U. S. AIR FORCE

PROJECT RAND

RESEARCH MEMORANDUM

Notes on Linear Programming: Part V

ALTERNATE ALGORITHM FOR THE REVISED SIMPLEX METHOD
Using a Product Form for the Inverse

George B. Dantzig
Wm. Orchard-Hays

RM-1268

ASTIA Document Number AD 114138

19 November 1953

Assigned to _____

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

ALTERNATE ALGORITHM FOR THE REVISED SIMPLEX METHOD

Using a Product Form for the Inverse*

George B. Dantzig

Wm. Orchard-Hays

Introduction

The simplex method is an algorithm for determining values for a set of n non-negative variables which minimizes a linear form subject to m linear restraints. It may be characterized briefly as a finite iterative procedure. Each iteration (or cycle) produces a new special solution to the restraint equations involving a subset of m of the variables, only one element of the subset changing on successive iterations; the remaining n-m variables are equated to zero. The vectors of coefficients corresponding to this subset of m variables are linearly independent and constitute a basis in m-dimension real vector space. Thus these bases are non-singular and possess inverses.

In the original simplex method [2a], [3], [4], each vector of coefficients associated with a variable is represented as a linear combination of the vectors of the latest basis. Since the changes of basis are step-wise, a simple recursion relation suffices to alter the representations on each iteration.

The revised simplex method [1], [5] differs from the original method in that it uses the same recursion relations to transform

---

* The authors wish to acknowledge that the present procedure was stimulated by earlier proposals by A. Orden along these lines.

only the inverse of the basis for each iteration. It has been introduced to reduce the quantity of writing at each step (which it does in general), and is particularly effective for linear programming models where the original matrix of coefficients is largely composed of zeros, as for example, in the transportation model [2c], [2d], or dynamic economic and production models [2b]. If the original method is used, these zeros would be replaced by non-zeros in the successive iterations and this greatly increases the computational effort. On the other hand, the revised method leaves those zeros intact.

In the important special case where the original matrix represents a dynamic situation, the matrix is composed of blocks of zeros in certain characteristic patterns, see [2b]. Here considerable advantage is gained by transforming only the inverse. This is the first step, however, and considerable research is under way which takes advantage of special ways of partitioning the matrix.

The revised method uses explicitly the concept of "prices" to determine which new variable to enter into the basic set in the next cycle. In many problems, due to the special nature of the coefficients in the optimizing form, prices often satisfy simple laws and can give clues on properties of optimizing solutions.

One important feature of the simplex method is the criterion for deciding which one of the vectors in the basis is to be replaced by a vector not in the basis to form the basis of the next iteration. When the constant terms of the restraint equations are not general, the choice of the vector to drop from the basis may be ambiguous and an arbitrary selection (as pointed out in unpublished examples

by Allan Hoffman and Philip Wolfe) may lead to non-convergence. Several devices exist, however, for perturbing the constant terms so as to avoid this difficulty. The earliest proposal along these lines [2a] consisted in modifying the vector of constant terms by a specially weighted combination of the unit vectors. This approach may be used conveniently both for the revised and original simplex methods, [1], [5]. With the original simplex method, there is another natural way to form the pertubation which consists in adding a weighted linear combination of the column vectors to the vector of constant terms. This was suggested first by Orden and developed independently by Charnes, [3].

Considerable attention has been paid to the above difficulty (called degeneracy) because it may lead to non-convergence. However from a practical point of view it usually does not lead to non-convergence. The type of problems in which it can cause non-convergence appear to be exceedingly rare. To date, there have been only two examples and both were artificially constructed for this purpose. Accordingly, the S.E.A.C. code and the electronic computer codes developed at RAND use an arbitrary selection criteria in case of ambiguity. In these codes, a deliberate decision was made to use a simple code in lieu of a more complex one needed to cover a possible case that may never arise in practice.

For those who wish to safeguard themselves from the possible (improbable) failure of the simplified rule used here, it is recommended that those rows of the inverse required (by the revised simplex method) to resolve ambiguity be computed directly from the product form as needed.

There are several advantages that accrue when the basis is represented as a product of "elementary" matrices,—for the basis, its transpose, its inverse and inverse transpose are readily available for vector multiplication.* By an "elementary matrix" is meant one formed from the identity matrix by replacing one column; thus an elementary matrix can be compactly recorded by the subscript of the altered column and the values of the elements in it. In the present procedure both the inverse and inverse transpose of the basis matrix are needed; more significant, however, is the fact that each iteration replaces one of the columns of the basis. In the product form of representation, this change can be conveniently effected by multiplying the previous matrix by an elementary matrix; thus, only one additional column of information need be recorded with each iteration. This approach places relatively greater emphasis on "reading" operations than "writing" and thereby reduces computation time.

The Problem

Determine values of n non-negative variables $(x_1, x_2, \ldots, x_n)$ which minimize the linear form

$$(1) \qquad a_{01}x_1 + a_{02}x_2 + \ldots + a_{0n}x_n$$

subject to the restraints

$$(2) \qquad \begin{cases} a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 & (x_j \geq 0 \text{ for } j = 1, \ldots, n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

---

* Using the I.B.M. Card Programmed Calculator, a novel feature results: when the inverse matrix is needed at one stage and its transpose at another, this is achieved simply by turning over the deck of cards representing the inverse.

or determine that no such values exist.

By changing signs of all coefficients in an equation in (2) if necessary it can be assumed that

$$(3) \qquad b_k \geq 0 \qquad\qquad k = 2, \ldots, m \ .$$

It has been found convenient to form a redundant equation* which is the negative sum of the equations in (2):

$$(4) \qquad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

where we define

$$(5) \qquad a_{1j} = -\sum_{k=2}^{m} a_{kj} \ , \qquad\qquad b_1 = -\sum_{2}^{m} b_k \ ,$$

and consider the following system in place of (1) and (2):

$$(6) \qquad \left\{ \begin{array}{lll} x_0 + (a_{01}x_1 + a_{02}x_2 + \ldots + a_{0n}x_n) & & = 0 \\ (a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n) + x_{n+1} & & = b_1 \\ (a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n) + x_{n+2} & & = b_2 \\ \vdots & & \\ (a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n) + x_{n+m} & & = b_m \end{array} \right.$$

where

$$x_j \geq 0 \ , \ (j = 1, \ldots, n) \text{ and } b_k \geq 0 \ , \ (k = 2, \ldots, m)$$

---

* This device as will become clearer later introduces a "dummy" set of unit vectors into the system. In many cases the original system provides some or all of the unit vectors. In this case the redundant equation is formed by omitting the corresponding equation in forming the sum (5) and omitting the corresponding variable in (6).

The new variables x_{n+k} measure the error between the left and right hand sides if a set of x_j do not satisfy (2) and $(-x_0)$ is the value of the linear form (1) to be minimized. Because of (5) it is easy to see that

$$x_{n+1} + x_{n+2} + \ldots + x_{n+m} = 0 \,. \tag{7}$$

If now we impose the conditions

$$x_{n+2} \geq 0, \ \ldots, \ x_{n+m} \geq 0 \tag{8}$$

then $(-x_{n+1})$ represents the absolute sum of errors of an approximate solution to (2). Accordingly, we consider only solutions to (6) in which (8) holds where one notes that there is no restriction on the sign of x_0 or x_{n+1}, (it is easy to construct such solutions). Next we try to maximize x_{n+1}, and if max $x_{n+1} = 0$, then <u>all</u> x_{n+k} <u>must vanish</u> and the remaining x_j satisfy (2). If max $x_{n+1} < 0$ then it is clear there is no solution to (2).

The <u>first phase</u> of the computational procedure will be to maximize x_{n+1} in (6) subject to (8). If a solution to (6) is obtained with $x_{n+1} = 0$ then the <u>second phase</u> will maximize x_0 (i.e. minimize the linear form) subject to the additional restriction $x_{n+1} = 0$.

Computational Procedure

For theoretical justification of the revised simplex method, the reader is referred to [1]. The procedure given here closely parallels [5] except for the modifications brought about by the use of the product form for the inverse of the basis. A proof given here that the terms of the products can be found by successive applications of step VIb of the procedure is the only result not found in [1].

We will now give the rules by which, starting with an initial basic solution and a corresponding initial basis, one can obtain successively new basic solutions and bases until (after a finite number of steps) one obtains an optimizing solution to (1) or determines that no solutions exist to (2). In certain cases solutions may exist to (2) for which there is no lower bound for the minimizing form (i.e. no upper bound for x_0), in which event a class of solutions will be constructed having this property.

Let the column vector of coefficients associated with a variable x_j in (6) be denoted by P_j; the column vector of constant terms by Q. It should be noted that $P_0, P_{n+1}, \ldots, P_{n+m}$ are m+1 component unit vectors. Denoting by U_k a unit vector with 1 in the (k+1)st position we have

$$P_0 = U_0 , \qquad P_{n+1} = U_1, \ldots, P_{n+m} = U_m \tag{9}$$

For the initial basic solution it is convenient to choose

$$x_0, x_{n+1}, x_{n+2}, \ldots, x_{n+m} \tag{10}$$

(the remaining variables being set equal to zero). The values of these variables are

$$x_0 = 0, x_{n+1} = b_1, \ldots, x_{n+m} = b_m \qquad (b_k \geq 0, k = 2,\ldots,m) \tag{11}$$

The initial basis B^0 is formed from the column of coefficients corresponding to the variables (10) in the basic solution. From (9) it is clear that B^0 is an identity matrix, i.e.

$$B^0 = [\, P_0, P_{n+1}, \ldots, P_{n+m} \,] = I \tag{12}$$

In general it may now be assumed that the variables in the k^{th} basic solution are

(13) $$x_0, x_{j_1}, \ldots, x_{j_i}, \ldots, x_{j_m}$$

and that the values of these variables which satisfy (6) are known,

(14) $$x_0 = v_0, \quad x_{j_1} = v_1, \quad x_{j_2} = v_2, \ldots, x_{j_m} = v_m \quad ;$$

and moreover that the inverse of the corresponding basis,

(15) $$[B^k]^{-1} = [P_0, P_{j_1}, \ldots, P_{j_m}]^{-1},$$

is known in the product form

(16) $$[B^k]^{-1} = E_k \cdot E_{k-1} \cdots E_1 \cdot E_0 \qquad k = 0, 1, \ldots$$

where $E_0 = I$ and E_r is an identity matrix except for column r whose components are

(17) $$\{\eta_{0r}, \eta_{1r}, \ldots, \eta_{mr}\}$$

The m+1 rows of a basis (or an E_r) will be referred to as row 0, row 1, ..., row m and similarly for columns. The following rules may now be applied to determine the (k+1)st solution from the k^{th} solution:

Iterative Procedure

Step I: Compute the row vector β by the appropriate formula

(18) $$\beta = U_1'[B^k]^{-1} \quad \text{if } x_{n+1} < 0 \text{ (phase I)}$$
$$= U_0'[B^k]^{-1} \quad \text{if } x_{n+1} = 0 \text{ (phase II)}$$

where U_k' is a unit row vector with 1 in position (k+1). (β is clearly row 1 of the inverse of the basis in phase I and row 0 in phase II).

Step II: Compute

$$\delta_j = (\beta P_j) \qquad (j = 1, 2, \ldots, n) \tag{19}$$

Step III:

(a) If there exist $\delta_j < 0$, choose s as the smallest index j such that

$$\delta_s = \min \delta_j < 0 \tag{20}$$

(b) If all $\delta_j \geq 0$, and

(i) if $x_{n+1} < 0$, (phase I), terminate procedure because no solution exists for (2)

(ii) if $x_{n+1} = 0$, (phase II), terminate procedure because x_0 is maximum and the basic solution is optimal.

Step IV: Compute the column vector $Y = (y_0, y_1, \ldots, y_m)$ where

$$Y = [B^k]^{-1} P_s \tag{21}$$

Step V: (a) If there exist $y_i > 0$, choose r as the smallest index i such that

$$v_r/y_r = \min(v_i/y_i) \qquad (y_i > 0,\ y_r > 0) \tag{22}$$

(b) If all $y_i \leq 0$, (possible only in phase II), terminate procedure because a class of solutions has been constructed such that $x_0 \to +\infty$ as $x_s \to +\infty$; the solutions are obtained by setting $x_j = 0$ except

$$\begin{cases} x_s \geq 0 & \text{arbitrary} \\ x_{j_i} = v_i - x_s \cdot y_i & (j_0 = 0,\ i = 0, 1, \ldots, m) \end{cases} \tag{23}$$

Step VI: (a) Replace variable x_{j_r} by variable x_s in (13) to form variables of the k+1st basic solution

(b) Form $[B^{k+1}]^{-1} = E_{k+1}[B^k]^{-1}$ where E_{k+1} is as an identity matrix except for row $r_{k+1} = r$ whose components are

$$\eta_{r,k+1} = 1/y_r, \quad \eta_{i,k+1} = -y_i/y_r \quad (i=0,1,\ldots,m,\ i\neq r) \tag{24}$$

(c) Form the values of new solution by setting

$$x_{j_i} = v_i + \eta_i v_r, \quad x_s = \eta_r v_r \quad (i\neq r,\ i=0,1,\ldots,m) \tag{25}$$

(d) Return to step I to start next cycle.

We shall now justify step VIb since this result is not contained in [1].

Theorem: The inverse of $[B^{k+1}]^{-1}$ may be written in the form

$$[B^{k+1}]^{-1} = [P_0, P_{j_1},\ldots, P_s,\ldots P_{j_m}]^{-1} = E_{k+1}[B^k]^{-1} \tag{26}$$

where P_s has replaced P_{j_r} and E_{k+1} is an identity matrix except for column r whose components are given by (24).

Proof: We simply note that

$$\left[E_{k+1}[B^k]^{-1}\right]P_{j_i} = E_{k+1}U_i = U_i \qquad (i \neq r), \tag{27}$$

$$\left[E_{k+1}[B^k]^{-1}\right]P_s = E_{k+1}Y = U_r \qquad ;$$

whence $E_{k+1}[B^k]^{-1}$ is the inverse of B^{k+1} by definition.

Vector Multiplication of a Matrix in Product Form

Consider the problem of computing a row vector β defined by multiplying $[B^k]^{-1}$ by a row vector $\alpha = U_0'$ or U_1', see (18).

$$\beta = \alpha E_k E_{k-1} \cdots E_0 \tag{28}$$

It is clear that β can be obtained by successive transformations on row vectors, i.e., forming $(\alpha)E_k$, $(\alpha E_k)E_{k-1}, \ldots$, etc. However, when any row vector $A = (a_0, a_1, \ldots, a_m)$ is transformed into a row vector $B = (b_0, b_1 \ldots, b_m)$ by multiplying A on the right by an elementary matrix E_ℓ one obtains simply

$$\begin{aligned} b_i &= a_i \qquad (i \neq r_\ell) \\ b_{r_\ell} &= \sum_0^m \eta_{i\ell}\, a_i \ . \end{aligned} \tag{29}$$

Consider next the problem of computing Y by relation (22).

$$Y = B^{-1}P_s = E_k E_{k-1} \cdots E_0 P_s \ . \tag{30}$$

It is clear that Y can be determined by successive transformations on column vectors, i.e., forming $E_0(P_s)$, $E_1(E_0P_s)$ ..., etc. However, when a column vector $C = \{c_0, c_1, \ldots, c_m\}$ is transformed into a column vector $D = \{d_0, d_1, \ldots, d_m\}$ by multiplying C on the left by a matrix of the special form E_ℓ, one obtains simply

$$\begin{aligned} d_i &= c_i + \eta_{i\ell} c_{r_\ell} \qquad (i \neq r_\ell) \\ d_{r_\ell} &= \eta_{r_\ell \ell}\, c_{r_\ell} \end{aligned} \tag{31}$$

From (29) and (31) it is clear that the only essential information contained in E_ℓ is the set of values $\eta_{i\ell}$ and the index r_ℓ. Note further that in (28), the E_ℓ are used in decreasing sequence of ℓ but from (29) it is not necessary to know r_ℓ until after the $\eta_{i\ell}$ have been used. On the other hand, in (30), the successive E_ℓ are used with increasing ℓ and it follows from (31) that it is necessary to know r_ℓ before using the $\eta_{i\ell}$. The perfect complementarity of the preceding two sentences, together with the fact that $\sum_{0}^{m} \eta_{i\ell}\, a_i$ can obviously be computed starting with i = m as well as with i = 0, makes it clear that (30) may be computed using the information in the reverse order of that used for (28).*

Let L_ℓ denote the ordered set of 'words' of information

$$(32) \qquad L_\ell = \left\{ r_\ell \; ; \; \eta_{0\ell} \, , \, \eta_{1\ell} \, , \, \cdots \, , \, \eta_{m\ell} \right\}.$$

Then each change of a column of B will produce a new $L_{\ell+1}$ which may be recorded in consecutive order to the previously computed $L_1, L_2, \ldots, L_\ell$.

In this compact manner only (m+1) additional words of information (and an index) are required to represent each successive inverse of the basis in product form.

---

* On the CPC, by punching two sets of instructions on each card—one being, in form, the reflection, in the vertical center line, of the other (with appropriate adjustments for difference in algorithms (29) and (31)—the transpose uses of the inverse in (28) and (30) may be accomplished by simply turning the cards over using the vertical center line of the card for the axis.

References

[1] G. B. Dantzig, Alex Orden, Philip Wolfe,
The Generalized Simplex Method
(RAND RM-1264 dated 5 April 1954).

[2] Activity Analysis of Production and Allocation,
T. C. Koopmans, Editor, (Wiley 1951)

(a) Dantzig, George B., Maximization of a Linear Function of Variables Subject to Linear Inequalities

(b) ———————, The Programming of Interdependent Activities: Mathematical Model

(c) ———————, Application of the Simplex Method to a Transportation Problem

(d) Koopmans, T. C. , A Model of Transportation

[3] A. Charnes, W. W. Cooper, and A. Henderson,
An Introduction to Linear Programming (Wiley 1953)

[4] A. Hoffman, M. Mannos, D. Sokolowsky, and N. Wiegmann
Computational Experience in Solving Linear Programs
N.B.S. Report 2501 dated May 15, 1953.

[5] G. B. Dantzig , Computational Algorithm of the Simplex Method
(RAND RM-1266 dated 26 October 1953)