U. S. AIR FORCE

PROJECT RAND

RESEARCH MEMORANDUM

NOTES ON LINEAR PROGRAMMING: PART XI
COMPOSITE SIMPLEX--DUAL SIMPLEX ALGORITHM--I

George B. Dantzig

RM-1274

ASTIA Document Number AD 114140

26 April 1954

Assigned to ______________________________

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

SUMMARY

This paper presents an idea by E. M. L. Beale for avoiding certain undesirable features of phase I of the simplex method, namely, the introduction of vectors in the basis that are not likely to be in the optimal solution of phase II.

COMPOSITE SIMPLEX—DUAL SIMPLEX ALGORITHM—I *

George B. Dantzig

Recently E. M. L. Beale has proposed a technique which may reduce the number of iterations of the simplex method in those cases where it is necessary to first find a "basic feasible solution." The first phase of the standard simplex method tries to drive out the artificial vectors in the basis and, in so doing, introduces vectors into the basis without any consideration of the form eventually to be minimized. This means that the second phase, where an optimum feasible solution is obtained, must go through at least one extra iteration for each vector brought into the basis in phase I, which later turns out not to be in the optimal basis of phase II. Beale introduced his device into a special variant of the dual simplex algorithm which he has developed. However, it can just as conveniently be introduced into the standard simplex set-up and we shall apply it in this form in this paper.**

We shall use the standard simplex criterion to decide which vector to bring into the basis in order to decrease the value of the linear form. We shall use a new criterion (in a little more precise way than that suggested by Beale) as to which vector to drive out of the basis in order to decrease a certain parameter α

---

*In Part XII of this series, a second Composite Simplex Algorithm is proposed by W. Orchard-Hays which leads to a simpler algorithm.

**It is conjectured that where n is considerably larger than m, the regular simplex procedure is likely to reach an optimum in fewer iterations than the dual. If so, this arrangement will be more efficient. See discussion in section (IV).

where α is a parameter used to "perturb" the original system of equations. At a certain stage in the problem it may happen that an optimal solution has been obtained before the parameter α has decreased to the value zero. ($\alpha = 0$ is the point at which the perturbation coincides with the original system.) However, if the solution is optimal for some $\alpha > 0$, it is, in fact, a basic feasible solution to the dual problem so that it is now convenient to switch over to the dual simplex algorithm (originated by C. E. Lemke) and complete the problem [5]. If the reader refers to references [1, 2, 3, 4], he will note that a standard system of equations and notation has been set up for both the original and dual systems so that it is possible to combine these into a composite algorithm which is convenient to apply. In addition to this, we shall make use of the product form for the inverse of the basis, [3], which has many advantages, particularly in reducing the quantity of writing.

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

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

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):

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

---

*This device, as will become clearer later, introduces a "dummy" or "artificial" set of unit vectors into the system. In many cases the original system may provide some (or all) of the unit vectors. In this case the redundant equation is best formed by omitting the corresponding equation in forming the sum (5), omitting the corresponding variable in (6), and introducing the corresponding non-artificial unit vectors and variables into the initial basis and solution.

where

(7) $x_j \geq 0$ $j = 1, 2, \ldots, (n+m)$

The new variables x_{n+k} can be thought of as measuring the error between the left- and right-hand sides of (2) when a set of x_j do not satisfy the system exactly. Because of (5) it is easy to see that

(8) $x_{n+1} + x_{n+2} + \ldots + x_{n+m} = \alpha$ $,(x_{n+i} \geq 0)$

and it is clear that if a solution is obtained for $\alpha = 0$, then $x_{n+k} = 0$ $(k = 1, 2, \ldots, m)$ so that system (6) reduces to system (2).

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 $Q + \alpha U_1$. 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

(9) $P_0 = U_0$, $P_{n+1} = U_1, \ldots, P_{n+m} = U_m$.

For the initial basic solution it is convenient to choose

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

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

(11) $x_0 = 0$, $x_{n+1} = (b_1 + \alpha)$, $x_{n+2} = b_2, \ldots, x_{n+m} = b_m$

where

(11.1) $(b_k \geq 0, k = 2, \ldots, m)$ and $b_1 = -\sum_2^m b_k \leq 0$.

In order for $x_{n+1} \geq 0$, it is necessary that $\alpha \geq -b_1 \geq 0$ initially.

The initial basis B^0 is formed from the column of coefficients corresponding to the variables (11) in the basic solution. From (10) it is clear that B^0 is an identity matrix, i.e.,

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

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 in terms of the parameter α in the form

(14) $x_{j_i} = v_i + \alpha \cdot u_i$ $(j_0 = 0;\ i = 0, 1, \ldots, m)$

where for the k^{th} basis there exists a smallest value of $\alpha = \alpha_k \geq 0$ such that $x_{j_i} \geq 0$; thus the basic solution can be written

(15) $\sum_0^m (v_i + \alpha \cdot u_i)P_{j_i} = Q + \alpha U_1$.

The inverse of the basis,

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

will be assumed to be known in the product form [3],

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

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

(18) $\{\eta_{0\ell}, \eta_{1\ell}, \dots, \eta_{m\ell}\}$.

The m+1 <u>rows</u> of a basis (or an E_ℓ) will be referred to as <u>row</u> 0, <u>row</u> 1, ..., <u>row</u> m (and similarly for <u>columns</u>). It should be noted that when it is required to multiply a row vector $(a_0, a_1, \dots, a_m)$ on the right by an elementary matrix E_ℓ, one obtains a new row vector $(a_0^*, a_1^*, \dots, a_m^*)$ where

(19) $$\begin{cases} a_i^* = a_i & (i \neq r_\ell) \\ a_r^* = \sum \eta_{i\ell} a_i & (r = r_\ell) \end{cases}$$;

similarly when it is required to multiply a column vector $\{c_0, c_1, \dots, c_m\}$ on the left by an elementary matrix E_ℓ, one obtains a new column vector $\{c_0^*, c_1^*, \dots, c_m^*\}$ where

(20) $$\begin{cases} c_i^* = c_i + \eta_{i\ell} c_r & (i \neq r_\ell) \\ c_r^* = \quad \eta_{r\ell} c_r & (r = r_\ell) \end{cases}$$.

For further references on the product form see [3].

<u>Iterative Procedure</u>

(I) <u>The Question of which Vector to Introduce into the Basis</u>: The first part of the simplex algorithm is concerned with the determination of the vector P_s to introduce into the next basis in order to increase the value of the solution x_0. For this purpose, the first step is the computation of the top or 0-row of the inverse of the basis B^k (denoted by β_0) by the formula

$$(21) \quad \beta_0 = U_0' \, [B^k]^{-1} = U_0' \cdot E_k \cdot E_{k-1} \cdots \cdot E_1 \, .$$

where U_0' is the unit row vector with 1 in the first component. β_0 is often called the "pricing **vector**." It is clear that β_0 can be obtained from (21) by successive multiplications of a row vector by an elementary matrix, see (19). With β_0 form the scalar products

$$(22) \quad \delta_j = (\beta_0 P_j) \qquad\qquad j = 1, 2, \ldots, n \quad .$$

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

$$(23) \quad \delta_s = \min \delta_j < 0 \quad .$$

The corresponding vector P_s is the one to be introduced into the basis.

(b) If <u>all</u> $\delta_j \geq 0$ and $\alpha_k = 0$ then the procedure <u>terminates</u> because x_0 is maximum and the optimal solution to (1) is $x_0 = v_0$, $x_{j_i} = v_i$, and $x_j = 0$ otherwise for $j \neq j_i$.

(c) If <u>all</u> $\delta_j \geq 0$ but $\alpha_k > 0$, then the next step is to apply the <u>Dual Simplex Algorithm</u> part of this procedure which is discussed in section (III).

(II) <u>The Question of which Vector to Drop from the Basis</u>: If a vector P_s is chosen by (Ia), the next step is the determination of the vector P_{j_r} to remove from the basis. For this purpose, it is necessary to represent P_s as a linear combination of the vectors in the basis:

$$(24) \qquad P_s = \sum_0^m y_i P_{j_i} = BY_s \qquad\qquad (j_0 = 0)$$

where the column vector of weights $Y_s = \{y_0, y_1, \ldots, y_m\}$ can be computed by

$$(25) \qquad Y_s = [B^k]^{-1} P_s = E_k \cdot E_{k-1} \ldots E_1 P_s \quad .$$

It will be noted that Y_s can be obtained from (25) by successive multiplications of a column vector by an elementary matrix, see (20). We now consider a class of solutions formed from (15) and (24)

$$(26) \qquad \sum_0^m (v_i + \alpha u_i - \theta y_i) P_{j_i} + \theta P_s = Q + \alpha U_1$$

where θ and α are free parameters chosen such that

$$(27) \qquad x_{j_i} = v_i + \alpha u_i - \theta y_i \geq 0 \qquad\qquad i = 1, 2, \ldots, m$$

$$(28) \qquad x_s = \theta \geq 0 \qquad \alpha \geq 0$$

and such that

(29) $\alpha = \text{Min} \geq 0$ and $\theta = \text{Max}$ (given the value α is minimum).

For the smallest value of α, the value of the solution is $v_0+\alpha\cdot u_0-\theta y_0$. Since $y_0 = \beta_0 P_s = \delta_s \leq 0$, it is clear that to maximize, the value θ is chosen maximum analogous to the regular simplex process. In order not to interrupt the discussion of the main routine we shall postpone the discussion of the evaluation of Min $\alpha = \alpha_{k+1}$ to section (IV). It should be noted that if $\alpha_k = 0$, then $\alpha_{k+1} = 0$ also, and the rules below for decidng which vector to drop from the basis are the same as the regular simplex process.

Once the minimum value of $\alpha = \alpha_{k+1}$ is determined, see section (IV), then the vector P_{j_r} to be dropped from the basis is chosen such that

(i) if there exist $y_i > 0$,

$$(30.1) \qquad \theta = \frac{v_r+u_r\cdot\alpha_{k+1}}{y_r} = \operatorname*{Min}_{y_i > 0} \frac{v_i+u_i\cdot\alpha_{k+1}}{y_i} , \quad (y_r > 0) ;$$

(ii) if all $y_i \leq 0$ and $\alpha_{k+1} > 0$,

$$(30.2) \qquad \theta = \frac{v_r+u_r\cdot\alpha_{k+1}}{y_r} = \operatorname{Max} \frac{v_i+u_i\cdot\alpha_{k+1}}{y_i}$$

(in order to have a basic solution with a smaller value of α);

(iii) if all $y_i \leq 0$ and $\alpha_{k+1} = 0$, then the <u>algorithm terminates</u> since

$$(30.3) \qquad x_{j_i} = v_i - y_i\theta, \quad x_s = \theta, \quad x_j = 0 \qquad (j \neq j_i \text{ or } s)$$

constitutes a class of feasible solutions whose values $x_0 = v_0 - y_0\theta$ have no upper bounds as $\theta \rightarrow +\infty$. Otherwise, having replaced P_{j_r} in the basis by P_s, a cycle has been completed and one returns to (I).

(III) <u>The Dual Simplex Algorithm</u> [4]: For a given $\alpha_k > 0$ it has been determined that <u>all</u> $\delta_j \geq 0$. The first step of the dual algorithm is to set

(31) $\alpha = 0$

(32) $x_{j_i} = \begin{cases} -x_{j_i} & \text{if } j_i > n \text{ and } x_{j_i} > 0 \\ +x_{j_i} & \text{otherwise} . \end{cases}$

The vector P_{j_r} to drop from the basis is chosen such that

(33) $x_{j_r} = \text{Min}\ x_{j_i} < 0$.

If <u>all</u> $x_{j_i} \geq 0$, <u>the iterative process is terminated because an optimal basic solution has been obtained</u>.

Assuming r can be determined the next step is to find the vector P_s to enter the basis. For this purpose form the r^{th} row of $[B^k]^{-1}$

(34) $\beta_r = U_r\ [B^k]^{-1} = U_r \cdot E_k \cdot E_{k-1} \cdots E_0$

by successive multiplications of row vectors by elementary matrices on the right (19). Set

(35) $\beta_r = \begin{cases} -\beta_r & \text{if } j_r > n \text{ and } x_{j_r} > 0 \\ +\beta_r & \text{otherwise} \end{cases}$

and determine

(36) $\rho_j = \beta_r P_j \qquad (j = 0, \ldots, n)$.

Then the vector P_s is determined such that

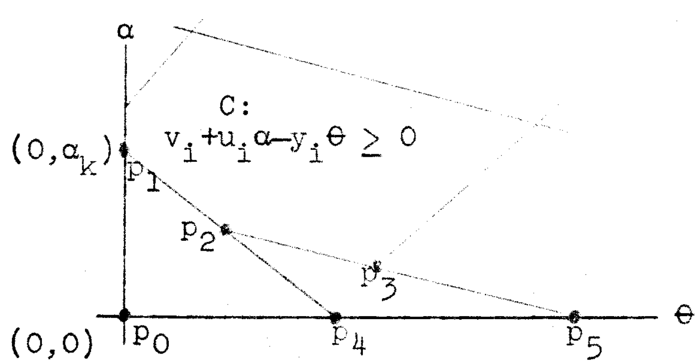
(37) $\phi = \frac{\delta_s}{\rho_s} = \underset{\rho_j < 0}{\text{Max}} \frac{\delta_j}{\rho_j} \leq 0 \qquad (\rho_s < 0) \quad (j = 1, \ldots, n)$.

If all $\rho_j \geq 0$, then the iterative process terminates; there is no feasible solution to the system of equations, see [4], otherwise, for the next basis, the new values of δ_j denoted by δ_j^* are

(38) $\delta_j^* = \delta_j - \phi\rho_j \geq 0 \qquad j = 0, \ldots, n$.

The dual simplex algorithm may now be iterated with the new basis. Having replaced P_{j_r} in the basis by P_s a cycle has been completed and one returns to (III).

(IV) How to Minimize α_{k+1}: Geometrically, conditions (27) and (28) define a convex C in the plane (θ,α)



where the equations of the bounding lines $v_i + \alpha \cdot u_i - \theta y_i = 0$ are known. It will be noted that $\theta = 0$, $\alpha = \alpha_k > 0$ satisfies (27) and (28) so that C is non-empty. The lowest point in the convex has the minimum value of $\alpha = \alpha_{k+1}$. Since the convex may be flat on the bottom, for this value, θ is chosen maximum (except in case where θ has no upper bound).

This sub-problem of determining the minimum α is itself a (m+2) x 2 linear programming problem which is more conveniently set up in its dual form. The dual of this sub-problem is to determine non-negative λ_i, i = 1, 2, ..., m+2 such that λ_0 = Max and λ_i satisfies

$$(39) \quad \begin{cases} \lambda_0 + v_1\lambda_1 + v_2\lambda_2 + \dots + v_m\lambda_m = 0 \\ \lambda_{m+1} + u_1\lambda_1 + u_2\lambda_2 + \dots + u_m\lambda_m = 1 \\ \lambda_{m+2} - y_1\lambda_1 - y_2\lambda_2 + \dots - y_m\lambda_m = 0 \end{cases} .$$

It will be noted that the optimum "price" vector, $\beta = [1, \alpha, \theta]$, for this sub-system satisfies

$$(40) \quad \alpha \geq 0, \quad \theta \geq 0, \quad v_i + u_i\alpha - y_i\theta \geq 0, \quad \alpha = \text{Min} \qquad ;$$

see [1]. We shall consider two alternative ways to solve (39): by the dual simplex method, [4], and the direct simplex method, [1]. In addition, there is, of course, the possibility of more rapid methods which estimate the minimum value of α.

(a) For the dual method, one may begin with the convenient basic solution corresponding to the point $p_1 = (0, \alpha_k)$ in the diagram. If so, then it is interesting to note that for each iteration of the dual simplex algorithm the solution will move from one neighboring point to the next on the convex, i.e., from p_1 to p_2 to p_3, etc. until a minimum is reached. The variables in the <u>initial</u> basic solution are $[\lambda_0, \lambda_{r_0}, \lambda_{m+2}]$, where λ_{r_0} is chosen such that

(41) $$v_{r_0} + u_{r_0} \cdot \alpha_k = 0 \qquad (u_{r_0} < 0) \quad .$$

(b) For the regular simplex method, one may begin with a convenient basic solution corresponding to the point p_0 = (0,0) in the diagram. It can be shown that the next iteration of the simplex algorithm will first choose a line $v_i + u_i \cdot \alpha - y_i \theta = 0$ which separates p_0 from the convex and is the <u>farthest distance</u> from p_0 (to be precise, a maximum weighted distance unless the boundary lines have been normalized, i.e., $u_i^2 + y_i^2 = 1$). This is the line joining p_1 to p_2, and p_4 becomes the next solution* in place of p_0. Upon iteration, the next solution point is p_5 and then finally p_3.

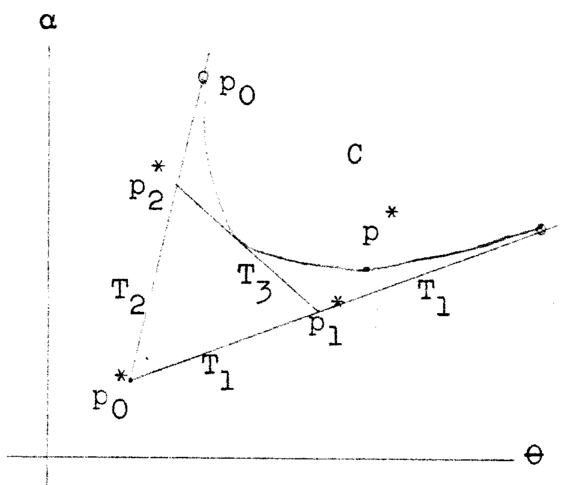
Further insight into how the simplex and dual simplex methods compare can be seen in the case where the boundary of the convex is considered as continuous. The problem is to determine a point p^* on C such that a tangent** line to C

---

*The rule for choosing p_4 after p_0 (and not p_1) is that $\alpha \geq c$ must contain the convex C where $\alpha = c$ passes through the point in question. It can be shown that the point with the smallest α coordinate always has this property.

**Strictly speaking the simplex algorithm works not with tangents but with boundaries of half spaces which cover C.

through p^* has a specified direction (we shall assume that it is parallel to the θ axis and must lie below C). The dual simplex starting at any point p_0 on boundary of C will slide down the curve to the lowest point p^*. The regular simplex starts with any point p_0^* with α coordinate less than that of p^* with tangent lines T_1, T_2 to C. Next a tangent line T_3 is constructed on the same side of C as p_0^* and at a <u>maximum distance</u> from p_0^*. A new "solution" point can now be obtained from one of the two intersections of T_3 with T_1 and T_2, i.e., either p_1^* or p_2^*, depending on which has the smallest α coordinate—in this case p_1^*. The algorithm is repeated now with p_1^* as solution point with tangent lines T_1 and T_3.



Since boundary of C is, in fact, a series of broken lines the dual algorithm (which proceeds from one neighboring vertex on C to the next) may require many iterations, whereas the regular simplex algorithm appears to close in rapidly on p^*. Since both algorithms are really the same algorithm being applied to a problem and its dual, this discussion must only be interpreted to mean that when n is large compared to m (or for the special case m is large compared to 2) the dual algorithm is likely to be slower than the regular one.

REFERENCES

[1] Dantzig, George B., Alex Orden, and Philip Wolfe, "The Generalized Simplex Method for Minimizing a Linear Form under Linear Inequality Restraints," RAND RM-1264, Rev. 5 April 1954.

[2] ——————, ——————, "Duality Theorems," RAND RM-1265, Rev. 30 October 1953.

[3] ——————, Wm. Orchard-Hays, "Alternate Algorithm for the Revised Simplex Method using a product form for the inverse," RAND RM-1268, 19 November 1953.

[4] ——————, "The Dual Simplex Algorithm," RAND RM-1270, Rev. 3 May 1954.

[5] Lemke, Carlton E., "The Dual Method for Solving the Linear Programming Problem," Carnegie Institute of Technology, Dept. of Math., Technical Report No. 29, March 4, 1953.