AN ALGORITHM FOR
THE \( n \times n \) OPTIMUM ASSIGNMENT PROBLEM

KABEKODE V. S. BHAT and BHARAT KINARIWALA

Abstract.
In this paper we present an algorithm for finding an optimum assignment for an \( n \times n \) matrix \( M \) in \( n \) iterations. The method uses systematic permutations on the rows of \( M \) and is based on the properties of optimum assignments. The implementation presented in the paper requires at most \( O(n^3) \) in time and \( n^2 + 6n \) memory locations for solving a dense \( n \times n \) problem.

Key words. Algorithm, assignment, permutation, cycles, cost.

1. Introduction.
Optimum resource allocation to several tasks is of considerable interest in many system problems. Computer and communication systems, management and society systems must frequently consider questions of resource allocation [1]–[8]. One formulation for an \( n \times n \) optimum assignment problem is given below.

Let \( M = [m_{ij}] \) be an \( n \times n \) matrix where \( m_{ij} \geq 0 \) is the cost of assigning resource \( i \) to task \( j \), \( i, j = 1, 2, \ldots, n \). Find an \( n \times n \) permutation matrix \( Q \) such that

\[
\text{Trace}[QM] = \min_{P \in \mathcal{L}} \{\text{Trace}[PM]\}
\]

where \( \mathcal{L} \) is the set of all \( n \times n \) permutation matrices, \( Q \) is an optimum permutation and \( \text{Trace}[QM] = w_0 \) the optimum cost for the problem. The problem has also been expressed as a linear programming model and several approaches are available for solving the problem [7]–[14].

In this paper we present a new algorithm for finding an optimum permutation. Our method with \( n \) iterations uses properties of optimum assignments and systematically permutes the rows of the matrix \( M \) so that at the end of iteration \( k \) an optimum assignment for the \( k \times k \) principal submatrix defined by the first \( k \) rows of the permuted \( M \) is along the principal diagonal of the submatrix. A modification of Dijkstra's algorithm [15], [16] is used for finding a zero cost cycle as well as other relevant costs that are needed in each iteration. The time required by the algorithm is at most \( O(n^3) \) which is of the same order as given in recent works [13], [14]. For a dense matrix the algorithm requires \( n^2 + 6n \) memory locations.

This work was supported by the National Science Foundation Grant NSF ENG 74-19788.

Received August 17, 1978. Revised April 5, 1979.
In section 2 of this paper we present relevant concepts useful in our algorithm. Section 3 presents the algorithm and its properties and gives an example for illustration.

2. Preliminaries.

We associate a directed weighted graph \( G(M) = (V, E) \) where \( V \) is a set of \( n \) vertices and \( E \) a set of \( |E| \) directed edges with \( M \) as follows: column \( i \) of \( M \) corresponds to vertex \( v_i \in V \) and \( m_{ij} \) \((0 \leq m_{ij} < \infty)\) is the cost of edge \((v_i, v_j) \in E\). For \( v_{j_1}, v_{j_2} \in V \) a path from \( v_{j_1} \) to \( v_{j_2} \) denoted by \( v_{j_1} \rightarrow v_{j_2} \) is a sequence of directed edges \((v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \ldots, (v_{j_{r-1}}, v_{j_r})\) where \( v_{j_1}, v_{j_2}, \ldots, v_{j_r} \) are all (with the possible exception of \( v_{j_1} \) and \( v_{j_r} \)) distinct. The cost of a path is given by the sum of the costs of the edges in the path. If there is a set of paths from \( v_i \) to \( v_j \), then a path with minimum cost in the set is denoted by \( v_i \ast v_j \). A path of the type \( v_i \rightarrow v_i \) is a cycle.

Given \( M \), a solution to the optimum assignment problem consists in finding \( S = \{t_1, t_2, \ldots, t_n\} \) a set of index pairs \((t_1, 1), (t_2, 2), \ldots, (t_n, n)\) where all \( t_i \) are distinct and such that \( \sum_{i=1}^{n} m_{t_i} = w_0 \). The set \( S \) is an optimum assignment for \( M \). In the graph context the following can be stated:

**Theorem 1.** An optimum assignment for \( M \) corresponds to a set of vertex disjoint cycles having total number of \( n \) edges and total cost \( w_0 \) in \( G(M) \).

**Proof.** Let \( E(S) \) denote the set of edges in \( G(M) \) corresponding to an optimum assignment \( S \). Each vertex of \( G(M) \) must have exactly one edge from \( E(S) \) directed towards the vertex and one edge from \( E(S) \) directed away from the vertex. Each vertex of \( G(M) \) must be in a cycle with all edges in \( E(S) \), and no vertex can be in more than one such cycle. Thus \( E(S) \) constitutes a set of vertex disjoint cycles with a total number of \( n \) edges. Total cost of the edges in \( E(S) \) must be \( w_0 \) if \( S \) is an optimum assignment.

While Theorem 1 serves to characterize optimum assignments, Lemmas 1 and 2 (which can easily be verified) are useful in finding optimum assignments.

**Lemma 1.** Let \( Q \) be an optimum permutation for the matrix \( \hat{P}M \) where \( \hat{P} \) is an \( n \times n \) permutation matrix. Then \( \hat{P} = Q \hat{P} \) is an optimum permutation for \( M \).

**Lemma 2.** Let \( M' = M + RU + UC^T \) be a modified cost matrix where \( R, C \) are \( n \)-dimensional vectors whose components are finite real numbers and \( U \) is a vector of \( n \) ones. Then \( M \) and \( M' \) have the same optimum assignments.

A \( k \times k \) matrix \( M \) has property \( A \) if all \( m_{ij} \geq 0 \) and \( m_{ii} = 0 \) for \( i = 1, 2, \ldots, k \). A matrix \( M \) has property \( B \) if it has property \( A \) and \( m_{kk} = 0 \). A \( (k + 1) \times (k + 1) \) matrix \( M^{(k+1)} = \begin{bmatrix} m_{ij}^{(k+1)} \end{bmatrix} \) is said to have resulted from bordering \( M^{(k)} \) if \( m_{ij}^{(k+1)} = m_{ij}^{(k)} \) for \( i, j = 1, 2, \ldots, k \).