbb{Z}$
A'	= the transpose of the matrix A
$[x]^I$	= the greatest integer less than or equal to x
$N(A)$	= the null space of the matrix A.

Also, the word "integer" is sometimes used as an adjective to describe matrices; e.g., the phrase "A is integer" should be interpreted to mean that all elements of A are integers.

II. A CONDENSED CUTTING PLANE METHOD

In this chapter we will use Gomory's cutting plane algorithm to derive a scheme for solving the interval integer programming (IIP) problem. For ease of exposition the matrix A will first be assumed nonsingular; the full row rank singular case will then follow directly. We will use a particular formulation to cast the IIP as a conventional integer programming problem and then construct the extended Tucker tableau needed to begin Gomory's cutting plane algorithm; we will then show that, by modifying the pivoting rules, the size of the tableau (and hence the computational effort required to keep the tableau updated) can be significantly reduced. It will be shown that the absence of nonnegativity restrictions on the variables allows us to regain primal feasibility without an explicit dual pivot operation in the event that a basic structural variable becomes negative after the introduction of a new Gomory cut. A convergence proof is not required, since the method we derive is simply a special case of Gomory's algorithm, which is known to converge finitely.

Let $A \in \mathbb{Z}^{m,m}$ be nonsingular and let b^-, b^+ and $c \in \mathbb{Z}^m$ with $b^- \leq b^+$ elementwise. Consider the IIP

$$(2.1) \qquad \max \quad cx$$
$$\text{subject to} \quad b^- \le Ax \le b^+, \quad x \in \mathbb{Z}^m,$$

equivalently written as a conventional integer program

$$(2.2) \qquad \max \quad cx - cz$$
$$\text{subject to} \quad Ax - Az \le b^+,$$
$$-Ax + Az \le -b^-,$$
$$x, z \in (\mathbb{Z}^+)^m.$$

Inserting the appropriate slack variables, we see that problem (2.2) may also be expressed as

$$(2.3) \qquad \max \quad cx - cz$$

subject to

$$\begin{bmatrix} A & -A & I & 0 \\ 0 & 0 & I & I \end{bmatrix} \begin{bmatrix} x \\ z \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} b^+ \\ b^+ - b^- \end{bmatrix},$$

$$x, z, s_1 \text{ and } s_2 \in (\mathbb{Z}^+)^m.$$

Notice that the last m constraints are simply upper bounds on the vector s_1 of slacks on the first m constraints; we shall make reference to this observation later.

The essence of the derivation of the condensed cutting plane method is to cast the original problem (2.1) in the form (2.3) and to apply Gomory's algorithm to the new formulation; we then show that most of the simplex tableau required for Gomory's method can either be ignored or dealt with implicitly. Appendix 1 contains a summary of Gomory's method; we shall employ the notation and tableaus given there.

The first step in Gomory's algorithm is the solution of the continuous linear program corresponding to (2.3), namely

(2.4) $\max \quad cx - cz$

subject to

$$\begin{bmatrix} A & -A & I & 0 \\ 0 & 0 & I & I \end{bmatrix} \begin{bmatrix} x \\ z \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} b^+ \\ b^+ - b^- \end{bmatrix} ,$$

$$x,\ z,\ s_1 \text{ and } s_2 \in (\mathbb{R}^+)^m .$$

Since (2.4) is simply the continuous interval program corresponding to (2.1), we may solve it directly using the same logic as was sketched in the first paragraph of the Introduction (and which is discussed more fully in Appendix 2). Let w^* be an optimal solution to

(2.5) $\max \quad cw$

$$\text{subject to } \quad b^- \leq Aw \leq b^+, \quad w \in \mathbb{R}^m;$$

then the vectors

$$(2.6) \qquad x_j^* = \begin{cases} w_j^*, & \text{if} \quad w_j^* \geq 0, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$(2.7) \qquad z_j^* = \begin{cases} -w_j^*, & \text{if} \quad w_j^* < 0, \\ 0, & \text{otherwise,} \end{cases}$$

are optimal for (2.4). The optimal slacks s_1^* and s_2^* are then uniquely determined.

With this information we are now able to construct an optimal basis matrix for (2.4). Let V be the $m \times m$ diagonal matrix defined by

$$(2.8) \qquad V_{ii} = \begin{cases} 1, & \text{if } x_j^* \geq 0, \\ -1, & \text{otherwise,} \end{cases}$$

and let β be the $m \times m$ diagonal matrix with

$$(2.9) \qquad \beta_{jj} = \begin{cases} 0, & \text{if } cA_j^{-1} \geq 0, \\ 1, & \text{otherwise,} \end{cases}$$

where A_j^{-1} is the jth column of A^{-1}. Then the $2m \times 2m$ matrix

$$(2.10) \qquad B = \begin{bmatrix} AV & \beta \\ 0 & I \end{bmatrix}$$

is an optimal basis matrix for (2.4).

Having solved (2.4) and determined the optimal basis (2.10), we now compute an optimal simplex tableau in extended Tucker form (see Appendix 1). In general, we know that if B is an optimal basis matrix for the linear program

$$\text{(2.11)} \qquad \max \; cx \quad \text{subject to} \quad Ax = b, \quad x \geq 0,$$

and A can be partitioned as

$$\text{(2.12)} \qquad A = [B \,|\, N],$$

then (except for permutations of rows or of columns) the optimal extended Tucker tableau is

$$\text{(2.13)} \qquad \begin{array}{c|c|c|}
 & 1 & -x_{N1} \;\; \cdots \;\; -x_{N(n-m)} \\
\hline
z & c_B x_B^* & -d \\
\hline
\begin{matrix} x_{B1} \\ \vdots \\ x_{Bm} \end{matrix} & x^* & B^{-1}N \\
\hline
\begin{matrix} x_{N1} \\ \vdots \\ x_{N(n-m)} \end{matrix} & 0 & -I \\
\hline
\end{array}$$

where

$$\text{(2.14)} \qquad x^* = B^{-1}b,$$

and

$$d = c_N - c_B(B^{-1}N), \tag{2.15}$$

the vector of reduced cost coefficients. c_B and c_N have the usual interpretation as the cost vector c partitioned as in (2.12). To compute this tableau for (2.4) we first partition the matrix

$$\begin{bmatrix} A & -A & I & 0 \\ 0 & 0 & I & I \end{bmatrix} \tag{2.16}$$

as

$$[B|N] = \left[\begin{array}{cc|cc} AV & \beta & -AV & I-\beta \\ 0 & I & 0 & I \end{array}\right] \tag{2.17}$$

The optimal basis given in (2.10) is placed in the first $2m$ columns, and the nonbasic columns (arranged in the particular configuration above) are put in the remaining $2m$ columns. Observe that AV is nonsingular (since both A and V are) and that hence B is nonsingular. If a square matrix is of the form

$$W = \begin{bmatrix} U & Z \\ 0 & I \end{bmatrix} \tag{2.18}$$

with U nonsingular, then W^{-1} exists and is

$$W^{-1} = \begin{bmatrix} U^{-1} & -U^{-1}Z \\ 0 & I \end{bmatrix}. \tag{2.19}$$

Therefore,

$$B^{-1} = \begin{bmatrix} VA^{-1} & -VA^{-1}\beta \\ 0 & I \end{bmatrix}. \tag{2.20}$$

With B^{-1} we may now compute all the elements of the extended Tucker tableau.

First,

$$B^{-1}N = \begin{bmatrix} VA^{-1} & -VA^{-1}\beta \\ 0 & I \end{bmatrix} \begin{bmatrix} -AV & I-\beta \\ 0 & I \end{bmatrix} \tag{2.21}$$

$$= \begin{bmatrix} -I & VA^{-1}\Sigma \\ 0 & I \end{bmatrix}, \tag{2.22}$$

where

$$\Sigma = (I-\beta) - \beta \tag{2.23}$$

$$= I - 2\beta. \tag{2.24}$$

Notice that Σ is simply a diagonal matrix with

$$\Sigma_{jj} = \begin{cases} 1, & \text{if } cA_j^{-1} \geq 0, \\ -1, & \text{otherwise.} \end{cases} \tag{2.25}$$

Next we compute the vector of reduced cost coefficients. It is clear from (2.17) that the vector of costs associated with the basic variables is

$$c_B = (cV \mid 0), \tag{2.26}$$

and that the vector of costs associated with the nonbasic variables is

$$c_N = (-cV \mid 0); \tag{2.27}$$

hence by (2.15) we have

$$d = c_N - c_B(B^{-1}N) \tag{2.28}$$

$$= (-cV \mid 0) - (cV \mid 0) \begin{bmatrix} -I & VA^{-1}\Sigma \\ 0 & I \end{bmatrix} \tag{2.29}$$

$$= (-cV \mid 0) - (-cV \mid cVVA^{-1}\Sigma) \tag{2.30}$$

$$= (-cV \mid 0) - (-cV \mid cA^{-1}\Sigma) \tag{2.31}$$

$$= (0 \mid -cA^{-1}\Sigma). \tag{2.32}$$

It should be noted in passing that from the definition of Σ the vector d given by (2.32) has nonpositive elements, and thus the completed tableau will be dual feasible, as we should expect.

We may now construct the entire optimal extended Tucker tableau for the continuous linear program (2.4); Figure 1 is a diagram of such a tableau. It is important to emphasize that the tableau can be constructed without actually solving problem (2.4) by the simplex method; the entire structure can be determined directly from the matrix A^{-1}, the signs of the elements of the vector cA^{-1}, and the vectors b^- and b^+.

We now set out to derive a condensed cutting plane method for

	1	$-x_{N1} \;\cdots\; -x_{Nm}$	$s_{N1} \;\cdots\; -s_{Nm}$
z	$c_B x_B^*$	0	$cA^{-1}\Sigma$
$x_{B1}, \ldots, x_{Bm}$	x_B^*	$-I$	$VA^{-1}\Sigma$
$s_{B1}, \ldots, s_{Bm}$	$b^+ - b^-$	0	I
$x_{N1}, \ldots, x_{Nm}$	0	$-I$	0
$s_{N1}, \ldots, s_{Nm}$	0	0	$-I$

Figure 1. Diagram of optimal extended Tucker tableau for problem (2.4).

evolving from an optimal continuous linear programming tableau to an optimal discrete solution. Recall from Appendix 1 that the rules for Gomory's method can be summarized as follows:

(i) If all x_{Bi} and s_{Bi} are integers the problem is solved; otherwise continue.

(ii) Let σ be the least index for which t_{i0} is noninteger. For each element $t_{\sigma j}$ of row σ let

$$f_{\sigma j} = t_{\sigma j} - [t_{\sigma j}]^I ,$$

the positive fractional part of element $t_{\sigma j}$. Define a new slack variable γ and append to the tableau the constraint

$$\gamma = -f_{\sigma 0} + \sum_{j=0}^{n-m} (-x_j) f_{\sigma j} \tag{2.33}$$

$$\geq 0. \tag{2.34}$$

(iii) Perform a dual simplex pivot operation with the appended row as the pivot row.

(iv) If some element of the 0th column becomes negative after step (iii), perform a dual simplex pivot operation with that negative element's row as the pivot row. Continue until all t_{i0} are nonnegative.

(v) If all t_{i0} are integers the integer programming problem is solved; otherwise return to step (ii).

Notice that each Gomory cut introduces a new variable (the slack γ) and increases the dimension of the problem by unity; this means that each new cut may allow one of the nonbasic variables to take on a nonzero value. Since no companion pair of structural variables x_{Bi} and x_{Ni} will ever take on nonzero values simultaneously, any nonbasic variables taking on positive values must be nonbasic slacks.

In the event that one of the basic structural variables becomes negative after the introduction of a cut, primal feasibility can be restored by simply replacing the negative variable x_{Bi} with its nonbasic counterpart of reversed sign, x_{Ni}; since this has no effect whatsoever on the value of the objective function it is an allowable substitution. The following proposition, stated without proof, shows that such a basis change can be performed without an explicit dual pivoting operation.

Proposition 1: Let $A = [B|N]$ be partitioned such that B is a nonsingular matrix. Suppose that column N_i is the negative of column B_j, and let B' and N' be the corresponding basic matrices with B_j and N_i interchanged. Then

(2.35) $$(B')^{-1} = E(B^{-1})$$

and

(2.36) $$(B')^{-1}N' = E(B^{-1}N)F,$$

where E and F are diagonal matrices of the appropriate dimensions with 1's along the diagonal except for -1's in the jth and ith positions respectively.

The significance of this result is that we now know precisely what the tableau would look like if we replaced the basic variable x_{Bi} with x_{Ni} through an explicit dual pivot; the row labelled x_{Bi} would have an entry of -1 in the column previously labelled x_{Ni} and zero's elsewhere, and the row labelled x_{Ni} would consist of a negative copy of the pivot row (<u>before</u> the pivot) except for a -1 in the column previously labelled x_{Ni}. To illustrate, consider the simple tableau

(2.37)

	1	$-x_{N1}$	$-x_{N2}$	$-s_{N1}$	$-s_{N2}$
z	10	0	0	5	6
x_{B1}	-2	-1	0	1	2
x_{B2}	3	0	-1	3	4
s_{B1}	10	0	0	1	0
s_{B2}	10	0	0	0	1
x_{N1}	0	-1	0	0	0
x_{N2}	0	0	-1	0	0
s_{N1}	0	0	0	-1	0
s_{N2}	0	0	0	0	-1

Pivoting on the circled element to eliminate the infeasibility in the first row yields the new tableau

(2.38)

	1	$-x_{B1}$	$-x_{N2}$	$-s_{N1}$	$-s_{N2}$
z	10	0	0	5	6
x_{B1}	0	-1	0	0	0
x_{B2}	3	0	-1	3	4
s_{B1}	10	0	0	1	0
s_{B2}	10	0	0	0	1
x_{N1}	2	-1	0	-1	-2
x_{N2}	0	0	-1	0	0
s_{N1}	0	0	0	-1	0
s_{N2}	0	0	0	0	-1

which has the structure anticipated above. Thus, when a basic structural variable x_{Bi} becomes negative we can restore primal feasibility by reversing the sign of every element in the row labelled x_{Bi} except for the -1 in the column labelled x_{Ni} and interchanging the labels x_{Ni} and x_{Bi} wherever they appear. For example, performing this operation on tableau (2.37) we have

(2.39)

	1	$-x_{B1}$	$-x_{N1}$	$-s_{N1}$	$-s_{N2}$
z	10	0	0	5	6
x_{N1}	2	-1	0	-1	-2
x_{B2}	3	0	-1	3	4
s_{B1}	10	0	0	1	0
s_{B2}	10	0	0	0	1
x_{B1}	0	-1	0	0	0
x_{N2}	0	0	-1	0	0
s_{N1}	0	0	0	-1	0
s_{N2}	0	0	0	0	-1

Next we give a proposition which allows us to drop from our computations all the columns corresponding to the nonbasic structural variables x_{Ni}, $i \in \{1, \ldots, m\}$. Intuitively, this is to be expected; the columns of the tableau simply contain the coefficients for expressing the nonbasic variables in terms of the basic variables. Since

(2.40) $$x_{Ni} = -x_{Bi} \quad \forall\, i \in \{1, \ldots, m\}$$

there is effectively no information contained in the columns corresponding to the x_{Ni}. We must verify, though, that the subsequent Gomory cuts do not destroy relationship (2.40).

Proposition 2: Let any sequence of Gomory cuts be performed on the optimal Tucker tableau for problem (2.4). For any $i \in \{1, \ldots, m\}$ the column corresponding to the nonbasic variable x_{Ni} consists of -1's in the rows labelled x_{Bi} and x_{Ni} and zero's elsewhere.

Proof: Let $\bar{B}$ be the basis matrix for the expanded linear program, i.e., the original linear program with the appended Gomory cuts. We know from the nature of Gomory's method that any appended constraint expresses restrictions only on the current non-basic variables. Since exactly one of the variables x_{Bi} and x_{Ni} is in the basis at any iteration, we know that a Gomory cut never explicitly constrains an original structural variable. Therefore the relationship

$$x_{Ni} = -x_{Bi} \qquad i \in \{1, \ldots, m\} \tag{2.41}$$

remains true throughout the computations, making the entry in the column labelled x_{Ni} -1 for the row labelled x_{Ni} and zero for all other rows corresponding to basic variables. The remaining rows in the tableau are merely bookkeeping rows, stating that for any non-basic variable y the relation

$$y = (-y)(-1) \tag{2.42}$$

holds. This means that the column labelled x_{Ni} has a -1 in the row labelled x_{Ni} and 0's in all the other rows corresponding to nonbasic variables. Therefore column x_{Bi} consists of zero's and two strategically placed -1's. This completes the proof.

We therefore see that, by Proposition 1, the columns corresponding to the nonbasic variables are not necessary for restoring primal feasibility in the event that a basic structural variable becomes negative, and by Proposition 2 that these columns are not needed for the generation of Gomory cuts. They will never contain any elements which are not integers and thus will never contribute any positive fractional parts to an appended constraint. If we adopt the implicit pivoting method discussed earlier when a basic structural variable becomes negative, then we may eliminate the columns corresponding to the nonbasic variables. This renders the rows corresponding to these variables useless, so they may be dropped as well. We are left with the abbreviated tableau shown in Figure 2. With this tableau, Gomory's algorithm may then be carried out using the same steps as outlined earlier in the chapter, except that in step (iv) primal infeasibilities in the structural variables are removed by the implicit method derived from Proposition 1.

We now show that the rows corresponding to the nonbasic slacks can be ignored and in fact dropped from the computations. It seems reasonable that this should be possible; for any $j \in \{1, \ldots, m\}$ the

	1	$-s_{N1} \quad \cdots \quad -s_{Nm}$
z	$c_B x_B^*$	$cA^{-1}\Sigma$
x_{B1} . . . x_{Bm}	x_B^*	$VA^{-1}\Sigma$
s_{B1} . . . s_{Bm}	$b^+ - b^-$	I
s_{N1} . . . s_{Nm}	0	$-I$

Figure 2. Abbreviated optimal extended Tucker tableau for problem (2.4).

identity

$$s_{Bj} + s_{Nj} = b_j^+ - b_j^- \tag{2.43}$$

holds, so that the vectors s_B and $b^+ - b^-$ contain all the information about the slack variables.

No row corresponding to a slack variable will ever serve as a source row for generating a Gomory cut since the source row is always the first row for which the tableau entry $t_{\sigma 0}$ is noninteger, and once the structural variables take on integer values the slacks must be integer as well. Thus the only purposes for these rows are (a) to signal that a slack variable has become negative and then (b) to serve as the pivot row to be used in removing the infeasibility. By (2.43) we see that if any slack s_{Bi} (or s_{Ni}) becomes negative then its companion slack s_{Ni} (or s_{Bi}) becomes greater than $b_j^+ - b_j^-$. We will capitalize on this observation in deriving a modified pivoting scheme that will allow us to delete the rows corresponding to the nonbasic slacks.

Suppose that a sequence of Gomory cuts has been performed on the starting tableau shown in Figure 2. Then the resulting tableau is of the form

(2.44)

$$
\begin{array}{c|c|c|}
 & 1 & \lambda \\
\hline
z & M & -d \\
\hline
x_B & x & T \\
\hline
s_B & u & W \\
\hline
s_N & b^{+}-b^{-}-u & -W \\
\hline
\end{array}
$$

where the vector λ of labels contains as many as m of the slacks introduced with the addition of each appended constraint. If any element x is negative, the signs in row i are adjusted in the manner previously discussed and the label x_{Bi} is replaced by x_{Ni}. If any element u_i is negative a conventional dual pivot operation is performed to restore primal feasibility. If for some $i \in \{1, \ldots, m\}$ we have

$$(b_i^{+} - b_i^{-}) - u_i < 0$$

then a dual pivot operation is performed on the appropriate row. This pivot operation would be carried out as follows:

(i) Let r be the index of the row labelled s_{Ni} and let s be such that

(2.45) $$\left|\frac{w_{0s}}{w_{rs}}\right| = \min_{j} \left|\frac{w_{0j}}{w_{rj}}\right| \quad \text{over all } j \ni w_{rj} < 0.$$

(ii) Divide every element of column s by w_{rs} and use this rescaled column to eliminate all other nonzero entries in row r.

Computationally, we see that the same pivot operation could be carried out by operating on the corresponding basic slack row labelled s_{Bi}:

(i) Let r be the index of the row labelled s_{Bi} and let s be such that

$$\left|\frac{w_{0s}}{w_{rs}}\right| = \min_j \left|\frac{w_{0j}}{w_{rj}}\right| \quad \text{over all } j \ni w_{rj} > 0. \tag{2.46}$$

(ii) Divide every element of column s by w_{rs} and use this column to (a) reduce element u_i to $b_i^+ - b_i^-$ and (b) eliminate all other nonzero entries in row r.

Since this pivoting operation can be performed without the benefit of an explicit representation of the portion of the tableau corresponding to the nonbasic slack variables we may delete that part of the tableau altogether and carry only the subtableau

(2.47)

$$\begin{array}{c|c|c|c}
 & 1 & \lambda & \\
\hline
z & M & -d & \\
\hline
x_B & x & T & \\
\hline
s_B & u & W & \boxed{b^+ - b^-}
\end{array}.$$

Motivated by this observation, we see that the starting tableau

(2.48)

$$\begin{array}{c|c|c|c}
 & 1 & -s_N & \\
\hline
z & cx^* & cA^{-1}\Sigma & \\
\hline
x_B & x^*V & VA^{-1}\Sigma & \\
\hline
s_B & b^+-b^- & I & \boxed{b^+-b^-} \\
\hline
\end{array}$$

is sufficient for generating the entire sequence of Gomory cuts necessary to solve the integer program (2.1). The vector (b^+-b^-) appended at the right of the original tableau is actually a vector of upper bounds on the basic slack variables.

The following summary of the condensed method we have derived is seen to contain several similarities with the conventional upper bounded linear programming algorithm. We consider problem (2.1), where A is nonsingular and all entries of A, b^-, b^+ and c are integers.

<u>Condensed Cutting Plane Algorithm (Nonsingular Case).</u>

(1) Solve the linear programming problem

$$\max \quad cx$$

$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x \in \mathbb{R}^m,$$

by setting

$$b_i^* = \begin{cases} b_i^-, & \text{if } (cA^{-1})_i < 0, \\ b_i^+, & \text{otherwise,} \end{cases}$$

and

$$x^* = A^{-1}b^*.$$

(2) Define the $m \times m$ diagonal matrices V and Σ by

$$\Sigma_{ii} = \begin{cases} +1, & \text{if } (cA^{-1})_i \geq 0, \\ -1, & \text{otherwise,} \end{cases}$$

and

$$V_{ii} = \begin{cases} 1, & \text{if } x_i^* \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

(3) Construct the tableau shown in Figure 3; we will denote the main body of the tableau as the matrix $T = [t_{ij}]$ where $i \in \{0, 1, \ldots, 2m\}$ and $j \in \{0, 1, \ldots, m\}$. We have used the integers, with their signs altered by the diagonal elements of V and Σ, as labels, in effect using the labels to record the true signs of our structural variables and to identify the basic slack variables.

(4) If $t_{i0} \in \mathbb{Z}^+$ $\forall\, i \in \{0, 1, \ldots, m\}$ the integer programming problem is solved. Otherwise, let r be the index of the first row for which t_{i0} is noninteger. Set

$$f_{rj} = t_{rj} - [t_{rj}]^I \quad \forall\, j \in \{0, 1, \ldots, m\}$$

and append the constraint

$$
\begin{array}{c|c|c|}
 & 1 & -(m+1)\Sigma_{11} \;\; \cdots \;\; -(2m)\Sigma_{mm} \\
\hline
z & cx^{*} & cA^{-1}\Sigma \\
\hline
\begin{array}{c} V_{11} \\ 2V_{22} \\ \vdots \\ mV_{mm} \end{array} & x^{*}V & VA^{-1}\Sigma \\
\hline
\begin{array}{c} (m+1)\Sigma_{11} \\ (m+2)\Sigma_{22} \\ \vdots \\ (2m)\Sigma_{mm} \end{array} & b^{+}-b^{-} & I \\
\hline
\end{array}
\qquad
\begin{array}{|c|}
\hline
b^{+}-b^{-} \\
\hline
\end{array}
$$

Figure 3. Abbreviated Tucker tableau for condensed cutting plane algorithm.

$$\gamma_i = -f_{r0} - \sum_{j=1}^{m} (-x_j) f_{rj} \geq 0$$

to the tableau as row $2m+1$. Perform a dual pivot operation with row $2m+1$ as the pivot row; after the pivot delete the new row (but leave its label on the appropriate column).

(5) Set

$$\tau_1 = \min\{t_{i0} \mid i = 1, \ldots, m\},$$

$$\tau_2 = \min\{t_{i0} \mid i = m+1, \ldots, 2m\},$$

$$\tau_3 = \min\{(b_i^+ - b_i^-) - t_{m+i,\,0} \mid i = 1, \ldots, m\},$$

$$\tau = \min\{\tau_1, \tau_2, \tau_3\}.$$

If $\tau \geq 0$ go to step (4); otherwise continue.

(6) If $\tau = \tau_1$ and r is the least index such that $t_{i0} = \tau$, then reverse the sign of each element in row r and reverse the sign of row r's label. If $\tau = \tau_2$ and $r \in \{m+1, \ldots, 2m\}$ is the least index such that $t_{i0} = \tau$ then perform a conventional dual pivot operation on row r. If $\tau = \tau_3$ and $r \in \{m+1, \ldots, 2m\}$ is the least index for which $t_{i0} = \tau$ then perform the following operation:

(i) Let $k = r - m$ and let

$$\left|\frac{a_{0s}}{a_{rs}}\right| = \min_j \left|\frac{a_{0j}}{-a_{rj}}\right| \quad \text{over all } j \ni a_{rj} > 0.$$

(ii) Divide every element of column s by t_{rs}, and use this rescaled column to reduce element t_{r0} to $b_k^+ - b_k^-$ and to eliminate every other nonzero entry from row r. Do not change any labels.

In case of a tie in taking $\min\{\tau_1, \tau_2, \tau_3\}$ we use the usual lexicographic ordering rule with the obvious modification.

Go to step (5).

As was mentioned previously, the algorithm is known to converge since it is logically equivalent to Gomory's original method.

Now consider the problem

$$\text{(2.49)} \qquad \begin{aligned} &\max \ cx \\ &\text{subject to} \ \ b^- \leq Ax \leq b^+, \quad x \in \mathbb{Z}^m, \end{aligned}$$

where A is an $m \times n$ matrix of full row rank and c is in the row space of A. As before, we assume that all elements of A, b^-, b^+ and c are integers.

It is known (see Appendix 2) that if A^- is a generalized inverse of A (i.e., that $A(A^-)A = A$), then a solution to the continuous linear program

$$\text{(2.50)} \qquad \max \quad cx$$
$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x \in \mathbb{R}^m$$

may be found by setting

$$\text{(2.51)} \qquad b_i^* = \begin{cases} b_i^+, & \text{if } (cA^-)_i \geq 0, \\ b_i^-, & \text{otherwise,} \end{cases}$$

and letting

$$\text{(2.52)} \qquad x^* = A^- b^*.$$

Since A is of full row rank, we know that a nonsingular $m \times m$ submatrix B exists. We partition A as

$$\text{(2.53)} \qquad A = [B|N].$$

A simple method of computing a generalized inverse of A is given by the following lemma, offered without proof.

Lemma: If A is partitioned as (2.53) then the $n \times m$ matrix

$$\text{(2.54)} \qquad H = \left[\frac{B^{-1}}{0}\right]$$

is a generalized inverse of A, i.e., $AHA = A$.

This lemma may be used to obtain a useful result:

Proposition 3: Consider the linear program (2.50) and assume that $A = [B|N]$ with B nonsingular. Let b^* be determined by

$$(2.55) \qquad b_i^* = \begin{cases} b_i^+, & \text{if } (cB^{-1})_i \geq 0, \\ b_i^-, & \text{otherwise.} \end{cases}$$

Then the vector

$$(2.56) \qquad x^* = (B^{-1}b^*|0)$$

is an optimal solution to (2.50).

Proof: We know from the previous lemma that the matrix $H = \left[\frac{B^{-1}}{0}\right]$ is a generalized inverse of A, so that setting $A^- = H$ and substituting into (2.51) and (2.52) give the desired results.

Suppose now that $A = [B|N]$ and that the continuous problem (2.50) has been solved as in Proposition 3. Define the $m \times m$ diagonal matrices V and Σ as before, i.e.,

$$(2.57) \qquad \Sigma_{ii} = \begin{cases} 1, & \text{if } (cA^{-1})_i \geq 0, \\ -1, & \text{otherwise,} \end{cases}$$

and

$$(2.58) \qquad V_{ii} = \begin{cases} 1, & \text{if } x_i^* \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

If we were to formulate (2.50) as a conventional linear program as

(2.59) $\max \quad cx - cz$

subject to

$$\begin{bmatrix} A & -A & I & 0 \\ 0 & 0 & I & I \end{bmatrix} \begin{bmatrix} x \\ z \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} b^+ \\ b^+ - b^- \end{bmatrix},$$

$$x,\ z,\ s_1,\ s_2 \geq 0,$$

we would have

(2.60)
$$\begin{bmatrix} BV & \beta \\ 0 & I \end{bmatrix}$$

as an optimal basis, where $\beta = (I-\Sigma)/2$, and thus

(2.61)
$$\begin{bmatrix} BV & \beta \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} VB^{-1} & -VB^{-1}\beta \\ 0 & I \end{bmatrix}.$$

The optimal extended Tucker tableau for (2.50) can then be computed, and it may be written as is shown in Figure 4. Adopting the simplified pivoting rules for removing primal infeasibilities in the basic structural variables as they occur and removing infeasibilities in the slack variables (both basic and nonbasic), we see that the tableau may be reduced to that shown in Figure 5. As before, the vector $(b^+ - b^-)$ appended at the right is not affected by any pivot operations; it is merely a list of the upper bounds on the basic slacks.

Unfortunately the size of the tableau cannot be reduced further,

	1	$-(x_B)$	$-(s_N)$	$-(x_N)$	$-(-x_N)$
z	$c_B x^*$	0	$cA^{-1}\Sigma$	0	0
x_B	Vx^*	$-I$	$VA^{-1}\Sigma$	$VB^{-1}N$	$-VB^{-1}N$
s_B	$b^+ - b^-$	0	I		
x_N	0			$-I$	
$-x_N$	0				$-I$
s_N	0		$-I$		
$-x_B$	0	$-I$			

Figure 4. Optimal extended Tucker tableau for problem (2.50).

	1	$-(s_N)$	$-(x_N)$	$-(-x_N)$	
z	$c_B x^*$	$cB^{-1}\Sigma$	0	0	
x_B	Vx^*	$VB^{-1}\Sigma$	$VB^{-1}N$	$-VB^{-1}N$	
x_N	0		$-I$		
$-x_N$	0			$-I$	
s_B	0	I			$b^+ - b^-$

Figure 5. Abbreviated optimal Tucker tableau for problem (2.50).

but we can simplify the pivoting procedure for removing primal infeasibilities in the nonbasic structural variables if they should occur. The basic idea is the same as that employed in devising the modified pivoting rules for the structural variables in the nonsingular case: by replacing a variable x_k with its companion variable $-x_k$, we may perform the pivot implicitly with a set of sign and label changes.

It is clear that at most one of the variables x_{Ni} and $-x_{Ni}$ will take on positive value at any given stage of the computations. Suppose that variable x_{Ni} has a negative value after a Gomory cut is introduced; variable $-x_{Ni}$ would be chosen as the variable to replace it (since $-x_{Ni}$'s reduced cost coefficient must be zero). The row labelled $-x_{Ni}$ has a -1 entry in the column labelled $-(-x_{Ni}) = x_{Ni}$ and zero's elsewhere; if a pivot is performed to replace x_{Ni} with $-x_{Ni}$ then the row labelled x_{Ni} will have a single entry of -1 in the appropriate column, and the row labelled $-x_{Ni}$ will consist of the original -1 entry with a negative copy of the row labelled x_{Ni} as it appeared before the pivot in the remaining entries.

The same pivot can be performed by changing the labels on the rows and columns corresponding to the two variables in question and performing these sign changes in the row of the variable that is being driven up to zero, in the same fashion as we perform analogous pivots on the basic structural variables.

We now summarize the condensed cutting plane method for the case in which A is singular.

<u>Condensed Cutting Plane Algorithm (Singular Case)</u>.

(1) Assume that A is partitioned as $[B|N]$ with B nonsingular. Solve the linear program

$$\begin{aligned} &\max \quad cx \\ &\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x \in \mathbb{R}^n, \end{aligned}$$

by setting

$$b_i^* = \begin{cases} b_i^-, & \text{if } (cB^{-1})_i < 0, \\ b_i^+, & \text{otherwise,} \end{cases}$$

and

$$x^* = (B^{-1}b^* | 0).$$

(2) Define the diagonal matrices V and Σ by

$$\Sigma_{ii} = \begin{cases} 1, & \text{if } (cB^{-1})_i \geq 0, \\ -1, & \text{otherwise,} \end{cases}$$

and

$$V_{ii} = \begin{cases} 1, & \text{if } x_i^* \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

(3) Construct the tableau shown in Figure 6 and denote the main body of the tableau by $T = [t_{ij}]$, where $i \in \{0, 1, \ldots, 2n\}$

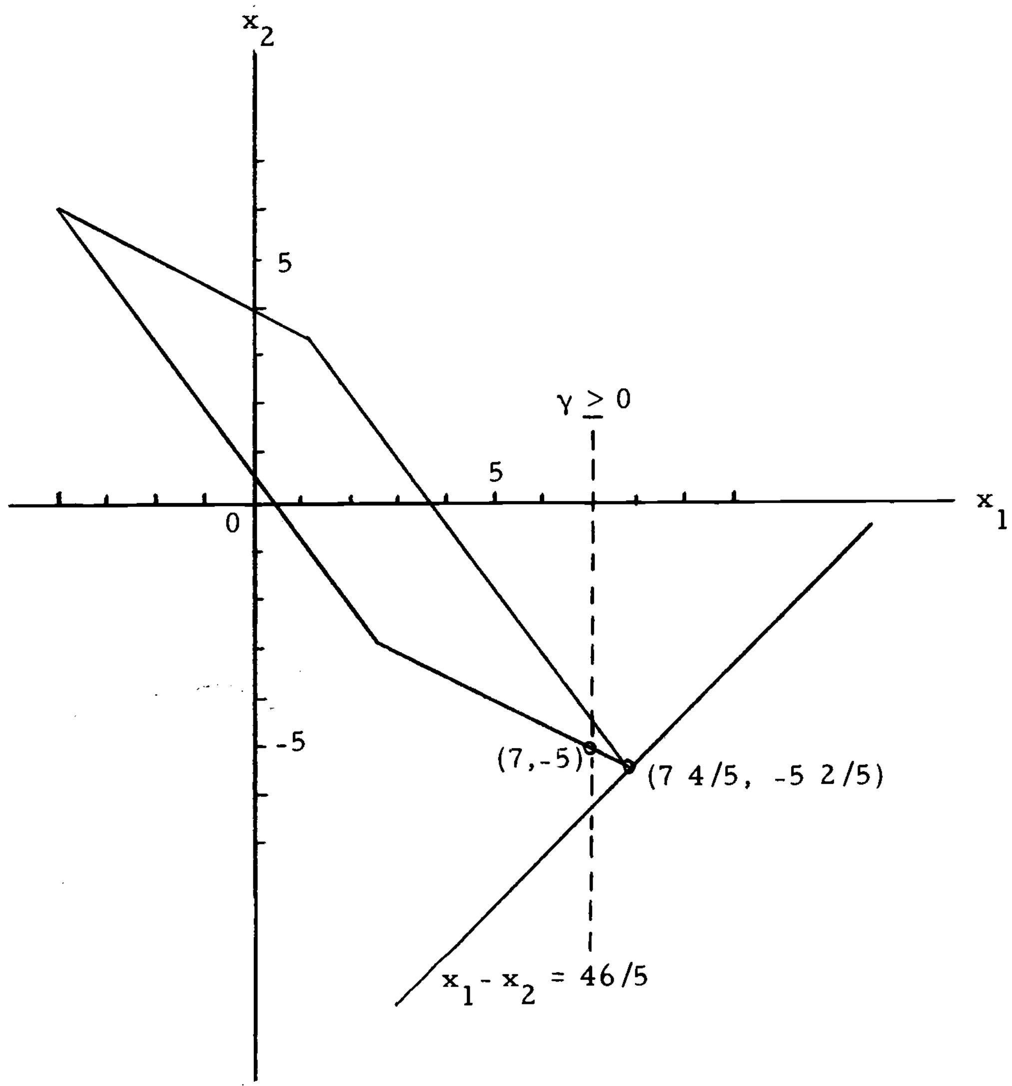

Figure 6. Geometrical representation of problem (2.62).

and $j \in \{0, 1, \ldots, 2n-m\}$. We do not use the labelling scheme employed in the nonsingular case but instead use literal labels to identify the variables.

(4) If t_{i0} is integer $\forall\, i \in \{1, \ldots, 2n-m\}$ the integer programming problem is solved. Otherwise, let r be the least index for which t_{i0} is noninteger; set

$$f_{rj} = t_{rj} - [t_{rj}]^I \quad \forall j \in \{0, 1, \ldots, 2n-m\}$$

and append the constraint

$$\gamma = -f_{r0} - \sum_{j=0}^{2n-m} (-x_j) f_{rj} \geq 0$$

to the tableau as row $2n+1$. Perform a dual pivot operation with the new row as the pivot row; after the pivot delete the new row (but leave the new slack label on the appropriate column).

(5) Set

$$\tau_1 = \min\{t_{i0} \mid i = 1, \ldots, m\}$$

$$\tau_2 = \min\{t_{i0} \mid i = m+1, \ldots, 2n-m\}$$

$$\tau_3 = \min\{t_{i0} \mid i = 2n-m+1, \ldots, 2n\}$$

$$\tau_4 = \min\{(b_i^+ - b_i^-) - t_{i0} \mid i = 2n-m+1, \ldots, 2n\}$$

$$\tau = \min\{\tau_1, \tau_2, \tau_3, \tau_4\}$$

If $\tau \geq 0$ go to step (4); otherwise continue.

(6) If $\tau = \tau_1$, τ_3 or τ_4 then the infeasibility may be removed as in the corresponding case when A is nonsingular. If $\tau = \tau_2$ we let $r \in \{m+1, \ldots, 2n-m\}$ be the least index for which $t_{i0} < 0$. Reverse the sign for every element of row r except for the entry -1 in the companion variable's column, and interchange the labels of the pivot row and its companion row. Replace the label on the companion variable's column with the label of the variable being driven up to zero. In case of any tie in choosing the variable to be replaced, use the usual lexicographic ordering rule with the obvious modification. Go to step (5).

Again, we need not give a convergence proof since we have merely a special case of Gomory's original algorithm.

To illustrate the workings of this scheme, we consider the problem

(2.62) $\max \quad x_1 - x_2$

subject to

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix} \leq \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 8 \\ 15 \end{bmatrix},$$

$$x_1, x_2 \in \mathbb{Z}.$$

Then

$$A^{-1} = \begin{bmatrix} -3/5 & 2/5 \\ 4/5 & -1/5 \end{bmatrix}$$

and

$$cA^{-1} = (1 \quad -1) \begin{bmatrix} -3/5 & 2/5 \\ 4/5 & -1/5 \end{bmatrix}$$

$$= (-7/5 \quad 3/5),$$

hence

$$b^* = \begin{bmatrix} -3 \\ 15 \end{bmatrix},$$

$$x^* = A^{-1}b^*$$

$$= \begin{bmatrix} 7\ 4/5 \\ -5\ 2/5 \end{bmatrix} .$$

We define the matrices

$$V = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$$

and

$$\Sigma = \begin{bmatrix} -1 & 0 \\ 0 & +1 \end{bmatrix}$$

and compute

$$VA^{-1}\Sigma = \begin{bmatrix} 3/5 & 2/5 \\ 4/5 & 1/5 \end{bmatrix},$$

$$cA^{-1}\Sigma = (-7/5 \quad 3/5) \begin{bmatrix} -1 & 0 \\ 0 & +1 \end{bmatrix},$$

$$= (7/5 \quad 3/5),$$

and

$$x^{*}V = (7\ 4/5 \quad 5\ 2/5).$$

Then the optimal condensed tableau is

	1	+3	-4	
z	13 1/5	7/5	3/5	
+1	7 4/5	3/5	2/5	
-2	5 2/5	4/5	1/5	
-3	11	1	0	11
+4	13	0	1	13

The row labelled z is the first row with a noninteger entry in the 0th row; for convenience, however, we will take the row labelled +1 as the source row. The tableau with the Gomory cut appended at the bottom is

	1	+3	-4	
z	13 1/5	7/5	3/5	
+1	7 4/5	3/5	2/5	
-2	5 2/5	4/5	1/5	
-3	11	1	0	11
+4	13	0	1	13
γ	-4/5	-3/5	-2/5	

Since $\left|\dfrac{3/5}{-2/5}\right| < \left|\dfrac{7/5}{-3/5}\right|$, we pivot on the circled element and obtain the updated tableau

	1	+3	γ	
z	12	1	3/2	
+1	7	0	1	
-2	5	1/2	1/2	
-3	11	1	0	11
4	11	-3/2	5/2	13

Notice that the Gomory cut row is dropped immediately after the pivot. Since all the elements of the 0th column are integers, the problem is solved. Assigning the algebraic signs of the labels to the first and second elements of the 0th column yields the optimal solution

$$x_1^* = 7$$

$$x_2^* = -5.$$

Figure 6 shows how the Gomory cut severs the linear programming solution from the feasible region and reveals the point (7 -5) as a vertex of the reduced feasible region.

III. AN IMPLICIT ENUMERATION SCHEME

Perhaps the most naive general approach to solving an optimization problem is to list all the possible solutions and then choose the best one. When the number of possible solutions is large (or infinite), the computational disadvantage of such a scheme is obvious. However, if the possible solutions can be investigated in an orderly sequence such that an optimal solution may be identified without necessarily evaluating all the remaining possible solutions, then considerable efficiency may be achieved. Optimization schemes that operate on this general principle are suggestively called implicit enumeration algorithms; both dynamic programming and branch-and-bound are examples of such schemes. The purpose of this chapter is to derive an implicit enumeration algorithm for the interval integer programming problem.

Again consider the IIP

$$(3.1) \qquad \max \; cx$$
$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x \in \mathbb{Z}^m,$$

where all elements of A, b^-, b^+ and c are integers and A is $m \times m$ and nonsingular. We will derive an implicit enumeration method for solving this problem, and we will also show that the algorithm can be used to solve problems of the form

$$(3.2)\qquad \max\ cx \quad \text{subject to}\quad b^- \leq Ax \leq b^+,\quad x \in \mathbb{Z}^n$$

where $A \in \mathbb{Z}^{m,n}$ is of full row rank and c is in the row space of A.

First, consider the continuous analog to problem (3.1),

$$(3.3)\qquad \max\ cx \quad \text{subject to}\quad b^- \leq Ax \leq b^+,\quad x \in \mathbb{R}^m;$$

we have seen previously that this problem is equivalent to

$$(3.4)\qquad \max\ \gamma z \quad \text{subject to}\quad b^- \leq z \leq b^+,\quad z \in \mathbb{R}^m,$$

where $\gamma = cA^{-1}$ and $z = Ax$. Since $x = A^{-1}z$ we may then express the original integer program (3.1) as

$$(3.5)\qquad \max\ \gamma z \quad \text{subject to}\quad b^- \leq z \leq b^+,\quad z \in \mathbb{R}^m,\ A^{-1}z \in \mathbb{Z}^m.$$

We will concentrate our efforts toward solving problem (3.5); if z^* is a solution to (3.5) then $x^* = A^{-1}z^*$ is a solution to (3.1). We will first find a solution $\bar{z}$ to the continuous linear program (3.4) and then enumerate, in an orderly fashion, a sequence of alternate (and possibly suboptimal) solutions until a z^* solving (3.5) is revealed.

As we have seen, the general solution to (3.4) is

$$\hat{z}_i = \begin{cases} b_i^+, & \text{if } \gamma_i > 0, \\ b_i^-, & \text{if } \gamma_i < 0, \\ \theta b_i^+ + (1-\theta)b_i^-, & \text{if } \gamma_i = 0, \end{cases} \tag{3.6}$$

where θ is an arbitrary number in $[0,1]$. For ease of discussion and to remove ambiguity, set $\theta = 1$ and take

$$\bar{z}_i = \begin{cases} b_i^+, & \text{if } \gamma_i \geq 0, \\ b_i^-, & \text{if } \gamma_i < 0, \end{cases} \tag{3.7}$$

as our optimal solution to (3.4). Notice that $\bar{z} \in \mathbb{Z}^m$, since both b^- and b^+ are elements of $\mathbb{Z}^m$. Moreover, it is clear that if z^* satisfies the condition that $A^{-1}z^* = x^* \in \mathbb{Z}^m$ then z^* must also be integer, for if $A^{-1}z^* \in \mathbb{Z}^m$ then $z^* = A(A^{-1}z^*) \in \mathbb{Z}^m$ since both A and $A^{-1}z^*$ have all integer elements. Thus in searching for a solution to (3.5) <u>we need only consider vectors $z \in \mathbb{Z}^m$.</u>

If $A^{-1}\bar{z} \in \mathbb{Z}^m$ then (3.5) is solved; if not, then we must find a feasible vector of integers z^* with the properties that $A^{-1}z^* \in \mathbb{Z}^m$ and that no feasible z' exists such that $A^{-1}z' \in \mathbb{Z}^m$ and $\gamma z' > \gamma z^*$. Finding such a vector z^* is logically equivalent to finding a vector of integers v^* with the properties that

(i) $b^- \leq (\bar{z} + v^*) \leq b^+$,

(ii) $A^{-1}(\bar{z} + v^*) \in \mathbb{Z}^m$,

(iii) $\not\exists$ v satisfying (i) and (ii) such that $\gamma(\bar{z} + v) > \gamma(\bar{z} + v^*)$.

Observe that the vector v^* can be viewed as a "correction" to the optimal linear programming solution $\bar{z}$, whose purpose is to force the components of $x^* = A^{-1}(\bar{z} + v^*)$ to take on integer values. Property (i) requires that the "corrected" vector be feasible, and property (iii) requires that v^* be a least costly feasible "correction" in the sense that v^* must cause a minimal decrease in the value of the objective function.

Property (i) is equivalent to

(i') $(b^- - \bar{z}) \leq v^* \leq (b^+ - \bar{z})$,

which together with (3.6) shows that

$$v_i^* \in \begin{cases} [b_i^- - b_i^+, 0], & \text{if} \quad \gamma_i \geq 0, \\ [0, b_i^+ - b_i^-], & \text{if} \quad \gamma_i < 0. \end{cases} \tag{3.8}$$

Notice that v_i^* and γ_i are of opposite sign except in the case that $\gamma_i = v_i = 0$. Property (ii) is equivalent to

(ii') $$A^{-1}v^* \equiv -A^{-1}(\bar{z}) \pmod 1,$$

where congruence is taken elementwise. Property (iii) is clearly equivalent to

$$\text{(iii')} \qquad \gamma v^* \geq \gamma v, \qquad \forall\, v \in \mathbb{Z}^m \quad \text{satisfying (i') and (ii')}.$$

Let

$$(3.9) \qquad \mathcal{V} = \{(v_1, \ldots, v_m) \mid v_i \in \mathbb{Z} \text{ and satisfies (3.8) } \forall\, i\};$$

then a vector v satisfies (i'), (ii') and (iii') if and only if it is a solution to

$$(3.10) \qquad \max \ \gamma v$$

$$\text{subject to} \quad A^{-1}v \equiv -A^{-1}\bar{z} \pmod 1, \quad v \in \mathcal{V}.$$

Define the ordering $\succcurlyeq$ on $\mathcal{V}$ by the relation

$$v \succcurlyeq v' \iff \gamma v \geq \gamma v';$$

then $v \succcurlyeq v'$ if and only if the correction v' is no more costly than v.

We may give a general statement of the enumerative algorithm for solving (3.1).

<u>Implicit Enumeration Algorithm (Nonsingular Case)</u>.

(1) Set $\gamma = cA^{-1}$ and set

$$\bar{z}_i = \begin{cases} b_i^+, & \text{if } \gamma_i \geq 0, \\ b_i^-, & \text{if } \gamma_i < 0. \end{cases} \qquad i = 1, 2, \ldots, m.$$

Define

$$\mathcal{V} = \{(v_1, \ldots, v_m) \mid v_i \in \mathbb{Z} \text{ and } v_i \text{ satisfies (3.8)}, \forall\ i\}.$$

Compute $-A^{-1}\bar{z} \pmod 1$.

(2) Order the set $\mathcal{V}$ by the ordering $\succcurlyeq$ defined in (3.10), and sort the elements of $\mathcal{V}$ so that

$$v^0 \succcurlyeq v^1 \quad \ldots \succcurlyeq v^M$$

where M is the cardinality of $\mathcal{V}$. Set $k = 0$.

(3) Let $k = k+1$. If $A^{-1}v^k \equiv -A^{-1}\bar{z} \pmod 1$ go to (4); otherwise go to (3).

(4) Set $x^* = A(\bar{z} + v^k)$; x^* is a solution to (3.1).

Before discussing the geometrical interpretation and some computational aspects of this scheme, we demonstrate its use with a sample calculation.

Consider the problem

(3.11) $\max\ x_1 - x_2$

subject to

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix} \leq \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 8 \\ 15 \end{bmatrix}, \quad x_1, x_2 \in \mathbb{Z},$$

so that in our earlier terminology

$$c = (1 \quad -1), \quad b^- = \begin{bmatrix} -3 \\ 2 \end{bmatrix}, \quad b^+ = \begin{bmatrix} 8 \\ 15 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} -3/5 & 2/5 \\ 4/5 & -1/5 \end{bmatrix} .$$

We compute

$$\gamma = cA^{-1}$$
$$= (-7/5 \quad 3/5)$$

and find that

$$\bar{z} = \begin{bmatrix} -3 \\ 15 \end{bmatrix}$$

and

$$-A^{-1}\bar{z} = \begin{bmatrix} -39/5 \\ 27/5 \end{bmatrix}$$

$$\equiv \begin{bmatrix} 1/5 \\ 2/5 \end{bmatrix} \text{ (mod 1)}.$$

The set of feasible corrections is seen to be

(3.12) $$\mathcal{V} = \{0, 1, \ldots, 11\} \times \{0, -1, \ldots, -13\}.$$

In Table I we list the initial six elements of the sorted list

$$v^0 \succcurlyeq v^1 \succcurlyeq \cdots \succcurlyeq v^M$$

together with their costs and their respective products $A^{-1}v^k$ (mod 1).

Since $v^2 = (0 \;\; -2)$ is the first correction in the list for which $A^{-1}v^k \equiv -A^{-1}\bar{z}$ we conclude that

$$\begin{aligned} z^* &= v^* + \bar{z} \\ &= (0 \;\; -2) + (-3 \;\; 15) \\ &= (-3 \;\; 13) \end{aligned}$$

is a solution to (3.5) and that

$$\begin{aligned} x^* &= A^{-1}z^* \\ &= \begin{bmatrix} -3/5 & 2/5 \\ 4/5 & -1/5 \end{bmatrix} \begin{bmatrix} -3 \\ 13 \end{bmatrix} \\ &= \begin{bmatrix} 7 \\ -5 \end{bmatrix} \end{aligned}$$

is an optimal solution to (3.1).

Table I. Six least costly corrections for problem (3.11).

v^k	Correction	Cost ($= \gamma v^k$)	$A^{-1}v^k$ (mod 1)
v^0	(0 0)	0	(0 0)'
v^1	(0 -1)	-3/5	(3/5 1/5)'
v^2	(0 -2)	-6/5	(1/5 2/5)'
v^3	(1 0)	-7/5	(2/5 4/5)'
v^4	(0 -3)	-9/5	(4/5 3/5)'
v^5	(1 -1)	-10/5	(0 0)'
⋮	⋮	⋮	⋮

It is interesting to reflect on the geometrical interpretation of this enumerative scheme for solving (3.1). Define

$$R_x = \{x \in \mathbb{R}^m \mid b^- \leq Ax \leq b^+\} \tag{3.13}$$

and

$$R_z = \{z \in \mathbb{R}^m \mid b^- \leq z \leq b^+\}; \tag{3.14}$$

these are the feasible regions for the continuous linear programs (3.3) and (3.4) respectively. Define

$$I_x = R_x \cap \mathbb{Z}^m \tag{3.15}$$

and

$$I_z = R_z \cap \mathbb{Z}^m; \tag{3.16}$$

I_x is the feasible region for (3.1) and I_z contains (but is not in general equal to) the feasible region for (3.5), which is

$$I'_z = \{Ax \mid x \in I_x\}. \tag{3.17}$$

R_x is a parallelopiped in $\mathbb{R}^m$, and Gomory's cutting plane method evolves from an optimal continuous solution (at a vertex of R_x) to an optimal discrete solution by introducing a sequence of constraints that progressively cuts away portions of R_x until an element of I_x is revealed as a vertex of the remaining region; this element of I_x is then a solution to the integer programming problem. The enumerative scheme may also be viewed as a cutting plane method, but as one that operates on R_z, which is a rectangular parallelopiped in $\mathbb{R}^m$ with elements of I_z at each vertex. Figures 7 and 8 show these regions for the sample problem just discussed. Let $\bar{z}$ be the optimal linear programming solution found in step (1) of the algorithm and let

$$\gamma z = \gamma \bar{z} \tag{3.18}$$

be the hyperplane through $\bar{z}$ perpendicular to γ. Let

$$= \{v^0 \succcurlyeq v^1 \succcurlyeq \ldots \succcurlyeq v^M\}$$

be the set of feasible integer corrections to $\bar{z}$ as defined earlier, and suppose the hyperplane $\gamma z = \gamma \bar{z}$ were given a parallel translation into R_z until it encountered an element of I_z. If we assume $v^0 = \vec{0}$, the first element of I_z the hyperplane would meet is $\bar{z} + v^1$; in fact, if the hyperplane were translated through R_z, it would meet the elements of I_z in precisely the order $\bar{z}, \bar{z} + v^1, \bar{z} + v^2, \ldots, \bar{z} + v^M$ (up to permutations of elements of for which $v^i \succcurlyeq v^j \succcurlyeq v^i$). Thus we may view the sequential enumeration of the elements of $\mathcal{V}$ as a progressive cutting away of the feasible region R_z by a sequence of cutting planes perpendicular to γ; when finally an integer point $\bar{z} + v^k$ is encountered for which $A^{-1}(\bar{z} + v^k) \equiv 0 \pmod 1$ the integer programming problem is

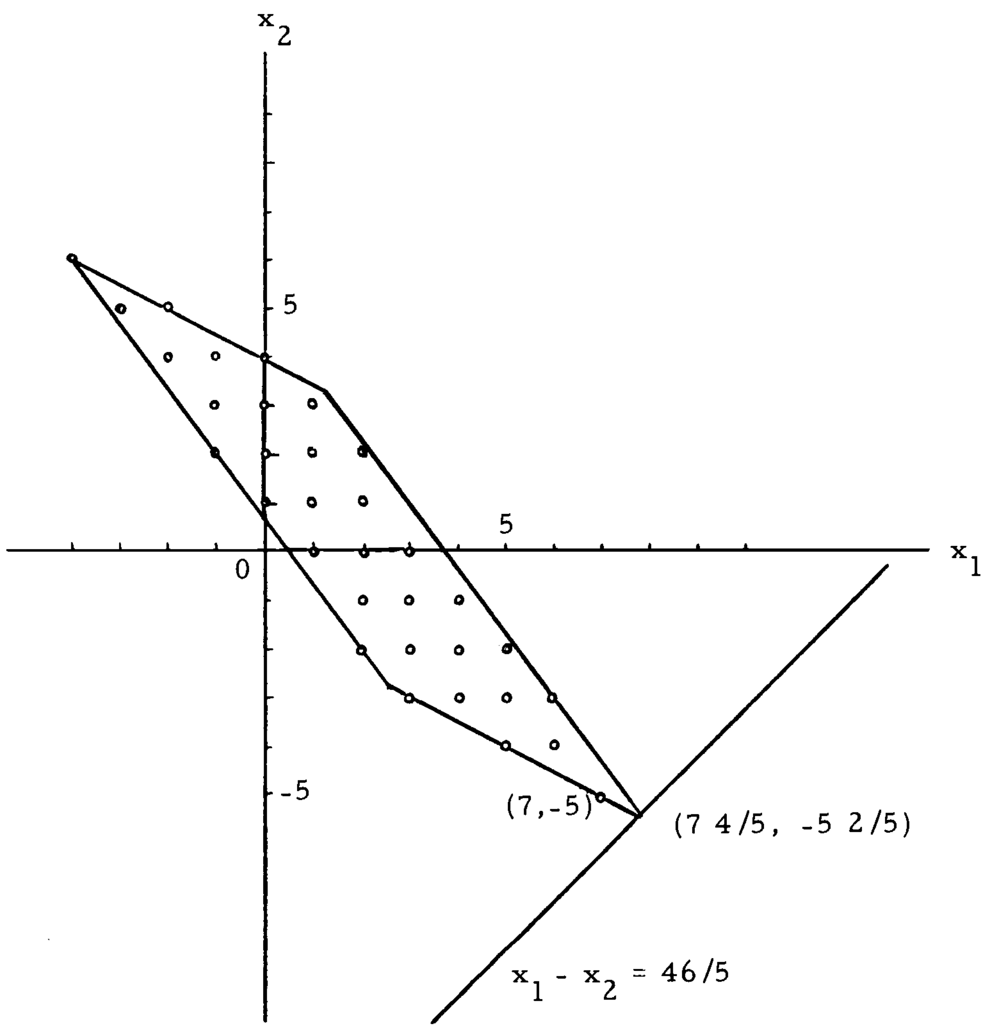

Figure 7. Geometrical representation of problem (3.11). The parallelogram is R_x, and the lattice points within the parallelogram are I_x.

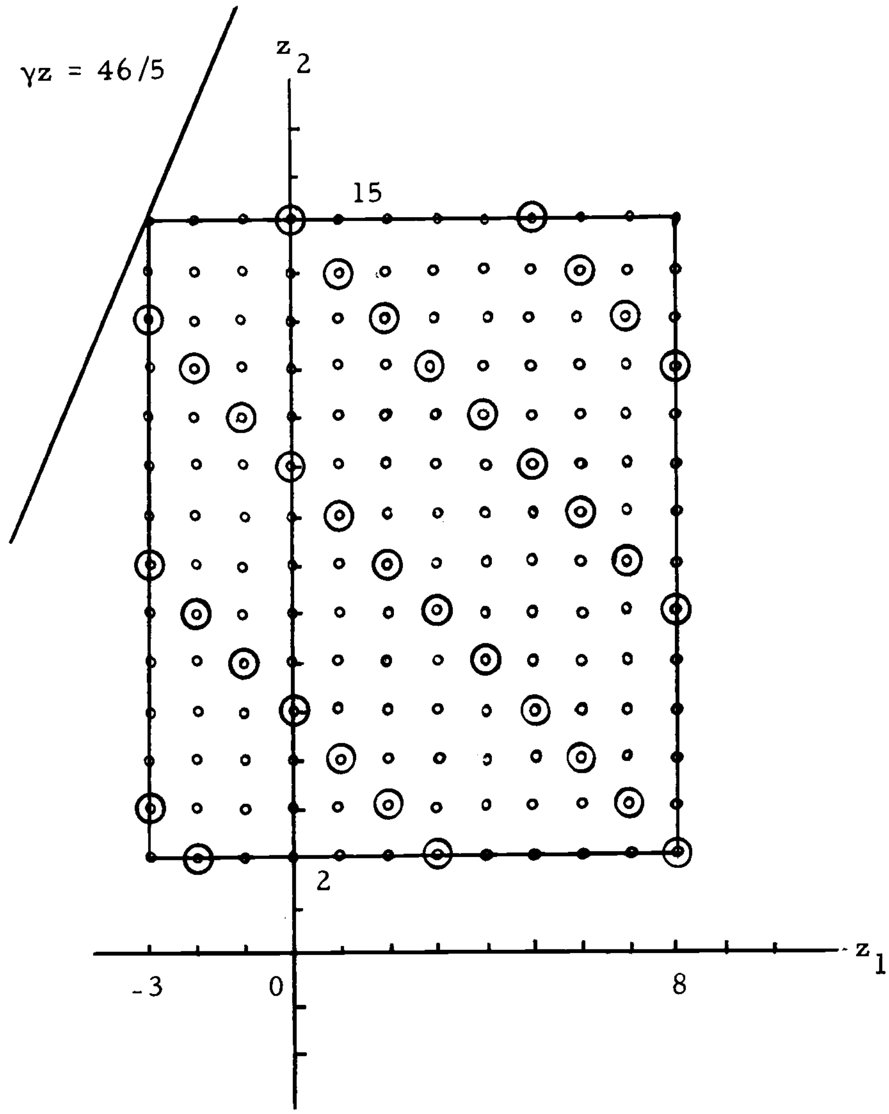

Figure 8. Geometrical representation of transformed equivalent of problem (3.11). Rectangular region is R_z; small circles (∘) denote points of I_z and large circles (○) denote points of I'_z.

essentially solved.

For problems of any appreciable size, the most demanding portion of the enumerative scheme from a computational viewpoint is the generation and sorting of the set $\mathcal{V}$ of feasible integer corrections. We give two results that provide some relief in constructing $\mathcal{V}$.

Proposition 4. Let $D = \det(A)$. Then $|v_i^*| \leq |D|-1$ for $i = 1, \ldots, m$ for the optimal $v^* \in \mathcal{V}$.

Proof: Let $i \in \{1, \ldots, m\}$; then A_i^{-1}, the ith column of A^{-1}, has entries all of which may be expressed as rational numbers with denominator $|D|$. Now, it is clear that for any integers a and b with $b \leq |D|-1$ we have

$$(3.18) \qquad (a|D|+b)A_i^{-1} \equiv bA_i^{-1} \pmod 1.$$

Since γ_i and v_i^* are of opposite sign, we know that in solving (3.10) we will want v_i^* no larger in absolute value than is necessary. From (3.18) we see that if $|v_i^*| \geq |D|$ we may replace v_i^* with $v_i^{*\prime} = v_i^* \pmod D$ without destroying the relationship $A^{-1}v^* = -A^{-1}\bar{z}$ and without prejudicing the optimality of v^*. We may therefore assume that $|v_i^*| \leq |D|-1 \;\; \forall\, i$. This completes the proof.

The significance of Proposition 4 is that the size of $\mathcal{V}$ may be effectively reduced by ignoring all $v \in \mathcal{V}$ for which some element v_i exceeds $|D|-1$; we may substitute for $\mathcal{V}$ the set

$$\mathcal{V}' = \{(v_1, \ldots, v_m | \, |v_i| \leq |D|-1 \text{ and } v_i \text{ satisfies (3.8)}, \ \forall i\},$$

which will be smaller than $\mathcal{V}$ whenever some $b_i^+ - b_i^- \geq |D|-1$.

In fact, even more is true; it can be shown (Gomory, 1969) that we may assume $\sum_1^m |v_i^*| \leq |D|-1$. Although this result is not difficult to prove in the context of Hu (1969) or Gomory (1969), it requires a great deal of effort to set up the algebraic machinery necessary for a compact proof. We content ourselves with simply stating the result and observing that it allows even further reduction of $\mathcal{V}$.

Next, we state and prove a result that allows us to establish an upper bound on the number of elements of $\mathcal{V}'$ that must be investigated before a solution v^* to (3.10) is found.

<u>Proposition 5</u>. Let N and k be positive integers. Then the number of vectors $u \in (\mathbb{Z}^+)^k$ for which

$$\vec{1} \cdot u = N$$

is given by

$$\sum_{j=1}^{k} \binom{k}{j} \binom{N-1}{j-1} . \tag{3.19}$$

<u>Proof</u>: It is well-known (see e.g., Hall (1967)) that N can be expressed as the sum of j positive integers in exactly

$$\binom{N-1}{j-1} \tag{3.20}$$

ways, where we agree to let $\binom{N-1}{j-1} = 0$ whenever $N < j$; this is simply the number of possible ways of placing $j-1$ markers in the spaces between N objects arranged in a line. Now, there are $\binom{k}{j}$ ways to choose j elements of u to take on positive values, so the total number of vectors $u \in (\mathbb{Z}^+)^k$ whose elements sum to N is

$$\sum_{j=1}^{k} \binom{k}{j} \binom{N-1}{j-1} .$$

This completes the proof.

From Proposition 5, then, we see that the number of vectors $u \in (\mathbb{Z}^+)^m$ for which

$$\sum_{i}^{m} u_i \leq |D| - 1$$

is

$$1 + \sum_{n=1}^{|D|-1} \sum_{j=1}^{m} \binom{m}{j} \binom{n-1}{j-1} ; \tag{3.21}$$

the 1 counts the zero vector and the double sum counts the nonzero vectors. Since

$$(3.22) \qquad \sum_{1}^{m} |v_i^*| \leq |D| - 1$$

and the sign of each v_i^* is predetermined (except for the trivial possibility that $v_i^* = 0$), we may use (3.21) to arrive at the following result.

<u>Proposition 6</u>. Consider problem (3.1) with $A \in \mathbb{Z}^{m,m}$ and $\det(A) = D$. If the implicit enumeration algorithm is employed to solve (3.1), then a solution will be revealed after enumerating no more than

$$(3.23) \qquad 1 + \sum_{n=1}^{|D|-1} \sum_{j=1}^{m} \binom{m}{j} \binom{n-1}{j-1}$$

elements of $\mathcal{V}'$, provided they are listed in descending $\succcurlyeq$ order and that all v not meeting condition (3.22) are ignored.

For example, if $A \in \mathbb{Z}^{2,2}$ and $\det(A) = 3$ then for any b^-, b^+ and $c \in \mathbb{Z}^2$ such that $b^- \leq b^+$ and $c \perp N(A)$ the problem (3.1) can be solved by querying no more than

$$1 + \sum_{n=1}^{3} \sum_{j=1}^{2} \binom{2}{j}\binom{n-1}{j-1} = 1 + \binom{2}{1}\binom{0}{0} + \binom{2}{2}\binom{0}{1}$$

$$+ \binom{2}{1}\binom{1}{0} + \binom{2}{2}\binom{1}{1}$$

$$= 1 + [2 + 0] + [2 + 1]$$

$$= 6$$

of the least costly correction vectors satisfying (3.22). We may verify that this is correct by observing that the only possible correction vectors v with $\sum |v_i| \leq |D| - 1 = 2$ are (up to changes in sign)

$$\begin{array}{cc} (0\ \ 0) & (1\ \ 1) \\ (0\ \ 1) & (2\ \ 0) \\ (1\ \ 0) & (0\ \ 2), \end{array}$$

which are precisely six in number. Notice that Proposition 6 establishes a bound that depends only on the size and determinant of A; it is totally independent of b^+ and b^-. If for some $i \in \{1, \ldots, m\}$ we have $(b_i^+ - b_i^-) < |D| - 1$ then strictly fewer vectors than the bound given by Proposition 6 will need to be queried, but it seems difficult to compute a precise upper bound in this case. One obvious bound can be computed by setting

$$(3.24) \qquad \delta_i = \min\{(b_i^+ - b_i^-), |D| - 1\} \quad \forall\, i \in \{1, \ldots, m\}$$

and computing

$$\prod_{1}^{m} \delta_i; \tag{3.25}$$

this is plainly an upper bound on the number of vectors $v \in \mathcal{V}'$ which must be considered, but it can be expected to severly overestimate.

We may make further use of Proposition 4 in showing how the problem

$$\begin{aligned} &\max \quad cx \\ &\text{subject to} \quad b^- \le Ax \le b^+, \quad x \in \mathbb{Z}^n, \end{aligned} \tag{3.26}$$

may be solved, where $A \in \mathbb{Z}^{m,n}$ is of full row rank and $c \perp N(A)$. As in Chapter II, we assume that A may be partitioned as

$$A = [B|N],$$

where $B \in \mathbb{Z}^{m,m}$ is nonsingular; then if we compute the particular generalized inverse

$$A^- = \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix} \tag{3 27}$$

and let

$$\gamma = cA^- \tag{3.28}$$

and

$$\bar{z}_i = \begin{cases} b_i^+, & \text{if } \gamma_i \geq 0, \\ b_i^-, & \text{if } \gamma_i < 0, \end{cases}$$

we have

$$\bar{x} = (B^{-1}\bar{z} \mid 0) \tag{3.29}$$

an optimal solution to the continuous analog of (3.26).

Now, let x^* be an optimal solution to the integer programming problem (3.26) and let $z^* = Ax^*$. If $B^{-1}z^* \in \mathbb{Z}^m$ then

$$x^{*\prime} = (B^{-1}z^* \mid 0)$$

is also a solution to (3.26). If $B^{-1}z^* \notin \mathbb{Z}^m$ then there exists a vector $\hat{x}_N$ such that

$$\hat{x}_B = B^{-1}(z^* - N\hat{x}_N) \in \mathbb{Z}^m, \tag{3.30}$$

which is equivalent to

$$B^{-1}N\hat{x}_N \equiv B^{-1}z^* \pmod 1. \tag{3.31}$$

Since all solutions to $Ax = z^*$ are equally costly[1] we know $cx^* = c_B(B^{-1}z^*) = c(x_B \mid x_N)$; we may then think of x_N as a gratis nonbasic correction to $(B^{-1}z^* \mid 0)$ which forces the basic variables to assume integer values. Then we may employ precisely the same

---

[1]Suppose $Ax = Ax' = z^*$; $c \perp N(A)$ so $\exists\, t \in \mathbb{R}^m \ni c = tA$. Then $cx = (tA)x = t(Ax) = t(Ax') = cx'$.

argument used to prove Proposition 4 to show that each $|x_{Ni}|$ may be assumed to be less than $|\det(B)|$ in absolute value; this is because every element of the matrix $B^{-1}N$ can be expressed as a rational number with denominator $\det(B) = D$.

Since each column of $B^{-1}N$ generates a cyclic subgroup of $\mathbb{R}^m \pmod 1$ whose order divides $|\det(B)|$ we know that for any integer k

$$-k(B^{-1}N)_i \equiv [|\det(B)| - k(\mathrm{mod}\,|\det(B)|)](B^{-1}N)_i \pmod 1$$

where $(B^{-1}N)_i$ is the ith column of $B^{-1}N$; thus we may assume that

$$(3.32) \qquad 0 \le \hat{x}_{Ni} \le |D| - 1, \qquad \forall\, i \in \{1, 2, \ldots, n-m\}$$

since there is no need for any nonbasic variable to assume a negative value. Therefore our original problem (3.26) is equivalent to

(3.33) $\quad \max \ cx$

subject to

$$\begin{bmatrix} b^- \\ 0 \end{bmatrix} \le \begin{bmatrix} B & N \\ 0 & I \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} \le \begin{bmatrix} b^+ \\ \delta \end{bmatrix}, \quad x_B \in \mathbb{Z}^m, \quad x_N \in \mathbb{Z}^{n-m},$$

where $\delta = (|D| - 1)\vec{1} \in \mathbb{Z}^{n-m}$. We see by inspection that the matrix

$$\begin{bmatrix} B & N \\ 0 & I \end{bmatrix}$$

is nonsingular, so we have expressed the original problem in a form that can be solved by the original enumerative algorithm for the nonsingular case.

The geometry of this transformation of problem (3.26) into problem (3.33) is interesting. The feasible region for (3.26) is a parallelopiped in $\mathbb{R}^n$ that is "open" in $n-m$ dimensions. Figure 9 is a sketch of a feasible region for a problem where $A \in \mathbb{Z}^{2,3}$ has full row rank. Here we have taken x_1 and x_2 to be the basic variables and x_3 as the single nonbasic variable. Now, when we introduce the additional constraints

$$0 \leq x_{Ni} \leq |D|-1 \qquad \forall i \in \{1, 2, \ldots, n-m\}$$

we are effectively closing up the open ends of the parallelopiped. By the rather careful argument used to derive inequality (3.32) we can be certain that we have not cut off all the optimal points in the feasible region; at least one x^* that solves (3.26) is certain to remain in the new compact feasible region. Figure 10 illustrates the manner in which this occurs.

In practice, one would not wish to explicitly construct the set $\mathcal{V}'$ of corrections and sort it in $\succcurlyeq$ order; an efficient method for

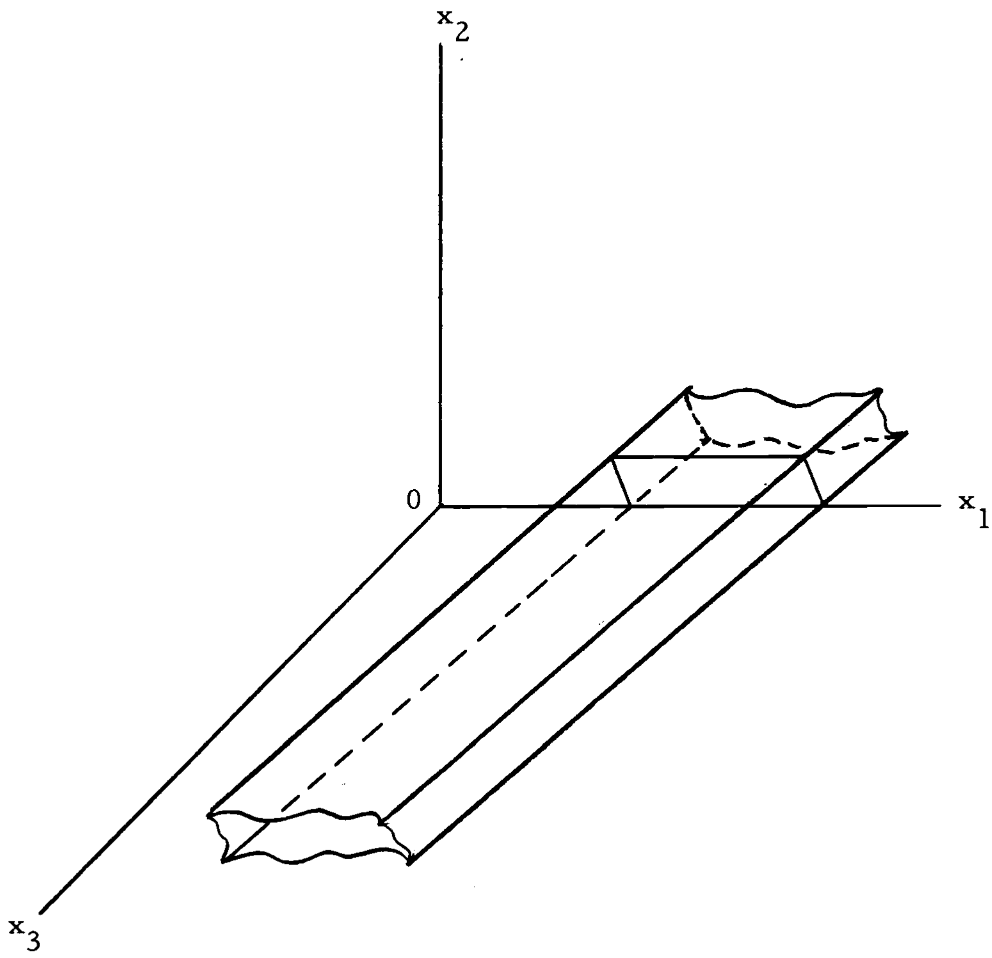

Figure 9. Sketch of a feasible region for a problem with $A \in \mathbb{Z}^{2,3}$.

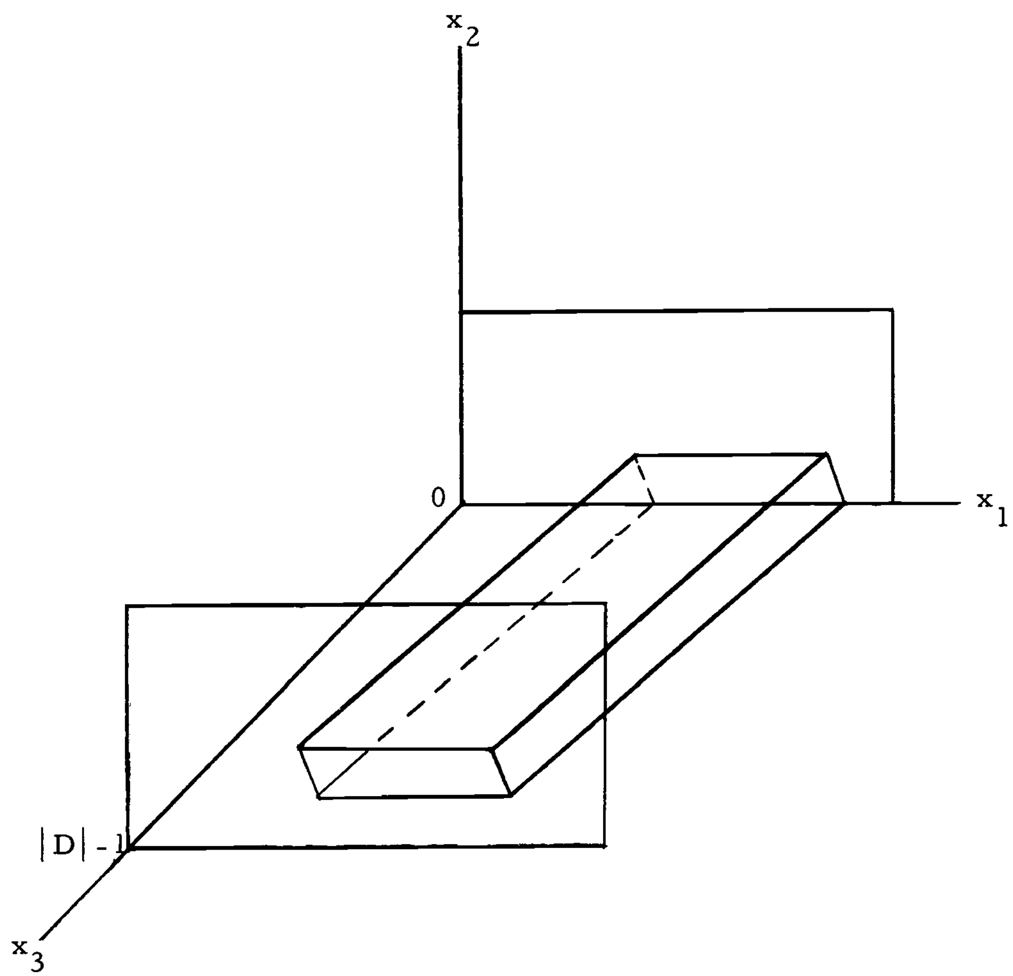

Figure 10. Sketch of a reduced feasible region for a problem with $A \in \mathbb{Z}^{2,3}$.

generating the elements of $\mathcal{V}'$ in $\succcurlyeq$ order would have obvious advantages. The task of generating these corrections in exactly the proper sequence appears to be a difficult combinatorial problem. It is not difficult to devise methods for generating these elements of $\mathcal{V}'$ in an orderly (if not ordered) fashion, though, so the proposed algorithm is of more than purely academic interest. A good deal of computational experimentation would be required to determine a generation scheme that would be in some sense best; we have not dealt with this problem in this paper.

BIBLIOGRAPHY

Balinski, M. L. 1967. Some general methods in integer programming. In: Nonlinear programming, ed. by J. Abadie, Amsterdam, North-Holland. p. 221-247.

Ben-Israel, A. and A. Charnes. 1968. An explicit solution of a special class of linear programming problems. Operations Research 16:1166-1175.

Ben-Israel, Adi and Philip D. Robers. 1970. A decomposition method for interval linear programming. Management Science 16:374-387.

Gomory, R. E. and W. J. Baumol. 1960. Integer programming and pricing. Econometrica 28:521-550.

Gomory, R. E. 1965. On the relation between integer and non-integer solutions to linear programs. Proceedings of the National Academy of Sciences 53:260-265.

____________ 1967. Integer faces of a polyhedron. Proceedings of the National Academy of Sciences 57:16-18.

____________ 1969. Some polyhedra related to combinatorial problems. Linear Algebra and Its Applications 2:451-558.

Hall, Marshall Jr. 1967. Combinatorial theory. Waltham, Blaisdell. 310 p.

Hoffman, A. J. and J. B. Kruskal. 1956. Integral boundary points of convex polyhedra. In: Linear inequalities and related systems, ed. by H. W. Kuhn and A. W. Tucker, Princeton, Princeton University Press. p. 233-246.

Hu, T. C. 1969. Integer programming and network flows. Reading, Addison-Wesley. 452 p.

Rubin, David S. 1969. Integral solutions of integral linear systems. Chicago. 21 p. (University of Chicago. Report no. 6930)

Shapiro, Jeremy F. 1968a. Dynamic programming algorithms for the integer programming problem-I: the integer programming problem viewed as a knapsack problem. Operations Research 16:103-121.

________________ 1968b. Group theoretic algorithms for the integer programming problem II: extension to a general algorithm. Cambridge. 33 p. (Massachusetts Institute of Technology, Operations Research Center. Technical Report no. 38. U.S. Army Research Office Contract no. DA-31-124-ARO-D-209. Department of the Army Project no. 20011501B704.)

APPENDICES

APPENDIX 1: GOMORY'S CUTTING PLANE METHOD

In this appendix we will give a brief summary of Gomory's cutting plane algorithm for solving integer programming problems. An elementary discussion of the algorithm can be found in Balinski (1967). Gomory's scheme consists of a sequence of operations performed on the optimal simplex tableau of the associated continuous linear program, and although his method can be adapted to suit any configuration of simplex tableau we will find it most convenient to employ the so-called extended Tucker tableau. We shall first sketch the simplex method in the setting of this particular tableau and then discuss the cutting plane method in that context.

Consider the linear programming problem

$$\text{(A1.1)} \qquad \max \ cx \qquad \text{subject to} \quad Ax \leq b, \quad x \geq 0.$$

If we introduce the appropriate nonnegative slack variables we may write this program as

$$\text{(A1.2)} \qquad \max \ cx \qquad \text{subject to} \quad x \geq 0, \quad s = b - Ax \geq 0$$

or, in matrix notation,

(A1.3) max cx

subject to

$$\begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} 0 & -I \\ b & A \end{bmatrix} \begin{bmatrix} 1 \\ -x \end{bmatrix} \geq \begin{bmatrix} 0 \end{bmatrix}.$$

For uniformity of notation it is convenient to let $a_{n+i,0} = b_i$ and $x_{n+i} = s_i$ for all $i \in \{1, \ldots, m\}$, so that problem (A1.3) becomes

(A1.4) max cx

subject to

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{m+n} \end{bmatrix} = \left[\begin{array}{c|c} 0 & -I \\ \hline a_{n+1,0} & \\ \vdots & A \\ a_{n+m,0} & \end{array}\right] \begin{bmatrix} 1 \\ -x_1 \\ -x_2 \\ \vdots \\ -x_n \end{bmatrix} \geq \begin{bmatrix} 0 \end{bmatrix}.$$

Now let $a_{0j} = -c_j$ for all $j \in \{1, \ldots, n\}$ and construct the extended Tucker tableau

(A1.5)

$$
\begin{array}{c|c|cccc|}
 & 1 & -x_1 & -x_2 & \cdots & -x_n \\
\hline
z & 0 & a_{01} & a_{02} & \cdots & a_{0n} \\
\hline
x_1 & & & & & \\
x_2 & & & & & \\
\vdots & 0 & & -I & & \\
x_n & & & & & \\
\hline
x_{n+1} & a_{n+1,0} & & & & \\
\vdots & & & A & & \\
x_{n+m} & a_{n+m,0} & & & & \\
\hline
\end{array}
\quad .
$$

The first row is essentially the objective functional, and each row after the first corresponds to one of the nonnegativity constraints in problem (A1.4). The two column vectors in (A1.4) are dropped and the two sets of labels are installed in their place.

Tableau (A1.5) can be thought of as the starting tableau for problem (A1.2) with the initial solution given by

(A1.6)
$$
x_j = \begin{cases} 0, & \text{if } \; j \in \{1, \ldots, n\}, \\ b_i, & \text{if } \; j = n+i \in \{n+1, \ldots, n+m\}. \end{cases}
$$

We will now show how one performs both primal and dual simplex pivot operations on (A1.5). We assume that either $a_{0j} \geq 0 \;\; \forall \, j$ (i.e. dual feasibility) or that $a_{n+i,0} \geq 0 \;\; \forall \, i$ (i.e., primal feasibility). We know that if a primal feasible solution exists then a two-phase method will produce one, and that similarly a dual feasible solution can be derived if one exists. We are not being overly restrictive, then, in assuming that we have either primal or dual feasibility at the start.

First, consider the case in which $a_{n+i,0} \geq 0 \;\forall i$; then the solution given by (A1.6) is primal feasible. To perform a primal simplex step we do the following:

(i) Let

$$a_{0s} = \min_j a_{0j}, \quad j \in \{1, \ldots, n\}.$$

If $a_{0s} \geq 0$ then the current solution is optimal; otherwise column s will be the pivot column.

(ii) Let

$$\left|\frac{a_{r0}}{a_{rs}}\right| = \min_i \left|\frac{a_{i0}}{a_{is}}\right|$$

over all i for which $a_{is} > 0$. Then a_{rs} is the pivot element.

(iii) Divide every element of column s by $-a_{rs}$ and use this rescaled column to eliminate all nonzero elements of row r.

(iv) Assign row r's label to column s.

As with any primal method, a nonbasic variable is first chosen to enter the basis and then a basic variable is selected to leave. After this pivot has been performed, the new element a_{00} is the value of the objective functional at the solution given by the remaining elements of the 0th column (together with their labels), and the elements of the 0th row are the "reduced costs" or "transformed cost coefficients" associated with the nonbasic variables. This interpretation of the 0th row and column will be applicable throughout the entire discussion.

Now consider the case in which $a_{0j} \geq 0 \quad \forall\, j \in \{1, \ldots, n\}$; then the solution (A1.6) is dual feasible. A dual simplex step is performed in the following manner:

(i) Let

$$a_{r0} = \min_i \, a_{i0}, \quad i \in \{1, \ldots, n+m\}.$$

If $a_{r0} \geq 0$ then the current solution is both primal and dual feasible, hence optimal. Otherwise row r will be the pivot row.

(ii) Let

$$\left|\frac{a_{0s}}{a_{rs}}\right| = \min_j \left|\frac{a_{0j}}{a_{rj}}\right|$$

over all j such that $a_{rj} < 0$. Then element a_{rs} is the pivot element.

(iii) Divide every element of column s by $-a_{rs}$ and use this rescaled column to eliminate all nonzero elements of row r.

(iv) Assign row r's label to column s.

As with any dual scheme, the variable to exit from the basis is chosen first and then a nonbasic variable is selected to replace it.

By iterated application of these two basic operations it is possible to solve any linear programming problem possessing an optimal feasible solution. If all elements of b are nonnegative we have an immediate primal feasible solution given by (A1.6), and performing primal simplex steps until all the reduced cost coefficients are nonnegative will yield an optimal solution given by the 0th column of the tableau and its adjoining vector of labels. Similarly, if all elements of the cost vector c are nonpositive then dual simplex steps, performed until all elements of the tableau's 0th column are nonnegative, will yield an optimal solution. As was mentioned previously, if neither b nor $-c$ is nonnegative then a standard two-phase scheme may be used to determine a primal or dual feasible solution (if one exists).

Notice that this configuration of the simplex algorithm is simply the usual simplex method with the basic variables expressed in terms of the nonbasic variables. This offers no particular advantage in solving the linear programming problem, but it proves to be most convenient for integer programming.

Consider the integer programming problem

$$\text{(A1.11)} \qquad \max \; cx \quad \text{subject to} \quad Ax \leq b, \quad x \in (\mathbb{Z}^+)^n,$$

where all elements of A, b and c are integers and A is of full row rank. As usual assume A to be $m \times n$. As we have seen above, the continuous linear program

$$\text{(A1.12)} \qquad \max \; cx \quad \text{subject to} \quad Ax \leq b, \quad x \geq 0,$$

can be solved by the primal or dual simplex method provided an optimal feasible solution exists. Let T be the optimal simplex tableau for the linear program (A1.12), written in extended Tucker form. Since the optimal solution is both primal and dual feasible, we have

$$\text{(A1.13)} \qquad t_{i0} \geq 0 \quad \forall\, i \in \{1, \ldots, n+m\},$$

and

$$\text{(A1.14)} \qquad t_{0j} \geq 0 \quad \forall\, j \in \{1, \ldots, n\}.$$

We also observe that if B is the optimal basis for problem (A1.12) then every element of the tableau T can be expressed as a rational number with denominator $|\det(B)|$. This is true of any optimal simplex tableau written in any form (not necessarily in the extended Tucker form) and is an easy exercise to verify.

Without going through the derivation, we now set down the procedure for solving the integer programming problem (A1.11) by Gomory's cutting plane method.

(i) Solve the continuous linear program (A1.12) corresponding to (A1.11).

(ii) If t_{i0} is integer for all $i \in \{1, \ldots, n+m\}$ the integer program is solved; otherwise let σ be the index of the first row for which a_{i0} is not integer. From this row (the source row) we derive an additional constraint (called a Gomory cut) as follows: let $f_{\sigma j}$ be the positive fractional part of $t_{\sigma j}$ $\forall$ $j \in \{0, \ldots, n\}$ and write

$$\text{(A1.15)} \qquad \sum_{j=1}^{n} (-x_j) f_{\sigma j} \geq f_{\sigma 0}$$

or, defining a new slack variable s, we may equivalently write

$$\text{(A1.16)} \qquad s = -f_{\sigma 0} - \sum_{j=1}^{n} (-x_j) f_{\sigma j} \geq 0.$$

Append the constraint (A1.16) to the bottom of the tableau T and perform a dual pivot operation, with the new row as the pivot row, to eliminate the primal infeasibility brought on by the fact that $-f_{\sigma 0}$ is negative.

(iii) If some a_{i0} with $i \in \{1, \ldots, n+m\}$ is negative after the pivot operation in (ii), perform a dual simplex step with row i as the pivot row to eliminate the infeasibility. Repeat until no primal infeasibilities remain.

(iv) Return to step (ii).

As an example we consider the following problem discussed in Hu (1969):

$$\text{(A1.17)} \qquad \max \quad 4x_1 + 5x_2 + x_3$$

subject to

$$\begin{bmatrix} 3 & 2 & 0 \\ 1 & 4 & 0 \\ 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \le \begin{bmatrix} 10 \\ 11 \\ 13 \end{bmatrix}, \quad x_i \text{ integer for } i = 1,2,3.$$

Since the right-hand side vector b is nonnegative, we may introduce slack variables x_4, x_5 and x_6 and construct the primal feasible extended Tucker tableau

(A1.18)

	1	$-x_1$	$-x_2$	$-x_3$
z	0	-4	-5	-1
x_1	0	-1	0	0
x_2	0	0	-1	0
x_3	0	0	0	-1
x_4	10	3	2	0
x_5	11	1	4	0
x_6	13	3	3	1

.

To solve the integer problem (A1. 17) we first solve the corresponding continuous linear program by performing primal simplex steps on the above tableau until all the reduced cost coefficients t_{0j} are non-negative. To perform the first primal pivot we find the smallest reduced cost coefficient to be -5, so that the column labelled $-x_2$ will be the pivot column. To select the pivot row we compute, for each row i whose intersection with the pivot column 2 is strictly positive, the ratio

(A1. 19)
$$\left|\frac{t_{i0}}{t_{i2}}\right|$$

and then select the row for which this ratio is minimal. Thus the row labelled x_5 is seen to be the pivot row and the circled entry is the pivot element. The resulting tableaus and circled pivot entries necessary to solve the continuous linear program are given below.

(A1. 20)

	1	$-x_1$	$-x_5$	$-x_3$
z	55/4	-11/4	5/4	-1
x_1	0	-1	0	0
x_2	11/4	1/4	1/4	0
x_3	0	0	0	-1
x_4	18/4	10/4	-2/4	0
x_5	0	0	-1	0
x_6	19/4	9/4	-3/4	1

(A1.21)

	1	$-x_4$	$-x_5$	$-x_3$
z	187/10	11/10	-7/10	-1
x_1	18/10	4/10	-2/10	0
x_2	23/10	-1/10	3/10	0
x_3	0	0	0	-1
x_4	0	-1	0	0
x_5	0	0	-1	0
x_6	7/10	-9/10	-3/10	1

(A1.22)

	1	$-x_4$	$-x_5$	$-x_6$
z	194/10	2/10	4/10	1
x_1	18/10	4/10	-2/10	0
x_2	23/10	-1/10	3/10	0
x_3	7/10	-9/10	-3/10	0
x_4	0	-1	0	0
x_5	0	0	-1	0
x_6	0	0	0	-1

This tableau is both primal and dual feasible and is therefore optimal. The solution to the continuous linear program is

(A1.23) $$x^* = (18/10 \;\; 23/10 \;\; 7/10),$$

and the corresponding objective functional value is

(A1.24) $$cx^* = 194/10.$$

Clearly x^* does not solve the integer programming problem, so we proceed to derive a Gomory cut. For ease of exposition we do not derive our first cut from the row of least index i for which t_{i0} is noninteger (which would be the 0th row) but instead take the row labelled x_3 as the source row; this will yield an integer solution after only one pivot. The positive fractional parts of the elements of the row labelled x_3 are 7/10, 1/10, 7/10 and 0 respectively, so the constraint corresponding to (A1.16) is

(A1.25) $$-7/10 - (-x_4)1/10 - (-x_5)7/10 \geq 0.$$

Introducing the new slack variable s and appending this new constraint to the bottom of tableau (A1.22) gives the tableau

(A1.26)

	1	$-x_4$	$-x_5$	$-x_6$
z	194/10	2/10	4/10	1
x_1	18/10	4/10	-2/10	0
x_2	23/10	-1/10	3/10	0
x_3	7/10	-9/10	-3/10	1
x_4	0	-1	0	0
x_5	0	0	-1	0
x_6	0	0	0	-1
s	-7/10	-1/10	-7/10	0

.

This tableau is dual feasible and has a primal infeasibility in the last row; thus the row labelled s will be the pivot row for a dual simplex step. Performing the ratio test indicated by (A1.10) shows that the column labelled $-x_5$ is the pivot column, so the circled element above is the pivot element. Pivoting on this element yields the tableau

(A1.27)

	1	$-x_4$	$-s$	$-x_6$
z	19	1/7	4/7	1
x_1	2	3/7	-2/7	0
x_2	2	-1/7	3/7	0
x_3	1	-6/7	-3/7	1
x_4	0	-1	0	0
x_5	1	1/7	-10/7	0
x_6	0	0	0	-1
s	0	0	-1	0

which solves the integer programming problem (A1.17).

APPENDIX 2: INTERVAL LINEAR PROGRAMMING

Consider the interval linear program

(A2.1) $\max \; cx$

$$\text{subject to} \quad b^- \leq Ax \leq b^+, \quad x \quad \text{real},$$

where A, b^-, b^+ and c have all real elements. Ben-Israel and Charnes (1968) discuss this problem in some detail, giving a non-iterative method of solution for problems satisfying the following three conditions:

(i) a feasible solution to (A2.1) exists,

(ii) c lies in the row space of A, i.e. $c \perp N(A)$,

(iii) A is of full row rank.

In this appendix we will summarize the relevant results of that paper.

For any given real $m \times n$ matrix A, the unique real matrix T such that

(A2.2) $$ATA = A$$

(A2.3) $$TAT = T$$

(A2.4) $$(AT)' = AT$$

(A2.5) $$(TA)' = TA$$

is called the <u>Moore-Penrose generalized inverse</u> of A, which we shall denote by A^+. For our purposes, only the first property (A2.2) is required; we will denote by $A(1)$ the set of matrices T such that

$$ATA = A. \tag{A2.6}$$

The main theorem in Ben-Israel and Charnes' paper is the following.

<u>Theorem</u> (Ben-Israel and Charnes). Let problem (A2.1) meet conditions (i), (ii) and (iii), and let $T \in A(1)$ be an $n \times m$ matrix with columns $t_1, \ldots, t_m$. Then the optimal solutions to (A2.1) form a manifold

$$S = \sum_{i \in I^-} t_i b_i^- + \sum_{i \in I^+} t_i b_i^+ + \sum_{i \in I^0} t_i[\theta b_i^+ + (1-\theta) b_i^-] + N(A), \tag{A2.7}$$

where $\theta \in [0, 1]$ and

$$I^- = \{i \mid ct_i < 0\}, \tag{A2.8}$$

$$I^+ = \{i \mid ct_i > 0\}, \tag{A2.9}$$

$$I^0 = \{i \mid ct_i = 0\}. \tag{A2.10}$$

The essence of the proof is that the hypotheses imply that problem (A2.1) is equivalent to the problem

$$\begin{aligned} &\max \quad cTz \\ &\text{subject to } b^- \leq z \leq b^+, \quad z \in \mathbb{R}^m \end{aligned} \tag{A2.11}$$

The solution to this problem is obvious; the optimal vector z^* is given by

$$z_i^* = \begin{cases} b_i^-, & \text{if } ct_i < 0, \\ b_i^+, & \text{if } ct_i > 0, \\ \theta b_i^+ + (1-\theta)b_i^-, & \text{otherwise,} \end{cases} \tag{A2.12}$$

where θ is an arbitrary number in $[0, 1]$. Then $x^* = Tz^*$ is an optimal solution to (A2.1). If y is any vector in $N(A)$ then $x^* + y$ is also optimal, so the optimal solutions form a manifold whose dimension is that of $N(A)$.

Ben-Israel and Charnes also give a lemma that suggests a method for computing these simple generalized inverses. The important part of the lemma is this:

Lemma (Ben-Israel and Charnes). Let A be a real $m \times n$ matrix with rank r and let E be any $m \times m$ real matrix such that

$$EA = \begin{bmatrix} I_r & \Delta \\ 0 & 0 \end{bmatrix} P, \tag{A2.13}$$

where P is a permutation matrix. Partition E as

$$E = \begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix}, \tag{A2.14}$$

where E_1 is $r \times r$. Then

(A2.15) $$T = P' \begin{bmatrix} E_1 & E_2 \\ 0 & 0 \end{bmatrix}$$

satisfies properties (A2.2) and (A2.3).

We see, then, that these generalized inverses may be computed by performing a sequence of row operations on the matrix A and recording these operations in an identity matrix appended to A. This process is remarkably similar to the usual method of inverting a nonsingular matrix by appending an identity matrix and reducing the original matrix to an identity, as the following example illustrates.

Example (Ben-Israel and Charnes). Consider the problem

(A2.16) $$\max \ 2x_1 - x_2 - x_3 + 3x_4$$

subject to

$$\begin{bmatrix} 0 \\ -3 \\ 1 \end{bmatrix} \le \begin{bmatrix} 1 & 2 & -1 & 0 \\ -1 & 0 & 1 & -1 \\ 2 & 1 & -3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \le \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} .$$

The identification of the vectors and matrix above with our earlier notation is obvious. Using the last lemma, we diagonalize a portion of the matrix A by a sequence of pivot operations and record them in an identity matrix appended on the right. The pivot elements are circled.

(A2.17)
$$\left[\begin{array}{rrrr|rrr} 1 & 2 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 & 1 & 0 \\ 2 & 1 & -3 & 1 & 0 & 0 & 1 \end{array}\right]$$

(A2.18)
$$\left[\begin{array}{rrrr|rrr} 1 & 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & -1 & 1 & 1 & 0 \\ 0 & -3 & -1 & 1 & -2 & 0 & 1 \end{array}\right]$$

(A2.19)
$$\left[\begin{array}{rrrr|rrr} 1 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1/2 & 1/2 & 1/2 & 0 \\ 0 & 0 & -1 & -1/2 & -1/2 & 3/2 & 1 \end{array}\right]$$

(A2.20)
$$\left[\begin{array}{rrrr|rrr} 1 & 0 & 0 & 3/2 & 1/2 & -5/2 & -1 \\ 0 & 1 & 0 & -1/2 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1 & 1/2 & 1/2 & -3/2 & -1 \end{array}\right].$$

Write

(A2.21)
$$E = \begin{bmatrix} 1/2 & -5/2 & -1 \\ 1/2 & 1/2 & 0 \\ 1/2 & -3/2 & -1 \end{bmatrix};$$

then we have

(A2.22)
$$EA = \left[\begin{array}{c|c} & 3/2 \\ I & -1/2 \\ & 1/2 \end{array}\right]$$

(A2.23)
$$= [I|\Delta] \ .$$

Therefore by the lemma

(A2.24)
$$T = \left[\begin{array}{c} E \\ \hline 0 \end{array}\right]$$

(A2.25)
$$= \left[\begin{array}{ccc} 1/2 & -5/2 & -1 \\ 1/2 & 1/2 & 0 \\ 1/2 & -3/2 & -1 \\ \hline 0 & 0 & 0 \end{array}\right]$$

is an element of $A(1)$. Computing directly we have

(A2.26)
$$cT = [2 \ \ -1 \ \ -1 \ \ 3] \left[\begin{array}{ccc} 1/2 & -5/2 & -1 \\ 1/2 & 1/2 & 0 \\ 1/2 & -3/2 & -1 \\ \hline 0 & 0 & 0 \end{array}\right]$$

(A2.27)
$$= [0 \ \ -4 \ \ -1],$$

and hence

$$I^- = \{2, \ 3\},$$

$$I^+ = \phi,$$

$$I^0 = \{1\}.$$

Therefore we have

(A2.28)
$$z^* = \begin{bmatrix} \theta \\ -3 \\ 1 \end{bmatrix},$$

and

(A2.29)
$$x^* = Tz^*$$

(A2.30)
$$= \frac{1}{2}\begin{bmatrix} \theta + 14 \\ \theta - 3 \\ \theta + 8 \\ 0 \end{bmatrix}$$

is an optimal solution to (A2.16).

Finally, it should be noted that if A is nonsingular then $A^- = A^{-1}$ is a suitable generalized inverse of A and the foregoing treatment is entirely applicable.