

AD 607025

10
Su

THE CROSS-SECTION METHOD
An Algorithm for Linear Programming

Jeremy J. Stone

✓
P-1490

September 16, 1958

Approved for OTS release

COPY	OF
HARD COPY	\$. 1 00
MICROFICHE	\$. 0 50

DDC 24
RECEIVED
OCT 5 1964
DDC-IRA C

The **RAND** Corporation

1700 MAIN ST. • SANTA MONICA • CALIFORNIA

ARCHIVE COPY

SUMMARY

In this paper an algorithm is presented which solves the linear programming problem. This algorithm combines the usual phase one (getting feasibility) and phase two (getting optimality) of the Simplex or Dual methods into a single phase. The algorithm begins with either a single activity (column) or a constraint (equation) and proceeds to add either activities or constraints one at a time, solving the subproblems which arise for their optimal solutions. The final solution is attained after adding the last activity or constraint. The algorithm promises to be an efficient one and has several advantages which arise from the information supplied about subproblems.

R-11

SUMMARY

In this paper an algorithm is presented which solves the linear programming problem. This algorithm combines the usual phase one (getting feasibility) and phase two (getting optimality) of the Simplex or Dual methods into a single phase. The algorithm begins with either a single activity (column) or a constraint (equation) and proceeds to add either activities or constraints one at a time, solving the subproblems which arise for their optimal solutions. The final solution is attained after adding the last activity or constraint. The algorithm promises to be an efficient one and has several advantages which arise from the information supplied about subproblems.

THE CROSS-SECTION METHOD

An Algorithm for Linear Programming

INTRODUCTION

Both the Simplex [2] and Dual [6] methods for solving linear programming problems require two distinct phases. In the first phase a feasible solution to the problem is found and in the second phase an optimal solution is secured. This paper develops a method for solving linear programming problems which combines these two phases.* For this and other reasons, which are examined in detail in Section 4, it promises to be more efficient than existing techniques. Furthermore, it has certain advantages which, apart from efficiency, make it superior to the Simplex or Dual methods as now applied. These advantages are due to the fact the method supplies, in addition to the final answer, the optimal solution to subproblems constructed by ignoring certain constraints (equations) or activities (columns) of the original problem. This characteristic of the technique permits one to compute without difficulty solutions to problems which differ only in certain activities or constraints, without solving separate linear programming problems. These advantages are described in Section 5.

The central contribution of this paper is a method of going in one "step" from an extreme point (e.p.) feasible dual

*The author is indebted to Dr. George B. Dantzig for posing a problem which inspired this investigation.

solution* of one linear programming problem to an e.p. feasible dual solution of another linear programming problem consisting of the first problem with an additional constraint. This leads to a technique in which an n-equation problem is treated by optimizing in successively larger (more constrained) subproblems. This technique is described in Section 3. The Dual Method is used to achieve optimization in each subproblem. Each time it is achieved for a subproblem, another constraint is brought in by our method, and the Dual Method is again applied. The constraints are introduced in such a way as to obtain a better bound on the value of the total n-equation problem (like an iteration of the Dual Method, the step by which the constraint is introduced drives the current value toward the optimal one). In the first section we develop the geometric significance of our method of going from one e.p. feasible dual solution to another while adding constraints. In the second section we derive the result algebraically.

1. GEOMETRIC MOTIVATION

For concreteness, we shall always deal with a problem requiring minimization. Thus, in the dual problem we shall

*In terms of (1) below, an extreme point feasible dual solution may be considered to be a set of numbers $(\pi_1, \dots, \pi_m)$ such that for m indices, $i_1, \dots, i_m$, we have

$$(\pi_1, \dots, \pi_m) \begin{pmatrix} a_{1i_k} \\ \vdots \\ a_{mi_k} \end{pmatrix} - c_{i_k} = 0 \quad k = 1, 2, \dots, m,$$

while for the other $n - m$ indices, the above equation holds with "less than or equal" replacing the equality sign.

always maximize. Our method requires that the primal problem be in equation (as opposed to inequality) form. This is no restriction, since "slacks" can always be added to a problem to put it in that form. Hence we take the problem to be that of (1).

$$\begin{aligned}
 & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \quad x_i \geq 0, \quad i = 1, 2, \dots, n \\
 & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
 & \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
 & \min \quad c_1x_1 + c_2x_2 + \cdots + c_nx_n
 \end{aligned}
 \tag{1}$$

We define the "kth subproblem" or "k-problem," $1 \leq k \leq n$, to be the linear programming problem which arises from (1) by removing the last $m-k$ constraints. We call an e.p. feasible dual solution to the kth subproblem a "kth solution."

The method may now be indicated. Given a kth solution, we attempt to find a $k+1$ st solution in the following way. Since the kth solution corresponds to a k -column basis (or k th basis) and the $k+1$ st solution will certainly require a $k+1$ -column basis, we attempt to find a single column which, added to the columns of the kth solution, will provide a $k+1$ st solution. If we assume for concreteness that the first k columns of (1) were in the basis of the kth solution, we may then be said to search for a column i , $i > k$, which will make the basis, B_{k+1} , indicated in (2) correspond to a $k+1$ st

$$(2) \quad B_{k+1} = \left(\begin{array}{cccc|c} a_{11}, & a_{12}, & \cdots, & a_{1k}, & a_{11} \\ a_{21}, & a_{22}, & \cdots, & a_{2k}, & a_{21} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{k1}, & a_{k2}, & \cdots, & a_{kk}, & a_{k1} \\ \hline a_{k+1,1}, & a_{k+1,2}, & \cdots, & a_{k+1,k}, & a_{k+1,1} \end{array} \right)$$

solution. In fact, there will always exist such an index i (assuming there exists a solution to (1)). Geometrically this can be appreciated as follows: Consider a 2nd solution to (1), and let $m = 4$ in that problem. This 2nd solution corresponds to an extreme point of the convex polyhedron indicated in Fig. 1. The two lines on which the point lies correspond to the columns of the 2nd basis (for definiteness we take these to be the first and second columns) and the point is on the appropriate side (determined by the inequality

$$\sum_{i=1}^2 \pi_i a_{ij} \leq c_j$$

and indicated by an arrow) of each of the other lines. When the third equation is added to the problem, the lines become planes and in particular the lines

$$\pi_1 a_{11} + \pi_2 a_{21} = c_1$$

$$\pi_1 a_{12} + \pi_2 a_{22} = c_2$$

of our 2nd basis become the planes

$$\pi_1 a_{11} + \pi_2 a_{21} + \pi_3 a_{31} = c_1$$

$$\pi_1 a_{12} + \pi_2 a_{22} + \pi_3 a_{32} = c_2.$$

The fact that we have decided to leave the two columns of the 2nd basis in the 3rd basis indicates that our 3rd solution will lie somewhere on the intersection of the two planes which the first two columns now determine. Our second solution is clearly of this type but, though feasible, it is no longer an extreme point. Hence we move this point up or down along the intersection of the two planes (we discuss later whether up or down) until we meet a third plane. (Our use of the word "move" is purely rhetorical since the change will not require a search but only a single decision.) This determines an extreme point and becomes our 3rd solution.

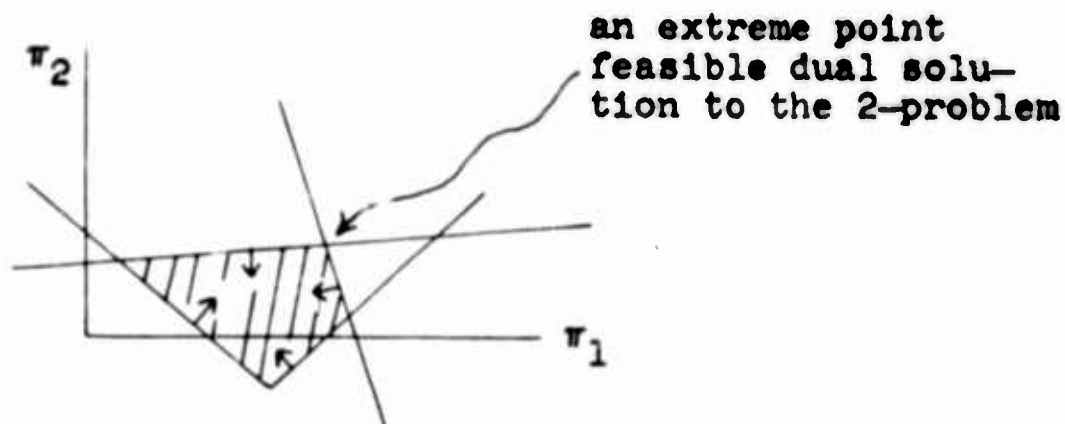


Fig. 1

2. ALGEBRAIC DERIVATION

We assume that we have a k th basis (corresponding to a k th solution) and that we are looking for an index i which will determine a $k+1$ st basis of the form assumed in (2). The dual variables corresponding to the k th basis are

$$(3) \quad (\pi_1, \pi_2, \dots, \pi_k) = (c_1, c_2, \dots, c_k) B_k^{-1}$$

and by assumption

$$(4) \quad (c_1, c_2, \dots, c_k) (B_k^{-1}) \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{kj} \end{pmatrix} - c_j \leq 0 \quad j = 1, 2, \dots, n.$$

We wish to choose an index 1 which satisfies the condition analogous to (4) for the $k+1$ st stage, or

$$(5) \quad (c_1, c_2, \dots, c_k, c_1) \begin{pmatrix} a_{11}, & \dots, a_{1k}, & a_{11} \\ \vdots & & \vdots \\ a_{k1}, & \dots, a_{kk}, & a_{k1} \\ a_{k+1,1}, & \dots, a_{k+1,k}, & a_{k+1,1} \end{pmatrix}^{-1} \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{kj} \\ a_{k+1,j} \end{pmatrix} - c_j \leq 0$$

$j = 1, 2, \dots, n.$

By straightforward calculation the reader may verify that the inverted matrix involved in (5), denoted B_{k+1}^{-1} , is related to B_k^{-1} as indicated in (6).

$$(6) \quad B_{k+1}^{-1} = \left(\begin{array}{c|c} X & \begin{matrix} y_1 \\ \vdots \\ y_k \end{matrix} \\ \hline z_1, \dots, z_k & t \end{array} \right) \quad X = B_k^{-1} - (B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} (z_1, \dots, z_k)$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} = -t(B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} \quad (z_1, \dots, z_k) = -t(a_{k+1,1}, \dots, a_{k+1,k})(B_k^{-1})$$

$$t = - \frac{1}{(a_{k+1,1}, \dots, a_{k+1,k})(B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} - a_{k+1,1}}$$

Applying (6), we can write (5) in the form

$$(c_1, \dots, c_k, c_1) \left[\left(\begin{array}{c|c} x & 0 \\ \hline 0, \dots, 0 & 0 \end{array} \right) + \left(\begin{array}{c|c} 0 & y_1 \\ \hline z_1, \dots, z_k & y_k \end{array} \right) + \left(\begin{array}{c|c} 0 & 0 \\ \hline 0, \dots, 0 & t \end{array} \right) \right] \begin{pmatrix} a_{1j} \\ \vdots \\ a_{k+1,j} \end{pmatrix} - c_j \leq 0.$$

Distributing the matrix multiplication over the three interior matrices and collecting the $-c_j$ with the first term, we get the following three summands, after substituting some of the values given in (6):

$$(8) \quad \left[(c_1, \dots, c_k)(B_k^{-1}) \begin{pmatrix} a_{1j} \\ \vdots \\ a_{kj} \end{pmatrix} - c_j \right] - (c_1, \dots, c_k)(B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} (z_1, \dots, z_k)$$

$$(9) \quad c_1(z_1, \dots, z_k) \begin{pmatrix} a_{1j} \\ \vdots \\ a_{kj} \end{pmatrix} - ta_{k+1,j}(c_1, \dots, c_k)(B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix}$$

$$(10) \quad ta_{k+1,j}c_1.$$

Combining (8), (9), and (10) and then substituting the values for $(z_1, z_2, \dots, z_k)$ and t given in (6), we have this equivalent

form of (7):

$$(11) \quad \left[(c_1, \dots, c_k) (B_k^{-1}) \begin{pmatrix} a_{1j} \\ \vdots \\ a_{kj} \end{pmatrix} - c_j \right] \\ - \frac{\left[(a_{k+1,1}, \dots, a_{k+1,k}) (B_k^{-1}) \begin{pmatrix} a_{1j} \\ \vdots \\ a_{kj} \end{pmatrix} - a_{k+1,j} \right]}{\left[(a_{k+1,1}, \dots, a_{k+1,k}) (B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} - a_{k+1,1} \right]} \left[(c_1, \dots, c_k) (B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix} - c_1 \right]$$

≤ 0 .

Thus the condition to be satisfied is of the form

$$(12) \quad \bar{c}_j \leq \frac{\bar{a}_j}{\bar{a}_1} \bar{c}_1 \quad j = 1, 2, \dots, n,$$

where the $\bar{c}_j$ are relative cost factors for the k th subproblem and the $\bar{a}_j$, $j = 1, \dots, n$, are the numbers which would be the relative cost factors for the k th subproblem if the coefficients of the next equation were taken as the cost form for the problem. Since B_k is a k th-basis, the $\bar{c}_j$, $j = 1, \dots, n$, are all non-positive. Using this fact, the reader can verify that if there exists an $\bar{a}_1 > 0$, then an index i_1 for which

$$(13) \quad \bar{a}_{i_1} > 0 \quad \text{and} \quad \frac{\bar{c}_j}{\bar{a}_j} \leq \frac{\bar{c}_{i_1}}{\bar{a}_{i_1}} \quad \text{for all } j \text{ such that } \bar{a}_j > 0$$

will satisfy (12). Similarly, if there exists an $\bar{a}_1 < 0$, then an index 1_2 for which

$$(14) \quad \bar{a}_{1_2} < 0 \quad \text{and} \quad \frac{\bar{c}_j}{\bar{a}_j} \geq \frac{\bar{c}_{1_2}}{\bar{a}_{1_2}} \quad \text{for all } j \text{ such that } \bar{a}_j < 0$$

will satisfy (12). We are precluded from choosing an index 1 for which $\bar{a}_1 = 0$ since (12) would not be defined. This corresponds to the fact that the addition of such a column makes B_{k+1} a singular matrix (without the necessary inverse). With this understanding, it is easily verified that any index 1 for which (12) holds must satisfy either (13) or (14). We now consider the implications of a choice between (13) and (14).

If in (5) we delete c_j and let the column of b's in (1) play the role of the j th column in (5), the left-hand side of that equation will represent the value associated with the $k+1$ st solution which we denote V_{k+1} . Reasoning similarly with (4) we can interpret the reduction of (5) to (11) as indicating that

$$(15) \quad V_{k+1} = V_k - \frac{\bar{b}}{\bar{a}_1} \bar{c}_1$$

where $\bar{b}$ is defined analogously to $\bar{a}_j$. Since we are interested in maximizing in the dual problem, we evidently should ascertain the sign of $\bar{b}$ and choose our index 1 in such a way

that $\frac{\bar{b}}{\bar{a}_1}$ is positive. Assuming that $\bar{b} \neq 0$,* this is always possible, provided that the equations have a feasible primal solution.

The reasoning is as follows: Let us pretend that the $k+1$ st equation represents the cost form for the problem; $\bar{b}$ becomes the difference between the present value and the value which must be achieved to satisfy the $k+1$ st equation. The $\bar{a}_j$ become the relative cost factors. If $\bar{b} < 0$, and $\bar{a}_j \geq 0$, $j = 1, \dots, n$, we would have a feasible dual solution to a problem of maximization (in the primal) in which more "value" was required to satisfy equation $k+1$. Since the value associated with the feasible dual solution is an upper bound on the value which can be attained by a feasible primal solution, there can exist no feasible primal solution to the $k+1$ -problem (and hence no such solution to the complete m -problem). A similar argument applies to the case in which $\bar{b} > 0$ and $\bar{a}_j \leq 0$, $j = 1, \dots, n$. This gives the result and allows us to say that the bound on the final optimal solution attained by bringing in the equation is no worse (and usually better) than the previous one.**

* If $\bar{b} = 0$, the $k+1$ st equation is satisfied, along with the first k equations, by variables which correspond to a k -column basis. In this case any column may be brought in which satisfies $\bar{a}_1 \neq 0$. If no such column exists, the $k+1$ st equation is a linear combination of the previous k equations and can thus be ignored.

** Dr. George Dantzig has pointed out that the result in this section bears a close relationship to the Parametric Linear Programming of William Orchard-Hays [7]. Looked at

3. ALGORITHM PRESENTATION

Our method of introducing equations suggests an algorithm, which will be discussed in this section, for solving linear programming problems. We propose to solve the m -equation problem by the following steps: Beginning with an (e.p.) optimal solution to a one-equation subproblem, we will introduce another constraint. This can be done, because an e.p. optimal solution to a problem corresponds to an e.p. feasible dual solution. Our method derives an e.p. feasible dual solution to the 2-problem, and using the Dual Method we achieve an optimal solution to this subproblem. We are then in a position to introduce another constraint, giving a three-equation subproblem. We continue in this way until we have brought in the last equation and used the Dual Method to find an optimal

(cont'd from page 10)

in this light, the application of the method corresponds to moving the right-hand side of a $k+1$ st subproblem "parametrically" from

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \\ \bar{b} + b_{k+1} \end{pmatrix} \quad \text{to} \quad \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \\ b_{k+1} \end{pmatrix}$$

Our approach differs from Parametric Linear Programming in that we do not necessarily begin with an optimal situation and do not seek to maintain optimality. Further, the movement indicated above is carried out not "parametrically" but in one step.

solution to that ultimate subproblem.*

There is only one difficulty to be considered with respect to this procedure. It is possible for all of the m possible one-equation subproblems to fail to have an (e.p.) optimal solution although the entire m -equation problem is quite unexceptional (i.e., has optimal programs).** In this case we employ a trick due to Beale [1] and begin by considering a fictitious equation,

$$(16) \quad \sum_{i=1}^n x_i + t = M,$$

where M is a very large positive number (thought of as larger than the sum of the x_i for any feasible primal solution, though in fact there may be no such number). If we assign t a zero cost this 1-problem has an (e.p.) optimal primal solution,

*We could have proposed in place of this procedure that the dual of the original problem be considered the primal problem and the above procedure followed. This would lead to maintaining feasibility in the original problem and successively adding activities (columns). This important possibility is discussed in Section 5.

**An example is shown below. Each equation is devoid of feasible dual solutions as a 1-problem but together they are not. An optimal solution is $x_1 = 0$, $x_2 = 2$, $x_3 = 1$.

	x_1	x_2	x_3	
π_1	1	-1	0	= -2
π_2	1	-1	-1	= -3
min	-1	2	-1	

and hence we are in a position to initiate our procedure. The addition of (16) to the original problem does not diminish (hence does not change) the set of feasible primal solutions to the problem if this set is bounded. It does, however, ensure that the set of feasible dual solutions is nonempty, and hence, by the Duality Theorem [4], it guarantees the existence of a solution (which our algorithm will find) to the modified problem. If the dual variable associated with (16) ever becomes zero (as it usually will early in the procedure), (16) can be ignored thereafter. However, it is possible that in the optimal solution this dual variable is not zero. This indicates that the set of values of feasible primal solutions to the original problem is not bounded below (i.e., the original problem had an infinite solution).

4. EFFICIENCY IN COMPUTATION

We turn now to considering the efficiency of our proposed algorithm. If we were to introduce one equation after another without attempting to find optimality in each subproblem, we would eventually reach an e.p. feasible dual solution to the entire problem.* Thus our result would be similar to that attained by a phase one algorithm for the Dual Method. However, because we would have introduced constraints in such a way as to increase the value of the dual solution, we might expect to

*If the artificial equation is required, the e.p. feasible dual solution will be associated with a slightly more constrained problem.

be somewhat nearer the optimal solution than would be the case if we had followed a phase one procedure.* Neglecting this advantage, we might consider the introduction of equations in our algorithm to be analogous to a phase one technique. Hence we might compare the number of multiplications required to introduce the entire set of equations by our method with the number of multiplications required by some phase one procedure. The number of multiplications we require is, of course, not dependent upon whether we pause to optimize in the subproblems or just pass quickly from one to the next. Having made the above comparison, we will then go on to compare the number of multiplications which might be necessary to reach optimality in all the subproblems with the number usually required to perform a phase two (i.e., reach optimality, assuming feasibility) with the Dual or Simplex Method.

At the $k+1$ st stage, two sets of k dual variables must be computed in our algorithm, one set corresponding to the true cost form and one set corresponding to the pretense that the next equation is the cost form. This requires $2k^2$ multiplications

*Only two approaches to the problem of constructing a phase one procedure for the Dual Method are known to us. Lemke in [6] proposes that a Simplex Method phase two be performed on a problem which he describes. G. Dantzig in [3] suggests, instead of a phase one for the Dual Method, that the Dual Method be applied to a problem related to the original one for which an e.p. feasible dual solution is immediate and that the Simplex Method be used, after the Dual Method has found the optimal solution, to go from the optimal solution of the related problem to the optimal solution of the original problem.

in all. Applying these sets of dual variables to the $(n-k)$ columns not yet in the basis, in order to compute the $\bar{c}_j$ and $\bar{a}_j$, requires $2k(n-k)$ multiplications. Determination of $\bar{b}$ requires k more multiplications, and a number of ratios not exceeding $n-k$ must then be computed.* Hence the transition requires less than

$$(17) \quad 2k^2 + 2k(n-k) + k + n - k = 2nk + n$$

multiplications. Summing (17) as k goes from 0 to $m-1$ will indicate the number of multiplications necessary to bring in all m equations. Hence we have

$$(18) \quad \sum_{k=0}^{m-1} 2nk + n = 2n \frac{(m-1)m}{2} + mn = nm^2.$$

Counting nm multiplications for a Simplex or Dual Method iteration, we interpret this result as the equivalent of m iterations. It seems reasonable that no phase one procedure can be consistently faster than this. For example, it would require at least m iterations to remove m artificial vectors, if these were used in a phase one procedure. A phase one procedure that involved an auxiliary Simplex or Dual problem would also require at least m iterations most of the time,

*The ratios $\bar{c}_j/\bar{a}_j$ need only be computed when $\bar{a}_j$ has the sign of $\bar{b}$.

if we may judge by computing experience.* We conclude, therefore, that the introduction of equations by our algorithm compares favorably in speed with a phase one technique as well as having the advantage mentioned at the start of this discussion.

The comparison of a phase two technique with the Dual Method iterations necessary for our algorithm to reach optimality in every subproblem is best carried out in geometric language. Geometrically our proposal reduces to considering certain cross-sections of the polyhedron of feasible dual solutions. At the k th stage the cross-section considered lies in the coordinate "face" determined by $\pi_1 = 0$, $i = k+1, k+2, \dots, m$. (In other words, the cross-section considered is the one achieved by ignoring the last $m-k$ equations.) Our algorithm then uses the Dual Method to optimize in that cross-section. When no better bound on the optimal value can be achieved in the cross-section, another equation is introduced (this operation also improves the bound) and the Dual Method is again applied. The cross-section now lies in a coordinate face determined by $\pi_1 = 0$, $i = k+2, k+3, \dots, m$. In this "higher dimensional" cross-section our algorithm prohibits us from considering any "vertices" (of the cross-section) which are associated with lower values of the dual

*This statement is appropriate as a comment on Lemke's method for constructing a phase one procedure (see previous footnote). It is difficult to determine the effect of G. B. Dantzig's proposal.

than have been already achieved. Hence we consider a truncated cross-section. It is our expectation that achieving optimality for the previous subproblem and introducing the next equation as we do (so as to achieve a better bound) will truncate the next cross-section radically, so that few iterations will be required to regain optimality. This expectation is one-half of the argument that our algorithm provides a good alternative to a phase one technique. The other half of our argument centers on the fact that, whatever the number of iterations required to achieve optimality for each subproblem may be, these iterations often concern a relatively small basis and hence involve fewer multiplications than a Simplex or Dual Method iteration. For example, let us consider the possibility that it requires two Dual Method iterations at each stage to achieve optimality for the subproblem. In this case, counting kn multiplications for a Dual Method iteration in a k -equation problem we would require

$$\sum_{k=2}^m 2kn = \frac{2nm(m+1)}{2} - 2n = nm(m+1) - 2n$$

multiplications. This result is somewhat less than the equivalent of $m+1$ Simplex or Dual Method iterations. With the m iterations required to introduce the equations included, our entire algorithm would take the equivalent of approximately $2m$ Simplex iterations and would probably be about on a par computationally with the techniques in use.

It is obviously possible that it will take somewhat more

than two Dual Method iterations to achieve optimality for each subproblem. It is relevant to point out that the new variable x_1 introduced into the basic solution at the $k+1$ th stage always enters at a nonnegative level. It is derived in a straightforward manner to be

$$(19) \quad x_1 = \frac{\bar{b}}{\bar{a}_1}$$

which is nonnegative since $\bar{a}_1$ is always chosen to be of the same sign as $\bar{b}$. Of course the $k+1$ st solution may not be optimal when the k th solution is, because any of the first k variables may become negative. The new values of the first k variables (primed) have the relationship to the old values indicated in (20).

$$(20) \quad \begin{pmatrix} x_1' \\ \vdots \\ x_k' \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} - \frac{\bar{b}}{\bar{a}_1} (B_k^{-1}) \begin{pmatrix} a_{11} \\ \vdots \\ a_{k1} \end{pmatrix}$$

In any case, the nonnegativity of x_1 allows for the possibility that as the equations are brought in an optimal solution is maintained from the k th to the $k+1$ st stage (without the use of any Dual Method iterations). *

* A related question to which the answer is definitely in the negative asks whether there always exists some ordering of the equations which will allow them to be brought in one by one while the k th solutions remain optimal at each stage. A counter-example is given below. Depending upon which equation

5. ADVANTAGES IN APPLICATION

In this section we discuss the flexibility and advantages of the algorithm proposed. We begin by presenting three types of situations in which the algorithm would prove useful.

A. A firm has listed the constraints which, it feels, might apply in a given problem. It recognizes that if certain constraints prove too restraining in terms of the optimal value, they might be altered or removed by some "nonlinear action, such as acquisition of more land or plant facilities, etc. Our recommendation is to place these alterable constraints last, and to use the cross-section method to achieve an optimal solution to the subproblem P, consisting of the other k constraints, and thereafter, for each alterable equation, to alternate bringing in the equation by our method and reaching optimality in that k+1-equation subproblem by the Dual Method, in order to gauge the difference between the optimal value of P and the optimal value of the subproblem consisting of P with an alterable

(cont'd from page 18)

is considered initially, column one or column 4 is introduced into the basis. However, neither of these columns appears in the only optimal solution to the problem. That solution is $x_1 = 0$, $x_2 = 4/3$, $x_3 = 4/3$, $x_4 = 0$.

	x_1	x_2	x_3	x_4	
π_1	1	1/2	1/4	1/8	1
π_2	1/8	1/4	1/2	1	1
min	-1	-1	-1	-1	

equation added. This will indicate the restraining effect of any single alterable equation on the set of unalterable equations.

B. A firm has listed the inequalities which it feels might apply in a given problem but recognizes that most of them are probably not constraining in an optimal solution. Rather than work with the entire set of inequalities, as would be necessary by a straightforward application of the Simplex Method, the firm should place at the end those inequalities which, in its judgment, are thought to be unimportant and then proceed to apply the technique indicated in Section 3 (after adding slack vectors). A check on the rate at which the optimal values to successive subproblems decrease should provide a good indication of how restraining the latest equations are. When the values cease to decrease, the firm might want to assume that the remaining equations are not constraining. In this case it would have only to check the feasibility (in the other inequalities) of the optimal solution to the last subproblem considered. Assuming that the constraints could be satisfied, the firm would have an optimal solution, without the labor involved in handling an oversized matrix.

C. A firm might be interested in a bound on the optimal value of a given linear programming problem. It might feel that if the optimal value v were greater than some v_0 , it would be pointless to follow some course of action associated with the problem. In this case, the procedure advised in

Section 3 would give successively better lower bounds as each subproblem was considered. If the lower bound indicated by some subproblem surpassed v_0 , the firm would know that the course of action considered could not be followed unless at least one of the constraints involved in that optimal subproblem were altered or removed.

The advantages discussed so far exhibit the virtues of a method which adds equations one at a time while completely solving the induced subproblems. These advantages would have their counterpart in a method which adds activities (columns) one at a time while completely solving the subproblems which arise. In order to get such a method we have only to apply the procedure we recommended in Section 3 to the dual of the original problem. Such a procedure maintains feasibility of the intermediate solutions to the original problem and hence provides upper bounds (according to our conventions), as opposed to the lower bounds utilized in example C. Furthermore, it allows one to alter activities, with all the advantages analogous to those described in examples A and B for changing equations.

REFERENCES

1. Beale, E. M. L., "An Alternative Method for Linear Programming," Proceedings of the Cambridge Philosophical Society, Vol. 50, Part 4, 1954, pp. 513-523.
2. Dantzig, G. B., "Maximization of a Linear Function of Variables subject to Linear Inequalities," in T. C. Koopmans (ed.), Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, John Wiley and Sons, 1951.
3. Dantzig, G. B., Notes on Linear Programming: Part VII — The Dual Simplex Algorithm, The RAND Corporation, Research Memorandum RM-1270, July 3, 1954.
4. Gale, D., Kuhn, H. W., and Tucker, A. W., "Linear Programming and the Theory of Games," in T. C. Koopmans (ed.), Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, John Wiley and Sons, 1951.
5. Koopmans, T. C. (ed.), Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, John Wiley and Sons, 1951.
6. Lemke, C. E., "The Dual Method of Solving the Linear Programming Problem," Naval Research Logistics Quarterly, Vol. 1, No. 1, March, 1954.
7. Orchard-Hays, William, "Revisions and Extensions to the Simplex Method (with Sidelights on Programming Techniques)," Econometrica, July, 1955.