COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut 06520

COWLES FOUNDATION DISCUSSION PAPER NO. 526

PRODUCTION SETS WITH INDIVISIBILITIES: GENERALITIES AND THE CASE OF TWO VARIABLES

Herbert E. Scarf

May 21, 1979

PRODUCTION SETS WITH INDIVISIBILITIES: GENERALITIES AND THE CASE OF TWO VARIABLES

by

Herbert E. Scarf

The research described in this paper was supported by a grant from the National Science Foundation. Some of the material was presented in the Fisher-Schultz lecture delivered at the 1978 European Meeting of the Econometric Society.

I am extremely grateful to Professors Sergiu Hart of Stanford University and Roger Howe of Yale University for the many interesting conversations I have had with them on the general topic of this paper.

TABLE OF CONTENTS

I. Introduction

The assumption of convex production sets plays an extremely important role in general equilibrium analysis. Its replacement by weaker and more plausible assumptions seems to me to be one of the major challenges of mathematical economics. One way of approaching this problem is by means of n person game theory, since this conceptual apparatus is considerably broader than the neo-classical model of equilibrium. There is, of course, no difficulty in constructing a variety of special models in which production exhibits increasing returns to scale, and for which the associated n person game has a non-empty core. But it seems to be impossible to raise these special examples to the level of generality that is customary in economic analysis. For this reason I am extremely doubtful that game theory will provide an illuminating analysis of non-convexities in production.

It is my belief that it is fruitful to study, for their own intrinsic interest, production sets in which the customary convexity assumption is relaxed. Of particular importance are those production sets in which a discrete collection of activities is available.

The mode of analysis adopted in the present paper will make use, in a rather surprising way, of techniques which were developed in the solution of an apparently unrelated problem--the approximation of fixed points of a continuous mapping. These techniques will be combined with the study of lattice points and convex polyhedra in what might be called the Geometry of Numbers without the customary symmetry assumption.

We shall associate with each discrete production set a canonical simplicial complex which may, if the set is finite, be shown to be homeomorphic to the unit simplex. A unique minimal collection of neighborhoods for which a local maximum is global, will then be defined for this production set. This will permit us to describe, in theory, a monotone algorithm for the solution of the general discrete programming problem. Prices and considerations of profitability, tied as they are to the neo-classical formulation, will play no role in the development of this algorithm.

The problem that arises in practice is that the system of neighborhoods may be extremely complex and difficult to determine. We may be content with heuristic neighborhoods which promise approximate solutions rather than global optima. If, however, the correct neighborhood systems are to be determined, some additional assumptions--such as the additivity assumption which leads to integer programming problems--are required.

But even if we restrict our attention to integer programs we seem to be faced with sophisticated and intriqueing mathematical problems which are far from a general resolution. We shall illustrate these problems in our description of a polynomial-time algorithm for integer programs with two variables. The application to integer programs with three variables is being examined by Sergiu Hart, Roger Howe and myself and will be reported on in a subsequent publication.

Let us consider a discrete production set X consisting of a set of vectors $\{x\}$ in R^{m+1}. Each specific vector in X represents a technically feasible production plan with inputs denoted by negative entries and outputs by positive entries. In subsequent sections of this paper we shall assume that the vectors in X arise from an activity analysis model with integral activity levels:

1.1 $$x = \begin{bmatrix} a_{01} & \cdots & a_{0n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} h ,$$

where $h = (h_1, \ldots, h_n)$ ranges over all integral points in R^n. For the moment, however, we take X to be completely general, aside from the following two assumptions:

1.2 [Assumption of Local Finiteness]. We assume that for any vector $c \in R^{m+1}$ the set of $x \in X$ satisfying $x \geq c$ is finite.

1.3. [Assumption of Non-Degeneracy]. No two vectors in X have the same i^{th} coordinate, for any i.

Figure 1 represents an example of a production set consisting of a finite list of vectors $x^0, x^1, \ldots, x^6$ in R^3. I have drawn through each vector the translate of the non-positive orthant having its vertex at that particular vector. This provides us with an intuitive picture

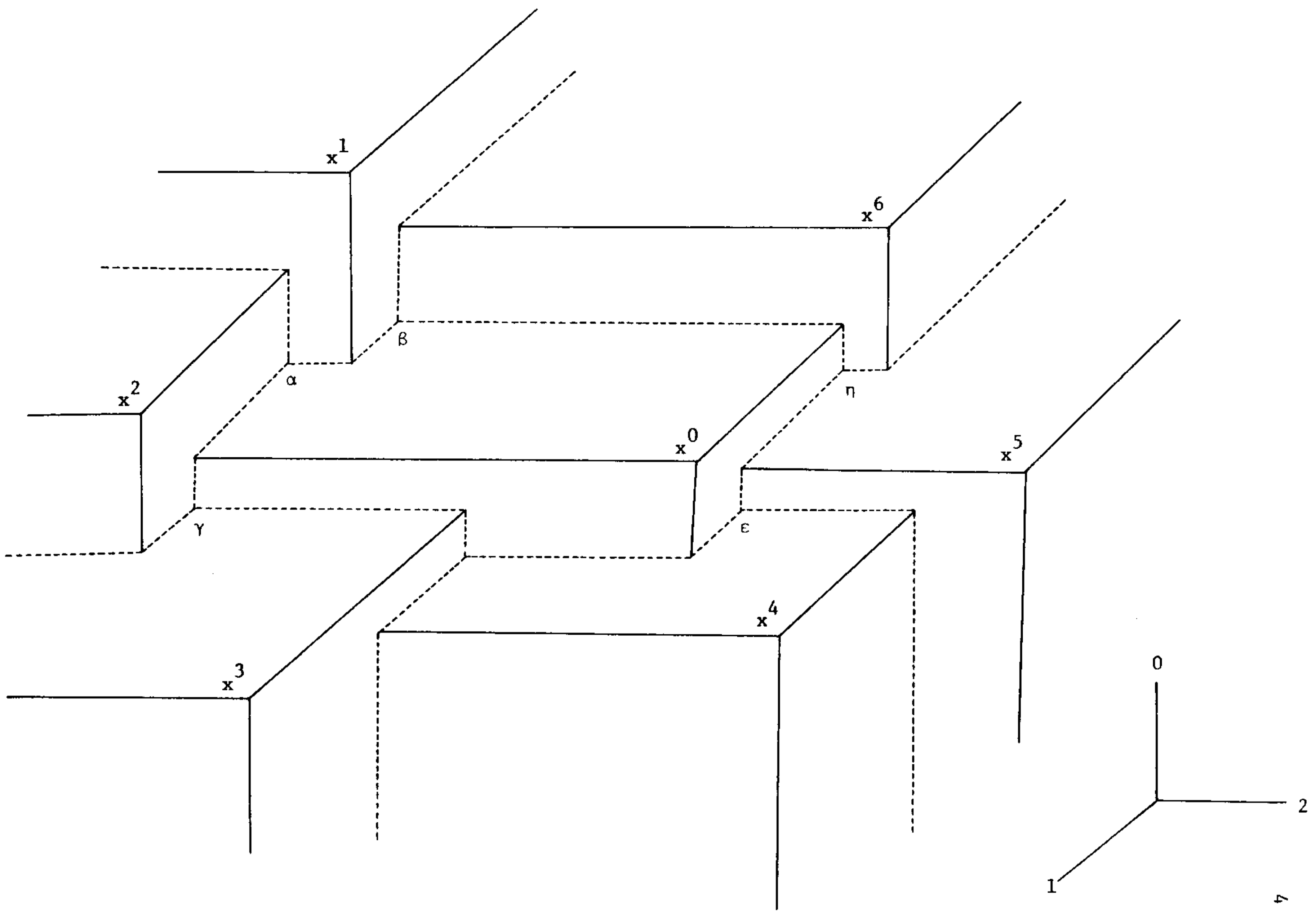


FIGURE 1

of what one might mean by the "upper surface" of a discrete production set and also reveals a surprising amount of structure, which will form the basis for much of our subsequent analysis.

We shall define, in a canonical fashion, a collection of m-simplices whose vertices are selected from the vectors in X. We begin the construction by translating the positive orthant of R^{m+1} parallel to itself until it contains none of the vectors in X. Then translate the orthant downwards, passing through none of the vectors in X, until no further reduction of any of the coordinates of its vertex is possible. The orthant will typically be stopped by a set of $m+1$ vectors in X, say $x^{j_0}, \ldots, x^{j_m}$. From the non-degeneracy assumption each coordinate hyperplane of the translated orthant will contain precisely one of these vectors. Moreover the vertex of the orthant will have its coordinate given by

$$\min[x^{j_0}, \ldots, x^{j_m}] ;$$

the coordinate-wise minimum of the $m+1$ vectors. These sets of $m+1$ vectors, which have elsewhere been given the name of primitive sets [Scarf, Hansen, 1973], will be the m-simplices of our collection. This definition may easily be seen to be equivalent to the following:

1.4 [Definition]. A set of $m+1$ vectors in X, $x^{j_0}, \ldots, x^{j_m}$ is said to be a primitive set if there is no vector x in X with

$$x > \min[x^{j_0}, \ldots, x^{j_m}] .$$

In Figure 1 the vectors x^0, x^1, x^2 form a primitive set with vertex α, and x^0, x^1, x^6 a primitive set with vertex β. In order to be somewhat more concrete let us imagine that the vectors in Figure 1 form a finite subset of a doubly infinite set of vectors given by

$$x = \begin{bmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} h ,$$

as h runs over all lattice points in the plane. The primitive set x^0, x^1, x^2 will then be generated by three lattice points, say h^0, h^1, h^2. Since $x_0^0 = \min[x_0^0, x_0^1, x_0^2]$ we see that $a_{01}h_1^0 + a_{02}h_2^0 < a_{01}h_1^j + a_{02}h_2^j$ (for $j = 1, 2$). Therefore the straight line $a_{01}h_1 + a_{02}h_2 = \text{const.}$ passing through h^0 has both points h^1 and h^2 on its increasing side. Similarly the line $a_{11}h_1 + a_{12}h_2 = \text{const.}$ passing through h^1 has h^0 and h^2 on its increasing side, and the line $a_{21}h_1 + a_{22}h_2 = \text{const.}$ passing through h^2 has h^0 and h^1 on its increasing side. The fact

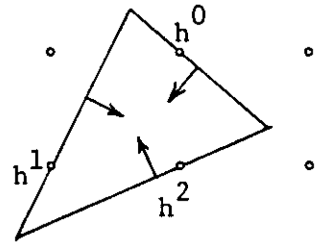


FIGURE 2

that x^0, x^1, x^2 form a primitive set may then be translated into activity level space by the statement that this triangle, whose three sides correspond to the three rows of A, contain no other lattice points. This is stronger than merely requiring that there be no other lattice points in the convex hull of h^0, h^1, h^2.

If the activity analysis model generating the set X is given by a matrix

$$\begin{bmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

with four, rather than three, rows, the vectors x will lie in R^4 and primitive sets will consist of sets of four vectors. The following figure illustrates a set of four lattice points in R^2 whose associated x vectors form a primitive set. The figure describes a quadrilateral

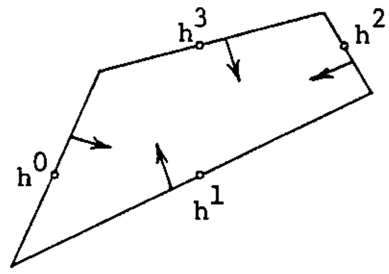


FIGURE 3

each of whose sides is associated with a given row of the matrix, and which contains no lattice points other than the four which define the primitive set.

It will be argued, later in this paper, that primitive sets contain no more than 2^n vectors if the set X is generated by an activity analysis model with n activities. If, for example $n = 2$, primitive sets will consist of either three or four lattice points in the plane. This leads to an apparent inconsistency in the definition of primitive sets as sets of $(m+1)$ vectors in R^{m+1}, if m is sufficiently large.

From a geometric point of view this arises because the set X does not have sufficiently high dimension to resist the downward movement of the positive orthant. Figure 4 illustrates this point with a set of points X which can be thought of as arising from an activity analysis matrix with three rows and one column. As we see, either the first or second

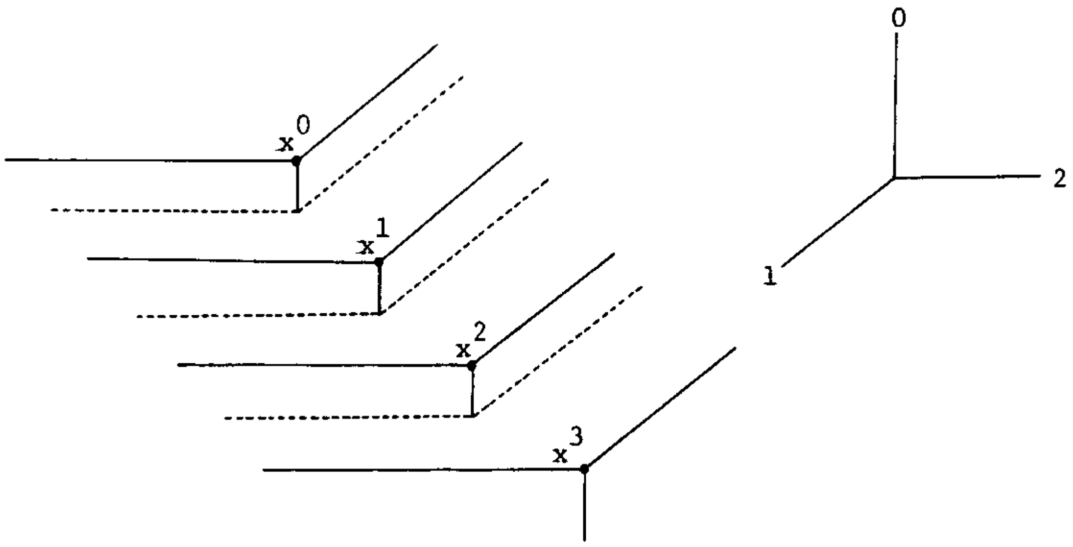


FIGURE 4

coordinate of the vertex of the translated positive orthant can always be reduced without passing through any of the points in X .

This difficulty may be overcome by the formal introduction of (m+1) "ideal" vectors $\xi^0, \xi^1, \ldots, \xi^m$, which are called slack vectors, because of an analogy with linear programming. As we shall see they simply indicate which coordinates of the vectors in X are being neglected at a given moment.

1.5 [Definition of the slack vectors]. The slack vector ξ^i is defined by saying that its i^{th} coordinate is <u>less than</u> the i^{th} coordinate of any of the vectors in X , and its j^{th} coordinate (for $j \neq i$) is <u>larger</u> than the j^{th} coordinate of any of the vectors in X .

The definition of primitive sets given in 1.4 is now extended to include primitive sets, some of whose members are slack vectors, and the remainder vectors in X. In Figure 4 the vectors x^0 and x^1, in conjunction with the slack vector ξ^1 form a primitive set, as do the triples (x^0, x^1, ξ^2) and (x^2, x^3, ξ^1).

If the set X, described in Figure 1, is assumed to be finite and consist of the seven points $x^0, \ldots, x^6$, there will be a number of primitive sets which involve slack vectors. Examples are (x^1, x^6, ξ^1) and (x^6, x^5, ξ^1) as well as (x^1, ξ^1, ξ^2).

II. Maximization Problems and the Local Neighborhood Structure

We shall be concerned with the problem of finding that vector x^j in X which maximizes x_0^j subject to the inequalities

$$\begin{aligned} x_1^j &\geq b_1 \\ &\vdots \\ x_m^j &\geq b_m , \end{aligned}$$

with $b_1, \ldots, b_m$ preassigned numbers. In the event that the vectors in X arise from an activity analysis model (1.1) with integral activity levels our problem becomes the customary integer programming problem

$$\max a_{01}h_1 + \ldots + a_{0n}h_n , \quad \text{subject to}$$

$$\begin{aligned} a_{11}h_1 + \ldots + a_{1n}b_n &\geq b_1 \\ &\vdots \\ a_{m1}h_1 + \ldots + a_{mn}h_n &\geq b_m \end{aligned}$$

and $h = (h_1, \ldots, h_n)$ a vector of integers.

Our purpose in this section will be to discuss one of the relationships between primitive sets and discrete maximization problems. A vector $x \varepsilon X$ is said to be efficient if there is no vector in X all of whose coordinates are strictly larger than those of x. The vectors in X which are not efficient are, clearly, contained in no primitive sets, since the downward movement of the positive orthant will be resisted before reaching such a vector.

The concept of primitive sets permits us to define a finite set of vectors which are neighbors of a given efficient vector in X.

2.1 [Definition]. Let x be an efficient vector in X. A vector x' in X (or one of the slack vectors) is defined to be a neighbor of x if they are both members of a common primitive set.

In Figure 1 the vector x^0 has six neighbors: the vectors $x^1, \ldots, x^6$. This will be seen to be the typical situation when the set X is generated by an activity analysis matrix with three rows and two columns.

2.2 [Theorem]. An efficient vector x in X has a non-empty set of neighbors. In particular for any $\ell = 0, 1, \ldots, m$ that vector x' in X (or the slack vector ξ^ℓ) whose ℓ^{th} coordinate is maximal, subject to $x'_i > x_i$ for all $i \neq \ell$, is a neighbor of x.

We demonstrate this theorem for the case $\ell = 0$, by the following geometrical argument. Translate the positive orthant so that its vertex coincides with x. Since x is efficient there will be no vectors in X in this positive orthant. Then translate the orthant by lowering the zeroth coordinate only until a vector $x' \varepsilon X$ is reached. This

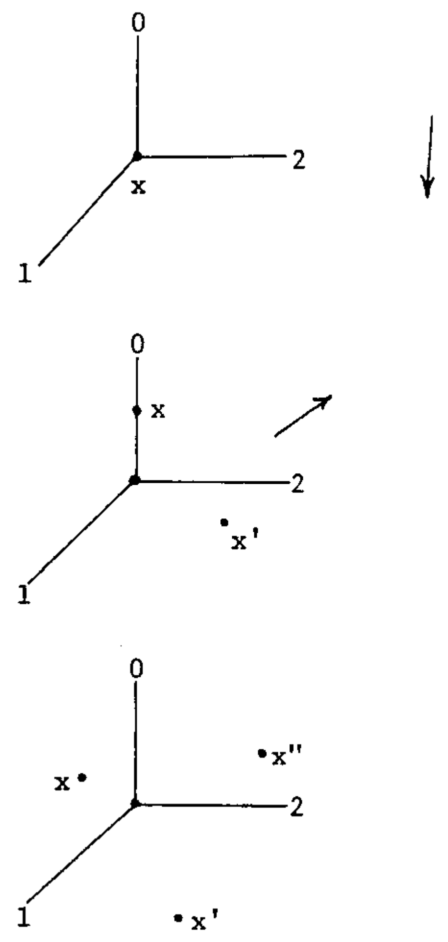


FIGURE 5

vector, whose zeroth coordinate is maximal, subject to $x'_i > x_i$ for $i = 1, 2, \ldots, n$, is the vector referred to in the statement of Theorem 2.2. (If no vector in X is ever reached by decreasing the zeroth coordinate, we use the notation x' for the zeroth slack vector ξ^0.) We then continue by decreasing the first coordinate of the vertex until a vector x'' (or the first slack vector) is reached. This construction, when continued through all of the coordinates, obviously leads to a primitive set containing x and x'.

The concept of the neighborhood of an efficient vector x in X may be applied to the problem of finding that vector x in X which maximizes x_0 subject to the inequalities

$$\begin{aligned} x_1 &\geq b_1 \\ &\vdots \\ x_m &\geq b_m \,. \end{aligned}$$

As the following theorem states an efficient vector which satisfies these inequalities is a global maximum if it is a local maximum when compared only with the finite set of its neighbors.

2.3. [Theorem]. Let x^* be an efficient vector in X and satisfy the inequalities $x_i^* \geq b_i$ for $i = 1, \ldots, m$. Assume that for every neighbor x' of x^* either

1. $x_i' < b_i$ for some $i = 1, \ldots, m$ or
2. $x_0' < x_0^*$.

Then x^* is that vector in X which maximizes x_0 subject to $x_i \geq b_i$ for $i = 1, \ldots, m$.

The proof of Theorem 2.3 is by induction on m; it is clearly correct if $m = 1$. Let us consider those points in X which satisfy the inequality $x_m \geq b_m$ and project them into R^m by disregarding the last coordinate. If T is used to denote the projection operator $T : (x_0, \ldots, x_{m-1}, x_m) \to (x_0, \ldots, x_{m-1})$, we define Y to be the discrete production set in R^m obtained by considering all of the points $y = Tx$ with x in X and $x_m \geq b_m$. The set Y, illustrated in Figure 6 clearly satisfies Assumptions 1.1 and 1.2.

As Figure 6 indicates the image Tx of an efficient vector in X need not be efficient in Y. There is one important case, however, in which this is so.

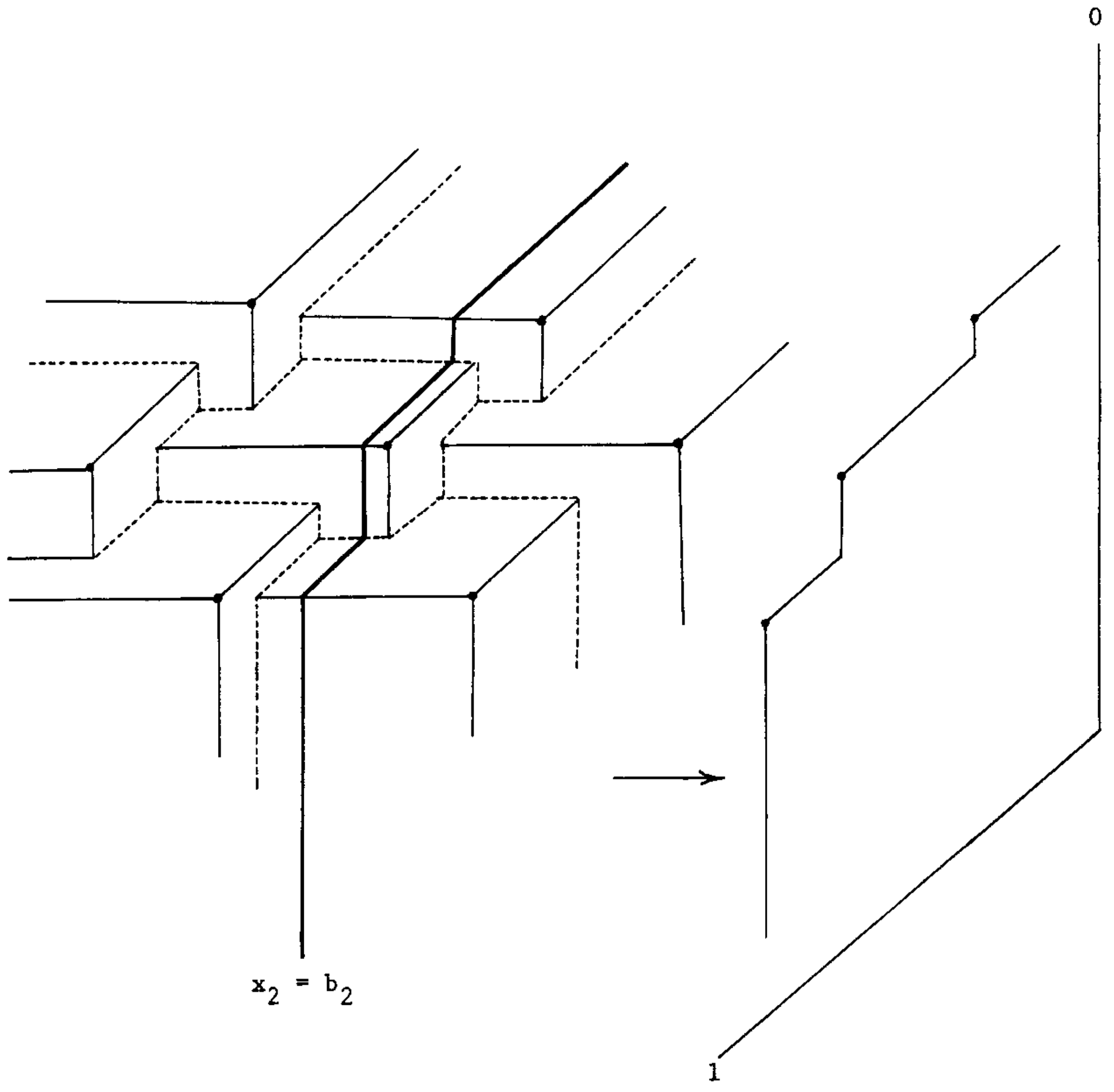


FIGURE 6

2.4 [Lemma]. Let x^* be a local maximum for our programming problem, i.e. satisfy the hypotheses of Theorem 2.3. Then $y^* = Tx^*$ is efficient in Y .

If this were not so there would be a vector x' in X satisfying

$x'_m \geq b_m$ and $x'_i > x_i$ for $i = 0, \ldots, m-1$. In fact we may take x' to be that vector in X which maximizes x'_m subject to $x'_i > x_i$ for $i = 0, \ldots, m-1$. But then by Theorem 2.2, x' is a neighbor of x^* , which satisfies

$$\begin{aligned} x'_0 &> x^*_0 \\ x'_1 &> x^*_1 \geq b_1 \\ &\vdots \\ x'_{m-1} &> x^*_{m-1} \geq b_{m-1} \\ x'_m &\geq b_m , \end{aligned}$$

contradicting the assumption that x^* is a local maximum.

Having demonstrated that $y^* = Tx^*$ is efficient in Y , we are now prepared to apply Theorem 2.3 by induction to sets of points in R^m . This is facilitated by the following lemma.

2.5 [Lemma]. Let $y = Tx$ be a neighbor of $y^* = Tx^*$ in Y . Then x is a neighbor of x^* in X .

The fact that $y = Tx$ is a neighbor of $y^* = Tx^*$ in Y will be revealed by their membership in a common primitive set in Y , composed, say, of the vectors

$$(y^*, y, y^2, \ldots, y^{m-1}) .$$

Each of these vectors is the image, under T of a vector in X , whose m^{th} coordinate is $\geq b_m$. Let the vectors in X be denoted, using an obvious notation, by $x^*, x, x^2, \ldots, x^{m-1}$. In order to demonstrate our Lemma it is sufficient to exhibit a vector x^m in X (with $x^m_m < b_m$)

so that $x^*, x, x^2, \ldots, x^m$ is a primitive set in X. But this vector may simply be defined to be the vector in X whose mth coordinate is maximal subject to

$$\begin{aligned} x_0^m &> \min[y_0^*, y_0, y_0^2, \ldots, y_0^{m-1}] \\ &\vdots \\ x_{m-1}^m &> \min[y_{m-1}^*, y_{m-1}, y_{m-1}^2, \ldots, y_{m-1}^{m-1}] . \end{aligned}$$

To complete the proof of Theorem 2.3 we observe that if x^* is a local maximum for the problem

$$\begin{aligned} &\max x_0 \\ &\quad x_1 \geq b_1 \\ &\quad\vdots \\ &\quad x_m \geq b_m , \end{aligned}$$

in X, then $y^* = Tx^*$ will be a local maximum for the problem

$$\begin{aligned} &\max \quad y_0 \\ &\quad y_1 \geq b_1 \\ &\quad\vdots \\ &\quad y_{m-1} \geq b_{m-1} \end{aligned}$$

in Y. For if there were a neighbor $y = Tx$ for which

$$\begin{aligned} y_0 &> y_0^* \\ y_1 &\geq b_1 \\ &\vdots \\ y_{m-1} &\geq b_{m-1} , \end{aligned}$$

then by Lemma 2.5, x would be a neighbor of x^* in X. But x would then satisfy $x_m \geq b_m$ in addition to

$$\begin{aligned} x_0 &> x_0^* \\ x_1 &\geq b_1 \\ &\vdots \\ x_{m-1} &\geq b_{m-1}, \end{aligned}$$

contradicting the assumption that x^* is a local maximum.

Having established that y^* is a local maximum in Y, our induction assumption permits us to conclude that y^* is a global maximum in Y. It is then immediate that x^* is a global maximum in X, for if x in X satisfies

$$\begin{aligned} x_0 &> x_0^* \\ x_1 &\geq b_1 \\ &\vdots \\ x_m &\geq b_m, \end{aligned}$$

it follows that $y = Tx$ is in Y and satisfies

$$\begin{aligned} y_0 &> y_0^* \\ y_1 &\geq b_1 \\ &\vdots \\ y_{m-1} &\geq b_{m-1}. \end{aligned}$$

This demonstrates Theorem 2.3. A converse to this Theorem will be given in Section V. We demonstrate there that any neighborhood system, for which a local maximum is global, must include the neighborhoods given by primitive sets.

Theorem 2.3 suggests an obvious algorithm for the solution of discrete programming problems. Begin with an efficient vector which satisfies the inequalities $x_i \geq b_i$ for $i = 1, \ldots, m$. Examine the neighbors of x. If each of them either violates one of the inequalities or gives a lesser value to the 0^{th} coordinate, we terminate with the global optimum. Otherwise replace x by one of its neighbors which satisfies the inequalities, and yields a higher value of the 0^{th} coordinate, and continue.

The ease with which this idea can be implemented depends on the ease with which the neighborhood structure associated with the technology X can be determined. It should not be surprising, therefore, if no structure whatsoever is imposed on X, that the determination of the neighborhood structure is as complex as solving the original programming problem itself. With complete generality the above algorithm will be at best a systematic way of organizing what is inevitably a search through the entire set X.

On the other hand if the set X has a sufficiently rich structure, the associated primitive sets and neighborhood structure may be quite easy to determine. I will illustrate this by anticipating a subsequent theorem which forms the basis for an extremely rapid algorithm (an algorithm which solves the problem in polynomial time--using the terminology of complexity theory) for the general integer programming problem with two variables.

Assume that the set X is generated by an activity analysis matrix with 3 rows and 2 columns whose entries have the following sign pattern

$$\begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix} .$$

In addition let $a_{i1} + a_{i2} > 0$ for $i = 1, 2$. Then it may be shown that the primitive sets correspond, in activity analysis space, to one of the two triangles illustrated in Figure 7, translated to an arbitrary

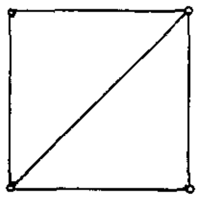


FIGURE 7

lattice point. Each lattice point will have, therefore, the six neighbors shown in Figure 8.

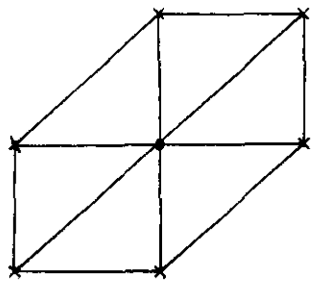


FIGURE 8

In order to solve the programming problem

$$\max a_{01}h_1 + a_{02}h_2$$
$$a_{11}h_1 + a_{12}h_2 \geq b_1$$
$$a_{21}h_1 + a_{22}h_2 \geq b_2 ,$$

and $h = (h_1, h_2)$ integral it is therefore sufficient to find a vector (h_1, h_2) which satisfies the two inequalities, and such that $(h_1 - 1, h_2)$ violates the first inequality and $(h_1 - 1, h_2 - 1)$ violates the second (or alternatively $(h_1, h_2 - 1)$ violates the second and $(h_1 - 1, h_2 - 1)$ the first). For if $(h_1 - 1, h_2 - 1)$ violates inequality 2, then so does

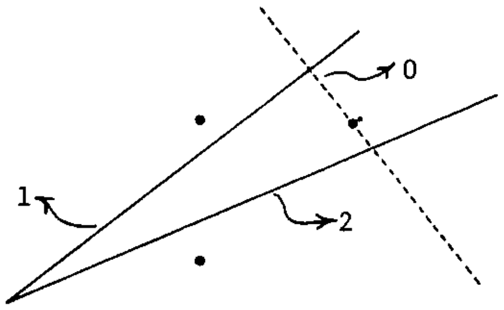


FIGURE 9

$(h_1, h_2 - 1)$, whereas the three other neighbors of (h_1, h_2) ($(h_1 + 1, h_2)$, $(h_1, h_2 + 1)$, $(h_1 + 1, h_2 + 1)$) all produce lower values of the objective function.

III. The Number of Binding Constraints for an Integer Program

We shall further illustrate the relationship between primitive sets and discrete programming problems by demonstrating a theorem on the maximal number of binding constraints in an integer programming problem with n variables. Consider the problem

$$\max a_{01}h_1 + \ldots + a_{0n}h_n , \quad \text{subject to}$$

3.1
$$\begin{gathered} a_{11}h_1 + \ldots + a_{1n}h_n \geq b_1 \\ \vdots \\ a_{m1}h_1 + \ldots + a_{mn}h_n \geq b_m , \end{gathered}$$

and with $h = (h_1, \ldots, h_n)$ integral. Non-negativity requirements on the variables, if any, will be incorporated in the constraints, whose number m will typically be larger than n. Assumptions 1.1 and 1.2 will be assumed to apply to the discrete production set generated by this model.

We assume that the inequalities have a feasible integral solution. Assumption 1.1 implies that there is a finite maximum and 1.2 that the optimal integral solution is unique.

3.2 [Definition]. A subset S of the inequalities is said to be binding, if the integer programming problem obtained by discarding the inequalities not in S has the same optimal solution.

The question to be raised is whether there is a function of n, say $f(n)$, such that an integer program with n variables always has a set of binding constraints of cardinality $f(n)$ or less. It is one of the major theorems of linear programming--in which the variables are not restricted to be integral--that a set of binding constraints of cardinality n can always be found. This result is, in fact, the basis for the simplex method for linear programming, which proceeds by systematically analyzing appropriate subsets of n inequalities. The result also leads to the pricing theorems of linear programming, with their important implications for the decentralization of economic activity.

Of course, it is conceivable that no function of n will suffice for integer programming, and that problems may be found with a fixed number of variables and an arbitrarily high number of constraints, none of which can be discarded without modifying the answer. The following theorem, first demonstrated by David Bell [Bell, 1977] and independently (though somewhat belatedly) by myself [Scarf, 1977]. states that the function $f(n) = 2^n - 1$ is the correct one for integer programming.

3.3 [Theorem]. An integer programming problem with n variables has a set of binding constraints of cardinality $2^n - 1$ or less.

At this point I will give Bell's argument for Theorem 3.3, rather than mine. Both arguments, however, make use of the following geometrical lemma, which seems to me to be at the heart of integer programming problems.

3.4 [Lemma]. Let P be a convex polyhedron in R^n, whose vertices are lattice points, and which contains no lattice points other than its vertices. Then the number of vertices is no larger than 2^n.

The unit cube in n space is an example of the type of convex polyhedron referred to in the lemma with a maximal number of vertices. It may be shown that when $n = 2$ any such polyhedron with 4 vertices is equivalent, under a unimodular transformation to the unit square, a fact that accounts for a good deal of the simplicity of programming problems with 2 variables. This simple characterization of the maximal polyhedra of Lemma 3.4 is, however, no longer correct when $n \geq 3$. The detailed study of these polyhedra is just being initiated.

The proof of Lemma 3.4 is quite simple. Let the vertices be $v^1, v^2, \ldots, v^k$. If $k > 2^n$ then there must be at least one pair of vertices, say v^1 and v^2, all of whose coordinates have the same parity, in terms of being even or odd. But then $(v^1 + v^2)/2$ is integral, contained in the polyhedron, and not a vertex. This completes the argument.

Let us return to the programming problem 3.1, which we assume to have an optimal solution h^0. Let ε be small and consider the polyhedron defined by

3.5
$$\sum_j a_{0j}h_j \geq \sum a_{0j}h_j^0 + \varepsilon$$
$$\sum_j a_{ij}h_j \geq b_i \quad \text{for} \quad i = 1, \ldots, m \; .$$

By the definition of h^0, this polyhedron is free of lattice points. We wish to show that there is a subset of 2^n or less of these inequalities (including the inequality derived from the objective function) so that the larger polyhedron obtained by deleting the remaining inequalities is also free of lattice points.

Every lattice point in R^n is, by construction, eliminated by at least one of these inequalities, and, of course, each inequality eliminates many lattice points. Bell's argument begins with the following classification of the $(m+1)$ inequalities 3.5.

3.6 [Definition]. The inequality $\sum a_{ij}h_j \geq b_i$ is said to be of type I if it eliminates a lattice point which is not eliminated by any other inequality. It is said to be of type II if every lattice point which it eliminates is also eliminated by some other inequality.

This definition is illustrated by the top drawing in Figure 10 which represents an integer program with 2 variables and 4 inequalities. The feasible set has been shaded and the objective function moved inwards slightly from the optimal solution. Inequality 0 is of course of type I. Of the remaining inequalities 1 and 2 are of type I, and 3 and 4 are of type II.

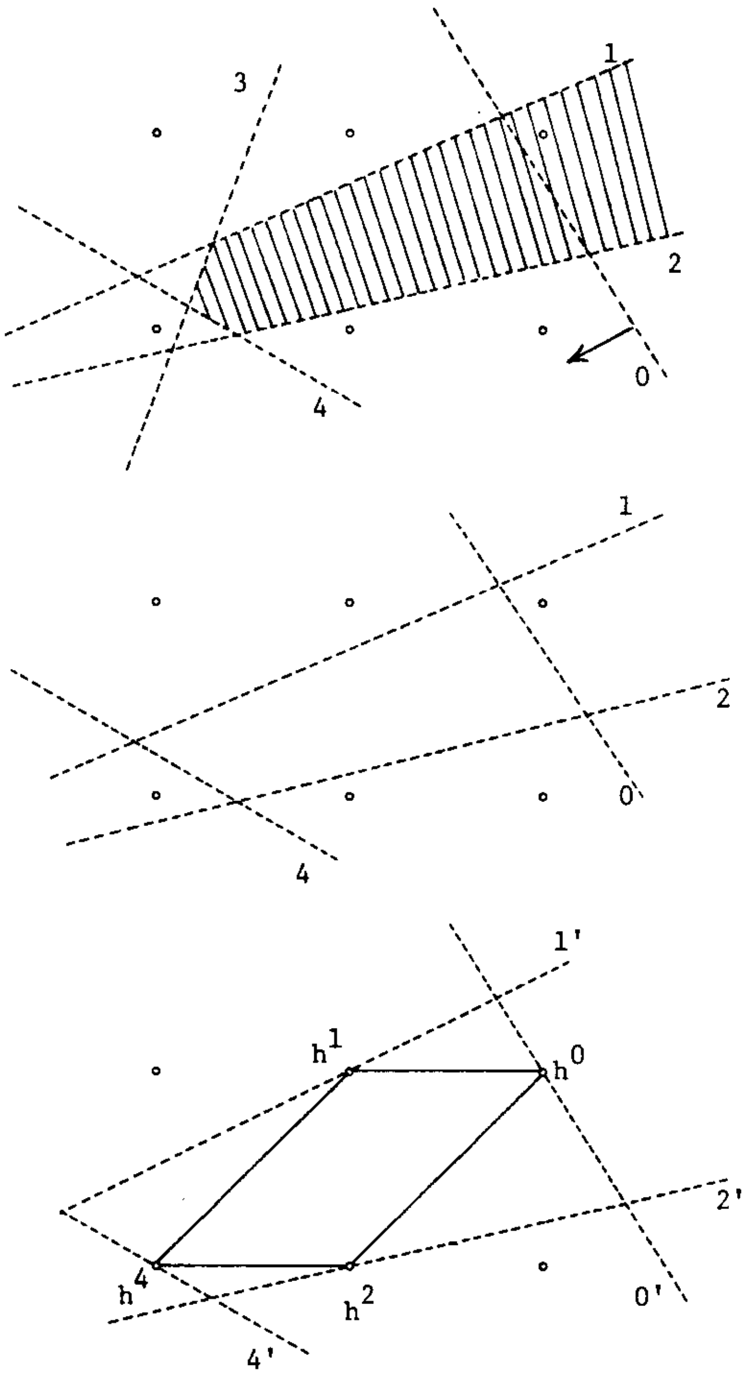


FIGURE 10

If an arbitrary inequality of type II is eliminated the convex polyhedron defined by the remaining inequalities will be enlarged, but it will still contain no lattice points. The resulting integer program will have a larger constraint set but the optimal solution will be unchanged. This is illustrated by the second drawing in Figure 10.

After an inequality of type II is eliminated, an inequality of type I will still be of type I, but inequalities of type II may change their character. If inequality 3 is eliminated in Figure 10, inequality 4 changes from an inequality of type II to an inequality of type I.

We may therefore continue the process of eliminating inequalities of type II, one at a time, until only type I inequalities remain. Theorem 3.3 will be demonstrated by showing that there can be no more than 2^n inequalities if they are all of type I. Consider the inequalities in the order of their subscripts, beginning with inequality 0. Relax inequality 0 until it hits the optimal solution of the programming problem. Relax each inequality, in turn, until it first hits a lattice point which it previously eliminated but which is not eliminated by any other inequality. When this relaxation is applied to any particular inequality in the sequence the convex polyhedron is enlarged but no lattice points are introduced into its interior. Moreover the inequalities remain as type I. When the process is completed each relaxed inequality will be associated with a specific lattice point which satisfies the remaining relaxed inequalities. This process is illustrated by the third drawing in Figure 10.

The process results in a set of lattice points $\{h^i\}$ for $i \varepsilon S$, where S is the set of indices referring to inequalities of type I. By the construction, the convex polyhedron formed by the relaxed inequalities contains no lattice points other than $\{h^i\}$, each of which is supported by its own translated inequality. H, the convex hull of the h^i is therefore a convex polyhedron, whose vertices are the h^i themselves, and which contains no other lattice points. It follows from Lemma 3.4 that the number of vertices (and therefore the number of inequalities of type I) is no larger than 2^n. This demonstrates Theorem 3.3.

It is a trivial matter to verify that the bound provided by 3.3 is sharp; i.e. that there are integer programs with n variables and $2^n - 1$ inequalities, whose optimal solution changes when any of the inequalities are discarded. Figure 11 illustrates this possibility for $n = 2$; a similar construction, based on the unit cube in n-space will work in general.

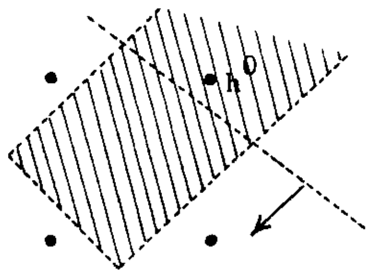


FIGURE 11

This observation casts some doubt on those methods for solving integer programs which examine subsets of n inequalities, solve the resulting programming problem and check to see whether the remaining inequalities are also satisfied. There may simply be no subset of n inequalities whose solution satisfies all of the constraints. The observation also reinforces our intuition that there are no useful pricing theorems for discrete programming problems.

Theorem 3.3 has been generalized by Alan Hoffman [Hoffman, 1978], who demonstrated that the maximum number of binding constraints in a programming problem with n integral variables and k real variables is no larger than $(k+1)2^n - 1$.

Bell's construction may be seen, quite easily, in terms of primitive sets. We define the set X to consist of the $m+1$ slack vectors $\xi^0, \ldots, \xi^m$ and the vectors $x = Ah$ as h ranges over all lattice points in R^n . By assumption the translate of the positive orthant in R^{m+1} with vertex at

$$\begin{matrix} \sum_{1}^{n} a_{0j} h_j^0 \\ b_1 \\ \vdots \\ b_m \end{matrix}$$

contains no vectors in X other than $x^0 = Ah^0$. We translate this vertex downward, lowering each coordinate in turn, until a primitive set is reached. The slack vectors in this primitive set will correspond to inequalities which are not binding and which may be discarded without changing the solution to the original programming problem. The existence of a set of binding constraints of cardinality $2^n - 1$ or less follows directly from the following theorem whose proof is an immediate consequence of Lemma 3.4.

3.5 [Theorem]. Let X consist of the slack vectors $\xi^0, \ldots, \xi^m$ and the points $x = Ah$ as h ranges over the lattice points in R^n . Then the number of non-slack vectors in a primitive set is less than or equal to 2^n .

IV. The Combinatorial Theorem Known as Sperner's Lemma

In this section we assume that X is a finite set. The following theorem is the analogue of Sperner's Lemma for primitive sets rather than simplicial subdivisions.

4.1 [Theorem]. Let each vector in X be given an integer label $\ell(x)$ selected from the set $(0, 1, \ldots, m)$. Let the i^{th} slack vector ξ^i be given the label $\ell(\xi^i) = i$ for $i = 0, 1, \ldots, m$. Then there exists at least one primitive set all of whose members have distinct labels.

Theorem 4.1 can be given a constructive argument based on the existence of a replacement operation for primitive sets. We consider an arbitrary primitive set $x^{j_0}, x^{j_1}, \ldots, x^{j_m}$, and ask whether there is a replacement for a given vector so that the new collection of $(m+1)$ vectors also forms a primitive set.

Before dealing with the general problem, let us examine geometrically the case in which the vectors in X lie in R^3. Figure 12 illustrates a primitive set composed of three vectors $x^{j_0}, x^{j_1}, x^{j_2}$. Without loss

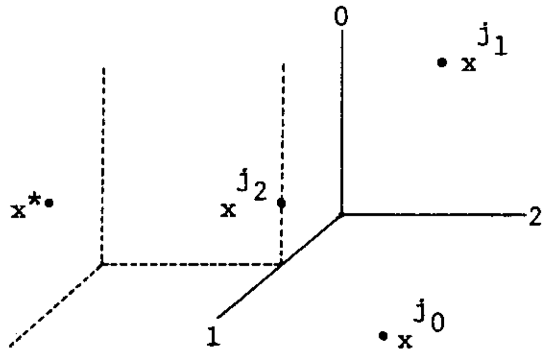


FIGURE 12

of generality I have selected the vectors so that

$$\min[x_i^{j_0}, x_i^{j_1}, x_i^{j_2}] = x_i^{j_i} \quad \text{for} \quad i = 0, 1, 2 .$$

In other words x^{j_i} lies on the i^{th} coordinate hyperplane of the translated orthant.

In order to remove, say, the vector x^{j_1}, we increase the first coordinate of the vertex of this orthant until we reach another vector in the primitive set, in this case x^{j_2}. We then decrease the second coordinate of the vertex until another vector $x*$ (or possibly the second slack vector) is reached. The unique replacement for x^{j_1} is $x*$.

In order to discuss the replacement operation for general values of m let us introduce a matrix whose columns are the $m+1$ vectors of a given primitive set

$$4.2 \qquad \begin{bmatrix} \underline{x_0^{j_0}} & x_0^{j_1} & \cdots & x_0^{j_m} \\ x_1^{j_0} & \underline{x_1^{j_1}} & & x_1^{j_m} \\ \vdots & \vdots & & \vdots \\ x_m^{j_0} & x_m^{j_1} & & \underline{x_m^{j_m}} \end{bmatrix} .$$

The vertex of the translated orthant associated with this primitive set is the vector of row minima. These must lie in different columns, since otherwise one of the columns themselves would be greater than the vector of row minima. I have assumed, without loss of generality, that the row minima--which are underlined--lie on the main diagonal.

In order to replace x^{j_0} we look at the second smallest entry in row 0; assumed in this case to be $x_0^{j_1}$. We then look through the set X to find that vector x which maximizes x_1 subject to

4.3 $$\begin{aligned} x_0 &> x_0^{j_1} \\ x_2 &> x_2^{j_2} \\ &\vdots \\ x_m &> x_m^{j_m} . \end{aligned}$$

The replacement for x^{j_0} is x and the new primitive set may be displayed as

4.4 $$\begin{bmatrix} x_0 & \underline{x_0^{j_1}} & \cdots & x_0^{j_m} \\ \underline{x_1} & x_1^{j_1} & & x_1^{j_m} \\ \vdots & \vdots & & \vdots \\ x_m & x_m^{j_1} & \cdots & \underline{x_m^{j_m}} \end{bmatrix} .$$

This construction clearly generates a new primitive set if there is a vector which satisfies 4.3. There certainly will be such a vector --the first slack vector ξ^1 will do--if the second smallest entry in row 0, is the zeroth coordinate of one of the vectors in X . And this will be the case unless the m vectors in the primitive set, other than the vector we are attempting to remove, are all slack vectors. In this case no replacement is possible.

For example, in Figure 1 the vector x^1 forms a primitive set in conjunction with the two slack vectors ξ^1 and ξ^2 , but it cannot be replaced. We shall take advantage of this type of exceptional primitive set by initiating our algorithm for a completely labeled set at one like this. Before examining this algorithm it is useful to verify that the replacement for x^{j_0} which has just been described is the unique replacement.

In order to see this let us consider the matrix 4.4 without assuming that we know the location of the row minima in the new primitive set. Aside from $x_m^{j_m}$ all of the entries in the last column are strictly larger than the corresponding entry in some other column. Since one of the row minima must appear in the last column we see that it must be $x_m^{j_m}$ which is the smallest entry in row m. Using precisely the same argument we see that $x_i^{j_i}$ is the smallest entry in row i for $i = 2, \ldots, m$.

The smallest entry in row 1 is either x_1 or $x_1^{j_1}$. If it is the latter the vector x must be that vector in X which maximizes x_0 subject to $x_i > x_i^{j_i}$ for $i = 1, \ldots, m$. But this is the vector x^{j_0} and we are back at the original primitive set. It follows that the disposition of row minima is that given by 4.4, an observation which determines the replacement uniquely. We summarize these observations in the following theorem.

4.5 [Theorem]. The replacement for a given vector in a primitive set exists and is unique, except for the case in which the primitive set consists of m slack vectors, and a single non-slack vector which we are attempting to remove. In this latter case no replacement exists.

The algorithm for finding a completely labeled primitive set begins with the primitive set consisting of the m slack vectors $\xi^1, \ldots, \xi^m$, and that particular vector x in the finite set X whose zeroth coordinate is maximal. If $\ell(x) = 0$ we have found a completely labeled primitive set since the slack vectors bear all of the remaining labels. If on the other hand $\ell(x) = i$ we remove the i^{th} slack vector and reach a new primitive set.

The algorithm will move through primitive sets whose $(m+1)$ vectors will bear all of the labels $1, 2, \ldots, m$. The algorithm terminates when the label 0 appears, and prior to termination each primitive set will contain precisely two vectors which have the same label. One of these vectors has just been introduced into the primitive set. We continue by removing the other vector with the doubled label.

A familiar graph theoretic argument demonstrates that we never return to a primitive set previously encountered. Consider a graph whose nodes represent the primitive sets through which the algorithm passes. Two nodes will be adjacent, and connected by an edge, if one of the primitive sets is obtained from the other by removing one of the vectors with a doubled label. Since this relationship is symmetric the edges need not be ordered.

The initial and terminal primitive sets have nodes which are adjacent to a single other node. Each intermediary node is adjacent to precisely two other nodes. If the algorithm were to return to a node previously visited, the first node which is encountered twice--if it is not the initial position--would necessarily be adjacent to at least three other nodes, which is impossible. If the first primitive set which is revisited were the initial position it would necessarily be adjacent to at least two other nodes--again impossible.

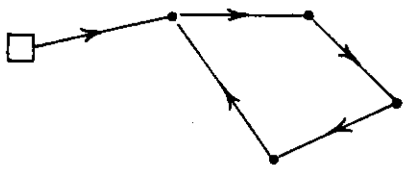


FIGURE 13

One final remark to complete the proof. Every replacement operation called for in the course of the algorithm can actually be carried out. If not we would be at a primitive set containing m slack vectors. The label zero would have already been brought in and the algorithm would have previously terminated unless the slack vectors are $\xi^1, \ldots, \xi^m$. But in this case we would have returned to our original primitive set, a possibility we have already ruled out. This completes our algorithm for Sperner's Lemma.

Let us enlarge our graph by considering all primitive sets whose $(m+1)$ vectors bear the labels $1, 2, \ldots, m$, rather than only those encountered in the course of the algorithm. As before, two nodes are adjacent if one is obtained from the other by removing one of the two columns with the doubled label.

Aside from the initial primitive set and the completely labeled primitive sets, each node is adjacent to two other nodes. The initial primitive set and the completely labeled primitive sets are adjacent to precisely one other node. Such a graph must have the form illustrated in Figure 14. This observation leads immediately to the following

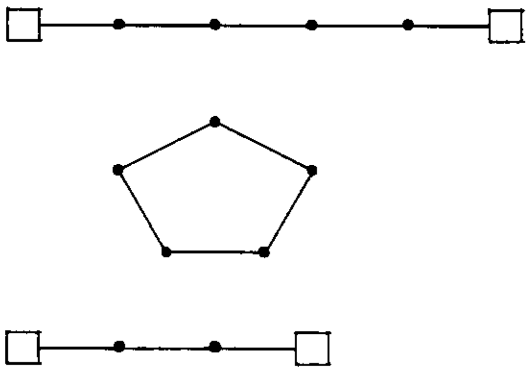


FIGURE 14

refinement of Sperner's Lemma.

4.6 [Theorem]. The number of completely labeled primitive sets is odd.

V. The Application of Sperner's Lemma to Discrete Programming Problems

Sperner's Lemma has become quite familiar, during the last decade, because of its use in the approximation of fixed points of a continuous mapping. It may be somewhat surprising, however, that it has an immediate application to discrete programming problems, as well.

Let us return to the problem of finding that vector in X which maximizes x_0 subject to the inequalities

$$\begin{gathered} x_1 \geq b_1 \\ \vdots \\ x_m \geq b_m \, . \end{gathered}$$

We shall assume that X is finite. If this is not so, then we restrict our attention to that finite subset satisfying $x_i \geq -M$ (with M a large positive number) for $i = 0, 1, \ldots, m$. We adopt the following labeling rule.

5.1 [Labeling Rule]. We label x in X with the label $\ell(x) = i$ if i is the largest index for which $x_i < b_i$. If $x_i \geq b_i$ for all $i = 1, 2, \ldots, m$, then $\ell(x) = 0$.

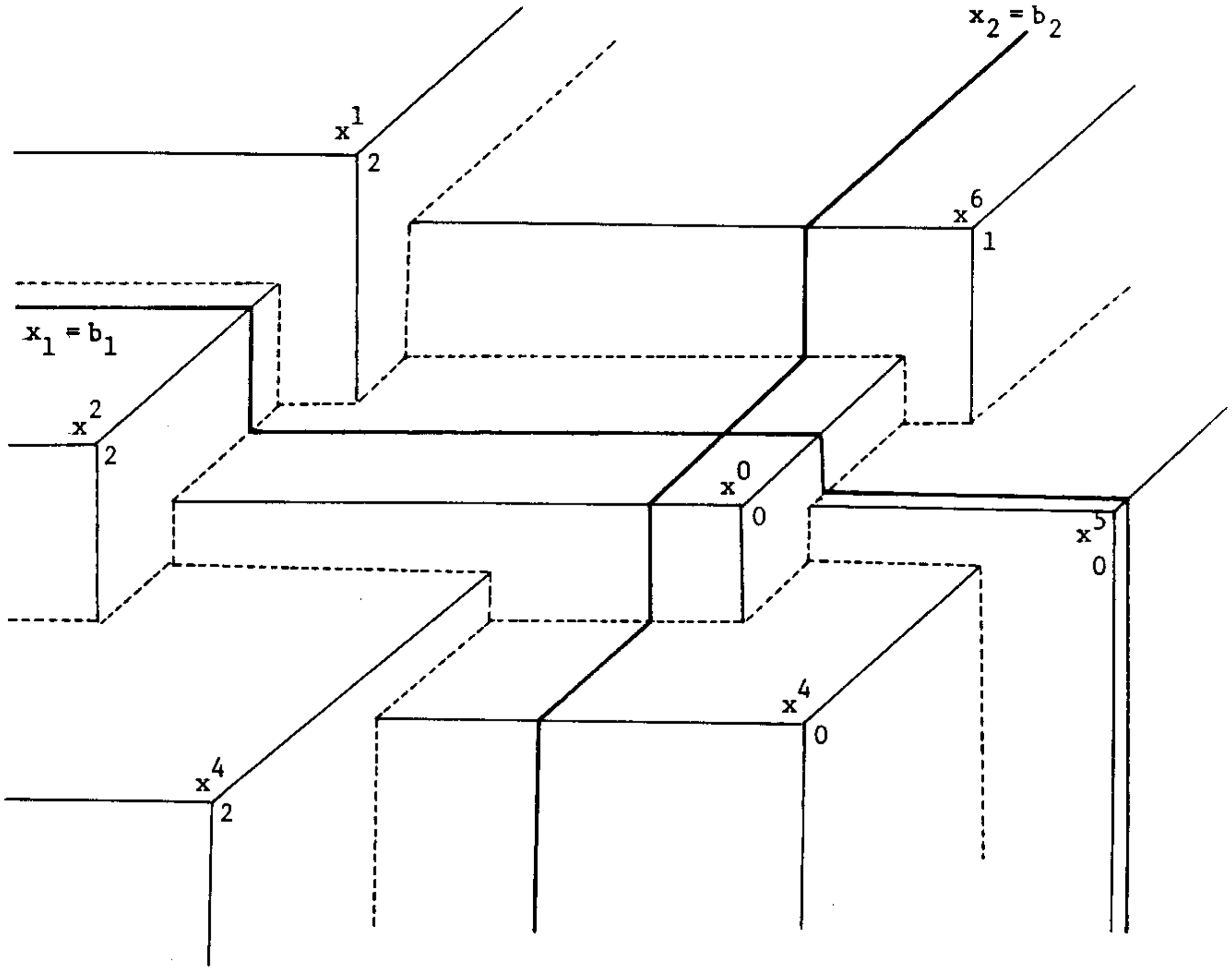


FIGURE 15

Figure 15 is a redrawing of Figure 1 with two additional lines $x_1 = b_1$ and $x_2 = b_2$. The labels for the seven vectors are given by 5.1. We see that there is only one completely labeled primitive set and that the vector in this primitive set with the label 0, x^0, is the optimal solution to the programming problem. The general argument is given in the following theorem.

5.2 [Theorem]. Let x be that vector in a completely labeled primitive set--labeled according to 5.1--with the label 0. If x is not a slack vector then it maximizes x_0 among all vectors in X which satisfy $x_i \geq b_i$ for $i = 1, \ldots, m$. If x is a slack vector, then it is the zeroth slack vector, and the constraints are infeasible.

Without loss of generality let us assume that the vectors in a completely labeled primitive set are given by $x^0, x^1, \ldots, x^m$, arranged in such a way that the row minima of

$$\begin{bmatrix} \underline{x_0^0} & x_0^1 & \cdots & x_0^m \\ x_1^0 & \underline{x_1^1} & & x_1^m \\ \vdots & \vdots & & \vdots \\ x_m^0 & x_m^1 & \cdots & \underline{x_m^m} \end{bmatrix}$$

lie on the main diagonal. The particular form of the labeling rule 5.1 permits us to argue that $\ell(x^i) = i$ for $i = 0, \ldots, m$. To see this, we observe first of all, that $\ell(x^i) > i$ for all i. This is clearly correct for $i = 0$. On the other hand if $\ell(x^i) < i$ for some $i \geq 1$ then $x_i^i \geq b_i$ and therefore no vector receives the vector i. These observations imply immediately that $\ell(x^i) = i$ for all i.

We see that $x_i^i < b_i$ for $i = 1, \ldots, m$ and that x^0, if it is not the zeroth slack vector, satisfies all of the constraints of the programming problem. But then it must be the global maximum, for if there were another vector x in X with $x_i \geq b_i$ for $i = 1, \ldots, m$ and $x_0 > x_0^0$, we would have $x_i > x_i^i$ for all i, which violates the definition of a primitive set. If, on the other hand, x^0 is the zeroth slack vector, then an identical argument implies that there is no x in X with $x_i \geq b_i$ for $i = 1, \ldots, m$. This demonstrates Theorem 5.2.

We see that Sperner's Lemma can be used to provide an algorithm for discrete programming problems. The difficulty in its implementation is the replacement operation, which requires a knowledge of all of the primitive sets associated with a given technology X. If this approach is to be made useful, research must focus on methods for determining these primitive sets when sufficient structure is placed on X. In subsequent sections we shall illustrate how this may be done when the technology is based on an activity analysis model with 2 integral activities. The case of 3 activities is much more difficult and will be presented in a separate paper.

We should remark that the completely labeled primitive set is identical with the one obtained by Bell in his proof that the maximum number of binding constraints in a integer program with n variables is $2^n - 1$.

As a final topic let us return to a general production set X. In Section II we defined two vectors to be neighbors if they were contained in a common primitive set. The major conclusion was that an efficient vector in X which was a local maximum when compared with its neighbors was, in fact, a global maximum. The following definition provides a generalization of this concept of neighborhoods.

5.3 [Definition]. A neighborhood structure is defined by associating with each efficient vector x in X a non-empty subset of neighbors $N(x) \subset X$. The assignment is arbitrary aside from the requirement that $y \in N(x)$ implies that $x \in N(y)$.

A neighborhood structure permits us to define a local maximum for the programming problem: find x in X so as to maximize x_0 subject to $x_i \geq b_i$ for $i = 1, \ldots, m$. We say that an efficient vector x in X is a local maximum if it satisfies the constraints and if every vector in $N(x)$ either violates one of the constraints or has a smaller zeroth coordinate.

Let us assume that we are given a neighborhood structure with the property that for each vector b , a local maximum is a global maximum. We shall demonstrate that for every x , the neighborhood $N(x)$ must contain all vectors which are in a common primitive set with x . This implies that primitive sets provide the unique, minimal neighborhood system for which a local maximum is global.

Suppose that x and y are in some common primitive set, but that y is not in $N(x)$, nor x in $N(y)$. Without loss of generality we can assume that $x_0 < y_0$. Consider a primitive set which contains both x and y , and whose columns are given by

$$\begin{bmatrix} - & \cdots & x_0 & \cdots & y_0 & & \\ & & \vdots & & & & \\ & & & & \vdots & & \\ & & \underline{x_i} & & & & \\ & & & & \underline{y_j} & & \\ & & \vdots & & & \ddots & \\ & & & & & & - \end{bmatrix}$$

with the row minima assumed to lie on the main diagonal. By repeated applications of the replacement operation, removing those vectors with smaller zeroth coordinate then that of x , we will obtain a primitive set--containing x and y --with x having the smallest zeroth coordinate. By a change of notation, if necessary, we shall assume that y has the smallest m^{th} coordinate, so that our matrix takes the form

$$\begin{bmatrix} \underline{x_0} & x_1^1 & \cdots & y_0 \\ x_1 & \underline{x_1^1} & & y_1 \\ \vdots & \vdots & & \vdots \\ x_m & x_m^1 & & \underline{y_m} \end{bmatrix} .$$

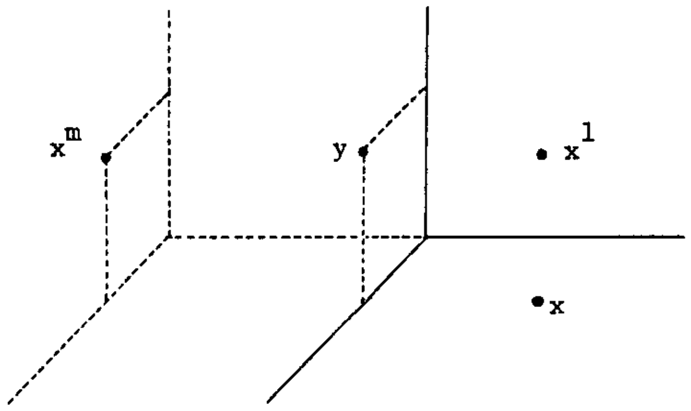
Now let $X^1 = X - \{y\}$. If x^m is that vector in X^1 whose m^{th} coordinate is maximal, subject to

$$\begin{aligned} x_0^m &> x_0 \\ x_1^m &> x_1^1 \\ &\vdots \\ x_{m-1}^m &> x_{m-1}^{m-1} , \end{aligned}$$

then $(x, x^1, \ldots, x^m)$ will be a primitive set in X^1, displayed by the matrix

$$\begin{bmatrix} \underline{x_0} & x_0^1 & \cdots & x_0^m \\ x_1 & \underline{x_1^1} & & x_1^m \\ \vdots & \vdots & & \vdots \\ x_m & x_m^1 & & \underline{x_m^m} \end{bmatrix} .$$

The relationship between these two primitive sets is illustrated by the following figure. Of course, y will be contained in the positive orthant whose vertex is $\min[x, x^1, \ldots, x^m]$.



Let us define a particular programming problem by selecting the vector b as

$$\min[x_1, y_1] \geq b_1 > x_1^1$$
$$\vdots$$
$$\min[x_m, y_m] \geq b_m > x_m^m .$$

Then it follows from the definition of primitive sets that x is that vector in X^1 which maximizes x_0 subject to $x_i \geq b_i$ for $i = 1, \ldots, m$. Since y is not in $N(x)$, x must be a local maximum in X using the neighborhood $N(x)$. But x is not a global maximum in X since y also satisfies the constraints and $y_0 > x_0$. This demonstrates the following theorem.

5.4 [Theorem]. A neighborhood system for which a local maximum is global, for all vectors b, must contain the neighborhood system defined by primitive sets.

VI. Index Theory and Monotonicity

In this section we shall use the concepts of index theory, applied to primitive sets, to analyze the graph of almost completely labeled primitive sets displayed in Figure 14. Our major conclusion will be that when the labeling rule is given by 5.1, the graph contains no cycles and is composed of a single chain connecting the initial primitive set with the unique completely labeled primitive set.

This result seems important to me for two reasons. First it implies that we need not start the algorithm with the primitive set consisting of the m slack vectors $\xi^1, \ldots, \xi^m$. Any primitive set whose $m+1$ members bear the labels $1, 2, \ldots, m$ will lie on the unique chain leading to the required answer. As we shall see this flexibility will be quite useful in discussing programming problems with two integral activities.

This result also suggests that our algorithm for discrete programming has captured one of the significant properties which differentiate algorithms for convex programming from the more subtle techniques required for fixed point computations.

Let us return to the general problem studied in Section IV. The set X is taken to be finite and the labels $\ell(x)$ are arbitrary members of the set $(0, 1, \ldots, m)$. As before the i^{th} slack vector will receive the label $\ell(\xi^i) = i$. We introduce the following definition of the index of a completely labeled primitive set.

6.1 [Definition]. Let $x^{j_0}, \ldots, x^{j_m}$ be a completely labeled primitive set arranged so that the row minima of the matrix

6.2 $$\begin{bmatrix} \underline{x_0^{j_0}} & x_0^{j_1} & \cdots & x_0^{j_m} \\ x_1^{j_0} & \underline{x_1^{j_1}} & & x_1^{j_m} \\ \vdots & \vdots & & \vdots \\ x_m^{j_0} & x_m^{j_1} & & \underline{x_m^{j_m}} \end{bmatrix}$$

lie on the main diagonal. Then

$$\text{index}(x^{j_0}, \ldots, x^{j_n})$$

is defined to be +1 if the permutation $\ell(x^{j_0}), \ldots, \ell(x^{j_m})$ is <u>even</u>, and -1 if the permutation is <u>odd</u>.

We shall demonstrate the following important generalization of Theorem 4.6 which states that the number of completely labeled primitive sets is odd.

6.3 [Theorem]. The number of completely labeled primitive sets with index +1 exceeds the number with index -1 by unity.

The proof of Theorem 6.3 is based on our ability to orient the graph of almost completely labeled primitive sets by a calculation which depends solely on the data involved in the particular primitive set being studied.

An orientation is a designation of the direction in which the vertices of each component of the graph are to be traversed. Consider a primitive

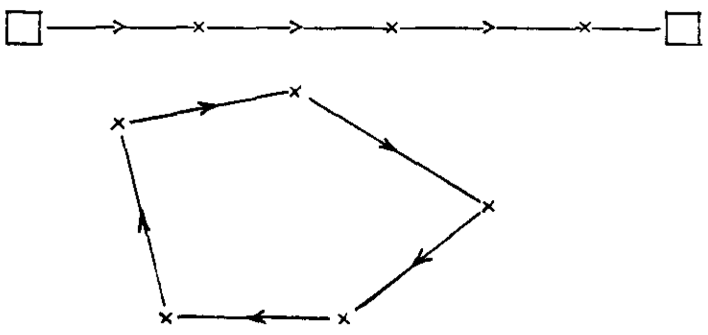


FIGURE 16

set $(x^{j_0}, \ldots, x^{j_m})$ whose members bear the labels $1, 2, \ldots, m$, and which is arranged so that the row minima of the matrix 6.2 lie on the main diagonal. Two of the vectors, say x^{j_α} and x^{j_β} have the same labels; aside from the initial primitive set one or the other of them will be removed.

The string of symbols

$$\ell(x^{j_0}), \ldots, \ell(x^{j_\alpha}), \ldots, \ell(x^{j_\beta}), \ldots, \ell(x^{j_m})$$

will not be a permutation of $(0, 1, \ldots, m)$, since the label 0 is missing. But both

$$6.4 \qquad \begin{cases} \ell(x^{j_0}), \ldots, 0, \ldots, \ell(x^{j_\beta}), \ldots, \ell(x^{j_m}), \text{ and} \\ \ell(x^{j_0}), \ldots, \ell(x^{j_\alpha}), \ldots, 0, \ldots, \ell(x^{j_m}) \end{cases}$$

will be permutations. In fact the two permutations will have opposite parity since they are obtained, one from the other, by a single transposition.

In order to orient the graph we must select one of the two vectors to be removed.

6.5 [Prescription of an Orientation]. Let us orient the graph of almost completely labeled primitive sets by removing that vector with the property that when we replace its label by 0 , the resulting permutation is <u>odd</u>.

Several remarks are in order. We must, first of all, verify that this orientation is consistent in the sense that if we move from one vertex to an adjacent one, the next step does not require us to return to the original vertex. Consider a primitive set whose columns form the matrix

$$\begin{bmatrix} \underline{x_0^{j_0}} & \cdots & x_0^{j_\alpha} & \cdots & x_0^{j_m} \\ \vdots & & \vdots & & \vdots \\ x_\alpha^{j_0} & & \underline{x_\alpha^{j_\alpha}} & & x_\alpha^{j_m} \\ \vdots & & \vdots & & \vdots \\ x_m^{j_0} & & x_m^{j_\alpha} & & \underline{x_m^{j_m}} \end{bmatrix}$$

arranged as usual so that the row minima lie on the main diagonal. Assume that $\ell(x^{j_0}), \ldots, 0, \ldots, \ell(x^{j_m})$ is odd, where $\ell(x^{j_\alpha})$ has been replaced by 0 ; x^{j_α} is to be removed, and replaced by a vector x .

We must demonstrate that 6.5 does not require us to remove x from the new primitive set. The permutation $\ell(x^{j_0}), \ldots, 0, \ldots, \ell(x^{j_m})$ obtained by replacing $\ell(x)$ by 0 in the new primitive set is identical with the previous permutation and is therefore odd. But a simple transposition (see, for example, 4.4) of two columns is required to bring the new primitive set to the form in which the row minima of the corresponding matrix lie on the main diagonal. This transposition will change the sign of the permutation so that the other vector with the doubled label is removed.

6.6 [Lemma]. A completely labeled primitive set which is reached by traversing the graph in the direction given by 6.5 has an index of +1. If such a set is obtained by moving in the opposite direction, the index is -1.

In order to demonstrate this lemma, return to the notation we have just used, and assume that the incoming vector x has the label 0, so that we have reached a completely labeled primitive set. The permutation of labels $\ell(x^{j_0}), \ldots, \ell(x), \ldots, \ell(x^{j_m})$ is, of course, odd, but the single transposition required to bring the row minima of the final matrix to the main diagonal will convert the permutation to an even one. A virtually identical argument will demonstrate that a completely labeled primitive set obtained by moving in the opposite direction has index -1. This demonstrates 6.6.

A single completely labeled primitive set is obtained by initiating the algorithm at the primitive set composed of the slack vectors $\xi^1, \ldots, \xi^m$ and the vector x in X whose zeroth coordinate is maximal. Since the permutation $0, \ell(\xi^1), \ldots, \ell(\xi^m)$ is the identity permutation, the orientation rule is consistent with removing that slack vector whose label duplicates that of x. Our arguments therefore imply that the primitive set obtained by our algorithm has an index of +1.

The remaining completely labeled primitive sets may be grouped in pairs. The two members of each pair will lie at opposite ends of a connected chain in the graph of almost completely labeled primitive sets, and will therefore--by Lemma 6.6--have opposite indices. This demonstrates Theorem 6.3.

This important result is all that can be said about the indices of completely labeled primitive sets when the labels $\ell(x)$ are arbitrary. But a considerable sharpening is available when the labeling rule 5.1 is used to solve the discrete programming problem: find that vector x in X whose zeroth coordinate is maximal, subject to the inequalities

$$\begin{aligned} x_1 &\geq b_1 \\ &\vdots \\ x_m &\geq b_m \, . \end{aligned}$$

Let us now assume that labeling rule 5.1 is being used and let $x^0, x^1, \ldots, x^m$ be a completely labeled primitive set, arranged in our customary way:

6.8 $$\begin{bmatrix} \underline{x_0^0} & x_0^1 & \cdots & x_0^m \\ x_1^0 & \underline{x_1^1} & & x_1^m \\ \vdots & \vdots & & \vdots \\ x_m^0 & x_m^1 & \cdots & \underline{x_m^m} \end{bmatrix}$$

Then, by the argument previously given, we must have $\ell(x^i) = i$ for all i . We see therefore that the index associated with every completely labeled primitive set is +1. This demonstrates the following theorem.

6.9 [Theorem]. The labeling rule 5.1 results in a unique completely labeled primitive set.

The graph of the almost completely labeled primitive sets is seen therefore to consist of a single chain, connecting the initial primitive set to the unique completely labeled primitive set, and possibly a number of cycles. In the remainder of this section we shall demonstrate that there are, in fact, no such cycles.

The argument will be based on a detailed examination of the labels associated with the almost completely labeled primitive sets which are assumed to appear in such a cycle. Consider such a primitive set, $x^0, \ldots, x^m$, again arranged in such a way that the row minima of

6.10
$$\begin{bmatrix} \underline{x_0^0} & x_0^1 & \cdots & x_0^m \\ x_1^0 & \underline{x_1^1} & & x_1^m \\ \vdots & \vdots & & \vdots \\ x_m^0 & x_m^1 & \cdots & \underline{x_m^m} \end{bmatrix}$$

lie on the main diagonal. As before, it is easy to verify that the labeling rule 5.1 implies that $\ell(x^i) \geq i$.

Let us define for each such primitive set an increasing sequence of indices $0 = i_0 < i_1 < \ldots < i_k$ by

6.11
$$\begin{aligned} \ell(x^0) &= i_1 > 0 \\ \ell(x^{i_1}) &= i_2 > i_1 \\ &\vdots \\ \ell(x^{i_{k-1}}) &= i_k > i_{k-1} \\ \ell(x^{i_k}) &= i_k . \end{aligned}$$

We have the following lemma:

6.12 [Lemma]. $\ell(x^i) = i$ for all columns $i \neq 0, i_1, \ldots, i_{k-1}$.

The argument is immediate. We let S be the set of indices i in $(0, 1, \ldots, m)$ with $i \neq 0, i_1, \ldots, i_{k-1}$. For each such i in S there must be some x^j with $\ell(x^j) = i$. But $j \neq 0, i_1, \ldots, i_{k-1}$ since the labels of $x^0, x^{i_1}, \ldots, x^{i_{k-1}}$ are not in S. We see that the set of indices $\{\ell(x^i)\}$ for i in S is precisely S itself. Lemma 6.12 follows from the observation that $\ell(x^i) \geq i$ for all i.

The two columns $x^{i_{k-1}}$ and x^{i_k} have the doubled label and one of them will be removed as we proceed around a cycle following the orientation given by 6.5. The permutation

$$\ell(x^0), \ldots, 0, \ldots, \ell(x^m) ,$$

where $\ell(x^{i_k})$ has been replaced by 0 may be brought to the identity permutation by precisely k transpositions. It follows that this permutation is odd if the number k is odd; otherwise it is even.

6.13 [Lemma]. If the orientation given by 6.5 is followed, we remove x^{i_k} when k is odd, and $x^{i_{k-1}}$ when k is even.

In order to obtain a contradiction to the existence of a cycle using these arguments, it is convenient to define

$$\alpha = \min[x^0, \ldots, x^m] ,$$

for each primitive set in a cycle, and to study the way in which the coordinates of α change as we follow the orientation 6.5. For example, α_0 will increase only if the vector x^0 is removed, and will decrease if x^i is removed and x^0 is that vector in the primitive set with the second smallest i^{th} coordinate.

Can α_0 be increased when the labeling rule 5.1 is followed? This can only occur when x^0 has one of the doubled labels, so that $\ell(x^{i_1}) = i_1$. But Lemma 6.13 then tells us that the vector x^{i_1} will be removed as we traverse the cycle with the orientation 6.5, and α_0 is not increased. If, however, α_0 is not increased throughout a cycle, it can never decrease, and must remain constant. It follows that the vector x^0 is contained in every primitive set in the cycle, and in fact retains its role as the vector with smallest zeroth coordinate.

The index $\ell(x^0) = i_1$ will therefore be unchanged throughout the cycle. By Lemma 6.13 none of the vectors x^i for $0 < i < i_1$ which appear in any primitive set in the cycle will ever be removed. Therefore the i^{th} coordinate of α, for $0 < i < i_1$ will never be increased. This implies that every one of these coordinates will remain constant throughout the cycle and therefore all of the vectors $x^0, x^1, \ldots, x^{i_1-1}$ will be contained in every primitive set in the cycle. Moreover they will retain their roles in bearing the row minima for rows $0, 1, \ldots, i_1-1$.

In order to argue that x^{i_1} is never removed, let us avail ourselves of the opportunity of moving around the cycle in the reverse orientation. If x^{i_1} bears the doubled label then we must have $\ell(x^{i_2}) = i_2$. It follows from 6.13 that x^{i_2} will be removed in the reverse orientation. The coordinate α_{i_1} is never increased in the reverse orientation. It must therefore stay constant regardless of the orientation.

The proof then verifies that α_i, for $i_1 < i < i_2$, never changes and continues with α_{i_2}. In discussing α_{i_ℓ} we use the orientation

6.5 if ℓ is odd, and the reverse orientation if ℓ is even. The final contradiction, of course, is that none of the vectors $x^0, \ldots, x^m$ are removed throughout the cycle.

6.14 [Theorem]. When labeling rule 5.1 is used, the graph of the almost completely labeled primitive sets contains no cycles.

VII. General Remarks on Polynomial Algorithms

It is appropriate at this point to make an excursion into a topic studied in computer science, that of polynomial algorithms for discrete programming problems. This topic will be of considerable interest in our study of integer programming problems with two variables.

The general integer program

$$\max \sum_1^n a_{0j}h_j$$

$$\begin{aligned} \sum_1^n a_{1j}h_j &\geq b_1 \\ &\vdots \\ \sum_1^n a_{mj}h_j &\geq b_m , \end{aligned}$$

with $h_1, \ldots, h_n$ integral is specified by an $(m+1)\times n$ matrix A and a right hand side b. Let us imagine that they are all integral and stored in the computer in terms of their binary representation. Each of these $(m+1)n+m$ numbers requires a certain number of binary bits for its representation. The total number of bits required to represent all of the numbers is an integer, say I, which measures the complexity of the problem.

Let us consider an algorithm which solves a given set of integer programs in a finite number of basic steps, each of which consists of a single addition, subtraction, multiplication, division, or elementary logical operation. The algorithm is said to be polynomial if there is a polynomial function $f(I)$ of the complexity of the problem so that the algorithm terminates in no more than $f(I)$ steps, for all instances of the class of integer programs being studied whose complexity is no more than I .

Let us discuss an example which illustrates the difference between polynomial and non-polynomial algorithms. Consider the discrete analogue of finding a zero of a function of a single variable. We are given a function $g(j)$ defined on the integers $0, 1, 2, \ldots, n+1$ and which takes on the values ± 1 . We are told that $f(0) = -1$, $f(n+1) = +1$, and asked to find a pair of adjacent integers where the function has different values.

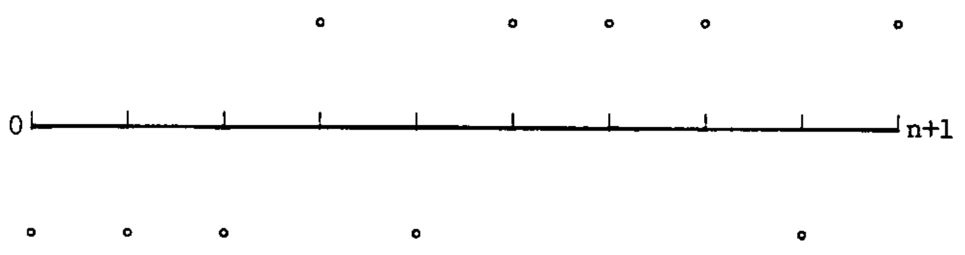


FIGURE 17

Let a basic step in an algorithm be a question as to the sign of $g(j)$ for a given integer j . A measure of the complexity of the problem (we are a bit loose here for illustrative purposes) is the integer n , which requires $\log_2 n$ bits for its binary representation.

One algorithm is to ask for the signs of $g(1), g(2), \ldots$ in order, and terminate when the first sign change occurs. This may require a total of n steps, and since n is not a polynomial function of $\log_2 n$, this will not be a polynomial algorithm.

Repeated bisection of the interval does, however, produce a polynomial algorithm. We bisect $[0, n+1]$ by asking for the sign of $g(j)$ where $j = [(n+1)/2]$ (the greatest integer in $(n+1)/2$). If this sign is positive there is a crossover in the interval $[0,j]$; if it is negative there is a crossover in the interval $[j, n+1]$. We continue by bisecting the interval in which a crossover is known to exist. Since each step reduces the interval by roughly a factor of 2, the process converges in essentially $\log_2 n$ steps--a linear function of the complexity of the problem.

A more interesting example is the Euclidean algorithm for finding the greatest common divisor (a,b) of two positive integers a and b . Assume that $a > b$ and define the sequence $r_1, r_2, \ldots, r_{k+1}$ by

$$\begin{aligned} a &= t_1 b + r_1 \ ; \quad 0 < r_1 < b \\ b &= t_2 r_1 + r_2 \ ; \quad 0 < r_2 < r_1 \\ r_1 &= t_3 r_2 + r_3 \ ; \quad 0 < r_3 < r_2 \\ &\ \ \vdots \\ r_k &= t_{k+2} r_{k+1} \ ; \end{aligned}$$

with $t_1, \ldots, t_{k+2}$, and $r_1, \ldots, r_{k+1}$ integral. It is a trivial matter to verify that r_{k+1} is, in fact, the greatest common divisor of a and b .

To illustrate the process consider the two integers 141 and 15. We have

$$141 = 9 \cdot 15 + 6$$
$$15 = 2 \cdot 6 + 3$$
$$6 = 2 \cdot 3 ,$$

from which we conclude that $(141, 15) = 3$.

The complexity of the problem is given by $\log_2 a + \log_2 b$, the number of bits required to store the two integers a and b . From a relationship like

$$r_{j-1} = t_{j+1} r_j + r_{j+1} \quad \text{with}$$

$r_{j-1} > r_j > r_{j+1}$, we conclude that $t_{j+1} \geq 1$, and therefore

$$r_{j-1} \geq r_j + r_{j+1} > 2r_{j+1} .$$

It follows that

$$\begin{aligned} &\log_2 r_{j-1} + \log_2 r_j \\ &\geq \log_2 2r_{j+1} + \log_2 r_j \\ &= \log_2 r_j + \log_2 r_{j+1} + 1 . \end{aligned}$$

Therefore the quantity $\log_2 r_{j-1} + \log_2 r_j$ decreases by at least one unit on every iteration of the Euclidean algorithm, from its initial value of $\log_2 a + \log_2 b$. Since, up to the last step, the r_j are all no less than one, the process must terminate in a number of steps no larger than $\log_2 a + \log_2 b$. The Euclidean algorithm is, indeed, a polynomial algorithm.

We shall see that this observation is central in demonstrating that our general methods lead to a polynomial algorithm for integer programming problems with two variables. An alternative polynomial algorithm for this problem had previously been developed by Kannan [Kannan, 1977].

VIII. Some Remarks on Polygons in the Plane

In order to demonstrate how some of our rather abstract ideas can be converted to practical algorithms we shall turn our attention to integer programming problems involving two integral activity levels and m inequalities. It will be necessary to review some elementary material on triangles and quadralaterials in the plane.

Let us begin by considering a triangle in the plane, whose three vertices h^0, h^1, h^2 are lattice points, and which contains no other lattice points in its interior or on the boundary. The area of such a triangle is given by 1/2 of the absolute value of either of the following determinants

$$\det\begin{vmatrix} 1 & 1 & 1 \\ h_1^0 & h_1^1 & h_1^2 \\ h_2^0 & h_2^1 & h_2^2 \end{vmatrix} \quad \text{or}$$

$$\det\begin{vmatrix} h_1^1 - h_1^0 & h_1^2 - h_1^0 \\ h_2^1 - h_2^0 & h_2^2 - h_2^0 \end{vmatrix} .$$

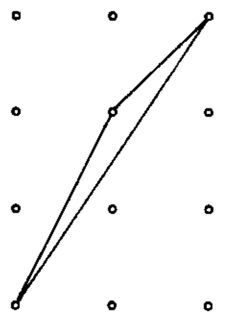


FIGURE 18

We have the following classical theorem.

8.1 [Theorem]. A necessary and sufficient condition that a triangle whose vertices are lattice points contain no other lattice points is that its area equal 1/2.

Consider a triangle whose vertices are lattice points and such that

$$\det \begin{vmatrix} 1 & 1 & 1 \\ h_1^0 & h_1^1 & h_1^2 \\ h_2^0 & h_2^1 & h_2^2 \end{vmatrix} = \pm 1 .$$

If a lattice point (h_1, h_2) is a convex combination of these three vertices then

$$\begin{pmatrix} 1 \\ h_0 \\ h_1 \end{pmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ h_1^0 & h_1^1 & h_1^2 \\ h_2^0 & h_2^1 & h_2^2 \end{bmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} ,$$

with $\alpha_i \geq 0$. But since the determinant is ± 1, the α_i's are integral.

They sum to 1 and therefore h must be one of the three vertices.

In order to argue the converse let us consider a triangle whose vertices are lattice points and whose area is larger than 1/2. We shall construct a fourth lattice point contained in the triangle.

Our argument will involve linear transformations $h' = Uh + k$ of the plane into itself, where k is an integral vector and U a unimodular matrix, i.e. a matrix with integral entries and determinant of ± 1. Such a transformation carries the lattice points in the plane onto themselves and preserves area.

By applying a translation we may assume that $h^0 = (0,0)$. But then h_1^1 and h_2^1 must be relatively prime, since if they had a common factor there would be a lattice point on the line connecting h^0 and h^1. It follows that there are integers p and q such that

$$p h_1^1 - q h_2^1 = 1 .$$

The linear transformation $h' = Uh$ where

$$U = \begin{bmatrix} p & -q \\ -h_2^1 & h_1^1 \end{bmatrix}$$

is unimodular, so that it is sufficient to consider the triangle whose three vertices are Uh^0, Uh^1, Uh^2 or

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} x \\ y \end{pmatrix} .$$

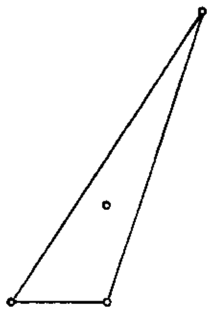


FIGURE 19

There is no loss in generality in assuming that $y > 0$, since this can be brought about by a unimodular reflection. We may also assume that $0 < x \leq y$ since this can be achieved by adding a suitable integral multiple of the second coordinate to the first coordinate of each of these vectors.

The area of this triangle is $y/2$; we therefore assume that $y \geq 2$. But the lattice point $(1,1)'$ is contained in such a triangle since

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{bmatrix} 0 & 1 & x \\ 0 & 0 & y \end{bmatrix} \begin{pmatrix} (x-1)/y \\ (y-x)/y \\ 1/y \end{pmatrix} .$$

This demonstrates Theorem 8.1.

In the course of this argument we have also verified that if the triangle (h^0, h^1, h^2) contains no other lattice point, then there is a unimodular transformation (including a translation) which brings this triangle to the form

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

If the 2nd coordinate is subtracted from the 1st—again a unimodular transformation—we see that such a triangle can be brought into the canonical form

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} .$$

Lemma 3.4 tells us that a convex polyhedron in the plane whose vertices are lattice points and which contains no other lattice points has either 3 or 4 vertices. We have, in the above argument, characterized such polyhedra with 3 vertices. Now let us consider one with 4 vertices h^0, h^1, h^2, h^3 .

Since the the triangle with vertices h^0, h^1, h^2 contains no other lattice points, there is a unimodular transformation which brings all four points to the form

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$$

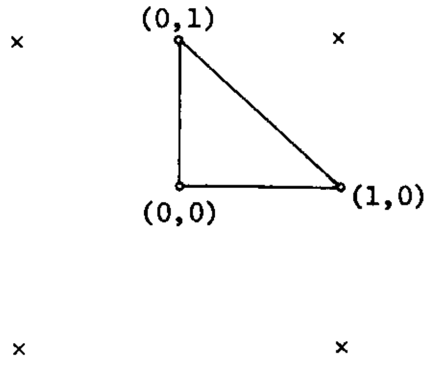


FIGURE 20

But the triangle formed by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$$

contains no additional lattice points; therefore

$$\det\begin{vmatrix} 1 & h_1 \\ 0 & h_2 \end{vmatrix} = h_2 = \pm 1 .$$

In a similar fashion $h_1 = \pm 1$, so that there are at most four possible locations for the point h, as illustrated in Figure 20. Of course, we cannot have $(h_1, h_2) = (-1,-1)$ since $(0,0)$ would not be a vertex of such a quadrilateral. The three possible quadrilaterals appear in Figure 21.

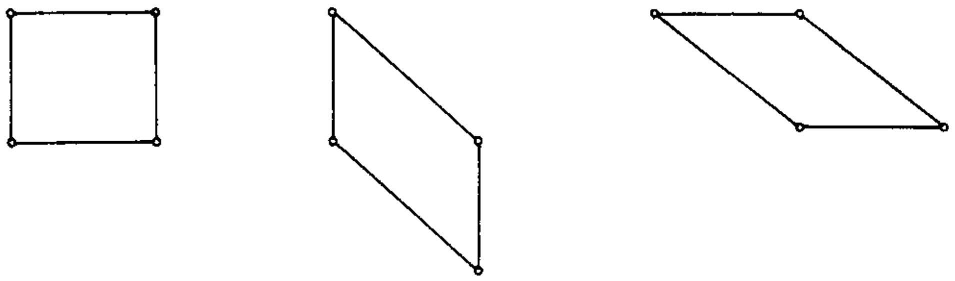


FIGURE 21

The first of these is the unit square, and the second, consisting of

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix},$$

can be brought to

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

by adding the 1st row to the 2nd. The third figure can be transformed by a similar unimodular transformation into the unit square.

8.2 [Theorem]. A convex polyhedron whose four vertices are lattice points, and which contains no other lattice points is equivalent to the unit square under a unimodular transformation.

Aside from translations the general example of such a quadrilateral is a regular parallelogram of unit area whose four vertices are given by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} p \\ q \end{pmatrix} \quad \begin{pmatrix} x-p \\ y-q \end{pmatrix} \quad \begin{pmatrix} x \\ y \end{pmatrix},$$

with x,y,p,q integers satisfying $py - qx = \pm 1$.

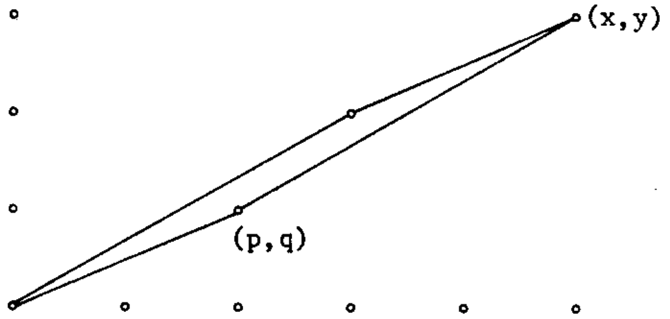


FIGURE 22

IX. Primitive Sets Associated with an Integer Program with Two Variables and Two Inequalities

In this section we shall give a complete description of the primitive sets which arise when the set X consists of all vectors of the form

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix},$$

where (h_1, h_2) ranges over the lattice points in the plane. To be specific we shall let the entries in A be integers. This causes some difficulty with the non-degeneracy assumption 1.3, since clearly two distinct vectors in X may have the same i^{th} coordinate. Any number of devices are available to resolve degeneracy; to be specific we adopt the following rule.

9.1 [Rule]. Let $x = (x_0, x_1, x_2)'$ be a vector in X with a zero coordinate. That coordinate will be considered positive if the vector x is lexicographically positive, negative if the vector is lexicographically negative, and zero if $x = 0$.

We note that this tie breaking rule has the property that if $x_i > y_i$ then $\alpha x_i + (1-\alpha)y_i > y_i$ for $0 < \alpha \leq 1$.

In order to begin our discussion of primitive sets let us assume that the entries in A have the following sign pattern

$$\begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix},$$

and in addition that $a_{i1} + a_{i2} > 0$ for $i = 1, 2$. The inequalities are to be interpreted according to 9.1 if necessary.

I claim that the following triple of activity levels,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

generates three vectors in X which form a primitive set. Let the three vectors be represented as columns in the following matrix, 9.2 $$\begin{bmatrix} 0 & a_{01} & \underline{a_{01}+a_{02}} \\ \underline{0} & a_{11} & a_{11}+a_{12} \\ 0 & \underline{a_{21}} & a_{21}+a_{22} \end{bmatrix} .$$

The sign pattern of A permits us to identify the row minima as indicated. Is there a vector

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$$

all of whose coordinates are strictly larger than these three row minima?

Consider the convex hull C of the four vectors

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} .$$

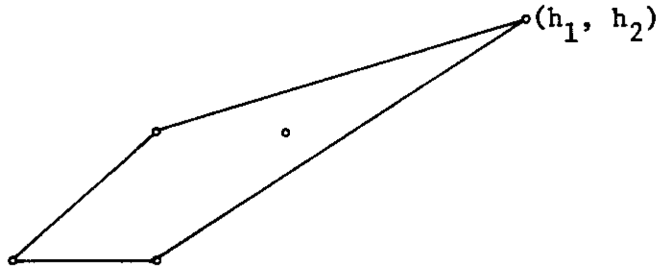


FIGURE 23

The vector (h_1, h_2) is clearly a vertex of the convex hull since it cannot be written as a convex combination of the remaining three vectors. But the first three vectors are also vertices of C . If for example

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \alpha_1 \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

with $\alpha_j \geq 0$, $\sum \alpha_j = 1$, then

$$0 = \alpha_1(a_{11}h_1 + a_{12}h_2) + \alpha_2 a_{11} + \alpha_3(a_{11} + a_{12}) .$$

But $a_{11}h_1 + a_{12}h_2$, a_{11} , and $a_{11} + a_{12}$ are all greater than 0 (at least in the lexicographic sense), which is impossible.

If there is a lattice point in C other than one of those four vertices it will also give rise to a vector in X which has all of its coordinates larger than the row minima of 9.2. It is therefore sufficient to assume that there are no lattice points in C other than the four vertices. It follows that h must be one of the following three vectors:

$$\begin{pmatrix} 0 \\ -1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

whose associated vectors in X are given by

$$\begin{pmatrix} -a_{02} \\ -a_{12} \\ -a_{22} \end{pmatrix} \quad \begin{pmatrix} a_{02} \\ a_{12} \\ a_{22} \end{pmatrix} \quad \begin{pmatrix} 2a_{01} + a_{02} \\ 2a_{11} + a_{12} \\ 2a_{21} + a_{22} \end{pmatrix} .$$

But $-a_{22} < a_{21}$, $a_{12} < 0$ and $2a_{01} + a_{02} < a_{01} + a_{02}$. This demonstrates that the three activity vectors

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

do indeed form a primitive set, and in precisely the same fashion so do

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

The translate of a primitive set by an integer vector is also a primitive set. This argues that every triangle in the simplicial subdivision of the plane illustrated in Figure 24 gives rise to a primitive set.

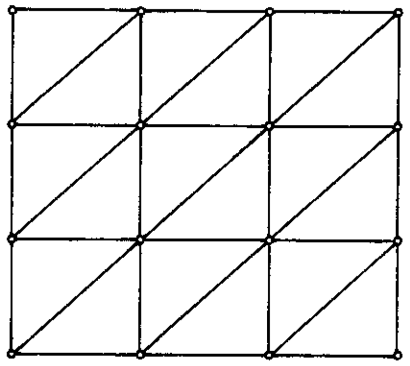


FIGURE 24

It is also easy to argue that these are the only primitive sets when the entries in the matrix A have the sign pattern we have assumed. One argument, based on Figure 24, is that a replacement for a vector in such a primitive set may obviously be found in such a way as to keep us within this class of primitive sets. The replacement operation is unique and therefore we never leave this class of primitive sets by an arbitrary series of replacements. If there were, however, a primitive set not in this class we would be in contradiction with the general observation that any two primitive sets can be connected by a sequence of replacement operations. We summarize our conclusions in the following theorem.

9.3 [Theorem]. Let the matrix A have the sign pattern

$$\begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix},$$

and assume that $a_{i1} + a_{i2} > 0$ for $i = 1, 2$. Then the collection of primitive sets is given by the simplicial subdivision of Figure 24.

Now let us turn our attention to determining the primitive sets associated with a general integral matrix A with 3 rows and 2 columns. In order to avoid the case in which all of the associated vectors in X lie on a single line in R^3 we shall assume that A has rank 2. We have the following Lemma which derives from Assumption 1.2: that for any vector c in R^3 the set of x in X with $x \geq c$ is finite.

9.4 [Lemma]. Let A have rank 2 and make Assumption 1.2 about the associated set of vectors $x = Ah$. Then there exists a positive vector π such that $\pi A = 0$. In other words all of the vectors in X lie on a plane through the origin with positive normals.

Assumption 1.2 implies that there is no lattice point h for which $Ah \geq 0$, with at least one inequality strict. If this were so the infinite set Ah, 2Ah, 3Ah, ... would violate Assumption 1.2 with $c = 0$. One consequence of this observation is that there is no real vector ξ with $A\xi > 0$, for if there were we could take a sufficiently high multiple of ξ and round to integers without disturbing the inequalities. But then by a standard convexity argument there is a non-negative, non-zero vector π such that $\pi A = 0$. In order to demonstrate Lemma 9.4 we must show that none of the coordinates of π are zero.

Suppose that $\pi_0 = 0$. Then rows 1 and 2 of A are linearly dependent. It follows that there are integers h_1 and h_2 not both zero such that

$$a_{11}h_1 + a_{12}h_2 = 0$$

$$a_{21}h_1 + a_{22}h_2 = 0 .$$

Since columns 1 and 2 are not linearly dependent it follows that $a_{01}h_1 + a_{02}h_2 \neq 0$. But then Ah (or $-Ah$) has all three coordinates non-negative, with at least one positive. This demonstrates Lemma 9.4.

The strategy in dealing with a general matrix A will be to show that there is a unimodular matrix

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} ,$$

such that $AU = B$ where B has the sign pattern

$$\begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix} ,$$

and $b_{i1} + b_{i2} > 0$ for $i = 1, 2$. Our previous result would then argue that

$$B\begin{pmatrix} 0 \\ 0 \end{pmatrix} , \quad B\begin{pmatrix} 1 \\ 0 \end{pmatrix} , \quad B\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and}$$

$$B\begin{pmatrix} 0 \\ 0 \end{pmatrix} , \quad B\begin{pmatrix} 0 \\ 1 \end{pmatrix} , \quad B\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{are}$$

both primitive sets. In terms of the matrix A this implies that

$$AU\begin{pmatrix}0\\0\end{pmatrix}, \quad AU\begin{pmatrix}1\\0\end{pmatrix}, \quad AU\begin{pmatrix}1\\1\end{pmatrix}, \quad \text{and}$$

$$AU\begin{pmatrix}0\\0\end{pmatrix}, \quad AU\begin{pmatrix}0\\1\end{pmatrix}, \quad AU\begin{pmatrix}1\\1\end{pmatrix}$$

are primitive sets. Primitive sets therefore form a simplicial subdivision of the plane based on the parallelogram

$$\begin{pmatrix}0\\0\end{pmatrix} \begin{pmatrix}u_{11}\\u_{21}\end{pmatrix} \begin{pmatrix}u_{12}\\u_{22}\end{pmatrix} \begin{pmatrix}u_{11}+u_{12}\\u_{21}+u_{22}\end{pmatrix},$$

cut into two triangles along the diagonal between $(0,0)'$ and $(u_{11}+u_{12}, u_{21}+u_{22})'$.

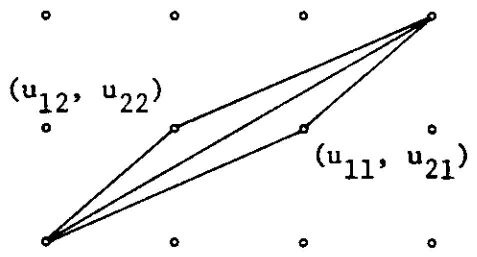


FIGURE 25

Of course, multiplication of the matrix A on the right by a unimodular matrix is equivalent to a series of elementary column operations --adding a multiple of one column to the other, and changing the sign of a column.

Let us begin by assuming that $a_{01} \geq a_{02} \geq 0$, and apply the Euclidean algorithm for determining the greatest common divisor of these two integers. We write $a_{01} = ta_{02} + r$ with $0 \leq r < a_{02}$, subtract t times the second column from the first, and continue until one of the entries in the top row is zero and the other equal to g.c.d.(a_{01}, a_{02}) . The negative of the resulting matrix will have the sign pattern

$$\begin{bmatrix} - & 0 \\ & \\ & \end{bmatrix},$$

and the two entries in the second column will be non-zero and of opposite sign. By multiplying this second column by -1 if necessary we reach a matrix B for which the following sign pattern prevails:

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

If $b_{21} \leq 0$ then since $\pi B = 0$, we have $b_{11} > 0$, and the sign pattern of the first column is as desired. On the other hand if $b_{21} > 0$, we find a positive integer t such that

$$0 \geq b_{21} - tb_{22} > -b_{22} .$$

If we then subtract t times column 2 from column 1 the matrix B will have the sign pattern

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

Of course, it need not yet be true that $b_{i1} + b_{i2} > 0$ for $i = 1, 2$. In order to obtain this it may be necessary to continue with the Euclidean algorithm on the last two rows of the matrix.

For example if $b_{11} + b_{12} < 0$, or $b_{11} < -b_{12}$ we initiate the Euclidean algorithm for the two positive integers b_{11} and $-b_{12}$. We write $-b_{12} = tb_{11} + r$ with $0 \leq r < b_{11}$. If we add t times column one to column two we obtain the matrix

$$\begin{bmatrix} - & - \\ + & -r \\ - & + \end{bmatrix} .$$

(The third entry in column two is, of course, positive from $\pi B = 0$.) In this new matrix $b_{11} + b_{12} > 0$. If, however, $b_{21} + b_{22} < 0$, or $b_{22} < -b_{21}$ we use the Euclidean algorithm for finding the greatest common divisor of these two numbers. The operation of the Euclidean algorithm for one of these two rows only facilitates the algorithm for the other row. At the worst we complete the algorithm for both rows and obtain

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

9.5 [Theorem]. Let A be a 3x2 matrix of integers of rank 2. Assume that there exists a positive vector π with $\pi A = 0$. Then the matrix can be transformed by a number of elementary column operations, into the form

$$AU = B = \begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix} ,$$

with $b_{i1} + b_{i2} > 0$ for $i = 1, 2$. Primitive sets are given by translations of one of the two triangles

$$\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad \begin{Bmatrix} u_{11} \\ u_{21} \end{Bmatrix} \quad \begin{Bmatrix} u_{11}+u_{12} \\ u_{21}+u_{22} \end{Bmatrix}, \quad \text{and}$$

$$\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad \begin{Bmatrix} u_{12} \\ u_{22} \end{Bmatrix} \quad \begin{Bmatrix} u_{11}+u_{12} \\ u_{21}+u_{22} \end{Bmatrix}.$$

Let us investigate the complexity of the matrix B by a careful examination of the process which is used to obtain this final matrix. The first part of the procedure carries out the Euclidean algorithm for finding the greatest common divisor of a_{01} and a_{02}. In no more than $\log_2 a_{01} + \log_2 a_{02}$ steps we reach a matrix with sign pattern

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

At every intermediary step we will have a matrix

$$B = \begin{bmatrix} b_{01} & b_{02} \\ b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

with, say, $b_{01} \geq b_{02} > 0$. We then write $b_{01} = tb_{02} + r$ with $t \geq 1$, $0 \leq r < b_{02}$ and transform the above matrix to

$$B' = \begin{bmatrix} b_{01} - tb_{02} & b_{02} \\ b_{11} - tb_{12} & b_{12} \\ b_{21} - tb_{22} & b_{22} \end{bmatrix}.$$

There are a variety of equivalent measures that can be used to estimate the complexity of the matrices encountered in this sequence. It will be convenient in analyzing the first part of the procedure to adopt the following expression

9.6 $$C(B) = \log_2 \max(|b_{01}|, |b_{02}|) + \frac{1}{2}\sum_{i=1}^{2} \log_2 \max(|b_{i1}|, |b_{i2}|) ,$$

as a measure of complexity rather than the more obvious choice $\max\limits_{i,j} \log_2|b_{ij}|$. We shall demonstrate

$$C(B') \leq C(B) + 1 .$$

It is clearly sufficient to show that

9.7 $$\log_2 \max(|b_{01}|, |b_{02}|) + \log_2 \max(|b_{11}|, |b_{12}|)$$

and

$$\log_2 \max(|b_{01}|, |b_{02}|) + \log_2 \max(|b_{21}|, |b_{22}|)$$

each increase by no more than one unit in passing from B to B' . Since the two arguments are identical we concentrate on the first of these.

The expression 9.7 changes from $\log_2 b_{01} + \log_2 \max(|b_{11}|, |b_{12}|)$ to $\log_2 b_{02} + \log_2 \max(|b_{11} - tb_{12}|, |b_{12}|)$. But $|b_{11} - tb_{12}| \leq |b_{11}| + t|b_{12}| \leq (t+1)\max(|b_{11}|, |b_{12}|)$, and certainly

$$\max(|b_{11} - tb_{12}|, |b_{12}|) \leq (t+1)\max(|b_{11}|, |b_{12}|) .$$

It follows that

$$\log_2 \max(|b_{11} - tb_{12}|, |b_{12}|) - \log_2 \max(|b_{11}|, |b_{12}|) \leq \log_2(t+1) .$$

On the other hand

$$\log_2 b_{02} - \log_2 b_{01} = \log_2(b_{02}/(tb_{02} + r)) \leq \log_2(1/t) .$$

The expression 9.7 changes therefore by an amount which is no larger than $\log_2(t+1)/t \leq 1$. This concludes our argument.

It follows that the complexity of the matrix obtained after the Euclidean algorithm terminates is less than or equal to

$$\log_2 a_{01} + \log_2 a_{02} + C(A) .$$

The Euclidean algorithm terminates with a matrix whose sign pattern is

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

We then subtract from the first column a non-negative integral multiple of the second column; in fact the smallest such multiple such that the resulting matrix, say B^* , has the sign pattern

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

Using virtually identical arguments as before, it may be shown that the measure of complexity C increases by no more than a single additional unit on this last transition. It follows that

$$C(B^*) \leq 1 + \log_2 a_{01} + \log_2 a_{02} + C(A) \ .$$

At this point we enter into a series of elementary operations which are designed to maintain the above sign pattern and ultimately to yield $b_{i1} + b_{i2} > 0$ for $i = 1, 2$. Assume for example that $b_{11} + b_{12} \leq 0$. We write $-b_{12} = tb_{11} + r$ with $t \geq 1$ and $0 \leq r < b_{11}$, and add t times column one to column 2:

$$\begin{bmatrix} b_{01} & tb_{01} + b_{02} \\ b_{11} & tb_{11} + b_{12} \\ b_{21} & tb_{21} + b_{22} \end{bmatrix} .$$

If $b_{21} + (tb_{21} + b_{22}) \leq 0$ we perform a simular step based on row 2 of this matrix. In other words we alternate between steps of the Euclidean algorithm for finding g.c.d. (b^*_{11}, b^*_{12}) and g.c.d. (b^*_{21}, b^*_{22}) . Since $\log_2|b^*_{i1}| + \log_2|b^*_{i2}|$ have already been shown to be bounded by linear functions of the complexity of the original matrix A , it follows that at most a polynomial number of iterations are required to bring the matrix to its final form.

We wish to verify that the complexity of the final matrix is bounded by a polynomial in the complexity of A .

Let us examine the change in $\log_2 \max(|b_{i1}|, |b_{i2}|)$ for each row separately, as we perform a typical elementary operation described above.

$i = 0$. We have

$$\log_2 \max(|b_{01}|, |tb_{01} + b_{02}|)$$
$$\leq \log_2 \max(|b_{01}|, (1+t)\max(|b_{01}|, |b_{02}|)$$
$$= \log_2(1+t) + \log_2 \max(|b_{01}|, |b_{02}|) ,$$

so that $\log_2 \max(|b_{01}|, |b_{02}|)$ increases by at most $\log_2(t+1)$.

$i = 1$. Since $b_{11} + b_{12} \leq 0$ it follows that $\max(|b_{11}|, |b_{12}|) = |b_{12}|$. On the other hand, $\max(|b_{11}|, |tb_{11}+b_{12}|) = b_{11}$ and therefore $\log_2 \max(|b_{11}|, |b_{12}|)$ increases by $\log_2 b_{11} - \log_2|b_{12}| = \log_2 b_{11}/|b_{12}|$ $\leq \log_2 1/t$.

$i = 2$. Since $|tb_{21}+b_{22}| \leq b_{22}$ it follows that

$$\log_2 \max(|b_{21}|, |tb_{21}+b_{22}|) \leq \log_2 \max(|b_{21}|, |b_{22}|) .$$

We see from this argument that the measure of complexity

$$\sum_{i=1}^{2} \log_2 \max(|b_{i1}|, |b_{i2}|)$$

increases by no more than one unit on each iteration. This demonstrates the following Theorem.

9.8 [Theorem]. The number of elementary operations required in Theorem 9.5, and the complexity of the final matrix B are both bounded by polynomial functions of the complexity of A .

This conclusion permits us to use the matrices A and B interchangeably in any discussion of polynomial algorithms.

X. The Algorithm for Integer Programs with Two Variables and Two Inequalities

In this section we shall show how the above considerations can be used to provide a polynomial algorithm for solving the integer program

$$\max a_{01}h_1 + a_{02}h_2$$

$$a_{11}h_1 + a_{12}h_2 \geq b_1$$

$$a_{21}h_1 + a_{22}h_2 \geq b_2 ,$$

with h_1, h_2 arbitrary integers. We continue to make the assumptions that A is of rank 2, and that there exists a positive vector π with $\pi A = 0$. The considerations of the previous section imply that there is no loss in generality in assuming that A has the sign pattern

$$\begin{bmatrix} - & - \\ + & - \\ - & + \end{bmatrix} ,$$

with $a_{i1} + a_{i2} > 0$ for $i = 1, 2$.

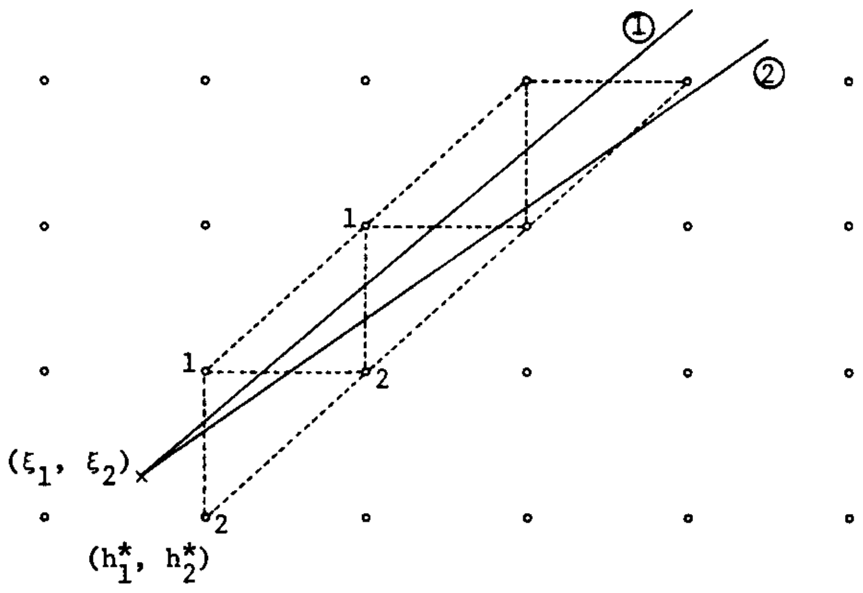


FIGURE 25

Figure 25 illustrates the constraint set for such a problem. We denote by (ξ_1, ξ_2) the typically non-integral solution of

$a_{i1}\xi_1 + a_{i2}\xi_2 = b_i$ for $i = 1, 2$, and by h_1^*, the smallest integer greater than or equal to ξ_1. We move up the line $h_1 = h_1^*$ until we find the largest integer h_2^* so that (h_1^*, h_2^*) violates the second inequality. Clearly (h_1^*, h_2^*) satisfies the first inequality since points which violate both inequalities have both coordinates less than those of ξ.

It is trivial to argue that (h_1^*, h_2^*+1) is the optimal solution to the programming problem if it also satisfies inequality number 1. We therefore assume that this is not the case; the two points (h_1^*, h_2^*+1) and (h_1^*, h_2^*) therefore have the labels 1 and 2 respectively.

The three points

$$\begin{pmatrix} h_1^* \\ h_2^* \end{pmatrix} \quad \begin{pmatrix} h_1^* \\ h_2^*+1 \end{pmatrix} \quad \begin{pmatrix} h_1^*+1 \\ h_2^*+1 \end{pmatrix}$$

form a primitive set. If the third point satisfies both inequalities it optimal since we would have a completely labeled primitive set. This third point clearly satisfies the first inequality since

$$a_{11}(h_1^*+1) + a_{12}(h_2^*+1) > a_{11}h_1^* + a_{12}h_2^* \geq b_1 .$$

If it is not optimal it must violate the second inequality and receive the label 2, as indicated.

We move to an adjacent primitive set by replacing (h_1^*, h_2^*) by (h_1^*+1, h_2^*+2). This new point will be optimal if it satisfies both inequalities. It definitely satisfies the second inequality since

$$a_{21}(h_1^*+1) + a_{22}(h_2^*+2) > a_{21}h_1^* + a_{22}(h_2^*+1) \geq b_2 .$$

If it is not optimal it must receive the label 1 as indicated, and the process continues by removing the vector (h_1^*, h_2^*+1) .

The process moves through a sequence of almost completely labeled primitive sets and terminates when a vector satisfying both inequalities is introduced. But we need not actually carry out the sequence of replacement steps, since the vectors being brought in lie on one of the two parallel lines:

$$(h_1, h_2) = (h_1^*+t, h_2^*+t) \quad \text{and}$$

$$(h_1, h_2) = (h_1^*+t, h_2^*+1+t) \ .$$

It is sufficient to examine each of these two lines, find the first lattice point satisfying both inequalities--simply by division--and compare the two values of the objective function.

We see that a fixed number of arithmetic operations is required to calculate the optimal solution, after the matrix has been brought into the correct form. This demonstrates that our methods provide a polynomial algorithm for the integer programming problem with 2 variables and 2 inequalities.

XI. Primitive Sets Associated with an Integer Program with Two Variables and Three Inequalities

Theorem 3.3 tells us that the solution of the integer programming problem

$$\begin{aligned} &\max a_{01}h_1 + a_{02}h_2 \\ &\quad a_{11}h_1 + a_{12}h_2 \geq b_1 \\ &\quad\quad\quad\vdots \\ &\quad a_{m1}h_1 + a_{m2}h_2 \geq b_m , \end{aligned}$$

and h_1 and h_2 integral, may be found by solving a suitable subproblem obtained by selecting 3 or fewer of the m inequalities. The number of subsets of 3 inequalities is $\binom{m}{3}$ which is polynomial in the data of the problem. If we can show that the integer program with 3 inequalities has a polynomial algorithm, then the rather inefficient algorithm which solves all of the problems obtained by selecting subsets of 3 inequalities, and checking to see whether the remaining inequalities are satisfied, will be polynomial in the data. For this reason we shall concentrate on matrices A which have 4 rows and 2 columns:

$$\begin{bmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} .$$

Let us consider a primitive set composed of four vectors h^0, h^1, h^2, h^3 in the plane. Each of these four vectors will be an extreme point in the convex hull of the four points, and, of course, there will be no other lattice points in this convex hull. It follows that the four vectors are the vertices of a parallelogram of unit area.

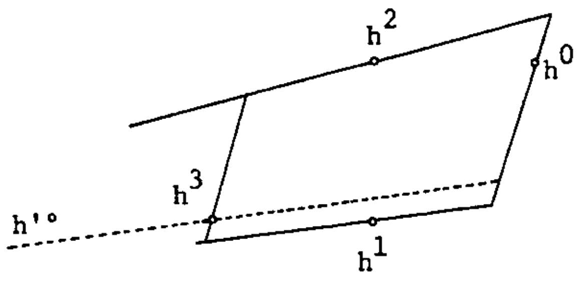


FIGURE 26

Assume that the line through h^i corresponds to the i^{th} row of the matrix A . In Figure 26, h^1 and h^2 are on opposite sides of the parallelogram, as are h^0 and h^3 . In attempting to remove, for example h^1 , we press in the side containing h^1 until we reach one of the other three vectors. The first vector to be reached will always be h^0 or h^3 ; in Figure 26 it is h^3 .

The replacement for h^1 will either be the 3rd slack vector, or a lattice point which forms a parallelogram of unit area in conjunction with h^0, h^2, h^3 . But this can only be the vector $h' = 2h^3 - h^1$, which is the replacement if and only if it satisfies the second inequality, i.e.

$$a_{21}h_1^2 + a_{22}h_2^2 < a_{21}h_1' + a_{22}h_2' .$$

Since $h' - h^2 = h^3 - h^0$ this is equivalent to

$$a_{21}h_1^0 + a_{22}h_2^0 < a_{21}h_1^3 + a_{22}h_2^3 .$$

But this means that if we press in the line through h^2 , as the first step in removing h^2 , we will reach h^0 before h^3 . The replacement for h^2 will therefore be $2h^0 - h^2$.

The original parallelogram and the two new parallelograms obtained by replacing h^1 and h^2 are shown in Figure 27. Of course the two

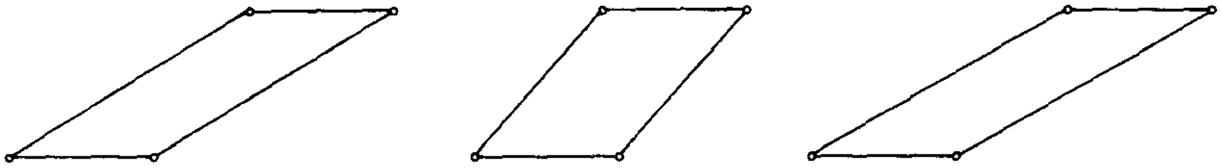


FIGURE 27

new parallelograms are always congruent.

In Figure 26, if we press in the side containing h^3 we first reach h^2 ; whereas if we press in the side containing h^0 we first reach h^1 . It follows that the replacement for h^3 is $2h^2 - h^3$, and the replacement for h^0 is $2h^1 - h^0$. The two new congruent parallelograms obtained by this replacement operation are drawn in Figure 28.

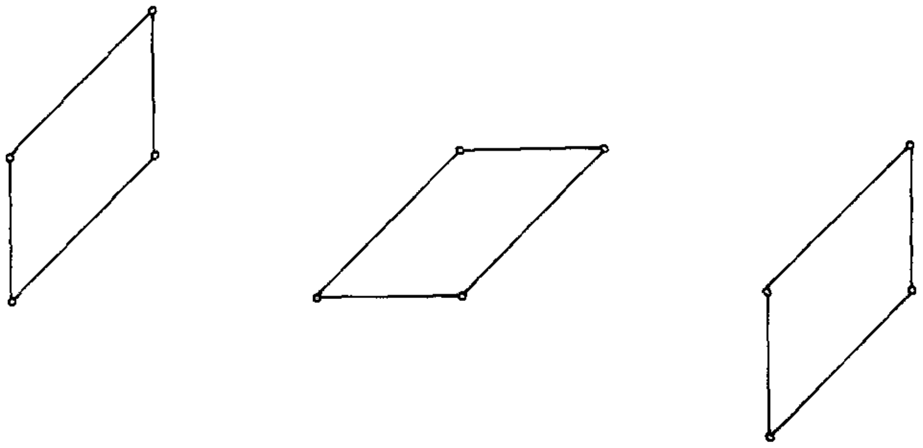


FIGURE 28

If, as in Figure 29, the same vector, say h^0 , is first reached by pressing in the line through h^1 <u>and</u> the line through h^2 , then the replacement for either one of them is the zeroth slack vector. In this case the chain of quadrilaterals will end with the two triangles

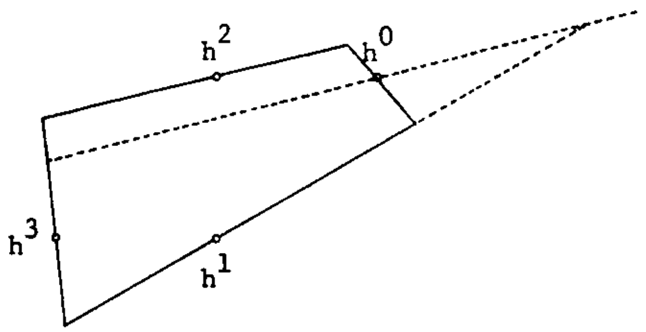


FIGURE 29

(h^0, h^2, h^3) and (h^0, h^1, h^3) both of which form primitive sets, in conjunction with the zeroth slack vector. We formalize this result in the following theorem.

11.1 [Theorem]. Let h^0, h^1, h^2, h^3 form a primitive set with the line through h^i corresponding to the i^{th} row of the matrix A. Let h^0 and h^3 be on opposite sides of the quadrilateral, so that $h^0 + h^3 = h^1 + h^2$.

1. Assume that when we press in the line through h^1 we first hit h^3 and when we press in the line through h^2 we first reach h^0. Then the replacement for h^1 is $2h^3 - h^1$ and the replacement for h^2 is $2h^0 - h^2$.

2. If on the other hand we reach the same point; say h^0, when pressing in the lines through h^1 <u>and</u> h^2, then the replacement for either h^1 or h^2 is the zeroth slack vector.

Theorem 11.1 tells us that the primitive sets with 4 vectors associated with a given 4x2 matrix A will form a chain of quadrilaterals. Each quadrilateral will be obtained from the previous one by the reflection of a given vertex about an adjacent vertex. The chain of quadrilaterals will terminate, on either side, when a pair of primitive sets is reached each consisting of a triple of vectors and the same slack vector.

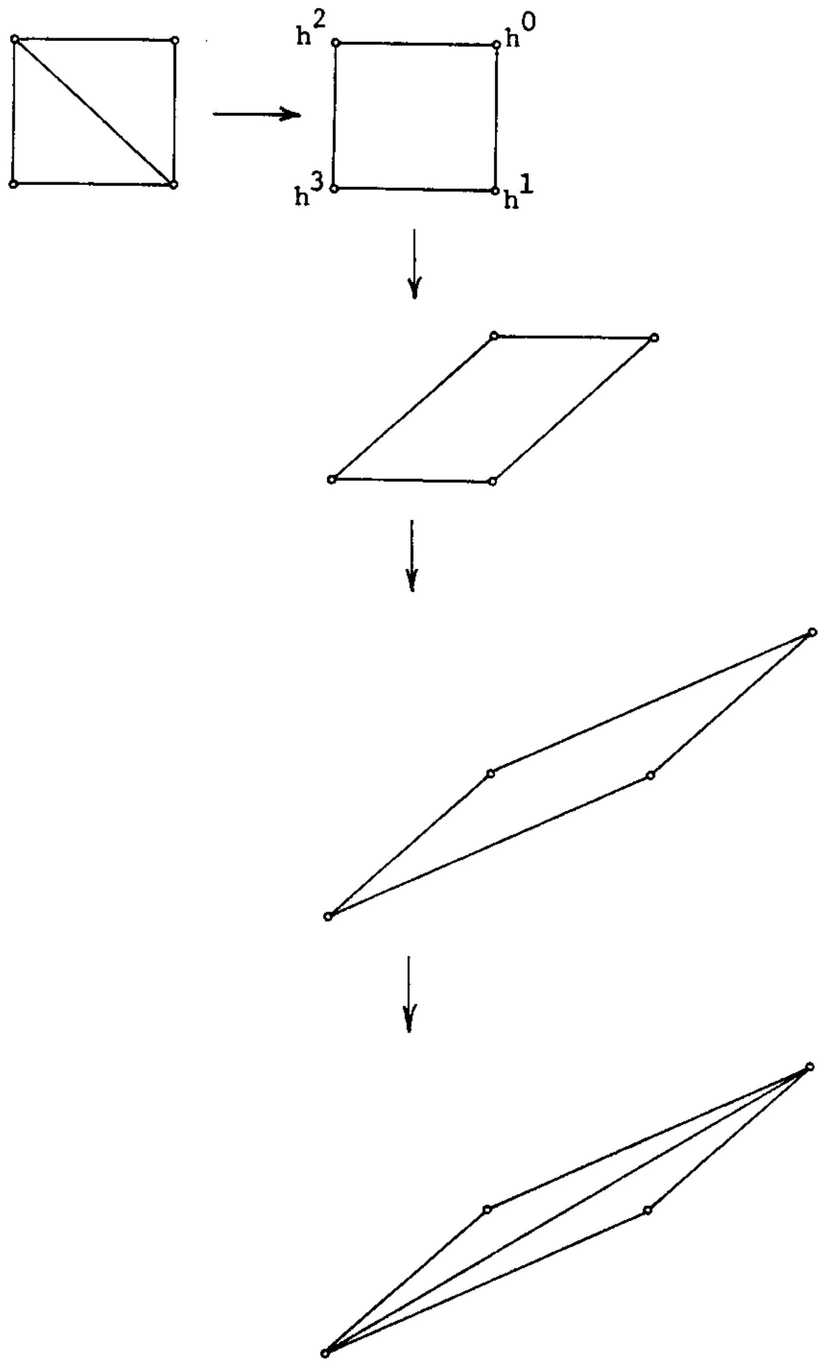


FIGURE 30

There is a very simple test for determining whether the four vertices of a parallelogram of unit area form a primitive set for the matrix A . Consider, for example, the unit square with vertices h^0, h^1, h^2, h^3 as in Figure 31. The four associated vectors in X are given by the

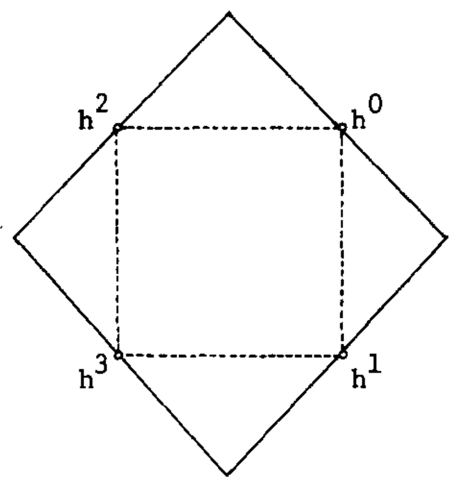


FIGURE 31

columns of the matrix

$$\begin{array}{cccc} h^0 & h^1 & h^2 & h^3 \\ \left[\begin{array}{c} \underline{a_{01}+a_{02}} \\ a_{11}+a_{12} \\ a_{21}+a_{22} \\ a_{31}+a_{32} \end{array}\right. & \begin{array}{c} a_{01} \\ \underline{a_{11}} \\ a_{21} \\ a_{31} \end{array} & \begin{array}{c} a_{02} \\ a_{12} \\ \underline{a_{22}} \\ a_{32} \end{array} & \left.\begin{array}{c} 0 \\ 0 \\ 0 \\ \underline{0} \end{array}\right] \end{array} .$$

If the four lines corresponding to the rows of A can be drawn as in Figure 31, then the four row minima lie in different columns; let us assume that they lie along the main diagonal. But conversely, if the four row minima lie in different columns the vertices of the unit square will form a primitive set, since there are obviously no other lattice points in the larger quadrilateral of Figure 31.

The particular location of the row minima in the above matrix is equivalent to the sign pattern

$$(11.2) \qquad \begin{bmatrix} - & - \\ - & + \\ + & - \\ + & + \end{bmatrix}$$

in the matrix A . More generally if there is a unimodular transformation of the matrix A , multiplying on the right by a unimodular matrix U , so that AU has the above sign pattern, then the quadrilateral with vertices

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} \quad \begin{pmatrix} u_{12} \\ u_{22} \end{pmatrix} \quad \begin{pmatrix} u_{11}+u_{12} \\ u_{21}+u_{22} \end{pmatrix}$$

will form a primitive set.

Let us consider a chain of quadrilaterals as depicted in Figure 30, with the initial quadrilateral being the unit square. Without loss of generality we assume that the row minima of

$$\begin{array}{cccc} \left[\begin{array}{cccc} \underline{a_{01}+a_{02}} & a_{01} & a_{02} & 0 \\ \utilde{a_{11}+a_{12}} & \underline{a_{11}} & a_{12} & 0 \\ a_{21}+a_{22} & a_{21} & \underline{a_{22}} & \utilde{0} \\ a_{31}+a_{32} & a_{31} & a_{32} & \underline{0} \end{array}\right] \\ h^0 \quad\quad h^1 \quad\quad h^2 \quad h^3 \end{array}$$

lie on the main diagonal so that the sign pattern of A is as 11.2. In this example if we press in the line through h^2 we first reach h^0 ; if we press in the line through h^1 we first reach h^3 . It follows that the second minima in rows 1 and 2 of the above matrix are as indicated. This is, of course, equivalent to saying that

$$a_{11} + a_{12} > 0$$

$$a_{21} + a_{22} < 0 \ .$$

The replacement for h^1 is $2h^0 - h^1$ and the resulting matrix is given by

$$\begin{bmatrix} \underline{a_{01} + 2a_{02}} & a_{01} + a_{02} & a_{02} & 0 \\ a_{11} + 2a_{12} & \underline{a_{11} + a_{12}} & a_{12} & 0 \\ a_{21} + 2a_{22} & a_{21} + a_{22} & \underline{a_{22}} & 0 \\ a_{31} + 2a_{32} & a_{31} + a_{32} & a_{32} & \underline{0} \end{bmatrix} .$$

We can associate with this new primitive set the matrix

$$\begin{bmatrix} a_{01} + a_{02} & a_{02} \\ a_{11} + a_{12} & a_{12} \\ a_{21} + a_{22} & a_{22} \\ a_{31} + a_{32} & a_{32} \end{bmatrix} ,$$

which will also have precisely the same sign pattern 11.2. This process can then be continued to reach the full chain of quadrilaterals. At each stage we have a matrix, say B, obtained from A by a unimodular transformation, and with the sign pattern 11.2. If

$$b_{11} + b_{12} < 0$$

$$b_{21} + b_{22} > 0$$

the next quadrilateral in the chain is associated with

$$\begin{bmatrix} b_{01}+b_{02} & b_{02} \\ b_{11}+b_{12} & b_{12} \\ b_{21}+b_{22} & b_{22} \\ b_{31}+b_{32} & b_{32} \end{bmatrix} ;$$

if

$$b_{11} + b_{12} > 0$$

$$b_{21} + b_{22} < 0 ,$$

the new matrix is

$$\begin{bmatrix} b_{01} & b_{01}+b_{02} \\ b_{11} & b_{11}+b_{12} \\ b_{21} & b_{21}+b_{22} \\ b_{31} & b_{31}+b_{32} \end{bmatrix} .$$

Finally if both $b_{11} + b_{12}$ and $b_{21} + b_{22}$ have the same sign the sequence of quadrilaterals terminates. We summarize these observations as follows.

11.3 [Theorem]. Let the four vertices of the unit square form a primitive set, so that the matrix A has the sign pattern

$$\begin{bmatrix} - & - \\ - & + \\ + & - \\ + & + \end{bmatrix} .$$

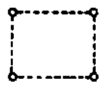
Perform a sequence of elementary column operations, replacing one of these columns by the sum of the two columns, so that the sign pattern is maintained. Each such matrix corresponds to a primitive set in the chain of quadrilaterals. The chain terminates when this operation can no longer be continued.

An example may be useful in illustrating these considerations. Let

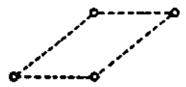
$$A = \begin{bmatrix} -4 & -1 \\ -2 & 5 \\ 1 & -2 \\ 1 & 1 \end{bmatrix},$$

with ties broken by the lexicographic rule. This matrix has the appropriate sign pattern so that the four vertices of the unit square form a primitive set. The following list of matrices illustrates the sequence followed in Theorem 11.3.

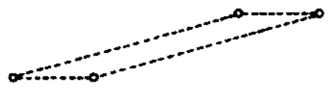
$$\begin{bmatrix} -4 & -1 \\ -2 & 5 \\ 1 & -2 \\ 1 & 1 \end{bmatrix}$$



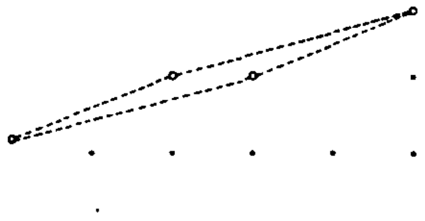
$$\begin{bmatrix} -4 & -5 \\ -2 & 3 \\ 1 & -1 \\ 1 & 2 \end{bmatrix}$$



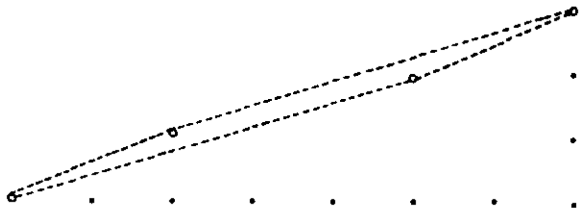
$$\begin{bmatrix} -4 & -9 \\ -2 & 1 \\ 1 & 0 \\ 1 & 3 \end{bmatrix}$$



$$\begin{bmatrix} -13 & -9 \\ -1 & 1 \\ 1 & 0 \\ 4 & 3 \end{bmatrix}$$



$$\begin{bmatrix} -22 & -9 \\ 0 & 1 \\ 1 & 0 \\ 7 & 3 \end{bmatrix}$$



Is there a calculation, which is polynomial in the data of the problem and which permits us to recognize the chain of quadrilaterals and the two simplicial subdivisions which appear at either end of the chain? The answer is quite immediate if two rows in the original matrix A are proportional to each other. In order to consider the more difficult and general case I shall assume that no two rows are proportional.

The four rows of A are selected from a larger problem satisfying the assumption of local finiteness 1.2. If these four rows yield primitive sets in conjunction with the remaining slack vectors then assumption 1.2 will apply just as well to the vectors in R^4 given by $x = Ah$. We may then use the argument of Section IX to conclude that there is a non-negative vector $\pi = (\pi_0, \pi_1, \pi_2, \pi_3)$, not all of whose coordinates are 0 , such that $\pi A = 0$. At least 3 of the coordinates of π are strictly positive, since otherwise two of the rows of A would be proportional.

If all four of the coordinates of π are strictly positive then, since the system $\pi A = 0$ consists of two homogeneous equations in four variables, there is a standard way of finding an alternative solution with three positive components. Let us assume that this has been done and that π_0, π_1, π_2 are positive and $\pi_3 = 0$.

We may then use the arguments of Theorem 9.5 to construct, in polynomial time, a unimodular matrix U , such that $B = AU$ has the following sign pattern

$$\text{(11.6)} \qquad \begin{bmatrix} - & - \\ - & + \\ + & - \\ b_{31} & b_{32} \end{bmatrix} .$$

Moreover $b_{11} + b_{12} > 0$, and $b_{21} + b_{22} > 0$, and the arguments of Theorem 9.5 may be used to demonstrate that the complexity of B is bounded by a polynomial function of the complexity of A .

Let us consider the following four cases for the sign patterns of the last row of B .

Case I. $b_{31} > 0$, $b_{32} > 0$.

In this case the matrix 11.6 has the appropriate sign pattern which permits us to identify a primitive set. The columns of

$$\begin{array}{c} \begin{bmatrix} \underline{b_{01}+b_{02}} & b_{01} & b_{02} & 0 \\ b_{11}+b_{12} & \underline{b_{11}} & b_{12} & 0 \\ b_{21}+b_{22} & b_{21} & \underline{b_{22}} & 0 \\ b_{31}+b_{32} & b_{31} & b_{32} & \underline{0} \end{bmatrix} \\ \begin{array}{cccc} h^0 \qquad & h^1 & h^2 & h^3 \end{array} \end{array}$$

will represent the final primitive set in the chain of quadrilaterals, which is then followed by the simplicial subdivision given by the two triangles (h^0, h^1, h^3) and (h^0, h^2, h^3) in conjunction with the third slack vector.

Case II. $b_{31} < 0$, $b_{32} < 0$.

In this case the arguments of Section IX permit us to conclude that

$$(h^0, h^1, h^3, \xi^3), \quad (h^0, h^2, h^3, \xi^3) \quad \text{and}$$

$$(h^0, h^1, h^3, \xi^0), \quad (h^0, h^2, h^3, \xi^0)$$

are <u>all</u> primitive sets. Since the replacement operation keeps us--subject to translations--in this class of primitive sets, it follows that there are no quadrilaterals associated with this 4x2 matrix. All of the primitive sets consist of triangles in conjunction with the zeroth or third slack vector.

Case III. $b_{31} > 0$, $b_{32} < 0$.

This case falls into two subcases depending on the sign of $b_{31} + b_{32}$. If $b_{31} + b_{32} > 0$, then the matrix

$$\begin{bmatrix} -b_{01} - b_{02} & b_{02} \\ -b_{11} - b_{12} & b_{12} \\ -b_{21} - b_{22} & b_{22} \\ -b_{31} - b_{32} & b_{32} \end{bmatrix}$$

has the sign pattern

$$\begin{bmatrix} + & - \\ - & + \\ - & - \\ - & - \end{bmatrix},$$

with the sums of rows zero and one positive. The sign pattern is as in Case II, and there are no quadrilateral primitive sets.

On the other hand if $b_{31} + b_{32} < 0$ then the matrix

$$\begin{bmatrix} -b_{01} - b_{02} & b_{01} \\ -b_{11} - b_{12} & b_{11} \\ -b_{21} - b_{22} & b_{21} \\ -b_{31} - b_{32} & b_{31} \end{bmatrix}$$

has the sign pattern

$$\begin{bmatrix} + & - \\ - & - \\ - & + \\ + & + \end{bmatrix},$$

with the sums of rows zero and two positive. The sign pattern is as in Case I.

Case IV. $b_{31} < 0$, $b_{32} > 0$.

We use the same arguments as in Case III.

We see that in all cases in which there is a chain of quadrilateral primitive sets, the above method will produce a matrix which represents a quadrilateral primitive set appearing at one end of the chain. Moreover the complexity of this matrix will be bounded by a polynomial function of the complexity of the original matrix A . To find the matrix at the other end of the chain let us assume that the sign pattern of the matrix already determined is

$$\begin{bmatrix} - & - \\ - & + \\ + & - \\ + & + \end{bmatrix} .$$

If we multiply the first column by -1, the sign pattern becomes

$$\begin{bmatrix} + & - \\ + & + \\ - & - \\ - & + \end{bmatrix} .$$

If $b_{01} + b_{02}$ and $b_{31} + b_{32}$ have the same sign this matrix represents the quadrilateral primitive set at the opposite end of the chain, which in this case contains only one such primitive set. On the other hand if the signs are opposite we replace one of these columns by the sum of the two columns, in such a way that the sign pattern is maintained, and continue until a primitive set yielding the opposite end of the chain is obtained.

One additional observation may be useful. The operation of replacing a column by the sum of the two columns is very slow when compared to the Euclidean algorithm and may lead to a chain of quadrilaterals--and therefore a number of neighbors of the origin--which is quite large in terms of the complexity of the problem. Consider, for example, the matrix

$$\begin{bmatrix} -1 & -1 \\ -1 & k \\ 1 & -k \\ 1 & 1 \end{bmatrix},$$

with k a large positive integer. Since

$$\begin{bmatrix} -1 & -1 \\ -1 & k \\ 1 & -k \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & j \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -1-j \\ -1 & k-j \\ 1 & -k+j \\ 1 & 1+j \end{bmatrix},$$

with the appropriate sign pattern for all $0 \le j < k$, it follows that for these values of j, all of the quadrilaterals

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} j \\ 1 \end{pmatrix} \quad \begin{pmatrix} j+1 \\ 1 \end{pmatrix}$$

are primitive sets. But then the origin has at least k neighbors, and therefore the number of neighbors is not a polynomial function of the data.

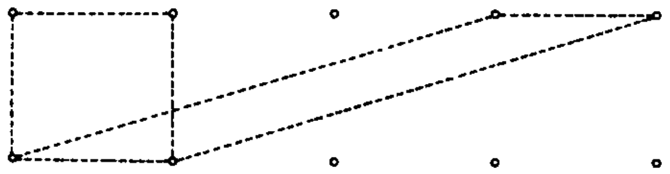


FIGURE 32

This observation is important enough to be put into the form of a theorem.

11.7 [Theorem]. The cardinality of the minimum neighborhood system associated with the general integer program is not polynomial in the data.

If the two variable problem is to have a polynomial algorithm it must mean that the chain of quadrilateral primitive sets has sufficient structure so that it can be inspected in polynomial time, even though it may contain a non-polynomial number of members.

Consider a matrix at one end of such a chain with sign pattern

$$\begin{bmatrix} - & - \\ - & + \\ + & - \\ + & + \end{bmatrix} .$$

It is elementary that there is a non-negative vector π such that $\pi B = 0$ with either $\pi_3 = 0$ and the remaining coordinates positive, or $\pi_0 = 0$ and the remaining coordinates positive. Let us assume that the former occurs. It follows that a matrix in the sequence obtained by replacing a column by the sum of two columns will never have $b_{i1} + b_{i2} < 0$ for both $i = 1, 2$. Until the chain is completed one of these sums will be negative and the other positive.

Assume that $b_{11} + b_{12} < 0$ and write $-b_{11} = tb_{12} + r$ with t a positive integer and $0 \leq r < b_{12}$. The matrices

$$\begin{bmatrix} b_{01} + jb_{02} & b_{02} \\ b_{11} + jb_{12} & b_{12} \\ b_{21} + jb_{22} & b_{22} \\ b_{31} + jb_{32} & b_{32} \end{bmatrix}$$

will have the same sign pattern as the original matrix for $j = 0, 1, \ldots, t$, so that the quadruples

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ j \end{pmatrix} \quad \begin{pmatrix} 1 \\ j+1 \end{pmatrix}$$

will all form primitive sets. Such a sequence of primitive sets will be called a link in the chain of quadrilaterals.

11.8 [Definition]. A sequence of quadrilateral primitive sets all of whose members lie on two parallel straight lines will be called a link in the chain of quadrilaterals.

The particular link illustrated above terminates when $j = t$ since $(b_{11} + tb_{12}) + b_{12}$ is no longer negative. If the chain of quadrilaterals does not terminate it must now be true that $(b_{21} + tb_{22}) + b_{22} < 0$. We then write $-b_{22} = t'(b_{21} + tb_{22}) + r'$, with t' a positive integer, and $0 \leq r' < (b_{21} + tb_{22})$. This gives a new link of quadrilateral primitive sets with t' members, illustrated in Figure 33. By arguments similar to those given in Section VII, it may be shown that the quantity

$$\begin{aligned} &\log_2 |b_{11}| + \log_2 |b_{12}| \\ + &\log_2 |b_{21}| + \log_2 |b_{22}| \end{aligned}$$

decreases by at least one unit in traversing a link in the chain. This demonstrates the following conclusion.

11.9 [Theorem]. The number of links in the chain of quadrilateral primitive sets is polynomial in the data of the problem.

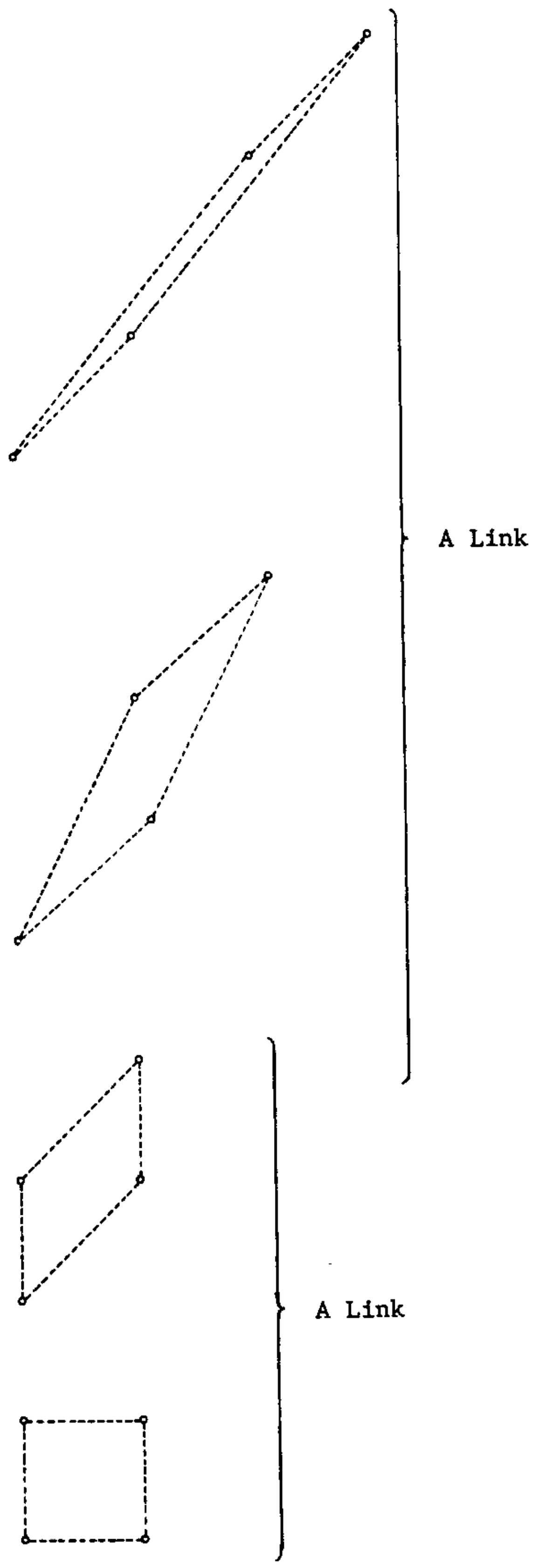


FIGURE 33

XII. The General Algorithm for Integer Programs with Two Variables

We are now in a position to discuss our algorithm for integer programs with two variables and three inequalities:

$$\begin{aligned} \max\ & a_{01}h_1 + a_{02}h_2 \\ & a_{11}h_1 + a_{12}a_2 \geq b_1 \\ & a_{21}h_1 + a_{22}h_2 \geq b_2 \\ & a_{31}h_1 + a_{32}h_2 \geq b_3 \end{aligned}$$

with h_1, h_2 integral. Let us assume that the methods of the last several sections have already been applied to determine the chain of quadrilateral primitive sets and the two simplicial subdivisions appearing at the ends of the chain. To be specific let the inequalities 0 and 3 relate to opposing vertices in each quadrilateral, and similarly for inequalities 1 and 2. In addition let one of the simplicial subdivisions involve the 3rd slack vector and the other the 1st slack vector. The case in which one of the subdivisions involves the zeroth slack vector will be commented on later.

We shall use the labeling rule which labels a vector $x = Ah$ with the largest subscript corresponding to a violated inequality, and with the label zero if all inequalities are satisfied. The algorithm of Section IV for solving integer programming problems is concerned with a sequence of almost completely labeled primitive sets, each of which bears the labels 1, 2, and 3, with one of them doubled.

Let us imagine that the algorithm has entered a chain of quadrilaterals and moved to the beginning of a link in the chain without having found a completely labeled primitive set. We wish to show that the passage through the link can be accomplished with an amount of numerical computation which is independent of the number of primitive sets in the link and more generally independent of the complexity of the problem.

Without loss of generality we may transform the problem so that the primitive sets in the link consist of the quadrilaterals

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ j \end{pmatrix} \quad \begin{pmatrix} 1 \\ j+1 \end{pmatrix}$$

for $j = 0, 1, \ldots, t$. The corresponding primitive sets will be given by the columns of

$$\begin{bmatrix} \underline{b_{01}+(j+1)b_{02}} & b_{01}+jb_{02} & b_{02} & 0 \\ b_{11}+(j+1)b_{12} & \underline{b_{11}+jb_{12}} & b_{12} & 0 \\ b_{21}+(j+1)b_{22} & b_{21}+jb_{22} & \underline{b_{22}} & 0 \\ b_{31}+(j+1)b_{32} & b_{31}+jb_{32} & b_{32} & \underline{0} \end{bmatrix}$$

with row minima lying on the main diagonal. The first of these primitive sets appears in Figure 34, and the reader may verify that the four dispositions of the labels shown in Figure 35 are exhaustive.

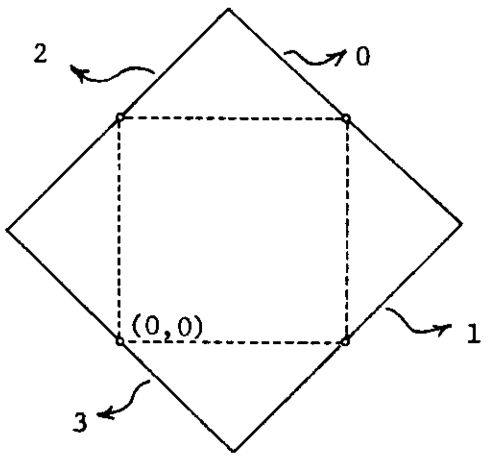


FIGURE 34

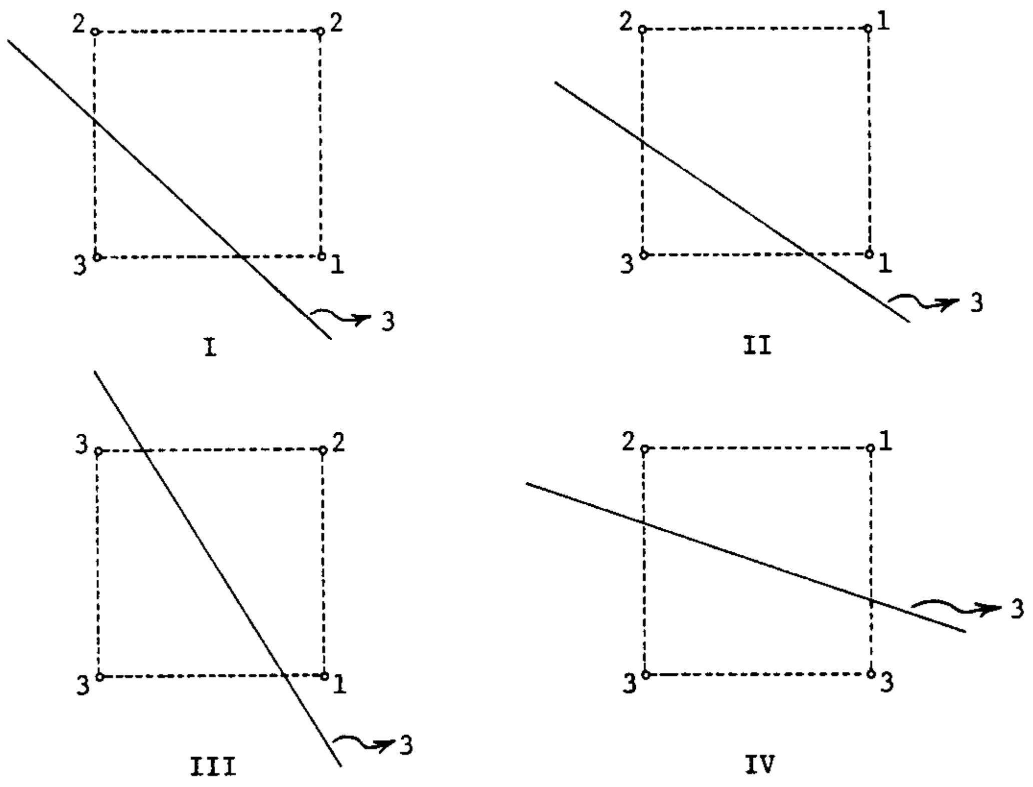


FIGURE 35

We shall examine each of these four possibilities in turn and use the orientation arguments of Section VI to determine which of the vectors with doubled labels is to be removed. As the reader will remember we remove the vector which yields an odd permutation of the integers 0, 1, 2, 3 when its label is replaced by the missing label 0 in the sequence

$$\ell(1,1),\ \ell(1,0),\ \ell(0,1),\ \ell(0,0)\ .$$

Case I

In this case the sequence of labels is (2,1,2,3) so that (0,1) is removed and replaced by (0,-1) , which bears the label 3. It follows that (0,0) is then removed and we continue down the line with 1st coordinate equal to zero, until the final primitive set in the link,

$$\begin{pmatrix} 0 \\ -t \end{pmatrix} \begin{pmatrix} 0 \\ -t+1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

is reached. Of course no calculation is necessary and we move immediately to this latter primitive set.

Case II

The sequence of labels is now (1,1,2,3) and therefore the vector (1,0) is removed and replaced by (1,2) . We continue moving up the line: if for every such point (1,2), (1,3), ..., (1,t) the label 1 is retained then we move immediately to the final primitive set

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \end{pmatrix} \begin{pmatrix} 1 \\ t+1 \end{pmatrix}.$$

On the other hand there may be some point on this line for which the label changes from 1 to either 0 or 2. By carrying out two divisions we can find the smallest value of j , say j^* , for which $(1,j)$ has a label different from 1. If this label is 0 then the primitive set given by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ j^*-1 \end{pmatrix} \begin{pmatrix} 1 \\ j^* \end{pmatrix}$$

bears all four labels and the vector $(1,j^*)$ is the global optimum for the programming problem.

On the other hand the label for $(1,j^*)$ may be 2, as in Figure 36. At this point the algorithm calls for the removal of (0,1) and its replacement by (0,-1) . But this vector has the label 3 as does every vector on the line $(0, h_2)$ with $h_2 \leq 0$. This means that we proceed immediately to the almost completely labeled primitive set

$$\begin{pmatrix} 0 \\ -t+j^*-1 \end{pmatrix} \begin{pmatrix} 0 \\ -t+j^* \end{pmatrix} \begin{pmatrix} 1 \\ j^*-1 \end{pmatrix} \begin{pmatrix} 1 \\ j^* \end{pmatrix}$$

which initiates the next link in the chain of quadrilaterals.

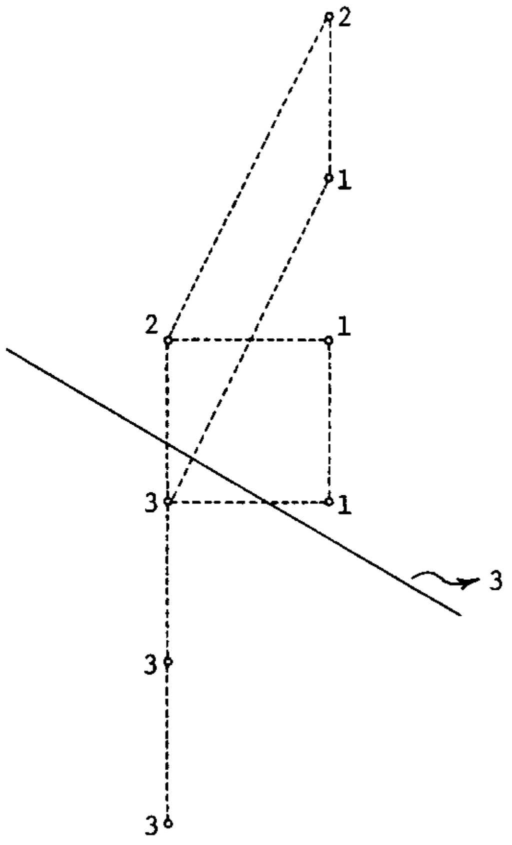


FIGURE 36

Case III

The sequence of labels for this case is $(2,1,3,3)$ so that $(0,1)$ is replaced by $(0,-1)$. Since all of the vectors on the line $(0, h_2)$ with $h_2 \leq 0$ have the label 3, we immediately move to the final quadrilateral in the link:

$$\begin{pmatrix} 0 \\ -t \end{pmatrix} \begin{pmatrix} 0 \\ -t+1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

Case IV

The sequence of labels for this final case is (1,3,2,3) so that (1,0) is removed and replaced by (1,2) . The subsequent argument is then identical with Case II.

To summarize we see that in two of the four cases we can move to the final quadrilateral in the link with no calculation at all. In the other two cases the passage through the link depends solely on determining the first point on the line (1,j) at which a label different from 1 first appears. Since the number of links in the chain is polynomial in the data of the problem we see that the passage through the entire chain of almost completely labeled quadrilaterals can be effected using a number of calculations which is polynomial in the data.

But even more can be said. We can assume that the initial quadrilateral in the chain is given by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} ,$$

which is immediately preceeded by primitive sets composed of translates of the two triangles

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and}$$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} ,$$

in conjunction with the 1^{st} slack vector. At the other end of the chain is another simplicial subdivision in conjunction with the 3^{rd} slack vector. If the algorithm passes through the chain it will never return to the chain since if it does it will enter through the latter simplicial subdivision and be forced to move through the chain in a direction opposite to that given by our orientation argument. It follows that our algorithm can be made polynomial in the data if we can recognize, in polynomial time, a completely labeled primitive set associated with the first simplicial subdivision or the initial almost completely labeled primitive set in the chain of quadrilaterals.

But this is quite easy. Let us solve the programming problem in which the first inequality is neglected. We adopt the labeling rule which labels a vector $x = Ah$, with the label 3 if x violates the third inequality, with the label 2 if it satisfies the third but violates the second inequality, and with the label 0 if both the second and third inequalities are satisfied.

Using our previous algorithm, we can find in polynomial time, a triangle in the appropriate simplicial subdivision whose three vertices bear the labels 3, 2, 0. One possibility is illustrated in Figure 37.

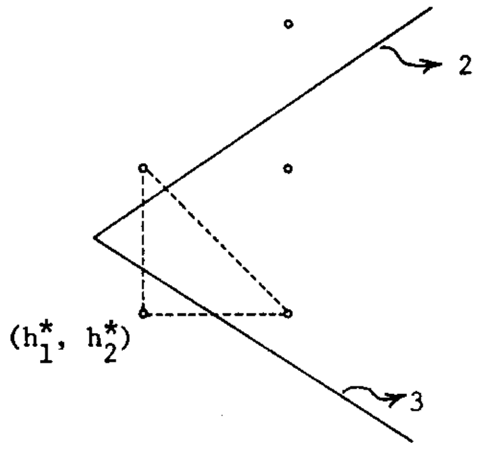


FIGURE 37

In this figure the vector (h_1^*, h_2^*) receives the label 3 since it violates the third inequality, and (h_1^*, h_2^*+1) the label 2, since it satisfies the third but violates the second inequality. The vector (h_1^*+1, h_2^*) satisfies both of these inequalities. If in addition it satisfied the first inequality--which was neglected in determining this vector--it would be true that (h_1^*+1, h_2^*) would be the maximum for the programming problem involving all three inequalities. In order to continue we assume therefore that (h_1^*+1, h_2^*) violates the first inequality of the programming problem.

Let the labeling procedure now be changed so that a vector $x = Ah$ receives as a label the largest subscript associated with a violated inequality. With this modification the vectors (h_1^*, h_2^*) , (h_1^*, h_2^*+1) , (h_1^*+1, h_2^*) receive the labels 3, 2, 1 respectively. These three vectors form a primitive set in conjunction with the first slack vector, which receives the label 1. We have therefore constructed an almost completely labeled primitive set. By removing the first slack vector, we introduce

$(h_1^* + 1,\ h_2^* + 1)$ and enter the chain of quadrilaterals in Cases I or II of Figure 37.

The solution of the programming problem in which the first inequality is neglected may also be revealed by a triangle of the form

$$\begin{pmatrix} h_1^* + 1 \\ h_2^* \end{pmatrix} \quad \begin{pmatrix} h_1^* \\ h_2^* + 1 \end{pmatrix} \quad \begin{pmatrix} h_1^* + 1 \\ h_2^* + 1 \end{pmatrix}$$

instead of the triangle in Figure 37. In this case $(h_1^* + 1,\ h_2^*)$ receives

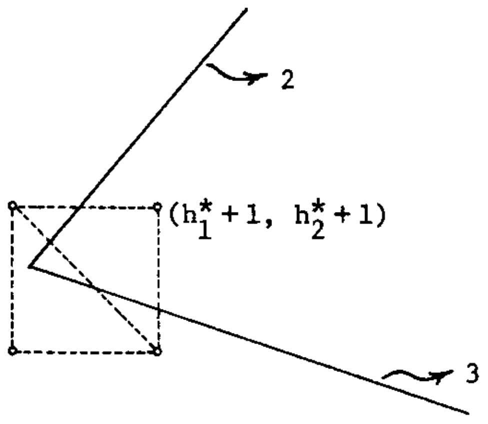


FIGURE 38

the label 3, $(h_1^*,\ h_2^* + 1)$ the label 2 and $(h_1^* + 1,\ h_2^* + 1)$ satisfies the 2nd and 3rd inequalities. If this latter vector also satisfies the 1st inequality it is the global maximum to the programming problem. As before we assume that $(h_1^* + 1,\ h_2^* + 1)$ violates the 1st inequality, and change the labeling procedure so that a vector $x = Ah$ is labeled with the largest subscript corresponding to a violated inequality. The primitive set consisting of these three vectors and the first slack vector form an almost completely labeled primitive set. When the first slack vector is removed we enter the chain of quadrilaterals in either Case III or IV.

One final remark: If the simplicial subdivision at the end of the chain involves the 0^{th} slack vector rather than the 3^{rd} slack vector, entering this subdivision means that the final primitive set involves the 0^{th} slack vector, and the problem is therefore infeasible. These arguments demonstrate the following theorem.

12.1 [Theorem]. Our algorithm may be accelerated so as to provide a polynomial algorithm for the general integer program with 2 variables.

BIBLIOGRAPHY

Bell, David E. (1977), A Theorem Concerning the Integer Lattice, Studies in Applied Mathematics, 56, 187-188.

Hoffman, Alan J. (1978A), Binding Constraints and Helly Numbers, to appear in Proceedings of the International Colloquim on Combinatorics, N.Y. Academy of Sciences.

__________ (1978B), Helly Numbers of Some Sets in R^n, to appear in Proceedings of the Second Summer School on Combinatorial Optimization, SOGESTA, Urbino.

Kannan, Rovindran (1977), A Polynomial Algorithm for the Two Variable Integer Programming Problem, Tech. Report No. 348, School of Operations Research, Cornell University.

Scarf, Herbert E. (1973) (with the collaboration of Terje Hansen), The Computation of Economic Equilibria, Yale University Press, New Haven.

__________ (1977), An Observation on the Structure of Production Sets with Indivisibilities, Proc, Nat. Acad. Sci. U.S.A. (74), 3637-3641.