Cowles Foundation Paper 516

ECONOMETRICA

VOLUME 49 JANUARY, 1981 NUMBER 1

PRODUCTION SETS WITH INDIVISIBILITIES PART I: GENERALITIES[1]

BY HERBERT E. SCARF

This paper and its sequel present a new approach to the study of production sets with indivisibilities and to the programming problems which arise when a factor endowment is specified. The absence of convexity precludes the use of prices to support efficient production plans and to guide the search for optimal solutions. Instead, we describe the unique minimal system of neighborhoods for which a local maximum is global, and discuss a related algorithm. The definition of this neighborhood system is based on techniques used in the computation of fixed points of a continuous mapping. In Part II of the paper this neighborhood system is investigated in the special case of two activities and it is shown that the algorithm may be accelerated so as to terminate in polynomial time.

INTRODUCTION

THE ASSUMPTION OF CONVEX PRODUCTION SETS plays a central role in neoclassical economic theory. Its replacement by weaker and more plausible assumptions seems to me to be one of the major challenges of mathematical economics. The present paper, and its sequel, present a new approach to the study of discrete production sets, and to the mathematical programming problems which arise when a particular factor endowment is specified.

The primary consequence of the convexity assumption is the existence of a vector of prices which supports an arbitrary efficient production plan. This leads immediately to the duality theorem for linear programming when the technology is given by an activity analysis model and to the existence of implicit prices for the general convex programming problem. The major economic theorems concerning decentralization of economic activity arise directly from this body of ideas.

Decentralized prices are no longer available when the production set displays increasing returns to scale, indivisibilities, or other forms of nonconvexity. There is no natural algorithm, based on prices, which verifies that a proposed solution to the associated programming problem is optimal and no corresponding theory of decentralization in production. Our major innovation, for the case of discrete production sets, will be to replace the concept of competitive prices by an entirely different analytical apparatus in order to solve the discrete programming problems arising from a specification of factor endowments.

The most important example of a discrete production set is an activity analysis model in which the activity levels are restricted to being integers rather than assuming arbitrary real values. We shall associate with each such vector of activity

[1] The research described in this paper and its sequel 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, Roger Howe of Yale University, and Michael Todd of Cornell University for the many interesting conversations I have had with them on the general topic of this paper.

levels a *neighborhood* consisting of a finite set of nearby activity vectors. The neighborhood system will be defined in a canonical fashion for each activity analysis model with discrete activity levels. A major conclusion of this paper will be the theorem that this neighborhood system is the unique, minimal neighborhood system for which a local maximum for the associated integer programming problem is global.

The method for constructing this neighborhood system will be based on the concept of *primitive sets*, introduced in the study of fixed point algorithms. If the neighborhood system for a particular activity analysis model were known we would have available to us an elementary algorithm for solving the associated integer programming problems. Given a vector of activity levels which satisfies the constraints of the programming problem we simply check the finite list of vectors in its neighborhood to see whether one of them satisfies the constraints and yields a superior value of the objective. If there is one we move to this vector and repeat the construction; if not, we have the optimal solution.

In this algorithm the test for optimality by means of competitive prices has been replaced by a search through a neighborhood which is intrinsically defined by the activity analysis model. The usefulness of the algorithm depends on the difficulty in determining this neighborhood system and on the complexity of its description. If the neighborhood system for a specific technology were extremely complex, the search for optimal solutions would necessarily be replaced by a series of heuristic tests which exploit the broad features of the system rather than its fine detail.

The theory of computational complexity is a major advance in our ability to describe and investigate the intrinsic complexity of mathematical programming problems. In this theory a basic distinction is made between algorithms which terminate in *polynomial* time—as a function of the size of the problem being studied—and those which require an *exponential* amount of time for their successful execution. In the second half of this paper these ideas will be introduced and we shall demonstrate that our methods lead to a polynomial algorithm for the general integer program with two variables. A study of the three variable problem is being carried out in collaboration with Sergiu Hart and Roger Howe.

Aside from the work of Herbert A. Simon and his collaborators the concept of computational complexity has not played an important role in economic theory. It does seem to me, however, that an assessment of the computational difficulties introduced by nonconvex production sets is central to economic analysis. I feel that the subject of computational complexity will illuminate our understanding of this question and provide a new bridge between mathematical programming and economic theory.

1. THE SIMPLICIAL COMPLEX

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:

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

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 assumption:

ASSUMPTION 1.2 (Non-Degeneracy): No two vectors in X have the same ith 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 nonpositive orthant having its vertex at that particular vector. This provides us with an intuitive picture 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.

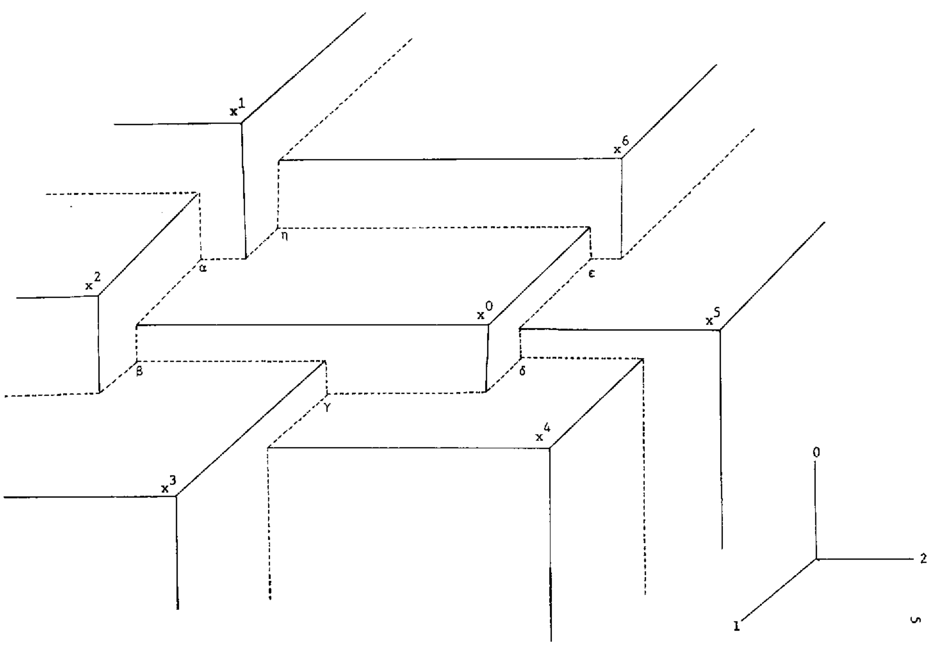

FIGURE 1

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 nondegeneracy 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 [**5**]), will be the m-simplices of our collection. This definition may easily be seen to be equivalent to the following:

DEFINITION 1.3: 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^2, x^3 a primitive set with vertex β. In order to be somewhat more concrete let us imagine that the vectors in Figure 1 arise from 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.

A translate of the positive orthant with vertex (c_0, c_1, c_2) corresponds, in activity level space, to a specific positioning of the three inequalities

$$a_{i1}h_1 + a_{i2}h_2 \geq c_i, \quad \text{for} \quad i = 0, 1, 2.$$

If there are no vectors x in this translated positive orthant, then this region in activity level space contains no lattice points. When the coordinates of the translated positive orthant are decreased, the corresponding inequalities are relaxed until three lattice points are reached. Primitive sets are seen, therefore, to correspond to all triples of lattice points, say h^0, h^1, h^2, arising by an arbitrary relaxation of the inequalities from a lattice free region. This is, of course, stronger than merely requiring that there be no lattice points in the convex hull of h^0, h^1, h^2. The nondegeneracy Assumption 1.2 is required to insure that there is no ambiguity about which lattice point is reached first when an inequality is relaxed.

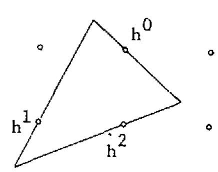

FIGURE 2

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 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.

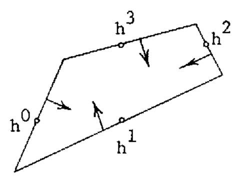

FIGURE 3

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

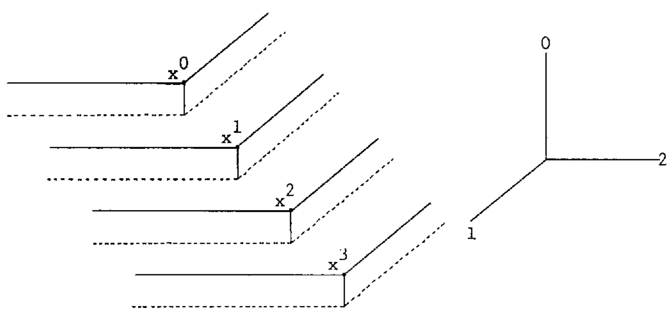

FIGURE 4

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 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.

DEFINITION 1.4 (Slack Vectors): The *slack vector* ξ^i is defined by saying that its ith coordinate is *less than* the ith coordinate of any of the vectors in X, and its jth coordinate (for $j \neq i$) is *larger* than the jth coordinate of any of the vectors in X.

The definition of primitive sets given in 1.3 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).

At various points in our subsequent arguments it will be useful to require that the number of primitive sets containing any specific vector in X be finite. Of course, this will automatically be satisfied if X is finite and the reader may wish to make this assumption and proceed directly to the next section. But since our primary application is to integer programming, in which case X is certainly infinite, some discussion of this point is required.

An example of a set X, some of whose members are contained in an infinite number of distinct primitive sets was given to me by Michael Todd in a private communication. Let $m = 2$, and let X consist of the origin $(0, 0, 0)$ and an infinite set of points on the plane $x_0 = 1$, given by $(1, -h, -1/h)$ where h is either a positive integer or the reciprocal of a positive integer.

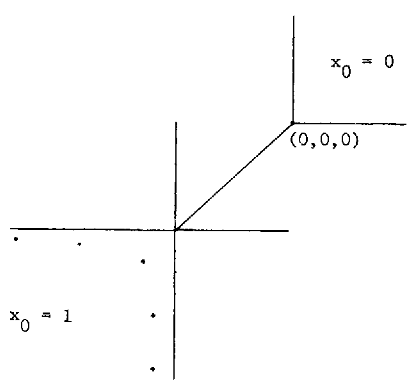

FIGURE 5

Primitive sets for this example will consist of the origin and any pair of adjacent points on the plane $x_0 = 1$; the origin will therefore be contained in an infinite number of distinct primitive sets.

This set X has a particular property whose absence is sufficient to rule out membership in an infinite number of primitive sets. Let us consider the problem of finding the vector x in X which maximizes x_2 subject to the single constraint $x_0 \geqslant b$. If, for example, $b=-1$, the vector (0, 0, 0) will have the largest second coordinate. On the other hand if $b=1/2$, the maximum will not be assumed; the second coordinate in the sequence $(1, -h, -1/h)$ can be made arbitrarily close to 0, but cannot attain this value.

Let X be a general set in R^{m+1}, S a subset of the indices $(0, 1, \ldots, m)$, and i^* a particular index in S. We shall consider the problem of finding that vector $x \in X$ which maximizes x_{i^*} subject to inequalities of the form

$$x_i \geqslant b_i \quad \text{for} \quad i \in S - \{i^*\}.$$

A maximum, if it exists, will certainly have the property that there is no vector $x' \in X$ with $x'_i > x_i$ for $i \in S$. This motivates the following definition.

DEFINITION 1.5: A set $S \subset (0, 1, \ldots, m)$ is defined to be an *efficient* set of indices if there is a vector $x \in X$ such that for no $x' \in X$ is $x'_i > x_i$ for all $i \in S$.

This property is a minimal requirement on a set of indices if the associated maximization problems are to have a solution. It also permits us to formulate the following basic assumption.

ASSUMPTION 1.6: Let S be an efficient set of indices, and Y a subset of X whose members satisfy the inequalities $y_i \geqslant b_i$ for $i \in S$. Let i^* be a particular index in S for which $y_{i^*} \leqslant c$ for all y in Y. Then there is a vector y^* in Y with $y^*_{i^*} > y_{i^*}$ for all other vectors in Y.

Assumption 1.6 is in the nature of a compactness assumption. It states that bounded maximization problems based on an efficient set of indices achieve their maxima, even when the allowable vectors are restricted to an arbitrary subset of X. The assumption will be useful in a variety of subsequent arguments in addition to its role in the following theorem, whose proof will be given in the Appendix.

THEOREM 1.7: *Assumption* 1.6 *implies that each vector in X is contained in a finite number of distinct primitive sets.*

In order to complete this section let us make some observations about the important case in which X consists of all vectors of the form $x = Ah$ with A an $(m+1) \times n$ matrix and h ranging over all integral vectors in R^n. At various points it will be useful to assume that the entries in A are themselves integral. Unfortunately this causes some difficulty with the nondegeneracy assumption since a pair of vectors x and x' can have an identical ith coordinate without the vectors being identical in all coordinates. But for our subsequent arguments all that is required is that the ith coordinates of the vectors in X be totally ordered, with an ordering $\}$ which is irreflexive, complete, and transitive. The natural ordering with these properties is the lexicographic ordering which states that $x'_i \} x_i$ if (i) $x'_i > x_i$ or (ii) $x'_i = x_i$ and the vector $x' - x$ is lexicographically positive.

We shall assume, without being explicit about it, that this ordering is used whenever it is necessary. The proof of the following theorem will also be deferred to the Appendix.

THEOREM 1.8: *Let the set X consist of the vectors $x = Ah$ with A an integral $(m+1)\times n$ matrix and h ranging over the lattice points in R^n. Assume that the lexicographic ordering is used to break ties. Then Assumption 1.6 is satisfied.*

2. 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

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

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

$$\begin{aligned} \max\ & a_{01}h_1 + \ldots + a_{0n}h_n, \quad \text{subject to} \\ & a_{11}h_1 + \ldots + a_{1n}h_n \geqslant b_1, \\ & \qquad \vdots \\ & a_{m1}h_1 + \ldots + a_{mn}h_n \geqslant b_m, \end{aligned}$$

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 $x \in X$ 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.

DEFINITION 2.1: 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.

THEOREM 2.2: *An efficient vector x in X has a nonempty set of neighbors. In particular for any $l = 0, 1, \ldots, m$, that vector x' in X (or the slack vector ξ^l) whose lth coordinate is maximal, subject to $x_i' > x_i$ for all $i \neq l$, is a neighbor of x.*

We demonstrate this theorem for the case $l = 0$, 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 $x' \in X$ is reached.

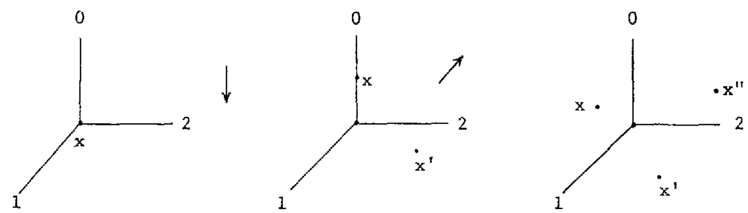

FIGURE 6

This 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 existence of the relevant vectors follows from Assumption 1.6.

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

$$x_1 \geqslant b_1,$$
$$\vdots$$
$$x_m \geqslant b_m.$$

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.

THEOREM 2.3: *Let x^* be an efficient vector in X and satisfy the inequalities $x_i^* \geqslant b_i$ for $i = 1, \ldots, m$. Assume that for every neighbor x' of x^* either (i) $x_i' < b_i$ for some $i = 1, \ldots, m$ or (ii) $x_0' < x_0^*$. Then x^* is that vector in X which maximizes x_0 subject to $x_i \geqslant b_i$ 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 $x_m \geqslant b_m$ 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 7 may easily be seen to satisfy Assumptions 1.2 and 1.6.

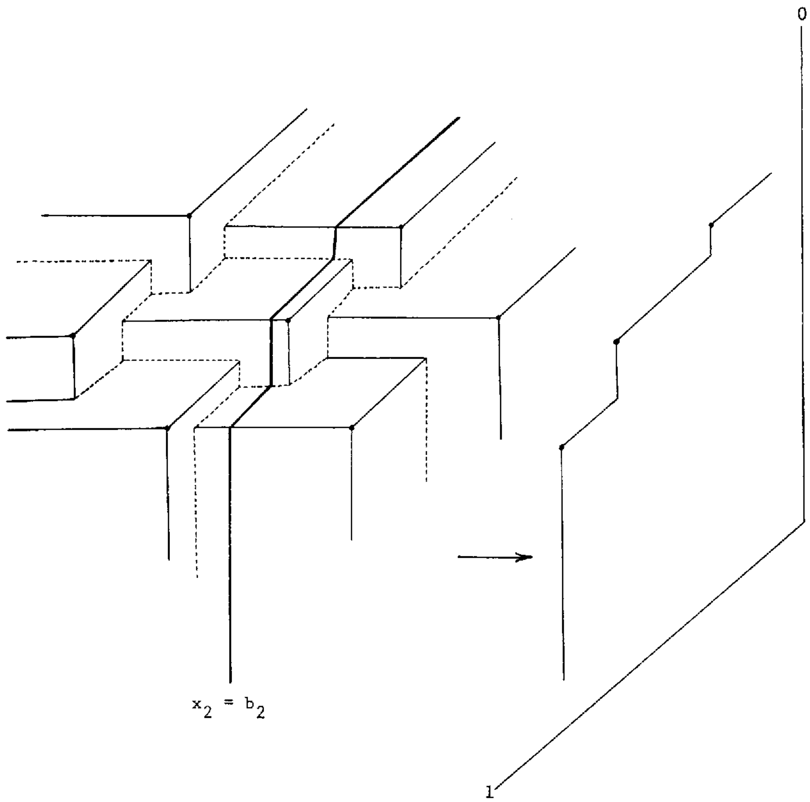

FIGURE 7

As Figure 7 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.

LEMMA 2.4: *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 (whose existence is guaranteed by Assumption 1.6) 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

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

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.

LEMMA 2.5: *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 the vector in X, whose mth 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

$$
\begin{aligned}
&x^m_0 > \min\,[y^*_0, y_0, y^2_0, \ldots, y^{m-1}_0],\\
&\quad\vdots\\
&x^m_{m-1} > \min\,[y^*_{m-1}, y_{m-1}, y^2_{m-1}, \ldots, y^{m-1}_{m-1}].
\end{aligned}
$$

(Assumption 1.6 is used to guarantee the existence of such a vector.)

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

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

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

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

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

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

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

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

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

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

This demonstrates Theorem 2.3. A converse to this Theorem will be given in Section 4. We demonstrate there that any neighborhood system, for which a local maximum is global, must include the neighborhoods given by primitive sets.

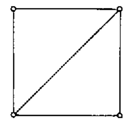

FIGURE 8

Theorem 2.3 suggests an obvious algorithm for the solution of discrete programming problems. Begin with an efficient vector which satisfies the inequalities $x_i \geqslant b_i$ 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 of the zeroth 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 zeroth 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, that if no structure whatsoever is imposed on X, 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 a 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 8, translated to an arbitrary lattice point. Each lattice point will have, therefore, the six neighbors shown in Figure 9.

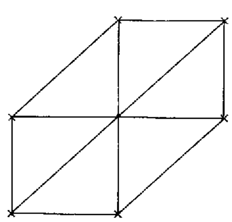

FIGURE 9

In order to solve the programming problem

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

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 $(h_1, h_2 - 1)$, 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.

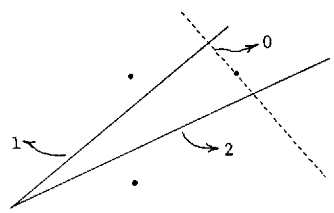

FIGURE 10

3. 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

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

$$(3.1) \qquad \begin{aligned} a_{11}h_1 + \dots + a_{1n}h_n &\geqslant b_1, \\ &\vdots \\ a_{m1}h_1 + \dots + a_{mn}h_n &\geqslant b_m, \end{aligned}$$

and with $h = (h_1, \dots, h_n)$ integral. Nonnegativity requirements on the variables, if any, will be incorporated in the constraints, whose number m will typically be larger than n. The lexicographic tie breaking rule will be used and as a consequence Assumptions 1.2 and 1.6 will apply to the discrete production set generated by this model.

We assume that the inequalities have a feasible integral solution, and a finite maximum. The maximizing x vector will, of course, be unique.

DEFINITION 3.2: 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 [**1**] and independently (though somewhat belatedly) by myself (Scarf [**6**]), states that the function $f(n)=2^n-1$ is the correct one for integer programming. An even earlier proof is given by Doignon [**2**].

THEOREM 3.3: *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.

LEMMA 3.4: *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 $n \geqslant 3$. 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), and denote its optimal solution by h^0. Let $\varepsilon > 0$ be small and consider the polyhedron defined by

$$\sum_j a_{0j}h_j \geqslant \sum a_{0j}h_j^0 + \varepsilon,$$

(3.5)

$$\sum_j a_{ij}h_j \geqslant b_i \quad \text{for} \quad i = 1, \ldots, m.$$

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).

DEFINITION 3.6: The inequality $\Sigma\, a_{ij}h_j \geqslant b_i$ 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 11 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.

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 11.

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 11, 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

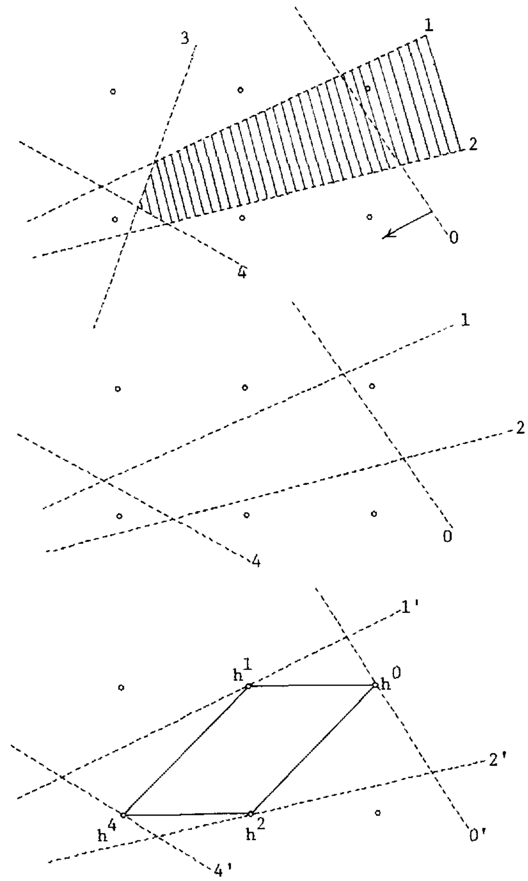

FIGURE 11

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 11.

The process results in a set of lattice points $\{h^i\}$ for $i \in S$, 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 12 illustrates this possibility for $n = 2$; a similar construction based on the unit cube in n-space will work in general.

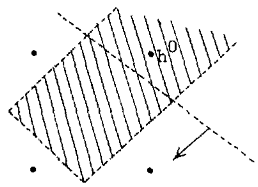

FIGURE 12

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.

Theorem 3.3 has been generalized by Alan Hoffman [**3, 4**], 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

$$\begin{array}{l} \sum_1^n a_{0j}h_j^0, \\ b_1, \\ \vdots \\ b_m, \end{array}$$

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.

THEOREM 3.5: *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 nonslack vectors in a primitive set is less than or equal to 2^n.*

4. THE REPLACEMENT OPERATION

In this section, 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.

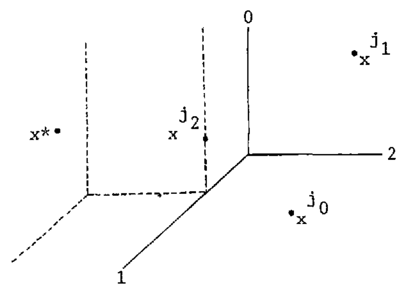

FIGURE 13

Before dealing with the general problem, let us examine geometrically the case in which the vectors in X lie in R^3. Figure 13 illustrates a primitive set composed of three vectors $x^{j_0}, x^{j_1}, x^{j_2}$. Without loss 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 ith 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:

$$(4.1) \qquad \begin{bmatrix} \underline{x_0^{j_0}} & x_0^{j_1} & \cdots & x_0^{j_m} \\ x_1^{j_0} & \underline{x_1^{j_1}} & & x_1^{j_m} \\ \vdots & \vdots & & \vdots \\ x_m^{j_0} & x_m^{j_1} & & \underline{x_m^{j_m}} \end{bmatrix}.$$

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

$$
(4.2)\qquad
\begin{aligned}
&x_0 > x_0^{j_1},\\
&x_2 > x_2^{j_2},\\
&\quad\vdots\\
&x_m > x_m^{j_m}.
\end{aligned}
$$

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

$$
(4.3)\qquad
\begin{bmatrix}
x_0 & \underline{x_0^{j_1}} & \cdots & x_0^{j_m}\\
\underline{x_1} & x_1^{j_1} & & x_1^{j_m}\\
\vdots & \vdots & & \vdots\\
x_m & x_m^{j_1} & & \underline{x_m^{j_m}}
\end{bmatrix}.
$$

Suppose that there is a nonslack vector in X satisfying (4.2). Let us use Assumption 1.6 to verify that there will be—among such vectors—one which maximizes x_1. Observe that the ith inequality, for $i \geqslant 2$, can be disregarded if the ith slack vector is a member of the original primitive set. But the remaining indices, augmented by the index 0, form an efficient set of indices, and Assumption 1.6 can then be applied.

On the other hand if there are no nonslack vectors satisfying (4.3), then the first slack vector ξ^1 will be the replacement for x^{j_0} 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.

It will be 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 (4.3) 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 (4.3), an observation which determines the replacement uniquely. We summarize these observations in the following theorem.

THEOREM 4.4: *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 nonslack vector which we are attempting to remove. In this latter case no replacement exists.*

The replacement step may be used to provide a converse to the theorem of Section 2. In that section 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.

DEFINITION 4.5: A neighborhood structure is defined by associating with each efficient vector x in X a nonempty 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 $x_i \geqslant b_i$ 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

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

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 than 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 mth coordinate, so that our matrix takes the form

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

Now let $X^1 = X - \{y\}$. If x^m is that vector (guaranteed by Assumption 1.6) in X^1 whose mth coordinate is maximal, subject to

$$x_0^m > x_0,$$
$$x_1^m > x_1^1,$$
$$\vdots$$
$$x_{m-1}^m > x_{m-1}^{m-1},$$

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 Figure 14. Of course, y will be contained in the positive orthant whose vertex is $\min[x, x^1, \ldots, x^m]$.

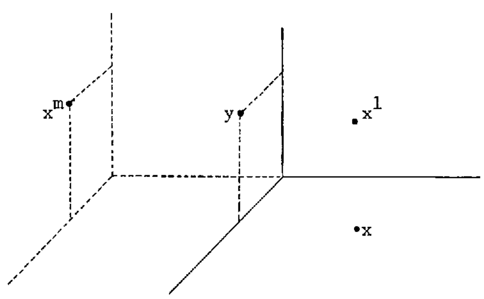

FIGURE 14

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

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

Then it follows from the definition of primitive sets that x is that vector in X^1 which maximizes x_0 subject to $x_i \geqslant b_i$ 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.

THEOREM 4.6: *A neighborhood system for which a local maximum is global, for all vectors b, must contain the neighborhood system defined by primitive sets.*

5. SPERNER'S LEMMA

The following Theorem is the analogue of Sperner's Lemma for primitive sets.

THEOREM 5.1: *Let X be a finite set, and assume that each vector x in X is given an integer label $l(x)$ selected from $(0, 1, \ldots, m)$. Let the ith slack vector ξ^i be given the label $l(\xi^i) = i$ for $i = 0, 1, \ldots, m$. Then there exists a primitive set all of whose labels are distinct.*

The argument for finding a completely labeled primitive set, when X is finite, 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 $l(x) = 0$ we have found a completely labeled primitive set since the slack vectors bear all of the remaining labels. If on the other hand $l(x) = i$ we remove the ith 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.

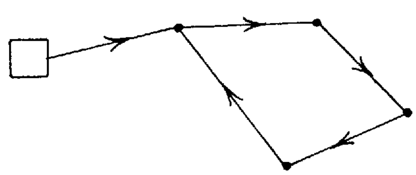

FIGURE 15

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 16. This observation leads immediately to the following refinement of Sperner's Lemma.

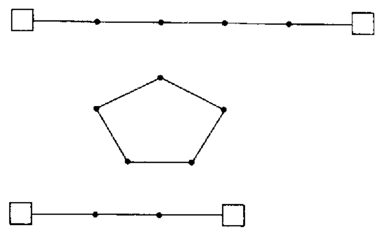

FIGURE 16

THEOREM 5.2: *The number of completely labeled primitive sets is odd.*

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

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

In order to simplify the exposition we continue to assume that X is finite; virtually all of our subsequent results have an analogue for infinite X satisfying 1.6. We adopt the following labeling rule.

LABELING RULE 5.3: We label x in X with the label $l(x) = i$ 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 $l(x) = 0$.

Figure 17 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 Labeling Rule 5.3. 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.

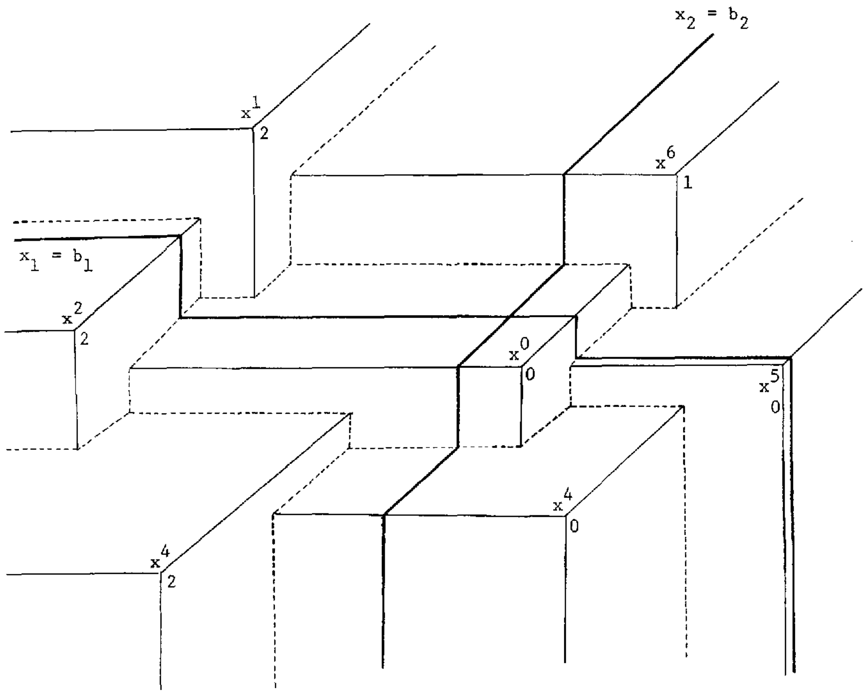

FIGURE 17

THEOREM 5.4: *Let x be that vector in a completely labeled primitive set—labeled according to Rule 5.3—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

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

lie on the main diagonal. The particular form of the Labeling Rule 5.3 permits us to argue that $l(x^i)=i$ for $i=0,\ldots,m$. To see this, we observe first of all, that $l(x^i)\geq i$ for all i. This is clearly correct for $i=0$. On the other hand if $l(x^i)<i$ for some $i\geq 1$ then $x_i^i\geq b_i$ and therefore no vector receives the label i. These observations imply immediately that $l(x^i)=i$ 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 5.4.

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 the second part of this paper 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 an integer program with n variables is 2^n-1.

6. INDEX THEORY

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 16. Our major conclusion will be that when the labeling rule is given by 5.3, the graph contains no cycles and is composed of a single path 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 path 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. In this regard the same arguments can be used to show that the minimum entry in row 0 is nonincreasing, in the sequence of primitive sets which arise when Sperner's Lemma is used to solve discrete programming problems.

Let us return to the general problem studied in Section 5. The set X is taken to be finite and the labels $l(x)$ are arbitrary members of the set $(0,1,\ldots,m)$. As before the ith slack vector will receive the label $l(\xi^i) = i$. We introduce the following definition of the index of a completely labeled primitive set.

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

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

lie on the main diagonal. Then

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

is defined to be $+1$ if the permutation $l(x^{j_0}), \ldots, l(x^{j_m})$ is *even*, and -1 if the permutation is *odd*.

We shall demonstrate the following important generalization of Theorem 5.2 which states that the number of completely labeled primitive sets is odd.

THEOREM 6.3: *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 with label 0 missing 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 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.

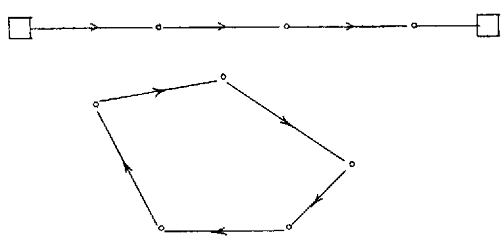

FIGURE 18

The string of symbols

$$l(x^{j_0}), \dots, l(x^{j_\alpha}), \dots, l(x^{j_\beta}), \dots, l(x^{j_m})$$

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

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

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.

Prescription of an Orientation 6.5: 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 *odd*.

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

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

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

We must demonstrate that Orientation 6.5 does not require us to remove x from the new primitive set. The permutation $l(x^{j_0}), \dots, 0, \dots, l(x^{j_m})$ obtained by replacing $l(x)$ by 0 in the new primitive set is identical with the previous permutation and is therefore odd. But a simple transposition (see, for example, (4.3)) 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.

Lemma 6.6: *A completely labeled primitive set which is reached by traversing the graph in the direction given by Orientation* 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 $l(x^{j_0}), \ldots, l(x), \ldots, l(x^{j_m})$ 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 Lemma 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 $0, l(\xi^1), \ldots, l(\xi^m)$ 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 path 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 $l(x)$ are arbitrary. But a considerable sharpening is available when the Labeling Rule 5.3 is used to solve the discrete programming problem: find that vector x in X whose zeroth coordinate is maximal, subject to the inequalities

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

Let us now assume that Labeling Rule 5.3 is being used and let $x^0, x^1, \ldots, x^m$ be a completely labeled primitive set, arranged in our customary way:

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

Then, by the argument previously given in the proof of Theorem 5.4, we must have $l(x^i) = i$ for all i. We see therefore that the index associated with every completely labeled primitive set is $+1$. This demonstrates the following theorem.

THEOREM 6.9: *The Labeling Rule* 5.3 *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 path, 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

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

lie on the main diagonal. As before, it is easy to verify that the Labeling Rule 5.3 implies that $l(x^i) \geq i$.

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

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

We have the following lemma:

LEMMA 6.12: *$l(x^i) = i$ 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 $i \neq 0, i_1, \ldots, i_k$. For each such i in S there must be some x^j with $l(x^j) = i$. But $j \neq 0, i_1, \ldots, i_{k-1}, i_k$ since the labels of $x^0, x^{i_1}, \ldots, x^{i_k}$ are not in S. We see that the set of indices $\{l(x^i)\}$ for i in S is precisely S itself. Lemma 6.12 follows from the observation that $l(x^i) \geq i$ 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 Prescription 6.5. The permutation

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

where $l(x^{i_k})$ 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.

LEMMA 6.13: *If the orientation given by Prescription 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 ith coordinate.

Can α_0 be increased when the Labeling Rule 5.3 is followed? This can only occur when x^0 has one of the doubled labels, so that $l(x^{i_1}) = i_1$. 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 $l(x^0) = i_1$ 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 ith 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 $l(x^{i_2}) = i_2$. It follows from Lemma 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.

THEOREM 6.14: *When Labeling Rule* 5.3 *is used, the graph of the almost completely labeled primitive sets contains no cycles.*

The same arguments may be used to demonstrate the following theorem which provides an analogy between our methods and dual feasible algorithms for the solution of programming problems.

THEOREM 6.15: *When Labeling Rule* 5.3 *is used,*

$$\alpha_0 = \min [x_0^0, \ldots, x_0^m]$$

is monotonic along the single path of almost completely labeled primitive sets.

Yale University

Manuscript received June 1979; revision received February, 1980.

APPENDIX

In this Appendix we demonstrate the two theorems enumerated at the end of Section 1.

THEOREM 1.7: *Assumption* 1.6 *implies that each vector in* X *is contained in a finite number of distinct primitive sets.*

Let us assume to the contrary that x is contained in an infinite sequence of primitive sets. As usual the coordinates of the vectors in these primitive sets will appear as the columns in a matrix, arranged so that the row minima lie along the main diagonal. By selecting suitable subsequences, without loss of generality we may assume the following: (i) In each of these primitive sets, x has the smallest zeroth coordinate. (ii) The same slack vectors, if any, appear in each primitive set. (iii) If the ith slack vector does not appear, then the minimum entry in the ith row in the sequence of matrices representing primitive sets is unambiguously decreasing, constant, or increasing.

Let S be the set of indices i, including 0, such that the ith slack vector does not appear, and such that the minimum entry in the ith row of the matrices representing this sequence of primitive sets are bounded from below. S is an efficient set of indices since if there were an x' with $x'_i > x_i$ for $i \in S$, then x' would violate the definition of a primitive set sufficiently far in the sequence.

Now let $i^* \neq 0$ be in S and let Y be the set of all vectors appearing in column i^* in this sequence of matrices. But then all of the vectors in Y have their ith coordinates bounded from below, for $i \in S$, and $y_{i^*} < x_{i^*}$ for all such y. It follows from Assumption 1.6 that one of these vectors has the largest i^*th coordinate, for all $y \in Y$. This implies that the minimum i^*th coordinate must be decreasing, or constant for every such i^* in S. But then, by selecting a suitable infinite subsequence *all* of the row minima corresponding to nonslack vectors must be constant or decreasing. This is inconsistent with there being an infinite number of primitive sets in the sequence, and demonstrates Theorem 1.7.

The next Theorem demonstrates that Assumption 1.6 is satisfied when X arises from an activity analysis model with integral activity levels.

THEOREM 1.8: *Let* X *consist of the vectors* $x = Ah$, *with* A *an integral* $(m+1)\times n$ *matrix and* h *ranging over all lattice points in* R^n. *Assume that for each* i, *ties in the* i*th coordinates of the vectors* x *are broken by the lexicographic tie breaking rule. Then Assumption* 1.6 *is satisfied.*

, Let S be an efficient set of indices; an assumption which in this case is equivalent to saying that there is no $x = Ah$ with $x_i > 0$ for all i in S (using the lexicographic tie breaking rule should any of the x_i be equal to 0).

Let Y be a subset of X whose ith coordinates are bounded from below for all i in S, let i^* be a particular index in S, and assume that y_{i^*} is bounded from above for all $y \in Y$. If there is no $y^* \in Y$ with $y^*_{i^*} > y_{i^*}$ for all other $y \in Y$, then there is an infinite sequence $y^v \in Y$ with $y^{v+1}_{i^*} > y^v_{i^*}$ in the lexicographic sense. But the i^*th coordinates of the vectors in Y can take on a finite number of values; we can therefore assume that $y^{v+1}_{i^*} = y^v_{i^*}$ and the vectors $y^{v+1} - y^v$ are lexicographically positive.

The numbers y^v_i are bounded from below for all i in S. We can therefore, by selecting suitable subsequences, assume that $y^{v+1}_i - y^v_i \geqslant 0$ for all i in S. But the lexicographic rule for breaking ties in the i^*th coordinate applies to these coordinates as well. It follows that $y^{v+1}_i - y^v_i$ is lexicographically positive for all i in S. This contradiction demonstrates Theorem 1.8.

REFERENCES

[1] BELL, DAVID E.: "A Theorem Concerning the Integer Lattice," *Studies in Applied Mathematics*, 56 (1977), 187–188.

[2] DOIGNON, JEAN-PAUL: "Convexity in Cristalographic Lattices," *Journal of Geometry*, 3 (1973), 77–85.

[3] HOFFMAN, ALAN J.: "Binding Constraints and Helly Numbers," to appear in *Proceedings of the International Colloquium on Combinatorics*, N.Y. Academy of Sciences, 1978.

[4] ———: "Helly Numbers of Some Sets in R^n," to appear in *Proceedings of the Second Summer School on Combinatorial Optimization*, SOGESTA, Urbino, 1978.

[5] SCARF, HERBERT E. (with the collaboration of Terje Hansen): *The Computation of Economic Equilibria*. New Haven: Yale University Press, 1973.

[6] ———: "An Observation on the Structure of Production Sets with Indivisibilities," *Proceedings of the National Academy of Sciences, U.S.A.*, 74 (1977), 3637–3641.