On Approximation Methods for the Assignment Problem*

JEROME M. KURTZBERG

Burroughs Corporation, Paoli, Pennsylvania

1. *Introduction*

This paper is concerned with approximation methods for handling the classical assignment problem. These methods permit solution of large scale assignment problems where exact methods are not economically feasible because of the extensive computation time requirements. Even though the development and analysis of these approximation methods are strongly directed toward digital computers, the methods are especially desirable for hand computation.

The assignment problem is defined in Section 1, and the results of a solution-time study of an exact method, the Munkres algorithm, are described. In the next section, three approximation methods are formulated. The results of an empirical investigation of error are given in Section 3. In the next section, the approximation methods are investigated analytically; the expected value and error bounds are established for the promising methods. In Section 5, computer mechanizations of the approximation methods are given and evaluated with respect to certain (defined) basic operations. Timing comparisons are then made, with respect to a particular computer, UNIVAC I. A summary of findings is given in the last section.

Definition of Assignment Problem. The statement of the assignment problem is as follows: There are n men and n jobs, with a cost c_{ij} for assigning man i to job j. It is required to assign all men to jobs such that one and only one man is assigned to each job and the total cost of the assignments is minimal.

The problem phrased in a more general way is: Given a matrix (c_{ij}) of real numbers, find a matrix (x_{ij}) such that $\sum_{i,j=1}^{n} c_{ij}x_{ij}$ is minimal, with $x_{ij} = x_{ij}^2$, $\sum_{j=1}^{n} x_{ij} = 1$, and $\sum_{i=1}^{n} x_{ij} = 1$, for all i and j. The matrix (c_{ij}) is termed the cost matrix; the matrix (x_{ij}) is termed the solution matrix or the permutation matrix (also occasionally termed the assignment matrix) [1, 6].

At this point, it may be mentioned that occasionally it is desired to maximize $\sum_{i,j=1}^{n} c_{ij}x_{ij}$, rather than to minimize it. A method yielding an optimal assignment for one always gives an optimal assignment for the other by a simple transformation of the cost matrix (c_{ij}). For example, if all c_{ij} lie in (a, b), the substitution $(d_{ij}) = (b - c_{ij})$ suffices.

Solution Techniques. Any exhaustive scheme of enumeration of all assignments to find the optimal solution is impractical. However, the assignment problem of order n can be expressed as a $2n \times n^2$ zero-sum two-person game,

* Received April, 1961; revised February, 1962 This paper is based upon the author's thesis: Kurtzberg, J. "Approximation methods for handling of large-scale assignment problems," June 1960, Univ. Pennsylvania, Moore School master's thesis.

which can be cast into a linear programming problem [12]; the general technique of the simplex method is then applicable [3, 6]. Many computer programs of the simplex method (and variants of it) exist for various computers [6]. Furthermore, there are a number of algorithms specifically directed at the solution of the assignment problem [1, 8, 9, 10, 11].

Since the assignment problem may be viewed as a special case of the general transportation (or distribution) problem [5, 7], methods suitable to that problem are quite applicable. There are methods designed to allow a "good initial start" for the simplex method operating upon a transportation problem [6]. (The purpose of this good initial start is to reduce the required number of iterations for an optimal solution.)

Methods such as these have been independently formulated by the writer, and evaluated in this paper, as techniques for obtaining near-optimal solutions of the assignment problem. The approximation methods considered and defined in Section 2 are termed (1) the suboptimization method, (2) the row/column-scan method, and (3) the matrix-scan method. Definition of these methods is preceded by the results of a timing study of an exact assignment algorithm.

Solution-Time Study of an Exact Method. A variant of Kuhn's method, developed by James Munkres [13], was adjudged particularly suitable for computer mechanization and was programmed by the writer on UNIVAC I. This was done to provide a necessary subroutine for one of the approximation methods (the suboptimization method, defined in Section 2.1) and to supply a basis for evaluating the worth of the approximation methods. The program that was developed was designed to solve assignment problems up to order 12. With this limit, extreme computer efficiency is possible for UNIVAC I because of the 12-digit word size of this machine. This size was deemed sufficient to carry out any pertinent analysis on a computer-programmed assignment algorithm without falling victim to degenerate cases.

The test cost matrices (for the timing study) were obtained by generating random numbers uniformly in the range (0, 1) and using these numbers as the matrix elements c_{ij}. There is no loss of generality in this choice of range, for if the c_{ij} are distributed uniformly in the interval (a, b), then the linear transformation $c'_{ij} = (c_{ij} - a)/(b - a)$ produces a set of c'_{ij} which lie in the range (0, 1) and are also distributed uniformly. The assignment is independent of the transformation, since a linear transformation does not destroy the relative ordering of the possible solutions.

The results of the timing study are tabulated below. (The time includes reading in the test matrix from tape.) The average sample size was approximately 100 test matrices. One would expect, due to the difficulty in timing the computer runs, more experimental error for small n.

n (size of matrix)	3	4	5	6	7	8	9	10	11	12
t (mean time per run in seconds)	1 3	1 5	2 5	3 3	4.7	6 3	9.4	12 1	15.4	19.2

Unfortunately, there exist qualitative differences (because of the word size and the internal memory size of UNIVAC I) among the ranges (2, 12), (13, 24), and (25, N), where N is arbitrarily large.

For any computer, increasing the matrix size beyond certain critical points introduces radically new problems, primarily of storage, necessitating different programming techniques. However, with this restriction understood, an orthogonal polynomial curve-fitting routine was employed to determine the form of a polynomial that would best fit the data gathered from the range (3, 12). This gave a cubic of the form

$$t = 2.8837 - 1.0511n + 0.1713n^2 + 0.0025n^3, \qquad t \text{ in seconds.}$$

This polynomial furnishes only a weak lower bound for the mean UNIVAC solution time for matrices in the range (13, 24), and an even weaker lower bound for matrices of order greater than 24.

2. *Approximation Methods*

Approximation methods must require relatively few basic operations, thereby yielding fast solution times, and yet not present extremely inefficient assignments. Ideally, these techniques should be directed toward solution of large problems where exact solutions are too costly or unavailable.

2.1. *Suboptimization Method.* This approach assumes the existence of an exact assignment algorithm that can accept an $R \times R$ matrix.

The given $N \times N$ cost matrix is partitioned into S^2 $R \times R$ (sufficiently small) submatrices. Each one of the $R \times R$ submatrices is individually solved for the true optimal assignment value. The assignment value $\sum c_{ij} x_{ij}$ associated with these $R \times R$ submatrices is now considered to constitute the elements of an $S \times S$ matrix. This $S \times S$ matrix is a "gross matrix" in the sense that each element represents a sub-block of assignments.

The true optimal assignment for this new matrix is obtained, thereby designating which (independent) sub-blocks of assignments are to be taken. Since the sub-blocks of assignments are independent in themselves, a feasible assignment for the $N \times N$ matrix is determined.

2.2. *Row/Column-Scan Method.* The row/column-scan algorithm requires two cycles through the matrix; each cycle produces a candidate assignment solution.

On the first cycle, all the rows are sequentially examined and the minimal uncovered element in each row selected. The row is then assigned to the column of the selected element and the column covered, i.e. all of the elements in that column are covered. Each row and column are thus uniquely assigned to one another, thereby constituting an assignment solution. A value is computed for this candidate assignment solution by summing the values of all the selected elements.

Next, on the second cycle, the columns of the original matrix are examined, and, as with the case of the rows, the minimal uncovered element in each column is selected and an assignment made. Again, as with the rows, none of the elements is initially covered. Completion of the column-by-column examination results in a new candidate assignment solution.

An assignment value associated with this new solution is computed and compared with the value associated with the row-by-row assignment solution. The minimum value determines the choice of assignment solution.

It may be desirable, especially for extremely high-order matrices that are tape limited, to mechanize the process by merely considering and operating upon "strings" of elements. A string may be either a row or a column of the matrix. Thus, there would be (for a matrix of order N) N strings for the rows and N strings for the columns. These $2N$ strings could be stored sequentially on tape and operated upon in two cycles, N subcycles in each cycle.

Section 5 presents two detailed computer variants of this algorithm, including the minimum and maximum number of required operations per mechanization. The variants could be implemented in any one of the above ways without altering the given required number of operations. The optimal implementation is a function of the particular computer available.

2.3. *Matrix-Scan Method.* The smallest uncovered element in the entire matrix is selected and the column and row associated with it assigned to one another. (Initially, all of the entries in the matrix are uncovered.) After the assignment is made, the row and column are covered; i.e. the elements in the assigned row and column are covered. This has the effect of reducing the order of the matrix by one.

The process is repeated until all of the rows have been assigned to columns. To produce a complete set of row-to-column assignments (an assignment solution) for a matrix of order N, N passes over the entire matrix are required. Section 5 presents two detailed computer variants of this algorithm, including the minimum and maximum number of required operations per mechanization.

3. *Empirical Error Investigations of Approximation Methods*

The value of each approximation method is dependent upon two factors: the time necessary for solution, given implicitly by the required number of basic operations, and the amount of error generated by the algorithm.

To gain insight into the relative worth of the approximation methods, experiments were conducted on low order matrices (of order 12) for which the true minimal and maximal assignments could be found.

The results of these empirical investigations indicated that the suboptimization method does not compete favorably with the other two approximation methods, even with respect to solution error. Since the solution-time cost is (by far) heaviest for the suboptimization method, and the results so discouraging, there does not appear to be much profit in it.

A total of 50 test matrices of order 12 was subdivided every possible way for the suboptimization method. The particular partition that gave the least error was a subdivision of four 6×6 submatrices. The average relative error with respect to a minimal solution for that partition was 47.1 percent. The greatest average relative error, 76.5 percent, again with respect to a minimal solution, was generated with a partition of sixteen 3×3 submatrices. With respect to a maximal solution, the corresponding average relative errors for the same partitions are 6.8 and 11.1 percent, respectively.

The row/column-scan method and the matrix-scan method were applied to several test matrices. The average relative errors with respect to minimal solutions were approximately 29 and 14 percent, respectively. The average relative errors with respect to maximal solutions were approximately 4.2 and 2.6 percent, respectively.

If the worst assignment for a minimal solution (corresponding to the maximal assignment) were selected for these test matrices, the relative error would be 603 percent. Correspondingly, the worst assignment for a maximal solution (equivalent to the minimal assignment) yields a relative error of 87.8 percent.

4. *Analytical Investigations of Approximation Methods*

In this section, an expression for the expected value of an assignment solution, in terms of the order of the matrix, is derived for two approximation methods: the row/column-scan and matrix-scan algorithms. (For appropriate definitions, see [2] or [4].) Of particular interest and importance is the error involved in solving high order problems.

Bounds are established for the absolute error of the expected solution values given by the row/column-scan and matrix-scan methods. In addition, bounds are found for the expected relative error of the maximal solution given by the above methods. The expected relative error is shown to vanish as the order of the matrix increases without bound. In the analysis, the elements in the matrices are independent and rectangularly distributed in [0, 1]. A generalization of results is then given for elements rectangularly distributed in $[a, b]$, an arbitrary finite interval.

4.1. *Row/Column-Scan Method.* For the remainder of this section, it is convenient to consider the maximal assignment instead of the minimal assignment. Solving the maximal assignment problem for the matrix $(d_{ij}) = (1 - c_{ij})$ is equivalent to solving the minimal assignment problem for the matrix (c_{ij}). The (c_{ij}) and (d_{ij}) matrices have the same assignment solution, and summation of the assigned elements in the (c_{ij}) matrix yields the correct minimal assignment value.

For a matrix of order n, there are n rows which must be assigned. Each row has a random variable associated with it: namely x_k, the value of the selected element. The value of the total assignment is simply $\sum_{k=1}^{n} x_k$.

The expected value for the total maximal assignment given by a row-scan method, in which only the rows and not the columns are examined, is now derived.

Let x_k be the maximum (uncovered) element of the $(n - k + 1)$th row. Then, the random variable x_k is the maximum of k random variables; namely, the contents of the surviving locations of the $(n - k + 1)$th row. Denote these contents by $1_1, 1_2, \cdots, 1_k$. Thus,

$$x_k = \max (1_1, 1_2, 1_3, \cdots, 1_k).$$

Let $F_k(x)$ denote the cumulative distribution function of x_k. Using the symbol

Pr for mathematical probability, $F_k(x)$ is computed as follows:

$$\begin{aligned} F_k(x) &= \Pr(x_k \leqq x) = \Pr[\max(1_1, 1_2, \cdots, 1_k) \leqq x] \\ &= \Pr(1_1 \leqq x, \quad 1_2 \leqq x, \quad \cdots, \quad 1_k \leqq x) \\ &= [\Pr(1_1 \leqq x)][\Pr(1_2 \leqq x)] \cdots [\Pr(1_k \leqq x)] \\ &= [F_{1_1}(x)][F_{1_2}(x)] \cdots [F_{1_k}(x)]. \end{aligned}$$

Now

$$F_{1_j}(x) = \begin{cases} 1, & \text{for} \quad x \geqq 1 \\ x, & \text{for} \quad 0 < x < 1 \\ 0, & \text{for} \quad x \leqq 0, \end{cases}$$

hence

$$F_k(x) = \begin{cases} 1, & \text{for} \quad x \geqq 1 \\ x^k, & \text{for} \quad 0 < x < 1 \\ 0, & \text{for} \quad x \leqq 0. \end{cases}$$

The associated density function, $f_k(x)$, is simply the derivative of $F_k(x)$,

$$f_k(x) = \begin{cases} 0, & \text{for} \quad x \geqq 1 \\ kx^{k-1}, & \text{for} \quad 0 < x < 1 \\ 0, & \text{for} \quad x \leqq 0. \end{cases}$$

Thus, the expected value of the random variable x_k is

$$\langle x_k \rangle = \int_{-\infty}^{\infty} x f_k(x)\, dx = \int_0^1 x(kx^{k-1})\, dx = \frac{k}{k+1}.$$

The expected value of the total assignment, denoted by $M_1(n)$, is given by $\sum_{k=1}^{n} \langle x_k \rangle$. Since the mean of a sum of random variables equals the sum of their means,

$$M_1(n) = \left\langle \sum_{k=1}^{n} x_k \right\rangle = \sum_{k=1}^{n} \frac{k}{k+1}.$$

An easily evaluated lower bound for $M_1(n)$ is

$$M_1(n) = \sum_{k=1}^{n} \frac{k}{k+1} = n - \sum_{k=1}^{n} \frac{1}{k+1} > n - \int_1^{n+1} \frac{dx}{x} = n - \ln(n+1).$$

The inequality holds for

$$\int_1^{n+1} \frac{dx}{x} = \sum_{k=1}^{n} \int_k^{k+1} \frac{dx}{x} > \sum_{k=1}^{n} \int_k^{k+1} \frac{dx}{k+1} = \sum_{k=1}^{n} \frac{1}{k+1}.$$

The expected value of the minimal solution, $m_1(n)$, for the matrix (c_{ij}) is now obtained by means of the results for $M_1(n)$. Let y_k denote the minimum (uncovered) element of the $(n - k + 1)$th row of (c_{ij}). Since $d_{ij} = 1 - c_{ij}$, it follows that $y_k = 1 - x_k$. Therefore,

$$m_1(n) = \left\langle \sum_{k=1}^{n} y_k \right\rangle = \left\langle \sum_{k=1}^{n} (1 - x_k) \right\rangle = n - \sum_{k=1}^{n} \frac{k}{k+1} = \sum_{k=1}^{n} \frac{1}{k+1}.$$

An upper bound for $m_1(n)$ is $\ln (n + 1)$, for, as seen before,

$$\int_1^{n+1} \frac{dx}{x} > \sum_{k=1}^{n} \frac{1}{k+1}.$$

The expression M_1 gives the expected value for only the maximum row-scan solution. In the row/column-scan method there are two tentative assignment solutions, one for the rows and one for the columns. The better of the two is selected. The expression for the expected value of the maximum column-scan solution is also given by M_1. Since the better (in this case, larger) of the two solutions is selected, the expected value for the complete method should be greater than M_1. However, that expected value is not immediately apparent, as the distributions for the row-scan and the column-scan assignments are not independent.

Similarly, m_1 provides an upper bound for the expected minimal solution given by the row/column-scan method. Thus, the expected minimum row/column-scan solution is less than $\ln (n + 1)$, and the expected maximum row/column-scan solution is greater than $n - \ln (n + 1)$.

A bound is now found for the expected value of the relative error given by the maximal row-scan method. This value tends to zero as the order of the matrix increases. Let $\sum_{k=1}^{n} t_k$ denote the true solution value. Then, the expected relative error is

$$\left\langle \left| \frac{\sum_{k=1}^{n} t_k - \sum_{k=1}^{n} x_k}{\sum_{k=1}^{n} t_k} \right| \right\rangle = \left\langle 1 - \frac{\sum_{k=1}^{n} x_k}{\sum_{k=1}^{n} t_k} \right\rangle$$

$$\leqq \left\langle 1 - \frac{\sum_{k=1}^{n} x_k}{n} \right\rangle = 1 - \frac{M_1(n)}{n} < \frac{\ln (n+1)}{n},$$

which goes to zero as n approaches infinity. This follows from the relations

$$n \geqq \sum_{k=1}^{n} t_k \geqq \sum_{k=1}^{n} x_k,$$

(which hold because of all entries lying between 0 and 1, and by the definition of true maximal solution), and from the relations

$$M_1(n) = \left\langle \sum_{k=1}^{n} x_k \right\rangle = \sum_{k=1}^{n} \frac{k}{k+1} > n - \ln (n + 1).$$

Furthermore, the same bound of $[\ln(n + 1)]/n$ holds for the relative error of the expected value of the row-scan method. This is so because both

$$\frac{\left\langle \sum_{k=1}^{n} x_k \right\rangle}{\left\langle \sum_{k=1}^{n} t_k \right\rangle} \text{ and } \left\langle \frac{\sum_{k=1}^{n} x_k}{\sum_{k=1}^{n} t_k} \right\rangle \geqq \frac{M_1(n)}{n}.$$

An upper bound for the absolute error for both the expected minimal and maximal solution values is now found. Let $e(n)$ and $E(n)$ denote the expected value of the true minimal and maximal solutions, respectively. Since $E(n) = n - e(n)$ and $M_1(n) = n - m_1(n)$, the absolute error bound is

$$| e(n) - m_1(n) | = | E(n) - M_1(n) | \leqq | n - [n - \ln(n + 1)] | = \ln(n + 1).$$

Again, the fact is stressed that this upper bound for absolute error refers to only a row-scan *or* a column-scan solution. Thus, there may be a significantly sharper bound for the absolute error of the expected solution given by a complete row/column-scan method.

4.2. *Matrix-Scan Method.* As with the case of the row/column-scan method, it is convenient to consider the maximal assignment solution instead of the minimal assignment method.

In the row/scan method, the column associated with the largest uncovered element of a row is covered; the members of other rows play no part in the choice of the largest element. In the matrix-scan method, however, the row *and* column of the largest uncovered element of the matrix are covered. Hence, the remaining elements, which are the candidates for a new choice, cannot be larger than the selected one, and thus are no longer rectangularly distributed in [0, 1].

Let the expected value given by the matrix-scan method be denoted by M_2. Suppose the maximum value in the matrix is x, where $0 \leqq x \leqq 1$. The remaining elements, after that maximum element is selected, are then independently and rectangularly distributed in $[0, x]$. Therefore, the expected value of the matrix-scan method on an $(n - 1) \times (n - 1)$ minor (i.e. the resulting uncovered elements) is $xM_2(n - 1)$. This relation follows by considering that if every element of the $(n - 1) \times (n - 1)$ minor is divided by x, an $(n - 1) \times (n - 1)$ matrix of elements in [0, 1] is obtained. Given that the maximum value is x, then $[x + xM_2(n - 1)]$ is the expected value of the matrix-scan; the probability that x is the maximum is $n^2x^{n^2-1}\,dx$. So, taking the product of the expected value (assuming x is the maximum) and the associated probability, and summing x over all possible values (i.e. integrating from 0 to 1), leads to the expression for the expected value of the matrix-scan method:

$$M_2(n) = \int_0^1 [x + xM_2(n - 1)]n^2x^{n^2-1}\,dx = \frac{n^2}{n^2 + 1}[1 + M_2(n - 1)].$$

A lower bound for $M_2(n)$ is now found, using the recursion relation

$$M_2(n) = \frac{n^2}{n^2 + 1}[M_2(n - 1) + 1] = \frac{1}{1 + \frac{1}{n^2}}[M_2(n - 1) + 1].$$

Since $\frac{1}{1+\frac{1}{n^2}} = 1 - \frac{1}{n^2} + \left(\frac{1}{n^2}\right)^2 - \cdots$, then

$$M_2(n) > \left[1 - \frac{1}{n^2}\right][M_2(n-1) + 1]$$

or

$$M_2(n) - M_2(n-1) > 1 - \frac{M_2(n-1)}{n^2} - \frac{1}{n^2};$$

and since $M_2(n-1) \leqq n - 1$,

$$M_2(n) - M_2(n-1) > 1 - \frac{n-1}{n^2} - \frac{1}{n^2} = 1 - \frac{1}{n}.$$

Taking the sum of both sides, with n going from 1 to T, gives the result

$$\sum_{n=1}^{T} [M_2(n) - M_2(n-1)] > \sum_{n=1}^{T} \left(1 - \frac{1}{n}\right),$$

which is equal to

$$M_2(T) - M_2(0) > T - \sum_{n=1}^{T} \frac{1}{n}.$$

Now, $M_2(0) = 0$, and since[1] $\gamma > \sum_{n=1}^{T} 1/n - \ln T$, changing the dummy variable (substituting n for T) results in the following lower bound for $M_2(n)$

$$M_2(n) > n - \gamma - \ln n.$$

An upper bound for $M_2(n)$ is now found by taking one more term in the expansion of $1/[1 + (1/n^2)]$ and substituting in the recursion expression for $M_2(n)$, as follows:

$$M_2(n) < \left[1 - \frac{1}{n^2} + \frac{1}{n^4}\right][M_2(n-1) + 1]$$

$$= 1 + M_2(n-1) - \left[\frac{1}{n^2} - \frac{1}{n^4}\right][M_2(n-1) + 1].$$

Thus,

$$M_2(n) - M_2(n-1) < 1 - \frac{1}{n^2} + \frac{1}{n^4} - \frac{M_2(n-1)}{n^2} + \frac{M_2(n-1)}{n^4}$$

$$< 1 - \frac{1}{n^2} + \frac{1}{n^4} - \frac{(n-1) - \gamma - \ln(n-1)}{n^2} + \frac{n-1}{n^4}$$

$$< 1 - \frac{1}{n} + \frac{\gamma}{n^2} + \frac{\ln n}{n^2} + \frac{1}{n^3}.$$

[1] Euler's constant, γ, is defined as $\gamma = \lim_{n\to\infty} \left[\sum_{k=1}^{n} \frac{1}{k} - \ln n\right]$, where $\sum_{k=1}^{n} \frac{1}{k} - \ln n$ approaches γ from below.

Taking the sum of both sides with n going from 1 to T gives the result

$$\sum_{n=1}^{T} [M_2(n) - M_2(n-1)] < \sum_{n=1}^{T} \left[1 - \frac{1}{n} + \frac{\gamma}{n^2} + \frac{\ln n}{n^2} + \frac{1}{n^3}\right]$$

or

$$M_2(T) < T - \sum_{n=1}^{T} \frac{1}{n} + \gamma \sum_{n=1}^{T} \frac{1}{n^2} + \sum_{n=1}^{T} \frac{\ln n}{n^2} + \sum_{n=1}^{T} \frac{1}{n^3}.$$

Thus, it follows that

$$n - \sum_{k=1}^{n} \frac{1}{k} < M_2(n) < n - \sum_{k=1}^{n} \frac{1}{k} + \gamma \sum_{k=1}^{n} \frac{1}{k^2} + \sum_{k=1}^{n} \frac{\ln k}{k^2} + \sum_{k=1}^{n} \frac{1}{k^3}.$$

Furthermore, since

$$M_1(n) = n - \sum_{k=1}^{n} \frac{1}{k+1} = n - \sum_{k=1}^{n} \frac{1}{k} + 1 - \frac{1}{n+1} > n - \sum_{k=1}^{n} \frac{1}{k},$$

the bound becomes

$$M_2(n) < M_1(n) + \gamma \sum_{k=1}^{n} \frac{1}{k^2} + \sum_{k=1}^{n} \frac{\ln k}{k^2} + \sum_{k=1}^{n} \frac{1}{k^3}.$$

Thus, $M_2(n) - M_1(n)$ increases with n, but since $\gamma \sum_{k=1}^{\infty} (1/k^2)$, $\sum_{k=1}^{\infty} (1/k^3)$, and $\sum_{k=1}^{\infty} (\ln k/k^2)$ are convergent (i.e. bounded by a constant),

$$M_2(n) - M_1(n) < c,$$

where c is the constant given by the sum of the limits of $\sum(1/n^3)$, $\gamma\sum(1/n^2)$, and $\sum \ln k/k^2$. This sum may be crudely estimated by noting that

$$\sum_{n=1}^{\infty} \frac{1}{n^3} < \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

and

$$\int_1^n \frac{\ln x}{x^2}\, dx = 1 - \frac{1 + \ln n}{n} > \sum_{k=1}^{n} \frac{\ln k}{k^2}.$$

Thus,

$$c < \gamma \frac{\pi^2}{6} + \frac{\pi^2}{6} + 1 < 3.6.$$

Therefore, the expected values of the matrix-scan and the row-scan methods are of the same order of magnitude.

Furthermore, since $M_2(n) > n - \sum_{k=1}^{n} 1/k > n - (\ln n + \gamma)$, it follows, by the same reasoning used for the row-scan method, that the bound on the expected relative error (and the relative error of the expected value) in the matrix-scan method is $(\ln n + \gamma)/n$, which goes to 0 as n approaches infinity.

Similarly, the bound on the absolute error of $M_2(n)$ and $m_2(n)$ is $\ln n + \gamma$.

4.3. *Generalization from* [0, 1] *to* [a, b]. The previous analysis assumes all matrix elements to be rectangularly distributed in [0, 1]. The results are readily extended to matrices with elements rectangularly distributed in $[a, b]$, an arbitrary finite interval. By employing the linear transformation

$$c_{ij} = \frac{e_{ij} - a}{b - a}, \quad \text{where} \quad e_{ij} \quad \text{is in} \quad [a, b],$$

and going through the appropriate manipulations, the following error bounds are obtained.

The expected relative error of the maximal solution with the row-scan method is bounded by $[(b - a) \ln (n + 1)]/nb$, and the expected absolute error is bounded by $(b - a) \ln (n + 1)$. Similarly, the expected relative error for the matrix-scan method is less than $[(b - a) (\ln n + \gamma)]/nb$, and the expected absolute error is less than $(b - a) (\ln n + \gamma)$.

5. *Solution-Time Evaluation of Approximate Methods*

For the two approximation methods that appear promising, there are two variant mechanizations. Flow charts are presented for the row-scan method and the matrix-scan method showing these candidate mechanizations.

Certain "basic operations" are defined, and the rival mechanizations are then analyzed for each method in terms of these operations. Timing estimates are made for the methods with respect to a given computer, UNIVAC I.

5.1. *Row/Column-Scan Mechanizations.* The procedures or steps which define the row/column-scan method for a minimal solution are:

(1) Select, in each row, the minimum element whose column has *not* been assigned to any other (previous) row.
(2) Assign the row to the column of the selected element.
(3) Indicate that the column has been assigned. (Cover the column.)
(4) After all rows have been processed, repeat the algorithm, treating the transpose of the matrix in similar fashion
(5) Of the two tentative assignments—the first from the original matrix, the second from the transpose—choose the one that is smaller.

The two variants differ in treatment of step (1). As the names of the mechanizations suggest, one first tests for a prospective new "smallest" row element and then determines its availability, i.e. whether or not its column has been previously assigned; in the other mechanization, the order of tests is reversed. The first is more favorable initially; the second shows to advantage in the terminal phase. However, in general it would be unwise to include controls to switch from one to another at the "breakpoint." The controls would cost more in extra operations than would be saved.

The flow charts for the element-indicator and indicator-element mechanizations are shown in Figure 1.

5.2. *Matrix-Scan Mechanization.* The procedure or steps which define the matrix-scan method for a minimal solution are:

(1) Select, in the matrix, the minimum element that does *not* have its row or column covered (i.e , whose row or column has not been previously assigned).
(2) Assign the row of the selected element to its column.

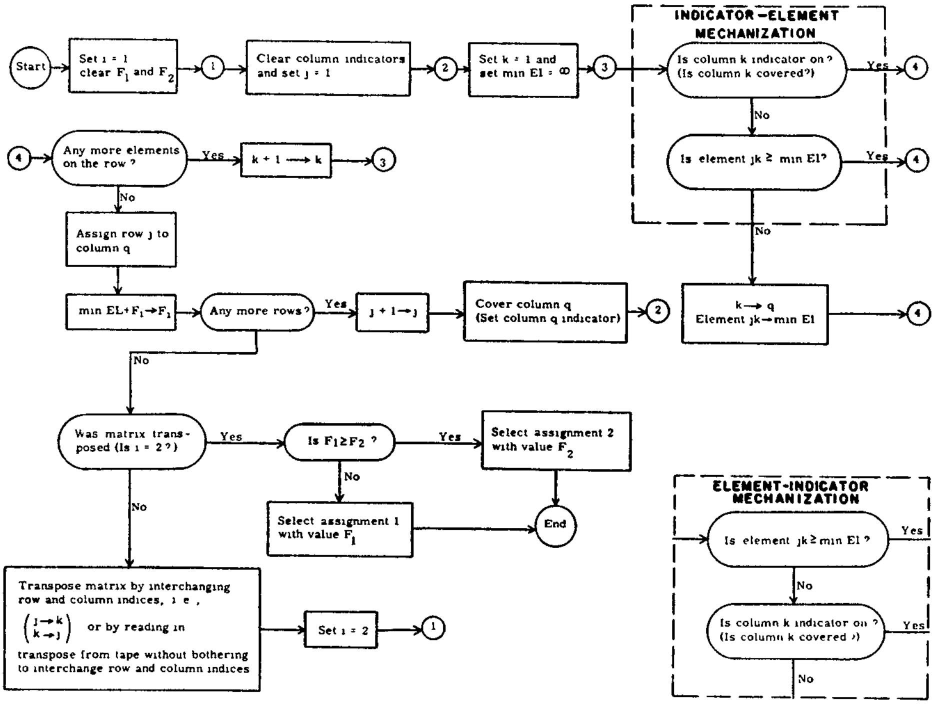

FIG 1. Row/column-scan method

(3) Indicate that the column and row are assigned. (Cover the row and column.)
(4) Test for completion of n assignments (for a matrix of order n); if not completed, continue recursively.

As with the row/column-scan method, the two variants differ in step (1); the discussion given for that method is also applicable here.

The flow charts for the element-indicator and indicator-element mechanizations are shown in Figure 2.

5.3. *Basic Computer Operations for the Approximation Methods.* To evaluate the computation speed of the approximation methods (and their candidate computer mechanizations), it is expedient to define certain operations, especially significant for the approximation methods. The techniques may then be analyzed in terms of these operations, thereby providing a basis for comparisons. The operations are as follows:

(1) Element comparison
(2) Indicator (sentinel) test
(3) Row-to-column assignment
(4) Candidate element replacement
(5) End-of-element-in-row/column test
(6) End-of-row/column test
(7) End-of-assignments test
(8) Matrix transposition (or interchange of indices)
(9) Solution-value comparison

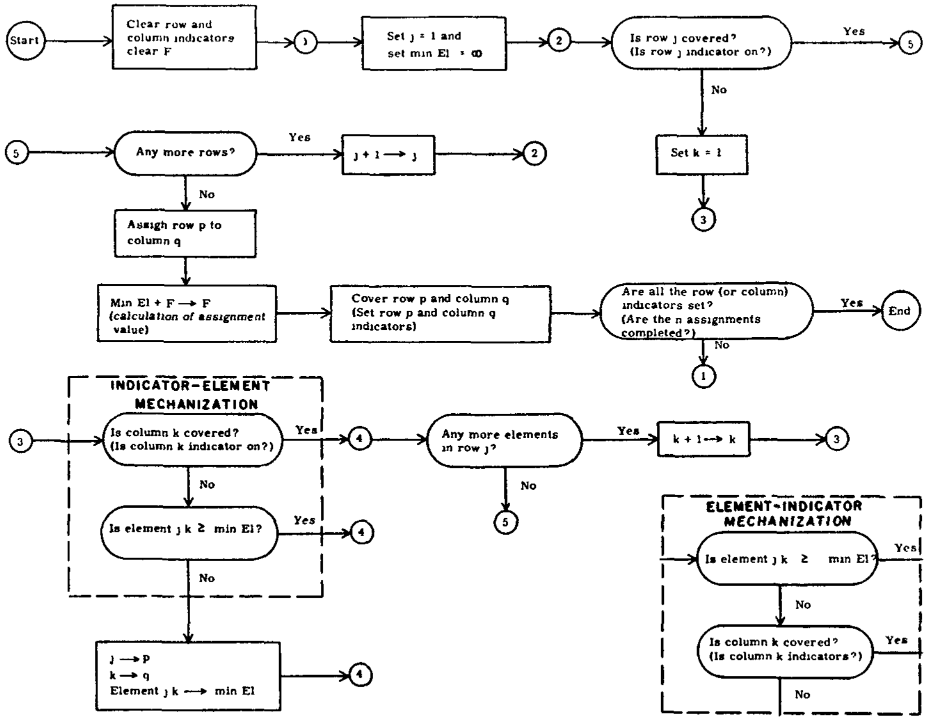

FIG. 2. Matrix-scan method

It is considered that some necessary operations not listed, such as advancing indices, clearing counters, resetting indices, presetting values, etc., are absorbed in the above operations. For example, summation of the selected elements to form the solution value is considered to be implicitly contained in the row-to-column assignment operation. Since the main purpose of the operations is to furnish a basis for comparison, no harm results from this artifice.

5.4. *Row/Column-Scan Operation Count.* For the minimum operation case (the case requiring the lowest total of equi-weighted operations), consider that the elements of the matrix are monotonically increasing across each row and the last element of the jth row is equal to or greater than the last element of the $(j + 1)$th row. Another minimum case is given by a matrix which has its main diagonal monotonically increasing, with the remaining set of elements monotonically increasing with respect to themselves.

For the maximum operation case, consider that the elements of the matrix are monotonically strictly decreasing across each row, and that the last element of the jth row is greater than the first element of the $(j + 1)$th row.

The estimate of the "average" number of operations is given by the average of the minimum operation and maximum operation cases, and is taken as a figure-of-merit for the techniques.

A summary of the operation counts for the variant mechanizations is given later (see Table 5).

TABLE 1. OPERATION COUNT ASSOCIATED WITH ROW-SCAN METHOD, ELEMENT-INDICATOR MECHANIZATION

Assignment Number	Element Comparison	Column-Indicator		Row-Col. Assignment	Element Replacement		End of Elements in Row Test	End of Rows Test
		Min. Case	Max. Case		Min. Case	Max. Case		
1	n	1	n	1	1	n	n	1
2	n	2	n	1	1	(n-1)	n	1
3	n	3	n	1	1	(n-2)	n	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	n	n	n	1	1	1	n	1
TOTAL	n^2	$\sum_{k=1}^{n} k$	n^2	n	n	$\sum_{k=1}^{n} k$	n^2	n

For the element-indicator mechanization, figures for the minimum and maximum number of operations for one sweep (row-scan) through the matrix, showing operation counts of the individual assignments, are listed in Table 1.

The operation count for one sweep (row-scan) through the matrix is, for the minimum count case,

$$n^2 + \sum_{k=1}^{n} k + n + n + n^2 + n = \frac{5n^2 + 7n}{2}$$

and for the maximum count case,

$$n^2 + n^2 + n + \sum_{k=1}^{n} k + n^2 + n = \frac{7n^2 + 5n}{2}.$$

For a complete operation count for a row/column-scan, the following additional operations must be taken into account: end-of-assignment test; matrix transposition (interchange of indices); and solution-value comparison.

The test for end-of-assignment in the row/column-scan method consists in determining if both the row *and* column scans have been completed. This operation must be done twice, once per sweep. The other two additional operations must be done once.

Thus, the complete operation count is, for the minimum count case,

$$2\left(\frac{5n^2 + 7n}{2} + 1\right) + 2 = 5n^2 + 7n + 4$$

and for the maximum count case,

$$2\left(\frac{7n^2 + 5n}{2} + 1\right) + 2 = 7n^2 + 5n + 4.$$

Therefore, the average of the minimum and maximum cases, or the figure-of-merit for the element-indicator mechanization is

$$\frac{(5n^2 + 7n + 4) + (7n^2 + 5n + 4)}{2} = 6n^2 + 6n + 4.$$

TABLE 2 OPERATION COUNT ASSOCIATED WITH ROW-SCAN METHOD, INDICATOR-ELEMENT MECHANIZATION

Assignment Number	Element Comparison	Column-Indicator		Row-Col. Assignment	Element Replacement		End of Elements in Row Test	End of Rows Test
		Min Case	Max Case		Min. Case	Max. Case		
1	n	n	n	1	1	n	n	1
2	(n-1)	n	n	1	1	(n-1)	n	1
3	(n-2)	n	n	1	1	(n-2)	n	1
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
n	1	n	n	1	1	1	n	1
TOTAL	$\sum_{k=1}^{n} k$	n^2	n^2	n	n	$\sum_{k=1}^{n} k$	n^2	n

For the indicator-element mechanization, the operation count (shown in Table 2) for one sweep (row-scan) through the matrix is, in the minimum count case,

$$\sum_{k=1}^{n} k + n^2 + n + n + n^2 + n = \frac{5n^2 + 7n}{2}$$

and, in the maximum count case,

$$\sum_{k=1}^{n} k + n^2 + n + \sum_{k=1}^{n} k + n^2 + n = 3n^2 + 3n.$$

For a complete operation count for a row/column-scan, one must, as in the element-indicator technique, take into account additional operations. The same remarks made about these additional operations also hold here.

Thus, the complete operation count is, in the minimum count case,

$$2\left(\frac{5n^2 + 7n}{2} + 1\right) + 2 = 5n^2 + 7n + 4,$$

and in the maximum count case,

$$2(3n^2 + 3n + 1) + 2 = 6n^2 + 6n + 4.$$

Therefore, the figure-of-merit for the indicator-element mechanization is

$$\frac{(5n^2 + 7n + 4) + (6n^2 + 6n + 4)}{2} = \frac{11n^2 + 13n + 8}{2}.$$

5.5. *Matrix-Scan Operation Count.* The minimum and maximum cases for the matrix scan method are the same as for the row/column-scan method.

For the element-indicator mechanization, the complete operation count (shown in Table 3) is, in the minimum count case,

$$n^2 + \sum_{k=1}^{n} kn + \sum_{k=1}^{n} k + n + n + \sum_{k=1}^{n} kn + n^2 + n = \frac{2n^3 + 7n^2 + 7n}{2}$$

and in the maximum count case,

$$n^2 + \sum_{k=1}^{n} kn + \sum_{k=1}^{n} kn + n + \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} kn + n^2 + n = \frac{11n^3 + 24n^2 + 13n}{6};$$

and the figure-of-merit is

$$\frac{1}{2}\left[\frac{2n^3 + 7n^2 + 7n}{2} + \frac{11n^3 + 24n^2 + 13n}{6}\right] = \frac{17n^3 + 45n^2 + 34n}{12}.$$

For the indicator-element mechanization, the complete operation count (shown in Table 4) is, for the minimum count case,

$$n^2 + \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} kn + n + n + \sum_{k=1}^{n} kn + n^2 + n = \frac{8n^3 + 21n^2 + 19n}{6}$$

and for the maximum count case,

$$n^2 + \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} kn + n + \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} kn + n^2 + n = \frac{5n^3 + 12n^2 + 7n}{3};$$

and the figure-of-merit is

$$\frac{1}{2}\left[\frac{8n^3 + 21n^2 + 19n}{6} + \frac{5n^3 + 12n^2 + 7n}{3}\right] = \frac{6n^3 + 15n^2 + 11n}{4}.$$

TABLE 3 Operation Count Associated with Matrix-Scan Method, Element-Indicator Mechanization

Assignment Number	Row Indicator	Element Comparison	Column-Indicator		Row-Col Assignment	Element Replacement		End of Elements in Row Test	End of Rows Test	End of Assignment
			Min Case	Max Case		Min Case	Max Case			
1	n	n^2	1	n^2	1	1	n^2	n^2	n	1
2	n	n^2-n	2	n^2-n	1	1	$(n-1)^2$	n^2-n	n	1
3	n	n^2-2n	3	n^2-2n	1	1	$(n-2)^2$	n^2-2n	n	1
n	n	n	n	n	1	1	1	. n	n	1
TOTAL	n^2	$\sum_{k=1}^{n} kn$	$\sum_{k=1}^{n} k$	$\sum_{k=1}^{n} kn$	n	n	$\sum_{k=1}^{n} k^2$	$\sum_{k=1}^{n} kn$	n^2	n

TABLE 4. Operation Count Associated with Matrix-Scan Method, Indicator-Element Mechanization

Assignment Number	Row Indicator	Element Comparison	Column-Indicator		Row-Col Assignment	Element Replacement		End of Elements in Row Test	End of Rows Test	End of Assignment
			Min Case	Max Case		Min Case	Max Case			
1	n	n^2	n^2	n^2	1	1	n^2	n^2	n	1
2	n	$(n-1)^2$	n^2-n	n^2-n	1	1	$(n-1)^2$	n^2-n	n	1
3	n	$(n-2)^2$	n^2-2n	n^2-2n	1	1	$(n-2)^2$	n^2-2n	n	1
							.			
n	n	1	n	n	1	1	1	n	n	1
TOTAL	n^2	$\sum_{k=1}^{n} k^2$	$\sum_{k=1}^{n} kn$	$\sum_{k=1}^{n} kn$	n	n	$\sum_{k=1}^{n} k^2$	$\sum_{k=1}^{n} kn$	n^2	n

5.6. *Time Comparisons of the Approximation Methods (with respect to UNIVAC I).* The computer computation times for the approximation methods are now presented as a function of the order of the assignment matrix. For the time comparisons, the approximation methods were programmed in the UNIVAC I code and time estimates were then made based upon the coding.

There are a number of jump discontinuities in the computation time of the approximation methods as the order of the problem increases. The input to the UNIVAC I computer must be in blocks of 60 words; thus, the input-time penalty increases according to the smallest integer equal to or greater than $n^2/60$. There is a critical point where the size of a problem is such that the entire matrix cannot

TABLE 5. SUMMARY OF OPERATION COUNT

Method	Mechanization	Minimum Operation Case	Maximum Operation Case	Figure-of-Merit
Row-Scan	Element-Indicator	$(5n^2 + 7n)/2$	$(7n^2 + 5n)/2$	$3n^2 + 3n$
* Row-Scan	Indicator-Element	$(5n^2 + 7n)/2$	$3n^2 + 3n$	$(11n^2 + 13n)/4$
Row/Column-Scan	Element-Indicator	$5n^2 + 7n + 4$	$7n^2 + 5n + 4$	$6n^2 + 6n + 4$
* Row/Column-Scan	Indicator-Element	$5n^2 + 7n + 4$	$6n^2 + 6n + 4$	$(11n^2 + 13n + 8)/2$
* Matrix-Scan	Element-Indicator	$(2n^3 + 7n^2 + 7n)/2$	$(11n^3 + 24n^2 + 13n)/6$	$(17n^3 + 45n^2 + 34n)/12$
Matrix-Scan	Indicator-Element	$(8n^3 + 21n^2 + 19n)/6$	$(5n^3 + 12n^2 + 7n)/3$	$(6n^3 + 15n^2 + 11n)/4$

* Recommended Technique

TABLE 6. UNIVAC I TIMING FOR OPERATIONS, ROW/COLUMN-SCAN METHOD, INDICATOR-ELEMENT MECHANIZATION

Operation	Frequency (Average)	Time (milliseconds) per Unit Operation	Total Time (milliseconds)
Element Comparison	$n^2 + n$	2	$2n^2 + 2n$
Column Indicator	$2n^2$	2	$4n^2$
Row-Column Assignment	$2n$	31	$62n$
Element Replacement	$(n^2 + 3n)/2$	2	$n^2 + 3n$
End of Elements in Row	$2n^2$	2	$4n^2$
End of Rows	$2n$	5	$10n$
End of Assignment (Was Matrix Transposed?)	2	2	4
Transpose Matrix (Interchange Indices)	1	6	6
Solution-Value Comparison	1	3	3
Read and Write (3 blocks)	1	510	510
Initialize	1	8	8

Total: $11n^2 + 77n + 531$

be contained simultaneously in the memory. If this critical size is exceeded, the time requirements of the matrix-scan method increase to another order of magnitude.

To avoid these problems, and to provide a simpler direct time comparison with the UNIVAC I program of an exact method (the Munkres algorithm) it is assumed that three UNIVAC blocks, or 180 words, are sufficient for input to the programs. Of course, for larger matrices, the input-time penalty could be adjusted accordingly.

For a matrix of order n, and with partitioning into $p \times p$ submatrices, the suboptimization method requires the exact solution of q^2 $p \times p$ matrices, and one $q \times q$ matrix. A weak lower bound on time required for each size matrix may be found by reference to the time equation of the Munkres algorithm in Section 1. For matrices in the range (2, 12), the time chart in Section 1 furnishes the (average) time actually consumed. An additional amount of time is consumed in the partitioning and forming of the $q \times q$ matrix. However, this time is negligible in comparison with the time consumed in solving the submatrices.

Table 6 lists the basic operations (Section 5.3) with the associated frequency for the optimal mechanization of the row/column-scan method (indicator-element mechanization) and the time (in milliseconds) for UNIVAC I coding for each operation. The time for "initialization" (setting up of certain controls) is also given, along with the time for tape reading and tape writing (three blocks). As was previously mentioned, certain operations are absorbed in the basic opera-

TABLE 7. UNIVAC I TIMING FOR OPERATIONS, MATRIX-SCAN METHOD, ELEMENT-INDICATOR MECHANIZATION

Operation	Frequency (Average)	Time (milliseconds) per Unit Operation	Total Time (milliseconds)
Row Indicator	n^2	2	$2n^2$
Element Comparison	$(n^3 + n^2)/2$	2	$n^3 + n^2$
Column Indicator	$(n^3 + 2n^2 + n)/4$	2	$n^3/2 + n^2 + n/2$
Row Column Assignment	n	35	35n
Element Replacement	$(2n^3 + 3n^2 + 7n)/12$	2	$n^3/3 + n^2/2 + 7n/6$
End of Elements in Row	$(n^3 + n^2)/2$	2	$n^3 + n^2$
End of Rows	n^2	6	$6n^2$
End of Assignment	n	2	2n
Read and Write (3 blocks)	1	510	510
Initialize	1	7	7

$$\text{Total: } \frac{17n^3}{6} + \frac{23n^2}{2} + 232\,n + 517$$

tions; summation of solution value, for example, is included in the row-column assignment operation.

The expression for the time (average case), converted to give time in seconds, is for the row/column-scan method,

$$t_1 = 0.011n^2 + 0.077n + 0.531.$$

For a 12×12 matrix, $t_1 \approx 3$ seconds. For the row-scan method, which requires approximately one-half the running time of the row/column-scan method, $t_2 \approx 1.8$ seconds. For the same size matrix, an exact method (the Munkres assignment program) took, on the average, 19.2 seconds. The corresponding time equation for the row-scan method is

$$t_2 = 0.0055n^2 + 0.0385n + 0.518.$$

Table 7 lists the basic operations and pertinent data for the matrix-scan method, element-indicator mechanization. As with the row/column-scan method, the time for "initialization" and tape reading and writing (three blocks) is also given.

The expression for the time (average case), converted to time in seconds, is

$$t_3 = 0.00283n^3 + 0.0115n^2 + 0.232n + 0.517.$$

For a 12×12 matrix, $t_3 \approx 9.8$ seconds, a saving of 9.4 seconds over the Munkres assignment program.

6. *Summary of Findings*

Of the three approximation methods defined and examined in this paper, only two appear to be worthy candidates for solving assignment problems. These two are the matrix-scan method and the row/column-scan method. Of these, the row/column-scan method is recommended.

However, it is possible that the main error concern might be with the worst possible solution, rather than with the expected solution. In that event, it is conjectured that the matrix-scan method would prove most satisfactory.

The expected value of the matrix-scan and the row-scan methods is of the same order of magnitude. The difference is bounded by a constant less than 3.6.

The upper bound on the absolute error of the expected value of the row-scan, or the column-scan, solution is $\ln(n + 1)$; for a maximal solution, the bound on the expected relative error, and on the relative error of the expected solution, is $[\ln(n + 1)]/n$. The corresponding bounds for the matrix-scan method for absolute error and relative error are $(\ln n + \gamma)$ and $(\ln n + \gamma)/n$, respectively. As yet, no expression has been found for the error generated by the combined-row/column-scan method; a weak upper bound is the upper bound for the row-scan method.

As regards solution time, the row/column-scan method is definitely superior; the matrix-scan method requires much more computation time. The operation-count difference of the two methods (the matrix-scan with the element-indicator mechanization and the row/column-scan with the indicator-element mechanization), representing the operations saved (for the average of the most favorable and unfavorable cases) is

$$\frac{17n^3 + 45n^2 + 34n}{12} - \frac{11n^2 + 13n + 8}{2} = \frac{17n^3 - 21n^2 - 44n - 48}{12}.$$

Furthermore, if the matrix is so large that it cannot be contained simultaneously in the computer, it is necessary, where the matrix-scan method is used, to read and reread into the computer segments of the matrix many times to obtain the solution. This necessity drastically increases the computation time. The row/column-scan method does not require the entire matrix to be held simultaneously in the computer. Thus, larger matrices may be solved than with the matrix-scan method, without causing an order-of-magnitude jump in the computation time.

If the time restriction is extremely critical, the row-scan method is suggested. With the indicator-element mechanization, the number of operations, for the average of the best and worst cases, for a problem of order n, is $(11n^2 + 13n)/4$.

Of course, the ultimate in a rapid rough-and-dirty approximation method would be to select, for each assignment problem, a fixed set of elements (for example, the main diagonal). The only two operations that are required for each individual selection are: row-to-column assignments and end-of-assignment test. Thus, only $2n$ operations are required with this method for an $n \times n$ matrix. The expected solution value for this rough-and-dirty method, assuming that the elements in the matrix are rectangularly distributed between 0 and 1, is $n/2$. Thus, the bound on the expected absolute error of the solution is $n/2$, and the bound on the expected relative error (for a maximal solution) is 1/2.

ACKNOWLEDGMENTS

This paper is based upon the author's thesis. For his thesis, the author wishes to thank his advisor, Dr. Benjamin H. Stevens, for his valuable direction, generosity of time, and constant encouragement; his second reader, Dr. Saul Gorn, for comments; and his third reader, Dr. Murray Gerstenhaber, for a suggestion of a simpler proof than the author had.

The author is greatly indebted to Dr. Louis Brickman and Dr. Leon Steinberg for their many helpful comments, suggestions of mathematical approaches, and constructive advice.

REFERENCES

1. Churchman; Ackoff; and Arnoff. *Introduction to Operations Research.* John Wiley, New York, 1957. Ch. 12, pp. 343–368.
2. Cramer, H. *Mathematical Methods of Statistics.* Princeton University Press, 1951.
3. Dantzig, G. The dual simplex algorithm. RAND Report RM-1270, RAND Corp., Santa Monica, Calif., 1954.
4. Feller, W. *Probability Theory and Its Applications.* John Wiley, New York, 1950.
5. Ford, L.; and Fulkerson, D. Solving the transportation problem. RAND Report RM-1736, RAND Corp., Santa Monica, Calif., 1956.
6. Gass, S. *Linear Programming.* McGraw-Hill, New York, 1958.

7. GERSTENHABER, M. A solution method for the transportation problem. *J. SIAM 6* (1958), 321–334.
8. KUHN, H. W. Hungarian method for the assignment problem. *Nav. Res. Logist. Quart. 2* (1955), 83–97.
9. KUHN, H. W. Variants of Hungarian method for assignment problems. *Nav. Res. Logist. Quart. 3* (1956), 253–258.
10. MOTZKIN, T. S. The assignment problem. *Proc. 6th Symp. Appl. Math.* VI, pp. 109–125, McGraw-Hill, New York, 1956.
11. MUNKRES, J. Algorithms for the assignment and transportation problems. *J. SIAM 5* (1957), 32–38.
12. VON NEUMANN, J. A certain-zero-sum two-person game equivalent to the optimal assignment problem. In H. Kuhn and A. Tucker (eds.), *Contribution to the Theory of Games II* (Ann. Math. Study No. 28), pp. 5–12, Princeton University Press, 1953.