J. Soc. Indust. Appl. Math.
Vol. 5, No. 1, March, 1957
Printed in U.S.A.

ALGORITHMS FOR THE ASSIGNMENT AND TRANSPORTATION PROBLEMS*†

JAMES MUNKRES

In this paper we present algorithms for the solution of the general assignment and transportation problems. In Section 1, a statement of the algorithm for the assignment problem appears, along with a proof for the correctness of the algorithm. The remarks which constitute the proof are incorporated parenthetically into the statement of the algorithm. Following this appears a discussion of certain theoretical aspects of the problem. In Section 2, the algorithm is generalized to one for the transportation problem. The algorithm of that section is stated as concisely as possible, with theoretical remarks omitted.

1. The Assignment Problem. The personnel-assignment problem is the problem of choosing an optimal assignment of n men to n jobs, assuming that numerical ratings are given for each man's performance on each job. An optimal assignment is one which makes the sum of the men's ratings for their assigned jobs a maximum. There are $n!$ possible assignments (of which several may be optimal), so that it is physically impossible, except for very small n, to consider all the different assignments one-by-one. The problem is to find a reasonably efficient algorithm for obtaining an optimal assignment. Such an algorithm has been given by H. Kuhn [3]; another algorithm, which is a variant of Kuhn's, appears in the present paper.

A mathematical statement of the problem follows: Let r_{ij} be the performance rating of man M_i for job J_j. A set of elements of a matrix are said to be *independent* if no two of them lie in the same line (the word "line" applies both to the rows and to the columns of a matrix). One wishes to choose a set of n independent elements of the matrix (r_{ij}) so that the sum of these elements is maximum. Let $r = \max_{i,j} r_{ij}$, and let $x_{ij} = r - r_{ij}$. An equivalent problem is to choose a set of n independent elements of the matrix $A = (x_{ij})$ such that the sum of these elements is minimum. This is the problem we consider. We assume for the present that the elements of A are integers.

Two remarks are in order: (1) There is a theorem of König which states: If A is a matrix, and m is the maximum number of independent zero ele-

* This work was done in connection with a project of the Operations Research group, Engineering Research Institute. The author is indebted to Dr. Jesse Wright for a valuable suggestion.

† Received by the editors August 8, 1956 and in revised form January 27, 1957.

ments of A, then there are m lines which contain all the zero elements of A. (2) It is readily seen that the solution of our problem is not changed if we replace the matrix (x_{ij}) by the matrix (y_{ij}), where $y_{ij} = x_{ij} - u_i - v_j$ (u_i and v_j are arbitrary constants).

These facts provide a basis for Kuhn's algorithm; M. Flood [1] has outlined it in the following form:

Step A. Subtract the smallest element in A from each element of A, obtaining a matrix A_1 with non-negative elements and at least one zero.

Step B. Find a minimal set S_1 of lines, n_1 in number, which contain all the zeros of A_1. If $n_1 = n$, there is a set of n independent zeros and the elements of A in these n positions constitute the required solution.

Step C. If $n_1 < n$, let h_1 denote the smallest element of A_1 which is not in any line of S_1. Then $h_1 > 0$. For each line in S_1, add h_1 to every element of that line; then subtract h_1 from every element of A_1. Call the new matrix A_2.

Step D. Repeat Steps B and C, using A_2 in place of A_1. The sum of the elements of the matrix is decreased by $n(n - n_k)h_k$ in each application of Step C, so the process must terminate after a finite number of steps.

To complete the algorithm, it is necessary to give a constructive procedure for carrying out Step B, i.e., for finding (1) a minimal set of lines which contain all zeros and (2) a maximal set of independent zeros. The present algorithm differs from Kuhn's at this point. Our algorithm follows.

In the course of the problem, certain lines will be distinguished; we will speak of these lines as *covered* lines. An element of the matrix is said to be non-covered, once-covered, or twice-covered, accordingly as it lies in precisely none, one, or two covered lines. We will distinguish some zero elements by means of asterisks and some by means of primes (we refer to "starred zeros" and "primed zeros", respectively).

* * *

Preliminaries. No lines are covered; no zeros are starred or primed. Consider a row of the matrix A; subtract from each element in this row the smallest element of this row. Do the same for each row of A. Then consider each column of the resulting matrix and subtract from each column its smallest entry. [This is similar to Step A above.]

Consider a zero Z of the matrix. If there is no starred zero in its row and none in its column, star Z. Repeat, considering each zero in the matrix in turn. Then cover every column containing a starred zero. [These starred zeros are independent.]

Step 1. Choose a non-covered zero and prime it. Consider the row containing it. If there is no starred zero in this row, go at once to Step 2. If there is a starred zero Z in this row, cover this row and uncover the column of Z.

Repeat until all zeros are covered. Go to Step 3.

Step 2. There is a sequence of alternating starred and primed zeros, constructed as follows: Let Z_0 denote the uncovered 0′. [There is only one.] Let Z_1 denote the 0* in Z_0's column (if any). Let Z_2 denote the 0′ in Z_1's row (we must prove that it exists). Let Z_3 denote the 0* in Z_2's column (if any). Similarly continue until the sequence stops at a 0′, Z_{2k}, which has no 0* in its column (this we must also prove). [Note that no column contains more than one 0* and no row more than one 0′, so that the sequence is uniquely specified. The sequence may, however, contain only one element. Now the column of Z_1 is not covered, so its row must be covered; hence there is a 0′ in this row (see Step 1). This 0′ serves as our Z_2. A similar argument applies to show that, given Z_{2i-1}, Z_{2i} exists. Now let us index the primed zeros 1, 2, 3, $\cdots$ in the order in which we primed them during Step 1. One sees readily from the directions in Step 1 that the index of Z_{2i} must be smaller than the index of Z_{2i-2}. It follows at once that the sequence does stop and, furthermore, that all the elements of the sequence $Z_0, \cdots, Z_{2k}$ are distinct elements of the matrix.]

Unstar each starred zero of the sequence and star each primed zero of the sequence. [The resulting set of starred zeros is easily seen to be independent. It is larger by one than the previous set of independent starred zeros.] Erase all primes, uncover every row, and cover every column containing a 0*. If all columns are covered, the starred zeros form the desired independent set. Otherwise, return to Step 1.

Step 3. [At this point, all the zeros of the matrix are covered. Each 0* is covered by precisely one line, so there are exactly as many covered lines as there are starred zeros. Now it is clear that any set of lines containing all the zeros of a matrix cannot contain fewer lines than the maximal number of independent zeros of the matrix. It follows that at this point, the starred zeros form a maximal set of independent zeros and the covered lines form a minimal set of lines containing all the zeros. Thus Steps 1 and 2, repeated several times perhaps, replace Step B of the previous outline.]

Let h denote the smallest non-covered element of the matrix; it will be positive. Add h to each covered row; then subtract h from each uncovered column. [This is the same transformation as is specified in Step C above.]

Return to Step 1, without altering any asterisks, primes, or covered lines. [One might think one should "erase all primes, uncover every row, and cover every column containing a 0*" before returning to Step 1, so that the input of Step 1 is the standard one. That this is unnecessary may be seen from the following argument: The effect of the transformation specified above is to decrease each non-covered element of the matrix by h, increase each twice-covered element by h, and leave each once-covered element unaltered. Each 0* and 0′ is once-covered, so each is still a zero

after the transformation. (Incidentally, this shows that $n_{k+1} \geqq n_k$, where n_i denotes the maximal number of independent zeros of A_i. A_k denotes the matrix before the transformation and A_{k+1} denotes the transformed matrix.) Let us index the primed zeros 1, 2, 3, $\cdots$ in the order in which they were primed previously. Imagine that we "erase all primes, uncover every row, and cover every column containing a 0*" before returning to Step 1. The directions in Step 1 tell us to find a non-covered zero, but do not specify which one of possibly several non-covered ones we should choose to prime. Hence we can consider the indexed zeros in the order of their indices, priming each one in turn, without violating the directions in Step 1. This will bring the configuration of asterisks, primes, and covered lines back to precisely the same one we had at the beginning of this paragraph. Hence there is no need to return to the standard input when passing from Step 3 to Step 1; and indeed such a return would be foolish.]

* * *

As remarked previously, Step 3 (or Step C) decreases the sum of the elements of the matrix, so that the algorithm has only a finite number of steps. One may prove the following stronger result: If $n_{k+1} = n_k$, then when one applies Step 1 to A_{k+1}, Step 2 will not occur, and one will begin Step 3 with *more* horizontal covered lines than in the application of Step 3 to A_k. [Every horizontal covered line of A_k is a covered line of A_{k+1} (see the preceding paragraph). The transformation by which we pass from A_k to A_{k+1} causes a zero Z to appear in some uncovered position. Z must have a 0* in its row (since otherwise Step 2 would apply to A_{k+1} and we would have $n_{k+1} > n_k$), so that this row is covered by the time we reach the beginning of Step 3.]

From this result it follows that our initial assumption that the elements of A were integers is not necessary to the operation of the algorithm. This assumption was used only to show that the algorithm had a finite number of steps. The result just proved shows that after at most n applications of Step 3, the maximal number of zeros in the matrix must be increased. The finiteness of the algorithm follows.

We may also use this result to obtain an absolute maximum for the number of operations needed to solve completely any $n \times n$ assignment problem, using the present algorithm. The operations considered are the following elementary ones: Scan a line, cover or uncover a line, add to or subtract from a line, star or unstar a zero, prime or unprime a zero. The maximum is obtained as follows: Suppose that we have a matrix with m starred independent zeros. We find a maximum for the number of operations necessary to obtain a matrix with $m + 1$ starred independent zeros. We assume the worst possible situation, in which each application of Step

3 serves only to increase the number of horizontal covered lines by one on the next repetition of Step 1, until finally one has all the covered lines horizontal. We note first that after the Preliminaries, each row and column of the matrix contains at least one zero and that this situation never changes. [The transformation of Step 3 removes only those zeros which are twice-covered. But each line containing such a zero contains also a 0*, which remains 0 after the transformation.]

In the worst situation, the Preliminaries require no more than $5n + 4$ operations, at the end of which one has one starred zero.

Suppose we begin Step 1 with m starred zeros, and no primed zeros or horizontal covered lines. The initial application of Step 1 requires at most $n + 4$ operations, ending with one horizontal covered line. (Each uncovered column contains a zero, so there will be at least one such line. We assume the worst situation, in which there is only one.) Step 3 can take no more than $2n + m$ operations (of which n are "scan a column", m are "add to a row", and n are "subtract from a column"), after which we apply Step 1 and require $n + 4$ operations to obtain a matrix with one more horizontal covered line (again, this is the worst situation possible). This repeats until we have m horizontal covered lines, at which point continuation of Step 1 must lead to Step 2 (each uncovered row contains a zero). This continuation requires at most $n + 1$ operations, and Step 2 takes at most $7m + 3$ operations (of which $2m$ are "scan a line", $2m + 1$ are "erase a prime or asterisk", $m + 1$ are "star a zero", and $2m + 1$ are "cover or uncover a line"). Then after at most

$$(n + 4) + (2n + m)(m - 1) + (n + 4)(m - 1) + (n + 1) + (7m + 3)$$

operations, one has a matrix with $m + 1$ starred independent zeros.

Sum this expression from $m = 1$ to $m = n - 1$, and add on the $5n + 4$ operations required initially. The final maximum on the number of operations needed is

$$(11n^3 + 12n^2 + 31n)/6.$$

This maximum is of theoretical interest, since it is so much smaller than the $n!$ operations necessary in the most straightforward attack on the problem.

2. The Transportation Problem. The transportation problem may be stated as follows: Let $D = (d_{ij})$ be an $n \times m$ matrix of non-negative integers and let r_i $(i = 1, \cdots, n)$ and c_j $(j = 1, \cdots, m)$ be positive integers such that $\sum r_i = \sum c_j = N$. Determine values of the variables x_{ij} which minimize the sum $\sum_{i,j} x_{ij} d_{ij}$, subject to the conditions

$$x_{ij} \geqq 0, \qquad \sum_{i=1}^{n} x_{ij} = c_j, \qquad \sum_{j=1}^{m} x_{ij} = r_i.$$

One may think of the following physical situation: There are N ships placed at positions $P_1, \cdots, P_n$, and r_i denotes the number of ships at position P_i. One wishes to move these ships to new positions $Q_1, \cdots, Q_m$, so that there will be c_j ships at position Q_j. The number d_{ij} is the cost of moving a ship from position P_i to position Q_j. The number x_{ij} stands for the number of ships we order to be moved from P_i to Q_j; it will be called the *quota* assigned to d_{ij}. The problem is to choose these quotas so that the total cost of moving the ships is as small as possible. Such a choice of quotas will be called an optimal assignment. There is always an optimal assignment for which the x_{ij} are integers, so that the troublesome question, "How does one move one-half a ship from P_i to Q_j?" need not arise. Our algorithm enables us to obtain such an integral-valued solution.

This problem is also called the distribution problem; see [1] for other interpretations of it. We note that L. Ford and D. Fulkerson have also given a generalization of Kuhn's method to this problem [2].

A statement of the algorithm follows. We work with the cost matrix $D = (d_{ij})$. In the course of the problem, we will distinguish certain lines of the matrix, calling them *covered* lines, and we will distinguish certain zero elements of the matrix by means of asterisks and primes (as in the preceding algorithm). In addition, we will assign to each element of the matrix a non-negative quota x_{ij}, which may be changed in the course of the problem. Each element of the matrix whose quota is positive will be called *essential* (these elements will always be zeros). At any stage of the problem, the number $c_j - \sum_i x_{ij}$ will be called the *discrepancy* of the j^{th} column at that stage, and the number $r_i - \sum_j x_{ij}$ will be called the discrepancy of the i^{th} row. These discrepancies will always be non-negative; when all of them vanish, the corresponding quotas are a solution to the problem.

* * *

Preliminaries. All quotas are zero; no lines are covered; no zeros are starred or primed. Subtract from each row of the matrix D its smallest entry; then subtract from each column of the resulting matrix its smallest entry.

Find a zero Z in the matrix. If the discrepancies of both its row and its column are positive, increase the quota assigned to Z until the smaller of these discrepancies is zero. Repeat, for each zero in the matrix. Then cover every column whose discrepancy is zero.

Step 1. Choose a non-covered zero and prime it. Consider the row containing it. If the discrepancy of this row is positive, go at once to Step 2. Otherwise (if this discrepancy is zero) cover the row; then star *each* twice-covered essential zero Z in the row and uncover Z's column.

Repeat until all zeros are covered. Go to Step 3.

Step 2. There is a sequence of alternating starred and primed zeros, constructed as follows: Let Z_0 denote the uncovered 0′ (there is only one). Let Z_1 denote the 0* in Z_0's column (if any). Let Z_2 denote the 0′ in Z_1's row. Let Z_3 denote the 0* in Z_2's column (if any). Similarly continue, until the sequence stops at a 0′, Z_{2k}, which has no 0* in its column. (Since no column contains more than one 0*, and no row more than one 0′, this sequence is unique. Note, however, that it may contain only one element.)

The discrepancy of Z_0's row is positive, the discrepancy of Z_{2k}'s column is positive, and the quota assigned to each 0* of the sequence $Z_0, \cdots, Z_{2k}$ is positive. Let h be the smallest of these positive numbers. Increase the quota of each 0′ in the sequence by h, and decrease the quota of each 0* in the sequence by h.

Erase all asterisks and primes, uncover every row, and cover every column whose discrepancy is zero. Return to Step 1.

Step 3. Let h denote the smallest non-covered element of the matrix; it will be positive. Add h to every covered row and subtract h from every uncovered column. Return to Step 1, without altering any asterisks, primes, or covered lines.

* * *

Note that in the case $c_j \equiv r_i \equiv 1$, the problem is the assignment problem, and the algorithm reduces to our algorithm for that problem.

REFERENCES

1. Flood, Merrill M., *The traveling-salesman problem*, Operations Research, 4 (1956), pp. 61–75.
2. Ford, L. R., Jr. and D. R. Fulkerson, *A simple algorithm for finding maximal network flows and an application to the Hitchcock problem*, RAND Corporation, Santa Monica, P-743, Sept. 26, 1955.
3. Kuhn, H. W., *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly, 2 (1955), pp. 83–97.

University of Michigan