

200099002

AN ALGORITHM FOR INTEGER
NEAR PROGRAMMING BY PARAMETRIC
MODIFICATION OF AN ADDED CONSTRAINT

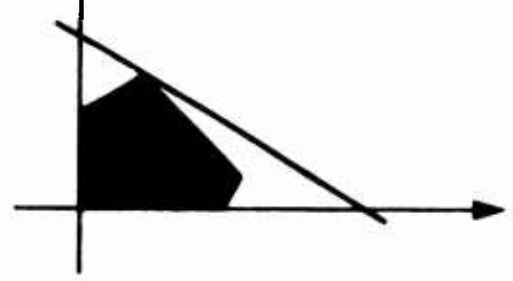
by
Hajime Eto

ORC 67-32
AUGUST 1967

DDC
RECORDED
OCT 24 1967
RECORDED

OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING



UNIVERSITY OF CALIFORNIA - BERKELEY

CLEARING HOUSE

RECEIVED
OCT 24 1967

AN ALGORITHM FOR INTEGER LINEAR PROGRAMMING
BY PARAMETRIC MODIFICATION OF AN ADDED CONSTRAINT

by

Hajime Eto
Operations Research Center
University of California, Berkeley

August 1967

ORC 67-32

This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83), the National Science Foundation under Grant GP-7417, and the U.S. Army Research Office-Durham, Contract DA-31-124-ARO-D-331 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ACKNOWLEDGEMENTS

I heartily thank Dr. Fred Glover, Miss Noreen Comotto and Miss Patricia Ward for their cordial suggestions in revising the paper.

ABSTRACT

To solve an integer linear program, we identify particular values that the objective function can assume on feasible lattice points. Thus, we reduce the problem of finding an *optimal* integer solution of n dimensions to that of finding a *feasible* integer solution of $n-1$ dimensions. A Branch and Bound Method is presented to solve the latter problem for the 0-1 case.

AN ALGORITHM FOR INTEGER LINEAR PROGRAMMING
BY PARAMETRIC MODIFICATION OF AN ADDED CONSTRAINT

by

Hajime Eto

1. INTRODUCTION

We define the standard linear programming problem to be that of finding a vector X to

$$(1.1) \quad \text{Maximize } z = CX$$

subject to the constraints

$$(1.2) \quad AX \leq B$$

$$(1.3) \quad X \geq 0$$

where A is $m \times n$, X is $n \times 1$, B is $m \times 1$ and C is $1 \times n$. The integer linear programming problem is a linear program in which X must also satisfy

$$(1.4) \quad X : \text{integral,}$$

i.e., each component of X must be an integer. We will call a vector X satisfying (1.1) and (1.2) a feasible continuous solution, and a feasible X that also maximizes $z = CX$ an optimal continuous solution. Feasible and optimal integer solutions are similarly defined by requiring that (1.3) be satisfied.

Rewriting the problem (1.1), (1.2) and (1.3) in the following manner:

$$(2.1) \quad \text{Maximize } z = CX$$

$$(2.2) \quad \text{Subject to } AX + A^1 Y = B$$

$$(2.3) \quad X, Y \geq 0$$

and applying the simplex method to this problem, we obtain a new representation:

$$\begin{aligned}
 &\text{Maximize} && z = C^0 U + K \\
 &\text{Subject to} && A^0 U + IW = B^0 \\
 &&& U, W \geq 0
 \end{aligned}$$

where U and W consist of the same variables as X and Y but in a different order. We call the components of Y the original basic variables and the components of W the current basic variables. (W and Y , or X and U may or may not be disjoint.) In an optimal representation, $C^0 \geq 0$ and $B^0 \geq 0$, and an optimal continuous solution is given by $U = 0$ and $W = B^0$.

General Ideas of an Algorithm for the Integer Programming Problem

Hereafter, we assume an optimal continuous solution has been obtained, yielding a value of $z = z_0$.

Let V denote the set of all the variables including the slack variables. Let us denote by G the set of all basic variables. We may exclude from G the degenerate variables, i.e., the variables which have zero components in B vector in the current tableau. And we include in G the dual degenerate variables, i.e., the variables whose shadow prices are zero in the current tableau.

Let F denote the set of all the variables with nonzero coefficients in the objective function. In many integer programming problems, most of the zero-one integer variables have zero coefficients in the objective (see Discussion 2 below) and F is a small subset of V . If a variable has an upper bound less than one in a range, it is dropped from F because it must take on value zero there.

Therefore, the intersection $F \cap G$ is a relatively small subset of V . Clearly, $F \cap G$ varies with the basis exchange because G does.

We can assume without loss of generality that all coefficients of the objective are integers and that their greatest common divisor is one. However, the greatest common divisor d of all coefficients of variables belonging to $F \cap G$ may be larger than one.

One of our basic ideas is to cut the feasible convex by a hyperplane parallel to the objective.

Experience indicates that an efficient version of Gomory's method of integer forms occurs by applying Gomory cuts to the value of the objective function first. (3) (1) (2). In the Gomory cutting method the value of the functional is limited to be integral. But from a well-known theorem (Lemma 1 below) in number theory, the value of the functional is confined to be multiples of d as defined above.

After a candidate for a permissible value of the functional is identified, we search for a lattice point at which that value is attained. In most integer programs, the upper bounds of unknowns are so low that each of them can take only a few values. Therefore, if we succeed in contracting the permissible range of values of the variables belonging to $F \cap G$ sufficiently, it may be practicable simply to enumerate the permissible combinations of integer values of these variables.

The other basic idea is to introduce the dual degenerate variables into the basis one at a time in order to determine a more restricted range of values of the variables.

2. THEORETICAL FOUNDATION OF THE ALGORITHM

Definition 1.

Let F be the set of the variables which may change the value of the objective function; i.e., $F = \{x_j \mid c_j \neq 0 \text{ and } u_j \neq 1\}$ where $u_j \geq x_j$.

Definition 2.

Let G_P be the set of all the variables whose shadow prices are zero at an extreme point P whether they may or may not be basic in an ordinary sense of the word "basic." If a degeneracy occurs at point P , all of the degenerate variables can be excluded from G_P ; i.e.,

$$G_P = \{x_j \mid \bar{c}_j = 0 \text{ and (the associated } \bar{b}_i \neq 0 \text{ if } x_j \text{ is basic)}\}$$

Definition 3.

Let d_P be the greatest common divisor (G.C.D.) of these c_j for which $x_j \in F \cap G_P$; i.e.,

$$d_P = \text{G.C.D. } (c_j \mid (c_j \neq 0) \text{ and } (\bar{c}_j = 0 \text{ at a point } P) \text{ and } (u_j \neq 1))$$

Lemma 1.

If c_j and x_j are integers, then $\sum_j c_j x_j = k d$ where k is an integer and d is the G.C.D. of c_j . The proof can be found in (4).

In what follows, some preparation is made for Theorem 1 associated with Figure 1.

Let Q_0 be a point inside a convex polyhedron. Let H be a hyperplane through Q_0 with the same slope as the objective function and let H intersect with the polyhedron at points $Q_1, Q_2, \dots, Q_N$. (Possibly

$Q_i = Q_j$ for $i \neq j$ if more than two constraints meet on the same point.)

Let us consider the smallest subpolyhedron containing $Q_0, Q_1, \dots, Q_N$.

Let $V_1, V_2, \dots, V_U$ denote extreme points of the subpolyhedron.

We can consider another parallel hyperplane $\bar{H}$ that intersects the polyhedron on the same edges as H does. Let $\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_N$ denote the intersecting points so that Q_i and $\bar{Q}_i$ are on the same edge for each i . Let us take a point $\bar{Q}_0$ on $\bar{H}$.

Theorem 1. (Theorem of a Possible Value of the Objective Function to a Feasible Lattice Point)

If there is any feasible lattice point L in the subpolyhedron $V_1 V_2 \dots V_U$, then the value of the objective function at L is $k d_1$, where k is an integer and d_1 is the G.C.D. at the point Q_1 (see Definition 3).

Proof:

Let Q_i and $\bar{Q}_i$ be on the edge $V_j V_k$ ($j \neq k$) and B_j, B_k be the basic variables at V_j, V_k respectively. Q_i and $\bar{Q}_i$ can be expressed uniquely as a linear combination of variables in the union of sets $(B_j \cup B_k)$. A similar linear combination which is unique holds for all i . Since Q_0 can be expressed uniquely as a linear combination of $(Q_1, Q_2, \dots, Q_N)$, Q_0 can be uniquely expressed as a linear combination of $(B_1 \cup B_2 \cup \dots \cup B_U)$.

If we cut the polyhedron by a hyperplane H parallel to the objective function, then all points on H are optimal feasible (not necessarily integer) points having the same value of the objective function.

Let us consider an extreme point of the cutting polyhedron on H , say Q_1 which is assumed to be on the edge $V_1 V_2$. By repeating the process of

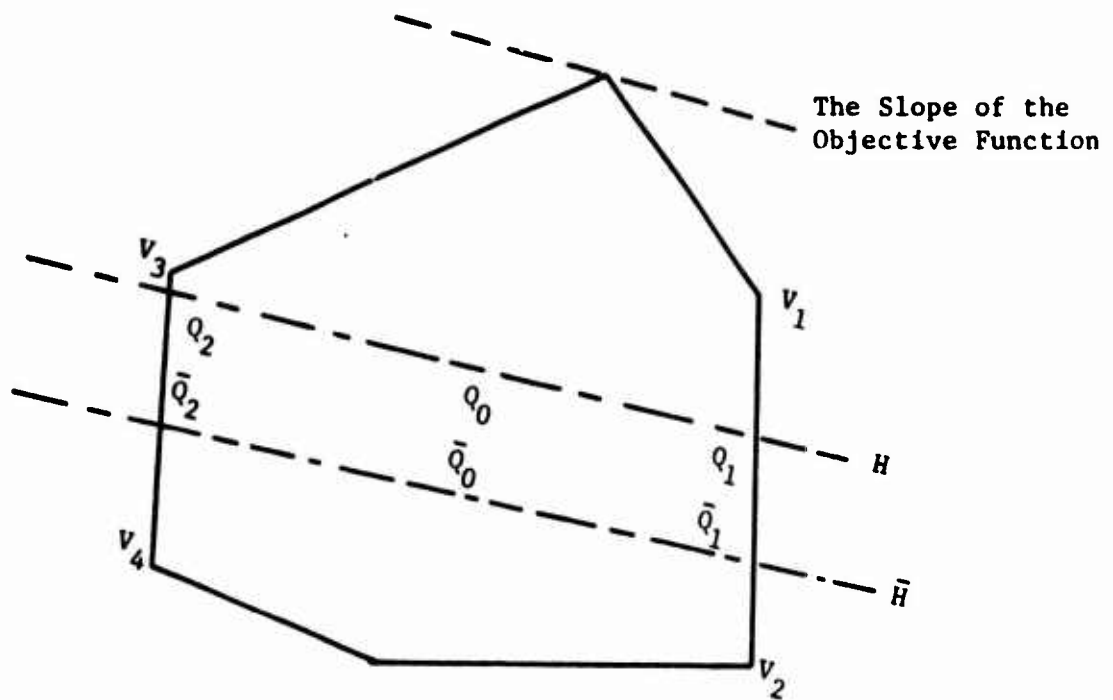


FIGURE 1: PARALLEL HYPERPLANES

appropriate basis exchange, we get all such $Q_2, Q_3, \dots, Q_N$ which have the same value of the objective function. This fact implies that the variable introduced into the basis at each time has zero shadow price; in other words, starting at Q_1 all the variables introduced into the basis belong to the set $G_1 = (B_1 \cup B_2)$. Furthermore, no other variables belong to this set, because otherwise H would have other extreme points having different values of the objective. Hence $G_1 = (B_1 \cup B_2) + S_1$ where S_1 is the set of all the variables whose shadow prices are zero at Q_1 . The same argument holds for $\bar{H}$ and $\bar{Q}_1$ and leads us to the conclusion that $\bar{G}_1 = (\bar{B}_1 \cup \bar{B}_2) + \bar{S}_1 = (B_1 \cup B_2) + S_1 = G_1$.

Similar results can be obtained for Q_i and $\bar{Q}_i$ for all i ; e.g., suppose $G_2 = (B_3 \cup B_4) + S_2 = \bar{G}_2 = (\bar{B}_3 \cup \bar{B}_4) + \bar{S}_2$ where Q_2 is on edge V_3V_4 . $(B_3 \cup B_4)$, the basis on Q_2 , is obtained from $(B_1 \cup B_2)$ uniquely by introducing a variable from the set S_1 into the basis and deleting a variable from the set $(B_1 \cup B_2)$ whose shadow price turns zero; in other words, a variable moves from S_1 to $(B_3 \cup B_4)$ and another variable moves from $(B_1 \cup B_2)$ to S_2 while the others remain unchanged. Thus $G_1 = G_2$. Furthermore, we see that $G_1 = \bar{G}_1 = G_2 = \bar{G}_2$ since $G_1 = \bar{G}_1$, and $G_2 = \bar{G}_2$.

The same argument leads us to the conclusion that $G_1 = G_2 = \dots = G_N = \bar{G}_1 = \bar{G}_2 = \dots = \bar{G}_N$. Since Q_0 and $\bar{Q}_0$ are expressed uniquely as linear combinations of $Q_1, \dots, Q_N$ and $\bar{Q}_1, \dots, \bar{Q}_N$, respectively, G_0 and $\bar{G}_0$ are the same.

Clearly, since the variable of the set F can change the value of the objective function, we can drop the variable which do not belong to set F from consideration. Thus, the theorem is proved with help of Lemma 1.

Q.E.D.

Corollary to Theorem 1. (Criterion of an Nonexistence of a Lattice Point)

If $b_i \not\equiv 0 \pmod{d^{(i)}}$ for some i , then no lattice point can exist in the subpolyhedron specified by G where $d^{(i)} = \text{G.C.D.}$

$(a_{ij} \mid x_j \in F_i \cap G)$ and $F_i = \{x_j \mid a_{ij} \neq 0 \text{ and } u_j \neq 1\}$ in (2.2).

Proof:

From Theorem 1, $\sum_{j \mid x_j \in F_i \cap G} a_{ij} x_j = k d^{(i)}$ for k integer, while $\sum_j a_{ij} x_j = \sum_{j \mid x_j \in F_i \cap G} a_{ij} x_j = b_i$ in the subpolyhedron specified by G . Hence $b_i = k d^{(i)}$ for k integer, thus $b_i \equiv 0 \pmod{d^{(i)}}$ for all i if any lattice point exists in the subpolyhedron.

Q.E.D.

Theorem 2. (Theorem of Enumerability)

If there is any lattice point in the subpolyhedron, then it is necessarily on a hyperplane $\bar{H}$ (or one of the hyperplanes if there is more than one such $\bar{H}$ in the same subpolyhedron) whose value of the objective function is $k d$ (for an integer k).

Proof: (See Figure 2.)

Let us assume on the contrary that a lattice point L is on a different and parallel hyperplane $\underline{H}$ whose objective function value is not $k d$ for any integer k . Here we assume without loss of generality that an intersecting point M of $\underline{H}$ with an edge is on the same edge as $\bar{Q}_1$. M is not necessarily a lattice point.

Since $\bar{H}$ has the same slope as the objective function, every point on $\underline{H}$ has the same value of the objective function. Thus the value of the objective function on M and the value of the objective function on L are equal to an integer value v . The basis which expresses $\bar{Q}_1$ also

expresses M . Thus we can see

$$(\bar{G}_1 \text{ at } \bar{Q}_1) = (G_M \text{ at } M) .$$

This implies v is equal to $k d$. Hence M is on one of the hyperplanes $\bar{H}$ having $k d$ as the value of the objective function, and so is L . This contradicts the assumption.

Q.E.D.

Theorem 3. (Theorem of the Criteria for Entering Another Subpolyhedron and Ending the Algorithm Unsuccessfully)

If $\bar{Q}_1$ is an infeasible point before a basis exchange and the algorithm moves to a feasible point $\bar{Q}_1'$ after an appropriate basis exchange, then $\bar{H}$ given by the added constraint is in another subpolyhedron. If feasibility can no longer be recovered by any basis exchange, then the added constraint is out of the feasible domain.

Proof follows from Figure 3. $\bar{H}$ is out of a subpolyhedron and $\bar{H}$ is out of the feasible domain. Theorem 3 justifies Step 3 and Step 9.

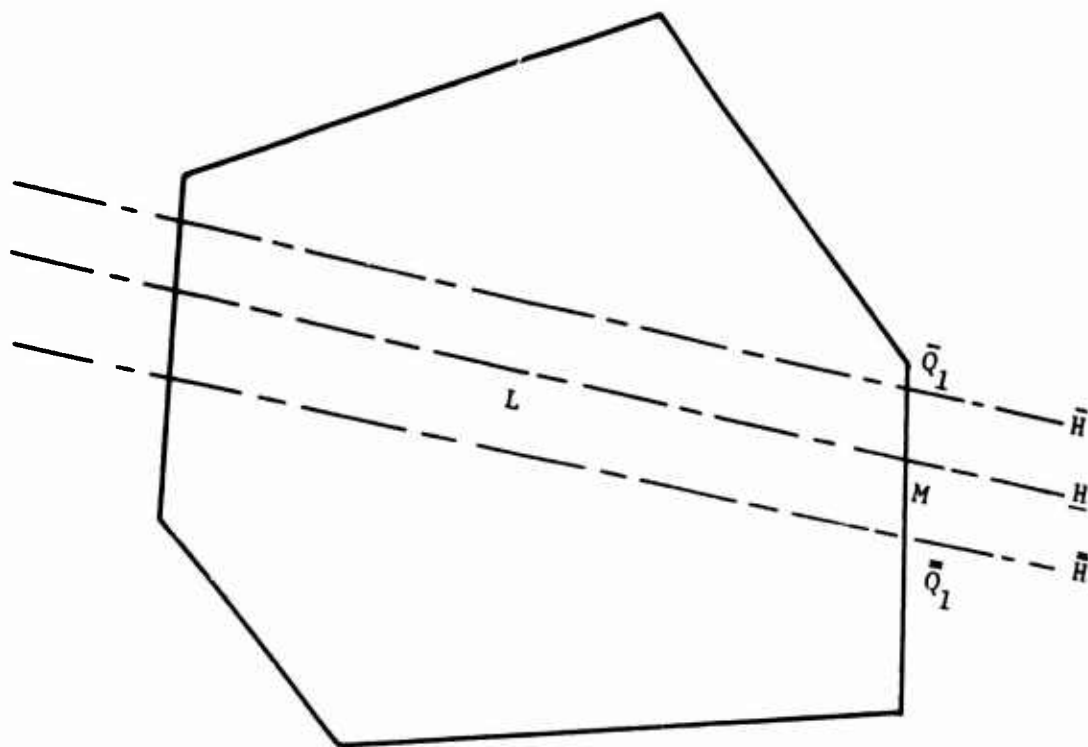


FIGURE 2: A LATTICE POINT ON A CUTTING HYPERPLANE

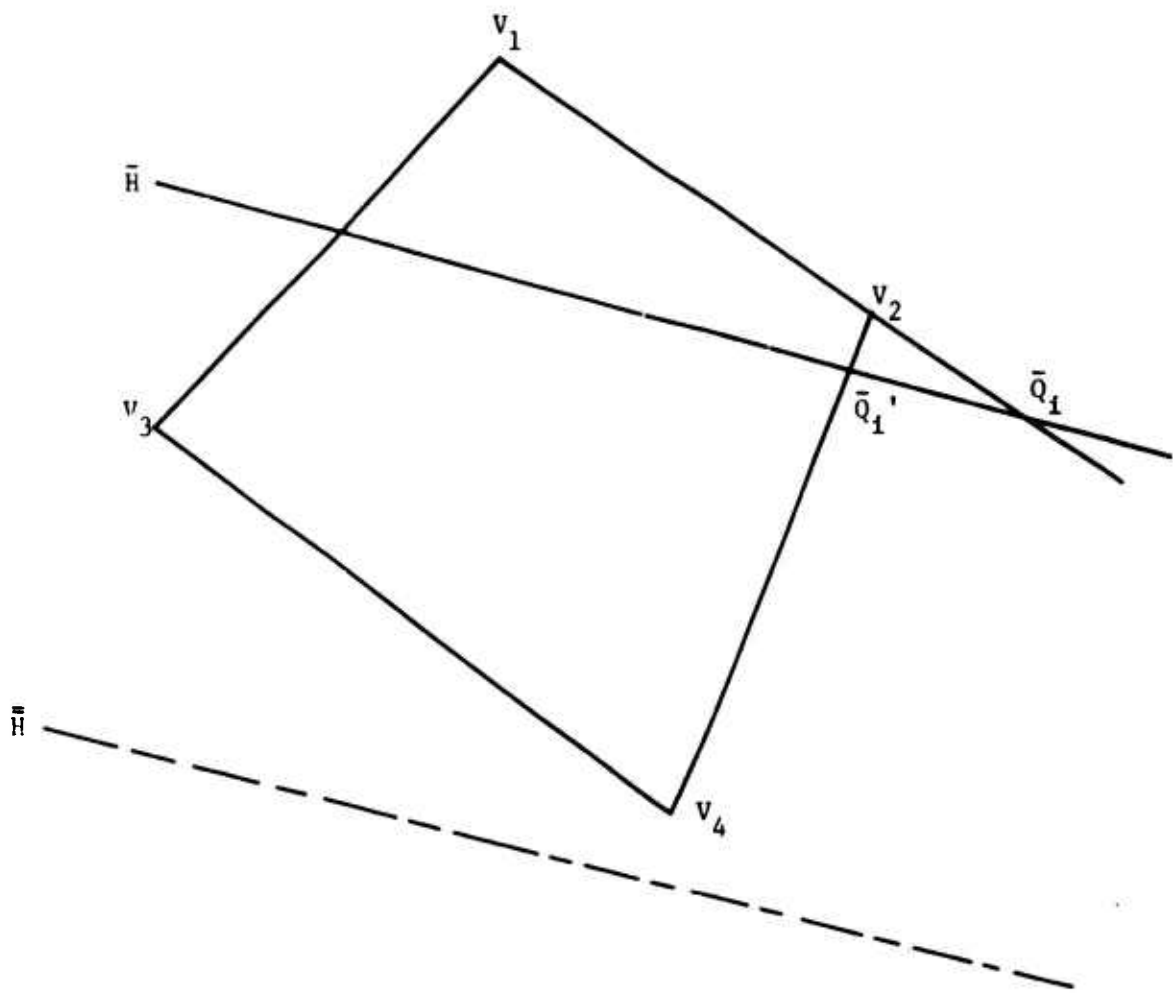


FIGURE 3: ENTRANCE INTO ANOTHER SUBPOLYHEDRON
AND UNSUCCESSFUL TERMINATION

3. DESCRIPTION OF THE ALGORITHM

Step 1.

Solve the standard Linear Programming Problem without integer requirement (1.4) and obtain z_0 , the optimal value of the objective.

Let P_0 stand for the optimal extreme point of the continuous problem.

Examine whether the solution is integral or not.

If integral, the problem is solved.

If otherwise, go to Step 2.

Step 2.

Define $d^{(1)} = \text{G.C.D. } (a_{ij} \mid x_j \in F_i \cap G) \text{ for all } i \text{ for a given } G \text{ where } F_i = \{x_j \mid a_{ij} \neq 0 \text{ and } u_j \neq 1\}.$

Examine whether $b_i \equiv 0 \pmod{d^{(1)}}$ for all i or not.

If not, then go to Step 9.

Otherwise, go to Step 3.

Step 3.

Add to (1.1), (1.2), (1.3) and (1.4) the following constraint

$r_0 :$

$$C X \leq k_0 d_0$$

where d_0 is the greatest common divisor of all the variables belonging to $F \cap G$.

Step 4.

Examine whether or not the solution subject to the augmented constraints in Step 3 is feasible without exchanging the basis.

If feasible, go to Step 5.

If otherwise, go to Step 9.

Step 5.

Obtain the intersecting points $Q_1^{(0)}, Q_2^{(0)}, \dots, Q_N^{(0)}$ of the added constraint r_0 with the edge of the convex hull by repeating the basis exchange. (One of the points, say $Q_j^{(0)} = (x_1^{(j)}, x_2^{(j)}, \dots, x_1^{(j)}, \dots, x_N^{(j)})$, is already obtained in Step 3.) In more detail, exchange an appropriate basic variable with one of the dual degenerate variables (i.e., a variable whose shadow price is zero in the current tableau) and get another intersecting point with the same value of the objective.

The number of dual degenerate variables is equal to $n-1$ (or less than $n-1$ if more than two constraints meet at the same point), where n is the number of the components of X excluding the slack variables.

If an infeasibility occurs during a basis exchange (i.e., if one of $Q_1^{(0)}, Q_2^{(0)}, \dots, Q_N^{(0)}$ is out of the feasible domain), go to Step 9.

If $\max_j x_i^{(j)} < 1$, then define F, F_1, d and $d^{(1)}$. If $b_1 \neq 0 \pmod{\text{new } d^{(1)}}$ for any i or the current value of objective is not $k'd$ for new d and any integer k' , then go to Step 8 after resetting old F, F_1, d and $d^{(1)}$ again. Otherwise, go to Step 6.

Step 6.

Obtain a lattice point (or lattice points) that can be expressed as a convex combination of the points $Q_1^{(0)}, Q_2^{(0)}, \dots, Q_N^{(0)}$.

This implies that the x_i belonging to $F \cap G$ must satisfy

$$\max \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(N)} \right) \geq x_i \geq \min \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(N)} \right) \text{ for all } i \in F \cap G \text{ where } Q_i^{(0)} = (x_1^{(j)}, x_2^{(j)}, \dots, x_{m+n-1}^{(j)}).$$

Step 7.

Examine whether a lattice point is obtained or not.

If obtained, the algorithm comes to an end.

If otherwise, go to Step 8.

Step 8.

Change the added constraint r_0 parametrically to r_0'

$$C X \leq (k_0 - 1)d_0$$

where C, X are the same as in Step 3 above. Solve the problem subject to the modified constraints. Go to Step 4.

Step 9. (From Step 4)

Identify the next $F \cap G$.

Go to Step 2.

The procedure below is similar to the above case if we substitute subscript 1 for subscript 0 and if we replace "add a constraint r_0 : $C X \leq k_0 d_0$ " with "modify the added constraint r_0 into r_1 such that $r_1 : C X \leq k_1 d_1$." If no extreme point can be found anymore, then the algorithm terminates unsuccessfully.

Remark to Step 2 and Step 3

A question arises how to know in advance which variables belong to $F \cap G$ after the addition of a constraint r_0 . They do not remain the same as on P_0 .

In order to identify the variables belonging to the current set of $F \cap G$, add for trial a constraint

$$C X \leq z_0'$$

where $z_0' = [z_0]$ if z_0 is not an integer, or $z_0' = z_0 - 1$ if z_0 is an integer. In other words, z_0' is the maximal integer less than z_0 .

We are in position to identify the variables in question by solving the problem under the added trial constraint. The trial constraint may happily coincide with the true constraint r_0 . This procedure is justified by Theorem 1 if the hyperplane given by z_0' and that given by $k_0 d_0$ are in the same polyhedron.

Remark to Step 9

A more detailed procedure to identify a new $F \cap G$ after $C X \leq (k_0 - 1)d_0$ becomes infeasible is as follows:

If $d_0 \geq 2$, then modify the added constraint $C X \leq k_0 d_0 - 1$ for trial and examine whether it is feasible without a basis exchange.

If it is feasible, then repeat the same procedure by reducing the right-hand side of the added constraint by 1 until it becomes infeasible. If it is infeasible, then a new $G \cap F$ can be identified in the same manner as in Remark to Step 2 and Step 3 after basis exchanges. The trial constraint may happily coincide with the true constraint r_1 .

Thus if the next subpolyhedron is small, we may "skip" it without jumping over any lattice points (Figure 4). This occurs because a subpolyhedron cannot contain a lattice point (from Theorem 1) if it does not

have any hyperplane which has an integer value of the objective. In Figure 4, a subpolyhedron $V_2V_3V_4V_5$ is skipped and a subpolyhedron $V_3V_5V_6V_7$ is examined.

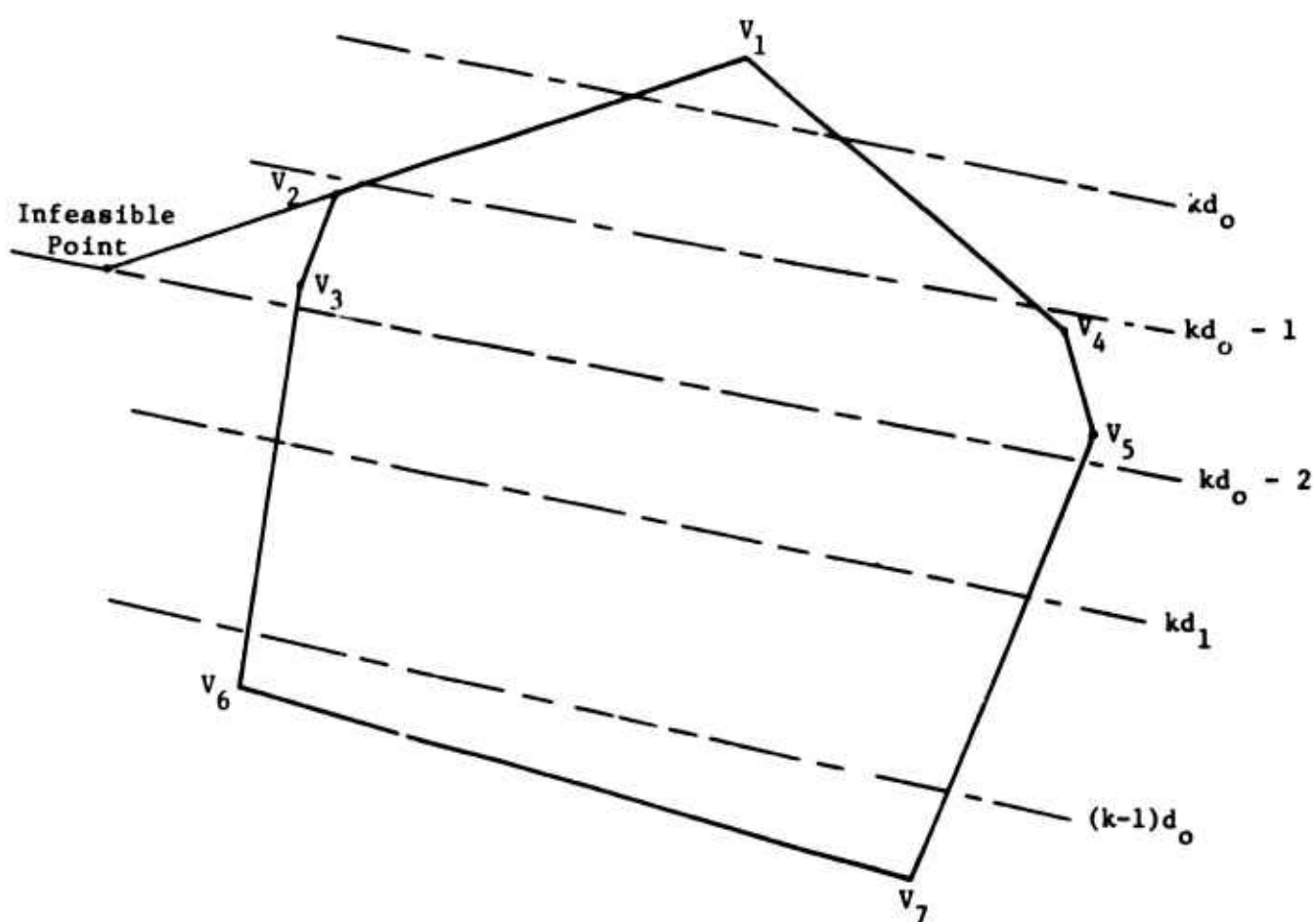


FIGURE 4: A SKIP OF A SUBPOLYHEDRON

4. FINITENESS PROOF

Let each hyperplane be specified by indices K, i where K denotes each subpolyhedron and i denotes each hyperplane in the same subpolyhedron. ($i = 0$ for K such that the K^{th} polyhedron does not contain a hyperplane given by a multiple of d_K .)

Theorem 4. (Theorem of Finiteness)

The algorithm terminates in a finite number of steps.

Proof:

The number of i (i.e., the number of times the constraint is modified) for each K is clearly finite since $d_K \geq 1$, and K is also finite because the number of the extreme points is finite unless the linear problem itself is unbounded. For each i, K , the number of basis exchanges is finite because a hyperplane is $n-1$ dimensional. After identifying the range of the values which each variable may assume, the enumeration of all permissible combinations of the integral values is finite.

5. A GRAPHIC EXPLANATION (FIGURE 5.)

Step 1.

We obtain P_0 , the optimal point with $z = z_0$. Examine whether P_0 is a lattice point or not. It is not a lattice point in this case, so we pass to Step 2.

Step 2.

Suppose $b_1 \equiv 0 \pmod{d^{(1)}}$ for all i .

Step 3.

Cut the convex by line (1)

$$C X \leq k_0 d_0$$

where d_0 is the greatest common divisor of all the variables belonging to $F \cap G$, and k_0 is the maximal integer such that $k_0 \cdot d_0 \leq z_0$. Suppose we obtain an optimal point $Q_1^{(0)}$.

Step 4.

In this case, it is feasible. So pass to Step 5.

Step 5.

By exchanging the basis, we obtain the other optimal extreme point $Q_2^{(0)}$.

Step 6.

We search for a lattice point on the segment $Q_1^{(0)} Q_2^{(0)}$ without success.

Step 7.

We pass to Step 8 because we have failed in obtaining a lattice

point.

Step 8.

Replace line (1) by line (2)

$$C X \leq (k_0 - 1)d_0$$

Solve the modified problem subject to (2) without the basis exchange.

Step 4.

It becomes infeasible without the basis exchange because we obtain as a solution the intersecting point of line (2) with line (5).

Step 9.

We identify a new $F \cap G$ below P_1 by replacing line (2) by line (3): $C X \leq k_0 d_0 - 3$ to obtain a new G after $C X \leq k_0 d_0 - 1$ and $C X \leq k_0 d_0 - 2$ are found to be still feasible.

Step 2.

Suppose $b_i \equiv 0 \pmod{d^{(1)}}$ for all i again.

Step 3.

Modify r_0 into r_1 such that

$$C X \leq k_1 d_1$$

where d_1 is the greatest common divisor of all the variables belonging to the current set of $F \cap G$, and k_1 is the

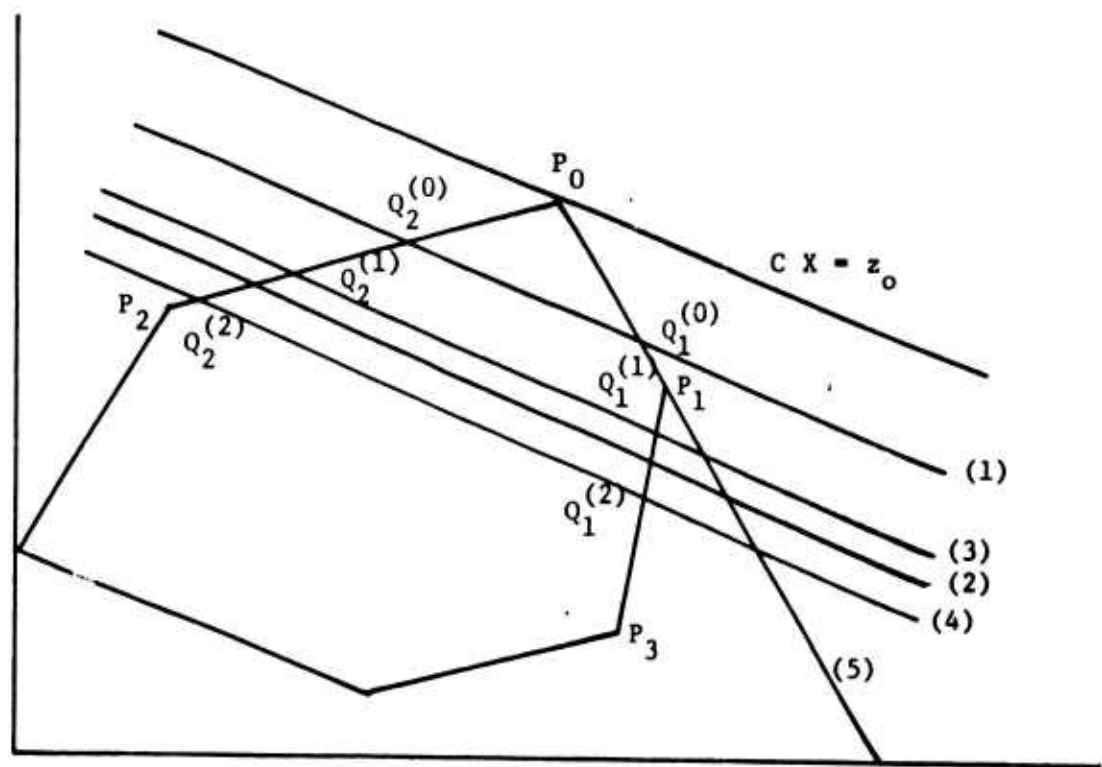


FIGURE 5: CUTTING THE CONVEX

maximal integer such that $CX \leq k_1 d_1$ is in the subpolyhedron $P_0 P_1 P_2 P_3$. We obtain $Q_1^{(1)}$ as a solution to the modified constraint.

Step 4.

It remains feasible without exchanging the basis.

Step 5.

We obtain the other intersecting point $Q_2^{(1)}$ besides $Q_1^{(1)}$ which was obtained in Step 3.

Step 6.

We search for a lattice point on the segment $Q_1^{(1)} Q_2^{(1)}$.

Step 7.

A lattice point is not obtained.

Step 8.

We change the added constraint r_1 to r_1' (line (4))

$$CX \leq (k_1 - 1)d_1,$$

and we obtain $Q_1^{(2)}$ as a solution without exchanging the basis.

Step 4.

The solution remains feasible.

Step 5.

We obtain the other intersecting point $Q_2^{(2)}$ by introducing the dual degenerate variable into the basis.

Step 6.

We search for a lattice point on the segment $Q_1^{(2)}Q_2^{(2)}$.

Step 7.

We succeeded in obtaining a lattice point on the segment,
successfully terminating the algorithm.

6. NUMERICAL EXAMPLE

$$\begin{aligned}
 \text{Maximize } z &= 7x_1 + 14x_2 + 10x_3 \\
 9x_1 + 5x_2 + 3x_3 &\leq 34 \\
 4x_1 + 5x_2 + 6x_3 &\leq 17 \\
 4x_1 + 10x_2 + 7x_3 &\leq 30
 \end{aligned}$$

Step 1.

We obtain as the optimal feasible solution

$$x_1 = 1.0, \quad x_2 = 2.6, \quad x_3 = 0.$$

and

$$z = 43.4$$

The solution above is not integral.

Step 2.

Add a constraint for trial

$$7x_1 + 14x_2 + 10x_3 \leq 43$$

partly to identify the variables belonging to $F \cap G$.

Step 4.

The solution is feasible.

Step 9 and Step 2.

After the basis exchange, we are in position to identify

x_1 , x_2 and all three slacks as the variables belonging to $G \cap F$.

$$d^{(1)} = \text{G.C.D. } (9, 5, 1) = 1, \quad d^{(2)} = \text{G.C.D. } (4, 5, 1) = 1,$$

$$34 \equiv 0 \pmod{1}$$

$$21 \equiv 0 \pmod{1}$$

$$d^{(3)} = \text{G.C.D. } (4, 10, 1) = 1$$

$$30 \equiv 0 \pmod{1}$$

Thus, a lattice point can exist because the slacks in G make

$d^{(i)} = 1$ for all i in this case.

Step 3.

The greatest common divisor of the coefficients of x_1 and x_2 is 7 and the maximal multiple of 7 not exceeding 43 is 42. So we add a constraint

$$7x_1 + 14x_2 + 10x_3 \leq 42,$$

and we obtain a solution

$$x_1 = \frac{4}{3}, \quad x_2 = \frac{7}{3}$$

and

$$x_3 = 0.$$

Step 4.

The solution above is feasible.

Step 5.

By introducing one of the dual degenerate variables into the basis, we obtain

$$x_1 = 0, \quad x_2 = 3$$

and

$$x_3 = 0$$

We successfully obtained an optimal integer solution.

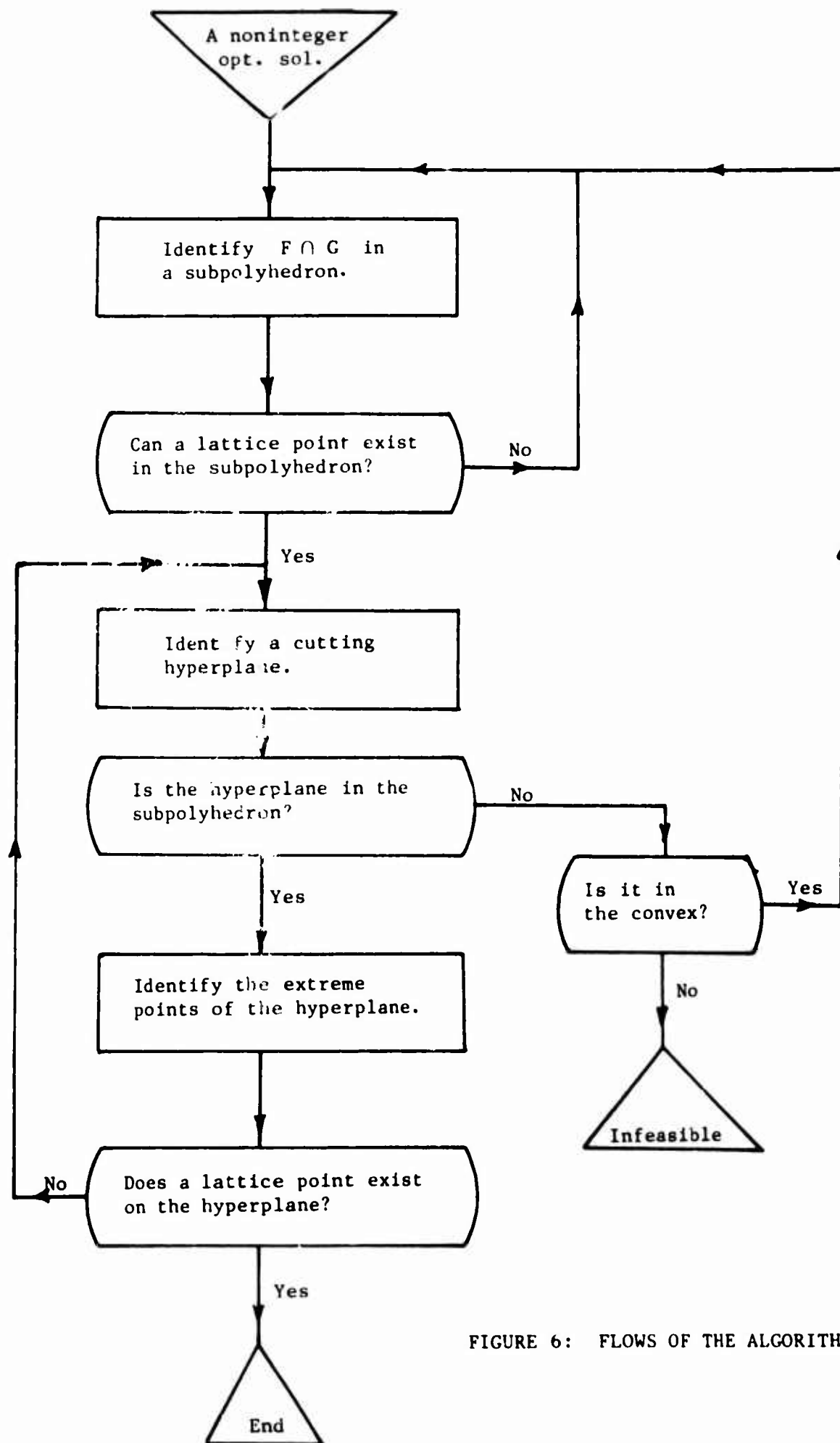


FIGURE 6: FLOWS OF THE ALGORITHM

7. A METHOD FOR FINDING LATTICE POINTS

A problem of searching for lattice points in Step 6 is left unsolved. One method is to repeat the same procedure, i.e., to reduce the problem again to a simpler problem to find a feasible solution in $n-2$ dimensions. At this time $d=1$ is uniformly used for a dummy objective function $z = \sum_j x_j$ in all subpolyhedrons; we do not have to identify $G \cap F$ anymore because the value of the objective is already fixed.

Another method is to use a branch and bound algorithm which will be presented below. Emphasis is placed on a 0-1 problem.

As stated in Theorem 2, all extreme points $\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_N$ of a hyperplane $\bar{H}$ can be obtained by repeating the introduction of variables into the basis whose shadow prices are zero. Let $(x_1^{(j)}, x_2^{(j)}, \dots, x_i^{(j)}, \dots, x_{m+n}^{(j)})$ denote the coordinate of $\bar{Q}_j$ ($j = 1, \dots, N$).

If there is any lattice point on $\bar{H}$ whose coordinate is $(x_1, x_2, \dots, x_i, \dots, x_{m+n})$ where x_i should be an integer for all i , then

$$l_i = \min_j (x_i^{(j)}) \leq x_i \leq \max_j (x_i^{(j)}) = u_i \quad \forall i$$

If $[l_i] = [u_i] < l_i$, then there cannot be any integer x_i between u_i and l_i .

Rule 1. (Criterion of Absence of a Lattice Point)

If $[u_i] < l_i$ hold for some i , then $\bar{H}$ is disregarded.

Rule 2. (Criterion of Zero Value of a Variable)

If $u_i < 1$, then $x_i = 0$.

Hereafter, we consider only 0-1 case.

Rule 3. (Criterion of Value One of a Variable)

If $\ell_1 > 0$, then $x_j = 1$. Rules 2 and 3 fix some variables on $\bar{H}$.

On a particular hyperplane $\bar{H}$, the value of the objective function is already fixed. Hence, we no longer care about optimizing it. The order in which the variables are assigned values is unimportant, contrary to the ordinary branch and bound algorithm in which we have to take into account the variables in a predetermined order. In our case, the only criterion that must be satisfied is $\sum_{j \in \text{FNG}} c_j x_j = k d$. Hence we will take the assignment of the variable's value in the simplest order, namely in the ascending order of the indices of the variables.

Hereafter, we assume $c_j \geq 0$ for all j . (If not, we can obtain it by resetting $x_j = 1 - x_j$.) For the sake of simplicity let us assume the first f variables are already fixed.

An Algorithm of Enumerations by Branch Method

- (1) If $\sum_{j=1}^{n'} c_j x_j = k d$ for $n' \leq n$, then we obtain an integer solution by assigning value zero to all other unfixed variables.
 - (1-1) If the solution is feasible, we have obtained an integer feasible solution.
 - (1-2) If the solution is not feasible, then disregard this branch and search another branch.
- (2) If $\sum_{j=1}^{n'} c_j x_j < k d$ for $n' = n$, then go back to the branch and search another branch for a larger value of $\sum_j c_j x_j$.
- (3) If $\sum_{j=1}^{n'} c_j x_j > k d$ for $n' < n$, then go back to the branch and search another branch for a smaller value of $\sum_j c_j x_j$.

A Numerical Example of Enumerations

Let the objective function be $9x_1 + 16x_2 + 4x_3 + 12x_4$. Let us assume x_1 never appears as a basic variable at this stage. Hence $d = \text{G.C.D.}(16, 4, 12) = 4$. Suppose $k=5$ is appropriate at this stage, therefore $z = 5 \times 4 = 20$. Let $\bar{Q}_1 = (0, 0.25, 1, 1)$, $\bar{Q}_2 = (0, 0.875, 0, 0.5)$, $\bar{Q}_3 = (0, 1, 1, 0) = \bar{Q}_4$. Thus, $0 \leq x_1 \leq 0$, $0.25 \leq x_2 \leq 1$, $0 \leq x_3 \leq 1$, $0 \leq x_4 \leq 1$. By Rule 2, $x_1 = 0$ and by Rule 3 $x_2 = 1$. The only unfixed variables are x_3 and x_4 . In Figure 7, a figure on the left side of a slash denotes a value taken on by a variable in the right column and a figure on the right side of the slash denotes a value of the objective function taken on a solution associated with an arc.

We start with Arc 1 to which a value 16 is associated as the objective function. 16 is smaller than 20, therefore we proceed until we reach or exceed 20. We consider Arc 2 where we reach an end of a branch, but its value 16 is too small, so we search another branch for a bigger value. (Procedure 2.) We consider Arc 3 but its value 28 is too large, therefore we go back. (Procedure 3.) We consider Arc 4 where its value is exactly 20. Thus, we assign x_4 a zero value. Hence, automatically we move to Arc 5 by disregarding Arc 6. (Procedure 1.) We examine whether Arc 5 is feasible or not. Arc 5 happens to coincide with $\bar{Q}_3$, therefore it is feasible. (Procedure 1-2.)

Another Numerical Example

Let us consider the same objective function and a different $k=6$. Hence, the criterion is $9x_1 + 16x_2 + 4x_3 + 12x_4 = 24$. Let $\bar{Q}_1 = (0, 1, 0, 2/3)$, $\bar{Q}_2 = (0, 1, 1, 1/3)$ and $\bar{Q}_3 = (0, 1, 1/2, 5/12) = \bar{Q}_4$. $u_4 = 2/3 [u_4] = 0 < \ell_4 = 1/3$. Hence, there cannot be any integer feasible solution. Thus, we must disregard this hyperplane. (Rule 1.)

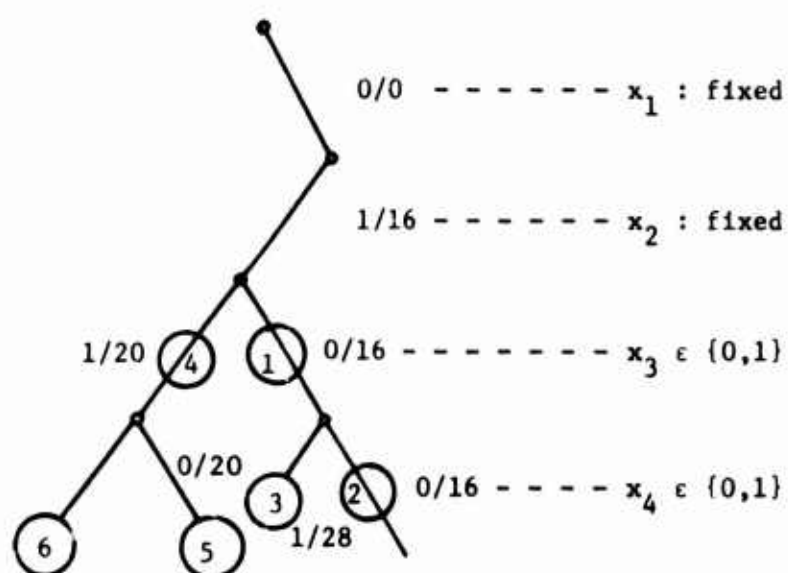


FIGURE 7: A TREE

The algorithm presented above is a branching algorithm; but it is a bounding algorithm in a somewhat different sense than usually conveyed by the term "branch and bound." For each $\bar{H}$ a value of the objective function is already fixed, so we do not need to evaluate it anymore. Instead, we give an upper and a lower bound to each variable.

A criterion function given here is the criterion to see whether an arc is feasible or not, as opposed to the ordinary branch and bound algorithm where the criterion is: which variable should be considered next.

No evaluation after setting a hyperplane $\bar{H}$ is made in the algorithm presented above.

8. DISCUSSIONS

Another Starting Procedure

We have started with an optimal continuous solution. But we can also start with a feasible integer solution if it is already given. In the latter case, we always have a feasible integer solution and we proceed to a better integer solution until an optimal continuous solution is reached or exceeded. The termination criterion is the same as in the former case.

The Possible Application of the Algorithm

Let us assume that the matrix A (in $AX = B$) is $m \times (n+m)$ (with m slack variables). Without degeneracy, m basic variables are associated at each $\bar{Q}_1$, and one extreme point is already identified on $\bar{H}$. Hence, $(n-1)$ nonbasic variables assume zero shadow prices at $\bar{Q}_1$. Thus, $(m + (n-1))$ variables belong to $\bar{G}_1$ at V_1 . This leaves out one variable. In particular if a slack variable is this exception, then $F \cap G_1 = F$. But consider the problem of m constraints $D_i(x_1, \dots, x_n) \geq 0$ ($i = 1, \dots, m$), of which at least m' constraints are required to be satisfied with integral x_j for all j . The problem is represented as follows:

$$\begin{aligned} D_i - \ell_i \lambda_i &\geq 0 & (i = 1, \dots, m) \\ \lambda_i &= m - m' \\ \lambda_i &\in \{0, 1\} \end{aligned}$$

where $\ell_i \leq D_i$ for each i . In this problem in addition to $(m+n)$ variables, there are m λ -variables whose coefficients are all zero in the objective function. The cardinal number of $G_1 = m+n-1 \geq$ the cardinal number of $F \cap G_1$

$$\geq \max(0, n-m-1)$$

while the total number of the variables equal $2m + n$. Thus we can see if n is not too large for m , the set $F \cap G_i$ is a relatively small subset of the set of all the variables.

When G consists mostly of slack variables and λ_i 's, d may be greater than one. Thus, a cutting hyperplane method may work efficiently. On the other hand, many $d^{(i)}$'s are equal to one because coefficients of slack variables are one in constraints.

When G consists mainly of nonslack variables, then the $d^{(i)}$'s may be greater than one and Step 2 may work efficiently in excluding from consideration subpolyhedrons which cannot contain a lattice point. On the other hand, the cutting hyperplane method may not work so well in this case because d may be close to one.

REFERENCES

- [1] Gomory, R. E., "Outline of an Algorithm for Integer Solutions to Linear Programs," Bull.Amer.Math.Soc., Vol. 64, (1958).
- [2] Gomory, R. E., "An Algorithm for the Mixed Integer Problem," RAND, Rm-2597, (1960).
- [3] Haldi, J. and L. M. Isaacson, "A Computer Code for Integer Solutions to Linear Programs," Opns.Res., Vol. 13, No. 6, pp. 946-959, (1965).
- [4] Hasse, H., VORLESUNGEN ÜBER ZAHLENTHEORIE, Berlin, (1949).

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1 ORIGINATING ACTIVITY (Corporate author)		2a REPORT SECURITY CLASSIFICATION
University of California, Berkeley		Unclassified
		2b GROUP
3 REPORT TITLE		
AN ALGORITHM FOR INTEGER LINEAR PROGRAMMING BY PARAMETRIC MODIFICATION OF AN ADDED CONSTRAINT		
4 DESCRIPTIVE NOTES (Type of report and inclusive dates)		
Research Report		
5 AUTHOR(S) (Last name, first name, initial)		
ETO, Hajime		
6 REPORT DATE	7a TOTAL NO OF PAGES	7b NO OF REFS
August 1967	35	4
8a. CONTRACT OR GRANT NO.	9a ORIGINATOR'S REPORT NUMBER(S)	
Nonr-222(83)	ORC 67-34	
b. PROJECT NO.		
NR 047 033		
c.	9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
Research Project No.: RR 003 07 01		
d.		
10. AVAILABILITY/LIMITATION NOTICES		
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.		
11. SUPPLEMENTARY NOTES Also supported by the Nat'l. Sci. Found. under Grant GP-7417, and the U.S. Army Res. Office-Durham under Contract DA-31-124-ARO-D-331.		12 SPONSORING MILITARY ACTIVITY
		MATHEMATICAL SCIENCE DIVISION
13 ABSTRACT		
To solve an integer linear program, we identify particular values that the objective function can assume on feasible lattice points. Thus, we reduce the problem of finding an <i>optimal</i> integer solution of n dimensions to that of finding a <i>feasible</i> integer solution of $n-1$ dimensions. A Branch and Bound Method is presented to solve the latter problem for the 0-1 case.		

DD FORM 1473
1 JAN 64

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Integer Linear Programming The Greatest Common Divisor Parallel Cutting Hyperplane						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those

imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.