RESEARCH
ANALYSIS
CORPORATION

An Algorithm for a Special Class of Generalized Transportation-Type Problems

PRO CON
RAC

ADVANCED RESEARCH DEPARTMENT
TECHNICAL PAPER RAC-TP-337
Published November 1968

An Algorithm for a Special Class of Generalized Transportation-Type Problems

by
Ronald L. Arms

RESEARCH ANALYSIS CORPORATION
McLEAN, VIRGINIA

RAC

FOREWORD

This paper describes an algorithm developed for solving a special class of generalized transportation-type problems of moderate size. Problems concerned with optimal allocations of resources subject to meeting a given set of requirements such as marketing, routing, production, and weapons allocation are frequently of the generalized transportation type.

The generalized transportation-type problem considered here is a linear programming problem with solutions giving the allocations x_{ij} of the jth resource to the ith operation such as to maximize a given profit function. The requirements specify the limits on each of the n available resources as well as the operational limits of each of the m operations. In addition the operational capacity of the ith operation when the jth resource is assigned to it is known. The structure of such problems (one constraint for each row i and each column j) enables an algorithm more efficient than the general simplex algorithm to be used for finding a solution.

The algorithm is intended to solve moderate-sized problems faster than will general simplex algorithms. It requires less computer storage than general simplex algorithms, thus making it particularly useful when a limited-capacity computer memory is all that is available.

Nicholas M. Smith
Head, Advanced Research Department

ACKNOWLEDGMENTS

The author wishes to acknowledge the encouragement of Dr. Jerome Bracken and Dr. Frederick Miercort in developing the algorithm for application to part of RAC's Nike X study and to thank Drs. D. Gross, C. M. Harris, and J. D. Pearson for their reviews, criticisms, and suggestions, which have largely been incorporated in the present version of the paper.

CONTENTS

Appendixes

Table

An Algorithm
for a Special Class of Generalized
Transportation-Type Problems

RAC

ABSTRACT

The algorithm described in this paper is used to solve a special class of linear programming problems characterized by constraint coefficient matrices having generalized transportation structure. Specifically, n available resources are allocated to m capacity-limited operations (where the operational capability of assigning the jth resource to the ith operation is known) such as to maximize the total profit for the system. The row-and-column structure of such problems permits an algorithm more efficient than the general simplex algorithm to be used to solve moderate-sized problems (problems where loop-tracing techniques or equivalent schemes are not required). It is not required in the problem statement that all the resources be allocated or that all operations be performed to capacity limits. It is characteristic of such problems, however, that the optimizing solution usually requires that at least one of the two conditions holds, i.e., either supply or demand is exhausted. The paper contains a description of the algorithm, a computer program, an example illustrating its application, and some comparisons with the general simplex algorithm in solving the same problem.

1. INTRODUCTION

The algorithm presented here yields optimal solutions to a special class of linear programming problems that are characterized by constraint coefficient matrices having generalized transportation structure.† The algorithm preserves primal feasibility and the complementary slackness condition at all times; hence feasibility of the dual constraints forms a set of necessary and sufficient conditions for testing optimality.

The need for the present algorithm arose initially in application to an optimal weapons-allocation problem as part of a larger nonlinear minimax problem employed in an earlier RAC study[1] in this area.

A specialized algorithm (similar to the one given here) for generalized transportation-type problems appears to have been first used by Ferguson and Dantzig.[2,3]

The algorithm can be divided computationally into two phases: (1) the matrix maximum phase and (2) the simplex phase. In phase 1 the algorithm permits only vectors associated with positive cost to enter the basis and only basis vectors associated with slack variables to leave the basis. In phase 2 the selection of the next neighboring vertex is currently made as it is done in most simplex algorithms (see Ref 4, Lecture V and the appendix).

The particular structure of the constraint coefficient matrix permits economy of computation by employing the equivalent of a doubly indexed simplex algorithm.

2. PROBLEM STATEMENT

The algorithm presented in subsequent sections yields an optimal solution to the following class of linear programming problems.

†The details of this structure will be considered in Sec 3, "Problem Structure."

Maximize

$$\sum_{i,j}^{m,n} c'_{ij}x'_{ij} \qquad \text{with respect to} \qquad x'_{ij}$$

subject to the constraints

$$\begin{aligned} &\sum_{j=1}^{n} d'_{ij}x'_{ij} \leq a_i \ , \ a_i > 0 \ ; \ (i = 1, \ldots, m) \\ &\sum_{i=1}^{m} h'_{ij}x'_{ij} \leq b_j \ , \ b_j > 0 \ ; \ (j = 1, \ldots, n) \\ &\left.\begin{aligned} x'_{ij} &\geq 0 \\ c'_{ij} &\geq 0 \\ d'_{ij}, h'_{ij} &> 0 \end{aligned}\right\} \begin{aligned} &(i = 1, \ldots, m) \\ &(j = 1, \ldots, n) \end{aligned} \end{aligned} \tag{1}$$

Under the correspondences

$$\begin{aligned} x_{ij} &= h'_{ij}x'_{ij} \\ d_{ij} &= d'_{ij}/h'_{ij} \\ c_{ij} &= c'_{ij}/h'_{ij} \end{aligned} \tag{2}$$

an optimal solution to Prob 1 can be found from finding an optimal solution to Prob 3.

Maximize

$$\sum_{i,j}^{m,n} c_{ij}x_{ij} \qquad \text{with respect to} \qquad x_{ij}$$

subject to the constraints

$$\begin{aligned} &\sum_{j=1}^{n} d_{ij}x_{ij} \leq a_i \ , \ a_i > 0 \ ; \ (i = 1, \ldots, m) \\ &\sum_{i=1}^{m} x_{ij} \leq b_j \ , \ b_j > 0 \ ; \ (j = 1, \ldots, n) \\ &\left.\begin{aligned} x_{ij} &\geq 0 \\ c_{ij} &\geq 0 \\ d_{ij} &> 0 \end{aligned}\right\} \begin{aligned} &(i = 1, \ldots, m) \\ &(j = 1, \ldots, n) \end{aligned} \end{aligned} \tag{3}$$

The algorithm finds an optimal solution to Prob 3.

It should be observed that both the row and column constraints are inequalities. It is characteristic of such problems that the optimizing solution has the property that either all the row constraints, or all the column constraints, or both all row and all column constraints are binding when all $c_{ij} > 0$. If equalities are imposed on the column constraints and the row inequalities are of either type, we have the generalized transportation problem considered by Hadley.[5]

If both row and column constraints are equalities, $\Sigma a_i = \Sigma b_j$, and $d_{ij} = 1$ for all i, j, the problem reduces to the standard transportation problem.

3. PROBLEM STRUCTURE: GENERAL DISCUSSION

The general simplex algorithm may be used to solve Probs 1 or 3. For large m and n, however, it is not practical to do so. Writing the components x_{ij} of $x \in E^{mn}$ using a single component subscript index k for x_k (as is done when using the general simplex algorithm), we see that the constraint matrix A contains $mn(m + n - 2)$ zeros.

If g is the component subscript indexing function $[g(i, j) = k]$ for the vector x, then for the problem

$$\begin{array}{lll} \max\limits_{x} \langle c,x \rangle & x \in E^{mn} & c \in E^{mn} \\ Ax \le b & b \in E^{m+n} & b > 0 \\ x \ge 0 & & \end{array} \tag{4}$$

A assumes either of the two structures

$$A = \left(\begin{array}{c|c|c|c} d_{11} \cdots d_{1n} & & & \\ & d_{21} \cdots d_{2n} & & \\ & & \cdots & \\ & & & d_{m1} \cdots d_{mn} \\ \hline I_n & I_n & I_n & I_n \end{array} \right) \tag{5}$$

when $g(i,j) = n(i-1) + j = k \; (1 \le i \le m, \; 1 \le j \le n)$

where I_n is the identity matrix of order n or

$$A = \left(\begin{array}{c|c|c|c} 1_m & & & \\ & 1_m & & \\ & & \cdots & \\ & & & 1_m \\ \hline \begin{matrix} d_{11} & & \\ & d_{21} & \\ & & \ddots \\ & & & d_{mn} \end{matrix} & \begin{matrix} d_{12} & & \\ & d_{22} & \\ & & \ddots \\ & & & d_{m2} \end{matrix} & \ddots & \begin{matrix} d_{1n} & & \\ & \ddots & \\ & & d_{mn} \end{matrix} \end{array} \right) \tag{6}$$

when $g(i,j) = m(j-1) + i = k \; (1 \le j \le n, \; 1 \le i \le m)$

where 1_m is a row vector of m ones.

†The right-hand side column vector b here includes $m + n$ components (m a_i and n b_j) as in Prob 3.

Any other indexing by g (besides interchanging upper and lower blocks) produces a less uniform structure for A. Structure 5 for A is associated with generalized transportation-type problems. When all $d_{ij} = 1$ the structure of A in Structure 5 is that of coefficient matrices associated with transportation problems. Structure 5 for A will be assumed when introducing suitable basis vectors for the solution space later on.

4. OPTIMALITY CRITERIA

The algorithm preserves primal feasibility P and the complementary slackness condition S at all times and uses the feasibility of the dual-programming problem constraints D as the optimality test criterion.

The three sets of Conditions P, D, and S are explicitly

$$P: \begin{cases} Ax + I_{m+n} x_s = b \\ x \geq 0 \\ x_s \geq 0 \end{cases} \qquad D: \begin{cases} A'w - I_{mn} w_s = c \\ w_s \geq 0 \\ w \geq 0 \end{cases} \tag{7}$$

$$S: \langle w, x_s \rangle + \langle w_s, x \rangle = 0$$

where $'$ denotes transposition
$\langle \, , \rangle$ denotes inner product
$b > 0$
$c > 0$

See Ref 6, Pt 2, p 58, for a discussion of Conditions 7.

Real vectors x, w that satisfy Conditions P, D, and S also solve the pair of dual linear programming problems

$$\max_x \ \langle c, x \rangle \quad \text{subject to P} \tag{8a}$$

$$\min_w \ \langle w, b \rangle \quad \text{subject to D} \tag{8b}$$

Problem 8a is solved with Conditions P and S always holding, hence Condition D becomes the set of necessary and sufficient conditions for optimality.

In practice the algorithm enforces the following stronger form $\bar{S}$ of Condition S, namely,

$$\langle w, x_s \rangle + \langle w_s, x \rangle = 0 \quad \underline{\text{at the component level,}} \tag{7$\bar{S}$}$$

i.e., $x_{ij} > 0 \Rightarrow w_{s_{ij}} = 0$; $x_{s_i} > 0 \Rightarrow w_i = 0$. Since the dual space constraint $A'w - I_{mn}w_s = c$ holds for all w, w_s throughout the algorithm, Conditions $7\bar{D}$ become the set of necessary and sufficient conditions for optimality, i.e.,

$$w_s \geq 0 \; , \; w \geq 0 \qquad (7\bar{D})$$

5. PROCESSES OF THE ALGORITHM

The general processes of the algorithm, the details of which will follow, are

(1) Generate basic primal feasible solution using complete or partial matrix maximum (Conditions 7P are satisfied by exactly $m + n$ positive primal variables x, x_s while the primal objective function is increased).

(2) Solve for the dual space variables u_i and v_j using Condition $7\bar{S}$ and knowledge of the structure of basis vectors associated with the positive primal variables.

(3) Perform optimality test (test Conditions $7\bar{D}$). If there are no violations the current basic primal solution is optimal.

(4) For nonoptimal solutions find the largest violation of Conditions $7\bar{D}$. Identify the associated vector for entry into the basis for the primal solution space.

(5) Find the representation of the entering vector in terms of vectors in current basis.

(6) Preserving primal feasibility Conditions P, select vector to leave the current basis.

(7) Express the solution in terms of the new basis.

(8) Return to step 2.

6. DETAILS OF THE ALGORITHM

The detailed description of the steps of the algorithm is presented here.

1. Generation of Basic Primal Feasible Solutions

The matrix maximum method of generating solutions x_{ij} is a process that makes allocations (assigns values to x_{ij}) to payoff elements c_{ij} of a matrix $\bar{P}$ of payoffs as follows:

Let

$$\bar{P} = \{c_{ij} | x_{in'} \neq 0 \text{ and } x_{m'j} \neq 0\} \qquad \begin{array}{l} (i = 1, \ldots, n) \\ (j = 1, \ldots, m) \\ n' = n + 1 \\ m' = m + 1 \end{array} \tag{9}$$

where

$$x_{in'} = a_i - \sum_{j=1}^{n} d_{ij} x_{ij}$$

$$x_{m'j} = b_j - \sum_{i=1}^{m} x_{ij}$$

$x_{in'}$ and $x_{m'j}$ represent the residual "slack" in row i and column j (Prob 3) after a set $\{x_{ij}\}$ of allocations has been made. In Conditions 7 x_s = $(x_{1n'}, \ldots, x_{mn'}, x_{m'1}, \ldots, x_{m'n})'$. If either $x_{in'}$ or $x_{m'j}$ = 0 then no further allocations involving row i or column j can be made since either the ith capacity has been achieved or the jth resource exhausted. Initially $\bar{P}$ is the matrix of all c_{ij} since $x_{in'} = a_i$ and $x_{m'j} = b_j$.

For each allocation x_{ij} of the matrix maximum method let

$$c_{k\ell} = \max_{ij} \{c_{ij} | c_{ij} \in \bar{P}\} \tag{10}$$

then choose

$$x_{k\ell} = \min \left\{ \frac{a_k - \sum_{j=1}^{m} d_{kj} x_{kj}}{d_{k\ell}} \; ; \; b_\ell - \sum_{i=1}^{n} x_{i\ell} \right\} = \min \left\{ \frac{x_{km'}}{d_{k\ell}} \; ; \; x_{m'\ell} \right\} \tag{11}$$

This choice for the value assigned to $x_{k\ell}$ eliminates either row k or column ℓ from the matrix $\bar{P}$ of payoffs for the next iteration. The new values $x'_{kn'}$, $x'_{m'\ell}$ for $x_{kn'}$ and $x_{m'\ell}$ are found as follows:

If

$$x_{k\ell} = \frac{x_{kn'}}{d_{k\ell}}$$

then

$$x'_{kn'} = x_{kn'} - d_{k\ell} x_{k\ell} = 0 \tag{12}$$

$$x'_{m'\ell} = x_{m'\ell} - x_{k\ell}$$

If

$$x_{k\ell} = x_{m'\ell}$$

then (13)

$$x'_{kn'} = x_{kn'} - d_{k\ell}x_{m'\ell}$$

$$x'_{m'\ell} = x_{m'\ell} - x_{k\ell} = 0$$

In the first case $\overline{P}_{new} = \overline{P} - \{c_{kj} | x'_{kn'} = 0\}$ $(j = 1, \ldots, n)$. In the second case $\overline{P}_{new} = \overline{P} - \{c_{i\ell} | x'_{m'\ell} = 0\}$ $(i = 1, \ldots, m)$.

If $\frac{x_{kn'}}{d_{k\ell}} = x_{m'\ell}$ then an arbitrary decision is made to perturb $x_{kn'}$ by a small amount epsilon.

The matrix maximum procedure can be terminated in either of two ways, by exhausting the matrix $\overline{P}$ of payoffs (complete matrix maximum) or by assigning a fixed number (less than the number of iterations required to exhaust the matrix of payoffs) of positive allocations x_{ij} to be made (partial matrix maximum).

Throughout the matrix maximum iterations exactly $m + n$ elements of the vector $\overline{X}$ of allocations $(x_{ij}, x_{in'}, x_{m'j})$ are positive. The vector $\overline{X}$ satisfies Conditions 7P for primal feasibility.

The matrix maximum procedure proceeds from a vertex of the solution space to a neighboring vertex as does the simplex procedure, but specifically it proceeds to the vertex that has one less positive slack component and one more positive nonslack component (i.e., component having positive cost c_{ij}); hence the former is more efficient using a per iteration comparison. The matrix maximum procedure is not sufficient, however, to achieve optimality in general.

2. Solving for Dual Space Variables

The vector $\overline{X}$, resulting from application of the matrix maximum process (partial or complete), is a candidate optimizing point since it is an extreme point (Ref 4, p 58) of the convex set K of points $(x,x_s)'$ satisfying Conditions 7P

$$Ax + I_{m+n}x_s = b, \quad x \geq 0, \; x_s \geq 0 \qquad (7P)$$

The linearly independent set (a basis) of $m + n$ vectors corresponding to $X^{o} = (x^{o}, x_s^{o})'$ (the subvector of positive components of $\bar{X}$) is defined as follows:

$$\begin{array}{lll} \text{If } x_{ij} > 0 \text{ then } d_{ij}\vec{e}_i + \vec{e}_{m+j}{}^{\dagger} & \text{is a member of the basis} & \\ \text{If } x_{in'} > 0 \text{ then } \vec{e}_i & \text{is a member of the basis} & (14) \\ \text{If } x_{m'j} > 0 \text{ then } \vec{e}_{m+j} & \text{is a member of the basis} & \end{array}$$

Recall that in the matrix maximum process if $x_{ij} > 0$ then not both $x_{in'} > 0$ and $x_{m'j} > 0$; hence if $d_{ij}\vec{e}_i + \vec{e}_{m+j}$ is a basis vector then not both $\vec{e}_i$ and $\vec{e}_{m+j}$ are basis vectors. Conversely, if both $x_{in'} > 0$ and $x_{m'j} > 0$ then $x_{ij} = 0$; hence if $\vec{e}_i$ and $\vec{e}_{m+j}$ are basis vectors then $d_{ij}\vec{e}_i + \vec{e}_{m+j}$ is not. The set of $m + n$ column vectors selected from the matrix (A, I_{m+n}) using Definition 14 and denoted by B (the ordered matrix of such column vectors) is thus linearly independent and satisfies the condition

$$BX^{o} = b$$

Hence $\bar{X}$ is an extreme point of K.

Corresponding to X^{o} satisfying the equation $BX^{o} = b$ is a vector w^{o} satisfying the equation $B'w^{o} = c^{o}$ where c^{o} is the vector of costs (payoffs) associated with X^{o}. If the $m + n$ components of w are written $w = (u_1, \ldots, u_m, v_1, \ldots, v_n)'$ then the scalar form of the equation $B'w = c$ or $w'B = c'$ is

$$\begin{array}{lll} d_{ij}u_i + v_j = c_{ij}^{o} & \text{if } x_{ij} > 0 & (m+n) \text{ equations} \\ u_i = 0 & \text{if } x_{in'} > 0 & \qquad (15) \\ v_j = 0 & \text{if } x_{m'j} > 0 & \end{array}$$

Solutions to Eqs 15 satisfy Condition 7S, $\langle w, x_s \rangle + \langle w_s, x \rangle = 0$. Since $x_{in'} > 0$ implies $u_i = 0$ and $x_{m'j} > 0$ implies $v_j = 0$ then $\langle w, x_s \rangle = 0$. Similarly, if $x_{ij} > 0$ implies $w_{s_{ij}} = d_{ij}u_i + v_j - c_{ij} = 0$ then $\langle w_s, x \rangle = 0$.

3. Optimality Test (Testing Conditions 7$\bar{D}$)

The set of necessary and sufficient Conditions 7$\bar{D}$ required for optimality of X^{o} is rewritten here for reference.

† $\vec{e}_i$ here is a unit column vector in E^{m+n}.
$\vec{e}_i = (0, 0, \ldots, 1, 0, \ldots, 0)'$.
↑ ith component

$$A'w - I_{mn}w_s = c$$

$$\left.\begin{aligned} w_s &\geq 0 \\ w &\geq 0 \end{aligned}\right\} (7\bar{D}) \qquad \textbf{(7D)}$$

The scalar form of Conditions 7D is

$$d_{ij}u_i + v_j - w_{s_{ij}} = c_{ij} \quad | \quad mn \text{ equations}$$

$$\left.\begin{aligned} w_{s_{ij}} &\geq 0 \quad | \quad mn \text{ inequalities} \\ u_i \geq 0 \,,\; v_j &\geq 0 \quad | \quad m + n \text{ inequalities} \end{aligned}\right\} (7\bar{D}) \qquad \textbf{(16)}$$

Let $(u_1^o, \ldots, u_m^o, v_1^o, \ldots, v_n^o)'$ be the solution to Eqs 15. Since Eq 16 must hold for optimality we must have $w_{s_{ij}}^o \geq 0, u_i^o \geq 0, v_j^o \geq 0$ [$(mn + m + n)$ inequalities] where

$$w_{s_{ij}}^o = d_{ij}u_i^o + v_j^o - c_{ij} \qquad \textbf{(17)}$$

If $w_{s_{ij}}^o \geq 0, u_i^o \geq 0, v_j^o \geq 0$ for all i,j then X^o is optimal and the algorithm is terminated. If, however, $w_{s_{ij}}^o < 0$ for some i,j or $u_i^o < 0$ or $v_j^o < 0$ for some i or j, an improvement in the solution X^o can be made.

4. Nonoptimal Solutions; Finding the Greatest Violation of the Dual Space Constraints; Identifying the Associated Vector for Entry into the Primal Solution Space Basis

The greatest violation, V, of the dual space constraints (Conditions $7\bar{D}$) is simply

$$V = \min \left\{ \min_{i,j} \{ w_{s_{ij}}^o \mid w_{s_{ij}}^o < 0 \} ; \min_i \{ u_i^o \mid u_i^o < 0 \} ; \min_j \{ v_j^o \mid v_j^o < 0 \} \right\} \qquad \textbf{(18)}$$

Depending on which of the above three bracketed minimums is largest in magnitude, the corresponding vector chosen to enter the new basis is one of the three types of vectors $d_{ij}\vec{e}_i + \vec{e}_{m+j}$, $\vec{e}_i$, or $\vec{e}_{m+j}$.

5. Finding the Representation of the New Basis Vector in Terms of the Current Basis Vectors

Consider the three cases (a) $V = w_{s_{\bar{i}\bar{j}}}^o$, (b) $V = u_{\bar{i}}^o$, (c) $V = v_{\bar{j}}^o$ that can result from Eq 18.† The vector equation to be solved for a singly indexed system is

$$\vec{A}_k = By_k \quad \text{or} \quad y_k = B^{-1}\vec{A}_k \qquad \textbf{(19)}$$

†The bar denotes the minimizing index or indexes in Eq 18.

where y_k is the vector of coordinates of $\vec{A}_k$ relative to the basis of column vectors of B.

Corresponding to cases a, b, or c the following vector equation is solved for $y_{ij}^{\bar{i}\bar{j}}$, the $m + n$ components of the entering basis vector.

$$\left.\begin{array}{ll} \text{(a)} & d_{\bar{i}\bar{j}}\vec{e}_{\bar{i}} + \vec{e}_{m+\bar{j}} \\ \text{(b)} & e_{\bar{i}} \\ \text{(c)} & \vec{e}_{m+\bar{j}} \end{array}\right\} = \sum_{\substack{i,j \\ x_{ij} > 0}} y_{ij}^{\bar{i}\bar{j}}(d_{ij}\vec{e}_i + \vec{e}_{m+j}) + \sum_{\substack{i \\ x_{in'} > 0}} y_{in'}^{\bar{i}\bar{j}}\vec{e}_i + \sum_{\substack{j \\ x_{m'j} > 0}} y_{m'j}^{\bar{i}\bar{i}}\vec{e}_{m+j} \qquad (20)$$

Equation 20 leads to the following three sets of scalar equations in $y_{ij}^{\bar{i}\bar{j}}$, $y_{ij}^{\bar{i}n'}$, or $y_{ij}^{m'\bar{j}}$:

For Eq 20a

$$\begin{array}{lll} \displaystyle\sum_{\substack{j \\ x_{ij} > 0}} y_{ij}^{\bar{i}\bar{j}} d_{ij} + y_{in'}^{\bar{i}\bar{j}} = d_{\bar{i}\bar{j}} & \text{if} \quad i = \bar{i} & \\ & & (i = 1, \ldots, m) \\ \displaystyle\sum_{\substack{j \\ x_{ij} > 0}} y_{ij}^{\bar{i}\bar{j}} d_{ij} = 0 & \text{if} \quad i \neq \bar{i} & \\ \displaystyle\sum_{\substack{i \\ x_{ij} > 0}} y_{ij}^{\bar{i}\bar{j}} + y_{m'j}^{\bar{i}\bar{j}} = 1 & \text{if} \quad j = \bar{j} & \\ & & (j = 1, \ldots, n) \\ \displaystyle\sum_{\substack{i \\ x_{ij} > 0}} y_{ij}^{\bar{i}\bar{j}} = 0 & \text{if} \quad j \neq \bar{j} & \end{array} \qquad (21)$$

For Eq 20b

$$\begin{array}{ll} \displaystyle\sum_{\substack{j \\ x_{ij} > 0}} y_{ij}^{\bar{i}n'} d_{ij} = \begin{cases} 1 & \text{if} \quad i = \bar{i} \\ 0 & \text{if} \quad i \neq \bar{i} \end{cases} & (i = 1, \ldots, m) \\ \displaystyle\sum_{\substack{i \\ x_{ij} > 0}} y_{ij}^{\bar{i}n'} = 0 & (j = 1, \ldots, n) \end{array} \qquad (22)$$

For Eq 20c

$$\begin{array}{ll} \displaystyle\sum_{\substack{j \\ x_{ij} > 0}} y_{ij}^{m'\bar{j}} d_{ij} = 0 & (i = 1, \ldots, m) \\ \displaystyle\sum_{\substack{i \\ x_{ij} > 0}} y_{ij}^{m'\bar{j}} = \begin{cases} 1 & \text{if} \quad j = \bar{j} \\ 0 & \text{if} \quad j \neq \bar{j} \end{cases} & (j = 1, \ldots, n) \end{array} \qquad (23)$$

6. Selecting the Vector To Leave the Current Basis

Once the vector that enters the new basis has been found, the associated positive components of the new primal solution X^{o}_{new} must also satisfy Conditions 7P for primal feasibility. Hence we have

$$BX^{o} = B_{new}X^{o}_{new} = b \qquad X^{o}_{new} > 0 \tag{24}$$

or

$$BX^{o} - \theta \begin{Bmatrix} (a) & d_{\bar{i}\bar{j}}\vec{e}_{\bar{i}} + \vec{e}_{m+\bar{j}} \\ (b) & \vec{e}_{\bar{i}} \\ (c) & \vec{e}_{m+\bar{j}} \end{Bmatrix} + \theta \begin{Bmatrix} (a) & d_{\bar{i}\bar{j}}\vec{e}_{\bar{i}} + \vec{e}_{m+\bar{j}} \\ (b) & \vec{e}_{\bar{i}} \\ (c) & \vec{e}_{m+\bar{j}} \end{Bmatrix} = b \tag{25}$$

$$= BX^{o} - \theta \begin{Bmatrix} (a) & By^{\bar{i}\bar{j}} \\ (b) & By^{\bar{i}n'} \\ (c) & By^{m'\bar{j}} \end{Bmatrix} + \theta \begin{Bmatrix} (a) & d_{\bar{i}\bar{j}}\vec{e}_{\bar{i}} + \vec{e}_{m+\bar{j}} \\ (b) & \vec{e}_{\bar{i}} \\ (c) & \vec{e}_{m+\bar{j}} \end{Bmatrix} = b \tag{26}$$

$$= B \left\{ X^{o} - \begin{matrix} (a) & \theta y^{\bar{i}\bar{j}} \\ (b) & \theta y^{\bar{i}n'} \\ (c) & \theta y^{m'\bar{j}} \end{matrix} \right\} + \theta \begin{Bmatrix} (a) & d_{\bar{i}\bar{j}}\vec{e}_{\bar{i}} + \vec{e}_{m+\bar{j}} \\ (b) & \vec{e}_{\bar{i}} \\ (c) & \vec{e}_{m+\bar{j}} \end{Bmatrix} = b \tag{27}$$

Since $X^{o}_{new} > 0$ we have in particular (a) $x^{o}_{\bar{i}\bar{j}} = \theta > 0$, or (b) $x^{o}_{\bar{i}n'} = \theta > 0$, or (c) $x^{o}_{m'\bar{j}} = \theta > 0$ corresponding to the new basis vector a, b, or c. The remaining $m + n - 1$ column vectors of B_{new} are determined by eliminating that column vector of B whose new associated primal solution component $x^{o}_{ij_{new}}$ vanishes.

This elimination is accomplished as follows. Writing the expressions in the left braces of Eq 27 in component form, we have

$$x^{o}_{ij_{new}} = \left\{ x^{o}_{ij} - \begin{matrix} (a) & \theta y^{\bar{i}\bar{j}}_{ij} \\ (b) & \theta y^{\bar{i}n}_{ij} \\ (c) & \theta y^{m'\bar{j}}_{ij} \end{matrix} \right\} \quad \text{for all } (i,j) \ni x_{ij} > 0 \tag{28a}$$

$$x^{o}_{in'_{new}} = x^{o}_{in'} - \theta y^{\bar{i}\bar{j}}_{in'} \qquad \text{for all } (i,n') \ni x_{in'} > 0 \tag{28b}$$

$$x^{o}_{m'j_{new}} = x^{o}_{m'j} - \theta y^{\bar{i}\bar{j}}_{m'j} \qquad \text{for all } (m',j) \ni x_{m'j} > 0 \tag{28c}$$

Since we want $x^{o}_{ij_{new}}$, or $x_{in'_{new}}$, or $x_{m'j_{new}}$ to vanish we select positive θ from Eq 29

$$\hat{\theta} = \min_{ij} \left\{ \frac{x^{o}_{ij}}{y_{\bar{i}\bar{j}}^{ij}} \,\middle|\, y^{ij}_{\bar{i}\bar{j}} > 0 \right\} \quad \begin{matrix} (i = 1, \ldots, m') \\ (j = 1, \ldots, n') \end{matrix} \tag{29}$$

If the minimizing indexes in Eq 29 are $(i,j) = (p,q)$ then $x^{o}_{pq_{new}} = 0$ and $d_{pq}\vec{e}_p + \vec{e}_{m+q}$ leaves the basis if $p \neq m'$ or $q \neq n'$, $\vec{e}_p$ leaves the basis if $a = n'$, and $\vec{e}_{m+q}$ leaves the basis if $p = m'$.

7. Expressing the Solution in Terms of the New Basis

The new solution X^{o}_{new} has components expressed by Eqs 28a to 28c with $\theta = \hat{\theta}$. In particular, as mentioned before, $x^{o}_{pq_{new}} = 0$ and $x^{o}_{\bar{i}\bar{j}} = \hat{\theta}$ for the primal variables associated with the leaving and entering vectors respectively.

8. Return to Step 2

Self-explanatory.

C. EFFECT OF NEW SOLUTIONS ON VALUE OF OBJECTIVE FUNCTION

There is associated with any violation of Conditions $7\overline{D}$ a new solution (X^{o}_{new}) to Conditions 7P that improves the value of the objective function $\langle c^{o}, X^{o}\rangle$ and at the same time eliminates the specific violation of $7\overline{D}$.

Recall from step 6 of the algorithm Conditions 7P are preserved when a new vector $\vec{A}_k$ enters the basis, thus

$$\begin{aligned} BX^{o} - \theta A_k + \theta\vec{A}_k &= b \\ BX^{o} - \theta By_k + \theta\vec{A}_k &= b \qquad \vec{A}_k = By_k \\ B(X^{o} - \theta y_k) + \theta\vec{A}_k &= b \end{aligned} \tag{30}$$

For the corresponding expression to the objective function value we have

$$\begin{aligned} &\langle c^{o},(X^{o} - \theta y_k)\rangle + \theta c_k \\ \text{or} \quad &\langle c^{o},X^{o}\rangle + \theta(c_k - \langle c^{o},y_k\rangle) \quad \text{(new objective function value)} \end{aligned} \tag{31}$$

The term $(c_k - \langle c^{o},y_k\rangle)$ corresponds to $(c_k - z_k)$ in general simplex notation and in the notation of this paper to (a) $-w_{s_k}$ for $1 \leq k \leq mn$ when $\vec{A}_k = d_{ij}\vec{e}_i + \vec{e}_{m+j}$, $k = n(i-1) + j$, or (b) $-u_i$ when $\vec{A}_k = \vec{e}_i$, $mn + k \leq mn + m$, or (c) $-v_j$ when $\vec{A}_k = \vec{e}_{m+j}$

RAC

$mn + m < k \le mn + m + n$. Thus for positive θ and any violation of Conditions 7$\overline{D}$, i.e., $w_{s_k} < 0$, $u_i < 0$, or $v_j < 0$, there is an associated improvement in the objective function value of magnitude (a) $-\theta w_{s_k}$, (b) $-\theta u_i$, or (c) $-\theta v_j$ when the vector (a) $\vec{A}_k$, (b) $\vec{e}_i$, or (c) $\vec{e}_{m+j}$ enters the new basis. Condition 7$\overline{S}$ guarantees that the violation will be eliminated for the next iteration.

Throughout the algorithm values for z_k ($z_k = \langle c^o, y_k \rangle$) are not computed using the y_k representation of $\vec{A}_k$ (i.e., the representation relative to basis vectors B), but from the relation $z_k = \langle w^o, \vec{A}_k \rangle$ which makes for greater efficiency in computation.

8. COMPUTATIONAL EXPERIENCE

The algorithm briefly called MATMAX was originally used to solve the linear subproblems described in Ref 1 with $m = 3$, $n = 4$. During the process of convergence to a single larger nonlinear programming problem solution to which the linear programming Prob 3 is only a constraint, it became necessary to solve the linear problems in the order of ten thousand times. The need for an algorithm faster than the standard simplex algorithm thus arose.

TABLE 1

Solution Times for MATMAX and Standard Simplex Algorithms[a,3]

Number of constraints		MATMAX, sec	Simplex,[7] sec	Simplex,[b] sec
m	n			
5	4	0.16	0.60	0.36
10	12	5.56	31.51	15.84
18	24	24.88	297.10	114.98

[a]Solution times are based on single precision operations in FORTRAN IV using the IBM 7044 computer.
[b]See App C.

The MATMAX algorithm has been compared for solution time with the simplex algorithm[7] and an even faster simplex algorithm given in App C. The A matrix (with identity) requires 19,988 = 42 × 474 storage locations for the $m = 18$, $n = 24$ simplex algorithm, thus limiting the size of "incore" comparison of the algorithms. Solution times for MATMAX and standard simplex algorithms are shown in Table 1.

The success of the algorithm currently depends on being able to solve the $m + n$ linear equations Eqs 15 in u_i and v_j sequentially, i.e., on solving for the nonzero u_i and v_j in terms of zero valued u_i and/or v_j.

If, however, it is not possible to solve the system of Eqs 15 sequentially during some iteration of the algorithm,an attempt is made to bypass the difficulty. The algorithm then attempts to proceed to the optimum by avoiding the particular vertex for which Eqs 15 could not be solved. Currently the algorithm returns to phase 1 (the matrix maximum phase),this time assigning one less x_{ij} having positive cost c_{ij} and one more positive slack variable (partial matrix maximum) than was assigned the prior solution of phase 1. Beginning with this solution in phase 2 (simplex) a new sequence of vertexes is generated for which the problem of nonsequential solvability of Eqs 15 is frequently avoided.

The above process has worked successfully on most problems of moderate size but has failed on one with $m = 30$, $n = 32$. In some situations more than one return to phase 1 may be required in order to find a sequence of vertexes for which Eqs 15 may be solved.

It is, of course, possible to solve Eqs 15 when sequential methods fail. However, the logic of loop-tracing techniques required in such situations is complex and is not currently employed. Alternative methods that use the sequential solvability of Eqs 15 as a secondary criterion for selecting the next neighboring vertex are under investigation.

APPENDIXES

Appendix A

COMPUTER PROGRAM FOR THE ALGORITHM

INTRODUCTION

A listing of the computer program for the algorithm follows. The algorithm has been used successfully on problems of moderate size (see Sec 8, "Computational Experience"). The comment cards identify appropriate subsections of the program as described in Sec 6, "Details of the Algorithm."

The success of the algorithm currently depends on being able to solve the $m + n$ linear equations Eqs 15 in u_i and v_j sequentially, i.e., on solving for the nonzero u_i and v_j in terms of zero valued u_i and/or v_j, as discussed in "Computational Experience."

Several working arrays are used for calculations; namely, the AE, ABASIS, YY, ABAR, BBAR, U, V, ISUM, and JSUM arrays.

The value of the objective function is printed out every kth iteration by setting IWRITE = k on the first input card. In addition, a detailed printout will be given every kth iteration by setting ITAB = 1 on the same card.

The computer program of App C accepts exactly the same input cards as the following program, with the exception as stated in App C.

The subroutine TODAY called twice in the program is used for timing purpose only. The general user should not call this subroutine.

PROGRAM

FORTRAN SOURCE LIST

```
ISN        SOURCE STATEMENT

  .  $IBFTC MATMAX
  1          DIMENSION CE(35,35),DE(36,36),AE(36,36),ABASIS(36,36),YY(36,36)
  2          DIMENSION ABAR(36),BBAR(36),ISUM(36),JSUM(36)
  3          DIMENSION AA(36),BB(36)
  4          DIMENSION XX(35,35),U(35),V(35)
  5  19001 FORMAT(1H1)
  6  19002 FORMAT(1H )
  7  19003 FORMAT(4I10)
 10  19004 FORMAT(5E12.6)
 11  19005 FORMAT(15X,6E16.6)
 12  19006 FORMAT(15X,4HMS= ,I2,4X,4HNS= ,I2,15X,30HFREQ. OF OBJ. FUNCT. PRIN
           1TOUT ,I2,9X,29HDETAILED PRINTOUT IF ITAB = 1,5X,5HITAB=,I2//)
 13  19007 FORMAT(30X,46HINPUT CONSTANTS FOR OPTIMAL ALLOCATION PROBLEM//)
 14  19008 FORMAT(15X,24HINPUT VALUES FOR CE(I,J)//)
 15  19009 FORMAT(15X,24HINPUT VALUES FOR DE(I,J)//)
 16  19010 FORMAT(15X,45HINPUT VALUES FOR AA(I) OF THE ROW CONSTRAINTS//)
 17  19011 FORMAT(15X,48HINPUT VALUES FOR BB(J) OF THE COLUMN CONSTRAINTS//)
 20  19012 FORMAT( 52HDEGENERACY OCCURS FOR ABAR(I) BBAR(J) TRY NEW DELTA1)
 21  19013 FORMAT( 7HIMAX = ,I3,7HJMAX = ,I3)
 22  19014 FORMAT( 13HABAR(IMAX) = ,E14.8,13HBBAR(JMAX) = ,E14.8)
 23  19015 FORMAT(//37HTOTAL EXECUTION TIME FOR ALGORITHM = ,F12.7.1X,4HSEC.)
 24  19016 FORMAT(8I10)
 25  19017 FORMAT(40X,23HDETAILS OF THE SOLUTION//)
 26  19018 FORMAT(//44HNUMBER OF ITERATIONS AFTER INITIAL SOLUTION ,I5)
 27  19019 FORMAT(//34HVALUE OF PRIMAL OBJECTIVE FUNCTION,E20.6)
 30  19020 FORMAT(//32HVALUE OF DUAL OBJECTIVE FUNCTION,E20.6)
 31  19021 FORMAT(10X,4HROW ,I2,5X,7HCOLUMN ,I2,5X,11HALLOCATION ,E12.6,5X,23
           1HRETURN FROM ALLOCATION ,E12.6)
 32  19022 FORMAT(//74HVALUES OF  THE DUAL SPACE VARIABLES (LAGRANGE MULTIPLI
           1ERS , SHADOW PRICES))
 33  19023 FORMAT(10X,4HROW ,I2,5X,6HU(I)= ,E12.6)
 34  19024 FORMAT(7X,7HCOLUMN ,I2,5X,6HV(J)= ,E12.6)
 35  19025 FORMAT(//37HMAXIMUM VIOLATION OF DUAL CONSTRAINTS,E15.6)
 36  19026 FORMAT(//10X,28HSTARTING NEW SEQUENCE NUMBER.I4)
 37  19027 FORMAT(6X,4H ROW,I3)
 40  19028 FORMAT(10X,41HTOTAL ITERATIONS IN LAST VERTICE SEQUENCE,I3)
 41  19029 FORMAT(10X,47HNUMBER OF POS. XX(I,J) TO BE ASSIGNED BY MATMAX,I3)
 42  19030 FORMAT(5X,10HITERATION ,I4,8X,28HPRIOR VALUE OF OBJ FUNCTION ,E12.
           16,8X,31HMAX. VIOLATION OF DUAL CONSTR. ,E12.6/)
 43  19031 FORMAT(//32HMATRIX MAXIMUM ITERATION NUMBER ,I4)
 44  19032 FORMAT(5X,19HALLOCATION SELECTED,4X,4HROW ,I4,4X,5HCOL. ,I4,4X,9HX
           1X(I,J)= ,F12.6)
 45  19033 FORMAT(5X15HENTERING VECTOR,2I5,8X,14HLEAVING VECTOR,2I5,8X,20HXX(
           1IENTER,JENTER) = ,F12.6//)
 46  19034 FORMAT(5X,30HVALUE OF OBJECTIVE FUNCTION = ,E12.6)
 47  19035 FORMAT(///35X,30HDETAILED INTERMEDIATE PRINTOUT//)
 50  19036 FORMAT(/10X,30HCURRENT SOLUTION ARRAY XX(I,J)//)
 51  19037 FORMAT(/15X,16HUNUSED RESOURCES,5X,26HCOLUMNS 1 THRU NS IN ORDER/)
 52  19038 FORMAT(/15X,17HUNUSED CAPACITIES,5X,23HROWS 1 THRU MS IN ORDER/)
 53  19039 FORMAT(/15X,50H  PRIOR VALUES OF THE DUAL VARIABLES U(I) IN ORDER)
 54  19040 FORMAT(/15X,50H  PRIOR VALUES OF THE DUAL VARIABLES V(J) IN ORDER)
 55          WRITE(6,19001)
 56          WRITE(6,19007)
 57          READ(5,19003)MS,NS,IWRITE,ITAB
 64          WRITE(6,19006)MS,NS,IWRITE,ITAB
```


RAC

```
ISN        SOURCE STATEMENT

 65        WRITE(6,19002)
 66        WRITE(6,19008)
 67        DO 19050 I=1,MS
 70        READ(5,19004)(CE(I,J),J=1,NS)
 75        WRITE(6,19027)I
 76 19050  WRITE(6,19005)(CE(I,J),J=1,NS)
104        WRITE(6,19002)
105        WRITE(6,19009)
106        DO 19060 I=1,MS
107        READ(5,19004)(DE(I,J),J=1,NS)
114        WRITE(6,19027)I
115 19060  WRITE(6,19005)(DE(I,J),J=1,NS)
123        WRITE(6,19002)
124        WRITE(6,19010)
125        READ(5,19004)(AA(I),I=1,MS)
132        WRITE(6,19005)(AA(I),I=1,MS)
137        WRITE(6,19002)
140        WRITE(6,19011)
141        READ(5,19004)(BB(J),J=1,NS)
146        WRITE(6,19005)(BB(J),J=1,NS)
153        WRITE(6,19001)
154         CALL TODAY(0,ITIME,IDAT)
     C---GENERATION OF BASIC FEASIBLE SOLUTION USING MATRIX MAXIMUM-19100-199
155        MTOTAL = MS * NS
156        MBIG   = MS + NS
157        DELTA1 = .1E-5
160        DELTA2 = .1E-4
161        NSS = NS + 1
162        MSS = MS + 1
163        MOUT = MBIG
164 19100  I1 = 0
165        TEMP2 = 0.
166        MMAT = 0
167        DO 19105 I=1,MS
170 19105  ABAR(I)=AA(I)
172        DO 19110 J=1,NS
173 19110  BBAR(J)=BB(J)
175        DO 19199 L=1,MOUT
176        AMAX = 0.
177        DO 19140 I=1,MS
200        IF(ABAR(I)-.1E-6)19140,19140,19120
201 19120  DO 19135 J=1,NS
202        IF(BBAR(J) - .1E-6)19135,19135,19125
203 19125  IF(CE(I,J) - AMAX)19135,19135,19130
204 19130  AMAX = CE(I,J)
205        IMAX = I
206        JMAX = J
207 19135  CONTINUE
211 19140  CONTINUE
213        IF(AMAX)19200,19200,19150
214 19150  ABARTP =  ABAR(IMAX)/ DE(IMAX,JMAX)
215        AE(IMAX,JMAX) = CE(IMAX,JMAX)
216        BBARTP = BBAR(JMAX)
217        IF(ABARTP - BBARTP)19160,19170,19180
     C---BRANCH 19170 IS FOR  DEGENERACY ---
```

```
ISN        SOURCE STATEMENT

220 19160 XX(IMAX,JMAX)=ABARTP
221       ABAR(IMAX)= 0.
222       BBAR(JMAX)= BBARTP - ABARTP
223       IF(ITAB - 1)19199,19165,19199
224 19165 TEMP2 = TEMP2 + CE(IMAX,JMAX) * XX(IMAX,JMAX)
225       MATMA = MMAT + 1
226       WRITE(6,19031)MATMA
227       WRITE(6,19032)IMAX,JMAX,XX(IMAX,JMAX)
230       WRITE(6,19034)TEMP2
231       GO TO 19199
232 19170 ABAR(IMAX)= ABAR(IMAX) + DELTA1
233       IDEGEN = IDEGEN + 1
234       IF(IDEGEN - MTOTAL)19150,19150,19175
235 19175 WRITE(6,19012)
236       WRITE(6,19013)IMAX,JMAX
237       WRITE(6,19014)ABAR(IMAX),BBAR(JMAX)
240       IDEGEN = 0
241       DELTA1 = 10. * DELTA1
242       GO TO 19170
243 19180 XX(IMAX,JMAX)=BBARTP
244       BBAR(JMAX) =0.
245       ABAR(IMAX) = ABAR(IMAX) - DE(IMAX,JMAX)* BBARTP
246       IF(ITAB - 1)19199,19185,19199
247 19185 TEMP2 = TEMP2 + CE(IMAX,JMAX) * XX(IMAX,JMAX)
250       MATMA = MMAT + 1
251       WRITE(6,19031)MATMA
252       WRITE(6,19032)IMAX,JMAX,XX(IMAX,JMAX)
253       WRITE(6,19034)TEMP2
254 19199 MMAT = MMAT + 1
    C------SOLVE FOR THE DUAL SPACE VARIABLES U(I) AND V(J) 19200-19299
256 19200 IDUAL = 0
257       DO 19206 I=1,MS
260       IF(ABAR(I) - .1E-6)19202,19202,19204
261 19202 U(I)= 1.E+35
262       GO TO 19206
263 19204 U(I)= 0.
264       IF(AA(I) - ABAR(I))19206,19206,19205
265 19205 IDUAL = IDUAL + 1
266 19206 CONTINUE
270       DO 19212 J=1,NS
271       IF(BBAR(J) - .1E-6)19208,19208,19210
272 19208 V(J)=1.E+35
273       GO TO 19212
274 19210 V(J) = 0.
275       IF(BB(J) - BBAR(J))19212,19212,19211
276 19211 IDUAL = IDUAL + 1
277 19212 CONTINUE
301       IF(IDUAL - 1)19215,19219,19219
302 19215 MOUT = MMAT - 1
303       IF(MOUT - 2)19411,19216,19216
304 19216 DO 19218 I=1,MSS
305       DO 19217 J=1,NSS
306       AE(I,J) = 0.
307 19217 CONTINUE
311 19218 CONTINUE
```

```
ISN        SOURCE STATEMENT

           IVERT = IVERT + 1
           WRITE(6,19026)IVERT
           WRITE(6,19028)I1
           WRITE(6,19029)MOUT
           IPRINT = 0
           GO TO 19100
     19219 IFINAL = 0
           MDUAL = 0
     19220 IF(IFINAL - MS)19222,19300,19300
     19222 MDUAL = MDUAL + 1
           IF(MDUAL - MBIG)19225,19225,19215
     19225 IFINAL = 0
           DO 19290 I=1,MS
           IF( U(I) - 1.E+35)19260,19230,19260
     19230 DO 19250 J=1,NS
           IF( V(J) - 1.E+35)19235,19250,19235
     19235 IF(AE(I,J))19250,19250,19240
     19240 U(I) = (CE(I,J) - V(J))/ DE(I,J)
           GO TO 19290
     19250 CONTINUE
           GO TO 19290
     19260 IFINAL = IFINAL + 1
           DO 19280 JJ =1,NS
           IF(V(JJ) - 1.E+35)19280,19265,19280
     19265 IF(AE(I,JJ))19280,19280,19270
     19270 V(JJ) = CE(I,JJ) - DE(I,JJ) * U(I)
     19280 CONTINUE
     19290 CONTINUE
           GO TO 19220
       C---OPTIMALITY  TEST  19300-19399-------
     19300 DIFMIN = 0.
           DO 19302 I=1,MS
           IF(U(I) - DIFMIN)19301,19302,19302
     19301 DIFMIN = U(I)
           IENTER = I
           JENTER = NSS
     19302 CONTINUE
           DO 19304 J =1,NS
           IF(V(J) - DIFMIN)19303,19304,19304
     19303 DIFMIN = V(J)
           JENTER = J
           IENTER = MSS
     19304 CONTINUE
           DO 19321 I=1,MS
           DO 19320 J=1,NS
           IF(AE(I,J))19320,19305,19320
     19305 DIF = DE(I,J) * U(I) + V(J) - CE(I,J)
           IF(DIF)19310,19320,19320
     19310 IF(DIF - DIFMIN)19315,19320,19320
     19315 DIFMIN = DIF
           IENTER = I
           JENTER = J
     19320 CONTINUE
     19321 CONTINUE
           IF(DIFMIN + DELTA2)19400,19325,19325
```

ISN SOURCE STATEMENT

```
      C---EXIT 19325 IS FOR OPTIMAL SOLUTIONS----ALL DIF ARE NON NEGATIVE
 19325 CONTINUE
        CALL TODAY(1,ITIME,IDAT)
        TIME = FLOAT(ITIME)/60.
       WRITE(6,19015)TIME
       WRITE(6,19018)I1
       WRITE(6,19025)DIFMIN
       WRITE(6,19001)
       WRITE(6,19017)
       PRIMAL = 0.
       DUAL   = 0.
       DO 19340 I=1,MS
       DO 19335 J=1,NS
       IF(AE(I,J))19335,19335,19330
 19330 TEMP = XX(I,J)*CE(I,J)
       PRIMAL = PRIMAL + TEMP
       WRITE(6,19021)I,J,XX(I,J),TEMP
 19335 CONTINUE
 19340 CONTINUE
       WRITE(6,19019)PRIMAL
       WRITE(6,19022)
       DO 19355  I=1,MS
       WRITE(6,19023)I,U(I)
       DUAL = DUAL+ AA(I) * U(I)
 19355 CONTINUE
       DO 19360 J=1,NS
       WRITE(6,19024)J,V(J)
       DUAL= DUAL + BB(J) * V(J)
 19360 CONTINUE
       WRITE(6,19020)DUAL
       CALL EXIT
      C---REPRESENTATION OF ENTERING VECTOR BY CURRENT BASIS 19400-19499---
 19400 ABAR(MSS)= 0.
       BBAR(NSS)= 0.
       NSPACE = 0
       DO 19401 I=1,MSS
       ISUM(I)= 0
       AE(I,NSS)=ABAR(I)
 19401 DE(I,NSS)=0.
       DO 19402 J=1,NSS
       JSUM(J)= 0
       AE(MSS,J) = BBAR(J)
 19402 DE(MSS,J) = 0.
       DO 19405 I=1,MSS
       DO 19404 J=1,NSS
       ABASIS(I,J) = AE(I,J)
       YY(I,J) = 0.
       IF(AE(I,J))19404,19404,19403
 19403 ISUM(I) = ISUM(I)+ 1
       JSUM(J) = JSUM(J)+ 1
       NSPACE = NSPACE + 1
 19404 CONTINUE
 19405 CONTINUE
       I1=I1+1
       IF(IENTER - MSS)19407,19406,19407
```

```
ISN        SOURCE STATEMENT

     19406 DE(IENTER,NSS)=0.
           DE(MSS,JENTER) = 1.
           GO TO 19410
     19407 IF(JENTER - NSS)19409,19408,19409
     19408 DE(IENTER,NSS)=1.
           DE(MSS,JENTER) = 0.
           GO TO 19410
     19409 DE(MSS,JENTER) = 1.
           DE(IENTER,NSS) = DE(IENTER,JENTER)
     19410 IF(II - NTOTAL)19414,19414,19411
     19411 WRITE(6,19005)DIFMIN
           WRITE(6,19003)II
           WRITE(6,19016)(ISUM(I),I=1,MS)
           WRITE(6,19016)(JSUM(J),J=1,NS)
           DO 19412 I=1,MSS
     19412 WRITE(6,19004)(AE(I,J),J=1,NSS)
           WRITE(6,19016)IENTER,JENTER
           DO 19413 I = 1,MSS
     19413 WRITE(6,19004)(ABASIS(I,J),J=1,NSS)
           GO TO 19325
     19414 CONTINUE
           NTOTAL = 0
           IREPRE = 0
     19415 IF(NTOTAL - MBIG)19417,19495,19495
     19417 IREPRE = IREPRE + 1
           IF(IREPRE - MBIG)19420,19420,19411
     19420 DO 19455 I=1,MS
           IF(ISUM(I) - 1)19455,19425,19455
     19425 DO 19445 J=1,NSS
           IF(ABASIS(I,J))19445,19445,19430
     19430 IF(J - NSS)19435,19440,19435
     19435 YY(I,J)= DE(I,NSS)/DE(I,J)
           DE(I,NSS)= YY(I,J)
           DE(MSS,J)= DE(MSS,J) - YY(I,J)
           GO TO 19450
     19440 YY(I,J)= DE(I,NSS)
           GO TO 19450
     19445 CONTINUE
     19450 ISUM(I)= ISUM(I)- 1
           JSUM(J)= JSUM(J)- 1
           NTOTAL = NTOTAL +1
           ABASIS(I,J)= 0.
     19455 CONTINUE
           IF(NTOTAL - MBIG)19460,19495,19495
     19460 DO 19490 J=1,NS
           IF(JSUM(J) - 1)19490,19470,19490
     19470 DO 19480 I=1,MSS
           IF(ABASIS(I,J))19480,19480,19475
     19475 YY(I,J)= DE(MSS,J)
           DE(I,NSS) = DE(I,NSS)-  DE(I,J)* YY(I,J)
           GO TO 19485
     19480 CONTINUE
     19485 ISUM(I)= ISUM(I)-1
           JSUM(J)= JSUM(J)-1
           ABASIS(I,J) = 0.
```

```
ISN        SOURCE STATEMENT

                 NTOTAL = NTOTAL + 1
           19490 CONTINUE
                 GO TO 19415
           19495 CONTINUE
      C---VECTOR TO LEAVE BASIS IS NOW DETERMINED----19500 - 19599
           19500 THETA  = 1.E+35
                 DO 19525 I=1,MSS
                 DO 19520 J=1,NSS
                 IF(AE(I,J))19520,19520,19504
           19504 IF(YY(I,J))19520,19520,19506
           19506 IF(I - MSS)19510,19508,19510
           19508 ALTEMP = BBAR(J)/YY(I,J)
                 GO TO 19516
           19510 IF(J - NSS)19514,19512,19514
           19512 ALTEMP = ABAR(I)/YY(I,J)
                 GO TO 19516
           19514 ALTEMP = XX(I,J)/YY(I,J)
           19516 IF(ALTEMP -  THETA)19518,19520,19520
           19518 THETA  =  ALTEMP
                 ILEAVE = I
                 JLEAVE = J
           19520 CONTINUE
           19525 CONTINUE
      C---TRANSFORM OLD SOLUTION IN TERMS OF CURRENT BASIS 19600-19699 ----
      C---NEW SOLUTION IS RETURNED TO 19200 -----
           19600 DO 19620  I=1,MS
                 DO 19615 J=1,NS
                 IF(AE(I,J))19615,19615,19605
           19605 IF(YY(I,J))19610,19615,19610
           19610 XX(I,J) = XX(I,J) - THETA*YY(I,J)
           19615 CONTINUE
           19620 CONTINUE
                 DO 19630 I=1,MS
                 IF(YY(I,NSS))19625,19630,19625
           19625 ABAR(I) = ABAR(I) - THETA  * YY(I,NSS)
           19630 CONTINUE
                 DO 19640 J=1,NS
                 IF(YY(MSS,J))19635,19640,19635
           19635 BBAR(J) = BBAR(J) - THETA  * YY(MSS,J)
           19640 CONTINUE
                 AE(ILEAVE,JLEAVE)= 0.
                 IF(ILEAVE - MSS)19642,19641,19642
           19641 BBAR(JLEAVE) = 0.
                 GO TO 19644
           19642 IF(JLEAVE - NSS)19644,19643,19644
           19643 ABAR(ILEAVE) = 0.
           19644 IF(IENTER - MSS)19650,19645,19650
           19645 BBAR(JENTER) = THETA
                 AE(MSS,JENTER) = THETA
                 GO TO 19700
           19650 IF(JENTER - NSS)19660,19655,19660
           19655 ABAR(IENTER) = THETA
                 AE(IENTER,NSS) = THETA
                 GO TO 19700
           19660 XX(IENTER,JENTER) = THETA
```

```
ISN        SOURCE STATEMENT

714          AE(IENTER,JENTER)= CE(IENTER,JENTER)
715          IF(ILEAVE - MSS)19675,19700,19675
716  19675  IF(JLEAVE - NSS)19680,19700,19700
717  19680  XX(ILEAVE,JLEAVE) = 0.
     C-----FREQUENCY OF OBJ. FUNCTION PRINTOUT CONTROLED BY SETTING IWRITE-------
720  19700  IPRINT = IPRINT + 1
721          IF(IPRINT - IWRITE)19200,19710,19710
722  19710  IPRINT = 0
723          DUAL = 0.
724          DO 19750 I=1,MS
725  19750  DUAL = DUAL+ AA(I) * U(I)
727          DO 19760 J=1,NS
730  19760  DUAL= DUAL + BB(J) * V(J)
732          WRITE(6,19039)I1,DUAL,DIFMIN
     C-----DETAILED PRINTOUT CONTROLED BY SETTING ITAB = 1 IN INPUT-------------
733  19800  IF(ITAB - 1)19200,19810,19200
734  19810  WRITE(6,19001)
735          WRITE(6,19035)
736          WRITE(6,19039)
737          WRITE(6,19005)(U(I),I=1,MS)
744          WRITE(6,19040)
745          WRITE(6,19005)(V(J),J=1,NS)
752          WRITE(6,19033)IENTER,JENTER,ILEAVE,JLEAVE,THETA
753          WRITE(6,19036)
754          DO 19820 I=1,MS
755          WRITE(6,19027)I
756  19820  WRITE(6,19005)(XX(I,J),J=1,NS)
764          WRITE(6,19037)
765          WRITE(6,19005)(BBAR(J),J=1,NS)
772          WRITE(6,19038)
773          WRITE(6,19005)(ABAR(I),I=1,MS)
1000         GO TO 19200
1001         END
```

Appendix B

SAMPLE PROBLEM

EXAMPLE

A short example is given here to illustrate (a) the type of problem to which the algorithm can be applied and (b) the details of the solution process. The example with some modifications is taken from Ref 3. The problem is one of allocating several types of commercial aircraft to various routes (e.g., New York to Dallas) in order to maximize overall profit (revenue less operational costs) for the system (see Fig. B1). The problem can be stated as follows:

Let m denote the total number of routes

n denote the total number of types of aircraft

a_i denote the anticipated number of passengers on route i per month

b_j denote the number of aircraft of type j

d_{ij} denote the monthly passenger-carrying capacity of aircraft type j on route i

r_i denote the revenue per passenger on route i

s_{ij} denote the monthly operational cost for operating aircraft type j on route i.

Then we seek to find the allocations x_{ij} of aircraft type j to route i for the system that maximize profit subject to constraints on the number of anticipated passengers using the various routes and the available number of each type of aircraft.

Specifically

maximize profit $z(x_{ij})$

Route i	Aircraft type j			
	1	2	3	4
1	1600			900
2	1500	1000	500	1100
3	2800	1400		2200
4	2300	1500	700	1700
5	8100	5700	2900	5500

a. Aircraft Passenger-Carrying Capacity per Month (d_{ij})

Route i	Aircraft type j			
	1	2	3	4
1	18,000			17,000
2	21,000	15,000	10,000	16,000
3	18,000	16,000		17,000
4	16,000	14,000	9,000	15,000
5	10,000	9,000	6,000	1,000

b. Operational Costs in Dollars per Month (s_{ij})

Fig. B1—Sample Data for Aircraft Allocation Problem

Blanks in cell i, j indicate that aircraft type j is never assigned to route i.

$m = 5$ (routes) $n = 4$ (types of aircraft)

Passenger data	Passenger fare data	Aircraft data
$a_1 = 25{,}000$	$r_1 = \$130.00$	$b_1 = 10$
$a_2 = 12{,}000$	$r_2 = \$130.00$	$b_2 = 19$
$a_3 = 18{,}000$	$r_3 = \$\ 70.00$	$b_3 = 25$
$a_4 = 9{,}000$	$r_4 = \$\ 70.00$	$b_4 = 15$
$a_5 = 60{,}000$	$r_5 = \$\ 10.00$	

$$z(x_{ij}) = \underbrace{\sum_{i=1}^{m} r_i \Big(\sum_{j=1}^{n} d_{ij} x_{ij} \Big)}_{\text{revenue}} - \underbrace{\sum_{i,j}^{m,n} s_{ij} x_{ij}}_{\text{operational costs}} \qquad \text{(B1)}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} (r_i d_{ij} - s_{ij}) x_{ij}$$

subject to

$$\sum_{j=1}^{n} d_{ij} x_{ij} \le a_i \qquad (i = 1, \ldots, m)$$

do not exceed passenger demand for each route i

$$\sum_{i=1}^{m} x_{ij} \le b_j \qquad (j = 1, \ldots, n)$$

do not exceed aircraft availability for each type j

Data for $m = 5$, $n = 4$ are given in Table 1 as taken from Ferguson-Dantzig.[3] It should be noted that if $c_{ij} = r_i d_{ij} - s_{ij}$ above, then the form of the problem is that of Prob 3. The Ferguson-Dantzig[3] example requires that all aircraft be allocated, i.e., $\sum_{i=1}^{m} x_{ij} = b_j \ (j = 1, \ldots, n)$. The algorithm here does not require that all resources be allocated. However, at least all the row or all the column constraints will be binding for an optimal solution. If all the column constraints are binding, then of course the equality constraints on the columns are satisfied.

The example of Ferguson-Dantzig[3] can be formulated from the example here for $m = 5$, $n = 4$ as follows:

Let

$$x_{i5} = a_i - \sum_{j=1}^{4} d_{ij} x_{ij} \qquad s_{i5} = r_i \qquad d_{i5} = 1 \tag{B2}$$

then we have

$$\begin{aligned} \max_{x_{ij}} z(x_{ij}) &= \max_{x_{ij}} \left[\sum_{i=1}^{5} r_i (a_i - x_{i5}) - \sum_{i=1}^{5} \left(\sum_{j=1}^{4} s_{ij} x_{ij} \right) \right] \\ &= \max_{x_{ij}} \left[\sum_{i=1}^{5} r_i a_i - \left\{ \sum_{i=1}^{5} \left(\sum_{j=1}^{4} s_{ij} x_{ij} + r_i x_{i5} \right) \right\} \right] \\ &= \sum_{i=1}^{5} r_i a_i - \min_{x_{ij}} \left\{ \sum_{i=1}^{5} \left(\sum_{j=1}^{5} s_{ij} x_{ij} \right) \right\} \end{aligned} \tag{B3}$$

subject to

$$\sum_{j=1}^{4} d_{ij} x_{ij} + x_{i5} = \sum_{j=1}^{5} d_{ij} x_{ij} = a_i \qquad (i = 1, \ldots, 4)$$

$$\sum_{i=1}^{4} x_{ij} = b_j \qquad (j = 1, \ldots, 5)$$

Ferguson and Dantzig[3] consider the problem

$$\min_{x_{ij}} \sum_{\substack{i=1 \\ j=1}}^{5,5} s_{ij} x_{ij}$$

subject to the two sets of constraints given.

Details of the solution are printed at each step of the solution for the problem expressed by Eq B1. A completely detailed description illustrating the algorithm on a step-by-step basis is presented between the first and second intermediate printouts (iterations) of phase 2 of the algorithm.

INPUT CONSTANTS

INPUT CONSTANTS FOR OPTIMAL ALLOCATION PROBLEM

MS= 5 NS= 4 FREQ. OF OBJ. FUNCT. PRINTOUT 1

INPUT VALUES FOR CF(I,J)

ROW 1	0.190000E 06	0.	0.	0.100000E 06
ROW 2	0.174000E 06	0.115000E 06	0.550000E 05	0.127000E 06
ROW 3	0.168000E 06	0.820000E 05	0.	0.137000E 06
ROW 4	0.145000E 06	0.910000E 05	0.400000E 05	0.104000E 06
ROW 5	0.710000E 05	0.480000E 05	0.230000E 05	0.450000E 05

INPUT VALUES FOR DE(I,J)

ROW 1	0.160000E 04	0.100000E 01	0.100000E 01	0.900000E 03
ROW 2	0.150000E 04	0.100000E 04	0.500000E 03	0.110000E 04
ROW 3	0.280000E 04	0.140000E 04	0.100000E 01	0.220000E 04
ROW 4	0.230000E 04	0.150000E 04	0.700000E 03	0.170000E 04
ROW 5	0.810000E 04	0.570000E 04	0.290000E 04	0.550000E 04

INPUT VALUES FOR AA(I) OF THE ROW CONSTRAINTS

0.250000E 05	0.120000E 05	0.180000E 05	0.900000E 04

INPUT VALUES FOR BB(J) OF THE COLUMN CONSTRAINTS

0.160000E 02	0.190000E 02	0.250000E 02	0.150000E 02

CR OPTIMAL ALLOCATION PROBLEM

REQ. OF OBJ. FUNCT. PRINTOUT 1   DETAILED PRINTOUT IF ITAB = 1   ITAB= 1

0.	0.100000E 06
0.550000E 05	0.127000E 06
0.	0.137000E 06
0.400000E 05	0.104000E 06
0.230000E 05	0.450000E 05

0.100000E 01	0.900000E 03
0.500000E 03	0.110000E 04
0.100000E 01	0.220000E 04
0.700000E 03	0.170000E 04
0.290000E 04	0.550000E 04

CONSTRAINTS

0.180000E 05	0.900000E 04	0.600000E 05

UMN CONSTRAINTS

0.250000E 02	0.150000E 02

PHASE 1, ITERATIONS

```
MATRIX MAXIMUM ITERATION NUMBER     1
     ALLOCATION SELECTED     ROW     1     COL.     1     XX(I,J)=  0.10000E 02
     VALUE OF OBJECTIVE FUNCTION =  0.19000E 07

MATRIX MAXIMUM ITERATION NUMBER     2
     ALLOCATION SELECTED     ROW     3     COL.     4     XX(I,J)=  0.81818E 01
     VALUE OF OBJECTIVE FUNCTION =  0.30209E 07

MATRIX MAXIMUM ITERATION NUMBER     3
     ALLOCATION SELECTED     ROW     2     COL.     4     XX(I,J)=  0.68182E 01
     VALUE OF OBJECTIVE FUNCTION =  0.38868E 07

MATRIX MAXIMUM ITERATION NUMBER     4
     ALLOCATION SELECTED     ROW     2     COL.     2     XX(I,J)=  0.45000E 01
     VALUE OF OBJECTIVE FUNCTION =  0.44043E 07

MATRIX MAXIMUM ITERATION NUMBER     5
     ALLOCATION SELECTED     ROW     4     COL.     2     XX(I,J)=  0.60000E 01
     VALUE OF OBJECTIVE FUNCTION =  0.49503E 07

MATRIX MAXIMUM ITERATION NUMBER     6
     ALLOCATION SELECTED     ROW     5     COL.     2     XX(I,J)=  0.85000E 01
     VALUE OF OBJECTIVE FUNCTION =  0.53583E 07

MATRIX MAXIMUM ITERATION NUMBER     7
     ALLOCATION SELECTED     ROW     5     COL.     3     XX(I,J)=  0.39828E 01
     VALUE OF OBJECTIVE FUNCTION =  0.54499E 07
     ITERATION     1        PRIOR VALUE OF OBJ FUNCTION  0.54499E 07        MAX. VIOLATION OF
```

I,J)= 0.10000E 02

I,J)= 0.81818E 01

I,J)= 0.68182E 01

I,J)= 0.45000E 01

I,J)= 0.60000E 01

I,J)= 0.85000E 01

I,J)= 0.39828E 01

).54499E 07   MAX. VIOLATION OF DUAL CONSTR. -0.96428E 05

B

DETAILED INTERMEDIATE PRINTOUT

PRIOR VALUES OF THE DUAL VARIABLES U(I) IN ORDER

0.	0.112207E 03	0.606489E 02	0.588046E 02

PRIOR VALUES OF THE DUAL VARIABLES V(J) IN ORDER

0.190000E 06	0.279310E 04	0.	0.357242E 04

ENTERING VECTOR 1 4 LEAVING VECTOR 2 4 XX(IENTER,

CURRENT SOLUTION ARRAY XX(I,J)

ROW 1	0.100000E 02	0.	0.	0.681818E 01
ROW 2	0.	0.120000E 02	0.	0.
ROW 3	0.	0.	0.	0.818182E 01
ROW 4	0.	0.600000E 01	0.	0.
ROW 5	0.	0.100000E 01	0.187241E 02	0.

UNUSED RESOURCES COLUMNS 1 THRU NS IN ORDER

0.	0.	0.627586E 01	0.

UNUSED CAPACITIES ROWS 1 THRU MS IN ORDER

0.286364E 04	0.	0.	0.

ITERATION 2 PRIOR VALUE OF OBJ FUNCTION 0.61074E 07 MAX. V

```
ILED INTERMEDIATE PRINTOUT

AL VARIABLES U(I) IN ORDER
22U7E 03     0.606489E 02     0.588046E 02     0.793103E 01

AL VARIABLES V(J) IN ORDER
9310E 04     0.               0.357242E 04
EAVING VECTOR     2     4        XX(IENTER,JENTER) =  0.68182E 01

)

             0.               0.681818E 01

00C0E 02     0.               0.

             0.               0.818182E 01

0000E 01     0.               0.

0C00E 01     0.187241E 02     0.

UMNS 1 THRU NS IN CRDER

             0.627586E 01     0.

NS 1 THRU MS IN CRDER

             0.               0.               0.
OF OBJ FUNCTION  0.61074E 07        MAX. VIOLATION OF DUAL CONSTR. -0.55661E 05
```


B

RAC

DETAILED DESCRIPTION OF ONE ITERATION OF PHASE 2

A detailed numerical examination of one iteration of phase 2 of the algorithm is presented here. Given the solution array **XX(I,J)** of the previous page, we wish to test optimality of the solution. Proceeding as in step 2 of the algorithm we solve Eqs 15 for $(u_1, \ldots, u_5, v_1, \ldots, v_4)$.

Since $x_{63} = 6.27586 > 0$ and $x_{51} = 2863.63 > 0$, we have $v_3 = 0$ and $u_1 = 0$ immediately, i.e., we solve for the zero-valued dual-space variables first. The remaining seven equations

$$d_{ij}u_i + v_j = c_{ij} \quad \text{if} \quad x_{ij} > 0$$

are solved sequentially. We have

$$x_{11} = 10 > 0 \Rightarrow v_1 = c_{11} = 190{,}000$$

$$x_{14} = 6.81818 > 0 \Rightarrow v_4 = c_{14} = 100{,}000$$

$$x_{53} = 18.7241 > 0 \Rightarrow u_5 = \frac{c_{53}}{d_{53}} = \frac{23{,}000}{2{,}900} = 7.93103$$

$$x_{52} = 1 > 0 \Rightarrow v_2 = c_{52} - d_{52}u_5 = 48{,}000 - (5700)(7.93103) = 2793.10$$

$$x_{22} = 12 > 0 \Rightarrow u_2 = \frac{c_{22} - v_2}{d_{22}} = \frac{115{,}000 - 2793.10}{1000} = 112.207$$

$$x_{42} = 6 > 0 \Rightarrow u_4 = \frac{c_{42} - v_2}{d_{42}} = \frac{91{,}000 - 2793.10}{1500} = 58.8046$$

$$x_{34} = 7.54545 > 0 \Rightarrow u_3 = \frac{c_{34} - v_4}{d_{34}} = \frac{137{,}000 - 100{,}000}{2200} = 16.8182$$

Moving on to step 3 of the algorithm the optimality test is now made (i.e., Conditions 7$\bar{D}$ in scalar form of the $m + n + mn = 29$ equalities of Eq 16 are tested). It is seen directly that u_i $(i = 1, \ldots, 5)$ and v_j $(j = 1, \ldots, 4)$ are nonnegative. Seven of the remaining $mn = 20$ inequalities; namely, $w_{s_{ij}} = d_{ij}u_i + v_j - c_{ij} = 0$ for $x_{ij} > 0$ (i.e., for x_{11}, x_{14}, x_{53}, x_{52}, x_{22}, x_{42}, and x_{34}) are equalities, hence only the remaining $20 - 7 = 13$ values of $w_{s_{ij}}$ are tested for nonnegativity. Computing directly we have

$$w_{s_{12}} = d_{12}u_1 + v_2 - c_{12} = (1)(0) + 2793.10 - 0 > 0$$

$$w_{s_{13}} = d_{13}u_1 + v_3 - c_{13} = (1)(0) + 0 - 0 \geq 0$$

$$w_{s_{21}} = d_{21}u_2 + v_1 - c_{21} = (1500)(112.207) + 190{,}000 - 174{,}000 > 0$$

$$w_{s_{23}} = d_{23}u_2 + v_3 - c_{23} = (500)(112.207) + 0 - 55{,}000 > 0$$

$$w_{s_{24}} = d_{24}u_2 + v_4 - c_{24} = (1100)(112.207) + 100{,}000 - 127{,}000 > 0$$

$$w_{s_{31}} = d_{31}u_3 + v_1 - c_{31} = (2800)(16.8182) + 190{,}000 - 168{,}000 > 0$$

$$w_{s_{32}} = d_{32}u_3 + v_2 - c_{32} = (1400)(16.8182) + 2793.10 - 82{,}000 = -55{,}661.42 < 0$$

$$w_{s_{33}} = d_{33}u_3 + v_3 - c_{33} = (1)(16.8182) + 0 - 0 > 0$$

$$w_{s_{41}} = d_{41}u_4 + v_1 - c_{41} = (2300)(58.8046) + 190{,}000 - 145{,}000 > 0$$

$$w_{s_{43}} = d_{43}u_4 + v_3 - c_{43} = (700)(58.8046) + 0 - 40{,}000 = 41{,}163.22 - 40{,}000 > 0$$

$$w_{s_{44}} = d_{44}u_4 + v_4 - c_{44} = (1700)(58.8046) + 100{,}000 - 104{,}000 > 0$$

$$w_{s_{51}} = d_{51}u_5 + v_1 - c_{51} = (8100)(7.93103) + 190{,}000 - 71{,}000 > 0$$

$$w_{s_{54}} = d_{54}u_5 + v_4 - c_{54} = (5500)(7.93103) + 100{,}000 - 45{,}000 > 0$$

It is seen at this point that the only violation and hence the maximum violation (step 4 of the algorithm) of Conditions $7\bar{D}$ of Eq 16 is

$$w_{s_{32}} = -55661.42 < 0$$

Hence the solution is not optimal and the vector $d_{32}\vec{e}_3 + \vec{e}_7$ enters the basis.

The vector selected to leave the basis is determined next, but first it is necessary to determine the components y_{ij}^{32} of the entering vector $d_{32}\vec{e}_3 + \vec{e}_7$ in terms of the current basis. The set of $m + n = 5 + 4 = 9$ linear equations Eq 21 is thus solved (step 5 of the algorithm). We have for the case at hand the following system:

$$y_{11}^{32}(1600) + y_{14}^{32}(900) + y_{15}^{32}(1) = 0$$

$$y_{22}^{32}(1000) = 0$$

$$y_{34}^{32}(2200) = d_{32} = 1400$$

$$y_{42}^{32}(1500) = 0$$

$$y_{52}^{32}(5700) + y_{53}^{32}(2900) = 0$$

$$y_{11}^{32} = 0$$

$$y_{22}^{32} + y_{42}^{32} + y_{52}^{32} = 1$$

$$y_{53}^{32} + y_{63}^{32} = 0$$

$$y_{14}^{32} + y_{34}^{32} = 0$$

The solution to the foregoing equations is

$$y_{22}^{32} = 0,\ y_{34}^{32} = \frac{1400}{2200},\ y_{42}^{32} = 0,\ y_{52}^{32} = 1,\ y_{11}^{32} = 0,$$

$$y_{53}^{32} = -\frac{5700}{2900},\ y_{63}^{32} = \frac{5700}{2900},\ y_{14}^{32} = -\frac{1400}{2200},\ y_{15}^{32} = \frac{(-900)(-1400)}{2200}$$

Thus the entering vector $d_{32}\vec{e}_3 + \vec{e}_7 = 1400\vec{e}_3 + \vec{e}_7$ can be written as the following linear combination of the current basis vectors.

$$1400\vec{e}_3 + \vec{e}_7 = \left\{ \begin{array}{l} -\frac{1400}{2200}[900\vec{e}_1 + \vec{e}_9] + \frac{(-900)(-1400)}{2200}[\vec{e}_1] \\ +\frac{1400}{2200}[2200\vec{e}_3 + \vec{e}_9] + 1[5700\vec{e}_5 + \vec{e}_7] \\ -\frac{5700}{2900}[2900\vec{e}_5 + \vec{e}_8] + \frac{5700}{2900}[\vec{e}_8] \end{array} \right.$$

The vector now selected to leave the basis is determined from the indexes that yield the minimum in the brackets. Using Eq 29 (step 6 of the algorithm) we have

$$\hat{\theta} = \min \left\{ \frac{8.18182}{\frac{1400}{2200}}, \frac{1}{1}, \frac{6.27586}{\left(\frac{5700}{2900}\right)}, \frac{2863.64}{\frac{(900)(1400)}{2200}} \right\}$$

$$= 1$$

and the minimizing indexes are (5,2), since $\frac{x_{52}}{y_{52}^{32}} = \frac{1}{1}$ yields the minimum value $\hat{\theta}$. Thus the vector $d_{52}\vec{e}_5 + \vec{e}_7 = 5700\vec{e}_5 + \vec{e}_7$ leaves the basis.

The new solution is evaluated (step 7) using Eqs 28 and appears in the next intermediate printout. The algorithm again returns to step 2 for the next iteration. The value of the objective function was increased by $\hat{\theta}(-u_{s_{32}})$ = \$55,661.42 during this iteration.

PHASE 2, REMAINING ITERATIONS

DETAILED INTERMEDIATE PRINTOUT

PRIOR VALUES OF THE DUAL VARIABLES U(I) IN ORDER

0. 0.112207E 03 0.168182E 02 0.588046

PRIOR VALUES OF THE DUAL VARIABLES V(J) IN ORDER

0.190000E 06 0.279310E 04 0. 0.100000

ENTERING VECTOR 3 2 LEAVING VECTOR 5 2 XX(IE

CURRENT SOLUTION ARRAY XX(I,J)

ROW 1

0.100000E 02 0. 0. 0.745455E

ROW 2

0. 0.120000E 02 0. 0.

ROW 3

0. 0.100000E 01 0. 0.754545E

ROW 4

0. 0.600000E 01 0. 0.

ROW 5

0. 0. 0.206897E 02 0.

UNUSED RESOURCES COLUMNS 1 THRU NS IN ORDER

0. 0. 0.431034E 01 0.

UNUSED CAPACITIES ROWS 1 THRU MS IN ORDER

0.229091E 04 0. 0. 0.

ITERATION 3 PRIOR VALUE OF OBJ FUNCTION 0.61630E 07 M

```
TAILED INTERMEDIATE PRINTOUT

DUAL VARIABLES U(I) IN ORDER
.122U7E (3      0.168182E 02      0.588046E 02      0.793103E 01

DUAL VARIABLES V(J) IN ORDER
.79310E 04      0.                0.100000E 06
LEAVING VECTOR    5    2          XX(IENTER,JENTER) =   0.10000E 01

J)

                0.                0.745455E 01

200CCE 02       C.                0.

D0000E 01       0.                0.754545E 01

D0000E 01       0.                0.

                0.206897E 02      0.

LUMNS 1 THRU NS IN ORDER

                0.431034E 01      0.

WS 1 THRU MS IN ORDER

                0.                0.                0.
CF OBJ FUNCTION  0.61630E 07        MAX. VIOLATION OF DUAL CONSTR. -0.26727E 05
```


B

```
                         DETAILED INTERMEDIATE PRINTOUT

          PRIOR VALUES OF THE DUAL VARIABLES U(I) IN ORDER
            0.               0.565455E 02     0.168182E 02     0.216970E 02     0.793

          PRIOR VALUES OF THE DUAL VARIABLES V(J) IN ORDER
            0.190000E 06     0.584545E 05     0.               0.100000E 06
   ENTERING VECTOR     2     3         LEAVING VECTOR     6     3         XX(IENTER,JENTER)

       CURRENT SOLUTION ARRAY XX(I,J)

   ROW   1
            0.100000E 02     0.               0.               0.882602E 01
   ROW   2
            0.               0.984483E 01     0.431034E 01     0.
   ROW   3
            0.               0.315517E 01     0.               0.617398E 01
   ROW   4
            0.               0.600000E 01     0.               0.
   ROW   5
            0.               0.               0.206897E 02     0.

          UNUSED RESOURCES       COLUMNS 1 THRU NS IN ORDER

            0.               0.               0.               0.

          UNUSED CAPACITIES       ROWS 1 THRU MS IN ORDER

            0.105658E 04     0.               0.               0.               0.
   ITERATION     4         PRIOR VALUE OF OBJ FUNCTION  0.62782E 07         MAX. VIOLATIO
```

PRINTOUT

IN ORDER
82E 02   0.216970E 02   0.793103E 01

IN ORDER
          0.100000E 06
    3   XX(IENTER,JENTER) =   0.43103E 01

          0.882602E 01
34E 01   0.
          0.617398E 01
          0.
97E 02   0.
ORDER
          0.
DER
          0.   0.
.62782E 07   MAX. VIOLATION OF DUAL CONSTR. -0.12853E 01

B

```
                    DETAILED INTERMEDIATE PRINTOUT

          PRIOR VALUES OF THE DUAL VARIABLES U(I) IN ORDER
           0.              0.565455E 02    0.168182E 02     0.2169

          PRIOR VALUES OF THE DUAL VARIABLES V(J) IN ORDER
           0.190000E 06    0.584545E 05    0.267273E 05     0.1000
    ENTERING VECTOR    5    5        LEAVING VECTOR    1    5      XX(I

        CURRENT SOLUTION ARRAY XX(I,J)

      ROW  1
           0.100000E 02    0.              0.               0.1000
      ROW  2
           0.              0.800000E 01    0.800000E 01     0.
      ROW  3
           0.              0.500000E 01    0.               0.5000
      ROW  4
           0.              0.600000E 01    0.               0.
      ROW  5
           0.              0.              0.170000E 02     0.

          UNUSED RESOURCES     COLUMNS 1 THRU NS IN ORDER

           0.              0.              0.               0.

          UNUSED CAPACITIES     ROWS 1 THRU MS IN ORDER

           0.              0.              0.               0.

TOTAL EXECUTION TIME FOR ALGORITHM =    0.7333333 SEC.

NUMBER OF ITERATIONS AFTER INITIAL SOLUTION      4

MAXIMUM VIOLATION OF DUAL CONSTRAINTS   0.
```

```
ILED INTERMEDIATE PRINTOUT

AL VARIABLES U(I) IN ORDER
5455E 02     0.168182E 02     0.216970E 02    -0.128527E 01

AL VARIABLES V(J) IN ORDER
4545E 05     0.267273E 05     0.100000E 06
EAVING VECTOR     1     5        XX(IENTER,JENTER' =   0.10700E 05

0

             0.                0.100000E 02

0000E 01     0.800000E 01      0.

0000E 01     0.                0.500000E 01

0000E 01     0.                0.

             0.170000E 02      0.

UMNS 1 THRU NS IN ORDER

             0.                0.

S 1 THRU MS IN ORDER

             0.                0.               0.107000E 05

0.7333333 SEC.

ION      4

0.
```


B

RAC

DETAILS OF THE SOLUTION

```
ROW   1      COLUMN   1      ALLOCATION   0.10000E 02      RETURN FROM ALLOCATION
ROW   1      COLUMN   4      ALLOCATION   0.10000E 02      RETURN FROM ALLOCATION
ROW   2      COLUMN   2      ALLOCATION   0.80000E 01      RETURN FROM ALLOCATION
ROW   2      COLUMN   3      ALLOCATION   0.80000E 01      RETURN FROM ALLOCATION
ROW   3      COLUMN   2      ALLOCATION   0.50000E 01      RETURN FROM ALLOCATION
ROW   3      COLUMN   4      ALLOCATION   0.50000E 01      RETURN FROM ALLOCATION
ROW   4      COLUMN   2      ALLOCATION   0.60000E 01      RETURN FROM ALLOCATION
ROW   5      COLUMN   3      ALLOCATION   0.17000E 02      RETURN FROM ALLOCATION

VALUE OF PRIMAL OBJECTIVE FUNCTION        0.629200E 07

VALUES OF  THE DUAL SPACE VARIABLES (LAGRANGE MULTIPLIERS , SHADOW PRICES)
          ROW   1      U(I)=   0.13[illegible]16E 02
          ROW   2      U(I)=   0.64000E 02
          ROW   3      U(I)=   0.22143E 02
          ROW   4      U(I)=   0.26667E 02
          ROW   5      U(I)=   0.
       COLUMN   1      V(J)=   0.16917E 06
       COLUMN   2      V(J)=   0.51000E 05
       COLUMN   3      V(J)=   0.23000E 05
       COLUMN   4      V(J)=   0.88286E 05

VALUE OF DUAL OBJECTIVE FUNCTION        0.629200E 07
```

N FROM ALLOCATION 0.19000E 07
N FROM ALLOCATION 0.10000E 07
N FROM ALLOCATION 0.92000E 06
N FROM ALLOCATION 0.44000E 06
N FROM ALLOCATION 0.41000E 06
N FROM ALLOCATION 0.68500E 06
N FROM ALLOCATION 0.54600E 06
N FROM ALLOCATION 0.39100E 06

RICES)

B

Appendix C

COMPARATIVE SIMPLEX ALGORITHM

This appendix contains a general simplex algorithm programmed specifically to solve Prob 3. The program accepts the same input data as the MATMAX algorithm with the exception of the IWRITE and ITAB information used in the MATMAX algorithm. The algorithm here has been used for comparative purposes, with particular attention given to the times required by the two algorithms to solve the same problem.

FORTRAN SOURCE LIST

```
ISN         SOURCE STATEMENT

  0 $IBFTC MATLIN
  1         DIMENSION CE(20,24),DE(21,25),AA(20),BB(24)
  2         DIMENSION AS(42,474),BS(45),CS(525)
  3         DIMENSION  IPATHS(45),CTS(45),BTS(45),ATSI(45)
  4         DIMENSION ZS(525),ZMCS(525),ATSJ(525)
  5  18001 FORMAT(1H1)
  6  18002 FORMAT(1H )
  7  18003 FORMAT(3I10)
 10  18004 FORMAT(6F12.6)
 11  18005 FORMAT(15X,6E16.6)
 12  18006 FORMAT(8X,2HMS,8X,2HNS,2X,15HPRINT FREQUENCY)
 13  18007 FORMAT(30X,46HINPUT CONSTANTS FOR OPTIMAL ALLOCATION PROBLEM//)
 14  18008 FORMAT(15X,24HINPUT VALUES FOR CE(I,J)//)
 15  18009 FORMAT(15X,24HINPUT VALUES FOR DE(I,J)//)
 16  18010 FORMAT(15X,45HINPUT VALUES FOR AA(I) OF THE ROW CONSTRAINTS//)
 17  18011 FORMAT(15X,48HINPUT VALUES FOR BB(J) OF THE COLUMN CONSTRAINTS//)
 20  18015 FORMAT(//37HTOTAL EXECUTION TIME FOR ALGORITHM = ,F12.7,1X,4HSEC.)
 21  18019 FORMAT(//34HVALUE OF PRIMAL OBJECTIVE FUNCTION,E20.6)
 22  18027 FORMAT(6X,4H ROW,I3)
 23  18020 FORMAT(10I10)
 24  18028 FORMAT(6X,7H COLUMN,I3)
 25         WRITE(6,18001)
 26         WRITE(6,18007)
 27         READ(5,18003)MM,NN
 32         WRITE(6,18006)
 33         WRITE(6,18003)MM,NN
 34         WRITE(6,18002)
 35         WRITE(6,18008)
 36         DO 18035 I=1,MM
 37         READ(5,18004)(CE(I,J),J=1,NN)
 44         WRITE(6,18027)I
 45  18035 WRITE(6,18005)(CE(I,J),J=1,NN)
 53         WRITE(6,18002)
 54         WRITE(6,18009)
 55         DO 18040 I=1,MM
 56         READ(5,18004)(DE(I,J),J=1,NN)
 63         WRITE(6,18027)I
 64  18040 WRITE(6,18005)(DE(I,J),J=1,NN)
 72         WRITE(6,18002)
 73         WRITE(6,18010)
 74         READ(5,18004)(AA(I),I=1,MM)
101         WRITE(6,18005)(AA(I),I=1,MM)
106         WRITE(6,18002)
107         WRITE(6,18011)
110         READ(5,18004)(BB(J),J=1,NN)
115         WRITE(6,18005)(BB(J),J=1,NN)
122         WRITE(6,18001)
123         MS = MM + NN
124         NS = MM * NN
125         DO 18110 I = 1,MM
126         DO 18110 J = 1,NN
127         K = NN *(I - 1) + J
130         CS(K) =-CE(I,J)
131  18110 AS(I,K) = DE(I,J)
134         DO 18120 I = 1,NN
```

```
ISN        SOURCE STATEMENT

135              DO 18120 J = 1,MM
136              II = MM + I
137              JJ = NN * (J - 1) + I
140  18120 AS(II,JJ) = 1.
143              DO 18130 I = 1,MM
144  18130 BS(I) = AA(I)
146              DO 18140 J = 1,NN
147              JJ = MM + J
150  18140 BS(JJ) = BB(J)
152              DO 18150 I = 1,MS
153              J = I + NS
154              IPATHS(I) = J
155  18150 AS(I,J) = 1.
157              NNS = MS + NS
           C --- SIMPLEX METHOD SOLUTION OF LINEAR PROGRAMMING PROBLEM
160              EPSLP = .1E-5
           C
161               CALL TODAY(0,ITIME,IDAT)
           C
           C --- BEGIN ITERATION
           C
162  18200 COST=0.
163              DO 18205 I=1,MS
164  18205 COST = COST + CTS(I)*BS(I)
           C
           C --- COMPUTE ZS AND ZMCS VECTORS
           C
166  18216 DO 18220 J=1,NNS
167              ZTS=0.
170              DO 18217 I=1,MS
171  18217 ZTS=ZTS + CTS(I)*AS(I,J)
173              ZS(J)=ZTS
174  18220 ZMCS(J)=ZS(J)-CS(J)
           C
           C --- SELECT MAXIMUM ZMCS.  IF NO POSITIVE, END.
           C
176  18230 CMAX=ZMCS(1)
177              JMAX=1
200              DO 18250 J=2,NNS
201              IF(CMAX-ZMCS(J))18240,18250,18250
202  18240 CMAX=ZMCS(J)
203              JMAX=J
204  18250 CONTINUE
206              IF(ZMCS(JMAX)-EPSLP)18800,18800,18260
           C
           C --- SELECT MINIMUM DS(I)=BS(I)/AS(I,JMAX) WHERE A(I,JMAX) IS POSITIVE
           C
207  18260 DSMIN=1.E+35
210  18270 DO 18350 I=1,MS
211              IF(AS(I,JMAX)-.1E-6)18350,18350,18300
212  18300 DST=BS(I)/AS(I,JMAX)
213              IF(DST-DSMIN)18310,18350,18350
214  18310 DSMIN=DST
215              IMIN=I
216  18350 CONTINUE
```

```
 ISN          SOURCE STATEMENT

       C
       C --- COMPUTE NEW MATRIX ATS
       C
 220           DO 18400 I=1,MS
 221           BTS(I)=AS(I,JMAX)    *    BS(IMIN)    /        AS(IMIN,JMAX)
 222 18400 ATSI(I)=AS(I,JMAX)/AS(IMIN,JMAX)
 224           TEMP=AS(IMIN,JMAX)
 225           TEMP2 = ZMCS(JMAX) / TEMP
 226           THETA = BS(IMIN) / AS(IMIN,JMAX)
 227           COST = COST - THETA * ZMCS(JMAX)
 230           WRITE(6,18019)COST
 231           DO 18410 J = 1,NNS
 232 18410 ATSJ(J)=AS(IMIN,J)
 234           DO 18525 I=1,MS
 235           IF(I-IMIN)18450,18500,18450
 236 18450 IF(ATSI(I))18455,18525,18455
 237 18455 BS(I)=BS(I)-BTS(I)
 240           DO 18475 J = 1,NNS
 241           IF(ATSJ(J))18460,18475,18460
 242 18460 AS(I,J) = AS(I,J) - ATSI(I) * ATSJ(J)
 243           IF(AS(I,J)-.1E-10)18462,18462,18475
 244 18462 IF(AS(I,J) - .1E-10)18475,18465,18465
 245 18465 AS(I,J)=0.
 246 18475 CONTINUE
 250           GO TO 18525
 251 18500 DO 18510 J=1,NNS
 252           AS(I,J)=AS(I,J)/TEMP
 253           IF(AS(I,J)-.1E-10)18502,18502,18510
 254 18502 IF(AS(I,J) - .1E-10)18510,18505,18505
 255 18505 AS(I,J)=0.
 256 18510 CONTINUE
 260           BS(I)=BS(I)/TEMP
 261 18525 CONTINUE
       C
 263           DO 18550 J=1,NNS
 264 18550 ZMCS(J) = ZMCS(J) - ATSJ(J) *TEMP2
       C --- SUBSTITUTE IPATH OF JMAX FOR IMIN, C OF JMAX FOR IMIN
       C
 266           IPATHS(IMIN)=JMAX
 267           CTS(IMIN)=CS(JMAX)
       C
       C --- TRANSFER BACK TO BEGIN ITERATION
       C
 270           GO TO 18230
       C
 271 18800 CONTINUE
 272            CALL TODAY(1,ITIME,IDAT)
 273            TIME = FLOAT(ITIME)/60.
 274           WRITE(6,18015)TIME
 275           WRITE(6,18020)(IPATHS(I),I=1,MS)
 302           WRITE(6,18005)(BS(I),I=1,MS)
 307           WRITE(6,18005)(CTS(I),I=1,MS)
 314           CALL EXIT
 315           END
```


RAC

REFERENCES

CITED REFERENCES

1. J. V. Michalowicz, J. Bracken, and L. Lyons, "Description of Initial Formulation of Nonlinear Programming Models for Optimizing Mixed Offensive and Defensive Weapons Systems," unpublished notes, Research Analysis Corp., 13 Sep 67.
2. A. R. Ferguson and G. B. Dantzig, "The Problem of Routing Aircraft—A Mathematical Solution," RM-1369, RAND Corp., 1 Sep 54.
3. ——, "The Problem of Routing Aircraft—A Mathematical Solution," Aeronaut. Eng. Rev., 14(4): 51–55 (Apr 55).
4. A. Charnes, W. W. Cooper, and A. Henderson, Introduction to Linear Programming, John Wiley and Sons, Inc., New York, 1953.
5. G. E. Hadley, Linear Programming, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1962.
6. A. J. Goldman and A. W. Tucker, "Theory of Linear Programming," in H. W. Kuhn and A. W. Tucker (eds), Linear Inequalities and Related Systems, Annals of Math. Studies, No. 38, Princeton University Press, 1956, pp 53–97.
7. R. Arms and J. Bracken, "An Improved Computer Program for the Simplex Method," unpublished notes, Research Analysis Corp., Sep 67.

ADDITIONAL REFERENCES

Dantzig, G. E., Linear Programming and Extensions, Princeton University Press, 1963.

Gass, S. I., Linear Programming: Methods and Applications, McGraw-Hill Book Co., New York, 1964, 2d ed.

Simonnard, S., Programmation Linéaire, Dunod, Paris, 1962.

RAC DISTRIBUTION LIST A

1 Sep 68

(Less these deletions: See additions on third page)

Address code	Agency	Number of copies
	DEPARTMENT OF DEFENSE	
D3	Armed Forces Staff College	1
A9	Assistant Secretary of Defense, Systems Analysis	2
A3	Assistant Secretary of Defense, International Security Affairs	1
C2	Defense Atomic Support Agency	1
C4	Defense Communications Agency (DCA)	1
C6	Defense Document Center (DDC)	20
C8	Defense Intelligence Agency, Medical Branch, Environment Division	1
C9	Defense Intelligence Agency (DIA SA-2B)	1
A2	Director of Defense Research and Engineering	1
C1	Advanced Research Projects Agency	1
B5	Weapons Systems Evaluation Group	1
D2	Industrial College of the Armed Forces	1
F3	Joint Support Command	1
C3	National Security Agency	1
D1	National War College	1
	JOINT CHIEFS OF STAFF	
B2	Joint Chiefs of Staff	1
B4	Joint War Game Control Group	1
B6	Special Study Group	1
	JOINT COMMANDS	
L5	Commander in Chief, Alaska (CINCAL)	1
L12	Commander in Chief, Europe (CINCEUCOM)	1
L6	Commander in Chief, Pacific (CINCPAC)	1
L7	Supreme Headquarters, Allied Powers, Europe (USNMR)	1
L16	US Strike Command, MacDill Air Force Base	1
	HEADQUARTERS, DEPARTMENT OF THE ARMY	
	Secretary of the Army	
E2	Office, Under Secretary of the Army, Operations Research	1
E4	Assistant Secretary of the Army (R&D)	1
M26	Data Systems Office, Research Division, Army Material Command	1
	CHIEF OF STAFF, US ARMY	
E33	Assistant Vice Chief of Staff of Army	5
	DEPARTMENT OF THE ARMY	
E23	Assistant Chief of Staff for Force Development	1
R11	Army Concepts Team, Vietnam	1
O24	Army War College	1
E10	Assistant Chief of Staff, Intelligence	2
O9	Army Intelligence School, Ft Holabird	1
E15	The Army Library, Attn: ASDIRS	1
E19	Chief of Communications-Electronics	1
O27	Academic Computer Center, West Point, N. Y.	1
E17	Chief of Engineers	1
E20	Director of Transportation	1
E27	Chief of Research and Development	1
R4	US Army Behavioral Science Research Laboratory	1
E21	Chief of Support Services	1
E14	Comptroller	1
E9	Deputy Chief of Staff for Logistics, Doctrine Branch	1
E8	Deputy Chief of Staff for Military Operations	2(4)[a]
P9	Army Strategic and Tactical Analysis Group	1
E7	Deputy Chief of Staff for Personnel	1
E18	Surgeon General	1
K11	Defense Logistics Studies Information Exchange, US Army Logistics Management Center, Ft Lee	2
	ARMIES, CONUS AND OVERSEAS	
H2	First	1
H3	Third	1
H4	Fourth	1
H5	Fifth	1
H6	Sixth	1
H8	Eighth	1

[a]Send 4 copies of RAC Strategic Studies Department documents, but only 2 copies of all other RAC documents.

RAC Distribution List A (continued)

Address code	Agency	Number of copies
	ARMY COMMANDS, AGENCIES, AND ELEMENTS	
L1	Army Air Defense Command	1
L10	Commanding General, US Army, Europe, Attn: AEAGC	1
J11	Assistant Chief of Staff, Developments, CDA	1
M19	US Army Strategic Communications Command, Technical Reference Center	1
L2	Combat Developments Command	1
L8	US Army, Alaska	1
L9	US Forces Southern Command, Europe	1
L11	US Army, Pacific	1
	ARMY MATERIEL COMMAND	
M7	Electronics Command, Ft Monmouth	1
I5	Edgewood Arsenal, Operations Research Group	1
J4	Armor Board, Ft Knox	1
J5	Artillery Board, Ft Sill	1
J8	Infantry Board, Ft Benning	1
J3	Arctic Test Center, Alaska	1
R6	Dugway Proving Ground	1
R9	White Sands Missile Range	1
M24	Weapons Command, Rock Island, Ill.	1
M22	Commanding General, US Army Aviation Materiel Command	1
	SEPARATE ACTIVITIES AND LABORATORIES	
M17	Ordnance Center and School	1
	COMBAT DEVELOPMENTS COMMAND SUBELEMENTS	
P2	Combat Arms Group, Ft Leavenworth	1
G2	Air Defense Agency, Ft Bliss	1
G3	Armor Agency, Ft Knox	1
G4	Artillery Agency, Ft Sill	1
G5	Aviation Agency, Ft Rucker	1
G7	Chemical, Biological, and Radiological Agency, Ft McClellan	1
G11	Communications-Electronics Agency, Ft Monmouth	1
G12	Engineer Agency, Ft Belvoir	1
G13	Infantry Agency, Ft Benning	1
G20	Intelligence Agency, Ft Holabird	1
P3	Combat Service Support Group, Ft Lee	1
O26	Institute of Advanced Studies, Carlisle Barracks	1
L44	Institute of Land Combat, Ft Belvoir	1
L45	Institute of Systems Analysis, Ft Belvoir	1
	SCHOOLS, ARMY	
O2	Air Defense, Ft Bliss	1
O3	Armor, Ft Knox	1
O4	Artillery and Guided Missile, Ft Sill	1
O15	Army Management, Ft Belvoir	1
O16	Army School, Europe	1
O17	Chemical, Biological, and Radiological, Campbell	1
O5	Chemical Corps, Ft McClellan	1
O14	Civil Affairs, Ft Gordon	1
O11	Command and General Staff, Ft Leavenworth	1
O6	Engineer (ESSY-L), Ft Belvoir	1
O7	Finance, Ft Benjamin Harrison	1
O10	Medical Field Service, Brooke	1
O19	Ordnance Guided Missile, Redstone Arsenal, Ala.	1
O21	Quartermaster, Ft Lee	1
O23	Special Warfare, Ft Bragg	1
O25	Transportation, Ft Eustis	1
O13	School of Americas	1
O8	Infantry School, Ft Benning	1
	US AIR FORCE	
T4	Headquarters, US Air Force (AFCSAI)	1
T8	Air University Library, Maxwell Field	1
	US MARINES	
S23	Dir MCLFDC, US Marine Corps School, Quantico, Va.	1
	US NAVY	
S1	Chief of Naval Operations, OP-96	1
S2	Chief of Naval Operations, OP03EG-CNO	1
S9	Naval War College, Newport, R.I.	1

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Research Analysis Corporation McLean, Virginia 22101	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

AN ALGORITHM FOR A SPECIAL CLASS OF GENERALIZED TRANSPORTATION-TYPE PROBLEMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical paper

5. AUTHOR(S) (First name, middle initial, last name)

Ronald L. Arms

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1968	52	10

8a. CONTRACT OR GRANT NO. DA 44-188-ARO-1 b. PROJECT NO. 008.134 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) RAC-TP-337 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office, Under Secretary of the Army Arms Control and Disarmament Agency

13. ABSTRACT The algorithm described in this paper is used to solve a special class of linear programming problems characterized by constraint coefficient matrices having generalized transportation structure. Specifically, n available resources are allocated to m capacity-limited operations (where the operational capability of assigning the jth resource to the ith operation is known) such as to maximize the total profit for the system. The row-and-column structure of such problems permits an algorithm more efficient than the general simplex algorithm to be used to solve moderate-sized problems (problems where loop-tracing techniques or equivalent schemes are not required). It is not required in the problem statement that all the resources be allocated or that all operations be performed to capacity limits. It is characteristic of such problems, however, that the optimizing solution usually requires that at least one of the two conditions holds, i.e., either supply or demand is exhausted. The paper contains a description of the algorithm, a computer program, an example illustrating its application, and some comparisons with the general simplex algorithm in solving the same problem. ( )

DD FORM 1 NOV 65 **1473**