U. S. AIR FORCE

PROJECT RAND

RESEARCH MEMORANDUM

Notes on Linear Programming — Part III:

COMPUTATIONAL ALGORITHM
OF THE
REVISED SIMPLEX METHOD

George B. Dantzig

RM-1266

ASTIA Document Number AD 114136

26 October 1953

Assigned to ______________________

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

SUMMARY

The computational procedure is given for the revised simplex method for minimizing a linear form under linear inequality restraints.

For theoretical justification of the revised simplex method, the reader is referred to the joint paper of Dantzig, Orden, and Wolfe, entitled "The Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Restraints," RAND RM-1264, dated 26 October 1953.

COMPUTATIONAL ALGORITHM OF THE REVISED SIMPLEX METHOD

George B. Dantzig

Introduction

The simplex method is an algorithm for determining values for a set of n nonnegative variables which minimizes a linear form subject to m linear restraints [1]. It may be characterized briefly as a finite iterative procedure. Each iteration produces a new special solution to the restraint equations involving a subset of m of the variables (called a basic set of variables), only one element of the basic set of variables changing on successive iterations; the remaining n - m variables are equated to zero. The matrix of coefficients (associated with the subset of m variables) is called a basis matrix. Each cycle (iteration) produces a new basis which differs from the previous one by only one column. These bases are nonsingular and possess inverses.

The revised simplex method [1] differs from the original simplex method [2], [3] in two main respects:

(a) The rule for which variable to drop (in the change of cycle) does not always lead to a unique choice in the original simplex method.* The algorithm presented here — where the inverse of the basis is available — gives a sharper

*In unpublished special examples, A. Hoffman and P. Wolfe have shown that the rule can lead to nonconvergence. Cases where this can occur appear to be so rare that to date only two examples have come to the author's attention and both were artificially constructed.

rule for choice which is unique, always leads to convergence, and usually takes little or no extra computational effort.

(b) The original simplex method transforms the coefficients associated with all the variables on each cycle by means of simple recursion relations. The revised method uses these same relations but transforms only the inverse of the basis.

The transformation of just the inverse (rather than the entire matrix of coefficients with each cycle) has been developed because it has several important advantages over the old method:

(a) When the original matrix of coefficients is largely composed of nonzero elements both methods yield the same number of multiplications per cycle (roughly m x n). However, in the original method (roughly) m x n new elements have to be recorded each time. In contrast, the revised method (by making extensive use of cumulative sums of products) requires the recording of about m^2 elements (and an alternative method [5] can reduce this to m if one goes back to the original weaker rule of which variable to drop).

(b) In most practical problems the original matrix of coefficients is largely composed of zero elements. The older procedure generated nonzero elements in its successive transformations of the matrix. The revised method works with the matrix in its original form and takes direct advantage of these zeros.

(c) 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 [4]. 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.

(d) The revised method uses 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 can be computed by simple rules and these often give clues on properties of optimizing solutions.

The Problem

Consider a system of equations in n <u>nonnegative</u> variables $(x_1, x_2, \cdots, x_n)$:

$$x_0 + a_{01}x_1 + a_{02}x_2 + \cdots + a_{0n}x_n = 0 \tag{1}$$

$$\left\{\begin{array}{l} a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{array}\right. \qquad (x_j \geq 0 \text{ for } j = 1, \cdots, n) \tag{2}$$

Our problem is to find values $x_j \geq 0$ $(j = 1, 2, \cdots, n)$ satisfying (2) such that x_0 in (1) is maximum. It is clear that this is the same as finding a solution to (2) which minimizes the linear form $\sum_{1}^{n} a_{0j}x_j$.

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

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

Based on a suggestion of W. Orchard-Hays, it has been found convenient to form a redundant equation which is the negative sum of the equations in (2):

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

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):

$$(6) \qquad \begin{cases} x_0 \;\; + (a_{01}x_1 + a_{02}x_2 + \cdots + a_{0n}x_n) = 0 & (x_j \geq 0,\ j=1,\cdots,n) \\ x_{n+1} + (a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n) = b_1 & \\ x_{n+2} + (a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n) = b_2 & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ x_{n+m} + (a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n) = b_m & \end{cases}$$

where we have introduced the variables x_{n+k} which measure the error between left and right-hand sides if a set of x_j do not satisfy (2). Because of (5) it is easy to see that

$$(7) \qquad x_{n+1} + x_{n+2} + \cdots + x_{n+m} = 0 .$$

If now we impose the conditions

(8) $$x_{n+2} \geq 0, \cdots, x_{n+m} \geq 0$$

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 all x_{n+k} must vanish and the remaining x_j satisfy (2). If max $x_{n+1} < 0$ then it is clear there is no solution to (2).

The first phase 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 second phase will maximize x_0 in (6) subject to the additional restriction $x_{n+1} = 0$.

Computational Procedure

Each cycle (or iteration) produces a "tableau" (see Tableau I) in which at the start of a cycle all entries are known except in the last column. The solution associated with a cycle is obtained by setting all variables $x_j = 0$ except those whose values are given in the tableau. The initial tableau has a very simple form and it is clear that the starting solution is simply $x_0 = 0$, $x_{n+1} = b_1, \cdots, x_{n+m} = b_m$ (see Tableau III). The following rules apply to all cycles:

Step I: Compute δ_j for $j = 1, 2, \cdots, n$ by the appropriate formula:

$$\delta_j = \beta_{10}a_{0j} + \beta_{11}a_{1j} + \cdots + \beta_{1m}a_{mj} \qquad (\text{if } x_{n+1} < 0)$$

$$= \beta_{00}a_{0j} + \beta_{01}a_{1j} + \cdots + \beta_{0m}a_{mn} \qquad (\text{if } x_{n+1} = 0) .$$

Step II: (a) If all $\delta_j \geq 0$ terminate procedure because

(i) if $x_{n+1} < 0$, then x_{n+1} is maximum and no solution exists for (1).

(ii) if $x_{n+1} = 0$, then x_0 is maximum and basic solution is optimum.

(b) Otherwise choose x_s, the variable to enter into next tableau in place of x_{j_r} (r to be determined), such that s is the smallest index satisfying

$$\delta_s = \min \delta_j < 0 .$$

Step III: Compute for $k = 0, 1, \cdots, m$,

$$y_k = \beta_{k0}a_{0s} + \beta_{k1}a_{1s} + \cdots + \beta_{km}a_{ms}$$

and enter values of y_k in last column for $k = 0, 1, 2, \cdots, m$.

Step IV: (a) If $x_{n+1} = 0$ and all $y_k \leq 0$, terminate procedure because a class of solutions has been constructed such that $x_0 \longrightarrow +\infty$ as $x_s \longrightarrow +\infty$; the solutions are obtained by setting $x_j = 0$ except

$$\begin{cases} x_s \geq 0 \quad \text{arbitrary} \\ x_{j_i} = v_i - x_s y_i \end{cases} \qquad (i = 0, \cdots, m) .$$

(b) Otherwise choose x_{j_r}, the variable to be replaced by x_s in the next tableau, such that r is chosen from those i where

$$y_i > 0 \qquad (i \neq 0)$$

and satisfies

$$(v_r/y_r) = \min (v_i/y_i) \, .$$

If two or more indices r_1, r_2, ··· are tied for minimum divide elements in the <u>first column of the inverse</u> in rows r_1, r_2, ... by corresponding y_{r_1}, y_{r_2}, ..., and take the index of the row with the minimizing ratio for r. If there still remain ties, repeat for those indices that are still tied using as ratio the corresponding entries in the <u>second column of the inverse</u> divided by their respective y_{r_i}. Ratios formed from successive columns of the inverse are used until all ties are resolved. (Since no two rows of an inverse can be proportional, a <u>unique</u> r will be chosen by the last column.)

<u>Step V</u>: Entries for row i of the next tableau (see Tableau II) are obtained by multiplying entries in row r by $\eta_i = -y_i/y_r$ and adding to entries in row i. The exception is row i = r which is formed by multiplying entries in row r by $\eta_r = 1/y_r$. The designations of the variables entering into the basic solution remain the same except x_{j_r} is replaced by x_s. The last column is left blank. The cycle is now complete; return to Step I.

Simplex Method

Row	Variables in Basic Solution	Values of the Basic Variables	Inverse of Basis				Y
			I. Tableau at End of Cycle				
0	x_0	v_0	β_{00}	β_{01}	$\cdots$	β_{0m}	y_0
1	x_{j_1}	v_1	β_{10}	β_{11}	$\cdots$	β_{1m}	y_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
r	x_{j_r}	v_r	β_{r0}	β_{r1}	$\cdots$	β_{rm}	y_r
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	x_{j_m}	v_m	β_{m0}	β_{m1}	$\cdots$	β_{mm}	y_m
			II. Tableau at Start of Next Cycle*				
0	x_0	$v_0+\eta_0 v_r$	$\beta_{00}+\eta_0\beta_{r0}$	$\beta_{01}+\eta_0\beta_{r1}$	$\cdots$	$\beta_{0m}+\eta_0\beta_{rm}$	
1	x_{j_1}	$v_1+\eta_1 v_r$	$\beta_{10}+\eta_0\beta_{r0}$	$\beta_{11}+\eta_1\beta_{r1}$	$\cdots$	$\beta_{1m}+\eta_1\beta_{rm}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	
r	x_s	$+\eta_r v_r$	$+\eta_r\beta_{r0}$	$+\eta_r\beta_{r1}$	$\cdots$	$+\eta_r\beta_{rm}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	
m	x_{j_m}	$v_m+\eta_m v_r$	$\beta_{m0}+\eta_m\beta_{r0}$	$\beta_{m0}+\eta_m\beta_{r1}$	$\cdots$	$\beta_{mm}+\eta_m\beta_{rm}$	
			III. Tableau at Start of First Cycle				
0	x_0	0	1	0	$\cdots$	0	
1	x_{n+1}	b_1	0	1	$\cdots$	0	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	
r	x_{n+r}	b_r	0	0	1	0	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	
m	x_{n+m}	b_m	0	0	$\cdots$	1	

*In Tableau II, x_s has replaced x_{j_r}, $\eta_r = 1/y_r$ and $\eta_i = -y_i/y_r$ $(i \neq r)$.

References

[1] G. B. Dantzig, A. Orden, P. Wolfe, The Generalized Simplex Method, RAND RM-1264, 5 April 1954.

[2] G. B. Dantzig, Maximizing of a Linear Function of Variables Subject to Linear Inequalities, Chap. XXI, Activity Analysis of Production and Allocation, T. C. Koopmans, Ed., John Wiley and Sons, 1951.

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

[4] G. B. Dantzig, The Programming of Interdependent Activities: Mathematical Model, Chap. II, Activity Analysis of Production and Allocation, T. C. Koopmans, Ed., John Wiley and Sons, 1951.

[5] G. B. Dantzig, William Orchard-Hays, The Product Form for the Inverse in the Simplex Method, RAND P-440, 9 October 1953.