MATHEMATICS OF OPERATIONS RESEARCH
Vol. 2, No. 2, May 1977
Printed in U.S.A.

NEW FINITE PIVOTING RULES FOR THE SIMPLEX METHOD*†

ROBERT G. BLAND

SUNY—Binghamton

A simple proof of finiteness is given for the simplex method under an easily described pivoting rule. A second new finite version of the simplex method is also presented.

1. A simple finite pivoting rule. Consider the canonical linear programming problem

$$\begin{aligned} &\text{maximize} && x_0, \\ &\text{subject to} && Ax = b, \\ & && x_j \geqslant 0 \quad \forall j \in E = \{1, \ldots, n\}, \end{aligned} \tag{1.1}$$

where A has $m + 1$ rows and $n + 1$ columns and is of full row rank. We denote the canonical simplex tableau for (1.1) corresponding to some basic set of variables with index set $B = \{B_0 = 0, B_1, \ldots, B_m\}$ by $(\bar{A}, \bar{b})$. It is assumed that the rows of $(\bar{A}, \bar{b})$ are ordered so that $\bar{a}_{i, B_i} = 1$; thus the ith row of the tableau represents the equation $x_{B_i} + \sum_{j \notin B} \bar{a}_{ij} x_j = \bar{b}_i$. If $\bar{b}_i \geqslant 0$ for $i = 1, \ldots, m$, then the tableau is (primal) feasible and the simplex pivoting rule permits the selection of any (nonbasic) variable x_k having $\bar{a}_{0k} < 0$ to enter the basis. If $\bar{a}_{0j} \geqslant 0$ for all $j \in E$, then the pivoting stops with the current tableau optimal. Having chosen a variable x_k to enter the basis, the simplex rule permits the selection of any basic variable x_{B_r} having $\bar{a}_{rk} > 0$ and

$$\frac{\bar{b}_r}{\bar{a}_{rk}} = \min\left\{ \frac{\bar{b}_i}{\bar{a}_{ik}} \ : \bar{a}_{ik} > 0 \right\}$$

to leave the basis. If $\bar{a}_{ik} \leqslant 0$ for $i = 1, \ldots, m$, then the pivoting stops with the current tableau indicating primal unboundedness and dual infeasibility.

A pivoting rule that is consistent with the simplex rule and further restricts the choice of either the pivot column or the pivot row is called a *refinement* of the simplex rule. We say that a refinement determines *a simplex* method, as opposed to *the simplex method*, which is used here as a generic term referring to the family of methods determined by all possible refinements.

It is very well known that the simplex method can fail to be finite because of the possibility of cycling. Certain refinements of the simplex pivoting rule, such as the lexicographic rule described in [3], restrict the selection of the pivot row in such a way that cycling cannot occur. The following refinement, which restricts the choice of both the pivot column and the pivot row, determines a simplex method that is, among all finite simplex methods known to us, the easiest to state, the easiest to implement, and the easiest to prove finite.

Let Rule I be the refinement of the simplex pivoting rule obtained by imposing the following restriction:

among all candidates to enter the basis, select the variable x_k having the lowest

* Received July 26, 1976; revised February 8, 1977.

AMS 1970 subject classification. Primary 90C05.

IAOR 1973 subject classification. Main: Programming: Linear.

Key words: Linear programming, simplex method, cycling, degeneracy.

† This research was performed under a research fellowship at CORE, Heverlee, Belgium.

index, i.e., pivot in the column k determined by

$$k = \min\{ j : \bar{a}_{0j} < 0\}; \tag{1.2(a)}$$

among all candidates to leave the basis, select the variable x_{B_r} having the lowest index, i.e., pivot in the row r determined by

$$B_r = \min\left\{ B_l : \bar{a}_{lk} > 0 \quad \text{and} \quad \frac{\bar{b}_l}{\bar{a}_{lk}} = \min\left\{ \frac{\bar{b}_i}{\bar{a}_{ik}} : \bar{a}_{ik} > 0 \right\}\right\}. \tag{1.2(b)}$$

THEOREM 1.1. *The simplex method under* Rule I *cannot cycle, hence it is finite.*

PROOF. Suppose to the contrary that for some linear programming problem P of form (1.1) and some initial feasible tableau, cycling occurs. (Note that given any feasible tableau, either optimality is verified, primal unboundedness is detected or Rule I uniquely determines a pivot element. Hence if cycling occurs, the cycle is unique.) Let $T \subseteq E$ be the index set of all variables that enter the basis during the cycle (so that $j \notin T$ implies that either x_j is never a basic variable during the cycle or x_j is always a basic variable during the cycle). Let $q = \max\{ j : j \in T\}$ and let $(\bar{A}', \bar{b}')$ be a tableau in the cycle such that Rule I specifies column q of $(\bar{A}', \bar{b}')$ as the pivot column. Let $Y = (y_0, \ldots, y_n)$ be defined by $y_j = \bar{a}'_{0j}$ for $j = 0, \ldots, n$. Then

$$y_0 = 1, \quad y_q < 0, \quad y_j \geqslant 0 \quad \forall j < q, \tag{1.3}$$

and Y is in the subspace of R^{n+1} generated by the rows of A.

Since x_q enters the basis during the cycle, x_q must also leave the basis during the cycle. Let $(\bar{A}'', \bar{b}'')$ be a tableau in the cycle corresponding to a set of basic variables $\{x_0 = x_{B_0}, x_{B_1}, \ldots, x_{B_r} = x_q, \ldots, x_{B_m}\}$ such that Rule I specifies a pivot in row r and, say, column t of $(\bar{A}'', \bar{b}'')$. Let $Z = (z_0, \ldots, z_n)$ with $z_{B_i} = \bar{a}''_{it}$ for $i = 0, 1, \ldots, m$, $z_t = -1$, and $z_j = 0$ otherwise, so that $z_0 = \bar{a}''_{0t} < 0$ and $z_q = a''_{rt} > 0$. Note that Z is in the orthogonal complement of the row space of A, which implies that $Y \cdot Z = 0$. Since $y_0 z_0 < 0$, it must be that $y_j z_j > 0$ for some j, $1 \leqslant j \leqslant n$. But $y_j \neq 0$ implies that x_j is a nonbasic variable in tableau $(\bar{A}', \bar{b}')$, and $z_j \neq 0$ implies that either x_j is a basic variable in $(\bar{A}'', \bar{b}'')$ or $j = t$. Hence $j \in T$, which implies that $j \leqslant q$. But $y_q < 0$ and $z_q > 0$, so $j < q$. It then follows from (1.3) that $y_j > 0$, which implies that $z_j > 0$. But $z_t = -1$, so $j \neq t$. Thus x_j is a basic variable in $(\bar{A}'', \bar{b}'')$; let $j = B_p$ so $\bar{a}''_{pt} = z_j > 0$.

Each pivot in the cycle must be degenerate, i.e., all variables remain fixed in value throughout the cycle. In particular, since $j \in T$ it must be that $x_j = 0$ during the cycle, implying that $\bar{b}''_p = 0$. However, we have now established that $j = B_p < q$, $a''_{pt} > 0$ and $\bar{b}''_p = 0$. This yields a contradiction since (1.2(b)) then precludes the possibility of pivoting x_q out of the basis in $(\bar{A}'', \bar{b}'')$. Hence cycling cannot occur, so monotonicity of the objective function value implies that the algorithm terminates after finitely many pivots. ∎

There are other simple proofs of Theorem 1.1. One could argue, for example, that subject to Rule I there can be at most one simplex pivot in column n. It then follows that there can be at most 2^j pivots in column $n - j$ for $j = 0, \ldots, n-1$.

It should be noted that if (1.2(a)) is dropped, so that only the selection of the pivot row is restricted, then cycling can occur (see the examples of Hoffman and Beale [3, pp. 229–230]).

2. A second finite simplex method.

The properties that render the simplex method finite under Rule I can be invoked to construct other finite versions of the simplex method. In this section we sketch a second finite simplex method.

Consider again the canonical linear programming problem P of form (1.1) with $m+1$ rows and $n+1$ columns. Suppose that $(\bar{A}, \bar{b})$ is a feasible tableau for P corresponding to some basis with index set $B = \{B_0 = 0, B_1, \ldots, B_m\}$. If for some $S \subseteq E$ and $k \in E \setminus S$ we have $S \cap B = \emptyset$, $\bar{a}_{0k} < 0$ and $\bar{a}_{0j} \geqslant 0$ for all $j \in E \setminus (S \cup \{k\})$, then we say that $(\bar{A}, \bar{b})$ is *reducible with respect to* S.

Observation 2.1. If $(\bar{A}, \bar{b})$ is reducible with respect to S, then for any $\hat{x} = (\hat{x}_0, \ldots, \hat{x}_n)$ satisfying $A\hat{x} = b$, $\hat{x}_j = 0\ \forall j \in S$, $\hat{x}_j \geqslant 0\ \forall j \in E \setminus (S \cup \{k\})$, and $\hat{x}_0 \geqslant \bar{b}_0$, it follows that $\hat{x}_k \geqslant 0$.

Suppose that $(\bar{A}, \bar{b})$ is reducible with respect to S, and we wish to solve the linear programming subproblem P' obtained from P by deleting S, i.e., by setting $x_j = 0$ for all $j \in S$. The tableau $(\bar{A}, \bar{b})$ represents a feasible solution of P'. By pivoting in column k of $(\bar{A}, \bar{b})$ we either solve P', or we produce a new feasible tableau $(\bar{A}', \bar{b}')$ in which x_k is a basic variable and $\bar{a}'_{0j} < 0$ for some $j \in E \setminus S$. In the latter case, we see from Observation 2.1 that we can ignore x_k as a candidate to leave the basis during all subsequent simplex pivots until P' is solved. Thus, the row of $(\bar{A}', \bar{b}')$ corresponding to the basic variable x_k is superfluous; we then say that the number of *active* constraints has been *reduced* to m. We can, in fact, delete x_k and its associated row from $(\bar{A}', \bar{b}')$, solve the remaining $m \times n$ *reduced problem*, which is equivalent to P', and restore the x_k-row when an optimal tableau for the reduced problem is at hand.

We will now show how to use these ideas to construct another finite simplex method. First consider a linear programming problem Q of the form

$$\begin{aligned} &\text{maximize} && x_0, \\ &\text{subject to} && Ax = b, \\ &&& x_j \geqslant 0 \quad \forall j \in E \setminus S, \\ &&& x_j = 0 \quad \forall j \in S, \end{aligned} \tag{2.1}$$

where $S \subseteq E$.

Suppose that $(\bar{A}^0, \bar{b}^0)$ is a feasible tableau for Q corresponding to the basic feasible solution $x^0 = (x_0^0, \ldots, x_n^0)$ and having $B^0 \cap S = \emptyset$, where B^0 is the index set of the basic variables. Consider the following procedure for solving Q.

Procedure A. (0) Initially let $i = 0$.

(1) Let $D^i = \{j \in E \setminus S : \bar{a}_{0j}^i < 0\}$. If $D^i = \emptyset$, then x^i is an optimal solution of Q. Otherwise select some $k \in D^i$ and let $S^{i+1} = S \cup D^i \setminus \{k\}$.

(2) Solve the linear programming subproblem Q^{i+1} obtained from Q by replacing S by S^{i+1} in (2.1). Since $B^i \cap S^{i+1} = \emptyset$, x^i is a basic feasible solution of Q^{i+1}. If Q^{i+1} is unbounded, then Q is unbounded. Otherwise let $(\bar{A}^{i+1}, \bar{b}^{i+1})$ be an optimal tableau for Q^{i+1} corresponding to some basis B^{i+1} having $B^{i+1} \cap S = \emptyset$ and let x^{i+1} denote the solution represented by $(\bar{A}^{i+1}, \bar{b}^{i+1})$. Increase i by 1 and go to (1).

Observation 2.2. Since $S^{i+1} \subsetneqq S^i$, for $i \geqslant 1$, Procedure A solves Q after solving only finitely many subproblems, say $Q^1, \ldots, Q^l$.

Observation 2.3. The tableau $(\bar{A}^i, \bar{b}^i)$ is reducible with respect to S^{i+1}, $i = 0, \ldots, l-1$. Hence in one simplex pivot we can either solve Q^{i+1} or reduce Q^{i+1} to an equivalent linear programming problem with one fewer constraint.

Given any linear programming problem P in form (1.1), we can apply Procedure A recursively to solve P starting from any feasible tableau for P. Initially we let $Q = P$ in Procedure A with $S = \emptyset$; and thus we create a sequence $P^1, \ldots, P^l$ of subproblems, each of form (2.1). Let the subset S for subproblem P^i be denoted by S^i. When subproblem P^i is created, we have a feasible tableau for P^i that is reducible with respect to S^i. A single pivot either solves P^i or reduces it to an equivalent $m \times n$ subproblem $\hat{P}^i$. In the latter case, we let $Q = \hat{P}^i$ and continue the process. Observations 2.2 and 2.3 can be applied (recursively) to show that such a recursive application of Procedure A will solve any linear programming problem P after finitely many pivots. (The recursion may effect successive reductions so that the number of rows in the reduced tableaux varies between 1 and m.)

Note that the recursive application of Procedure A as described above is a simplex method, in spite of the fact that some of the pivots are performed in reduced tableaux. Observation 2.1 implies that the same sequence of pivots in the full tableau conforms to the simplex rule. Let Rule II refer to the refinement of the general simplex pivoting rule that is implicit in the recursive application of Procedure A described above. The reader will note that in contrast with Rule I, Rule II does not uniquely determine the pivot element; there may be some freedom in the selection of both the pivot column and the pivot row.

3. Concluding remarks. That the simplex method is finite under Rules I and II is of some conceptual or pedagogical interest, but finiteness, by itself, is not particularly interesting from a computational standpoint. However, Rules I and II do have some interesting computational properties. (For example, in any problem requiring a "large" number of pivots under Rule I, the pivots will "concentrate" in the lower-indexed columns. Similarly, in any problem requiring a "large" number of pivots under Rule II, the pivots will "concentrate" in reduced tableaux having a "small" number of rows relative to the original problem.) We will not pursue the computational properties of Rules I and II here; we intend to explore that subject (and make precise the two roughly stated observations given above) separately.

It is noticeable that Rules I and II ignore the magnitudes of the tableau entries $\bar{a}_{0j}$ in the selection of a pivot column. This is a reflection of the broader context in which these rules arose: a combinatorial abstraction of linear programming in which only the signs of the tableau entries retain significance. We will conclude by briefly relating how these refinements of the simplex pivoting rule arose in that context.

Most interesting theorems concerning linear programming can be phrased as sign properties of the vectors in complementary orthogonal subspaces of R^n. Rockafellar suggested in [6] that such results ought to generalize in an appropriately axiomatized system of *oriented matroids*. Several equivalent axiomatizations of oriented matroids have since been given by Bland and Las Vergnas [1], [2], [4] and in the thesis of Lawrence [5], where previously unpublished work on another equivalent axiomatization by the late Jon Folkman is presented and extended. All of the results regarded by Rockafellar as susceptible to abstraction do indeed generalize in the context of oriented matroids. While we were able to find a nonconstructive proof of the generalization of the "complementarity" form of the linear programming duality theorem (primal and dual feasibility imply the existence of a complementary pair of feasible solutions), we had hoped to establish that result by a constructive, simplex-like approach. (We have recently learned that Lawrence [5] had already proved this theorem, but his proof is also nonconstructive.) This constructive approach would require a purely combinatorial proof, of finiteness of the simplex method. (Most proofs of finiteness, including those presented above, invoke monotonicity of the objective function value. This property cannot, as far as we know, be nicely translated into the matriod context.) We have now succeeded in constructively proving the "complementarity" form of the duality theorem for oriented matroids (and the stronger "schema" form) by a pivoting method that specializes (when the oriented matroid comes from a real vector space) to the simplex method under Rule II. The more general matroid results, including the purely combinatorial proof of finiteness of Rule II, will appear in a separate paper.

References

[1] Bland, R. G. (1974). Complementary Orthogonal Subspaces of R^n and Orientability of Matroids. Dissertation, Cornell University.

[2] Bland, R. G. and Las Vergnas, M. (to appear). Orientability of Matroids. *J. Combinatorial Theory(B).*

[3] Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton University, Princeton.

[4] Las Vergnas, M. (1974). Matroides Orientables, (preprint). Announced in *C. R. Acad. Sci. Paris*, t.280.

[5] Lawrence, J. (1975). Oriented Matroids. Dissertation, University of Washington.

[6] Rockafellar, R. T. (1969). The Elementary Vectors of a Subspace of R^n. In R. C. Bose and T. A. Dowling, eds., *Combinatorial Mathematics and Its Applications*. University of North Carolina, Chapel Hill, pp. 104–127.

CORE, DE CROYLAAN 54, B-3030, HEVERLEE, BELGIUM