W. E. Donath

Algorithm and Average-value Bounds for Assignment Problems

Abstract: A new suboptimal intermediate-speed algorithm which uses $n^2 \ln n$ steps is developed for the assignment problem. Upper and lower bounds are derived, using this algorithm and other methods, for the average values of three classes of $n \times n$ assignment problems:

1. When the elements of the matrix are random numbers uniformly distributed over the range 0 to 1, the average optimal value is smaller than 2.37 and larger than 1 for problems with large n. Experimentally the value is about 1.6.
2. When the elements of the matrix are random numbers such that the probability of being less than x is $x^{k+1}(k \neq 0)$, asymptotic expressions for the upper and lower bounds of the average optimal value are $C_k n^{k/(k+1)}$ and $C_k[(k+1)/k]n^{k/(k+1)}$, respectively.
3. When each column of the matrix is a random permutation of the integers 1 to n, asymptotic upper and lower bounds are $2.37n$ and $1.54\,n$, respectively. Experimentally the value is about $1.8\,n$.

Introduction

The assignment problem can be defined as follows: Given an $n \times n$ matrix a_{ij}, find the permutation p_i on n numbers such that the value or "cost"

$$C = \sum_{i=1}^{n} a_{ip_i} \tag{1}$$

is a minimum. As a simple example of such a problem, consider the case of n men being assigned to n different jobs, where the cost of training each man for each job is different. We want to find the man-to-job assignment such that the total training cost is a minimum. If we let each man be identified by the index i and each job by j and denote the training cost as a_{ij}, the problem is exactly in the form of Eq. (1). Because of its wide range of application, this problem has been studied from various points of view by a number of authors.[1-11]

Algorithms exist[1-9] that are guaranteed to find the optimal solution and much of the past work is concerned with finding solution methods. It is known that such methods exist and that it is possible to find an optimal solution within n^3 steps.[5,6] However, it is also of considerable interest to know the expected average minimal value of the solution to any given set of problems or, alternatively, what bounds exist for the average optimal cost. Kurtzberg[10] studied this question (in connection with the development of suboptimal algorithms) for the case in which the a_{ij} are random numbers uniformly distributed in the range 0 to 1 and found bounds $n(n+1)$ and $\ln n$ for this problem.

We derive a new upper bound of 2.37 for the average result of this problem using a new method of proof. Our experimental results indicate that the average optimal value is about 1.6. We also extend Kurtzberg's method to the case in which the elements of the matrix have a probability distribution $p(x)$ with values

$$p(x) = \begin{cases} 0, & x < 0; \\ (k+1)x^k, & 0 \leq x \leq 1; \quad \text{or} \\ 0, & x > 0. \end{cases} \tag{2}$$

In addition, we consider the case in which the columns of each matrix are random permutations of the integers 1 to n. Our methods give lower and upper bounds of $1.54\,n$ and $2.37\,n$ for large n, while experimentally the result is found to be about $1.8\,n$.

In the first section we describe the new algorithm and derive a theorem that gives an upper bound of the average cost for $n \times n$ matrices whose columns are random permutations of the integers 1 to n. Then we prove a lemma that allows us to derive the corresponding result for $n \times n$ matrices whose elements are random and uniformly distributed over the range 0 to 1. The derivation of upper and lower bounds for the nonuniform-distribution case is described in the next section. The results are based

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

on the column-scan method already applied by Kurtzberg to the uniform-distribution case. Finally we derive a lower bound for the random-permutation matrices. Here an enumeration method is used to obtain the result.

Multiple-column-scan algorithm and derived bounds

The algorithm will generate suboptimal results; i.e., it will not solve Eq. (1) exactly, but will yield an approximate solution. It could perhaps be described as an abbreviated version of the optimal method developed by Karp[5] and by Edmonds[6] or as an expanded version of the column scanning procedure used by Kurtzberg.[10] In the column-scan method each column is searched in turn for the lowest-valued element (LVE) among those that are in "free" rows. Initially all rows are free, so the search of the first column necessarily yields the minimal entry. The row in which the LVE is found is assigned to the column being searched. All elements in an assigned row are exempt from examination during the search of subsequent columns.

In the multiple-column-scan method we again consider each column in turn, searching for the LVE, but exempting no element from examination. If the LVE is in a free row, that row is assigned to the column being scanned. If, however, the LVE found during the scan of column i is in a row that has previously been assigned to another column, say k_1, a secondary scanning procedure is begun.

We call column i the primary scanning column and k_1 the first secondary scanning column. The secondary scanning procedure can lead to a reassignment of rows, so we institute a current-assignment list and update it at each step. After the first scan of column i, in which the conflict of row requests was discovered, columns i and k_1 both have the same row assignment in the list.

Now, for both of the scanning columns, we look for the next-LVE in the column, i.e., the next-larger entry compared with the one in the current-assignment list. If the next-LVE is found in a free row for one of the scanning columns, the scan for column i is terminated and new assignments are generated as follows: If the free row has been found for column i, that row is assigned to column i and the assignment of the secondary column is left unchanged. Should the free row have been found for the secondary column k_1, however, this assignment is made and column i is given the row assignment that necessitated the secondary scanning step, i.e., its entry in the current-assignment list.

When it happens that neither of the next-LVE's is found in a free row, we temporarily make the assignments indicated by the next-LVE's and update the current-assignment list. Then we add to the list another secondary scanning column k_2 which is the column previously assigned to the row now in conflict with the current assignment of the primary scanning column. Next we scan columns i, k_1 and k_2 for the next-LVE in each relative to the value of the entry for each in the current-assignment list.

The assignment procedure for three (or more) columns scanned jointly is a simple extension of the two-column procedure. If a free row is found, it is assigned to the corresponding column. If this latter column was a secondary scanning column, the primary column is given the row assignment that the secondary column had at the termination of scanning for primary column $i - 1$ and all other columns retain the assignments which existed at that time. If free rows are found during any scanning step for more than one column (primary or secondary), the choice of which column to reassign is arbitrary.

Again, should no column have a free row as its current assignment, we add another column to the set of secondary scanning columns and repeat the multiple-scan procedure. The column to be added is always that one whose previously assigned row is in conflict with the new current assignment of the primary column. (Whenever the scan of the primary column results in a previously assigned row, that row will have been assigned to a column which is not in the set of secondary scanning columns at the current stage.) Note that on the sth scanning of the primary column i there exist s scanning columns and that by the ith scanning of the primary column there certainly exists a free row.

Let us denote by q_{is} the probability that, after the sth scanning step, the scanning process has *not* terminated for the ith column. We can see that

$$q_{i0} = 1,$$

$$q_{i1} = \frac{i-1}{n} \quad \text{and}$$

$$q_{i2} \le \left(\frac{i-1}{n}\right)\left(\frac{i-2}{n-1}\right)^2.$$

[One factor $(i - 2)/(n - 1)$ corresponds to the probability that the secondary scanning column does *not* find a free row. Should this column not be assigned to the row containing the column's minimal element, this factor would be smaller because at least one assigned row would be excluded from the scan for this secondary column.] Continuing the sequence we have

$$q_{i3} \le \left(\frac{i-1}{n}\right)\left(\frac{i-2}{n-1}\right)^3\left(\frac{i-3}{n-2}\right)^2,$$

$$\vdots \qquad \vdots$$

$$q_{is} \le q_{i,s-1}\left(\frac{i-s}{n+1-s}\right)\prod_{j=1}^{s-1}\left(\frac{i-j-1}{n-j}\right). \tag{3}$$

We can also see that

$$q_{is} \le \left(\frac{i-1}{n}\right)^{s(s+1)/2} \tag{4}$$

because

$$\frac{i-j-1}{n-j} < \frac{i-1}{n} \quad \text{if} \quad 0 \le j \le i \le n.$$

We can now make use of the algorithm to prove the first theorem.

• *Theorem 1*

Consider the set of all $n \times n$ matrices such that each column is a permutation of the integers 1 to n. The average cost $\bar{C}$ for assignment problems of this class is less than $2.37n$.

Proof

If we apply the above algorithm to this problem, the incremental cost of assigning a column on the sth step (each step is the finding of the next-LVE for all scanning columns) to some column i is just s, independent of whether the column is a primary or a secondary scanning column. The probability of terminating on the sth step is given by

$$q_{i,s-1} - q_{is}, \tag{5}$$

so that the average cost $\bar{C}$ is bounded:

$$\begin{aligned}
\bar{C} &\le \sum_{i=1}^{n} \sum_{s=1}^{i} s(q_{i,s-1} - q_{is}) \qquad (6)\\
&\le \sum_{i=1}^{n} \sum_{s=0}^{i} q_{is}\\
&\le \sum_{i=1}^{n} \left(1 + \sum_{s=1}^{n} q_{is}\right)\\
&\le \sum_{i=1}^{n} \left[1 + \sum_{s=1}^{n} \left(\frac{i-1}{n}\right)^{s(s+1)/2}\right]\\
&< n + n \sum_{s=1}^{n} \int_0^1 x^{s(s+1)/2}\, dx\\
&< n\left\{1 + \sum_{s=1}^{\infty} [1 + s(s+1)/2]^{-1}\right\}\\
&< (2\pi n/\sqrt{7}) \tanh (\pi\sqrt{7}/2)\\
&< (2.3722 \cdots)n.
\end{aligned} \tag{7}$$

We can also compute an upper bound of the average computing time. In solving an assignment problem it is advantageous to initially apply a sorting procedure to each column to facilitate finding the entries in order (i.e., to generate a ranking matrix). The sort requires a number of computing steps of the order of $n^2 \ln n$. At the sth scanning step of our procedure we require s computing steps, so the average computing time required is

$$\bar{T}_n = \sum_{i=1}^{n} \sum_{s=1}^{i} s q_{is}. \tag{8}$$

We find

$$\begin{aligned}
\bar{T}_n &\le \sum_{i=1}^{n} \sum_{s=1}^{i} s\left(\frac{i-1}{n}\right)^{s(s+1)/2}\\
&< \sum_{s=1}^{n} \sum_{i=1}^{n} s\left(\frac{i-1}{n}\right)^{s(s+1)/2}\\
&< \sum_{s=1}^{n} ns \int_0^1 x^{s(s+1)/2}\, dx\\
&< \sum_{s=1}^{n} ns[1 + s(s+1)/2]^{-1}\\
&< 2n \ln n.
\end{aligned} \tag{9}$$

We can see that for large n the computing time is dominated by the initial sorting process. The permutation matrices just considered for the integers 1 to n, however, are equivalent to the ranking matrices and the initial sort is not necessary.

Uniform distributions

We are interested in the relation between permutation matrices and matrices in which the elements are random variables. Let us denote the set of all matrices in which each of the n columns is a permutation of n integers by R_n. We now consider the set of problems A_n, where $A \in A_n$ is an $n \times n$ matrix whose elements are random variables identically but independently distributed. We also consider the ranking matrix $M(A)$, where each element of A is replaced by its column ranking. That is, if a_{ij} is the sth-LVE in column j, it is replaced by s: $a_{ij} \rightarrow s$. We note that, for some $X \in R_n$ and some $A \in A_n$, the probability that $M(A) = X$ is just $|R_n|^{-1}$. We now prove the following lemma.

• *Lemma*

If the distribution of the random variable used in generating A_n is such that the expected value $\mathrm{E}_i(z_1, \cdots, z_n)$ of the ith lowest number of n numbers $z_1, \cdots, z_n$ satisfies

$$\mathrm{E}_i(z_1, \cdots, z_n) \le i\alpha \tag{10}$$

for some α, and if there exist assignments for every $X \in R_n$ having a cost $C(X)$ such that

$$|R_n|^{-1} \sum_{X \in R_n} C(X) \le \beta \tag{11}$$

for some β, then there exist assignments for all $A \in A_n$ having costs $C(A)$ such that

$$\mathop{\mathrm{E}}_{A \in A_n} [C(A)] \le \alpha\beta. \tag{12}$$

Proof

Given $X \in R_n$, consider all $Y \in A_n$ such that $M(Y) = X$. We consider the cost $C(X)$ that enters Eq. (11), i.e., which must exist to give an expectation less than β. Assume that

Table 1 Summary of computed (experimental) average optimal costs.

n	Number of matrices	Average optimal cost: Uniformly distributed random variable matrices ($\bar{C}$)	Average optimal cost: Integer permutation matrices ($\bar{C}/n$)
10	25	1.474	1.624
20	25	1.374	1.750
30	25	1.561	1.738
40	25	1.628	1.767
50	25	1.516	1.781
75	10	1.541	1.841
100	10	1.626	1.800

the ranking matrix assigns the columns to their r_1th, r_2th, $\cdots$, r_nth rows so that

$$C(X) = r_1 + r_2 + \cdots + r_n. \tag{13}$$

The expectation $\mathrm{E}\langle C[\{Y : M(Y) = X\}]\rangle$ is related to $C(X)$ by

$$\begin{aligned}\mathrm{E}\langle C[\{Y : M(Y) = X\}]\rangle &\leq \sum_i \mathrm{E}_{r_i}(z_1, z_2, \cdots, z_n)\\ &\leq \alpha \sum r_i\\ &\leq \alpha C(X),\end{aligned} \tag{14}$$

because the columns are mutually independent and, for each individual column, we have the set of all sequences of n random numbers in a specified order. The number of elements in $\{Y : M(Y) = X\}$ is independent of Y and it follows that

$$\begin{aligned}\mathrm{E}[C(A_n)] &= \sum_{X \in R_n} \mathrm{E}\langle C[\{Y : M(Y) = X\}]\rangle\, |R_n|^{-1}\\ &\leq \alpha \sum_{X \in R_n} C(X)\, |R_n|^{-1}\\ &\leq \alpha\beta.\end{aligned} \tag{15}$$

We can now easily prove the final theorem of this section:

• *Theorem 2*

It is given that A_n is a set of matrices whose elements are random numbers uniformly distributed in the range 0 to 1. Then

$$\mathrm{E}[C(A_n)] < 2.37n/(n + 1). \tag{16}$$

Proof

For n random numbers uniformly distributed in the range 0 to 1, the average value of the ith lowest number is

$$\mathrm{E}_i(z_1, \cdots, z_n) = i/(n + 1), \tag{17}$$

so that α of the preceding lemma is just $(n + 1)^{-1}$. Theorem 1 specifies $\beta < 2.37n$ and by substituting these values in Eq. (12) we obtain Eq. (16).

The upper bound of the average value of A_n is thus 2.37. Previous upper and lower bounds of A_n with uniform distribution were due to Kurtzberg[10] and are (asymptotically) $\ln n$ and 1, respectively. To estimate the *actual* average optimal cost, random-number matrices were generated on a computer and the optimal cost was determined using the Hungarian method.[3] The results are listed in Table 1; it appears that the average optimal cost for large n is about 1.6. We also computed the upper bound of the average optimal cost as a function of n using Eqs. (3) and (6), the Lemma and Theorem 2. These values are given in Table 2.

Nonuniform distributions

One may encounter assignment problems generated in a two-, three- or higher-dimensional environment and it is often adequate to approximate the resulting cost matrices by independent random variables with linear, quadratic or higher-dimensional distribution functions. We now study the case in which the probability $p(x)dx$ that a given random number is in the range $[x, x + dx]$ is

$$p(x)dx = (k + 1)x^k dx, \quad 0 \leq x \leq 1. \tag{18}$$

We use the methods Kurtzberg[10] developed for the uniform distribution, i.e., for the lower bound of A_n we assume that every column is assigned to its LVE and for the upper bound we let column 1 be assigned to its LVE, column 2 to the LVE of the $n - 1$ remaining rows, column 3 to the LVE of the $n - 2$ remaining rows, etc. Let e_{kn} be the expectation value of the lowest of n random variables with the distribution Eq. (18). Then

$$e_{kn} = \int_0^1 \left[1 - \int_0^x p(x')\, dx'\right]^{n-1} x p(x)\, dx \tag{19}$$

and, if

$$P(x) \equiv \int_0^x p(x')\, dx',$$

then

$$\begin{aligned}e_{kn} &= \int_0^1 nx[1 - P(x)]^{n-1}\, dP(x)\\ &= \int_0^1 P(x)^{1/(k+1)}[1 - P(x)]^{n-1} dP(x).\end{aligned}$$

This latter expression is recognized as a beta function[12] and can also be written as

$$\begin{aligned}e_{kn} &= n\Gamma(n)\Gamma[k/(k + 1)]/\Gamma[n + k/(k + 1)]\\ &= \prod_{j=1}^{n} j[j - 1/(k + 1)]^{-1}.\end{aligned} \tag{20}$$

Table 2 Computed (experimental) upper bounds of the average optimal cost for uniformly distributed random variable matrices.

n	3	4	5	10	20	30	40	50	75	100
Upper bound	1.0417	1.1889	1.3003	1.6103	1.8474	1.9530	2.0153	2.0572	2.1211	2.1583

For $k > 0$ an asymptotic expression for large n is

$$\ln e_{kn} = \sum_{j=1}^{n} \ln \{1 - [j(k+1)]^{-1}\}$$
$$= -\sum_{j=1}^{n}\sum_{l=1}^{\infty} [j(k+1)]^{-l}/l$$
$$= -\sum_{l=1}^{n}\sum_{j=1}^{n} [j(k+1)]^{-l}/l$$
$$\approx -(k+1)^{-1} \ln n + \mathcal{O}(1),$$

so that

$$e_{kn} \approx C_k n^{-1/(k+1)}. \tag{21}$$

To obtain L, the asymptotic lower bound of the average result, we set (the method of Kurtzberg[10])

$$L = ne_{kn}$$
$$\approx C_k n^{k/(k+1)}. \tag{22}$$

The asymptotic upper bound U is similarly given by

$$U = \sum_{m=1}^{n} e_{km}$$
$$\approx \sum_{m=1}^{n} C_k m^{-1/(k+1)}$$
$$\approx C_k \int_0^n m^{-1/(k+1)}\, dm$$
$$\approx C_k[(k+1)/k] n^{k/(k+1)}. \tag{23}$$

The following values of C_k were computed: $C_1 = 0.7535$, $C_2 = 0.8953$, $C_3 = 0.9474$, $C_4 = 0.9602$ and $C_5 = 0.9742$. Note that the ratio U/L is $(k+1)/k$.

Lower bound of permutation matrices

Kurtzberg's method in the case of permutation matrices leads to the result that L (the lower bound) is n. We use an enumeration technique to give a somewhat larger (i.e., better) bound. The number W_n of distinct problems (wherein each column of the assignment matrix is a random permutation of the integers 1 to n) is $W_n = (n!)^n$.

Let W_k be the number of feasible solutions with cost k. We use the method of generating functions to derive a formula for W_k, i.e., set

$$W_n(x) = \sum_{k=n}^{\infty} W_k x^k. \tag{24}$$

Before solving the assignment problem there can be $n!$ column-to-row assignments and for each column the assigned row can be ordered $1, 2, \cdots, n$, while the other rows can have their elements arranged in $(n-1)!$ ways. To enumerate all possibilities we write the generating function $W_n(x)$ as

$$W_n(x) = n![(n-1)!(x + x^2 + x^3 + \cdots + x^n)]^n$$
$$= n![(n-1)!x(1-x^n)/(1-x)]^n. \tag{25}$$

The essential step in the further development here is that we look for the smallest value of k, say K, such that

$$\sum_{k \le K} W_k \ge W_n = (n!)^n, \tag{26}$$

which gives the best possible set of feasible solutions to this set of problems.

We assume that K is in the range $n \le K < 2n$. Then the W_k of interest are not affected if we simplify $W_n(x)$ to

$$W_n(x) = n![(n-1)!\, x/(1-x)]^n. \tag{27}$$

For this range of values we see that

$$W_k = \binom{k-1}{n-1} n!(n-1)!^n \tag{28}$$

and

$$\sum_{k \le K} W_k = \binom{K}{n} n!(n-1)!^n. \tag{29}$$

The optimal K is related to n by the inequality

$$\frac{K!}{(K-n)!} \ge \left[\frac{n!}{(n-1)!}\right]^n = n^n. \tag{30}$$

We use Stirling's approximation for large K,

$$K! \approx (2\pi)^{\frac{1}{2}} K^{k+\frac{1}{2}} e^{-k},$$

and therefore have to solve

$$(K + \tfrac{1}{2}) \ln K - (K - n - \tfrac{1}{2}) \ln (K - n) \approx n \ln n. \tag{30}$$

For large n we find

$$K \approx (1.5422\cdots)n \tag{31}$$

and we see that K is indeed less than $2n$ as we previously assumed.

The average cost $\bar{C}$ of this best possible set of feasible solutions,

Table 3 Summary of asymptotic bounds of average optimal solutions of the assignment problem.

Class ($n \times n$ matrix)	Bound Lower	Upper
Uniformly distributed random variables	1.00	$2.37\cdots$
Nonuniformly distributed random variables, $p(x) = (k+1)x^k$	$C_k n^{k/(k+1)}$	$C_k\left(\frac{k+1}{k}\right)n^{k/(k+1)}$
Random permutations of integers	$(1.54\cdots)n$	$(2.37\cdots)n$

Table 4 Comparison of algorithms for the assignment problem.

Average	*Method* *Row-scan*	*Multiple-column-scan*	*Hungarian (or dual)*
Optimal value (for uniformly distributed random variables)	$\ln n$	2.2	1.6
Computing time (proportionality)	n^2	$n^2 \ln n$ or $n \ln n$†	$\leq n^3$

† The lower value applies when the ranking matrix is available.

$$\bar{C} = \sum_{k\leq K} kW_k / \sum_{k\leq K} W_k, \tag{32}$$

is evaluated by considering

$$\begin{aligned} x\partial W(x)/\partial x &= \sum_{k\geq n} kx^k W_k \\ &= nn![(n-1)!]^n \\ &\qquad \times [x^n(1-x)^{-n} + x^{n+1}(1-x)^{-n-1}], \end{aligned} \tag{33}$$

which gives us a generating function for kW_k. Summing the first term in Eq. (33) over $k \leq K$, we obtain a contribution $n\binom{K}{n}$ to the numerator in Eq. (32), while the contribution from the second term is $n\binom{K}{n+1}$; for the denominator we get $\binom{K}{n}$. Because $\bar{C}$ is the best possible average cost, it is equivalent to the lower bound of the average optimal cost and therefore

$$\begin{aligned} L &= n\left[1 + \binom{K}{n+1}\binom{K}{n}^{-1}\right] \\ &= n[1 + (K-n)/(n+1)] \\ &= Kn/(n+1) \\ &\approx (1.5422\cdots)n. \end{aligned} \tag{34}$$

Again, to estimate the actual asymptotic average optimal cost, random-permutation matrices were generated on the computer and the optimal cost was determined by the Hungarian method.[3] The results are listed in the final column of Table 1; we conjecture that the average optimal cost is about $1.8n$.

Summary

There are basically two results, a new suboptimal algorithm with well-characterized performance and a better knowledge of bounds of average optimal solutions for three classes of assignment problems. These bounds are summarized in Table 3. The algorithms now available for this type of problem are compared in Table 4. The multiple-column-scan method developed here is intermediate in speed between the row-scan method and the Hungarian method and is also intermediate in value of the average optimal result. If, however, the ranking matrix is already known, the new algorithm is faster than the other two methods. In this case the multiple-column-scan method is a good candidate for solving the traveling-salesman problem when used with a dynamic programming approach.[14]

Acknowledgments

I am very much indebted to a number of people, A. J. Hoffman, A. G. Konheim, J. M. Kurtzberg and R. L. Russo, who were helpful with both suggestions and encouragement. In particular, A. J. Hoffman pointed out the need for the lemma and A. G. Konheim suggested a possible method of proof.

References

1. H. W. Kuhn, "Hungarian Method for the Assignment Problem," *Naval Res. Log. Quart.* **2,** 83 (1955).
2. H. W. Kuhn, "Variants of the Hungarian Method for Assignment Problems," *Naval Res. Log. Quart.* **3,** 253 (1956).
3. J. Munkres, "Algorithms for the Assignment and Transportation Problems," *SIAM J. Appl. Math.* (formerly *J. Soc. Indust. Appl. Math.*) **5,** 32 (1957).
4. L. R. Ford and D. R. Fulkerson, "Solving the Transportation Problem," *Report RM-1736,* RAND Corporation, Santa Monica, California 1956.
5. R. M. Karp, unpublished work.
6. J. Edmonds, unpublished work.
7. M. L. Balinski and R. E. Gomory, "A Primal Method for the Assignment and Transportation Problems," *Management Sci.* **10,** 578 (1964).
8. A. J. Hoffman and H. M. Markowitz, "A Note on Shortest Path, Assignment and Transportation Problems," *Naval Res. Log. Quart.* **10,** 375 (1963).
9. M. Klein, "A Primal Method for Minimal Cost Flows," *Management Sci.* **14A,** 205 (1967).

10. J. M. Kurtzberg, "On Approximation Methods for the Assignment Problem," *J. ACM* **9**, 419 (1962).
11. T. S. Motzkin, "The Assignment Problem," *Proceedings of the Sixth Symposium on Applied Mathematics,"* McGraw-Hill Book Co., Inc., New York 1956, p. 109.
12. D. V. Widder, *Advanced Calculus,* Prentice-Hall, Inc., New York 1947, p. 308.
13. W. E. Donath, "Statistical Properties of the Placement of a Graph," *SIAM J. Appl. Math.* **16**, 439 (1968).
14. M. Bellmore and G. L. Nemhauser, "The Traveling Salesman Problem: A Survey," *Operations Res.* **16**, 538 (1968).

Received January 29, 1969