MANAGEMENT SCIENCE
Vol. 10, No. 3, April, 1964
Printed in U.S.A.

A PRIMAL METHOD FOR THE ASSIGNMENT AND TRANSPORTATION PROBLEMS*†

M. L. BALINSKI[1] AND R. E. GOMORY[2]

This paper describes a simple calculation for the assignment and transportation problems which is "dual to" the well-known Hungarian Method. While the Hungarian is a dual method, this method is primal and so gives a feasible assignment at each stage of the calculation. Bounds on the number of steps required for the assignment and transportation problems are given. They are the same as the best bounds known for the Hungarian Method.

1. Introduction

Perhaps the best known, most widely used, and most written about method for solving the assignment problem is the "Hungarian Method". Originally suggested by Kuhn [8] in 1955, it has appeared in many variants (e.g., [4], [5], [9], [10], [11]). It provided essential ideas for the early methods used in solving network flow problems [5], it has been extended to solve the transportation problem [5], [11], and it has even been "generalized" to solve the linear programming problem [3]. It is a dual method with a feasible assignment being obtained only at the last computational step.

This paper presents a primal method for the assignment and transportation problems which is a method "dual to" the Hungarian Method. Where the Hungarian Method provides at each intermediate computational step a dual feasible vector (U, V) and a corresponding (infeasible) primal vector X orthogonal to (U, V), the present method provides at each step a feasible X (a complete assignment or transportation solution) and a corresponding orthogonal (U, V). In addition to the advantage of being primal, and so providing a constantly improving solution, the method seems to be extremely simple to describe, explain, and—it would seem—program.

With this method, we are able to bound the number of steps required to solve the assignment and transportation problems. Different bounds can be obtained from different variants of the procedure. The best bounds are $n(n + 1)/2$ labeling passes for the $n \times n$ assignment problem and $(\sum_j c_j)$ min (m, n) passes for the m source n sink transportation problem with total demand of $\sum_j c_j$. These bounds are the same as the best known bounds for the Hungarian Method.

Remarkably enough, the variant that gives the best bounds requires a special rule of choice very similar to the simplex method's "most negative column" rule.

* Received July 1963.

† This research was supported in part by the Office of Naval Research under Contract No. Nonr-3775(00), NR 047040.

[1] University of Pennsylvania. This author's work was supported by the Princeton University, Office of Naval Research Logistics Project.

[2] Thomas J. Watson Research Center, IBM Corporation.

Since many methods for obtaining optimal solutions to assignment problems have been proposed, it is natural to ask if the primal method described here is really new. To the best of our knowledge, it really is new. The only other primal method maintaining orthogonality is the primal simplex method; but there a key idea is that of a *basic* feasible solution. This makes application of a simplex method to the assignment problem practicularly messy and difficult due to "degeneracy". In contrast, the idea of a basic solution is unnecessary for the method described below. The other apparent candidates for comparison are the "out-of-kilter method" [6] or the method of Gleyzal [7]. But these are not methods which preserve—or attempt to preserve—"orthogonality" between current values for primal and dual variables. Moreover, in the out-of-kilter method, the "kilter numbers" are monotone non-decreasing, and this is clearly not the case in the primal method given below. The method of Beale [2], which is also applicable to problems with convex costs $a_{ij}(x_{ij})$, shares with this method its primal character and its avoidance of the notion of basis. Again, orthogonality is not preserved.

These rather sketchy and general remarks will be amplified in a planned paper devoted to classifying and explaining the connections between the various known computational procedures (simplex, out-of-kilter, etc.) for the assignment problem, the transportation problem, and more general linear programs.

2. A Primal Method for the Assignment Problem

As is well known, the $n \times n$ assignment problem is equivalent to the dual linear programs

$$\begin{array}{lll} (1) & \text{minimize} & \alpha(X) = \sum_{i,j} a_{ij}x_{ij} \\ & \text{constrained by} & \sum_j x_{ij} = 1, \quad \text{all} \quad i \\ & & \sum_i x_{ij} = 1, \quad \text{all} \quad j \\ & & x_{ij} \geqq 0 \quad \text{all} \quad (i,j) \end{array}$$

or

$$\begin{array}{lll} (2) & \text{maximize} & \beta(U, V) = \sum_i u_i + \sum_j v_j \\ & \text{constrained by} & u_i + v_j \leqq a_{ij}, \quad \text{all} \quad (i,j). \end{array}$$

The criterion for optimality for an X satisfying the constraints of (1) and (U, V) satisfying the constraints of (2) is the orthogonality condition:

$$(3) \qquad (a_{ij} - u_i - v_j)x_{ij} = 0 \quad \text{all} \quad (i,j).$$

The primal method described below obtains, in a finite number of steps, optimal solutions to the pair of dual programs (1) and (2), and at every step of this method an orthogonal pair is at hand. This means that X is a feasible assignment (X satisfies (1) and every component is 0 or 1) and U, V satisfies conditions (3).

Suppose, then, that such a pair is at hand at some stage of the computation.

If $u_i + v_j \leqq a_{ij}$ for all (i, j), then X and U, V constitute optimal solutions to their respective problems.

Otherwise, there exists some entry (k, l) with

$$u_k + v_l > a_{kl}\,. \tag{4}$$

The rules specified below then show that in one computational step *either* a new feasible assignment X' and orthogonal mate U', V' is obtained which satisfy

$$\alpha(X') < \alpha(X), \tag{5a}$$

$$u'_i + v'_j \leqq a_{ij} \quad \text{if} \quad u_i + v_j \leqq a_{ij}\,, \tag{5b}$$

$$u'_k + v'_l \leqq a_{kl}\,, \tag{5c}$$

or a new orthogonal mate U', V' for X is obtained which satisfies

$$u'_i + v'_j \leqq a_{ij} \quad \text{if} \quad u_i + v_j \leqq a_{ij}\,, \tag{6a}$$

$$u'_k + v'_l - a_{kl} < u_k + v_l - a_{kl}\,. \tag{6b}$$

Thus, either a truly better assignment X' is found, or a new orthogonal mate which "better meets" (in the sense of (6a) and (6b)) the constraints of the dual problem (2).[3]

A Computational Step

Given an orthogonal pair, X (which we will consider to be in matrix form) and U, V (or u_i associated with each row i of X and v_j associated with each column j of X) with specified entry (k, l) such that $u_k + v_l > a_{kl}$, use the following labelling procedure until further labelling is impossible.

1(a) Assign the label $\{k\}$ to column l.

(b) If column j has been labelled, then label with $\{j\}$ that one row i having $x_{ij} = 1$.

(c) If row i, $i \neq k$, has been labelled, then label with $\{i\}$ all columns j having $u_i + v_j = a_{ij}$ which have not yet been labelled.

Let $\{R(i)\}$ and $\{Q(j)\}$ denote, respectively, row i's label and column j's label. *Either* 2) row k receives a label $\{R(k)\}$ *or* 3) not.

2. If row k receives a label, then by following the labelling in "backward order" a sequence of entries x_{ij} forming a loop is uniquely defined:

$$x_{kl} = 0, \qquad x_{k,R(k)} = 1, \qquad x_{Q(R(k)),R(k)} = 0, \cdots, x_{kl} = 0. \tag{7}$$

Number $x_{kl} = 0$ the first, $x_{k,R(k)} = 1$ the second, etc., entries in the loop. Then every odd numbered entry of the loop is zero, and every even numbered entry of the loop is one.

Define

[3] The same primitive idea is used in [1] and [3] for the general linear programming problem: if improvement in the primal (or dual) problem is impossible due to degeneracy, then improvement is attempted in a dual (or primal) problem.

$$(8)\qquad \begin{cases} x'_{ij} = 1 & \text{if} \quad x_{ij} \text{ is an odd numbered loop entry,} \\ x'_{ij} = 0 & \text{if} \quad x_{ij} \text{ is an even numbered loop entry,} \end{cases}$$

$$(9)\qquad x'_{ij} = x_{ij} \quad \text{if} \quad x_{ij} \text{ is not in the loop.}$$

Also, define $\epsilon = u_k + v_l - a_{kl} > 0$ and

$$(10)\qquad u'_i = u_i \quad \text{all} \quad i,$$

$$(11)\qquad v'_j = v_j \quad \text{all} \quad j \neq l, \qquad v'_l = v_l - \epsilon$$

$X' = (x'_{ij})$ is clearly a new assignment. $U' = (u'_i)$, $V' = (v'_j)$ is orthogonal to X' because if $x'_{ij} = 1$ $(i, j) \neq (k, l)$ then $u'_i = u_i$ and $v'_j = v_j$ so $u'_i + v'_j = u_i + v_j = a_{ij}$; while for $x'_{kl} = 1$, $u'_k + v'_l = a_{kl}$ due to the choice of ϵ. Thus orthogonality of X and U, V and the particular choice of ϵ assure the orthogonality of X' and U', V'. Also, $\alpha(X') < \alpha(X)$ because

$$\alpha(X) - \alpha(X') = \textstyle\sum_I a_{ij} - \sum_{I'} a_{ij}$$

where $I = \{(i, j) \mid (i, j)$ in the loop (7) and $x_{ij} = 1\}$ and $I' = \{(i, j) \mid (i, j)$ in the loop (7) and $x'_{ij} = 1\}$. Hence, there being two such entries in each row and column of the loop

$$\begin{aligned} \alpha(X) - \alpha(X') &= \textstyle\sum_I (a_{ij} - u_i - v_j) - \sum_{I'} (a_{ij} - u_i - v_j) \\ &= -(a_{kl} - u_k - v_l) = \epsilon > 0, \end{aligned}$$

since for all entries in the loop (IUI') except (k, l), $a_{ij} - u_i - v_j = 0$.

Finally (5b) holds because all totals $u_i + v_j$ are either unchanged or decreased. (5c) holds due to the orthogonality.

This completes a step in the case where row k is labelled.

3. If row k is not labelled, let $\epsilon = \min_J \{a_{ij} - u_i - v_j\}$ where $J = \{(i, j) \mid u_i + v_j \leqq a_{ij}$ and row i is labelled and column j is not labelled$\}$; or $\epsilon = u_k + v_l - a_{kl}$ if J is a void set of entries. Then define $X' = X$ and

(12) $u'_i = u_i + \epsilon$ if row i is labelled, $u'_i = u_i$ otherwise;

(13) $v'_j = v_j - \epsilon$ if column j is labelled, $v'_j = v_j$ otherwise. If $u'_k + v'_l > a_{kl}$ (Case 3a), specify entry (k, l) for the next step[4]; otherwise (Case 3b), specify any entry (i, j) with $u'_i + v'_j > a_{ij}$. If all $u'_i + v'_j \leqq a_{ij}$, stop.

$X' = X$ and U', V' are orthogonal because if $x'_{ij} = x_{ij} = 1$ either both u_i and v_j were labelled or neither was labelled so $u'_i + v'_j = u_i + v_j = a_{ij}$. (6a) holds because the only totals $u_i + v_j$ that are increased are those whose indices (i, j) are in J. And ϵ was chosen small enough that the corresponding inequalities remain satisfied. Also $\epsilon > 0$ which implies (6b).

If the computational step produces Case 3a, then at the next step an addi-

[4] Notice that in this case the labelling can be kept unchanged to begin the next step. This gives the same result as starting over would for those zeros *used* in the previous labelling are still zeros, since both their row and their column were labelled.

tional column is always labelled. This is because we have for one of the $(i, j) \epsilon J$, $u'_i + v'_j = a_{ij}$.

To *initiate* the algorithm, take any orthogonal pair with X a feasible assignment; e.g., $X = I(x_{ii} = 1 \text{ all } i; x_{ij} = 0 \; i \neq j)$ and $U = 0$, $V = (v_i) = (a_{ii})$. This completes the description of the algorithm.

The primal method described above must terminate in a finite number of steps for in one step a new orthogonal pair is obtained which satisfies either (5a, b, c) or (6a, b). The second alternative can occur at most n times in succession, since after each occurrence of this alternative, at least one more column is labelled. But each occurrence of the first alternative makes at least one more dual inequality satisfied; thus, since once a dual inequality is satisfied, it always remains satisfied (by (5b) and (6a)), and there are only a finite number n^2 of such inequalities, the method cannot take more than $n \cdot n^2 = n^3$ steps.

To reduce this bound to $0(n^2)$ one refinement is needed in the choice of the origin of labelling.

At every point in the description of the algorithm where the instructions say to choose an a_{kl} with $u_k + v_l > a_{kl}$ as the origin for labelling, the k should be chosen to maximize $(u_k + v_l - a_{kl})$ in column l. Specifically,

(a) in starting, choose any column containing an element a_{kl} with $u_k + v_l > a_{kl}$; then chose the row so that the difference $(u_k + v_l - a_{kl})$ is maximized.

(b) The same procedure should be followed after completing Step (2).

(c) After completing a Step (3), if $\max_i (u'_i + v'_l - a_{il}) > 0$, keep column l and choose the k that maximizes. Otherwise, choose any column just as after Step (2) or at the start.

To get the improved bound, we reason as follows:

If (k, l) is chosen to maximize $\epsilon = (u_i + v_l - a_{il})$ and Step (2) occurs, the change in the dual variables will make $u'_i + v'_l \leqq a_{il}$ for *all* entries i in column l. This is easily verified. So in one labelling pass, we satisfy all inequalities in one column.

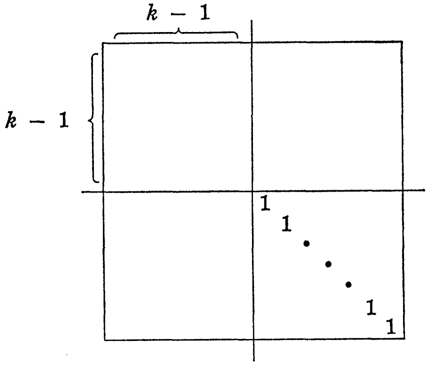

FIGURE 1

If row k is not labelled so Step (3) occurs, there are two possibilities (if it is not already true that $u'_i + v'_l \leqq a_{il}$ all i):

3(α) k still maximizes in its column; i.e., $k = i$ maximizes $u'_i + v'_l - a_{il} > 0$. In this case (k, l) is still the origin of labelling for the next step.

3(β) (k', l), $k' \neq$ the previous k, maximizes $u'_i + v'_l - a_{il} > 0$ over column l and so becomes the origin of labelling for the next step.

If (3α) occurs $(n - 1)$ times in succession, the only unlabelled column left would be that column l' having $x_{kl'} = 1$. Otherwise, k would be labelled. So Case (2) will occur at the next labelling. Thus, after n labellings, all inequalities in the column will be satisfied.

If (3α) occurs $0, \cdots, (n - 2)$ times and is followed by (3β), the very next labelling will bring on Case (2). For the maximum can shift from k to k' upon changing the variables after a Case (3α) only if row k' was labelled while (by assumption) row k was not. The next labelling, which starts from (k', l), will make use of all the old labels since the column is unchanged, so k' will be labelled again and Case (2) occurs. Again in n steps, all column l inequalities are satisfied.

Proceeding in this way, all inequalities will be satisfied after at most n^2 steps. A further modification can bring the bound down to $n(n + 1)/2$ steps.

Let us assume a starting solution $x_{ii} = 1$, all i. An appropriate set of dual variables satisfying the orthogonality condition is $u_i = a_{ii}$, all i, $v_j = 0$ all j. Let us assume that our calculation has reached a point where all dual inequalities $u_i + v_j \leqq a_{ij}$ are satisfied for $j < k$. We will show how at most k labellings will extend this condition to column k.

The calculation follows the procedure just described with the maximizing method of row choice. The entry maximizing $(u_i + v_k - a_{ik})$ in column k is selected and labelling proceeds exactly as before, except for one change. Columns $k' > k$ are never labelled.

The following consequences are easily seen:

(i) After at most k steps, all inequalities in column k are satisfied. This is by the same reasoning as above but confined to the first k columns.

(ii) Rows with index $i > k$ are never reached by the labelling process. This is because of the diagonal 1's in these rows and the fact that columns with index $j > k$ are not labelled.

Statement (i) shows that, after repeating the process on columns $k + 1$, etc., after at most $1 + 2 + \cdots k + \cdots n = n(n + 1)/2$ labellings all inequalities will have been satisfied. If they are kept that way, the problem will have been solved in at most $n(n + 1)/2$ steps.

Statement (ii) implies that inequalities indexed in the lower right of Figure 1 will be completely unchanged and, hence, orthogonality preserved. It also shows that, since entries in the lower left box are in unlabelled rows, these cannot become unsatisfied. Those in the upper left are preserved as here our previous argument applies. Satisfied inequalities in the upper right box may be lost, but to make the argument go through, we need preserve only those in columns $j \leqq k$, which we do.

3. Example

Consider the four by four assignment problem defined by the cost matrix

$$A = \begin{array}{|cccc|} \hline 2 & 3 & 1 & 1 \\ 5 & 8 & 3 & 2 \\ 4 & 9 & 5 & 1 \\ 8 & 7 & 8 & 4 \\ \hline \end{array}$$

In the computation below, the numbers on the left-hand margin of each "box" are current values of u_i and, in brackets, any labels they may receive; the numbers on the top margin are current values of the v_j and, in brackets, the labels they are given.

Within the box an uncircled number in the (i, j) position is $a_{ij} - u_i - v_j$, and the corresponding x_{ij} is zero; a circled number in the (i, j) position means $a_{ij} - u_i - v_j = 0$ and $x_{ij} = 1$.

A star indicates the origin of labelling.

1) Start with $x_{ii} = 1, u_i = 0, v_i = a_{ii}$. Label, Case (3a) occurs.

$$\begin{array}{r|cccc|} & & \{1\} & & \\ & 2 & 8 & 5 & 4 \\ \hline 0 & \textcircled{1} & -5^* & -4 & -3 \\ \{2\}\ 0 & 3 & \textcircled{1} & -2 & -2 \\ 0 & 2 & 1 & \textcircled{1} & -3 \\ 0 & 6 & -1 & 3 & \textcircled{1} \\ \hline \end{array}$$

$$\epsilon = 3$$

2) Change variables, continue the labelling, Case (2) occurs.

$$\begin{array}{r|cccc|} & \{2\} & \{1\} & & \\ & 2 & 5 & 5 & 4 \\ \hline \{1\}\ 0 & \textcircled{1} & -2^* & -4 & -3 \\ \{2\}\ 3 & 0 & \textcircled{1} & -5 & -5 \\ 0 & 2 & 4 & \textcircled{1} & -3 \\ 0 & 6 & 2 & 3 & \textcircled{1} \\ \hline \end{array}$$

$$\epsilon = 2$$

3) Change variables, label, Case (3a) occurs.

$$\begin{array}{r|cccc|} & & & \{2\} & \\ & 2 & 3 & 5 & 4 \\ \hline 0 & 0 & \textcircled{1} & -4 & -3 \\ 3 & \textcircled{1} & 2 & -5^* & -5 \\ \{3\}\ 0 & 2 & 6 & \textcircled{1} & -3 \\ 0 & 6 & 4 & 3 & \textcircled{1} \\ \hline \end{array}$$

$$\epsilon = 2$$

4) Change variables, continue the labelling, Case (2) occurs.

$$\begin{array}{cc|cccc|}
 & & \{3\} & & \{2\} & \\
 & & 2 & 3 & 3 & 4 \\ \hline
 & 0 & 0 & \textcircled{1} & -2 & -3 \\
\{1\} & 3 & \textcircled{1} & 2 & -3^* & -5 \\
\{3\} & 2 & 0 & 4 & \textcircled{1} & -5 \\
 & 0 & 6 & 4 & 5 & \textcircled{1} \\ \hline
\end{array}$$

$$\epsilon = 3$$

5) Change variables, label, Case (3a) occurs.

$$\begin{array}{cc|cccc|}
 & & & & & \{2\} \\
 & & 2 & 3 & 0 & 4 \\ \hline
 & 0 & 0 & \textcircled{1} & 1 & -3 \\
 & 3 & 0 & 2 & \textcircled{1} & -5^* \\
 & 2 & \textcircled{1} & 4 & 3 & -5 \\
\{4\} & 0 & 6 & 4 & 8 & \textcircled{1} \\ \hline
\end{array}$$

$$\epsilon = 4$$

6) Change variables, continue labelling, Case (3b) occurs.

$$\begin{array}{cc|cccc|}
 & & \{1\} & \{4\} & & \{2\} \\
 & & 2 & 3 & 0 & 0 \\ \hline
\{2\} & 0 & 0 & \textcircled{1} & 1 & 1 \\
 & 3 & 0 & 2 & \textcircled{1} & -1^* \\
\{1\} & 2 & \textcircled{1} & 4 & 3 & -1 \\
\{4\} & 4 & 2 & 0 & 4 & \textcircled{1} \\ \hline
\end{array}$$

$$\epsilon = 1$$

7) Change variables, continue labelling, Case (2) occurs.

$$\begin{array}{cc|cccc|}
 & & \{1\} & \{4\} & \{1\} & \{3\} \\
 & & 1 & 2 & 0 & -1 \\ \hline
\{2\} & 1 & 0 & \textcircled{1} & 0 & 1 \\
\{3\} & 3 & 1 & 3 & \textcircled{1} & 0 \\
\{1\} & 3 & \textcircled{1} & 4 & 2 & -1^* \\
\{4\} & 5 & 2 & 0 & 3 & \textcircled{1} \\ \hline
\end{array}$$

8) Solution given by circled 1's.

$$\begin{array}{c|cccc|}
 & 1 & 2 & 0 & -2 \\ \hline
1 & \textcircled{1} & 0 & 0 & 2 \\
3 & 1 & 3 & \textcircled{1} & 1 \\
3 & 0 & 4 & 2 & \textcircled{1} \\
5 & 2 & \textcircled{1} & 3 & 1 \\ \hline
\end{array}$$

4. A Primal Method for the Transportation Problem

Consider, now, the transportation problem which we take to be

(14) minimize $\alpha(X) = \sum_{i,j} a_{ij}x_{ij}$

constrained by[5]

$$\sum_j x_{ij} = b_i , \qquad \sum_i x_{ij} = c_j , \qquad x_{ij} \geqq 0.$$

The dual problem is

$$\text{(15)} \quad \text{maximize} \qquad \beta(U, V) = \sum_i b_i u_i + \sum_j c_j v_j$$

constrained by

$$u_i + v_j \leqq a_{ij} , \quad \text{all} \quad (i, j).$$

The primal method for the transportation problem described below is a direct extension of the method for the assignment problem given in Section 2.

If in our transportation problem we replace each sink by c_j identical sinks, each of demand 1, we have a problem to which the assignment method just described would apply. One can verify that the unit bound on sinks alone (not sources) is sufficient to justify the reasoning and procedures given above. The assignment calculation on this enlarged matrix of $\sum c_j$ columns would, however, involve a great deal of redundancy. Removing this redundancy, one obtains the transportation algorithm described below which is directly applicable to the original problem (14).

The algorithm may be described as a sequence of at most n stages, one for each column, with each stage made up of steps analogous to those defined in Section 2.

At the beginning of each stage, an orthogonal pair, X feasible for (14) and orthogonal U, V, i.e., satisfying

$$\text{(16)} \qquad (a_{ij} - u_i - v_j)x_{ij} = 0 \quad \text{all} \quad (i, j)$$

is at hand. If $u_i + v_j \leqq a_{ij}$ for all (i, j), then X and U, V constitute optimal solutions to their respective problems.

Otherwise, there exists some entry (k, l) with

$$\text{(17)} \qquad u_k + v_l > a_{kl} .$$

The rules specified below then show that in one computational *stage l* (for column l) a new orthogonal pair, X' and U', V' with X' feasible, is obtained which satisfies $\alpha(X') \leqq \alpha(X)$ and

$$\text{(18a)} \qquad u'_i + v'_j \leqq a_{ij} \quad \text{if} \quad u_i + v_j \leqq a_{ij}$$

$$\text{(18b)} \qquad u'_i + v'_l \leqq a_{il} \quad \text{all} \quad i.$$

Thus, each stage l, which takes at most mc_l steps, assures that at least one more column of dual inequalities is satisfied (18b). This is the same as in the variant of the method for an m by m assignment problem which leads to at most m steps being required to satisfy one more column of dual inequalities, i.e., to complete a "stage".

At every *step* within a stage l (initiated by an inequality (17)) an equivalent

[5] It is assumed that the b_i and c_j are nonnegative integers with $\sum_i b_i = \sum_j c_j$.

derived transportation problem is considered which is identical to the problem (14) except that column l is replaced by two columns, $l(1)$ and $l(2)$, having nonnegative integer "demands" $c_{l(1)}$ and $c_{l(2)}$ $(c_l = c_{l(1)} + c_{l(2)})$, with "costs" $a_{il(1)} = a_{il(2)} = a_{il}$ all i. At every step there is at hand a derived transportation problem and its dual, together with a feasible $\hat{X}$ for the primal and orthogonal $\hat{U}$, $\hat{V}$. In one step *either* a new derived transportation problem is obtained with demands c'_{l_1} and c'_{l_2} $(c_l = c'_{l_1} + c'_{l_2})$ and an orthogonal pair of solutions, $\hat{X}'$ feasible and $\hat{U}'$, $\hat{V}'$ which satisfies

$$\text{(19a)} \qquad \alpha(\hat{X}') < \alpha(\hat{X})$$

$$\text{(19b)} \qquad c'_{l(1)} > c_{l(1)} \qquad (c'_{l(2)} < c_{l(2)})$$

$$\text{(19c)} \qquad \hat{u}'_i + \hat{v}'_j \leqq a_{ij} \quad \text{if} \quad \hat{u}_i + \hat{v}_j \leqq a_{ij}$$

or a new orthogonal mate $\hat{U}'$, $\hat{V}'$ is found for the same derived transportation problem and a feasible X which satisfies (19c) and

$$\text{(20)} \qquad \hat{u}'_k + \hat{v}^i_{l(2)} - a_{kl(2)} < \hat{u}_k + \hat{v}_{l(2)} - a_{kl(2)}\,, \qquad \text{for some } k.$$

A Computational Stage l

Given an orthogonal pair for (14) and (15), X feasible and U, V, and a specified column l such that $u_k + v_l - a_{kl} = \max_i \{u_i + v_l - a_{il}\} > 0$, define an initial derived transportation problem and orthogonal pair, $\hat{X}$ feasible and $\hat{U}$, $\hat{V}$, by setting $c_{l(1)} = 0$, $c_{l(2)} = c_l$; $\hat{x}_{il(1)} = 0$ and $\hat{x}_{il(2)} = x_{il}$ all i, $\hat{x}_{ij} = x_{ij}$ otherwise; $\hat{u}_i = u_i$ all i, $\hat{v}_{l(1)} = a_{kl} - u_k$, $\hat{v}_{l(2)} = v_l$, and $\hat{v}_j = v_j$ otherwise. Note that the choice of $\hat{v}_{l(1)}$ satisfies all dual inequalities in $l(1)$.

A Computational Step.

Given an orthogonal pair, $\hat{X}$ feasible and $\hat{U}$, $\hat{V}$ with $\max_i \{\hat{u}_i + \hat{v}_{l(2)} - a_{il(2)}\} = \hat{u}_k + \hat{v}_{l(2)} - a_{kl(2)} > 0$ and $c_{l(2)} > 0$, use the following labelling procedure until further labelling is impossible:

1.(a) Assign the label $\{k\}$ to column $l(2)$.

(b) If column j has been labelled, then label with $\{j\}$ all rows i having $\hat{x}_{ij} > 0$ which have not yet been labelled.

(c) If row i $(i \neq k)$ has been labelled, then label with $\{i\}$ all columns j having $\hat{u}_i + \hat{v}_j = a_{ij}$ which have not yet been labelled.

Either 2. row k receives a label *or* 3. it does not.

2. If row k receives a label a loop is uniquely defined:

$$\text{(21)} \qquad \hat{x}_{kl(2)} = 0, \quad \hat{x}_{kR(k)} > 0, \quad \hat{x}_{QR(k),R(k)} \geqq 0, \cdots, \hat{x}_{kl(2)} = 0.$$

Number $\hat{x}_{kl(2)}$ the first, $\hat{x}_{kR(k)} > 0$ the second, etc., entries in the loop. Then every odd numbered entry of the loop is nonnegative and every even numbered entry is positive. Let

$$\delta = \min \{\hat{x}_{ij} \mid \hat{x}_{ij} \text{ even numbered loop entry}\}.$$

δ is always $\geqq 1$.

Define

$$\text{(22a)}\qquad \begin{aligned} \hat{x}'_{kl(1)} &= \hat{x}_{kl(1)} \quad \text{if } \hat{x}_{kl(1)} \text{ is an even numbered loop entry,} \\ &= \hat{x}_{kl(1)} + \delta \quad \text{otherwise,} \end{aligned}$$

$$\text{(22b)}\qquad \hat{x}'_{kl(2)} = 0;$$

$$\text{(22c)}\qquad \begin{aligned} \hat{x}'_{ij} &= \hat{x}_{ij} + \delta \quad \text{if } \hat{x}_{ij} \text{ is an odd numbered loop entry,} \\ &= \hat{x}_{ij} - \delta \quad \text{if } \hat{x}_{ij}((i,j) \neq (k, l(1))) \text{ is an even numbered loop entry,} \\ &= \hat{x}_{ij} \quad \text{otherwise; and} \end{aligned}$$

$$\text{(23)}\qquad c'_{l(1)} = c_{l(1)} + \delta, \qquad c'_{l(2)} = c_{l(2)} - \delta \quad \text{and}$$

$$\text{(24)}\qquad \hat{u}'_i = u_i, \qquad \hat{v}'_j = v_j \quad \text{all } i \text{ and } j.$$

If $c'_{l(2)} \neq 0$, start a new computational step. If $c'_{l(2)} = 0$, stop; stage l is complete. Let $x_{il} = \hat{x}'_{il(1)}$, $x_{ij} = \hat{x}'_{ij}\, j \neq l$, and $u_i = \hat{u}'_i$, $v_l = \hat{v}'_{l(1)}$ to obtain an orthogonal pair for (14) and (15), X feasible and U, V. Start a new stage if necessary.

There are two points in this procedure that would seem to require explanation. One is in connection with orthogonality. The orthogonality between the new primal solution, $\hat{X}'$, and the dual variables $\hat{U}'$, $\hat{V}'$ is assured by the same reasoning as in the assignment case except in the case of the components $\hat{x}_{kl(1)}$ and $\hat{x}_{kl(2)}$. The component $\hat{x}_{kl(2)}$ is and remains zero, so it poses no problem. Similarly, as we will show, the current $a_{kl(1)} - \hat{u}_k - \hat{v}_{l(1)}$ is zero, which assures the orthogonality of the remaining component. To see this, we need the two following facts: (i) The $a_{il(1)} - \hat{u}_i - \hat{v}_{l(1)}$ and $a_{il(2)} - \hat{u}_i - \hat{v}_{l(2)}$ differ by a constant $\hat{v}_{l(2)} - \hat{v}_{l(1)}$ independent of the row i. (ii) The $a_{il(1)} - \hat{u}_i - \hat{v}_{l(1)}$ are nonnegative and at least one is zero. (This is so at the beginning of a stage, and just as in the assignment problem these properties are preserved.) We know that $a_{kl(2)} - \hat{u}_k - \hat{v}_{l(2)}$ is minimal in column $l(2)$. Because of (i) $a_{kl(1)} - \hat{u}_k - \hat{v}_{l(1)}$ is minimal in column $l(1)$ and because of (ii) it must be zero. This disposes of the orthogonality question.

The other point is the reason why the demands $c_{l(1)}$ and $c_{l(2)}$ can be changed by δ. We want to maintain the property that the demand of a sink equals the sum of the variables in its column. The changes in the primal variables given in (22) above are equivalent to a change of amount δ completely around the loop, as in the assignment problem, followed by a shift of δ between $\hat{x}_{kl(2)}$ and $\hat{x}_{kl(1)}$. The change around the loop leaves column sums unaltered so the net effect is a shift of amount δ between the demands supplied to sinks $l(1)$ and $l(2)$.

This completes a step in the case where row j is labelled.

3. If row k is not labelled, let $J = \{(i, j) \mid \hat{u}_i + \hat{v}_j \leqq a_{ij}$, row i labelled, column j not labelled$\}$

and

$$\text{(25)}\qquad \epsilon = \min\,[\min_J \{a_{ij} - \hat{u}_i - \hat{v}_j\},\, \hat{u}_k + \hat{v}_{l(2)} - a_{kl(2)}] > 0.$$

Define $\hat{X}' = \hat{X}$ and

$$\text{(26)}\qquad \begin{aligned} \hat{u}'_i &= \hat{u}_i + \epsilon \quad \text{if row } i \text{ labelled} \\ &= \hat{u}_i \quad \text{otherwise} \end{aligned}$$

$$\begin{aligned} \hat{v}'_j &= \hat{v}_j - \epsilon \quad \text{if column } j \text{ labelled} \\ &= \hat{v}_j \quad \text{otherwise.} \end{aligned} \tag{27}$$

If for some i, $\hat{u}'_i + \hat{v}'_{l(2)} > a_{il(2)}$ start another computational step. Labels are maintained. If $\hat{u}'_i + \hat{v}'_{l(2)} \leqq a_{il(2)}$ all i, stop; stage l is complete. Let $x_{il} = \hat{x}'_{l(1)} + \hat{x}'_{il(2)}$, $x_{ij} = \hat{x}'_{ij}$, $j \neq l$; and $u_i = \hat{u}'_i$, $v_l = \hat{v}'_{l(1)} = \hat{v}'_{l(2)}$ (see below) $v_j = \hat{v}'_j$, $j \neq l$, to obtain an orthogonal pair for (14), (15), X feasible and U, V. Start a new stage if necessary.

The reason for the equality of $\hat{v}'_{l(1)}$ and $\hat{v}'_{l(2)}$ when a stage ends in this way is as follows. Throughout the computation, any column j always has $a_{ij} - \hat{u}_i - \hat{v}_j = 0$ for some i; also, as we remarked above, corresponding entries $a_{il(1)} - \hat{v}_i - \hat{v}_{l(1)}$ and $a_{il(2)} - \hat{u}_i - \hat{v}_{l(2)}$ in columns $l(1)$ and $l(2)$ always differ by a constant $\hat{v}_{l(1)} - \hat{v}_{l(2)}$. Also throughout stage l, $a_{il(1)} - \hat{u}_i - \hat{v}_{l(1)} \geqq 0$ for all i. When stage l is completed as above $a_{il(2)} - \hat{u}_i - \hat{v}_{l(2)} \geqq 0$ for all i also. Thus, the minimal entries in columns $l(1)$ and $l(2)$ are both zero at this point. Since these columns differ by a constant, the constant can only be zero, i.e., $\hat{v}_{l(1)} - \hat{v}_{l(2)} = 0$.

To *initiate* the algorithm take any orthogonal pair for (14) and (15), with X a feasible solution containing no loops among its positive entries and U, V then computed to satisfy the orthogonality conditions (16).

The bound on the algorithm follows from the fact that every time the row of the labelling source is labelled, $c_{l(1)}$ is increased by δ an integer $\geqq 1$, and $c_{l(2)}$ is decreased the same amount. Therefore, after at most c_l of these events $c_{l(2)}$ represents a zero demand sink and can be dropped. (See end of stage under 2 above.) All dual inequalities in column $l(2)$ (and so in l) are now satisfied.

The same reasoning as in the assignment case shows that after at most m labellings the row of the labelling source must be labelled, or else all dual inequalities in its column (in this case $l(2)$) are already satisfied. (See end of stage under 3 above.)

Consequently, after at most mc_l labellings all dual inequalities in column l are satisfied (i.e., it takes at most mc_l steps to complete stage l). Since any satisfied dual inequality remains satisfied, it follows that after at most

$$m(\textstyle\sum_j c_j)$$

labellings all dual inequalities are satisfied and the optimal X is obtained.

Because the roles of source and sink are interchangeable, the final bound for an m source n sink transportation problem is

$$(\textstyle\sum c_j)\min(m, n).$$

It is also easy to describe an algorithm in which each stage aims at making at least one more dual inequality (k, l) satisfied given (17) initially—rather than aiming at making at least one more column l of dual inequalities satisfied given (17) initially. Such an algorithm would then be similar to the method first described for the m by m assignment problem where it takes at most m steps to satisfy one more dual inequality, and hence at most m^3 steps to solve the problem. The bound for such an algorithm becomes, however, $[\min(m^2, n^2)][\sum_j c_j]$.

Notice, finally, that the primal algorithm given in this section for the transportation problem may have at hand at any step of the procedure a feasible X for (14) (arising, perhaps, from an $\hat{X}$) which contains loops among its positive entries, i.e., a feasible X which is not a basic feasible solution for (14).

5. Example

Consider the transportation problem defined by

3	6	3	1	1	$b_1 = 4$
2	4	3	2	7	$b_2 = 5$
1	1	2	1	2	$b_3 = 6$
$c_1 =$ 2	$c_2 =$ 2	$c_3 =$ 3	$c_4 =$ 4	$c_5 =$ 4	

In the computation below the numbers on the left and top margins are the u_i and v_j, respectively, and in brackets any labels they may receive. Within a box an uncircled number in the (i, j) position is $a_{ij} - u_i - v_j$ and the corresponding x_{ij} is zero; a circled number in the (i, j) position is the value of the current x_{ij} and $a_{ij} - u_i - v_j$ is zero (where the x_{ij}, u_i and v_j are the primal and dual variables for the transportation problem under consideration). A star indicates the origin of labelling.

1. Start with an initial feasible X and orthogonal U, V. Select column 2.

	3	6	3	2	3
0	②	②	0	−1	−2
0	−1	−2	③	②	4
−1	−1	−4	0	②	④

Stage 2

1. Replace column 2 by two columns. Label, case 2 occurs.

		{1}		{1}	{2}		{3}
		3	2	3	2	3	6
{6}	0	②	4	0	−1	−2	②
{3}	0	−1	2	③	②	4	−2
{4}	−1	−1	0	0	②	④	−4*
			0= $c_{l(1)}$				2= $c_{l(2)}$

$\delta = 2$

2. Stage 2 complete since $c_{l(2)} = 0$. Select column 1.

	3	2	3	2	3	6
0	②	4	②	−1	−2	0
0	−1	2	①	④	4	−2
−1	−1	②	0	0	④	−4
		2= $c_{l(1)}$				0= $c_{l(2)}$

Stage 1

2. Label, case 3 occurs.

$$\begin{array}{rr|ccccc|c}
 & & \{2\} & & \{1\} & \{2\} & & \{3\} \\
 & & 2 & 2 & 3 & 2 & 3 & 3 \\
\hline
\{6\} & 0 & 1 & 4 & \textcircled{2} & -1 & -2 & \textcircled{2} \\
\{3\} & 0 & 0 & 2 & \textcircled{1} & \textcircled{4} & 4 & -1 \\
 & -1 & 0 & \textcircled{2} & 0 & 0 & \textcircled{4} & -1^* \\
\hline
 & & 0= & & & & & 2= \\
 & & c_{l(1)} & & & & & c_{l(2)}
\end{array}$$

$$\epsilon = 1$$

3. Label, case 2 occurs.

$$\begin{array}{rr|ccccc|c}
 & & & & \{1\} & & & \{2\} \\
 & & 1 & 2 & 2 & 1 & 3 & 2 \\
\hline
\{6\} & 1 & 1 & 3 & \textcircled{2} & -1 & -3 & \textcircled{2} \\
\{3\} & 1 & 0 & 1 & \textcircled{1} & \textcircled{4} & 3 & -1^* \\
 & -1 & 1 & \textcircled{2} & 1 & 1 & \textcircled{4} & 0 \\
\hline
 & & 0= & & & & & 2= \\
 & & c_{l(1)} & & & & & c_{l(2)}
\end{array}$$

$$\delta = 1$$

4. Label, case 3 occurs.

$$\begin{array}{rr|ccccc|c}
 & & & & \{1\} & & & \{2\} \\
 & & 1 & 2 & 2 & 1 & 3 & 2 \\
\hline
\{6\} & 1 & 1 & 3 & \textcircled{3} & -1 & -3 & \textcircled{1} \\
 & 1 & \textcircled{1} & 1 & 0 & \textcircled{4} & 3 & -1^* \\
 & -1 & 1 & \textcircled{2} & 1 & 1 & \textcircled{4} & 0 \\
\hline
 & & 1= & & & & & 1= \\
 & & c_{l(1)} & & & & & c_{l(2)}
\end{array}$$

$$\epsilon = 1$$

5. Stage 1 complete since column $l(2)$ entries all nonnegative. Select column 5.

$$\begin{array}{r|ccccc|c}
 & 1 & 2 & 1 & 1 & 3 & 1 \\
\hline
2 & 0 & 2 & \textcircled{3} & -2 & -4 & \textcircled{1} \\
1 & \textcircled{1} & 1 & 1 & \textcircled{4} & 3 & 0 \\
-1 & 1 & \textcircled{2} & 2 & 1 & \textcircled{4} & 1 \\
\hline
 & 1= & & & & & 1= \\
 & c_{l(1)} & & & & & c_{l(2)}
\end{array}$$

Stage 5

5. Label, case 3 occurs.

$$\begin{array}{rr|ccccc|c}
 & & & \{3\} & & & & \{1\} \\
 & & 1 & 2 & 1 & 1 & -1 & 3 \\
\hline
 & 2 & \textcircled{1} & 2 & \textcircled{3} & -2 & 0 & -4^* \\
 & 1 & \textcircled{1} & 1 & 1 & \textcircled{4} & 7 & 3 \\
\{6\} & -1 & 1 & \textcircled{2} & 2 & 1 & 4 & \textcircled{4} \\
\hline
 & & & & & & 0= & 4= \\
 & & & & & & c_{l(1)} & c_{l(2)}
\end{array}$$

$$\epsilon = 1$$

6. Continue the labelling, case 2 occurs.

$$\begin{array}{cc|ccccc|c}
 & & \{3\} & \{3\} & & \{3\} & & \{1\} \\
 & & 1 & 1 & 1 & 1 & -1 & 2 \\
\hline
\{1\} & 2 & \textcircled{1} & 3 & \textcircled{3} & -2 & 0 & -3^* \\
 & 1 & \textcircled{1} & 2 & 1 & \textcircled{4} & 7 & 4 \\
\{6\} & 0 & 0 & \textcircled{2} & 1 & 0 & 3 & \textcircled{4} \\
\hline
 & & & & & & 0= & 4= \\
 & & & & & & c_{l(1)} & c_{l(2)}
\end{array}$$

$\delta = 1$

7. Label, case 3 occurs.

$$\begin{array}{cc|ccccc|c}
 & & \{3\} & \{3\} & & \{2\} & & \{1\} \\
 & & 1 & 1 & 1 & 1 & -1 & 2 \\
\hline
 & 2 & 0 & 3 & \textcircled{3} & -2 & \textcircled{1} & -3^* \\
\{1\} & 1 & \textcircled{1} & 2 & 1 & \textcircled{4} & 7 & 4 \\
\{6\} & 0 & \textcircled{1} & \textcircled{2} & 1 & 0 & 3 & \textcircled{3} \\
\hline
 & & & & & & 1= & 3= \\
 & & & & & & c_{l(1)} & c_{l(2)}
\end{array}$$

$\epsilon = 1$

8. Continue the labelling, case 2 occurs.

$$\begin{array}{cc|ccccc|c}
 & & \{3\} & \{3\} & \{3\} & \{2\} & & \{1\} \\
 & & 0 & 0 & 1 & 0 & -1 & 1 \\
\hline
\{3\} & 2 & 1 & 4 & \textcircled{3} & -1 & \textcircled{1} & -2^* \\
\{1\} & 2 & \textcircled{1} & 2 & 0 & \textcircled{4} & 6 & 4 \\
\{6\} & 1 & \textcircled{1} & \textcircled{2} & 0 & 0 & 2 & \textcircled{3} \\
\hline
 & & & & & & 1= & 3= \\
 & & & & & & c_{l(1)} & c_{l(2)}
\end{array}$$

$\delta = 3$

9. Stage 5 complete since $c_{l(2)} = 0$. Select column 4.

$$\begin{array}{c|ccccc|c}
 & 0 & 0 & 1 & 0 & -1 & 1 \\
\hline
2 & 1 & 4 & 0 & -1 & \textcircled{4} & -2 \\
2 & \textcircled{1} & 2 & 0 & \textcircled{4} & 6 & 4 \\
1 & \textcircled{1} & \textcircled{2} & \textcircled{3} & 0 & 2 & 0 \\
\hline
 & & & & & 4= & 0= \\
 & & & & & c_{l(1)} & c_{l(2)}
\end{array}$$

Stage 4

9. Label, case 3 occurs.

$$\begin{array}{cc|ccccc|c}
 & & \{2\} & \{3\} & \{2\} & & & \{1\} \\
 & & 0 & 0 & 1 & -1 & -1 & 0 \\
\hline
 & 2 & 1 & 4 & 0 & 0 & \textcircled{4} & -1^* \\
\{6\} & 2 & \textcircled{1} & 2 & 0 & 1 & 6 & \textcircled{4} \\
\{1\} & 1 & \textcircled{1} & \textcircled{2} & \textcircled{3} & 1 & 2 & 0 \\
\hline
 & & & & & 0= & & 4= \\
 & & & & & c_{l(1)} & & c_{l(2)}
\end{array}$$

$\epsilon = 1$

10. Stage 4 is complete since column $l_{(2)}$ entries all nonnegative.

$$\begin{array}{c|ccccc|c|}
 & -1 & -1 & 0 & -1 & -1 & -1 \\ \hline
2 & 2 & 5 & 1 & 0 & \textcircled{4} & 0 \\
3 & \textcircled{1} & 2 & 0 & 0 & 5 & \textcircled{4} \\
2 & \textcircled{1} & \textcircled{2} & \textcircled{3} & 0 & 1 & 0 \\ \hline
 & & & & 0= & & 4= \\
 & & & & c_{l(1)} & & c_{l(2)}
\end{array}$$

All entries are nonnegative; hence, the optimal solution is displayed in

$$\begin{array}{c|ccccc|}
 & -1 & -1 & 0 & -1 & -1 \\ \hline
2 & 2 & 5 & 1 & 0 & \textcircled{4} \\
3 & \textcircled{1} & 2 & 0 & \textcircled{4} & 5 \\
2 & \textcircled{1} & \textcircled{2} & \textcircled{3} & 0 & 1 \\ \hline
\end{array}$$

The cost of this solution is 23.

References

1. Balinski, M. L. and Gomory, R. E., "A Mutual Primal-Dual Simplex Method", in *Recent Advances in Mathematical Programming* (Graves and Wolfe, eds.), McGraw-Hill, New York, 1963.

2. Beale, E. M. L., "An Algorithm for Solving the Transportation Problem when the Shipping Cost Over Each Route is Convex", *Naval Research Logistics Quarterly,* Vol. 6 (1959), pp. 43–56.

3. Dantzig, G. B., Ford, L. R., Jr. and Fulkerson, D. R., "A Primal-Dual Algorithm for Linear Programs", in *Linear Inequalities and Related Systems* (Kuhn and Tucker, eds.), Annals of Mathematics Study No. 38, Princeton University Press, Princeton, 1956.

4. Flood, Merrill M., "The Travelling Salesman Problem", *Operations Research,* Vol. 4 (1956), pp. 61–75.

5. Ford, L. R., Jr. and Fulkerson, D. R., "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem", *Canadian Journal of Mathematics,* Vol. 9 (1957), pp. 210–218.

6. Fulkerson, D. R., "An Out-of-Kilter Method for Minimal Cost Flow Problems", *Journal of the Society for Industrial and Applied Mathematics,* Vol. 9 (1961), pp. 18–27.

7. Gleyzal, A., "An Algorithm for Solving the Transportation Problem", *Journal of Research of the National Bureau of Standards,* Vol. 54, No. 4, April 1955, pp. 213–216.

8. Kuhn, H. W., "The Hungarian Method for the Assignment Problem", *Naval Research Loqistics Quarterly,* Vol. 2 (1955), pp. 83–97.

9. ——, "Variants of the Hungarian Method for Assignment Problem", *Naval Research Logistics Quarterly,* Vol. 3 (1956), pp. 253–258.

10. Motzkin, T. S., "The Assignment Problem", *Proc. of Symposia in Applied Mathematics,* Vol. VI—Numerical Analysis, McGraw-Hill, New York, 1956.

11. Munkres, James, "Algorithms for the Assignment and Transportation Problems", *Journal of the Society for Industrial and Applied Mathematics,* Vol. 5 (1957), pp. 32–38.