JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematics and Mathematical Physics
Vol. 69B, No. 3, July–September 1965

A Primal (All-Integer) Integer Programming Algorithm*

Richard D. Young

Associate Professor of Economics, Rice University

(June 12, 1965)

The algorithm is most closely related to three existing procedures; the simplex method of G. B. Dantzig for linear programming problems, the Gomory all-integer integer programming algorithm, and the direct algorithm for integer programming of Ben-Israel and Charnes.

The algorithm is similar to the Gomory all-integer algorithm in these respects: (i) it is an all-integer algorithm; (ii) it uses the same cut generation procedure; (iii) it uses the cut row as the pivot row; and (iv) the pivot coefficient always has unit value. While the dual method provides the vehicle for moving from tableau to tableau in the Gomory all-integer algorithm, the simplex method has the analagous role in the primal algorithm. Thus in a general sense this algorithm is a primal analog to the (dual) Gomory all-integer algorithm.

The direct algorithm of Ben-Israel and Charnes also has the above similarities to the Gomory all-integer algorithm, but has one significant difference: an iteration or cycle of the direct algorithm must frequently include the solution of an "auxiliary problem" (which is itself an integer programming problem) or a determination that no solution to the "auxiliary problem" exists. In contrast, the cycles of the primal algorithm include only the adjoining of a Gomory cut and the execution of the change of basis procedure of the simplex method.

The procedure of the algorithm and the proof of finiteness are founded on a classification of cycles of the algorithm and on two theorems. Two types of procedural restrictions are imposed as a basis for proving finiteness: (a) selection of the incoming variable is subjected to regulation (beyond that required by the simplex method), and the rules applied are a function of the type of cycle being executed; (b) selection of the row used as the source of the data for the Gomory cut is restricted (in addition to the restriction implied by (ii), (iii), and (iv) above) in certain cycles of the algorithm.

Part I. A Primal Algorithm: Antecedents, Goals, and Problems

1.1. Introduction and Summary

The principal result reported in this paper is a new primal (all-integer) integer programming algorithm and a proof that the algorithm is finite—i.e., that it always terminates in a finite number of cycles. The general idea of a primal algorithm is not new.[1] The difficulty has been in the development of some specific primal algorithm—or a class of primal algorithms—which (a) can be shown to be finite and (b) is capable of obtaining a solution without recourse to supplemental *ad hoc* procedures and problems.

In part II we shall give a careful statement of the algorithm, and in part III present a proof that the algorithm is finite. In part I we discuss connections between the primal algorithm and several related topics. Although we shall summarize the relations of the primal algorithm to existing integer programming techniques, we do not intend to undertake here a careful and exhaustive review of the literature.[2] Our intent, rather, is to provide a summary in which, by selection and emphasis, we can spotlight critical aspects of the primal algorithm in terms of similarities and contrasts with existing techniques. We shall also comment on the general significance of integer programming, the particular significance of improved computational techniques for integer programming, and the possible contribution of the primal algorithm to improved computational efficiency.

1.2. Integer Programming: Definitions

Integer programming problems may be regarded as a special[3] class of linear programming problems in which the variables are required to take on zero or positive integral values. Mixed integer programming problems only require that some proper subset of the variables be restricted to zero or integral values.

*The author gratefully acknowledges the many helpful suggestions made by W. W. Cooper, R. E. Gomory, T. C. Hu, F. K. Levy, G. L. Thompson, and F. M. Tonge.
This study was supported, in part, by funds made available by the Ford Foundation to the Graduate School of Business, Stanford University. However, the conclusions, opinions, and other statements in this publication are those of the author and are not necessarily those of the Ford Foundation.
Presented at the Advanced Study Institute on Integer Programming and Network Flow, Tahoe City, Calif., July 1, to August 21, 1965, sponsored by the Office of Naval Research, The North Atlantic Treaty Organization, and the University of California.

[1] See, e.g., Ben-Israel and Charnes [1], and Harris [14].

[2] This task has been nicely accomplished in Ben-Israel and Charnes [1, pp. 227–238.]

[3] We acknowledge, of course, the existence and usefulness of alternative viewpoints, from which, e.g., ordinary linear programming problems are represented as a special case, at one extreme of a scale which continues through mixed integer programming problems to integer programming problems at the opposite extreme. And in terms of the characteristics required of real world problems for valid formulation as linear or integer programming problems, real world integer programming problems constitute the more general—i.e., the less restricted—class.

The primal algorithm and most of the other integer programming procedures discussed in this paper are integer—as distinct from mixed integer—programming algorithms.

A special class of linear programming problems exists in which all the basic solutions are also integral solutions. Transportation problems in which all demands and capacities are given as integers are typical of this class. Such problems, which have been termed "implicit integer programming"[4] problems, will obviously yield an integral solution to standard linear programming techniques which locate an optimal basic solution; these problems, therefore, generate no need for special integer programming solution techniques.

The primal algorithm, in common with some other integer programming techniques such as the Gomory all-integer algorithm and the Ben-Israel and Charnes direct algorithm, requires that the initial statement of the problem be in terms of a system of constraints and a criterion function in which all the constants are given as integers. In principle this requirement is not restrictive, since appropriate rescaling of equations and/or variables will convert any system given in rational constants to a system of integers.

1.3. Integer Programming: Significance and Computational Limitations

The potential significance of integer programming is directly related to the significance of the problems which are amenable to valid formulation as integer programming problems.[5] Some measure of the scope of integer programming applications is provided by Charnes and Cooper:

[The general definition of integer programming]

. . . carries within it as varied (and curious) a variety of problems as construction of Latin squares, analyses of switching circuits, solution of interrelated "either-or" refinery-equipment-running plans, sequencing or staging operations as in job-shop scheduling, and a general solution of the problem of optimization of an arbitrary piecewise linear functional over a disjoint union of convex polyhedra. It by-passes the historical (and ineffective) methods of scanning the possible local critical or local optimal points and proceeds directly to a global optimum, thereby providing a constructive calculus for such problems which, when perfected, may have efficiency comparable to the usual linear-programming methods on the usual linear-programming problems . . .[6]

Implicitly included in this listing are such specific problems as fixed-charge problems and more generally many of the problems which require nonlinear or dynamic programming formulations. It is probably no exaggeration to contend that the class of real world integer programming problems is at least as extensive and important as is the class of real world linear programming problems.

The ability to formulate a problem in terms of a mathematical system may be of great value for purposes of clarification and understanding. But with respect to the utility and knowledge provided by actual applications it is of limited value if the resulting mathematical problem cannot be efficiently solved. While it is not correct either to state or to suggest that current integer programming solution techniques are too inefficient to be of practical value, it is fair to state that a significant increase in integer programming computational efficiency is required to make integer programming a practicable tool with power that is comparable to ordinary linear programming. In a subsequent section we shall advance some general qualitative arguments suggesting that the primal algorithm may contribute to an increase in computational efficiency. Such comments are necessarily speculative; they are not designed to develop faith in a conclusion that would be better based on experimentation; they are designed to suggest some of the motivation behind the development of the primal algorithm and to offer some hypotheses that can now be investigated with the primal algorithm.

1.4. Solution Techniques: Search Routines

There are many possible methods of solving integer programming problems. We may distinguish two general classifications for these methods: (i) combinatorial search routines and (ii) cutting plane methods. While there is clearly some overlap in these categories, the distinction is useful to identifying the conceptual orientation of most methods for solving integer programming problems.

We shall give brief descriptions of some search routines. As a preliminary, we note that if the linear programming problem which contains the integer programming problem is bounded, attention can be confined to a finite set of integer solutions which can be systematically and exhaustively listed. Then the optimum can be selected, in accordance with any criteria that may be specified. Thus the existence of a finite algorithm is easily established.

Two methods, one developed by Land and Doig,[7] another by Szwarc[8] and Elmaghraby, exemplify combinatorial search routines. Both of these methods work with a tree graph which has nodes corresponding to all the solutions of a given integer programming problem. In both cases the tree graphs contain nodes that do not correspond to integer solutions. Both methods search the tree in a pattern designed to guarantee that the first integer solution node located will be an optimal integer solution node. Both methods guide the search process with information provided by solving parametric linear programming problems.

This general similarity should not suggest an absence of significant difference between these techniques. The Land and Doig "tree" has integer solutions only at its terminal nodes. Each node cor-

[4]The term is due to Ben-Israel and Charnes [1]. For an analysis which defines this class of problems see Hoffman and Kruskal [15].

[5]Important papers on the economic significance of integer programming include Gomory and Baumol [13], and Weingartner [20]. For a discussion of managerial and other applications see Dantzig [4].

[6]Quoted from Charnes and Cooper [3, p. 695].

[7]See [16].

[8]See [18].

responds to a collection of subsidiary constraints of the form

$$x_i = k_i, \qquad i \epsilon I,$$

where x_i is a variable and k_i is a positive integer or zero. At the terminal nodes the set I contains a sufficient collection of indices to fully determine a solution; at the preterminal nodes these subsidiary constraints do not fully determine a solution. In the Szwarc and Elmaghraby "tree" each node corresponds to a solution, which may or may not be integral, to the linear programming problem that contains (in its solution set) the solutions to the given integer programming problem. The structure of this tree is such that earlier nodes on a given branch or path have better criterion values than later nodes.

The Szwarc and Elmaghraby routine is restricted in its application to problems in which each integer variable must assume either the value 1 or 0. This restriction is shown by Szwarc and Elmaghraby to preclude certain nonbasic solutions to the linear programming problem from being uniquely optimal integer solutions; and this permits construction of a search process that systematically ignores such nonbasic solutions. The restriction to zero-one variables is not a serious limitation in principle, since reformulations exist[9] whereby any bounded integer programming problem can be reduced to this form. But the fact that these two methods are of a tree search variety and therefore require exponentially increasing time and memory requirements *is* a serious limitation for practical computation.

The "Stopped Simplex Method" of G. L. Thompson[10] is also a search routine, but utilizes a multidimensional search method that does not search the tree of possibilities in the usual manner. Instead, tests are made to show when enough search has been made so that a complete search of the entire tree is unnecessary. The memory requirements for the program are fixed in size and go up linearly with problem size.

The inadequacy of our simple classification for uses beyond the role of an expository device is revealed by the all-integer algorithm of F. Glover.[11] This algorithm proceeds from tableau to tableau by algebraic transitions that are evidently more elemental and flexible than the usual pivot operation, and progresses to an optimum solution through the generation of a sequence of successively greater lower bounds on the values of the variables in an optimum integer solution. This might be classed as a search technique. However it appears that the Gomory all-integer algorithm (a cutting plane method) can be represented as a special case of the Glover procedure. Thus we proceed to the discussion of cutting plane techniques with the realization that those methods may also be capable of interpretation as combinatorial search procedures.

[9] See [5, pp. 515–516].
[10] See [19].
[11] See [7].

Still another basically different approach to integer programming is contained in the recent paper[12] of R. E. Gomory, in which he considers very general types of "round-off" procedures to go from the continuous to the integer solution of a programming problem. Imbedded in that method is an auxiliary dynamic programming problem.

1.5. Solution Techniques: Cutting Plane Methods

While there are other conceptual designs for integer programming algorithms, most attention has centered on cutting plane methods.[13] These methods use a standard linear programming algorithm to locate a basic solution to the linear programming problem.[14] We shall call this solution the trial solution. If the trial solution is not integral, cutting plane methods generate and adjoin to the tableau a new constraint, called a cut, that is designed to destroy the feasibility of the trial solution while leaving undisturbed the feasibility of every integer solution. Each cycle of a cutting plane algorithm typically contains these two steps.

Preparatory to distinguishing among cutting plane algorithms, we note that the trial solution which is interdicted by the cut may or may not be the basic solution of the tableau to which the cut is adjoined. We illustrate three possible situations. Figure 1/1 depicts the case where the trial solution and the tableau basic solution are identical. The points in the triangle A were feasible before the cut was adjoined and are made infeasible by the cut. The points in region B remain feasible after the cut has been adjoined. A and B have similar interpretations in figure 1/2. Figure 1/2 illustrates the case where the cut intersects the tableau basic solution while eliminating the trial solution which is an extreme point adjacent to the tableau basic solution. Figure 1/3 is a schematic presentation of a third case, in which the tableau basic solution and the trial solution are distinct vertices and both solutions are infeasible with respect to the cut. The tableau basic solution, the trial solution, and the solution at point d are all assumed to satisfy primal optimality conditions, while none of these points, with the possible exception of point d, satisfies primal feasibility conditions.

It is desirable, of course, to organize the procedure so that the sequence of tableau basic solutions and trial solutions that are selected must lead to the optimum integral solution in a finite number of cycles. We may distinguish a cutting plane algorithm by the characteristics of the tableau basic solutions it utilizes and by the method used to generate cuts.

The first efforts to achieve a finite (cutting plane) algorithm shared a common method of locating the tableau basic and trial solutions but used different methods of generating cuts. For these methods the

[12] See [12].
[13] For a systematic discussion of cutting plane methods, see Ben-Isreal and Charnes [1, pp. 234 ff.] and Charnes and Cooper [3, pp. 698 ff.]. The Manne and Markowitz paper [17] is an early example of cutting plane methods.
[14] i.e., the linear programming problem which contains, in its set of solutions, all the solutions to the integer programming problem.

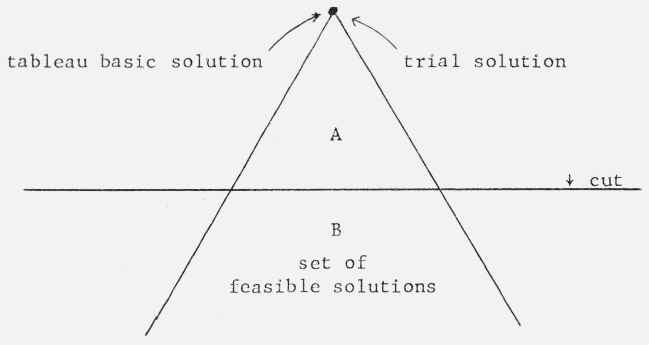


FIGURE 1/1.

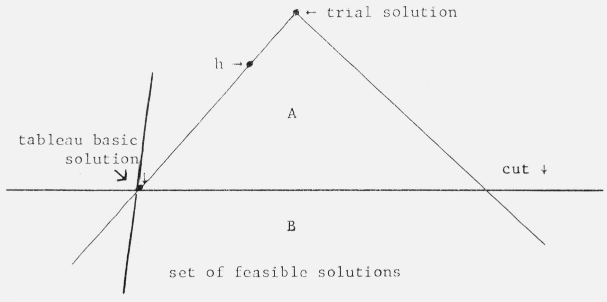


FIGURE 1/2.

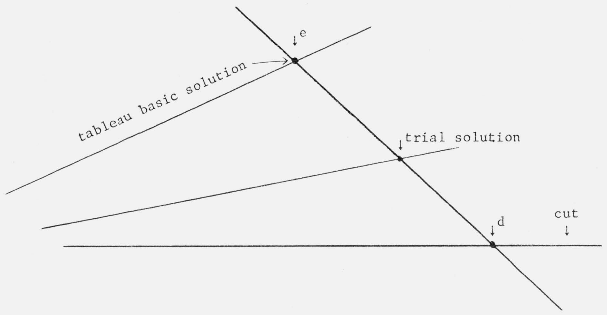


FIGURE 1/3.

trial solution was identical to the tableau basic solution, as in figure 1/1. The natural choice for the tableau basic solution was the optimal solution to the linear programming problem, since cutting the solution set back from the optimum solution would seem intuitively to lead to the integer optimum in a fairly direct fashion.

Several methods were[15] suggested for generating cuts—by Markowitz and Manne, by Dantzig, by Charnes and Cooper, and by Gomory.

The cut generation procedure developed by Gomory should be distinguished because he developed a proof of finiteness, and because most subsequent development of integer programming algorithms has been based on his work. A brief description of the cut-generation procedure of the Gomory algorithm—sometimes called the method of integer forms—will be given presently.

Subsequent to the development of the first Gomory algorithm, three other integer programming algorithms have been developed which employ a common method of generating cuts but differ with respect to the character of the tableau basic solutions. These algorithms are the Gomory all-integer algorithm, the direct algorithm developed by Ben-Israel and Charnes and the primal algorithm which is the subject of this paper. The Gomory all-integer algorithm uses the dual method to locate the trial and tableau basic solutions, while the direct algorithm and the primal algorithm use the (primal) simplex method. These three algorithms will all be described subsequently.

The optimum solution sought by any cutting plane method should have the following three properties. The solution must satisfy the usual linear programming tests for (i) optimality, (ii) feasibility, and in addition, (iii) the solution must be in integers. The cutting plane algorithms discussed here differ with respect to which of these properties characterize the tableau basic solutions generated by the algorithm *en route* to the optimal solution. The original Gomory algorithm generates a sequence of tableau basic solutions which are feasible and optimal but not integral before the solution generated by the final cycle of the algorithm. The Gomory all-integer algorithm generates a sequence of tableau basic solutions which are integral and optimal (i.e., dual feasible) but not (primal) feasible before the final solution. In contrast, our primal algorithm and the direct algorithm generate solutions which are integral and feasible but are not optimal before the final solution.

As we have noted figure 1/1 represents the typical relations among the cut, the trial solutions and the tableau basic solution in the original Gomory algorithm. Figure 1/2 represents these relations for the primal and the direct algorithms, while figure 1/3 represents these relations for the Gomory all-integer algorithm.

1.6. The Original Gomory Algorithm

The first step of the original Gomory algorithm is location of the optimal solution to the linear programming problem which contains in its solution set the solution to the given integer programming problem. Suppose we express the tableau which corresponds to this solution by the following matrix equation:

$$IX_B+AX_N=G, \tag{1.1}$$

where I is an m by m identity matrix, A is an m by n-m matrix of constants, G is an m by 1 vector of nonnegative constants, X_B is an m by 1 vector of variables, and X_N is an n-m by 1 vector of variables. The optimal solution is $X_B=G\geqslant 0$; $X_N=0$. If G only contains integral or zero components then the desired optimal solution to the integer programming problem has been attained. If G contains some nonintegral components, then a cut can be generated which will render the solution to (1.1) infeasible.

The following procedure is used to generate the required cut. To write the new equation we must

[15] Recently F. Glover has developed a very general formulation for cut generation. See [8] and Harris [14].

consider in detail a row (or equation) from (1.1) which has a fraction on the right side. Accordingly, let

$$x_r+\sum_j a_{r,j}x_j=g_r \tag{1.2}$$

be an equation from (1.1) where g_r is nonintegral, x_r is a component of X_B, and the summation ranges over all components x_j of X_N. In addition we require these definitions. For a given real number y, we define

$$_I[u]\equiv \text{the largest integer} \leq y.$$

We call $_I[y]$ the integer part of y. We symbolize the fractional part of y by

$$_f[y]\equiv y-{}_I[y]$$

We note that $y={}_f[y]+{}_I[y]$, and that $_f[y]$ is never negative and is equal to zero when y is an integer.

Now to write the Gomory cut we first select some row, such as (1.2), from the tableau (1.1), the only special characteristic of the row being its fractional component in the G vector. Then we use the data of that row, (1.2), to generate the following new equation:

$$s-\sum_j {}_f[a_{r,j}]\,x_j=-{}_f[g_r] \tag{1.3}$$

where s is a new slack variable, and the summation ranges over the indices of the nonbasic variables. This equation is adjoined to the system (1.1); the new s variable is inserted in the basis, which is thereby extended into one more dimension.

Gomory has proved that any feasible and integral solution to the system (1.1) must determine a nonnegative (and integral) value of s.[13] This proof holds for any system such as (1.1) independent of whether the basic solution associated with the identity matrix is optimal.

When the cut (1.3) is adjoined to (1.1) the basic solution is changed by letting $s=-{}_f[g_r]$, and is no longer feasible. A new cycle is initiated by solving for the optimal solution to the new linear programming problem generated by adjoining (1.3) to (1.1). The most convenient method of reoptimizing is the dual method, since the system that results from adjoining (1.3) to (1.1) is dual feasible and has one basic variable with a negative value.

Gomory proves that this algorithm will proceed to an optimal solution for a given integer programming problem in a finite number of cycles.[16]

1.7. The Gomory All-Integer Algorithm [17]

The preceding section discussed Gomory's method of integer forms. Dr. Gomory has also introduced another algorithm called the all-integer algorithm. There are several important differences between the Gomory all-integer algorithm and the original Gomory algorithm. In addition to the all-integer characteristic of this algorithm, there is a more general cut generation mechanism. The tableau basic solution and the trial solution coincide in the method of integer forms. In the all-integer algorithm they are distinct. In the method of integer forms the tableau basic solutions are optimal, feasible, and noninteger. In the all-integer algorithm the tableau basic solutions are integral, optimal—i.e., dual feasible—and (primal) infeasible.

Figure 1/3 can provide some insight into the mechanics of the all-integer algorithm. The tableau basic solution is integral and optimal and nonfeasible. The trial solution, which shares the line e with the tableau basic solution, is a solution that would become the basic solution after the execution of the usual dual method change of basis procedure. The trial solution is optimal and is typically nonintegral and nonfeasible. Instead of moving the tableau basic solution to the trial solution the cut is adjoined. The cut is so constructed that after it has been adjoined an ordinary dual method change of basis will move the solution to point d, which is integral and optimal, and may or may not be feasible.

This case is special: the trial solution need not be infeasible with respect to the cut and need not be on the same edge as point d and the tableau basic solution.

This algorithm requires that the all-integer property be present in the initial statement of the problem as a system of equations,—i.e., all the constants must be given as integers.[18] The means by which this property is preserved in subsequent tableaus are both simple and ingenious. If the pivot coefficient used in accomplishing the transition from a given to a subsequent tableau is equal to ± 1 (where the sign depends on whether the dual or the simplex method is being used) then the resulting tableau will be all-integer provided the initial tableau is all-integer. This conclusion follows from inspection of the formulas which describe the simplex method (or dual method) change of basis procedure.

To insure that a unit pivot coefficient is always used, the all-integer algorithm generates a cut in the course of each cycle of the algorithm. The cut generating mechanism and the method of employing it are such as to guarantee:

(i) That the cut can serve as the pivot row,

(ii) that the pivot coefficient (at the intersection of the cut row with the pivot column) always has the value -1, and

(iii) that the cut only contains integer constants.

Every prefinal tableau basic solution of the Gomory all-integer algorithm is dual feasible,[19] is associated with an all-integer tableau and has some primal infeasibility. Thus we may let (1.1) serve as a representation of such a tableau (less the criterion function information) if we assume all-integer data and that G is not $\geqslant 0$.

To provide a cut which satisfies the requirements listed above, in addition to preserving the feasibility

[13] See Gomory [9] and [10].
[16] See Gomory, [10].
[17] See Gomory, [11].

[18] This is not a significant limitation. See section 1.1.
[19] i.e., the solution would be optimal if there were no primal infeasibility.

of every integer solution to (1.1) while increasing the infeasibility of the tableau basic solution, the all-integer algorithm uses the following formulation:

$$s+\sum_{j} {}_I[a_{v,j}/\lambda]\cdot x_j+{}_I[1/\lambda]\cdot x_v={}_I[g_v/\lambda]. \qquad (1.4)$$

We shall presently discuss the determination of a value for the positive parameter λ. The constants in (1.4) with the subscript v are taken from some row v of (1.1) that has been selected as the source of the data for the cut (1.4). Gomory has proved[20] that for any row v and any positive λ, every feasible integer solution to (1.1)[21] determines a solution to (1.4) in which s is an integer $\geqslant 0$.

Thus to generate a specific cut from a specific tableau requires the selection of a source row v and the determination of a value for λ. Since the cut is to be used as the pivot row it is required by the dual method that ${}_I[g_v/\lambda]<0$. This requires that the source row be selected from among the rows i which have $g_i<0$. The value of λ is determined by two requirements: (a) the pivot coefficient must be -1, which can always be secured by making λ sufficiently large, and (b) the value of λ should be as small as possible—consistent with the satisfaction of (a)—in order to achieve as large as possible a change in the criterion function as a result of the subsequent pivot on the cut row.[22] Gomory provides an algebraic routine which will select λ so as to satisfy (a) and (b).

Roughly, then, each cycle of the Gomory all-integer algorithm consists of a cycle of the dual method in which the execution of the change of basis procedure is preceded by augmenting the system with the cut (1.4). This cut, once adjoined, will qualify as the pivot row and will have a pivot coefficient of -1. Gomory proves that this algorithm is finite and bases his proof on, among other things, the fact that

$${}_I[g_v/\lambda]<0.$$

This guarantees that each cycle will result in a finite "lexicographic" decrease (assuming the goal is maximization in the primal problem) in the column vector $\begin{bmatrix} z \\ G \end{bmatrix}$ where the scalar quantity z is the criterion value of the basic solution of (1.1) and G is the right-hand side in (1.1).

1.8. An Analogous Primal Algorithm: Motives and Problems

A natural sequel to the Gomory all-integer algorithm is the development of an integer programming algorithm that is related to the simplex method as the Gomory all-integer algorithm is related to the dual method. Most of the details of such a procedure can be derived, as we shall show presently, in a straightforward way from the Gomory all-integer algorithm. There are several motives which might propel such a development. We shall be content here with a brief discussion of some of these motives.

In the development of linear programming solution techniques, much progress in the development of special algorithms has been based on the joint existence of the primal (simplex) and dual methods, which have provided the foundation for a variety of composite algorithmic procedures. A primal counterpart to the Gomory all-integer algorithm might open the way to a class of composite integer programming algorithms. And such a class might well contain efficient algorithms based on special problem structures.

Additionally there are situations that intrinsically favor a procedure which proceeds to an optimum solution through a sequence of primal-feasible tableaus. Development of interpretative connections between the mathematical operations and the real-world counterparts of the elements of the mathematical system may be easier with a primal-feasible system. Where calculations cannot be continued until a known optimum solution is obtained, the current primal-feasible basic solution—which in an all-integer system is also integral—may be useful as a "good" answer—capable of execution and possibly controlled by suitable bounding techniques—to the real-world decision problem. Finally it may be convenient and useful in many cases to express good solutions—achieved by heuristic or other means—as initial basic solutions to an integer programming problem. Such advanced starts should reduce the calculation required to solve the integer programming problem. Thus a primal integer programming algorithm might provide the most convenient means to test, calibrate, and improve the power of a heuristic solution technique.[23]

As indicated, the procedures of the Gomory all-integer algorithm provide an easily followed model for the development of the details of a primal algorithm. The integer programming problem would be given in terms of an initial primal-feasible all-integer system of equations with an all-integer criterion function. A typical cycle of such an algorithm would include the procedures of a cycle of the simplex algorithm. A cycle would also include adjoining a Gomory cut—generated from the formula (1.4)—in such a way that the cut qualifies as the pivot row and has a pivot coefficient with value 1.

For the purposes of exposing critical problems with a minimum of extraneous detail we shall outline a simplified algorithm. Suppose the following problem has been given:

$$\text{maximize} \qquad \sum_{j=1}^{n} \gamma_j x_j \qquad (1.5)$$

[20] See Gomory [11].
[21] The proof does not depend on the properties of the basic solution (such as primal infeasibility) in (1.1).
[22] Gomory suggests the possibility of using other rules for determining λ. See [11, p. 198].

[23] The direct algorithm of Ben-Israel and Charnes, discussed below, provides a convenient method of incorporating an advanced start as a basic solution. The advanced start is expressed as a solution to the "auxiliary problem" (defined below). This will lead, by a simple and direct procedure, to a basic solution that corresponds to the advanced start.

subject to

$$x_i + \sum_{j=m+1}^{n} a_{i,j} x_j = g_i \geqslant 0, \qquad i=1, 2, \ldots, m. \quad (1.6)$$

$$x_j \geqslant 0 \text{ and integral}, \qquad j=1, 2, \ldots, n.$$

We assume all the γ_j, $a_{i,j}$ and g_i constants are given as integers or zeros. We associate the basic solution, $x_i=g_i$, $i<m+1$, $x_j=0$, $m+1 \leqslant j \leqslant n$, with this system of equations. We assume this solution is not optimal, i.e., for at least one x_j, $m+1 \leqslant j \leqslant n$,

$$c_j = \gamma_j - \sum_{i=1}^{m} a_{i,j}\gamma_i > 0. \quad (1.7)$$

In (1.7) the term c_j has the meaning of the term usually symbolized by $c_j - z_j$ in the linear programming literature. The γ's in (1.7) are original criterion coefficients and it is assumed for notational convenience that the variable that is basic in row i is x_i.

Now we describe the events of one cycle of a simplified primal algorithm, which will achieve all the features specified in the previous paragraph. We shall call this the *rudimentary primal algorithm.*

I. Select an incoming variable x_J according to the usual simplex method criterion for making that choice.

II. Select as the source row, v, the row which would be the (natural) pivot row[24] given the prior selection of x_J as the incoming variable.

III. Set $\lambda = a_{v,J}$, where $a_{v,J}$ is the natural pivot coefficient, implied by the selection of x_J.

IV. Adjoin the Gomory cut (1.4) to the system. The slack variable s will enter the basis and have the initial value ${}_I[g_v/a_{v,J}]$. (We shall assume that $a_{v,J}>1$ and that therefore x_v does not appear in the cut. In cases where $a_{v,J}=1$, we shall assume that no cut need be adjoined.)

V. Execute the usual simplex change of basis procedure with the cut serving as pivot row and the column associated with x_J serving as the pivot column.

It is easily demonstrated that this procedure does provide a cut which qualifies as the pivot row and has the pivot coefficient equal to 1.

The central difficulty with this procedure is in proving that it is finite. The core of the difficulty arises from the possible occurrence of

$$g_v/a_{v,J} < 1 \quad (1.8)$$

for every permissible selection, J, of a pivot column (and the natural pivot row, v, determined by the prior selection of x_j). When (1.8) is true, the cut has

$${}_I[g_v/a_{v,J}] = 0. \quad (1.9)$$

Thus s is placed in the basis at a zero level and, as a result of executing the pivot operation, x_J replaces s in the basis at a zero level. The basic solution in the tableau that results from such a cycle has been changed (in relation to the previous tableau) in composition and dimension but not with respect to the value of any basic or nonbasic variable. Thus no change occurs in the G column or in the criterion value of the basic solution.

The possibility that (1.8) may occur precludes a direct (and comparatively simple) proof of finiteness analogous to that developed for the Gomory all-integer algorithm. It is also impossible to remedy this difficulty by a straightforward appeal to the degeneracy theory developed for linear programming situations. Those procedures, and the arguments that prove they will avoid cycling, are based on the assumption of a convex polyhedron of solutions which has a finite *and fixed* number of extreme points. The nature of the cutting plane methods is to systematically alter and frequently to increase the number of extreme points on the set of solutions.

Two primal algorithms have been developed which share, in a general way, the procedures and problems which have been described and discussed in this section. They are (1) the direct algorithm developed by A. Ben-Israel and A. Charnes and (2) the primal algorithm which is the subject of this paper. The essential difference between these two procedures is their contrasting response to the problem posed by the possibility (1.8).

Some summary intuitive notion of the characteristics of these primal algorithms may be developed by the following brief discussion of figure 1/2. Here we have a simple representation of the typical situation for the primal algorithms. Since the tableau basic solution is feasible and integral with these algorithms, this solution must be feasible with respect to the cut. (Neither of the Gomory algorithms shares this characteristic.) The trial solution is a (typically) nonintegral solution must be feasible with respect to the cut, procedure were carried out without adjoining the cut. The cut is shown intersecting the tableau basic solution, which is the geometric equivalent of the occurrence of (1.8). If (1.8) is not true then the cut will not intersect the tableau basic solution, but will intersect instead some point (with integral coordinates) such as h in figure 1/2, or even possibly the trial solution point. In these circumstances the change of basis procedure "moves" the basic solution along the edge (connecting the tableau basic solution and the trial solution) to the point where the cut and the edge intersect.

In the following discussion we shall see that the direct algorithm systematically avoids the situation where (1.8) is true and the cut intersects the tableau basic solution, while the primal algorithm permits this circumstance.

1.9. The Ben-Israel and Charnes Direct Algorithm

The direct algorithm of Ben-Israel and Charnes will now be reviewed. The direct algorithm calls for a procedure essentially identical to the rudimentary primal algorithm described in the previous section

[24] The natural pivot row is a row which minimizes the pivot ratio, $g_i/a_{i,J}$, over the set of rows, i, which have $a_{i,J}>0$. Since we shall make the natural pivot row the source row, the index v may represent both concepts. In part II, we shall restrict v to identifying the source row.

whenever a nonbasic variable x_J exists such that

$$g_v/a_{v,J} \geqslant 1, \tag{1.10}$$

and x_J would improve the criterion value of the solution by becoming positive,—i.e., $c_J = \gamma_J - \sum_i a_{i,J}\gamma_i > 0$. Any nonbasic variable which satisfies those two conditions may be selected as the incoming variable. If no nonbasic variable satisfies both conditions, but some nonbasic variable exists which would improve the solution by becoming positive, then (1.8) is true and the direct algorithm invokes a special procedure. In these circumstances it is necessary to solve an auxiliary problem,[25] the goal of which is the generation of a new nonbasic variable—which we shall label x^0.

To describe the necessity for and purpose of the auxiliary problem, let (1.6) represent the constraints of the current tableau, and let (1.5), represent the criterion function. Also suppose we have a value

$$c_j = \gamma_j - \sum_{i=1}^{m} a_{i,j}\gamma_i \tag{1.11}$$

(for each nonbasic variable x_j).

Now if we let (1.6) be represented by the equivalent matrix equation

$$IX_B + AX_N = G$$

where X_B contains the x_i with $1 \leqslant i < m+1$, while X_N contains x_j with $m+1 \leqslant j \leqslant n$, and the coefficients of A and G are given by (1.6), a succinct algebraic test is available to distinguish (1.8) from (1.10). Let A_l symbolize a column of A and let $C+$ denote the set of all such columns A_l for which $c_l > 0$. If (1.10) is true then

$$A_l \leqslant G \qquad \text{for some } A_l \epsilon C+. \tag{1.12}$$

If this were not the case then in particular $A_J \leqslant G$ would be false, which would imply the existence of a row r, for which $a_{r,J} > g_r$. Since the smallest pivot ratio for column J is, by (1.10), $\geqslant 1$, this must contradict one of the following: (i) $G \geqslant 0$, (ii) (1.10), or (iii) the definition of the row v in (1.10). Thus (1.10) implies (1.12). Accordingly, if (1.12) is not true, then (1.10) must be false, which implies (1.8) is true.

In these terms the solution of the auxiliary problem is called for whenever (1.12) is not true. The goal of the auxiliary problem is the generation of a new column A^0 and a corresponding new nonbasic variable x^0, to be adjoined to A and X_N respectively. To solve the auxiliary problem, A^0 must make (1.12) true. Thus it is required that $A^0 \leqslant G$ and $A^0 \epsilon C+$. Finally A^0 is required to be a nonnegative integer combination of the existing columns of A.

Thus we may state the goal of the auxiliary problem as follows: find a vector ϕ, with typical component ϕ_l and the same dimension as X_N, which satisfies

$$\sum_l A_l\phi_l = A^0 \leqslant G \tag{1.13}$$

$$\sum_l c_l\phi_l = c_0 > 0 \tag{1.14}$$

$$\phi_l \geqslant 0 \text{ and integral for all } l. \tag{1.15}$$

We note that the auxiliary problem requires an integer solution and in general it will not be a priori evident that a solution does or does not exist for the auxiliary problem. If the auxiliary problem can be (and is) solved, the column A^0 is adjoined to A and a new variable x^0 is identified with this column. Clearly x^0 qualifies as the incoming variable which satisfies (1.10) as well as the usual requirement that $c_0 > 0$. If the auxiliary problem cannot be solved—i.e., if no vector $\phi = (\phi_1, \ldots, \phi_l, \ldots, \phi_{\text{last}})$ exists which satisfies (1.13), (1.14), and (1.15)—then the current basic solution is optimal. Ben-Israel and Charnes provide a proof of this proposition.

Thus the direct algorithm only executes cycles in which $c_J > 0$ and

$$1 \leqslant {}_I[g_v/a_{v,J}] \leqslant g_v/a_{v,J}.$$

This condition implies that each cycle of the algorithm results in at least a unit increase in the criterion function value of the solution. A proof of finiteness follows directly from the assumed boundedness of the given problem.[26]

The central weakness of the direct algorithm is the absence of a general method for solving the auxiliary problem or for making positive identification of every situation where no solution exists.[27] Ben-Israel and Charnes discuss several devices that will simplify or solve the auxiliary problem in particular cases. The strength of this algorithm, aside from the fact that it is a primal procedure, is in the efficiency of the procedure when applied to problems where special structure permits solution of the auxiliary problem in a simple and reliable fashion. In section 1.11 we shall present an example problem which is solved with the direct algorithm.

1.10. The Primal Algorithm

Now we shall provide a brief description of some of the distinguishing characteristics of the primal algorithm. The primal algorithm pursues an alternative course to that taken by the direct algorithm. The primal algorithm avoids whenever possible a selection of the incoming variable x_J which will lead to (1.8).

[25] The primal algorithm contains no such procedural detour.

[26] Charnes and Cooper in [3, chs. VII and XII] provide the theoretical foundation for assuming a bounded solution set.

[27] ". . . the auxiliary problem may be no smaller in size than the original problem. . . . However it may often be completely transparent, particularly for special structures. We will treat the auxiliary problem as a "black box" in presenting . . . a direct algorithm for integer programming. Although no simple systematic way to resolve this difficulty is given, we will show that in spite of it the direct algorithm is effective at getting feasible "close to optimal" solutions, or dually feasible integer solutions to which one may apply Gomory's algorithm." Quoted from Ben-Israel and Charnes [1, pp. 249–250].

If no other choice is available (and the primal optimality conditions are not satisfied) then an incoming variable which implies (1.8) is selected and a cut is generated which has

$$s = {}_I[g_v/a_{v,J}] = 0. \tag{1.16}$$

Thus the major task associated with this algorithm is establishing a guarantee that (1.16) will occur for at most a finite sequence of successive cycles.

In the detailed description of the algorithm given in part II, several departures are taken from the rudimentary primal algorithm described in section 1.8. There elaborations, while consistent with the goals behind the rudimentary algorithm, complicate and constrain the selection of the incoming variable and of the source row. These elaborations serve two general purposes: to guarantee a finite algorithm and to avoid[28] arbitrary restriction of choice beyond that required to attain a finite algorithm. The rudimentary primal algorithm may be regarded as a prototype for our primal algorithm and can usefully serve as a comparatively simple vehicle for introducing a procedural outline of the primal algorithm, provided the necessity of subsequent elaboration is borne in mind. It has been our intent to provide in part I a description of some of the major characteristics of the primal algorithm, and to discuss these characteristics in terms of contrasts and similarities to existing integer programming techniques, particularly the Gomory all-integer algorithm and the Ben-Israel and Charnes direct algorithm, which are the "closest relatives" to the primal algorithm. In the following section, which concludes part I, we present two example problems with solutions by the rudimentary primal algorithm. We also present a solution to the second problem by the direct algorithm.

1.11. Exemplification

In this section two small example problems are solved by the rudimentary primal algorithm. Both examples are chosen because they have been used elsewhere to illustrate the operation of some of the other integer programming algorithms discussed in this chapter. An interested reader will therefore be able to make comparisons of the examples given here with the referenced source of the problem.

Problem #1. The first problem was used by Charnes and Cooper[29] to illustrate the operation of Gomory's method of integer forms. The problem is

$$\begin{array}{ll} \max & 3x + y \\ \text{subject to} & 2x + 3y \leq 6 \\ & 2x - 3y \leq 3 \\ & x, y \geq 0 \\ & x, y \text{ to be integer.} \end{array}$$

[28] This goal is not perfectly satisfied as will be observed from part II. In addition we specifically disavow any suggestion that the procedures of the primal algorithm are necessary to secure finiteness even though they are sufficient. See [21].

[29] See Charnes and Cooper [3, pp. 702–709].

Converting this problem to one in equation form we adjoin two slack variables, t_1 and t_2, and obtain the following initial tableau. The first row contains the $-c_j$ values, or in the usual linear programming terminology, $z_j - c_j$ values. P_0 denotes the constant column.

(T1)

	P_0	x	y	t_1	t_2
Z	0	-3	-1	0	0
t_1	6	2	3	1	
t_2	3	2	-3		1

According to the usual simplex criteria both x and y are eligible candidates for the incoming variable. Either of these variables would lead to $g_v/a_{v,J} \geq 1$. If $x_J = x$ this quantity is 3/2; and if $x_J = y$, $g_v/a_{v,J} = 6/3$. Thus either x or y may be selected as the incoming variable. We shall arbitrarily select y. Then a cut must be adjoined to (T1) using the formula (1.4) with the natural pivot row (t_1 row) serving as the source row v and with $\lambda = a_{v,J} = 3$. The resulting cut is

$${}_I\left[\frac{6}{3}\right] = s_1 + {}_I\left[\frac{2}{3}\right]x + {}_I\left[\frac{3}{3}\right]y + {}_I\left[\frac{1}{3}\right]t_1.$$

or

$$2 = y + s_1 \tag{1.17}$$

The tableau (T1) with (1.17) and the new variable s_1 adjoined becomes

(T1′)

		P_0	x	$\overset{\downarrow}{y}$	t_1	t_2	s_1
	Z	0	-3	-1	0	0	0
*	t_1	6	2	3	1	0	0
	t_2	3	2	-3	0	1	0
$\leftarrow$	s_1	2	0	1	0	0	1

Here y is designated as the incoming variable and s_1 as the outgoing variable. We shall repeat the conventions employed in (T1′) in subsequent tableaus: the arrows designate the incoming and outgoing variables, the (*) designates the natural pivot row and the source row, and the new cut equation appears below a horizontal dashed line. Since (T1′) contains (T1) we shall in subsequent tableaus only present the primed version which contains the cut and designates the source row.

When the indicated simplex method pivot operation is applied to (T1′) the result is (T2), which is contained in (T2′).

	P_0	↓ x	y	t_1	t_2	s_1	s_2
Z	2	−3	0	0	0	1	0
* t_1	0	2	0	1	0	−3	0
t_2	9	2	0	0	1	3	0
y	2	0	1	0	0	1	0
←s_2	0	1	0	0	0	−2	1

(T2′)

The only eligible incoming variable in (T2) is x. If $x=x_J$ then the t_1 row is the natural pivot. This row becomes the source row v with $a_{v,J}=\lambda=2$. The new cut, which is the bottom row of (T2′), contains as coefficients (in the P_0, x, y, t_1, t_2, and s_1 columns) the integer parts of the quotients that result from dividing the t_1 row by 2. The new variable s_2 is of course the slack associated with the new cut. The simplex change-of-basis procedure is applied to (T2′) to yield (T3).

	P_0	x	y	t_1	t_2	↓ s_1	s_2
Z	2	0	0	0	0	−5	3
←t_1	0	0	0	1	0	1	−2
t_2	9	0	0	0	1	7	−2
y	2	0	1	0	0	1	0
x	0	1	0	0	0	−2	1

(T3)

In tableau (T3) only s_1 qualifies as an incoming variable. The natural pivot row is the t_1 row and the natural pivot coefficient is equal to 1. Thus no cut need be generated.[30] Pivoting as indicated by the arrows in (T3) yields (T4).

	P_0	x	y	t_1	t_2	s_1	↓ s_2	s_3
Z	2	0	0	5	0	0	−7	0
s_1	0	0	0	1	0	1	−2	0
* t_2	9	0	0	−7	1	0	12	0
y	2	0	1	−1	0	0	2	0
x	0	1	0	2	0	0	−3	0
←s_3	0	0	0	−1	0	0	1	1

(T4′)

In (T4) s_2 is the only permissible incoming variable. The natural pivot row is the t_2 row which becomes the source row v for generating the cut, and $a_{v,J}=\lambda=12$. Applying the formula (1.4) yields the cut in the bottom row of (T4′). Then s_2 replaces s_3 in the basis to yield (T5).

	P_0	x	y	↓ t_1	t_2	s_1	s_2	s_3	s_4
Z	2	0	0	−2	0	0	0	7	0
s_1	0	0	0	−1	0	1	0	2	0
* t_2	9	0	0	5	1	0	0	−12	0
y	2	0	1	1	0	0	0	−2	0
x	0	1	0	−1	0	0	0	3	0
s_2	0	0	0	−1	0	0	1	1	0
←s_4	1	0	0	1	0	0	0	−3	1

(T5′)

[30] See step IV of the rudimentary algorithm in section 1.8.

In (T5) there is again only one permissible incoming variable: t_1. The associated natural pivot row is the t_2 row. This row is made the source row and $\lambda=a_{v,i}*=5$. Applying (1.4) we get the cut row—the s_4 row in (T5′). When t_1 has replaced s_4 in the basis the result is (T6), which satisfies the primal optimality conditions. Thus the basic solution in (T6) is optimal.

	P_0	x	y	t_1	t_2	s_1	s_2	s_3	s_4
Z	4	0	0	0	0	0	0	1	2
s_1	1	0	0	0	0	1	0	−1	1
t_2	4	0	0	0	1	0	0	3	−5
y	1	0	1	0	0	0	0	1	−1
x	1	1	0	0	0	0	0	0	1
s_2	1	0	0	0	0	0	1	−2	1
t_1	1	0	0	1	0	0	0	−3	1

(T6)

In this example problem we have not dropped s variables which reenter the basis—e.g., s_1 and s_2 in (T6). We shall discuss in appendix A the conditions under which such s variables can be dropped from the system. The possibility of eliminating such variables is of course important to limiting the size of the system.

We may note that in only two cycles of the algorithm in the above problem was there a nonzero entry in the P_0 column of the (ultimate) pivot row. Hence we may conclude that a different course would have been followed had the same problem been solved with the Ben-Israel and Charnes direct algorithm. In this regard note tableau (T2)—i.e., (T2′) without the s_2 row. The direct algorithm would not have chosen x as the incoming variable since the x column in (T2) is *not* less than or equal to the P_0 column. Instead the direct algorithm would have instituted a search (the auxiliary problem) for a (nonnegative) integer combination of the nonbasic columns of (T2) which is less than P_0 and has a negative first (Z row) component. Such a combination exists, namely s_1+x. The Ben-Israel algorithm would create a new variable, say t_3, defined by $t_3=s_1+x$ and adjoin the appropriate column to (T2). This new variable t_3 would be designated as the incoming variable and the cycle would be accomplished by executing steps II–V of the rudimentary primal algorithm.

To provide a more comprehensive example of the contrast between the rudimentary primal algorithm and the direct algorithm we shall present solutions of a second example problem by both algorithms.

Problem #2. We provide below, in a quotation,[31] the statement and solution of a problem by the direct algorithm of Ben-Israel and Charnes. This problem has also been solved elsewhere by the original Gomory algorithm.[32] The statements, near the end of the quotation, which argue for the optimality of basic solution of tableau (4) merely assert that only the three listed combinations would have a negative first component, and that none of these combinations is less than or equal to the stipulations column. Hence there is no solution to the auxiliary problem.

[31] The quotation is from [1, pp. 256–257.]
[32] See [10, pp. 297–299.]

$$\begin{aligned}
&\text{Max } z = 3x_1 - x_2\\
&3x_1 - 2x_2 \leqq 3\\
&-5x_1 - 4x_2 \leqq -10\\
&2x_1 + x_2 \leqq 5\\
&x_i \text{ integer} \geqq 0\\
&i = 1, 2
\end{aligned}$$

We rewrite the constraints as

$$\begin{aligned}
3x_1 - 2x_2 + x_3 &= 3\\
-5x_1 - 4x_2 + x_4 - x_6 &= -10\\
2x_1 + x_2 + x_5 &= 5\\
x_i \text{ integer} &\geqq 0\\
i &= 1, \ldots, 6
\end{aligned}$$

where x_3, x_4, x_5 are slacks, and x_6 is a variable bearing a heavy penalty.

In this example we omit the unit vectors in all the tableaus.

(1)

	P_0	P_1	↓ P_2	P_4
	$-10M$	$-5M-3$	$-4M+1$	M
P_3	3	3	-2	0
* P_6	10	5	4	-1
P_2	5	2	1	0
$\leftarrow S_1$	2	1	1	-1

(2)

	P_0	↓ P_1	P_4	S_1
	$-2M-2$	$-M-4$	$-3M+1$	$4M-1$
* P_3	7	5	-2	2
P_6	2	1	3	-4
P_5	3	1	1	-1
P_2	2	1	-1	1
$\leftarrow S_2$	1	1	-1	0

(3)

	P_0	P_4	S_1	S_2	↓ $P_7 = 3P_4 + 2S_1 + 3S_2$
	$-M+2$	$-4M-3$	$4M-1$	$M+4$	$-M+1$
P_3	2	3	2	-5	-2
$\leftarrow P_6$	1	4	-4	-1	1
P_5	2	2	-1	-1	1
P_2	1	0	1	-1	-1
P_1	1	-1	0	1	0

(4)

	P_0	P_4	P_6	S_1	S_2
	1	-7	$M-1$	3	5
P_3	4	11	2	-6	-7
P_7	1	4	-1	-4	-1
P_5	1	-2	-1	3	0
P_2	2	-4	1	-3	-2
P_1	1	-1	0	0	1

Tableau (4) is optimal since it is case (A2) [see above, p. 222]. This is easy to check, since here one has to check only three nonnegative integer combinations of the nonbasic columns with $c_k > c_B^T\, y_{Bk}$, namely

$$P_4 + S_1,\ P_4 + 2S_1,\ P_4 + S_2$$

and none of them is such that

$$y_{Bo} \geqq y_{Bk}$$

The optimal solution is

$$\begin{aligned}
x_1 &= 1\\
x_2 &= 2\\
x_3 &= 4\\
x_4 &= 3 \text{ (this follows from } x_7 = 1\\
x_5 &= 1 \text{ and the definition of } P_7)\\
x_6 &= 0
\end{aligned}$$

with value $z = 1$.

The primal algorithm could yield an identical procedure to that followed by the direct algorithm through the generation of tableau (3). In tableau (3) if P_4 is selected as the incoming variable, $g_v/a_{v,J} = 1/4 < 1$ results. This is the circumstance that distinguishes the procedure of the two algorithms. The direct algorithm evokes the auxiliary problem in this situation. The incoming variable P_7 in tableau (3) is the result of successfully solving the auxiliary problem.

The primal algorithm simply makes P_4 the incoming variable. The resulting cut has a zero in the P_0 column, which is the circumstance the direct algorithm avoids.

The solution of this problem by the primal algorithm is recorded in the sequence of tableaus (T3′) through (T10) below.

We have dropped s variables when they have reentered the basis. Following the practice of Ben-Israel and Charnes, we have omitted the basis columns in all these tableaus.

(T3′)

	P_0	↓ P_4	s_1	s_2
Z	$-M+2$	$-4M-3$	$4M-1$	$M+4$
P_3	2	3	2	-5
* P_6	1	4	-4	-1
P_5	2	2	-1	-1
P_2	1	0	1	-1
P_1	1	-1	0	1
$\leftarrow s_3$	0	1	-1	-1

(T4′)

				↓
	P_0	s_3	s_1	s_2
Z	$-M+2$	$4M+3$	-4	$-3M+1$
P_3	2	-3	5	-2
* P_6	1	-4	0	3
P_5	2	-2	1	1
P_2	1	0	1	-1
P_1	1	1	-1	0
P_4	0	1	-1	-1
←s_4	0	-2	0	1

(T5′)

		↓		
	P_0	s_3	s_1	s_4
Z	$-M+2$	$-2M+5$	-4	$3M-1$
P_3	2	-7	5	2
* P_6	1	2	0	-3
P_5	2	0	1	-1
P_2	1	-2	1	1
P_1	1	1	-1	0
P_4	0	-1	-1	1
←s_5	0	1	0	-2

(T6′)

				↓
	P_0	s_5	s_1	s_4
Z	$-M+2$	$2M-5$	-4	$-M+9$
P_3	2	7	5	-12
P_6	1	-2	0	1
P_5	2	0	1	-1
P_2	1	2	1	-3
* P_1	1	-1	-1	2
P_4	0	1	-1	-1
←s_6	0	-1	-1	1

(T7)

			↓	
	P_0	s_5	s_1	s_6
Z	$-M+2$	$M+4$	$-M+5$	$M-9$
P_3	2	-5	-7	12
←P_6	1	-1	1	-1
P_5	2	-1	0	1
P_2	1	-1	-2	3
P_1	1	1	1	-2
P_4	0	0	-2	1

Tableau (T7) is noteworthy in several respects. In contrast to the preceding tableaus, the incoming variable has a natural unit pivot. Thus no cut is required. Rows P_6 and P_1 are "tied" as candidates for the natural pivot row. Here we arbitrarily choose the P_6 row, and since the pivot row has a positive quantity in the P_0 column, the next tableau, (T8), must have a changed—and improved—basic solution.

(T8′)

				↓
	P_0	s_5	P_6	s_6
Z	-3	9	$M-5$	-4
* P_3	9	-12	7	5
s_1	1	-1	1	-1
P_5	2	-1	0	1
P_2	3	-3	2	1
P_1	0	2	-1	-1
P_4	2	-2	2	-1
←s_7	1	-3	1	1

Note in (T8′) that again the pivot row, this time the cut, has a positive entry in the stipulations column. This implies a new and improved solution will occur in (T9).

(T9′)

		↓		
	P_0	s_5	P_6	s_7
Z	1	-3	$M-1$	4
P_3	4	3	2	-5
s_1	2	-4	2	1
* P_5	1	2	-1	-1
P_2	2	0	-1	-1
P_1	1	-1	0	1
P_4	3	-5	3	1
s_6	1	-3	1	1
←s_8	0	1	-1	-1

(T10)

	P_0	s_8	P_6	s_7
Z	1	3	$M-4$	1
P_3	4	-3	5	-2
s_1	2	4	-2	-3
P_5	1	-2	1	1
P_2	2	0	-1	-1
P_1	1	1	-1	0
P_4	3	5	-2	-4
s_6	1	3	-2	-2

We note that the optimal solution is first attained in (T9′) but that the transition to (T10) is required to prove that fact.

From this example we observe that the direct algorithm achieves its objective with greater dispatch than does the primal algorithm. This fact would suggest the advantage of combining the two procedures to achieve the efficiency (in certain circumstances) of the direct algorithm combined with the guaranteed finiteness of the primal algorithm.

Part II. Description of the Algorithm

2.1. Introduction

Our goal in part II is to provide a comprehensive and explicit statement of the primal algorithm. The details of the procedure are presented in sections 2.3 through 2.8. Section 2.9 contains a flow-chart summary of the algorithm. Some notational conventions are explained in section 2.2.

2.2. Notational Conventions and Assumptions

The assumptions and notational conventions listed below will be employed here and in part III.

1. We assume[1] a given, bounded, and solvable integer programming problem which, at some stage can be written in matrix terms as

maximize

$$C \cdot X_N \tag{2.1}$$

subject to

$$IX_B + AX_N = G \geqslant 0$$

$$X_B, X_N \geqslant 0 \text{ and integral.}$$

All the constants in (2.1) are assumed to be integers. I is an m by m identity matrix. A is an m by $n-m$ matrix. G is an m by 1 vector and C is a 1 by $n-m$ vector. X_B is an m by 1 vector of basic variables, and X_N is an $n-m$ by 1 vector of nonbasic variables.

The system (2.1) may or may not be the original or given form of the constraints. In any event let γ_r represent the criterion coefficient associated in the original statement of the problem, with a typical variable x_r, which may be a component of either X_B or X_N in (2.1). We note that the components of C correspond (one-to-one) to the components of X_N. The typical component c_j of C is related to the original criterion coefficient by

$$c_j = \gamma_j - \sum_i \gamma_i \cdot a_{i,j}, \tag{2.2}$$

in which the summation index i ranges over the set of all rows of A in (2.1), and γ_i is the original criterion coefficient of the basic variable associated with the column of I which has a 1 in row i.

The systems which evolve from (2.1) as a result of our attempt to solve (2.1) will contain new (slack) variables introduced into the system as part of new equations. To designate a scalar variable from (2.1), or from a successor system to (2.1) we shall frequently use the symbol u_i when it is desirable to avoid distinguishing the variable either as a scalar component of X_B or X_N in (2.1) or as a (slack) variable generated at some later stage.

The system (2.1) contains no slack variables created as the result of adjoining Gomory cuts to the system. Accordingly we may regard (2.1) as a representation of the original system of equations. We shall see subsequently that (2.1) may also serve as the representation of a tableau that follows a transition[2] cycle. For a more general representation of any tableau we employ the following notation

$$I^{(k)} \cdot U_B + A^{(k)} \cdot U_N = G^{(k)} \tag{2.1a}$$

where $I^{(k)}$ is a $m^{(k)}$ by $m^{(k)}$ identity matrix; A is a $m^{(k)}$ by $n^{(k)} - m^{(k)}$ matrix; $G^{(k)}$ is a $m^{(k)}$ by 1 vector, and U_B and U_N are variable vectors of appropriate dimensions. The interpretations made of (2.1a) differ from (2.1) in several respects. The variable designation U, instead of X means that the components, u_j, of U in (2.1a) may represent either structural variables of the given problem (x variables such as appear in (2.1)) or slack variables of Gomory cuts (s variables) which have been adjoined to the system. The tableau designator k appears as a superscript in (2.1a). In the following elaborations of (2.1a) we shall drop this superscript in the interest of less cluttered notation and with the understanding that the suppression of the superscripts does not suggest a general absence of change in the constants of the system (2.1a) as a function of the cycle or tableau index k.

We shall let (2.1a) represent either the original given tableau or any subsequent tableau which is generated in the course of solving the problem. We rely on the boundedness of the given problem, and the fact that the procedures of the primal algorithm do nothing to enlarge the original solution set, as the foundation for a system of subsidiary constraints to (2.1a). These constraints are

$$I \cdot U_B \leqslant G_{LB},$$

$$I \cdot U_N \leqslant G_{LN},$$

or

$$I \cdot H_B + I \cdot U_B = G_{LB} > 0 \tag{2.1b}$$

$$I \cdot H_N + I \cdot U_N = G_{LN} > 0 \tag{2.1c}$$

where the vectors H_B, and G_{LB} have the same dimensions as U_B, while H_N and G_{LN} have the same dimensions as U_N. The identity matrices have appropriate dimensions. G_{LB} and G_{LN} are vectors of positive integer constants. H_B and H_N are vectors of nonnegative variables. The vectors G_{LB} and G_{LN} are limits respectively on the values of U_B and U_N. The

[1] The theory which permits the assumption of boundedness to be made generally for linear programming problems is given by Charnes and Cooper [3, pp. 187–191]. Comparatively minor modifications are required to apply this theory to the primal integer programming situation. The assumption of solvability is also based on linear programming precedents. If an initial feasible basis is not available artificial variables may be adjoined to provide an initial basis. A "Phase—I" problem is then solved with the objective of securing a basic solution in which the sum of the artificial variables is minimized. If the optimal sum of artificials is zero feasibility is assured; otherwise there is no feasible solution.

[2] Transition cycles will be defined below; essentially they are cycles in which $g_v/a_{v,J} \geqslant 1$ and $c_J > 0$.

typical component, h_d, of either H_N or H_B is simply a slack variable representing the amount by which the current value of u_d falls short of its limiting value g_{Ld}.

The values of the components of G_{LB} and G_{LN} are specified so as to insure that constraints (2.1b) and (2.1c) are redundant in this sense; every feasible solution to (2.1a) is also feasible for (2.1b) and (2.1c). We may also express (2.1b) in terms of U_N by substitution from (2.1a). The result is

$$I \cdot H_B - A \cdot U_N = G_{LB} - G. \qquad (2.1d)$$

Now we may combine (2.1a) (2.1c) and (2.1d) in the following system

$$\begin{bmatrix} I & & & A \\ & I & & -A \\ & & I & I \end{bmatrix} \begin{bmatrix} U_B \\ H_B \\ H_N \\ U_N \end{bmatrix} = \begin{bmatrix} G \\ G_{LB} - G \\ G_{LN} \end{bmatrix} \qquad (2.1e)$$

$$U_B, H_B, H_N, U_N \geqslant 0 \text{ and integral.}$$

Subsequently it will be necessary to base definitions and procedures on elements of the system (2.1e) which are not explicitly present in (2.1a). This does not imply a computational requirement for continually keeping account of the full system (2.1e), since (2.1e) can always be constructed at any stage from (2.1a) and the (fixed) vectors G_{LB} and G_{LN}.

2. Rows of the system will usually be designated by the subscript i, and columns usually by the subscript j. Tableaus will be designed by k and t.

3. The incoming variable is identified by the index J; thus u_J signifies the incoming variable.

4. The cycle of the algorithm which transforms tableau k to tableau $k+1$ will be labeled cycle $k+1$. The incoming variable for cycle $k+1$ is $u_{J(k)}$ since u_J is determined from the data of tableau k.

5. The symbol v will be used to index source row for the Gomory cut. The source row for cycle $k+1$ will be designated $v(k)$.

6. The natural pivot row, determined after u_J is selected and before the cut is adjoined, is identified by the index I. The actual pivot or cut row is identified by the index p.

7. Data from, or derived from, the column $J(k)$ *in tableau* k are usually identified by the subscript J. Thus $a_{i,J}^{(k)}$ and $a_{i,J}^{(k')}$ are *not* in the same column while $a_{i,J}^{(k)}$ and $a_{i,J(k)}^{(k')}$ are in the same column of different tableaus.

8. If a vector A is lexicographically greater than a vector B we shall symbolize this relation by $A >_L B$.

9. The symbol $_I[y]$ means the integer part of y, i.e., the largest integer $\leqq y$. For example, $_I[11/2] = 5$; $_I[-5/2] = -3$; $_I[1/3] = 0$.

2.3. General and Preliminary Cycle Description

This algorithm, in common with other linear programming procedures—e.g., the simplex method—generates a sequence of tableaus. Each such tableau may be represented as a system of equations such as (2.1).[3] A cycle of the algorithm, or a complete iteration is defined here to include the decisions and algebraic manipulations required to accomplish the transition from any given tableau to the subsequent tableau.

A cycle of the primal algorithm includes the decisions and procedures that constitute a cycle of the simplex algorithm: a pivot row and a pivot column are selected and the pivot element thereby determined is used to execute the usual simplex change of basic procedure. Additionally the typical cycle of the primal algorithm includes adjoining a Gomory cut to the system of equations or tableau before the simplex change of basis procedure is executed.

The primal algorithm we shall describe here differs from the simplex algorithm in this fundamental respect: to execute a cycle, the simplex algorithm requires no more information than is contained in the current tableau, while, as will presently become apparent, the primal algorithm requires in addition some information from the history of the computations that led to the current tableau.

From the description already given we may conclude that a cycle of the algorithm must include the following decisions and procedures:

1. Selection of the income variable;
2. selection of the row (equation) in the tableau which will serve as the source, or source row, for the Gomory cut;
3. selection of the particular Gomory cut to be derived from that source row;
4. adjoining the cut to the tableau;
5. selection of the outgoing variable (or pivot row);
6. execution of a change of basis in accordance with the usual simplex procedure.

Several of the above steps are virtually automatic and require no special description here. If steps 1 through 4 have been completed, then steps 5 and 6 are completely specified by the usual simplex procedures applied to the tableau with the cut adjoined. If steps 1, 2, and 3 have been completed, then step 4 is also essentially mechanical.

All aspects of the algorithm will subsequently be elaborated. For the moment it will simplify matters if we confine our attention to some aspects of and constraints on the choices made in steps 1, 2, and 3, above.

Now we shall state two characteristics of the algorithm which contribute to the following effect: step 3 will be completely determined by the decisions made in steps 1 and 2; and the range of choice available in

[3] This is sometimes referred to as the Beale form of the tableau. See Charnes and Cooper [3, pp. 198ff] for a detailed discussion covering the interpretative significance of this formulation.

step 2 will be restricted. These two characteristics are

(i) the selection of the Gomory cut will be made in such a way that after the cut has been adjoined to the system, the cut will be selected as the natural pivot row in the new tableau, and

(ii) the pivot coefficient (i.e., the coefficient in the tableau which is common to the pivot row and the column of the incoming variable) will always have the value 1.

First we shall discuss the way these characteristics contribute to making step 3 automatic. We shall suppose that an incoming variable u_J and a source row v have been selected. Then the set of Gomory cuts which can be derived from row v is given in (2.3) below as a function of the positive parameter λ.[4]

$$s+\sum_j {}_I[a_{v,j}/\lambda]u_j+{}_I[1/\lambda]u_v={}_I[g_v/\lambda] \tag{2.3}$$

In (2.3) we have assumed for notational convenience that the basic variable u_v is associated with row v. Thus we know from (2.1) that the coefficient of u_v in row v is unity. In terms of (2.3), the choice required by step 3 is the assignment of a specific value for λ. If steps 1 and 2 have been made in such a way as to satisfy (i), we must have

$${}_I[a_{v,j}/\lambda]=1 \tag{2.4}$$

to satisfy (ii).

To satisfy (2.4) it is necessary[5] that

$$a_{v,J}\geqslant\lambda\geqslant a_{v,J}/2 \tag{2.5}$$

We shall resolve the problem of determining λ within the range (2.5) by stipulating

$$\lambda=a_{v,J}. \tag{2.6}$$

We shall find that the stipulation (2.6) has the useful effect of simplifying the selection of the source row so as to satisfy (*i*), given a prior selection of the incoming variable.

To see this, we may assume that the incoming variable u_J has been selected. Also we assume that from the data of (2.1)—specifically from the column G and the column of A corresponding to u_J (in which g_i and $a_{i,J}$ respectively represent the components of a typical row i)—we have calculated

$$\theta_J=\min_{i\epsilon I'}[g_i/a_{i,J}],$$

where $i\epsilon I'$ if and only if $a_{i,J}>0$. Note that the set of rows I' specifically cannot include the Gomory cut to be written as part of the cycle being discussed. In other words, we presume that $\theta_J\geqslant 0$, is determined *before* the cut is written. Then to satisfy (i), (presuming that (ii) can be satisfied as in (2.6)), it is necessary and sufficient that

$$0\leqslant {}_I[g_v/a_{v,J}]\leqslant\theta_J, \tag{2.7}$$

where v is the source row. We shall satisfy (i) if we select as source row any row v satisfying (2.7). It should be noted that at least one such row is always available: the natural pivot row—i.e., the row I for which $g_I/a_{I,J}=\theta_J$ which gives ${}_I[g_I/a_{I,J}]<\theta_J$ if θ_J is fractional and ${}_I[g_I/a_{I,J}]=\theta_J$ otherwise.

Thus if λ is determined by (2.6) and the source row is selected so as to satisfy (2.7), (i) and (ii) will always be satisfied.

For the sake of convenience we have foregone some of the freedom provided by (2.5) in the selection of λ. It may be noted parenthetically that there appears to be no straightforward line of reasoning—such as that applied in the Gomory all integer algorithm [6]—whereby selecting λ (either as large or) as small as possible is desirable.

It should be noted that (ii) guarantees the all integer character of the algorithm if the first tableau (the original statement of the problem) contains only integers.

Given a selection of the incoming variable u_J and of the source row v, we shall define

$$\theta_J'\equiv{}_I[g_v/a_{v,J}].$$

We may also note that

$$\theta_J\geqslant\theta_J'={}_I[\theta_J]$$

follows from (2.6), (2.7) and the definitions of ${}_I[y]$, θ_J', and θ_J.

For the sake of definiteness we have assumed in the preceding discussion that we are able to select an incoming variable and a source row. Rules covering these decisions—1 and 2 on our list on p. 226—remain to be determined, although we have by (2.7) narrowed the range within which the selection of the source row must be made. And we have made a commitment to a sequence of decision making in which the incoming variable is determined first and the subsequent selection of the source row is influenced—via (2.7)—by the prior selection of the incoming variable. The details of the selection of the incoming variable u_J and of the source row, v, are given in sections 2.6 and 2.8, respectively. In the next section, 2.4, we undertake a brief discussion of some aspects of the algorithm that depend on the characteristics we have already specified. In section 2.5 we discuss two problems that are implicit in the procedures we have specified in this section.

[4] This formulation is given by Gomory in [11].

[5] Since we require that $\lambda>0$, (2.5) implies $a_{v,J}>0$. Since by assumption we are dealing with a bounded problem, for any selection u_J, there must be at least one row v for which $a_{v,J}>0$.

[6] See Gomory, [11, pp. 197–198].

2.4. Cycle Classification: Transition Cycles and Stationary Cycles

In the proof of finiteness in part III and in the subsequent description of the primal algorithm in part II, much depends on an organization of the operation of the algorithm in terms of cycle categories. In this section we shall introduce the major distinctions to be used as a basis for classifying cycles of the algorithm.

We may begin by noting two possibilities: the value of

$$\theta_J' = {}_I[g_v/a_{v,J}] \tag{2.8}$$

where v is the index of the source row, may be zero or a positive integer. The expression (2.8) assigns the value of s, the slack variable in the Gomory cut (2.3). θ_J' also is the value assumed by u_J, the incoming variable, upon entering the basis.

Consider the case in which (2.8) is equal to zero. Adjoining the Gomory cut (2.3) to the tableau and installing s in the basis results in a degenerate solution. We permit this degeneracy. In case other rows are "tied" with the cut row as the natural pivot row, i.e., if $\theta_J' = \theta_J$, then the cut row is arbitrarily established as the pivot row. In part III we prove that this procedure does not lead to endless cycling. As we noted in part I, the subsequent pivot on the cut row results in a "new" basic solution in which all variables retain the values they had in the previous basic solution. Cycles in which the incoming variable enters the basis at a zero level—or equivalently in which $\theta_J' = 0$—will be called stationary cycles.

Cycles in which the incoming variable enters the basis at some positive integral level—or in which $\theta_J' \geqslant 1$—will be classified transition cycles. Transition cycles are so called because as the result of such cycles the solution actually moves to a new feasible lattice point. We shall further restrict the definition of transition cycles to cycles which yield an improvement in the criterion value of the solution.[7] Thus any variable u_j for which

$$c_j > 0 \text{ and } \theta_j = \min_{a_{i,j}>0} [g_i/a_{i,j}] \geqslant 1 \tag{2.9}$$

will yield a transition cycle if $u_J = u_j$. We shall use the symbol T to designate the set of all nonbasic variables which would, if designated the incoming variable, lead to a transition cycle. Thus T is set of all nonbasic u_j for which (2.9) is satisfied.

Corresponding to the interpretation of transition cycles as moves from one lattice point to another lattice point, a stationary cycle may be interpreted as moving the solution an infinitesimal distance along the edge that connects the tableau basic solution with the trial solution—i.e., the basic solution that would result from pivoting on row I and column J *without* adjoining a Gomory cut. The distinction between stationary and transition cycles will be significant in the remaining description of the mechanics of the algorithm and in the proof that the algorithm is finite.

Since each Gomory cut adjoined to the system brings an associated s variable into the basis, and since each such cut row immediately becomes the pivot row, the set of nonbasic variables will typically contain some of these s variables. We shall now define an s variable more precisely as *any slack variable from a Gomory cut which has been generated since the most recent transition cycle.* The other variables—which were part of the system that resulted from the most recent transition cycle—are *all* defined as x variables.

We shall classify every stationary cycle either as an x cycle or as an s cycle. If the incoming variable is an s variable, the cycle is defined as an s cycle. If the incoming variable is an x variable, the cycle is defined as an x cycle. Now we have three fundamental types of cycles: transition cycles, x cycles, and s cycles.

As implied by the definition of transition cycles and the discussion in part I of the primal algorithm and the direct algorithm of Ben-Israel and Charnes, the essential difficulty of constructing a finite primal algorithm resides in showing, if one starts with the initial tableau (or a tableau that results from a transition cycle), that a finite member of stationary cycles will be sufficient to achieve either (a) another transition cycle or (b) a tableau in which primal optimality conditions are satisfied. Thus, our description of the primal algorithm will concentrate on the details of stationary cycles, and will be limited to the following brief discussion of the details of executing a transition cycle.

The pivot operation in a transition cycle may be accomplished by selecting, as u_J, an arbitrary element of T. Any row, v, for which $1 \leqslant {}_I[g_v/a_{v,J}] \leqslant \theta_J$ can be used as the source row for a Gomory cut and the cut can then be employed as the pivot row. If $a_{v,J} = 1$, then it is permissible to pivot on the row v without adjoining a cut. When $a_{v,J} = 1$ it is also possible to adopt the special procedure described below in section 2.5.

Following a transition cycle a number of housekeeping details need specification—e.g., what is to be done with nonbasic and basic s variables? etc. Many procedures are possible. We shall be content here to outline a simple procedure taken from Ben-Israel and Charnes [1], as an example which will prevent troublesome growth in the number of equations and variables.

Let (2.1) represent the original tableau. We note that each transition cycle specifies the coordinates of a new and better feasible lattice point than was previously available; and the solution that results from a transition cycle can be (uniquely) expressed as a nonnegative integer combination, $A^0 = \sum_l \phi_l A_l$, of the columns of A in (2.1). Following the procedure of Ben-Israel and Charnes, the column A^0 and an associated variable x^0 may be adjoined to A and X_N in (2.1) and a pivot operation may then be executed

[7] The rules for selection of the incoming variable given in section 2.6 below preclude a selection u_J such that $c_J = 0$ and $\theta_J \geqslant 1$.

(according to the rules of the rudimentary primal algorithm) to bring x^0 into the basis at a positive level.

This procedure will (i) generate a feasible, all integer tableau in which the new and improved solution is basic, (ii) add one new row and one new column to the system, and (iii) require that a record be kept of the vector ϕ which relates the new variable x^0 to the variables in X_N. The tableau that results from this procedure is either optimal or serves as the starting point for a new sequence of stationary cycles. We emphasize that *each* new solution resulting from a transition cycle is expressed by the procedure we have described in terms of the *original* A matrix. This permits discarding all information and variables related to previous transition cycles.

2.5. Special Procedures

In this section we describe two supplementary procedures designed to overcome difficulties which would result from unmodified application of the procedures given in section 2.3.

We begin with a description of a special procedure to be used when

$$a_{v,J}=1. \qquad (2.7)$$

Then (2.6) and (2.3) imply that the basic variable u_v has a unit coefficient in the cut equation. Adjoining the cut to the tableau would create *a second* nonzero element in the (otherwise "basic") column corresponding to u_v. To avoid this difficulty, we introduce (and use) the following "weakened" cut:

$$s+\sum_j {}_I[a_{v,j}/\lambda]\cdot u_j={}_I[g_v/\lambda]. \qquad (2.3\text{w})$$

When $\lambda>1$, (2.3w) is identical to (2.3). When $\lambda=1$, (2.3w) is weaker than (2.3) in this sense: every solution which is feasible with respect to (2.3) is also feasible with respect to (2.3w). To demonstrate this it is sufficient to rewrite (2.3) as

$$s={}_I[g_v/\lambda]-\sum_j {}_I[a_{v,j}/\lambda]\cdot u_j-{}_I[1/\lambda]\cdot u_v,$$

and note that since $u_v\geqslant 0$ in any feasible solution, if the coefficient of u_v is arbitrarily set at zero (as in the cut (2.3w)), then the value of s cannot decrease. Thus any set of specific values for the u variables which determine a nonnegative s in (2.3) will also determine a nonnegative s in (2.3w). We may conclude, then, that (2.3w) carries the same guarantee against interdiction of an integer solution as does the Gomory cut (2.3).

When the cut (2.3w) is adjoined with $\lambda=a_{v,J}=1$, the source row and the cut row are identical except for the basis entries: there is a 1 in the cut row and the s column and a 1 in the source row and u_v column. Then, as the result of pivoting with the cut serving as the pivot row, the source row, v, is changed to the following form

$$u_v-s=0$$

where u_v remains a basic variable and s is a (newly) nonbasic variable. This permits us to interpret s as a nonbasic proxy for the zero-level basic variable u_v. The row v will not be changed by subsequent pivots until and unless s becomes the incoming variable at some later stage.

One of the implications of this procedure that will be of interest later is this: since the cut row *always* serves as the pivot row, no x variable will ever be removed from the basis by a stationary cycle. We may note that this procedure is a departure from the rudimentary primal algorithm described in part I: in the rudimentary algorithm if the source row has a 1 in the pivot column, then the source row is used as the pivot row and no cut is adjoined.

The second supplementary procedure is applied whenever an s cycle occurs. An s cycle has the following effects: the s variable created during the cycle is driven out of the basis; the incoming s variable reenters the basis. If there were no means of eliminating s variables from the system a large number of s cycles would lead to a large number of new rows and variables, since each cycle creates a new row and variable.

This difficulty is avoided by the following procedure: after each s cycle the incoming variable (which has become basic with a 1 in the pivot row) is dropped from the system, along with the rest of the pivot (cut) row. The justification for this procedure is given in appendix A.

As a result of the procedures described in this section we may conclude that the algorithm will exhibit the following properties: (i) each x cycle will increase both the number of rows and the number of columns (or variables) by one, and (ii) each s cycle will leave the number of rows and the number of columns unchanged. Thus the number of nonbasic s variables will always equal the number of x variables which have entered the basis at a zero level as the result of x cycles.

2.6. Selection of the Incoming Variable

A formal statement of the rules for selection of the incoming variable is given at the end of this section. As will be observed from those rules, a transition cycle is executed whenever possible. For transition cycles the incoming variable is selected from the set T. If the set T is empty, a stationary cycle must occur and the selection of the incoming variable must lead to either an x cycle or an s cycle. For stationary cycles, the incoming variable is selected from a set $E(+)_0$ which we shall now proceed to define. It will be convenient to use the following definitions as a basis for discussion:

$$C+=\{u_j|c_j>0\} \qquad (2.10)$$

$$X+=\{u_j | u_j \epsilon C+ \text{ and } u_j \text{ is an } x \text{ variable}\} \quad (2.11)$$

$$x_0 = \text{A single (nonbasic) variable, arbitrarily selected from } X+. \quad (2.12)$$

Every sequence of stationary cycles follows a transition cycle, or follows the initial tableau of the problem. We shall establish the tableau that has resulted from the most recent transition cycle (or the initial tableau of the problem) as the natural starting point, or set of initial conditions, for any sequence of stationary cycles. Accordingly, we let t signify the tableau index of (the initial tableau or) the tableau that has resulted from the most recent transition cycle. Then the index of a typical succeeding stationary cycle is given by $t+k$, with $k \geqslant 1$. The s variable created during cycle $t+k$ (which generates the tableau indexed by $t+k$) is designated by s_{t+k}.

The set $E[+)_0$ will be defined as a subset of a set E_0. The composition of the set E_0 will vary from tableau to tableau: thus the symbol $E_0^{(t+k)}$ is used to denote a set, associated with tableau $t+k$, from which the incoming variable for cycle $t+k+1$ is selected. We shall first develop a formal definition for E_0. The definition of E_0 is recursive: i.e., $E_0^{(t+k)}$ is defined in terms of $E_0^{(t+k-1)}$. The initial set is defined by:

$$E_0^{(t)} = \{x_0\} \quad (2.13)$$

Although, as indicated, $E_0^{(t+k)}$ is essentially a function of $E_0^{(t+k-1)}$, we shall write the recursion formula in terms of another set, $S^{(t+k)}$, which is the set of all nonbasic s variables in tableau $t+k$.

$$S^{(t+k)} = (E_0^{(t+k-1)} \cup \{s_{t+k}\}) - \{u_{J(t+k-1)}\} \quad (2.14)$$

$$E_0^{(t+k)} = S^{(t+k)}, \text{ if } S^{(t+k)} \cap C+ \neq \phi, \text{ and} \quad (2.15)$$

$$= S^{(t+k)} \cup \{x_0\}, \text{ otherwise.} \quad (2.16)$$

From (2.13) and (2.14) we can generate

$$\begin{aligned} S^{(t+1)} &= (E_0^{(t)} \cup \{s_{t+1}\}) - \{u_{J(t)}\} \\ &= (\{x_0\} \cup \{s_{t+1}\}) - \{x_0\} \\ &= \{s_{t+1}\} \end{aligned}$$

In the above we have assumed

$$u_{J(t)} = \{x_0\},$$

which is based on the hypothesis that cycle $t+1$ is a stationary cycle and therefore the incoming variable, $u_{J(t)}$, for cycle $t+1$ must be in $E_0^{(t)}$. Generally after tableau $t+k$ is generated by cycle $t+k$, we first revise $S^{(t+k)}$ by (2.14). This keeps $S^{(t+k)}$ coincident with the set of nonbasic s variables by (i) adding to $S^{(t+k)}$ the s variable newly created by the preceding cycle and (ii) deleting the variable which entered the basis during the previous cycle. Then, by (2.15), $E_0^{(t+k)}$ is made equivalent to $S^{(t+k)}$ unless $S^{(t+k)}$ fails to contain a variable in $C+$. In the latter case x_0—selected by (2.12)—is included in $E_0^{(t+k)}$. In the normal[8] operation of the algorithm, if x_0 is added to $S^{(t+k)}$ to form $E_0^{(t+k)}$, then x_0 will be chosen as the incoming variable for cycle $t+k+1$ and will therefore not be included in $S^{(t+k+1)}$ as defined by (2.14). Thus normally if $E_0^{(k+t)} = S^{(k+t)}$ the next cycle, $t+k+1$, is an s cycle, and if $E_0^{(t+k)} \neq S^{(k+t)}$ the next cycle is on x cycle with $u_{J(k+t)} = x_0$.

Before defining $E[+)_0$ as a subset of E_0 we must first define a special row of the tableau. Let the tableau be given by

$$IU_B + A^{(k)}U_N = G^{(k)},$$

or in expanded form

$$\begin{bmatrix} I & & & A^{(k)} \\ & I & & -A^{(k)} \\ & & I & I \end{bmatrix} \begin{bmatrix} U_B \\ H_B \\ H_N \\ U_N \end{bmatrix} = \begin{bmatrix} G^{(k)} \\ \\ G_{LB} - G^{(k)} \\ G_{LN} \end{bmatrix} \quad (2.17)$$

Let x_0 be the x variable that has most recently been selected according to (2.12) and introducing into E_0 according to (2.16). The special row referred to above is the limit row associated with x_0. If x_0 has not entered the basis then this limit row has the form

$$h_0 + 1 \cdot x_0 = g_{Lo}$$

where h_0 is basic and x_0 is nonbasic. If x_0 has entered the basis during some preceeding cycle, then this limit row is

$$h_0 + \sum_j (-a_{0,j}) = g_{Lo} - g_0 = g_{Lo}$$

where h_0 is basic and the summation is over the index set of nonbasic variables.

If we define $\mathbf{A}^{(k)} = \begin{bmatrix} -A^{(k)} \\ I \end{bmatrix}$

$$G_L^{(k)} = \begin{bmatrix} G_{LB} - G^{(k)} \\ G_{LN} \end{bmatrix}$$

$$H = (H_B, H_N)$$

[8] Variations of the algorithm are evidently possible in which x_0 might not immediately enter the basis after being included in E_0. This raises no essential difficulty with the procedure or set definitions given here and only contradicts—for a limited sequence of tableaus—our *interpretation* of $S^{(t+k)}$ as the set of all nonbasic s variables. By placing restrictions on the assignment of rows to the sequence of indices in (2.22) below we can insure that x_0 will always enter the basis immediately. However, we need not and do not make this assumption; see also appendix B.

we may rewrite (2.17) as

$$\begin{bmatrix} I & & A^{(k)} \\ & I & \mathbf{A}^{(k)} \\ & & \end{bmatrix} \begin{bmatrix} U_B \\ H \\ U_N \end{bmatrix} = \begin{bmatrix} G^{(k)} \\ G_L^{(k)} \\ \end{bmatrix} \tag{2.18}$$

where U_B, H, U_N are nonnegative and integral. Now we may give as a general expression for the limit row 0,

$$h_0 + \sum_j \mathbf{a}_{0,j} = g_{L0}. \tag{2.19}$$

The set $E[+)_0$ may be defined in terms of this row, as follows:

$$E[+)_0 = \{u_j | u_j \epsilon E_0 \text{ and } \mathbf{a}_{0,j} \geqslant 0\}. \tag{2.20}$$

This set contains u_J for any stationary cycle.
We shall also define, for the later reference,

$$E(+)_0 = \{u_j | u_j \epsilon E_0 \text{ and } \mathbf{a}_{0.j} > 0\}. \tag{2.21}$$

Selection of the incoming variable within $E[+)_0$ is accomplished by reference to a collection of vectors $\mathbf{R}_j$. One such vector is associated with each $u_j \epsilon E[+)_0$. To define $\mathbf{R}_j$ we require notational conventions to distinguish those rows of the tableau associated with x variables that have entered the basis *since* the most recent transition cycle and *before* the entry of the variable x_0 discussed in the previous paragraph. We shall let the sequence of indices

$$\mathbf{1, 2, \ldots, i, \ldots, r} \tag{2.22}$$

symbolize these rows. The assignment of particular rows to particular indices is arbitrary. The assignment of rows to indices may be revised after each x cycle and after each s cycle which decreases[9] $c_J/\mathbf{a}_{0,J}$. The vector $\mathbf{R}_j$ associated with the variable u_j is defined by

$$\mathbf{R}_j \equiv \left(\frac{c_j}{\mathbf{a}_{0,j}}, \frac{a_{\mathbf{1},j}}{\mathbf{a}_{0,j}}, \frac{a_{\mathbf{2},j}}{\mathbf{a}_{0,j}}, \ldots, \frac{a_{\mathbf{r},j}}{\mathbf{a}_{o,j}} \right). \tag{2.23}$$

The incoming variable is selected by choosing the $u_j \epsilon E[+)_0$ associated with the lexicographically largest $\mathbf{R}_j$.

We may now summarize the procedure for selection of the incoming variable with the following collection of rules.

Rule 1(J) The incoming variable u_J will always have $c_J \geqslant 0$.

Rule 2(J) If possible, the incoming variable should lead to a transition cycle (which will improve the solution): i.e., whenever possible, we select u_J such that $\theta'_J \geqslant 1$ and $c_J > 0$. An arbitrary choice is permissible among several variables, in the set T, which satisfy these criteria.

Rule 3(J) If no variable satisfies the criteria in Rule 2(*J*) and if $C+$ is not empty, then u_J is selected from $E[+)_0$. The particular variable selected from E_0 as u_J is associated with the lexicographically largest vector $\mathbf{R}_j$ over $E[+)_0$.

Rule 3(J) and the definition of $\mathbf{R}_j$ generate a problem of interpretation, namely: how is the lexicographic priority of a vector $\mathbf{R}_J$ determined when $\mathbf{a}_{0,j}=0$? We require the following rule to resolve this difficulty:

Rule 3a(J) To determine the lexicographic order of two vectors $\mathbf{R}_j$ and $\mathbf{R}_{j'}$ when (all) the components of one or both of the vectors have zero denominators, we employ the following conventions:

Case I: $\mathbf{a}_{0,j'} > 0$ and $\mathbf{a}_{0,j} = 0$.

Ia. If the first component of $\mathbf{R}_j$ with a nonzero numerator has a positive numerator, then:

$$\mathbf{R}_{j'} <_L \mathbf{R}_j.$$

Ib. If the first component of $\mathbf{R}_j$ with the nonzero numerator has a negative numerator, then:

$$\mathbf{R}_j <_L \mathbf{R}_{j'}.$$

Case II: $\mathbf{a}_{0,j'}=0$, $\mathbf{a}_{0,j}=0$. Compare $\mathbf{R}_j$ to $\mathbf{R}_{j'}$ to find the first component in which the numerators are unequal. Let $a_{i,j'}$ and $a_{i,j}$ symbolize this first unequal pair of numerators. Then, $a_{i,j} > a_{i,j'}$ implies: $\mathbf{R}_j >_L \mathbf{R}_{j'}$.

The rules for the selection of the incoming variable raise two natural questions:

1. What happens if $C+ \neq \emptyset$, so that the current basis is not optimal, and $T=\emptyset$, so that a transition cycle cannot occur, and $C+ \cap E[+)_0 = \emptyset$?

2. Are we assured that rule 3(J) as supplemented by Rule 3a(J) will always yield a unique selection of the incoming variable? The answer to (1) is that the indicated situation cannot occur. This is demonstrated[10] in part III, after the necessary foundation (theorem I) has been established. The answer to (2) is that these rules do invariably yield a unique choice. This is proved in appendix B.

2.7. Some Implications and Explanation of the Rules for Selecting the Incoming Variable

This section covers two topics. First there is a listing of some implications of the rules for selection at the incoming variable. Then there follows a short discussion of some of the rationale for these rules.

The following propositions can be proved on the basis of the rules for the selection of the incoming

[9] i.e., for each cycle k for which

$$\frac{c_{J(k)}^{(k)}}{\mathbf{a}_{0,J(k)}^{(k)}} < \frac{c_{J(k)}^{(k-1)}}{\mathbf{a}_{0,J(k-1)}^{(k-1)}}$$

[10] See below, p. 241 proposition (3.18) and the proof of corollary IIA.

variable and appropriate theorems from part III. They are stated here in the hope that they may expedite insight into some characteristics of the algorithm.

1. For all $u_j \epsilon E_0$, $c_j > 0 \Longrightarrow \mathbf{a}_{0,j} > 0$.
2. $c_J > 0 \Longleftrightarrow \mathbf{a}_{0,J} > 0$.
3. $c_J = 0 \Longleftrightarrow \mathbf{a}_{0,J} = 0$.
4. $\mathbf{a}_{0,J} = 0 \Longrightarrow$ the first component in $\mathbf{R}_J$ with a non-zero numerator has a positive numerator.
5. Only a finite number of successive cycles can occur with $\mathbf{a}_{0,J} = 0$; therefore, only a finite number of cycles can elapse while $x_0 \epsilon E_0$.
6. The vector $\mathbf{R}_J$ is lexicographically positive.
7. $\mathbf{R}_{J(t+k)}^{(t+k)}$ undergoes a monotonic lexicographic decrease as k increases.

While the source of much of the rationale for the incoming variable selection in stationary cycles is theorem I of part III, some aspects of the motivation for this procedure may be usefully discussed here. Our main goal in this duscussion is to develop the essentials of an interpretive connection between the "original" tableau (by which term we include any tableau that results from a transition cycle) and the data of the tableaus that are generated in the course of a succeeding sequence of stationary cycles. To support this goal we require the following notational development.

We shall suppose that a transition cycle has occurred and that the constraints of the resulting tableau may be represented by

$$I \cdot X_B + A \cdot X_N = G, \tag{2.1}$$

in which I is an m by m identity matrix, A is an m by $n-m$ matrix and G is an m by 1 vector. All the components of A and G are assumed to be integers and the components of G are nonnegative. The variable column vectors X_B and X_N have appropriate dimensions; the basic solution is $X_B = G$, $X_N = 0$. By definition (2.1) contains no s variables. Let the criterion function (to be maximized) which is associated with the above constraints be

$$z = C \cdot X_N$$

where a typical (integer) component of C is defined by

$$c_j = \gamma_j - \sum_i a_{i,j}\gamma_i, \tag{2.2}$$

and where γ_j and the γ_i are criterion coefficients given in the original statement of the problem. The summation is over all rows i. For convenience we have assumed that row i of the tableau is associated with the basic variable x_i. We assume that at least one c_j is positive.

Now we assume that a sequence of k successive stationary cycles has occurred. As a result the system (2.1) will have evolved to this form:

maximize

$$\begin{bmatrix} C_X^{(k)}, & C_S^{(k)} \end{bmatrix} \cdot \begin{bmatrix} X_{N''} \\ S \end{bmatrix} \tag{2.24}$$

subject to

$$I \cdot \begin{bmatrix} X_B \\ X_{N'} \end{bmatrix} + \begin{bmatrix} {}_xA^{(k)}, & {}_sA^{(k)} \end{bmatrix} \cdot \begin{bmatrix} X_{N''} \\ S \end{bmatrix} = \begin{bmatrix} G_B^{(k)} \\ G_{N'}^{(k)} \end{bmatrix} = G^{(k)}. \tag{2.24}$$

The terms in (2.24) have this interpretation:

X_B:	a subvector that contains the variables that were basic in (2.1) after the latest transition cycle and remain basic after cycle k,
$X_{N'}$:	a subvector that contains those variables of X_N in (2.1) that were nonbasic after the latest transition cycle and are basic after cycle k,
S:	a subvector that contains the s variables that have been created by the cycles after the latest transition cycle and are basic after cycle k,
$X_{N''}$:	a subvector that contains the variables of X_N in (2.1) that remain nonbasic after cycle k,
${}_xA^{(k)}$:	a submatrix that consists of the columns of $A^{(k)}$ that correspond to components of $X_{N''}$
${}_sA^{(k)}$:	a submatrix that consists of columns $A^{(k)}$ that correspond to components of S,
$C_X^{(k)}$:	a subvector that consists of the components of $C^{(k)}$ that correspond to components of $X_{N''}$,
$C_S^{(k)}$:	a subvector that consists of the components of $C^{(k)}$ that correspond to components of S,
$G_B^{(k)}$:	a subvector of $G^{(k)}$ which corresponds to X_B,
$G_{N'}^{(k)}$:	a subvector of $G^{(k)}$ which corresponds to $X_{N'}$,

The dimensions of terms in (2.23) are as follows: X_B and $G_B^{(k)}$ are m by 1; $G_{N'}^{(k)}$, $X_{N'}$ and S are k' by 1;

$X_{N''}$ is	$(n-m-k')$ by 1;
${}_xA^{(k)}$ is	$(m+k')$ by $(n-m-k')$;
${}_sA^{(k)}$ is	$(m+k')$ by (k'); and

$k' \leqslant k$ is the number of x cycles in the sequence of k stationary cycles.

All the constants in (2.24) are integers and $G^k \geqslant 0$. The basic solution is $X_B = G_B^{(k)}$, $X_{N'} = G_{N'}^{(k)} X_{N''} = 0$, $S = 0$.

Since the cycles which have converted the system (2.1), (2.2) to the system (2.24) are stationary cycles, G in (2.1) is equal to $G_B^{(k)}$ in (2.24), and all the components of $G_{N'}^{(k)}$ are zeros. These conclusions follow from the fact that a stationary cycle always has a zero in the constant column and pivot row, and only adds zeros to each component of the constant column.

In summary form our rationale for certain aspects of the incoming variable selection procedure is based on these considerations:

1. Each column of ${}_sA^{(k)}$ in (2.24) is equivalent (in a sense to be discussed presently) to an integer combination, $\sum_l \phi_l A_l$, of the columns A in (2.1). Thus each component s_j of S represents a potential solution to the auxiliary problem[11] of Ben-Israel and Charnes.

This establishes a linkage between the problem in "original" form and any tableau generated in the course of an immediately following sequence of uninterrupted stationary cycles. It will be recalled that a solution to the auxiliary problem must have (i) all ϕ_l nonnegative and integer, (ii) $\sum_l \phi_l A_l \leqslant G$ and (iii) $\sum_l \phi_l c_l \geqslant 1$. A "potential solution", $\sum_l \phi_l A_l$, which is related to a s variable s_j of (2.24) need not satisfy all three of these conditions. If all three conditions are satisfied then $s_j \epsilon T$. Typically the potential solution to the auxiliary problem associated with the incoming variable s_J of an s cycle will satisfy (iii), will not satisfy (ii), and may or may not satisfy (i).

2. Each stationary cycle, k, generates a new submatrix ${}_sA^{(k)}$ and hence a new collection of potential solutions to the auxiliary problem.

3. The vectors $X_{N'}$ and $X_{N''}$ in (2.24) constitute a partition of the vector X_N in (2.1). There exists, then, a corresponding partition of the columns of A in (2.1). The integer combination of columns of A in (2.1) that corresponds to a particular s_j in (2.24) only has nonzero weights for the columns $A_{j'}$ of A that correspond to variables $x_{j'}$ of X_N which are also in $X_{N'}$. During a sequence of s cycles the composition of the vector $X_{N'}$ in (2.24) does not change. Thus a sequence of s cycles (which includes cycle k) generates potential solutions to the auxiliary problem in which nonzero weights are only assigned to columns $A_{j'}$ in (2.1) corresponding to variables $x_{j'}$ of $X_{N'}$.

4. When, in some tableau k, the condition $C+\cap S^{(k)} = \emptyset$ occurs, the following conclusions are implied: (i) an x cycle is a necessary condition for obtaining a better solution, (ii) since an x cycle would expand and redefine $X_{N'}$, no solutions to the auxiliary problem exist which assign nonzero weights only to those columns of A that correspond to x variables in the vector $X_{N'}$ as currently constituted.

5. Thus we generally interpret s cycles as generating new integer combinations in which nonzero weights are associated only with the variables in $X_{N'}$. By contrast an x cycle expands and redefines $X_{N'}$ and thereby expands the scope of the solutions to the auxiliary problem associated with the s cycles that follow. Our definitions (2.16) and (2.17) which determine the constitution of E_0 insure that an x cycle will only occur if no possiblity remains for a solution to the auxiliary problem which has zero weights for all $A_{j''}$ associated a variable $x_{j''}$ in $X_{N''}$.

The relation between a column ${}_sA^{(k)}$ in (2.24) and an integer combination $\sum_l \phi_l A_l$ over the columns of A in (2.1) will now be developed.

Since the process that leads from (2.1) to (2.24) is one of pivoting and adjoining Gomory cuts, every solution to (2.24) must also be a solution to (2.1). Moreover all solutions to (2.1) can be expressed in terms of values for the "independent" variables X_N, and the corresponding values for components x_i of X_B are uniquely determined by (2.1). Thus if the values of the variables in X_N are known for some solution to (2.24) this is sufficient to determine a corresponding solution to (2.1).

Now let ${}_jA^{(k)}_{N'}$ represent the part of a column of ${}_sA^{(k)}$ which relates a typical s variable, s_j, to the basic variables of $X_{N'}$. If a solution (not necessarily feasible) to (2.24) has $s_j=1$ and all other nonbasic variables in (2.24) equal to zero, the value of $X_{N'}$ in this solution is

$$X_{N'} = 0 - {}_jA^{(k)}_{N'} \cdot 1. \tag{2.25}$$

This determines a corresponding solution to (2.1), which is not necessarily feasible. We may also express (2.25) as a solution, $\sum_l \phi_l A_l$, to the auxiliary problem by defining

$$\phi_l \equiv -a^{(k)}_{i,j}$$

in which the index i corresponds the same component of $X_{N'}$ that corresponds to the column A_l of A.

The preceding discussion has been aimed at showing some of the rationale for the definitions (2.14), (2.15) and (2.16). We shall make no such protracted attempt to develop a rationale for selecting u_J from $E[+)_0$ or for the use of the lexicographic domination test with the vector $\mathbf{R}$. We shall be content to note that because of the restriction (2.15) on the composition of E_0, it turns out that restricting the selection of u_J to $E[+)_0$ is equivalent to requiring that $c_J \geqslant 0$.

The role of the lexicographic test in terms of the vector $\mathbf{R}$ is closely related to the development in part III. We can, however, at least offer the following remark here.

The proofs of part III establish that

$$\mathbf{R}_{J(k)} >_L \mathbf{R}_{J(k+1)}$$

where k and $k+1$ are s cycles. This can be shown to imply that the potential solution to the auxiliary problem, $\sum_l \phi_l A_l$ related to incoming variable, s_J, for cycle k, cannot be identical to the solution $\sum_l \phi_l' A_l$ related to the incoming variable for cycle $k+1$, or the solution related to any succeeding s cycle.

In the next section we give the procedure for selecting the source row.

[11] See above, section 1.9.

2.8. Rules for the Selection of the Source Row

The rules for selection of the source row fall into two categories; special and normal. The normal rules are in essence the source row selection rules given in the rudimentary primal algorithm in part I. In the formal statement of the rules given below, rule 1(v) is the normal rule and rules 2(v), 3(v), and 4(v) are special.

Particular circumstances are required to evoke the special rules. The rules are designed to insure that these special circumstances persist for at most a finite subsequence of cycles. Thus the "normal" (i.e., nonspecial) circumstances must reoccur at finite intervals. This fact is used as a basis for proving that the algorithm is finite. It will probably not be transparent to many readers how the "normal" circumstances contribute to a proof of finiteness or how the special rules work to eradicate the circumstances that bring these rules into operation. Accordingly it may be most efficient to postpone a serious attempt to appreciate the rationale for the special rules until these rules are cited in the proofs given in part III. It should also be recalled that alternative special source row selection rules may be employed.

The rules for source row selection are founded on two definitions. The first of these is comparatively simple: the definition of the set $V(J)$, the set of row indices from which the source row, v, may be selected when u_J is the incoming variable and the normal source row solution rules are operative: Recalling (2.7) we define[12]

$$V(J) \equiv \{i \mid {}_I[g_i/a_{i,J}] \leq \theta_J\}\cdot \tag{2.26}$$

The second concept to be defined is both more complex and less obviously germane to the source row selection decision. As we have indicated, and will eventually prove, the vector $\mathbf{R}_J$ undergoes a strict lexicographic decrease from cycle to cycle during a sequence of s cycles. Let the first component of $\mathbf{R}_J$ which decreases (as the result of a given cycle) be called the *change component*. Let k be a tableau generated in a sequence of s cycles and let $\mathbf{i}$ be any index in the sequence (2.22)—i.e., $\mathbf{i}$ is the index of a component[13] of $\mathbf{R}_{J(k)}$. Let $k_{\Delta\mathbf{i}} \leq k$ be the index of the most recent cycle in which $\mathbf{i}$ was the change component. We may now define the component $\mathbf{i}$ of $\mathbf{R}_{J(k)}$ to be *out of bounds* if the following are true:

(i) $\mathbf{a}_{0,J} > 0$,
(ii)[14] $\mathbf{a}_{0,J} > g_{L0}$ or $\mathbf{a}_{\mathbf{i},J} > g_{L\mathbf{i}}$,
(iii) for each cycle k', where $k_{\Delta\mathbf{i}} \leq k' \leq k$, the change component has been a component identical to or after $\mathbf{i}$ in $\mathbf{R}_{J(k')}$,
(iv) for every tableau k', where $k_{\Delta\mathbf{i}} \leq k' \leq k$, (i) and (ii) have been true or $\mathbf{a}_{0,J}^{(k')} = \mathbf{a}_{\mathbf{i},J}^{(k')} = 0$.

The rules for selection of the source row, v, follow:
Normal Source Row Selection:
Rule 1(v) Any row $i \epsilon V(J)$ may be selected as the source row if:
(i) a transition cycle is being executed, or
(ii) an x cycle is being executed, or
(iii) an s cycle is being executed, and no component of $\mathbf{R}_J$ is out of bounds, and $\mathbf{a}_{0,J} > 0$.
Special Source Row Selection:
Rule 2 (v) The limit row $\mathbf{i}$, i.e., the row with the typical coefficient $\mathbf{a}_{\mathbf{i}j}$, is the source row if:
(i) $\mathbf{a}_{0,J} > 0$, and
(ii) $\mathbf{i}$ is the smallest[15] index $(\mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{r})$ of $\mathbf{R}_J$ which is out of bounds, and
(iii) $\mathbf{a}_{\mathbf{i},J}/\mathbf{a}_{0,J} \geq g_{L\mathbf{i}}/g_{L0}$
Rule 3 (v) The limit row 0, i.e., the row with the typical coefficient $\mathbf{a}_{0j}$, is the source row if:
(i) $\mathbf{a}_{0,J} > 0$, and
(ii) row $\mathbf{i}$ is the first component of $\mathbf{R}_J$ which is out of bounds, and
(iii) If $\mathbf{a}_{\mathbf{i},J}/\mathbf{a}_{0,J} < g_{L\mathbf{i}}/g_{L0}$
Rule 4(v) The row $\mathbf{i}$ i.e., the row with the typical coefficient $a_{\mathbf{i},j}$, is the source row if:
(i) $\mathbf{a}_{0,J} = 0$, and
(ii) $\mathbf{i}$ is the smallest index of a component of $\mathbf{R}_J$ which has the form:

$$\frac{\text{positive integer.}}{\text{zero}}$$

It is easily verified that rules 2(v), 3(v), and 4(v) always select a source row which satisfies (2.7), since in all cases ${}_I[g_v/a_{v,J}] = 0$.

As we have indicated the "special" rules are designed to insure convergence of the algorithm. Theorems VI and VII of part III rely specifically on properties of these rules. Rule 4(v) is designed to guarantee that only a finite number of successive cycles can have $\mathbf{a}_{0,J} = 0$. This is proved in corollary IIA of part III. In all cases the desired result is established on the basis of the properties of the source row selection rules and theorem II of Part III. This collection of rules is sufficient to guarantee finiteness, but it is not unique[16] in this respect. An analysis of theorems VI and VII of part III will suggest alternative possibilities to the rules given here.

2.9. Summary Flow Chart

The primal algorithm is presented in summary flow chart form in figure 2/1. While most of the terminology of the flow chart is defined there, some terms such as s_{t+k}, g_{Lj}, and g_{L0} must be located in the appropriate section of part II for a definition.

[12] In the interest of simplicity we may interpret $V(J)$ as a subset of the rows of the tableau. However, there is no logical barrier to letting i in (2.26) range over a set which includes indices that correspond to every positive linear combination of the rows of the tableau.

[13] Strictly speaking, $\mathbf{i}$ indexes the numerator of a component of $\mathbf{R}_J$. Since the denominator is identical for all components, we may let the index of the numerator serve to indentify the entire component.

[14] When the index $\mathbf{i}$ is the first index in $\mathbf{R}_J$ (i.e., the index of the component $c_J/\mathbf{a}_{0,J}$ we *need* not define a limit on the row of the numerator; if no limit has been defined on the c row then the first component can be out of bounds only if $\mathbf{a}_{0,J} > g_{L0}$.

[15] We have assumed here that no limit is placed on the c row. If such a limit is available then the formulae of rule 2(v) and rule 3(v) may be extended to cover the first numerator in $\mathbf{R}_J$.

[16] See [21].

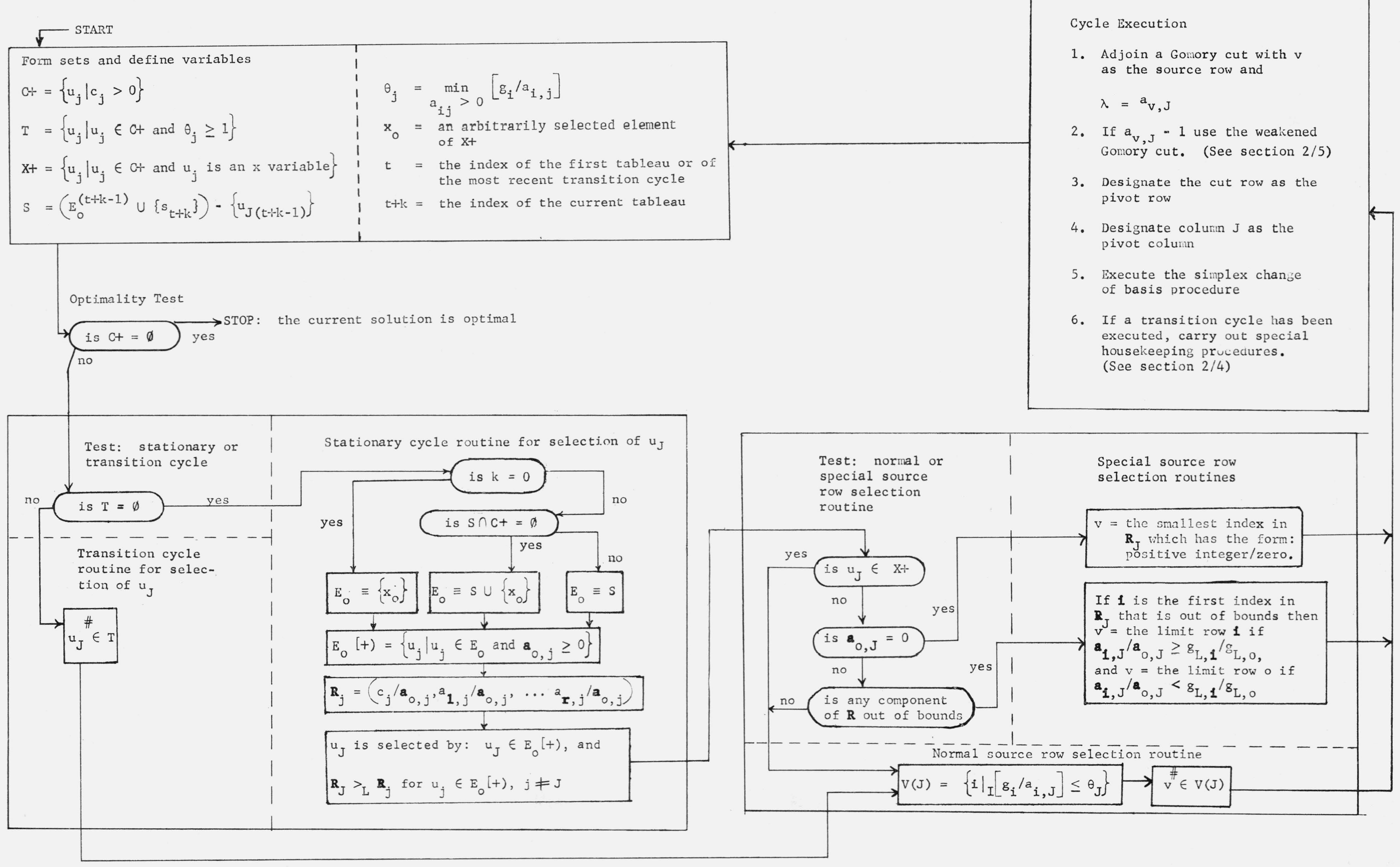

START
Form sets and define variables
$C+ = \{u_j | c_j > 0\}$
$T = \{u_j | u_j \in C+ \text{ and } \theta_j \geq 1\}$
$X+ = \{u_j | u_j \in C+ \text{ and } u_j \text{ is an x variable}\}$
$S = (E_o^{(t+k-1)} \cup \{s_{t+k}\}) - \{u_{J(t+k-1)}\}$
$\theta_j = \min_{a_{ij} > 0} [g_i/a_{i,j}]$
x_o = an arbitrarily selected element of X+
t = the index of the first tableau or of the most recent transition cycle
t+k = the index of the current tableau
Optimality Test
is C+ = ∅
yes
STOP: the current solution is optimal
no
Test: stationary or transition cycle
is T = ∅
no
yes
Transition cycle routine for selection of u_J
$\overset{\#}{u_J} \in T$
Stationary cycle routine for selection of u_J
is k = 0
yes
no
is S ∩ C+ = ∅
yes
no
$E_o \equiv \{x_o\}$
$E_o \equiv S \cup \{x_o\}$
$E_o \equiv S$
$E_o[+] = \{u_j | u_j \in E_o \text{ and } \mathbf{a}_{o,j} \geq 0\}$
$\mathbf{R}_j = (c_j/\mathbf{a}_{o,j}, a_{\mathbf{1},j}/\mathbf{a}_{o,j}, \ldots a_{\mathbf{r},j}/\mathbf{a}_{o,j})$
u_J is selected by: $u_J \in E_o[+]$, and
$\mathbf{R}_J >_L \mathbf{R}_j$ for $u_j \in E_o[+]$, $j \neq J$
Test: normal or special source row selection routine
yes
is $u_J \in X+$
no
is $\mathbf{a}_{o,J} = 0$
yes
no
is any component of R out of bounds
yes
no
Special source row selection routines
v = the smallest index in $\mathbf{R}_J$ which has the form: positive integer/zero.
If $\mathbf{i}$ is the first index in $\mathbf{R}_J$ that is out of bounds then v = the limit row $\mathbf{i}$ if $\mathbf{a}_{\mathbf{i},J}/\mathbf{a}_{o,J} \geq g_{L,\mathbf{i}}/g_{L,o}$, and v = the limit row o if $\mathbf{a}_{\mathbf{i},J}/\mathbf{a}_{o,J} < g_{L,\mathbf{i}}/g_{L,o}$
Normal source row selection routine
$V(J) = \{i | {}_I[g_i/a_{i,J}] \leq \theta_J\}$
$\overset{\#}{v} \in V(J)$
Cycle Execution
1. Adjoin a Gomory cut with v as the source row and $\lambda = a_{v,J}$
2. If $a_{v,J}$ - 1 use the weakened Gomory cut. (See section 2/5)
3. Designate the cut row as the pivot row
4. Designate column J as the pivot column
5. Execute the simplex change of basis procedure
6. If a transition cycle has been executed, carry out special housekeeping procedures. (See section 2/4)

FIGURE 2/1.
References to Definitions and Discussion

1. The symbol # means that an arbitrary choice is permissible within the indicated set.
2. Most of the terms used but not defined in the flow chart are defined in section 2.2.
3. The routines for selection of u_J are described and some of the terms in those routines are defined in section 2.6.
4. The routines for selection of the source row are described and some of the terms in those routines are defined in section 2.8.
5. The cycle execution procedures are described in sections 2.3, 2.4, and 2.5.

Part III. Proof of Finiteness

3.1. Introduction

Part III is devoted to proving that the algorithm described in part II will locate an optimal integer solution in a finite number of cycles. It will become apparent quickly that the categories of cycles: transition cycles, stationary cycles, x cycles, s cycles have a special relation to the problem of proving finiteness; this is discussed in section 3.2.

It may be helpful to remark that the integer and feasible solution which occupies the basis positions may be (i) optimal for the linear programming problem which contains the given integer programming problem; or (ii) optimal for the integer but not the linear programming problem; or (iii) optimal for neither the linear nor the integer programming problem. The proof we give here does not explicitly distinguish between cases (ii) and (iii). We shall prove that a finite number of cycles in which the basic solution does not change is sufficient to achieve condition (i). Hence if (iii) describes the current basic solution a finite number of cycles must be sufficient to yield a new basic solution.

In section 3.2 we discuss the general organization of our finiteness proof.

3.2. A General Outline of the Proof

In proving that a finite number of cycles of the algorithm is sufficient to yield an optimal solution to the given problem we shall make repetitive use of the following simple device. To show that a sequence of cycles is finite we establish a two way classification of cycles whereby every cycle is called either an A cycle or a B cycle. Then we show that the total number of A cycles must be finite. This reduces the task of proving finiteness to the simpler requirement of proving that every subsequence of B cycles which follows or preceeds an A cycle is finite. The forthcoming proof uses this procedure twice, and in that process three elemental cycle types are distinguished. These cycle types and the relations between them are shown schematically in figure 3/1.

The definitions of the cycle types shown have already been given.[1]

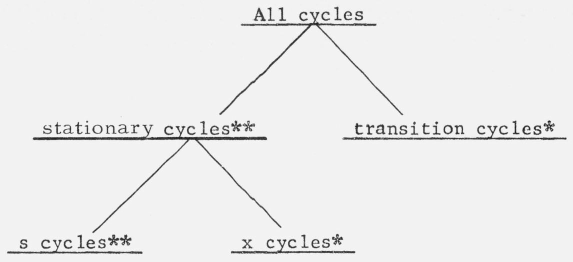
The diagram suggests the three major tasks which we must accomplish to achieve the desired proof. We must prove that:

(1) Only a finite number of transition cycles can occur;

(2) only a finite number of x cycles can occur (following the original tableau or any transition cycle); and

(3) every subsequence of s cycles following any x cycle is finite.

[1] See part II, pp. 228 and 229.



* A cycles

** B cycles

FIGURE 3/1.

The rest of part III is devoted to proving these three propositions. We are able to prove (1) and (2) by a direct and simple arguments in sections 3.3. and 3.4. The proof of (3), which is the task of the remaining sections of part III, is more difficult. In section 3.5 we outline our approach to proving (3) and there discuss the roles of the remaining sections in terms of that outline.

3.3. Proof That the Total Number of Transition Cycles Is Finite

A transition cycle always has $c_J > 0$ (which implies $c_J \geqslant 1$ since c_J must be an integer), and $\theta'_J \geqslant 1$; therefore each transition cycle results in at least a unit improvement in the criterion function value of the solution. Since the given problem is assumed to be bounded, a finite number of transition cycles must be sufficient to increase the criterion value from its level at the initial solution to the optimal criterion value.

3.4. Proof That the Total Number of x Cycles Is Finite

The original problem is assumed to have a finite number, M', of variables. In section 2.4 we showed that after any transition cycle it is possible to start with a system of equations which has at most $M'+1$ variables. In section 2.5 we described a procedure that guarantees that each variable which is nonbasic in the original tableau or immediately after a transition cycle can enter the basis only once during the following sequence of uninterrupted stationary cycles. Since each x cycle brings one x variable into the basis, the total number of x cycles (following any transition cycle or the original tableau) cannot exceed $M'+1$. Thus only a finite number of x cycles is possible.

3.5. s Cycles: Formulation, Goals, and a Guide to Subsequent Sections

To lay the foundation for a proof that every subsequence of s cycles which follows an x cycle is finite we shall in this section undertake a preliminary analysis. We begin with a reduction of (2.24) in terms of which our organization of the problem of proving finiteness may be stated. When this has been accomplished, we shall terminate this section with a summary discussion of the content of the remaining sections in this chapter.

The rest of part III is focused on a typical x cycle and the typical sequence of s cycles that follows it. We shall show that a finite sequence of s cycles is sufficient to achieve the condition:

$$C_S \leqslant 0 \text{ or } S \cap C+ = \emptyset \cdot \tag{3.1}$$

when (3.1) occurs an arbitrary x variable, $(x_0 \epsilon X+)$, enters E_0. Within a finite number of cycles after the occurrence of (3.1), another x cycle must occur.[2] Therefore out main problem is in developing a proof that a finite sequence of s cycles is sufficient to yield (3.1).

We shall let $\tilde{k}$ index the tableau which is the site of the initial x cycle. A typical subsequent tableau will be indexed by $\tilde{k}+t$. Because our remaining concern (after cycle $\tilde{k}+1$) is with s cycles and the condition (3.1) it will simplify matters to focus our attention on a truncated version of (2.24):

maximize

$$C_S^{(\tilde{k}+t)} \cdot S, \tag{3.2}$$

subject to

$$I \cdot \begin{bmatrix} X_B \\ X_{N'} \end{bmatrix} + {}_sA^{(\tilde{k}+t)} \cdot S = \begin{bmatrix} G_B^{(\tilde{k}+t)} \\ \\ G_{N'}^{(\tilde{k}+t)} \end{bmatrix} = G^{(\tilde{k}+t)}, \tag{3.3}$$

$$S \geqslant 0 \text{ and integral.}$$

In the above system we note that $G_{N'}{}^{(\tilde{k}+t)} = 0$. More precise notation would call for $S^{(k+t)}$ since the composition of this vector varies from cycle to cycle. We shall omit the superscript where there is no danger of confusion. By contrast, the variable vector $X_{N'}$ does not change (within the sequence of s cycles) as a function of t.

We may legitimately restrict our attention to (3.2) and (3.3) since this truncated system must contain the pivot column for any s cycle and since all the significant effects of the sequence of s cycles on the criterion function are reflected in (3.2). We assume of course that the pivot operations of the sequence of s cycles are carried out on the full system (2.24).

Each s cycle, $\tilde{k}+t$, $(t=2, 3, \ldots,)$ generates a new problem of the form (3.2), (3.3). It will be convenient to use an equivalent inequation form of these problems:

maximize

$$C_S^{(\tilde{k}+t)} \cdot S \tag{3.2}$$

subject to

$${}_sA^{(\tilde{k}+t)} \cdot S \leqslant G^{(\tilde{k}+t)} \tag{3.4}$$

$$S \geqslant 0 \text{ and integral.}$$

The dual[3] problem is

minimize

$$W'^{(\tilde{k}+t)} \cdot G^{(\tilde{k}+t)} \tag{3.5}$$

subject to

$$W'^{(\tilde{k}+t)} \cdot {}_sA^{(\tilde{k}+t)} \geqslant C_S^{(\tilde{k}+t)} \tag{3.6}$$

$$W'^{(\tilde{k}+t)} \geqslant 0.$$

The sequence of s cycles generates a sequence of pairs of primal and dual problems of the form (3.2), (3.4) and (3.5), (3.6). Let an optimal solution to (3.2), (3.4) be symbolized by $S^{*(\tilde{k}+t)}$. Then we shall have achieved (3.1) when, for some t,

$$C_S^{(\tilde{k}+t)} \cdot S^{*(k+t)} \leqslant W'^{(\tilde{k}+t)} \cdot G^{(\tilde{k}+t)} \leqslant 0. \tag{3.7}$$

The proof to be developed here will concentrate on showing that a finite number of s cycles will yield

$$W'^{(\tilde{k}+t)} = 0. \tag{3.8}$$

This is of course equivalent to (3.1) since (3.8) implies (3.7) which implies (3.1).

To develop the result (3.8), we shall construct a sequence of feasible solutions, one for each of the dual problems (3.5), (3.6). Each of the solutions will have exactly one variable positive, and this variable will correspond to the same row in each tableau. It will be shown that the value of this single positive dual variable declines monotonically as t increases and that it must eventually become equal to zero after a finite number of s cycles.

[2] A proof follows from these considerations: (i) in the first cycle for which $x_0 \epsilon E_0$, an s cycle can occur only if s_J has $c_J = \mathbf{a}_{0.J} = 0$; (ii) but such a cycle does not alter the data of the limit row 0 or the row c, thus every successive s cycle must have $c_J = \mathbf{a}_{0.J} = 0$; (iii) corollary IIA, below, implies that such a sequence must be finite. Thus eventually x_0 must be selected as the incoming variable. See also appendix B.

[3] This is the dual to the primal problem (3.2), (3.4) *without* the integer restriction. This lack of congruity does not invalidate (3.7), since the inequality is valid for the primal without the integer restrictions, and further restriction on the primal can only reduce the left side of (3.7). Moreover in a tableau $k+t$, for which (3.7) holds, the integer restriction in (3.4) is redundant.

In the next section, 3.6, we shall establish the basis for proving that the sequence of dual solution values decreases monotonically. In the following section, 3.7, we specify the sequence of dual solutions and apply the results of section 3.6 to show the monotonic decrease. In section 3.8, we shall establish the foundation for showing that the monotonic decrease in the value of the dual solution is of sufficient magnitude to realize (3.8) in a finite number of cycles. In section 3.9 five theorems are stated and proved to establish (3.8). The final section 3.10 relates theorem VII to (3.8).

3.6. Theorem I and Corollaries

In this section we give a theorem which is fundamental to much of the subsequent development in part III. We start with some necessary definitions, and proceed to a preliminary discussion in which we provide, along with more terminology, a lemma basic to theorem I. Then we state and prove theorem I and several corollaries. Algebraic proofs of the major propositions of this section are given in appendix C. The presentations and proofs here are informal, geometric and heuristic.

Definitions. In theorem I below the goal, roughly stated, is to show that if a set $F^{(k)}$ has certain properties, then in the next tableau a set which is a successor to $F^{(k)}$, and is designated $F^{(k+1)}$, also has these same properties. Here the successor relationship will be defined. $F^{(k)}$ is a set of nonbasic variables in tableau k. Let $D^{(k+1)}(F^{(k)})$ signify a set of nonbasic variables in the tableau $k+1$ whose definition is relative to the set $F^{(k)}$. $D^{(k+1)}(F^{(k)})$ is the "descendent" of $F^{(k)}$ in the next tableau.

$$D^{(k+1)}(F^{(k)}) \equiv \begin{Bmatrix} s_{k+1} \text{ and all the elements} \\ \text{of } F^k \text{ except } u_{J(k)} \end{Bmatrix}$$

$$D^{(k+t)}(F^{(k)}) \equiv D^{(k+t)}(D^{(k+t-1)}(\ldots(D^{(k+1)}(F^{(k)}))\ldots)).$$

In theorem I the following will be assumed:

$$F^{(k+1)} \equiv D^{(k+1)}(F^{(k)})$$

$$F^{(k+t)} \equiv D^{(k+t)} F^{(k)}).$$

Preliminary discussion. Let a tableau k be given and let j index a typical nonbasic variable u_j in tableau k. Let two rows of the tableau indexed d and n, be selected. Then we shall let the ordered pair

$$(a^{(k)}_{d,j}, a^{(k)}_{n,j})$$

represent the column j.

Since the elemental objects discussed in this section are pairs of numbers, it is both useful and appropriate to rely on geometric representation of assumed and implied situations.

The pairs or points associated with any set of the (nonbasic) columns of a single tableau may be graphically

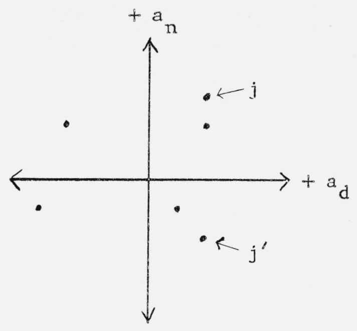


FIGURE 3/2.

represented in a diagram such as figure 3/2. In figure 3/2, two points are labeled j and j'; this designation reflects the association of these (illustrative) points with (hypothetical) columns j and j', or variables u_j and $u_{j'}$. Since the tableau under consideration is tableau k, the set of points (or indices, or columns, or variables) will be designated $F^{(k)}$. The subset of $F^{(k)}$ which consists of points to the right of the vertical axis will be designated $F(+)^{(k)}$.

The general problem under consideration in this section is the following: given a set $F^{(k)}$, which is represented graphically as in figure 3/2, how is a similar representation of $F^{(k+1)}$ related to the representation of $F^{(k)}$, when $F^{(k+1)} = D^{(k+1)}(F^{(k)})$.

In considering sets $F^{(k)}, F^{(k+1)}, \ldots$, derived from tableaus $k, k+1, \ldots$, the rows d and n remain fixed for all tableaus under consideration. The selection of rows d and n is arbitrary except for this: neither row can serve as the pivot row for any cycle in the sequence $k+1, k+2, \ldots$, under consideration.

We recall that if $u_j \in F^{(k)}$ and $u_j \neq u_{J(k)}$ then $u_j \in F^{(k+1)}$. In the subsequent analysis the definition of $F^{(k)}$ will imply that $u_{J(k)} \in F^{(k)}$; the variable which is displaced from the basis by the entry of $u_{J(k)}$ is an s variable, and this variable, s_{k+1}, "replaces" $u_{J(k)}$ in $F^{(k+1)}$.

To state the lemma that is fundamental to theorem I we must define another concept: the ratio

$$\frac{a^{(k)}_{n,J}}{a^{(k)}_{d,J}}$$

associated with the incoming variable $u_{J(k)}$. This ratio is identical to the slope[4] of a line determined by the origin and the point $(a_{d,J}, a_{n,J})$; an illustration is provided in figure 3/3, where the point J is arbitrarily selected. This line is designated L_J. The slope of L_J is $R_J = a_{n,J}/a_{d,J}$.

Now we may state our lemma which we shall call the "rule of parallel movement:"

Every point in $\mathrm{F}^{(k)}$ *will "move" to "its" position in* $\mathrm{F}^{(k+1)}$ *along a line parallel to* L_J.

[4] We assume for the moment that the point $(a_{d,J}, a_{n,J})$ does not occur at the origin.

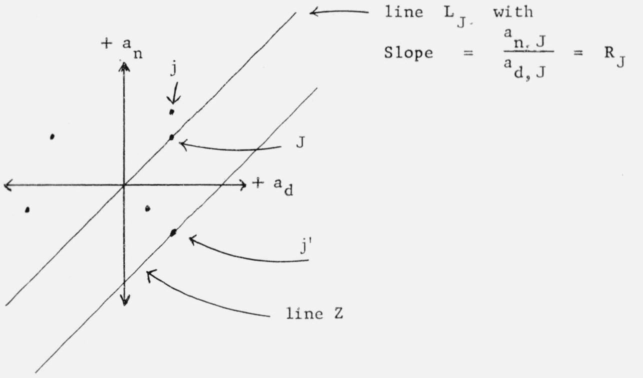


FIGURE 3/3.

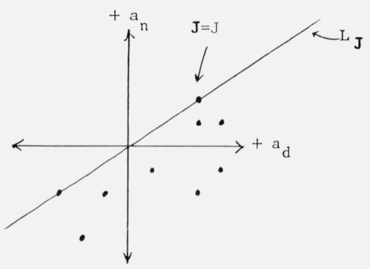


FIGURE 3/4.

As a specific example, in figure 3/3 the line Z (which is parallel to L_J) will contain the point corresponding to j' in $F^{(k+1)}$.

Proof of this rule follows directly from the observation that, excepting the the pivot row, the procedure for changing the basis in the simplex method calls for algebraically adding multiples of the pivot column (J) to all the other columns. In terms of the points represented in figure 3/3, multiples of J are added to the other points such as j'. The rules for vector addition apply here—specifically the parallelogram law applies—whence it follows that each point is displaced (if at all) along a line parallel to L_J. The point J will be replaced by a point associated with s_{k+1}. In the latter event we may note that the column associated with s_{k+1} in tableau $k+1$ is the negative of the column associated with $u_{J(k)}$ in tableau k (with the exception of the pivot row). Hence the "movement" from the point J to the point that replaces it in $F^{(k+1)}$ is along, and therefore parallel to, the line L_J.

THEOREM I. *The hypothesis of theorem I is as follows: a set of points* $F^{(k)}$ *exists and has the following properties (which are exemplified by the set of points depicted in figure 3/4):*

(I1) *At least one point in* $F^{(k)}$ *is to the right of the vertical axis, (i.e.,* $F(+)^{(k)}$ *is not empty).*

(I2) *Let a line* $L_{\mathbf{J}}$ *be determined by the origin and a point*[5] (J) *in* $F(+)$ *which maximizes the ratio* $a_{n,j}/a_{d,j}$ *over all* $j \in F(+)^{(k)}$. *Then it is our assumption that the line* $L_{\mathbf{J}}$ *will include or be above all other points in* $F^{(k)}$.

[5] We distinguish $\mathbf{J}$ from J only because we leave open the possibility that $(a_{d,J}, a_{n,J})$ may occur at the origin. Otherwise (see (I3)) we could identify J with $\mathbf{J}$.

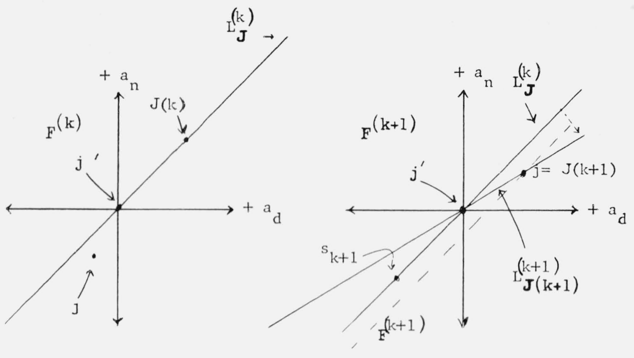


FIGURE 3/5.

(I3) *The incoming variable* u_j *for cycle* $k+1$ *is represented by a point that is on the line* $L_{\mathbf{J}}$ *and is either to the right of the vertical axis or is located at the origin.*

The conclusion of theorem I is the following: (II) is true for $F^{(k+1)} => $ (I2) *is true for* $F^{(k+1)}$.

PROOF OF THEOREM I. We shall consider two cases: (i) $L_{\mathbf{J}(k)} = L_{\mathbf{J}(k+1)}$ and (ii) $L_{\mathbf{J}(k)} \neq L_{\mathbf{J}(k+1)}$. In case (i), (I2) must hold for $F^{(k+1)}$. If some $u_j \in F^{k+1}$ were above $L_{\mathbf{J}(k+1)}$ this would imply either that (I2) was not satisfied for $F^{(k)}$ or that the rule of parallel movement had been violated in the transition from tableau k to tableau $k+1$. In case (ii), the rule of parallel movement and the definition of $L_{\mathbf{J}}$ in (I2) imply that the transition from $L_{\mathbf{J}(k)}$ to $L_{\mathbf{J}(k+1)}$ is accomplished by a clockwise rotation of the line to the position $L_{\mathbf{J}(k+1)}$. Since $L_{\mathbf{J}(k+1)}$ is therefore above $L_{\mathbf{J}(k)}$ to the left of the vertical axis, (I2) and the rule of parallel movement guarantee that no point on the left of the vertical axis in $F^{(k+1)}$ will be above $L_{\mathbf{J}(k+1)}$. By the definition of $L_{\mathbf{J}(k+1)}$, the rotation from the position of $L_{\mathbf{J}(k)}$ to the position of $L_{\mathbf{J}(k+1)}$ is terminated when the first point in $F(+)^{(k+1)}$ "touches" the line. Thus all points in $F(+)^{(k+1)}$ are on or below $L_{\mathbf{J}(k+1)}$. Figure 3/5 contains a hypothetical illustration of case (ii).

In the following group of corollaries to theorem I it is implicit in each case that the assumed conditions of theorem I are satisfied. We shall generally designate the ratio $a_{n,j}^{(k)}/a_{d,j}^{(k)}$ by $R_j^{(k)}(n, d)$. In the statement of the corollaries below, we shall simplify $R_j^{(k)}(n, d)$ to $R_j^{(k)}$, since only one pair of rows—n and d—are considered. If $(a_{d,J(k)}, a_{n,J(k)})$ should occur at the origin we shall stipulate that $R_{J(k)}$ is equal to the slope of $L_{\mathbf{J}}^{(k)}$. Proofs for the corollaries are given in appendix C.

COROLLARY IA. *If all points on* $L_{\mathbf{J}}^{(k)}$ *are to the left of or on the vertical axis in* $F^{(k+1)}$, *then*

$$R_{J(k)}^{(k)} > R_{J(k+1)}^{(k+1)}$$

COROLLARY IB. *Some point* j *(where* $j \in F^{(k)}$ *and* $j \neq J(k)$*) on* $L_{\mathbf{J}}^{(k)}$ *is on the right of the vertical axis in* $F^{(k+1)}$, *if and only if*

$$R_{J(k)}^{(k)} = R_{J(k+1)}^{(k+1)}.$$

Corollary IC

$$R^{(k)}_{J(k)} \geqslant R^{(k+1)}_{J(k+1)}.$$

Corollary ID. *If* $(a^{(k)}_{d,J(k)}, a^{(k)}_{n,J(k)})$ *is the only point in* $F^{(k)}$ *on the line* $L^{(k)}_{J}$, *then*

$$R^{(k)}_{J(k)} > R^{(k+1)}_{J(k+1)}.$$

Corollary IE. *Let* $E^{(k)}_{n}$ *designate the set of all points in* $F^{(k)}$ *and on* $L^{(k)}_{J}$. *For* $t>0$, *if*

$$R^{(k)}_{J(k)} = R^{(k+t)}_{J(k+t)}, \text{ then } u_{J(k+t)} \in D^{(k+t)}(E^{(k)}_{n}).$$

Corollary IF. *If* $R^{(k)}_{J(k)} = R^{(k+t)}_{J(k+t)}$, *then*

$$E^{(k+t)}_{n} = D^{(k+t)}(E^{(k)}_{n}).$$

Corollary IG. *If* $R^{(k+1)}_{J(k+1)} < R^{(k)}_{J(k)}$, *then for all* $u_j \in E^{(k+1)}_{n}$,

$$a^{(k+1)}_{d,j} \geqslant 0.$$

Corollary IH. *If* $R^{(k)}_{J(k)} \geqslant 0$, $R^{(k+1)}_{J(k+1)} < 0$, *and* $a^{(k+1)}_{d,j} \leqslant 0$, *then*

$$a^{(k+1)}_{n,j} \leqslant 0.$$

Corollary IJ. *If* (i) *the assumed conditions of theorem I are satisfied by* $F^{(k)}$,
(ii) $F(+)^{(k+t)}$ *is not empty for* $t \leqslant t'$, *and*
(iii) $u_{J(k+t)}$ *is selected in conformity to* (I3) *for* $t \leqslant t'$,
then the assumed conditions of theorem I are satisfied for $F^{(k+t')}$.

We close this section with the statement of a self-evident proposition that will be useful later.

If $F^{(k)}$ is contained in the half-open half space to the right of the vertical axis and including the lower half of the vertical axis, and if (I1) is satisfied, then (I2) must be satisfied. (3.9)

3.7. The Sequence of Dual Solutions

In this section we shall specify the sequence of dual solutions to (3.5), (3.6) which was described generally in section 3.5. With the aid of theorem I we shall show that the sequence of solution values declines monotonically.

Our starting point is the system (2.24) when the tableau index is $\tilde{k}$. Recall that $\tilde{k}+1$ is the index of a typical x cycle, that $C^{\tilde{k}}_{S} \leqslant 0$, that $u_{J(\tilde{k})}$ is an x variable, and that $c^{(k)}_{J(\tilde{k})} > 0$. We assume that a constraint which bounds the incoming variable has been adjoined to (2.24). This constraint is

$$x_{J(\tilde{k})} \leqslant g^{(\tilde{k})}_{LJ(\tilde{k})}. \tag{3.10}$$

Now let ${}_xA^{(\tilde{k})}_{J(\tilde{k})}$ signify the column of ${}_xA^{(\tilde{k})}$ which corresponds to $x_{J(\tilde{k})}$, and consider the system which results from adjoining this column and variable to (3.4), (and (3.3)).

$${}_sA^{(\tilde{k})} \cdot S + {}_xA^{(\tilde{k})}_{J} \cdot x_{J(\tilde{k})} \leqslant G^{(\tilde{k})}. \tag{3.11}$$

As the result of the cycle $\tilde{k}+1$, the system (3.11) acquires the form of (3.4) with $t=1$. Generally, the "descendants" of the system (3.11) have the form of (3.4). To be more precise let a set E_0 be defined[6] by

$$E^{(\tilde{k})}_{0} = \{u_j | u_j \text{ is a component of } S^{(\tilde{k})} \text{ or } u_j = x_{J(\tilde{k})}\}. \tag{3.12}$$

Then defining $E^{(\tilde{k}+t)}_{0} = D^{(\tilde{k}+t)}(E^{(\tilde{k})}_{0})$, it is clear that $E^{(\tilde{k}+1)}_{0}$ contains exactly the components of $S^{(\tilde{k}+1)}$ and $E^{(\tilde{k}+t)}_{0}$ contains exactly the components of $S^{(\tilde{k}+t)}$.

The criterion function corresponding to (3.11) is

$$C^{(\tilde{k})}_{S} S + c^{(\tilde{k})}_{J(\tilde{k})} \cdot x_{J(\tilde{k})}, \tag{3.13}$$

which is to be maximized.

Next we shall construct an initial feasible solution for the dual problem to (3.11), (3.13). Consider the row of (3.11) which corresponds to the bounding constraint (3.10):

$$0 \cdot S + 1 \cdot x_{J(\tilde{k})} \leqslant g^{(\tilde{k})}_{LJ(\tilde{k})}. \tag{3.14}$$

We shall assign this row the (arbitrary) subscript 0. Thus $g_{L0} \equiv g^{(\tilde{k})}_{LJ(\tilde{k})}$ and $\mathbf{a}^{(\tilde{k})}_{0,j} = 0$ unless $j=J$, in which case $\mathbf{a}^{(\tilde{k})}_{0,J} = 1$.

The dual variable, $\mathbf{w}^{(\tilde{k})}_{0}$, associated with the row (3.14) is assigned the value $c^{(\tilde{k})}_{J(\tilde{k})}$. The rest of the dual variables are zero. This solution is obviously feasible: the dual constraint associated with the J column is exactly satisfied; the other dual constraints are also satisfied since $\mathbf{w}^{(\tilde{k})}_{0} \cdot \mathbf{a}_{0,j} = 0$ and $c^{(\tilde{k})}_{j} \leqslant 0$ for all columns j corresponding to components of S.

To show that every subsequent dual problem (3.5), (3.6) has a feasible solution in which $\mathbf{w}^{(\tilde{k}+t)}_{0} \geqslant 0$ and $w^{(\tilde{k}+t)}_{i} = 0$ if i is not the index of the limit row 0, we apply the results of the preceding section.

We identify the row n of theorem I with the row c of (3.13) and (3.12). We identify the row d with the limit row 0 in (3.1) and (3.4). We identify $F^{(k)}$ with $E^{(\tilde{k})}_{0}$ and define[7] $E(+)_0$ analogously to $F(+)$.

With respect to these identifications we note that the assumed conditions of theorem I are satisfied in tableau $\tilde{k}$. (I1) is satisfied because $\mathbf{a}_{0,J} = 1$. (I2) is satisfied because $\mathbf{a}_{0,J} = 1$ and $c_J > 0$ while $\mathbf{a}_{0,j} = 0$ and $c_j \leqslant 0$ for all $j \neq J$, which with (3.9) implies (I2). (I3) is satisfied since x_J is the only element in $E(+)^{(\tilde{k})}_{0}$.

We note that

$$\mathbf{w}^{(\tilde{k})}_{0} = \frac{c^{(\tilde{k})}_{J(\tilde{k})}}{\mathbf{a}^{(\tilde{k})}_{0,J(\tilde{k})}}.$$

[6] The definition (3.12) is consistent with and more special (in that it relates to $\tilde{k}$) than the definitions given in section 2.6.

[7] This is consistent with (2.21).

In general if (I1), (I2) and (I3) are true, a feasible solution to dual problem (3.5), (3.6) is given by

$$\mathbf{w}_0^{(\tilde{k}+t)} = \frac{c_{J(\tilde{k}+t)}^{(\tilde{k}+t)}}{\mathbf{a}_{0,\,J(\tilde{k}+t)}^{(\tilde{k}+t)}}, \qquad (3.15)$$

$w_i^{(\tilde{k}+t)} = 0$, if i is not the index of the limit row 0. (3.16)

Thus to show that a solution (3.15), (3.16) is available for each dual problem (3.5), (3.6) we must show that the assumed conditions of corollary IJ are satisfied in every tableau $\tilde{k}+t$ which results from an s cycle. Since we have already shown that condition (i) of Corollary IJ holds, we shall now deal with (iii) and (ii). The selection of the incoming variable according to rules 3(J) and 3a(J) of section 2.6 will always conform to (I3). For all pertinent cases (I1) must also be satisfied, since if $E(+)_0^{(\tilde{k}+t)}$ is empty for some t, then we can conclude from corollary IH that for some $t' \leq t$, $c_j \leq 0$ for all $u_j \epsilon E_0^{(\tilde{k}+t')}$.

Thus corollary IJ applies. This establishes the sequence of feasible dual solutions (3.15), (3.16). Corollary IC implies

$$\mathbf{w}_0^{(\tilde{k}+t)} \geqslant \mathbf{w}_0^{(\tilde{k}+t+1)}. \qquad (3.17)$$

Since g_{L0} does not change during the sequence of s cycles, (3.17) implies a monotonic decrease in the value of the dual solution.

It is convenient to establish here, for later reference, the following proposition:

$$\textit{if } \mathbf{a}_{0,\,J(\tilde{k}+t)}^{(\tilde{k}+t)} = 0, \textit{ then } c_{J(\tilde{k}+t)}^{(\tilde{k}+t)} = 0. \qquad (3.18)$$

The rules for selecting u_J imply $c_J \geqslant 0$. But $c_J > 0$ is incompatible with the hypothesis of (3.18) because the finite ratio $c_{J(\tilde{k})}^{(\tilde{k})} / \mathbf{a}_{0,\,J(\tilde{k})}^{(\tilde{k})}$ stands first in a monotonically decreasing sequence. Thus $c_{J(\tilde{k}+t)}^{(\tilde{k}+t)} = 0$.

In the next section we establish the basis for showing that the monotonic decrease in (3.17) is of sufficient magnitude to achieve (3.8) for a finite t.

3.8. Theorem II and Corollaries

Theorem II deals with the effects of a change of basis on the source row, v, when $\lambda = a_{v,\,J}$ and the cut row serves as the pivot row. Corollaries IIA and IIB develop implications of theorem II that are useful to proving finiteness.

THEOREM II. *Let the incoming variable for the cycle* $\mathrm{k}+1$ *be* $\mathrm{u}_{\mathrm{J(k)}}$ *and let* v *be the index of the source row. It is assumed that once the new equation has been adjoined to the system, it qualifies as the pivot row for bringing* $\mathrm{u}_{\mathrm{J(k)}}$ *into the basis; and it is assumed that* $\mathrm{a}_{\mathrm{v,\,J}} > 0$. *The cut is*

$$\mathrm{s}_{\mathrm{k}+1} + \sum_{\mathrm{j}} \left[\frac{\mathrm{a}_{\mathrm{v,\,j}}^{(\mathrm{k})}}{\mathrm{a}_{\mathrm{v,\,J(k)}}^{(\mathrm{k})}}\right]_I \mathrm{u}_{\mathrm{j}} = \left[\frac{\mathrm{g}_{\mathrm{v}}^{(\mathrm{k})}}{\mathrm{a}_{\mathrm{v,\,J(k)}}^{(\mathrm{k})}}\right]_I. \qquad (3.19)$$

With (3.19) serving as the pivot row, $\mathrm{u}_{\mathrm{J(k)}}$ *is brought into the basis. The following results are implied:*

$\mathrm{a}_{\mathrm{v,\,j}}^{(\mathrm{k+1})} \geqslant 0$, *for every* j *that is not the index of* $\mathrm{s}_{\mathrm{k}+1}$. (3.20)

$\mathrm{a}_{\mathrm{v,\,j}}^{(\mathrm{k+1})} < \mathrm{a}_{\mathrm{v,\,J(k)}}^{(k)}$, *for all* j. (3.21)

PROOF OF THEOREM II. Both results stem from the formula for $a_{v,j}^{(k+1)}$:

$$a_{v,j}^{(k+1)} = a_{v,j}^{(k)} - \left[\frac{a_{v,j}^{(k)}}{a_{v,J(k)}^{(k)}}\right]_I \cdot a_{v,J(k)}^{(k)}. \qquad (3.22)$$

Let $a_{v,j}^{(k)}$ be expressed as a function of $a_{v,J(k)}^{(k)}$, as follows:

$$a_{v,j}^{(k)} = T \cdot a_{v,J(k)}^{(k)} + r_{v,\,j}, \text{ in which} \qquad (3.23)$$

$$a_{v,J(k)}^{(k)} > r_{v,\,j} \geqslant 0, \text{ and } T \text{ is an integer.} \qquad (3.24)$$

Then

$$T = \left[\frac{a_{v,j}^{(k)}}{a_{v,\,J(k)}^{(k)}}\right]_I. \qquad (3.25)$$

Substitution of (3.23) and (3.25) into the right side of (3.22) yields

$$a_{v,j}^{(k+1)} = r_{v,\,j} \text{ for every } u_j \text{ except } s_{k+1}. \qquad (3.26)$$

The coefficient of row v in the (newly nonbasic) column $k+1$ is

$$a_{v,\,k+1}^{(k+1)} = -a_{v,\,J(k)}^{k}. \qquad (3.27)$$

Thus (3.20) follows from (3.26) and (3.24); (3.21) follows from (3.26), (3.24), and (3.27) since the source row will always have $a_{v,J(k)}^{(k)} > 0$.

COROLLARY IIA. *If* $\mathrm{k}+1, \mathrm{k}+2, \ldots, \mathrm{k}+\mathrm{t}, \mathrm{k}+\mathrm{t}+1, \ldots$ *is a sequence of successive* s *cycles, and if*

$$\mathbf{a}_{0,\,\mathrm{J(k)}}^{(\mathrm{k})} = 0, \textit{ then } \mathbf{a}_{0,\,\mathrm{J(k+t)}}^{(\mathrm{k+t})} > 0,$$

for some finite t.

PROOF OF IIA. Let $\mathbf{a}_{0,J(k)}^{(k)} = 0$ and let $\mathbf{i}$ be the first index of a component of $\mathbf{R}_J^{(k)}$ which has a nonzero numerator. From (3.18) we may conclude $\mathbf{i} \geqslant 1$. Rule 3a(J)Ib of section 2.6 implies $a_{\mathrm{i},J(k)}^{(k)} > 0$ if $E(+)_0^{(k)}$ is not empty. Since we have shown[8] $E(+)_0^{(k)}$ is not empty, we have $a_{\mathrm{i},J(k)}^{(k)} > 0$. Rule 4($v$) in section 2.7 determines that row $\mathbf{i}$ is the source row for cycle $k+1$. Then as a consequence of theorem II we have

$$0 \leqslant a_{\mathrm{i},\,J(k+1)}^{(k+1)} < a_{\mathrm{i},\,J(k)}^{(k)} \qquad (3.28)$$

If $\mathbf{a}_{0,\,J(k+1)}^{(k+1)} > 0$ the corollary is proved.

Suppose $\mathbf{a}_{0,\,J(k+1)}^{(k+1)} = 0$. Since all the components preceding $a_{\mathrm{i},\,J}/\mathbf{a}_{0,\,J}$ in $\mathbf{R}_{J(k)}^{(k)}$ have the form 0/0, the

[8] See above section 3.7.

rows which provide the data for these components will not be altered by the change of basis in cycle $k+1$. Let $\mathbf{i}'$ index the numerator of any component which precedes $\mathbf{i}$ in $\mathbf{R}$. Then if $\mathbf{a}^{(k+1)}_{0,\,J(k+1)}=0$, we must also have $a^{(k+1)}_{\mathbf{i}',\,J(k+1)}=0$. We can rule out $a^{(k+1)}_{\mathbf{i}',\,J(k+1)}>0$ since this implies $a^{(k)}_{\mathbf{i}',\,J(k+1)}>0$ and $\mathbf{a}^{(k)}_{0,\,J(k+1)}=0$ which contradicts the selection of $u_{J(k)}$ over $u_{J(k+1)}$ as the incoming variable for cycle $k+1$. We can rule out $a^{(k+1)}_{\mathbf{i}',\,J(k+1)}<0$ since this contradicts the selection of $u_{J(k+1)}$ as the incoming variable for cycle $k+2$. Thus if $\mathbf{a}^{(k+1)}_{0,\,J(k+1)}=0$, row $\mathbf{i}$ is again the source row, or $a^{(k+1)}_{\mathbf{i},\,J(k+1)}=0$. If $a^{(k+1)}_{\mathbf{i},\,J(k+1)}>0$, then $\mathbf{i}$ is the source row for cycle $k+2$ and the tableau indices in (3.28) can all be increased by 1. Thus a finite sequence of cycles must result in $a_{\mathbf{i},\,J}=0$. Then the index of the first nonzero numerator in $\mathbf{R}_J$ must become $\geqslant \mathbf{i}+1$.

We have just observed that a finite number of cycles is sufficient to increase the index of the first nonzero numerator in $\mathbf{R}_J$ by at least one. Since the number of indices in $\mathbf{R}$ is finite, a finite number of successive cycles for which $\mathbf{a}_{0,\,J}=0$ will lead to $a_{\mathbf{i},\,J}=0$ for all indices $\mathbf{i}$ in $\mathbf{R}$. This is a contradiction since it implies[9] that u_J corresponds to a null column vector in ${}_sA^{(k+t)}$ for some finite t'. Hence for some $t<t'$ we must have $\mathbf{a}^{(k+t)}_{0,\,J}>0$.

COROLLARY IIB. *The hypothesis consists of three conditions that are assumed to hold for a sequence of* s *cycles* k+1, . . . , k+t, . . . , k+t':

$$\textit{if } \mathbf{a}^{(k+t)}_{0,\,J(k+t)}=0 \textit{ then } a^{(k+t)}_{i,\,J(k+t)}=0, \textit{for } t<t' \tag{3.29}$$

$$\textit{if } a^{(k+t)}_{i,\,J(k+t)}>g_i \textit{ then row } i \textit{ is the source row for cycle } k+t+1, \textit{for } t<t'; \tag{3.30}$$

$$a^{(k)}_{i,\,J(k)}>g_i. \tag{3.31}$$

The following condition is implied.

$$a^{(k+t')}_{i,\,J(k+t')}\leqslant g_i \text{ and } \mathbf{a}^{(k+t')}_{0,\,J(k+t')}>0 \text{ for some finite } t'. \tag{3.32}$$

PROOF OF IIB. In each tableau $k+t$ one of the following cases must occur:

$$a^{(k+t)}_{i,\,J(k+t)}>g_i \tag{3.33}$$

$$\mathbf{a}^{(k+t)}_{o,\,J(k+t)}=0 \text{ and } a^{(k+t)}_{i,\,J(k+t)}=0 \tag{3.34}$$

$$a^{(k+t)}_{i,\,J(k+t)}\leqslant g_i \text{ and } \mathbf{a}^{(k+t)}_{0,\,J(k+t)}>0. \tag{3.35}$$

If (3.33) occurs $a^{(k+t+1)}_{i,\,J(k+t+1)}$ is reduced, because of (3.30) and (3.21), by at least a unit. This reduction is not lost by the occurrence of (3.34), since a cycle for which (3.34) holds does not alter the data of row i. Corollary IIA implies that (3.34) can only occur successively for a finite number of cycles; thus (3.33) must reoccur at finite intervals. A finite[10] number of occurrences of (3.33) is sufficient force the occurrence of (3.35).

[9] See appendix B.

3.9. Theorems III, IV, V, VI, and VII

Definitions. In this section we state and prove five theorems. The first three theorems (III, IV and V) show that a sequence of s cycles generates a sequence of vectors $\mathbf{R}_{J(k)}$ with the property $\mathbf{R}_{J(k)} >_L \mathbf{R}_{J(k+1)}$. The next two theorems (VI and VII) use this result to show that a finite sequence of s cycles must reduce the first component of $\mathbf{R}_J$ to zero.

The definitions below are a necessary preliminary to the statement of the theorems. We shall provide some informal discussion of the concepts defined below since these definitions constitute the linkage between the rules of part II which govern the procedure of the primal algorithm and the general concepts developed by theorem I and its corollaries.

The procedure which selects u_J in a stationary cycle may be usefully viewed as a process of progressive elimination, as follows.

First the set of all nonbasic variables is narrowed to E_0. Then the first component of $\mathbf{R}$ is used as a test to select a subset of E_0 in which all elements are tied with maximal values for $c_j/\mathbf{a}_{0,\,j}$. We call this subset E_c. Next the second component of $\mathbf{R}$ is used to select a subset of E_c in which all elements are tied with equal and maximal values for the first two components of R. We call this subset E_1. This process continues until a single element subset $E_\mathbf{i}$ is obtained and we have, by definition, $E_\mathbf{i}=\{u_J\}$. We shall provide formal definitions for this nested sequence of subsets. Before that it may be helpful to give an example and a graphical presentation.

In figure 3/6(a) we present a tableau segment showing some of the coefficients at the intersection of the columns of five nonbasic variables in E_0 with the rows that provide the data for the first four components of $\mathbf{R}$. In figure 3/6(b) there are four pair of axes corresponding to the first four components of $\mathbf{R}$. All points corresponding to $u_j \in E_0$ are plotted on the top graph, all points corresponding to $u_j \in E_c$ are plotted on the second graph, and so on.

	u_1	u_2	u_3	u_4	u_5
c row	4	1	2	0	-1
$\mathbf{a}_0$ row	4	4	2	0	-1
a_1 row	2	—	1	0	-3
a_2 row	-2	—	-1	-1	—
a_3 row	0	—	2	—	—

FIGURE 3/6(a)

[10] It is necessary to assume, to achieve this result, that the "starting" value of $a_{i,\,J}$ on the left hand side of (3.31) is finite. This will be assumed, not on the basis that any $a^k_{i,\,j}$ generated as a result of the operation of the algorithm is certain to be bounded, since the latter assumption actually can be shown to directly imply the conclusion which we shall subsequently deduce with the aid of theorem II and its corollaries.

A weaker assumption is sufficient namely: for every finite k, a finite bound $M(k)$ exists such that $|a^k_{i,j}|<M(k)$. This assumption is also implicit in corollary IIA.

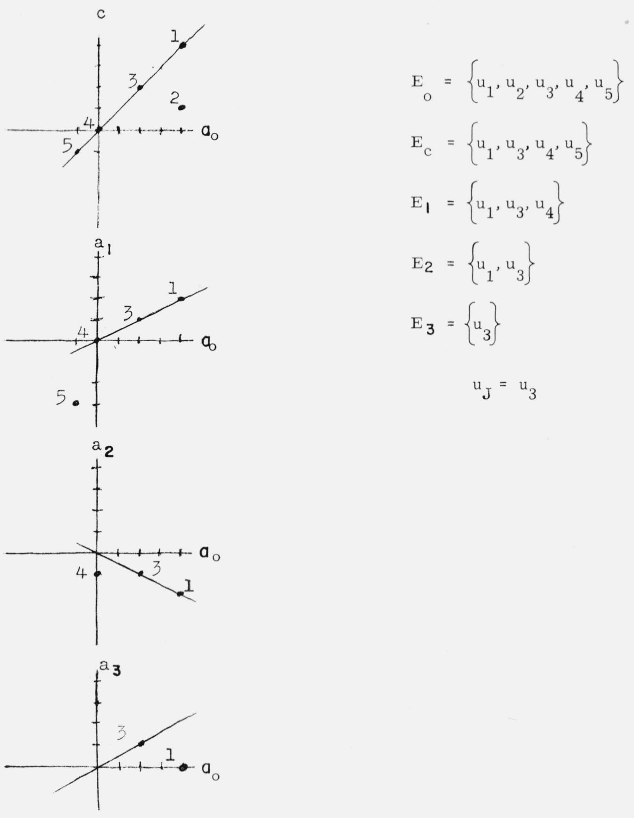

c
1
3
2
4
5
a_o
a_1
a_2
a_3
$E_o = \{u_1, u_2, u_3, u_4, u_5\}$
$E_c = \{u_1, u_3, u_4, u_5\}$
$E_1 = \{u_1, u_3, u_4\}$
$E_2 = \{u_1, u_3\}$
$E_3 = \{u_3\}$
$u_J = u_3$

FIGURE 3/6(b).

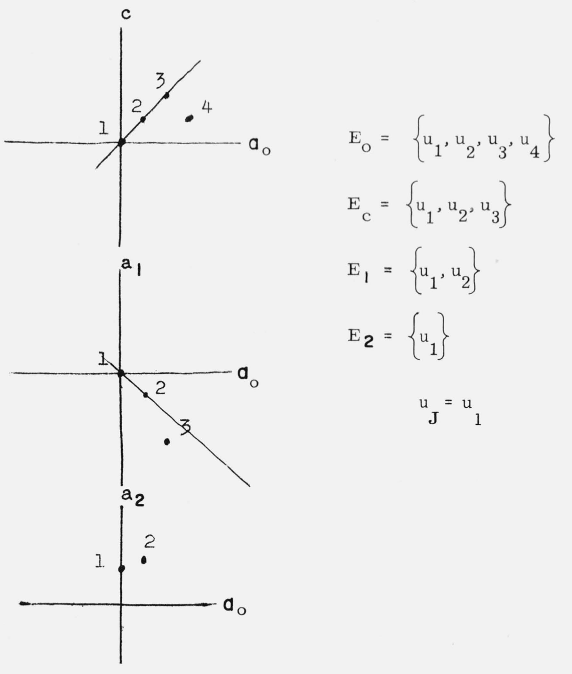


FIGURE 3/6(c).

Figures 3/6(a) and 3/6(b) are representations of the same data. In figure 3/6(c) we represent graphically a distinct situation in which $\mathbf{a}_{0,J}=0$. We note that since the horizontal dimension is identical for all graphs in figure 3/6(b) or 3/6(c) all the points which represent a given variable are on a single vertical line. Therefore movement of a point horizontally (from tableau to tableau) on one graph implies horizontal movement of corresponding points on all graphs.

The reader may find diagrams such as 3/6(b) and 3/6(c) helpful as a means of visualizing and keeping track of various details in the proofs that follow. We shall now return to the task of developing formal definitions.

The set $E_0^{(k)}$ has already been defined.[11] Here we define a sequence of nested subsets of $E_0^{(k)}$.

$$E_c^{(k)} \equiv \{u_j | u_j \epsilon E_0^{(k)} \text{ and } c_j^{(k)}/\mathbf{a}_{0,j}^{(k)} = \max_{u_{j'}\epsilon E(+)_0^{(k)}} [c_{j'}^{(k)}/\mathbf{a}_{o,j'}^{(k)}]\}$$

$$E_{\mathbf{i}}^{(k)} \equiv \{u_j | u_j \epsilon E_{\mathbf{i}-1}^{(k)} \text{ and } a_{\mathbf{i},j}^{(k)}/\mathbf{a}_{0,j}^{(k)} = \max_{u_{j'}\epsilon E(+)_{\mathbf{i}-1}^{k}} [a_{\mathbf{i},j'}^{(k)}/\mathbf{a}_{o,j'}^{(k)}]\} \cdot \qquad (3.36)$$

This sequence of subsets of E_o has the same order of subscripts as the sequence of numerators in $\mathbf{R}$. Thus if $\mathbf{i}=\mathbf{1}$, then $\mathbf{i}-1=c$. The component $a_{\mathbf{i},j}^{(k)}/\mathbf{a}_{0,j}^{(k)}$ of $\mathbf{R}_j^{(k)}$ is derived from data from the same row $\mathbf{i}$ which is used in the definition of $E_{\mathbf{i}}^{(k)}$. We use (3.36) to define subsets $E_c^{(k)}, E_{\mathbf{1}}^{(k)}, \ldots, E_{\mathbf{N}(k)}^{(k)}$, where the definition of the last index $\mathbf{N}(k)$ depends on whether $\mathbf{a}_{o,J(k)}^{(k)}=0$ or $\mathbf{a}_{o,J(k)}^{(k)}>0$. If $\mathbf{a}_{o,J(k)}^{(k)}=0$, $\mathbf{N}(k)+1$ is the index of the first positive numerator in $\mathbf{R}_{\mathrm{J}}^{(k)}$. For example, in figure 3/6(c) $\mathbf{N}(k)=1$. If $\mathbf{a}_{0,J(k)}^{(k)}>0$, $\mathbf{N}(k)+1$ is the first set in the sequence $E_0^{(k)}, E_c^{(k)}, E_{\mathbf{1}}^{(k)}, \ldots$ which contains only one element. The existence of such a set is guaranteed (see appendix B). In figure 3/6(b), $\mathbf{N}(k)=2$.

As with the definition of $\mathbf{R}$, a problem occurs in interpreting (3.36) in relation to variables u_j which have $\mathbf{a}_{0,j}^{(k)}=0$. As we have observed, set E_c is the subset of all $u_j \epsilon E_0$ for which the first component of $\mathbf{R}_j$ is not dominated by some other $u_{j'}\epsilon E_0$. The subset $E_{\mathbf{1}}$ is the subset of $u_j \epsilon E_c$ for which the first two components of $\mathbf{R}_j$ are not dominated by some other $u_{j'}\epsilon E_c$, and so on.

Now if $\mathbf{a}_{oj}=0$ and $c_j=0$ one cannot determine the lexicographic priority of $\mathbf{R}_j$ and $\mathbf{R}_{j'}$, when $\mathbf{a}_{0,j'} \geqslant 0$, by reference to the first component of $\mathbf{R}$. Therefore we shall include in E_c all $u_j \epsilon E_0$ for which $c_j=0$ and $\mathbf{a}_{0,j}=0$, and we shall include in $E_{\mathbf{i}}$ all $u_j \epsilon E_{\mathbf{i}-1}$ for which $\mathbf{a}_{o,j}=0$ and $a_{\mathbf{i}',j}=0$ for all $\mathbf{i}' \leqslant \mathbf{i}$. This equivalent to stipulating $0/0 \equiv \max_{u_{j'}\epsilon E(+)_{\mathbf{i}-1}^{(k)}} [a_{\mathbf{i},j'}^{(k)}/\mathbf{a}_{o,j'}^{(k)}]$ in (3.36).

[11] See p. 230.

In the following theorems and proofs the symbol $\frac{a^{(k)}_{\mathbf{i},j}}{\mathbf{a}^{(k)}_{0,j}}$ will be abbreviated to $R^{(k)}_j(\mathbf{i})$.

We will employ the symbol $B(k)$ to represent the following circumstance:

$$B(k) \equiv \begin{cases} (1)\ (I1), (I2), \text{ and } (I3) \text{ are true with} \\ \quad F^{(k)} \equiv E^{(k)}_{\mathbf{i}} \\ \quad \text{row } d \equiv \text{the limit row } 0 \\ \quad \text{row } n \equiv \text{row } \mathbf{i}+1 \\ \quad \text{for all } \mathbf{i} < \mathbf{N}(k). \\ (2)\ \text{If } \mathbf{a}^{(k)}_{0,J(k)} > 0, \text{ then (1) is also true for} \\ \quad \mathbf{i} = \mathbf{N}(k). \\ (3)\ \text{If } \mathbf{a}^{(k)}_{0,J(k)} = 0, \text{ then } \mathbf{a}^{(k)}_{0,j} \geqslant 0 \text{ for all} \\ \quad u_j \epsilon E^{(k)}_{\mathbf{N}(k)}. \end{cases}$$

THEOREMS

THEOREM III: $\mathrm{k} = \tilde{\mathrm{k}}$ *implies* $\mathrm{B(k)}$

THEOREM IV: $\mathrm{B(k)}$ *implies* $\mathbf{R}^{(\mathrm{k})}_{\mathrm{J}} >_{\mathrm{L}} \mathbf{R}^{(\mathrm{k}+1)}_{\mathrm{J}}$

THEOREM V: $\mathrm{B(k)}$ *implies* $\mathrm{B(k+1)}$

THEOREM VI: *For every* k *there exists a finite* t *such that* $\mathrm{R}^{(\mathrm{k})}_{\mathrm{J}}(\mathrm{c}) > \mathrm{R}^{(\mathrm{k}+\mathrm{t})}_{\mathrm{J}}(\mathrm{c})$

THEOREM VII: *For every* k *there exists a finite* t *such that* $\mathrm{R}^{(\mathrm{k})}_{\mathrm{J}}(\mathrm{c}) > 0$ *implies* $\mathrm{R}^{(\mathrm{k}+\mathrm{t})}_{\mathrm{J}}(\mathrm{c}) \leqslant 0$.

PROOFS

PROOF OF THEOREM III. From the definition of $E_{\mathbf{i}}$, ($\mathbf{i} \leqslant \mathbf{N}(k)$), $E_{\mathbf{i}}$ must contain at least one element of $E(+)_{\mathbf{i}-1}$. Hence (I1) is satisfied for all $\mathbf{i} \leqslant \mathbf{N}(k)$.

It will be recalled[12] that in tableau $\tilde{k}$, $\mathbf{a}_{0,j} = 0$ if j is the index of an s variable; if j is the index of the unique x variable in $E^{(\tilde{k})}_0$, then $\mathbf{a}_{0,j} = 1$. Since every $E_{\mathbf{i}}$ is a subset of E_0, $\mathbf{a}_{0,j} \geqslant 0$ for every $u_j \epsilon E^{(\tilde{k})}_{\mathbf{i}}$. From (3.9) we can conclude that (12) is satisfied for $\mathbf{i} < \mathbf{N}(k)$ if $\mathbf{a}_{0,J} = 0$ and for $\mathbf{i} \leqslant \mathbf{N}(k)$ if $a_{0,J} > 0$.

The satisfaction of (I3) is guaranteed by the rules 3(J) and 3a(J) for the selection of the incoming variable. This may be confirmed by noting that $E_{\mathbf{i}}$ consists of the variables which are tied with equal and maximal values of the first $\mathbf{i}+1$ components—c, **1**, **2**, . . . , $\mathbf{i}$—of **R**.

Since all $u_j \epsilon E_{\mathbf{i}}$ (for all $\mathbf{i}$) have $\mathbf{a}_{0,j} \geqslant 0$, condition (3) must also be satisfied.

PROOF OF THEOREM IV. Corollary IC implies that $R^{(k)}_J(\mathbf{i}+1) \geqslant R^{(k+1)}_J(\mathbf{i}+1)$ for all $\mathbf{i} < \mathbf{N}(k)$. To establish the conclusion required by the theorem it is only necessary to show that

$$R^{(k)}_J(\mathbf{i}+1) > R^{(k+1)}_J(\mathbf{i}+1) \tag{3.37}$$

when $\mathbf{i} = \mathbf{N}(k)$. If $\mathbf{a}^{(k)}_{0,J(k)} > 0$, (3.37) follows from the definition of $\mathbf{N}(k)$ and corollary ID. If $\mathbf{a}^{(k)}_{0,J} = 0$, then from the definition of $\mathbf{N}(k)$, $a^{(k)}_{\mathbf{i}+1,J} > 0$. In this situation row $\mathbf{i}+1$ will be selected as the source row, and we can conclude from theorem II that $a^{(k+1)}_{\mathbf{i}+1,J} < a^{(k)}_{\mathbf{i}+1,J}$ which from the definition of lexicographic priority given in part II,[13] implies (3.37).

PROOF OF THEOREM V. To prove theorem V we must show, among other things, that condition (1) of $B(k+1)$ holds regardless of the value of $\mathbf{a}^{(k+1)}_{0,J}$. Regarding condition (1) we shall show first that (I1) and (I3) are satisfied. Each set $E^{(k+1)}_{\mathbf{i}}$ must, definitionally contain some $u_j \epsilon\ E(+)^{(k+1)}_{\mathbf{i}-1}$ which is also an element of $E(+)^{(k+1)}_{\mathbf{i}}$. We may rule out $E(+)^{(k+1)}_0 = \emptyset$ since this hypothesis together with corollary IH implies[14] $E^{(k+1)}_0 \cap C+ = \emptyset$.

Thus (I1) must be satisfied. The Rules 3(J) and 3a(J) of part II imply that (I3) must be satisfied. Accordingly in the remainder of this proof we shall assume that condition (1) of $B(k+1)$ has been established once (I2) has been established.

For any $\mathbf{i} \leqslant \mathbf{N}(k)$, $B(k+1)$ is implied by $B(k)$ and theorem I if $E^{(k+1)}_{\mathbf{i}} = D^{(k+1)}(E^{(k)}_{\mathbf{i}})$. Let $\mathbf{i}'$ designate the smallest index in **R** for which $R_{J(k)}(\mathbf{i}') > R_{J(k+1)}(\mathbf{i}')$. Theorem IV implies $R_{J(k)}(\mathbf{i}) = R_{J(k+1)}(\mathbf{i})$ for all $\mathbf{i} < \mathbf{i}'$. The proof of theorem IV also implies that $\mathrm{i}' \leqslant \mathbf{N}(k)+1$. Corollary IF establishes that for $\mathbf{i} < \mathbf{i}'$, $E^{(k+1)}_{\mathbf{i}} = D^{(k+1)}(E^{(k)}_{\mathbf{i}})$.

Thus $B(k+1)$ is implied for $\mathbf{i} < \mathbf{i}'$.

Now consider the sets $E^{(k+1)}_{\mathbf{i}}$ with $\mathbf{i} \geqslant \mathbf{i}'$. If $\mathbf{a}^{(k)}_{0,J} = 0$ then condition (3) of $B(k)$ implies

$$\mathbf{a}^{(k+1)}_{0,j} \geqslant 0 \text{ for all } \mu_j \epsilon E^{(k+1)}_{\mathbf{i}'}. \tag{3.38}$$

If $\mathbf{a}^{(k)}_{0,J} > 0$, corollary IG implies (3.38).

Since all $E_{\mathbf{i}}$ with $\mathbf{i} > \mathbf{i}'$ are subsets of $E'_{\mathbf{i}}$, (3.38) also holds for every u_j in each such $E_{\mathbf{i}}$. This permits the use of (3.9) to establish (I2) for

$$F^{(k+1)} = E^{(k+1)}_{\mathbf{i}}$$

$$\text{row } d = \text{limit row } 0$$

$$\text{row } n = \text{row } \mathbf{i}+1$$

for all $\mathbf{i}$ such that $\mathbf{i}' \leqslant \mathbf{i} \leqslant \mathbf{N}(k+1)$ if $\mathbf{a}^{(k+1)}_{0,J} > 0$, and for $\mathbf{i}' \leqslant \mathbf{i} < \mathbf{N}(k+1)$ otherwise. If $\mathbf{a}^{(k+1)}_{0,J} = 0$, and $\mathbf{N}(k+1) \geqslant \mathbf{i}'$, condition (3) is satisfied by (3.38).

PROOF OF THEOREM VI. We shall derive a contradiction from the assumption that theorem VI is false, i.e., that for some k

$$R^{(k)}_J(c) = R^{(k+t)}_J(c) \text{ for every finite } t. \tag{3.39}$$

Theorem IV indicates that a statement such as (3.39) cannot be made for every index in **R**. Therefore there must be a smallest index $\mathbf{i}'$ for which (3.39) never holds—with $\mathbf{i}'$ substituted for c—for any value of k.

[12] See section 3.7.

[13] See section 2.6, rule 3a(J).

[14] See the proof of IH in appendix C.

Let k' signify the last cycle which accomplishes a decrease in $R_J(\mathbf{i}'-1)$. Corollary IIA implies that $\mathbf{a}_{0,J}^{(k'+t)}>0$ for some finite t. For notational convenience we shall assume $\mathbf{a}_{0,J}^{(k')}>0$.

Corollary IE and the hypothesis

$$R_J^{(k')}(\mathbf{i}'-1)=R_J^{(k'+t)}(\mathbf{i}'-1) \qquad (3.40)$$

imply that $u_{J(k'+t)}\epsilon D^{(k'+t)}(E_{\mathbf{i}'-1}^{(k')})$ for every finite t. In tableau k' the set $E_{\mathbf{i}'-1}^{(k')}$ can be represented on a diagram such as figure 3/7.

The entire set $E_{\mathbf{i}'-1}^{(k')}$ is (as a consequence of corollary IG and the definition of k') to the right of or on the vertical axis and on or below the line OA. The point $J(k')$ is somewhere on the line OA (to the right of the vertical axis). The selection of a positive slope for the line OA is arbitrary. The slope could be negative, but in any case must be finite.

The horizontal line through C and B represents a limit, equal to the distance from C to the origin, on the value of the basic x variable associated with row $\mathbf{i}'$. If the point on figure 3/7 associated with u_J falls below the line CB, then the limit row $\mathbf{i}'$ qualifies as an acceptable source row in $V(J)$.

Successive occurrence of the u_J point below the line CB and successive selection of $\mathbf{i}'$ as the source row will, (as a consequence of theorem II) eventually force the entire set $E_{\mathbf{i}'-1}$ above the line CB. The line AB represents a similar limit on the basic x variable associated with row 0. Similar comments apply to the use of the limit row 0 as a source row. The special source row selection rule applies whenever the point associated with u_J occurs to the right of or below point B. If the u_J point is outside the region OABC and above the extension of the line segment OB then the limit row 0 is the source row. If the u_J point is below the line segment OB (and outside OABC) then the limit row $\mathbf{i}'$ is the source row.

As t increases the set $D^{(k'+t)})E_{\mathbf{i}'+1}^{(k')}$ will be moved to positions that remain below lines such as (first) OA′

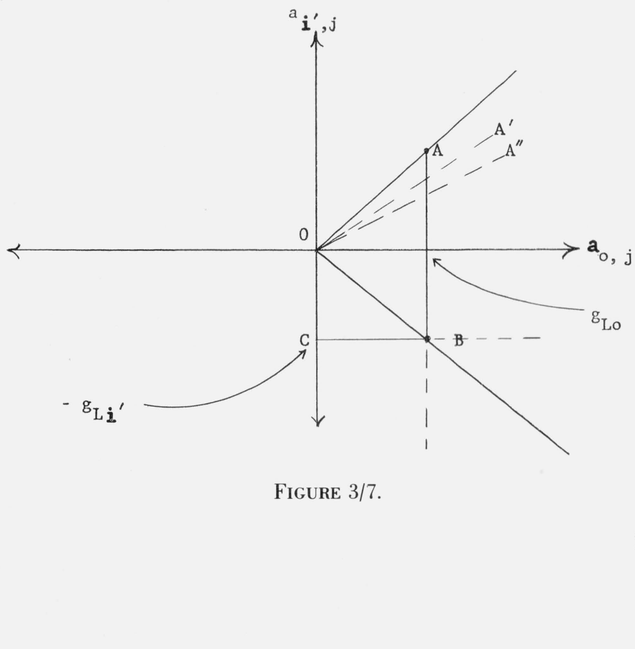


FIGURE 3/7.

and (later) OA″, which result from clockwise rotations of $R_J(\mathbf{i}')$ from the initial position of the line OA. We shall show that a finite number, t, of cycles is sufficient to rotate this line to a position coincident with the vertical axis—and thereby insure that no element of $D^{(k'+t)}(E_{\mathbf{i}'-1}^{(k')})$ is to the right of the vertical axis. This would (by realizing the assumed conditions of corollary IA for the index $\mathbf{i}'-1$) contradict the definition of $\mathbf{i}'$, and thereby prove theorem VI.

To prove that $D^{(k'+t)}(E_{\mathbf{i}'-1}^{(k')})$ will eventually be swept from the right-hand side of the vertical axis we shall temporarily adopt this hypothesis: the point $(\mathbf{a}_{0,J}^{(k'+t)}, a_{\mathbf{i}',J}^{(k'+t)})$ always occurs within the closed region OABC in figure 3/7 and never occurs at the origin. By taking t large enough we can accomplish an arbitrary number of decreases in the ratio $a_{\mathbf{i}',J}/\mathbf{a}_{0,J}$. Since the components of $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ must be integers and the point must be within the region OABC, only a finite number of different ratios can occur. Thus the ratio must, after a finite number of cycles, attain the following form: negative integer/zero.

Since it cannot be guaranteed that the point $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ will never occur at the origin or outside of the region OABC, we must relax that assumption.

Corollary IIA guarantees that subsequences of cycles during which $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ occurs at the origin are of finite duration. Such cycles alter no aspect of the set of points $(\mathbf{a}_{0,j}, a_{\mathbf{i}',j})$ for $u_j\epsilon E_{\mathbf{i}'-1}$ since these cycles only add a zero vector to each point in the set. Therefore the cycles which decrease the ratio $\mathbf{a}_{0,J}/a_{\mathbf{i}',J}$ must have $\mathbf{a}_{0,J}>0$. Thus, for the purposes of proving finiteness, it is permissible to ignore the possibility that the point $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ may occur at the origin, since such occurrences are finite in number and neutral in effect.

Suppose now that $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ occurs outside the region OABC—or more precisely, below and/or to the right of the point B. Then either row $\mathbf{i}'$ or the limit row 0 is made the source row, depending on whether $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ is below or above the line determined by the origin and point B. Theorems III, IV, and V, and corollary IIB combine to guarantee that a finite number of cycles will be sufficient to force the recurrence of $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ in the region OABC.

Let $k'+t'$ represent a cycle which decreases $a_{\mathbf{i}',J}/\mathbf{a}_{0,J}$. Let $\bar{t}'$ be the smallest value of t for which $\bar{t}'\geqslant t'$ and for which $(\mathbf{a}_{0,J}^{(k'+\bar{t}')}, a_{\mathbf{i}',J}^{(k'+\bar{t}')})$ is in the region OABC. Let t'' be the smallest value of t for which $t''\geqslant\bar{t}'$ and for which the cycle $k'+t''$ decreases $a_{\mathbf{i}',J}/\mathbf{a}_{0,J}$. Then we may define $\bar{t}''$ analogously to t', and define t''' analogously to t'', etc. Consider the sequence of tableaus $k'+\bar{t}'$, $k'+\bar{t}''$, $k'+\bar{t}'''$, etc. The ratio $a_{\mathbf{i}',J}/\mathbf{a}_{0,J}$ is reduced in every succeeding tableau in this sequence and the point $(\mathbf{a}_{0,J}, a_{\mathbf{i}',J})$ is in the area OABC for every tableau in the sequence. As we have already observed: this must lead to a tableau in which $a_{\mathbf{i}',J}/\mathbf{a}_{0,J}$ has the form: negative integer/zero.

PROOF OF THEOREM VII. To prove theorem VII we note that theorem VI provides the same guarantee that $c_J/\mathbf{a}_{0,J}$ will decrease in the finite number of cycles as was available in the proof of theorem VI for the index $\mathbf{i}'$ (and the ratio $a_{i',J}/\mathbf{a}_{0,J}$). The proof of theorem VII is a straightforward application of the same considerations that were used to show, in the proof of theorem VI, that a finite number of cycles is sufficient to drive the ratio $a_{i',J}/\mathbf{a}_{0,J}$ to negative infinity. The only difference, which is not substantial, is that here it is only necessary to show that a finite sequence of cycles will drive $c_J/\mathbf{a}_{0,J}$ to zero.

3.10. Proof That Every Subsequence of s Cycles Is Finite

In section 3/7 the connection was established between the solution (3.15), (3.16) to the dual problem (3.5), (3.6) and the ratio $c_J/\mathbf{a}_{0,J}$. Now we may apply theorem VII and corollary IH to establish the existence of a tableau in which a zero vector is a feasible solution to the dual problem, which establishes the sufficient condition (3.1) for another x cycle.

Appendix A. Discarding s Variables That Reenter the Basis

In this appendix we shall show that we may legitimately drop an s variable from the tableau after the variable has reentered the basis as the result of an s cycle. In this circumstance we also eliminate the row of the tableau associated with the (newly basic) s variable. This row is, of course, the row that served as pivot row in the s cycle that brought the s variable into the basis.

We shall show that in the circumstances we have described, nothing of significance to the operation of the primal algorithm is lost by eliminating the s variable and its associated row. To do this we shall show that the s variable could remain basic and nonnegative and its associated row need not be selected as the source row if the s variable were permitted to remain in the tableau. Thus the presence of the s variable in the tableau is not required to prevent violation of the Gomory cut in which the s variable has the role of a slack variable. And if the row associated with the s variable were never selected as the source row, the presence of this row would have no effect on the course of operation of the algorithm.

We shall assume that an s variable has entered the basis as the result of an s cycle. For definiteness and notational convenience we shall assume that this basic variable s_i is associated with row i. Now we shall consider a typical subsequent stationary cycle. This cycle might in general require the normal source row selection routine or a special source row selection routine. It is unnecessary to consider the case of a special source row selection routine in detail, since these routines always select a row associated with a basic x variable as the source row: therefore, row i could not be the source row in such a cycle.

Suppose the normal source row selection routine were employed. We note that s_i could only be driven negative and row i could only be the source row if $a_{i,J} > 0$. However, if $a_{i,J} > 0$, there must also exist a basic x variable associated with a row h such that $a_{h,J} > g_h \geqslant 0$. The nonexistence of such a basic x variable implies a contradiction: a transition cycle would be possible if row i were ignored[1]; and this would imply that an otherwise feasible integer solution is interdicted by the Gomory cut in which s_i is the slack variable. Thus a row $h \neq i$ exists which is in $V(J)$ and can be selected as the source row. This is sufficient to guarantee that row i need never be the source row.

Since all stationary cycles have a zero in the pivot (cut) row and the constant column, s_i cannot be driven negative by such a cycle. In a transition cycle we must have (by the argument in the preceding paragraph) $a_{i,J} < 0$, and therefore a transition cycle cannot drive s_i negative.

Appendix B. A Proof That Rule 3(J) and 3a(J) Will Always Lead to a Unique Selection of u_j

To show that no pair of variables u_j, u_{j^*}, ϵE_0 can have $\mathbf{R}_j = \mathbf{R}_{j^*}$, we shall utilize the notational foundation established in section 2.7 of part II.

First, we shall show that if u_j and u_{j^*} are both s variables then $\mathbf{R}_j = \mathbf{R}_{j^*}$ is impossible. To accomplish this we shall, as a preliminary, establish the following proposition:

If $a_{i',j} = Ka_{i',j^*}$, *for some number* K *and for every row index* i' *associated with a component of* $X_{N'}$ *then* $a_{i,j} = Ka_{i,j^*}$ *for every row* i *of the tableau.* (B.1)

To prove (B.1) we rely on the fact that every feasible solution to (2.24) is also a feasible solution to (2.1). Suppose the assumed conditions in (B.1) are satisfied, but $a_{i,j} \neq Ka_{i,j^*}$ for some row i corresponding to a component of X_B. This will lead to a contradiction. Consider the two feasible solutions below. (α is chosen sufficiently small to insure that both solutions are feasible.)

Solution 1	Solution 2
$u_j = \alpha \cdot K$ All other nonbasic variables in (2.24) are = zero	$u_{j^*} = \alpha$ All other nonbasic variables in (2.24) are = zero

All components of $X_{N'}$ take on identical values in solution 1 and solution 2. Regarding both of these solutions as solutions to (2.1) in which the values of the variables in X_B are functions of the variables in X_N (which con-

[1] We have assumed here, implicitly and for convenience, that s_i is the only s variable in the basis at the beginning of the cycle under discussion.

sists of the variables in $X_{N'}$ and $X_{N''}$), the values of all the variables in X_B must be equal in solution 1 and solution 2. This contradicts the above contrapositive hypothesis ($a_{i,j} \neq Ka_{i,j^*}$) whereby the value of the component of X_B associated with the row i would have a different value in the two solutions. Hence (B.1) is true.

Now to establish our proof we must consider three possible cases: (i) u_j and u_{j^*} are both s variables; (ii) u_j is an x variable and u_{j^*} is an s variable; (iii) u_j and u_{j^*} are both x variables.

In case (i) we assume that $\mathbf{R}_j = \mathbf{R}_{j^*}$ and show that this must lead to the conclusion that either u_j or u_{j^*} is a redundant variable. There are three cases: (ia) $\mathbf{a}_{0,j} = \mathbf{a}_{0,j^*} = 0$; (ib) $\mathbf{a}_{0,j} \neq 0$, and $\mathbf{a}_{0,j^*} = 0$; (ic) $\mathbf{a}_{0,j} \neq 0$, and $\mathbf{a}_{0,j^*} \neq 0$. In case (ia) we must have $a_{i',j} = a_{i',j^*}$ for every row i' associated with a component of $X_{N'}$. This satisfies the hypothesis of (B.1) with $K = 1$ and therefore leads to the conclusion that u_j and u_{j^*} are identical variables—one of which might be eliminated. In case (ib) every component of $\mathbf{R}_{j^*}$ has the form 0/0. This satisfies the hypothesis of (B.1) with $K = 0$, and implies that u_{j^*} is associated with a null vector. In case (ic) the hypothesis of (B.1) is again satisfied with $K = \mathbf{a}_{0,j}/\mathbf{a}_{0,j}$, and we may conclude that u_j is proportional to u_{j^*} and that one of these variables may be discarded.

More precisely, an all integer vector $\mathbf{A}_{j^{**}}$ with an associated variable $u_{j^{**}}$ must exist such that $\mathbf{A}_j$ and $\mathbf{A}_{j^*}$ can each be expressed as a positive integer multiple of $\mathbf{A}_{j^{**}}$. It is evident that no solution possibilities are lost if $u_{j^{**}}$ replaces u_j and u_{j^*}. (It is possible, of course, that $u_{j^{**}}$ is equal to u_j or u_{j^*}.) Since u_j and u_{j^*} are s variables, every feasible integer solution must determine integral values for these variables; and this rules out integer solutions in which $u_{j^{**}}$ has an integral value while the implicitly determined value of u_j or u_{j^*} is fractional. Therefore, substitution of $u_{j^{**}}$ for u_j and u_{j^*} introduces no new solutions. Thus Rules 3(J) and 3a(J) are capable of discriminating between any two nonredundant s variables.

Now we turn to cases (ii) and (iii). First we shall show that case (iii) is impossible. Consider the first cycle, $t+1$, following a transition cycle. $E_0^{(t)}$ contains a single x variable from $X+$. Assuming without loss of generality that the succeeding cycles are stationary cycles, the incoming variable $u_{J(t)}$ must be the x variable in $E_0^{(t)}$. This choice is clearly unambiguous by Rule 3(J). $S^{(t+1)}$ must consist of the single variable s_{t+1} which, having just been driven from the basis, cannot be an element of $C+$. Therefore $E^{(t+1)}$ consists of s_{t+1} and some x variable chosen from $X+$. The x variable is selected as the incoming variable on the basis of the first component of $\mathbf{R}$. Here again the rule makes a unique selection of u_J. $S^{(t+2)}$ will include s_{t+1} and s_{t+2}. $C+$ may or may not include s_{t+1}; hence some s cycles may occur before an x variable is brought into E_0 because $S \cap C+ = \emptyset$. During any such s cycles, E_0 consists entirely of s variables and therefore, as we have shown, rule 3(J) and rule 3a(J) must uniquely select the incoming variable. When another x variable is eventually brought into E_0, there are two possibilities: the x variable either is or is not immediately designated as u_J. If the x variable is selected as the incoming variable immediately we would return to the circumstances we have just discussed: all $u_j \epsilon E_0$ are s variables. Hence we need only discuss the case in which the x variable does not immediately become the incoming variable. Then $\mathbf{a}_{0,J}$ and c_J must both be equal to zero and therefore the simplex change of basis procedure will not change the value of $\mathbf{a}_{0,j}$ or c_j for any u_j in the transition to the succeeding tableau. The x variable remains the only element in E_0 and $C+$. This precludes another x variable entering E_0 until the single x variable in E_0 has become the incoming variable and has been thereafter deleted from E_0. Thus E_0 can contain at most one x variable.

In case (ii) u_j is an x variable in E_0. We have just shown that u_j must be x_0 and no other x variable can be in E_0. Since x_0 is not basic, we must have $\mathbf{a}_{0,j^*} = 0$, while $\mathbf{a}_{0,j} = 1$. Thus $\mathbf{R}_j = \mathbf{R}_{j^*}$ only if every component of $\mathbf{R}_{j^*}$ has the form 0/0; and this implies, because of (B.1), that u_{j^*} is a null vector.

Appendix C. Proof of Theorem I and Corollaries

1.1. Introduction

The purpose of this appendix is to provide an algebraic restatement and proof of theorem I and the corollaries that appear in part III. While the connections between the algebraic terminology used here and the geometric development in part III are not developed explicitly, reading this appendix in parallel with the analogous development in part III should reveal the relations, which are both simple and standard, between the algebra here and the geometry there.

1.2. Definitions

The analysis in this appendix is focused on some implications of the simplex change of basis procedure when certain conditions exist in the tableau that precedes the change of basis. Accordingly, our notational requirements will include (i) the usual algebraic representation for the tableaus, (ii) a convention to distinguish the (original and derived) data of the given tableau from the corresponding data of the tableau that results from the change of basis, and (iii) some special symbols to express the concepts in terms of which we state the assumed and implied conditions of the theorem and the corollaries.

1. The given tableau may be expressed by the matrix equation

$$IX_B + AX_N = G \geqslant 0 \qquad \text{(C.1)}$$

or equivalently by

$$x_i + \sum_{j=m+1}^{n'} a_{i,j}x_j = g_i > 0, \qquad i = 1, 2, \ldots, m. \qquad \text{(C.2)}$$

The current basic solution is $x_i = g_i$, $(i = 1, 2, \ldots, m)$, and $x_j = 0$, $(j = m+1, m+2, \ldots, n')$.

2. The two rows d and n are selected from the m rows given in (C.2). These two rows are selected arbitrarily except that neither row may serve as the pivot row in the change of basis procedure.

3. We shall designate the pivot row by the index p and the pivot column by the index J.

4. F is a set of nonbasic variables—i.e., a set whose members are also components of X_N.

5. $F(+) \equiv \{x_j | x_j \epsilon F \text{ and } a_{d,j} > 0\}$.

6. $$R^* = \max_{x_j \epsilon F(+)} [a_{n,j}/a_{d,j}]. \qquad \text{(C.3)}$$

7. For every $x_j \epsilon F$ we define

$$\Delta_j = R^* a_{d,j} - a_{n,j}. \qquad \text{(C.4)}$$

Thus

$$R^* a_{d,j} = a_{n,j} + \Delta_j. \qquad \text{(C.5)}$$

8. We shall use the symbol ˆ to identify data of the tableau that results from carrying out the change of basis operation on the system (C.1), (C.2). Thus, the coefficient in row i and column j of the new tableau will be signified by $\hat{a}_{i,j}$. The set which "descends" from F (according to the rules specified by section 3.6) will be signified by $\hat{F}$. In terms of this notation we shall symbolize the descendant relationship of $\hat{F}$ to F by $\hat{F} = \hat{D}(F)$.

1.3. General Analysis and Proof of Theorem I

In this section we shall state theorem I in terms of the notation we have established here and prove the theorem. The analysis on which this proof is based will also serve as the basis for the proofs of the corollaries in the next section.

THEOREM I. *The hypothesis is*

$$\mathrm{F}(+) \neq \emptyset, \qquad \text{(C.6)}$$

$$\Delta_j \geqslant 0, \textit{ for all } x_j \epsilon \mathrm{F}; \qquad \text{(C.7)}$$

$$\Delta_J = 0; \textit{ and} \qquad \text{(C.8)}$$

$$a_{d,J} \geqslant 0. \qquad \text{(C.9)}$$

The conclusion is

$$\hat{\mathrm{F}}(+) \neq \emptyset \Rightarrow \hat{\Delta}_j \geqslant 0 \textit{ for all } x_j \epsilon \hat{\mathrm{F}}. \qquad \text{(C.10)}$$

General analysis and proof. We shall let j be the index of an arbitrarily selected element of F and show that (C.10) holds for x_j. The following formulas describe the effect of the change of basis procedure on the data of column j in rows n and d.

$$\hat{a}_{n\,j} = a_{n,j} - a_{p,j} a_{n\,J}; \qquad \text{(C.11)}$$

$$\hat{a}_{d,j} = a_{d,j} - a_{p,j} a_{d,J}. \qquad \text{(C.12)}$$

To simplify the above expressions we have assumed that $a_{p,J} = 1$. While the cycles of the primal algorithm always satisfy this assumption, it is not necessary to the proof of theorem I or the corollaries.

Since the existence of R^* is guaranteed by (C.6), we may multiply both sides of (C.12) by R^*. The result is

$$R^* \hat{a}_{d,j} = R^* a_{d,j} - R^* a_{p,j} a_{d,J}. \qquad \text{(C.13)}$$

Now we may use (C.5) as a basis for substitution into the right side of (C.13). This leads to

$$R^* \hat{a}_{d,j} = a_{n,j} + \Delta_j - a_{p,j} a_{n,J} \qquad \text{(C.14)}$$

which has been simplified by substitution from (C.8). We can employ (C.11) as a basis for substitution into the right side of (C.14) to obtain

$$R^* \hat{a}_{d,j} = \hat{a}_{n,j} + \Delta_j, \qquad \text{(C.15)}$$

which is, incidentally, an algebraic expression of the rule of parallel movement stated in part III.

We shall prove theorem I for two mutually exclusive and collectively exhaustive cases: (i) $\hat{a}_{d,j} = 0$, and (ii) $\hat{a}_{d,j} \neq 0$. Case (i) can be proved on the basis of (C.15). If $\hat{a}_{d,j} = 0$, then the right side of (C.15) must equal zero. Since (C.7) requires that Δ_j be nonnegative, $\hat{a}_{n,j}$ must be nonpositive. By analogy to (C.4) we have

$$\hat{\Delta}_j = \hat{R}^* \hat{a}_{d,j} - \hat{a}_{n,j}. \qquad \text{(C.16)}$$

Now if $\hat{F}(+) \neq \emptyset$, then $\hat{R}^*$ exists and is finite. Therefore, $\hat{\Delta}_j = -\hat{a}_{n,j} \geqslant 0$.

Before proceeding to a proof for case (ii) we shall develop algebraically some further implications of our assumptions. If $\hat{a}_{d,j} \neq 0$ we may divide (C.15) by $\hat{a}_{d,j}$, which results in

$$R^* = \hat{a}_{n,j}/\hat{a}_{d,j} + \Delta_j/\hat{a}_{d,j}, \text{ or} \qquad \text{(C.17)}$$

$$R^* - \Delta_j/\hat{a}_{d,j} = \hat{a}_{n,j}/\hat{a}_{d,j}. \qquad \text{(C.18)}$$

We shall use (C.18) to relate R^* to $\hat{R}^*$. By analogy to (C.3) the definition of $\hat{R}^*$ is

$$\hat{R}^* = \max_{x_j \epsilon \hat{F}(+)} [\hat{a}_{n,j}/\hat{a}_{d,j}] = \max_{x_j \epsilon \hat{F}(+)} [R^* - \Delta_j/\hat{a}_{d,j}]. \qquad \text{(C.19)}$$

We shall use the index $\hat{J}^*$ to designate a variable in $\hat{F}(+)$ which has $\hat{a}_{n,\hat{J}^*}/\hat{a}_{d,\hat{J}^*} = \hat{R}^*$. The hypothesis of (C.10) establishes the existence of $x_{\hat{J}^*}$. From (C.19) it must be that for all $x_j \epsilon \hat{F}(+)$,

$$\Delta_{\hat{J}^*}/\hat{a}_{d,\hat{J}^*} \leqslant \Delta_j/\hat{a}_{d,j}. \qquad \text{(C.20)}$$

We can use (C.18) and the definition of $\hat{J}^*$ as a basis for substitution into (C.19) to secure the following relation between R^* and $\hat{R}^*$:

$$\hat{R}^* = R^* - \Delta_{\hat{J}^*}/\hat{a}_{d,\hat{J}^*}. \qquad \text{(C.21)}$$

Now we are ready to undertake an algebraic revision of (C.16) which will provide the basis for the proof. First $a_{d,j}$ is factored out of the right side, yielding,

$$\hat{\Delta}_j = \hat{a}_{d,j}[\hat{R}^* - \hat{a}_{n,j}/\hat{a}_{d,j}]. \tag{C.22}$$

Next we use (C.18) as a basis for substitution into (C.22). The result is

$$\hat{\Delta}_j = \hat{a}_{d,j}[\hat{R}^* - R^* + \Delta_j/\hat{a}_{d,j}]. \tag{C.23}$$

Finally we substitute on the basis of (C.21) into (C.23) to obtain

$$\hat{\Delta}_j = \hat{a}_{d,j}[\Delta_j/\hat{a}_{d,j} - \Delta_{\hat{J}^*}/\hat{a}_{d,\hat{J}^*}]. \tag{C.24}$$

To prove theorem I for case (ii) we shall consider two subcases: (iia) $\hat{a}_{d,j} > 0$, and (iib) $\hat{a}_{d,j} < 0$. In case (iia), (C.24) and (C.20) imply $\hat{\Delta}_j \geqslant 0$. In case (iib) both of the fractions in the brackets on the right side of (C.24) must be nonpositive. This implies $\hat{\Delta}_j \geqslant 0$.

1.4. Proofs for Selected Corollaries to Theorem I

In this section we shall present proofs for most of the corollaries to theorem I which are given in chapter III. We omit proofs of IE and IJ. The proof of IE is omitted since this corollary differs trivially from IF. Corollary IJ follows by induction from theorem I. The assumed conditions of theorem I are implicitly included with the assumed conditions of each of the following corollaries.

COROLLARY IA

$$[\Delta_j = 0 \Rightarrow \hat{a}_{d,j} \leqslant 0] \Rightarrow R^* > \hat{R}^*. \tag{C.25}$$

PROOF. The assumed condition implies $\hat{a}_{d,j} > 0$ $\Rightarrow \Delta_j \neq 0 \Rightarrow \Delta_j > 0$. Therefore, since by definition $\hat{a}_{d,\hat{J}^*} > 0$, we must have $\Delta_{\hat{J}^*} > 0$. These facts, in conjunction with (C.21) imply $R^* > \hat{R}^*$.

COROLLARY IB *There exists an* x_j *such that* $\Delta_j = 0$ *and* $\hat{a}_{d,j} > 0$ *if and only if* $R^* = \hat{R}^*$. (C.26)

PROOF. If $R^* = \hat{R}^*$, then from (C.21), $\Delta_{\hat{J}^*}/\hat{a}_{d,\hat{J}^*} = 0$. Since by definition $\hat{a}_{d,\hat{J}^*} > 0$, the "if" part of the corollary is satisfied by $x_j = x_{\hat{J}^*}$.

Now if we assume x_j exists, we can conclude from (C.20) that $\Delta_{\hat{J}^*} = 0$. Then (C.21) implies $R^* = \hat{R}^*$.

COROLLARY IC

$$R^* \geqslant \hat{R}^*.$$

PROOF. A proof follows directly from (C.21) since $\Delta_{\hat{J}^*} \geqslant 0$ and $\hat{a}_{d,\hat{J}^*} > 0$.

COROLLARY ID

$$\Delta_j > 0 \textit{ for all } j \neq J \Rightarrow R^* > \hat{R}^*. \tag{C.27}$$

PROOF. Since $x_J \notin \hat{F}$, clearly $x_{\hat{J}^*} \neq x_J$. Moreover, $x_{\hat{J}^*} \neq x_p$ (where x_p is the newly nonbasic variable that "replaces" x_J in $\hat{F}$), since $-a_{d,J} = \hat{a}_{d,p} \leqslant 0$, and $\hat{a}_{d,\hat{J}^*} > 0$. Therefore $\Delta_{\hat{J}^*} > 0$, which together with (C.21) implies $R^* > \hat{R}^*$.

COROLLARY IF *Before stating this corollary we shall define the sets* E_n *and* $\hat{E}_n$.

$$E_n \equiv \{x_j | x_j \epsilon F \textit{ and } \Delta_j = 0\}$$

$$\hat{E}_n \equiv \{x_j | x_j \epsilon \hat{F} \textit{ and } \hat{\Delta}_j = 0\}.$$

The corollary is

$$R^* = \hat{R}^* \Rightarrow \hat{E}_n = D\hat{\ }(E_n). \tag{C.28}$$

PROOF. We shall first consider any $x_j \epsilon E_n$ which is also in $\hat{F}$. Here there are two cases: (i) $\hat{a}_{d,j} = 0$, and (ii) $\hat{a}_{d,j} \neq 0$. In case (i), $\Delta_j = 0$ and (C.15) imply $\hat{a}_{n,j} = 0$. This permits the conclusion, from (C.16), that $\hat{\Delta}_j = 0$. In case (ii) we note that the term in brackets on the right side of (C.23) vanishes. Hence $\hat{\Delta}_j = 0$.

It remains to consider x_p, the newly nonbasic variable which replaces x_J in $\hat{F}$. Since $a_{n,p} = 0 = a_{d,p}$, we may use (C.4) to infer that $\Delta_p = 0$. Depending on whether $\hat{a}_{d,p}$ is or is not equal to zero we can apply an analysis identical to case (i) or case (ii) above to conclude in either event that $\hat{\Delta}_p = 0$.

We have shown that if $R^* = \hat{R}^*$ then $x_j \epsilon D\hat{\ }(E_n) \Rightarrow x_j \epsilon E_n$. Now we shall show that if $R^* = \hat{R}^*$, $x_j \epsilon \hat{E}_n \Rightarrow x_j \epsilon D\hat{\ }(E_n)$. This is true if $\hat{\Delta}_j = 0 \Rightarrow \Delta_j = 0$. Again we consider two cases (i) $\hat{a}_{d,j} = 0$ and (ii) $\hat{a}_{d,j} \neq 0$. In case (i) (C.16) and $\hat{\Delta}_j = 0$ imply $\hat{a}_{n,j} = 0$, which, with (C.15) implies $\Delta_j = 0$. In case (ii) $\hat{\Delta}_j = 0$ and (C.23) imply $\Delta_j = 0$.

COROLLARY IG

$$R^* > \hat{R}^* \Rightarrow [\hat{\Delta}_j = 0 \Rightarrow \hat{a}_{d,j} \geqslant 0] \tag{C.29}$$

PROOF. By rearrangement of the terms in (C.16) we obtain

$$\hat{R}^* \hat{a}_{d,j} = \hat{a}_{n,j} + \hat{\Delta}_j. \tag{C.30}$$

If (C.30) is subtracted from (C.15) the result is

$$\hat{a}_{d,j}(R^* - \hat{R}^*) = \Delta_j - \hat{\Delta}_j \tag{C.31}$$

Rearrangement of (C.31) gives

$$\hat{a}_{d,j} = \frac{\Delta_j - \hat{\Delta}_j}{R^* - \hat{R}^*} \tag{C.32}$$

By hypothesis the denominator on the right side of (C.32) cannot be negative. Therefore, $\Delta_j = 0$ implies $a_{d,j} \geqslant 0$.

COROLLARY IH

$$\left.\begin{array}{l} R^* \geqslant 0 \\ \hat{R}^* < 0 \\ \hat{a}_{d,j} \leqslant 0 \end{array}\right\} \Rightarrow \hat{a}_{n,j} \leqslant 0.$$

PROOF. This follows directly from (C.15) where by hypothesis the left side must be nonpositive. Hence $\hat{a}_{n,j} \leqslant 0$, since $\Delta_j \geqslant 0$.

We observe that the proof is independent of the assumption $\hat{R}^* < 0$. This assumption is included for a psychological rather than a logical purpose: to emphasize the application of the corollary to the tableau in which $\hat{R}^*$ first "goes negative."

References

[1] Ben-Israel, A., and A. Charnes, On some problems of diophantine programming, Cahiers, pp. 215–280 (1962).

[2] Charnes, A., Optimality and Degeneracy in Linear Programming, Econometrica **20,** No. 2, 160–170 (April 1952).

[3] Charnes, A., and W. W. Cooper, Management Models and Industrial Applications of Linear Programming (Wiley & Sons, New York and London, 1961).

[4] Dantzig, George B., On the significance of solving linear programming problems with some integer variables, Econometrica **28,** No. 1, 30–44 (January, 1960).

[5] Dantzig, George B., Linear Programming and Extensions, (Princeton University Press, Princeton, N.J., 1963).

[6] Dantzig, G. B., L. R. Ford, and D. R. Fulkerson, A Primal-Dual Algorithm for Linear Programs, in H. W. Kuhn and A. W. Tucker (eds.), Linear Inequalities and Related Systems, Annals of Mathematics Studies, No. 38, 171–181 (Princeton University Press, Princeton, N.J., 1956).

[7] Glover, F., An all-integer integer programming algorithm, unpublished working paper, Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pa., December, 1963.

[8] Glover, F., Generalized cuts in diophantine programming, Ph. D. Thesis, Carnegie Institute of Technology, 1965.

[9] Gomory, R. E., Essentials of an algorithm for integer solutions to linear programs, Bulletin of the American Mathematical Society **64,** No. 5 (1958).

[10] Gomory, Ralph E., An Algorithm for Integer Solutions to Linear Programs, in Robert L. Graves and Philip Wolfe (eds.), Recent Advances in Mathematical Programming, pp. 269–302 (McGraw-Hill Book Co., New York, N.Y., 1963).

[11] Gomory, Ralph E., An All-Integer Integer Programming Algorithm, in John F. Muth and Gerald L. Thompson (eds.), Industrial Scheduling pp. 193–206 (Prentice-Hall, Englewood Cliffs, New Jersey, 1963).

[12] Gomory, R. E., On the relation between integer and noninteger solutions to linear programs, Proc. Natl. Acad. Sci. **53,** 260–265 (1965).

[13] Gomory, Ralph E., and William J. Baumol, Integer programming and pricing, Econometrica **28,** No. 3, 521–550 (July, 1960).

[14] Harris, P. M. J., An algorithm for solving mixed integer linear programmes, Operational Research Quarterly, **15,** No. 2, 117–132 (June 1964).

[15] Hoffman, A. J., and J. G. Kruskal, Integral Boundary Points of Convex Polyhedra, in H. W. Kuhn and A. W. Tucker (eds.), Linear Inequalities and Related Systems, Annals of Mathematics Studies, No. 38, 223–246 (Princeton University Press, Princeton, New Jersey, 1956).

[16] Land, A. H., and A. G. Doig, "An automatic method of solving discrete programming problems, Econometrica **28,** No. 3, 497–520 (July, 1960).

[17] Markowitz, H. M., and A. S. Manne, On the solution of discrete programming problems, Econometrica **25,** No. 1, 84–110 (Jan., 1957).

[18] Szwarc, Wlodzimierz, The mixed integer programming problem when the variables are zero or one, unpublished working paper, Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pennsylvania, May, 1963.

[19] Thompson, G. L., The stopped simplex method: I. Basic theory for mixed integer programming; integer programming, pp. 159–182, Revue Francaise De Recherche Operationnelle, (1964).

[20] Weingartner, H. Martin, Mathematical Programming and the Analysis of Capital Budgeting Problems (Prentice-Hall, Englewood Cliffs, New Jersey, 1963).

[21] Young, R. D., Primal integer programming: example problems and alternative algorithms, in preparation.

(Paper 69B3–154)