THE OPTIMAL ASSIGNMENT PROBLEM

A Thesis

Presented to

the Division of Mathematics

Emporia State University

In partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Laura J. Oliver

July, 1992

AN ABSTRACT OF THE THESIS

Laura J. Oliver for the Master of Science

in Mathematics presented on May 7 ,1992

Abstract Approved: Elizabeth S. Yanik

This thesis is intended for an audience familar with basic linear algebra. The topic covered is the assignment problem. Chapter 1 is the introduction to the problem and the assumptions made. Chapter 2 discusses Kuhn's algorithm for solving the assignment problem. An example is done throughout the discussion. Chapter 3 discusses the Hungarian method for solving the assignment problem. An example follows this method. Chapter 4 discusses finding the optimal solution and Chapter 5 summarizes the two methods and gives a direction for further study.

Elizabeth G. Yanik

Approved for the Major Division

Faye N. Vowell

Approved for the Graduate Council

Table of Contents

CHAPTER 1

Introduction

The assignment problem is a well studied problem in combinatorics and operations research. The problem has arisen in a variety of contexts and, hence, it has received many names such as the marriage problem, the maximal transversal problem, and the name which we will use in this paper, the assignment problem.

Let us first describe it in the context of the marriage problem. Given n men, m_1, m_2, m_3, . . . , m_n and k women, w_1, w_2, w_3, . . . , w_k we assign to each pair of a woman and man a 1 if they are willing to marry and a 0 if not. The marriage problem asks, "What is the maximum number of marriages possible." Actually the problem considered in this work is a bit more general. Suppose to each man, and each woman, a positive integer could be assigned (a happiness rating?) to their future as a married couple. The question addressed here is how the marriages should be arranged to maximize the total level of happiness. [4, p.620]

The problem in this work will consider the case n=k. Mathematically stated the assignment problem is as follows: Let R be an n x n matrix with positive integer entries. The optimal assignment problem consists of choosing n entries, so that no two chosen entries come from the same row or column and such that the sum of the entries is maximized.

Consider the assignment problem in the following context. Given n individuals and n jobs, the entry in the i^{th} row j^{th} column, r_{ij}, of R denotes the rating value given to the i^{th} individual doing the j^{th} job. The optimal assignment problem seeks to determine the "best" possible pairing of individuals to jobs.

Mathematically, the assignment problem can be stated as follows:

$$\text{Maximize: } \sum_{i,j=1}^{n} r_{ij_i} \text{ with } (j_1, j_2, j_3, \ldots, j_n)$$
$$\text{a permutation of } (1, 2, 3, \ldots, n)$$

The General Assignment Problem makes several assumptions:

A1. The number of individuals equals the number of jobs.

A2. Each individual can do each job.

A3. No two individuals are paired to the same job; and no individual is paired to more than one job.

This paper will cover two methods of solving the General Assignment Problem. First, Kuhn's method will be discussed which deals with using the dual of the problem and second, the Hungarian method which uses a more direct approach to solving the problem.

CHAPTER 2

Kuhn's Algorithm

STATEMENT OF THE PROBLEM

The first algorithm to be discussed, Kuhn's method, solves the problem by finding a solution to what Kuhn referred to as a dual problem. Kuhn's dual is stated in the following manner

$$\text{minimize} \quad \sum_{i=1}^{n} (u_i + v_i)$$

$$\text{Subject to} \quad u_i + v_j \geq r_{ij}$$

Note that if u_i's and v_{j_i}'s can be found such that $u_i + v_{j_i} = r_{ij_i}$, where v_{j_i} is the job assigned to individual i and $(j_1, j_2, j_3, \ldots, j_n)$ is a permutation of (1, 2, 3, . . ., n), then the original maximizing problem has been solved.

Since $u_i + v_j \geq r_{ij}$ for all i, j

Then $\sum_{i=1}^{n} (u_i + v_i) \geq \sum_{i=1}^{n} r_{ij_i}$ with $(j_1, j_2, j_3, \ldots, j_n)$ a

permutation of (1, 2, 3, . . ., n)

Then $\min \sum_{i=1}^{n} (u_i + v_i) \geq \max \sum_{i=1}^{n} r_{ij_i}$ r_{ij}

Thus if there exists a combination of u_i, and v_{j_i} such that $u_i + v_{j_i} = r_{ij_i}$ for i = 1, 2, 3, . . . n, and $(j_1, j_2, j_3, \ldots, j_n)$ is a permutation of (1, 2, 3, . . ., n) then the minimization problem and the original maximization problem have been solved.

It is interesting to note that this method predates the simplex method.[1,p.206] The label of dual in Kuhn's method is consistent with the use of that term in the theory of

linear programming in the following sense. The optimal assignment problem can be stated as follows:

Maximize: $\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}r_{ij}$

Subject to: $\sum_{i=1}^{n} x_{ij}=1$ for each j,

$\sum_{j=1}^{n} x_{ij}=1$ for each i,

$x_{ij}=0,1,$

where $x_{ij} = 1$ means individual i is paired to job j, and $x_{ij} = 0$ means that individual i is not paired to job j. It has been shown by [1, p. 500] that this is equivalent to

Maximize: $\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}r_{ij}$

Subject to: $\sum_{i=1}^{n} x_{ij}=1$ for each j,

$\sum_{j=1}^{n} x_{ij}=1$ for each i,

$x_{ij}\geq 0.$

Then the linear programming dual of this problem is given by

Minimize: $\sum_{i=1}^{n} u_i+v_i.$

Subject to: $u_i + v_j \geq r_{ij}$.

u_i, v_j unrestricted.

Kuhn's algorithm considers the problem from the dual point of view. Kuhn has developed a computational method that will use this dual in an effective manner.

Recall the assumptions A1 - A3 made in Chapter 1. For Kuhn's method we also assume that the entries r_{ij} in the R matrix are positive integers. Before discussing this method further some definitions are required.

An assignment is a set of k pairings, $k \leq n$, of individuals to jobs. A transfer is a shifting of individuals paired to jobs so as to increase the number of pairings. An individual i is qualified to do job j if $u_i + v_j = r_{ij}$. Note each individual can do each job, but each individual is not qualified to do each job.

The following notation will also be used. Let Q denote the qualified matrix, with entries $q_{ij} = 1$ if individual i is qualified for job j; and 0 otherwise.

A $\underline{1}^*$ in (i,j) in Q implies that the i^{th} individual has been assigned the j^{th} job.

Kuhn's method of solving the assignment problem has three major components, first, the initialization stage which initializes the variables and gives an initial assignment; second, Routine I, which is used to attempt transfers to improve a given assignment; and third, Routine II, used to decrease u's and v's to increase the number of individuals qualified for jobs.

INITIALIZATION

In the initialization stage Kuhn used the following method for determining the initial u_i's and v_j's.

Let a_i be the maximum of $\{ r_{ij} \mid 1 \leq j \leq n \}$ and let $a = \sum_{i=1}^{n} a_i$. Let b_j be the maximum of $\{ r_{ij} \mid 1 \leq i \leq n \}$ and let $b = \sum_{j=1}^{n} b_j$.

Since we want to minimize $\sum_{i=1}^{n} u_i + v_j$, we will use the minimum of a and b.

If $a \leq b$ let $u_i = a_i$ $i = 1, 2, 3, \ldots, n$ and

$v_j = 0$ $j = 1, 2, 3, \ldots, n$.

If $a > b$ let $u_i = 0$ $i = 1, 2, 3, \ldots, n$ and

$v_j = b_j$ $j = 1, 2, 3, \ldots, n$.

Clearly either choice for the initial assignment of u_i's and v_j's is a feasible solution to the dual problem and preference is given to the one producing the smaller initial sum.

Once the u_i's and v_j's have been initialized the qualified matrix Q must be constructed. Recall the entries, q_{ij}, of Q are 1 if $u_i + v_j = r_{ij}$ and 0 otherwise. Once Q has been constructed, an assignment must be made. Kuhn's method for creating an initial assignment is as follows: Proceed down column j for $j = 1, 2, 3, \ldots, n$. The first 1 located in column j that does not have a 1^* in its row becomes a 1^*. Proceed to the next column until all columns have been checked. After this is completed, there should be no row or column with more than one 1^* in it. The initial assignment is done; Q is ready to be sent to Routine I. An example of the initialization of the u's and v's along with the construction of Q follows.

Given the rating matrix $R = \begin{bmatrix} 9 & 9 & 8 & 9 & 7 \\ 1 & 5 & 3 & 6 & 6 \\ 3 & 4 & 4 & 3 & 2 \\ 4 & 4 & 4 & 5 & 3 \\ 3 & 3 & 2 & 8 & 7 \end{bmatrix}$ recall that a_i is the maximum entry in row i, $\sum_{i=1}^{n} a_i = a$, b_j is the maximum entry in column j, and $\sum_{j=1}^{n} b_j = b$. Since in our example, a = 32 and b = 42, the following initial assignment is made. Let $u_1 = 9$, $u_2 = 6$, $u_3 = 4$, $u_4 = 5$, $u_5 = 8$; and $v_j = 0$ for j = 1, 2, 3, . . . n. Given these u's and v's, the following Q is produced.

$$Q = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Next an initial assignment is needed. Recall Kuhn's method for an initial assignment is to proceed down each column. If a 1 is found without a 1^* in its row, then the 1 becomes a 1^*. Using this method the initialization of Q produces the following.

$$Q = \begin{bmatrix} 1^* & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1^* & 1 \\ 0 & 1^* & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Now Q has been constructed and an initial assignment has been made. Q is ready for Routine I.

ROUTINE I

Routine I is used to determine if it is possible to make a transfer in order to increase the number of pairings in the given assignment. Given the Q matrix, Routine I has two possible outcomes:

IA - A transfer has been made, thus improving the number of pairings in the assignment. The altered Q is sent to Routine I to determine if any other transfers are possible.

IB - No transfers were made in the Q entered in Routine I. Q is now ready to be sent to Routine II.

An equivalent definition for a transfer is needed before Routine I can be discussed. A transfer is a shifting of 1's (people qualified for jobs) and 1^*'s (people assigned to jobs) to increase the number of 1^*'s.

A definition for an essential row is also needed before discussing the algorithm for Routine I. A row that has a 1^* that is needed for a transfer, but can not be used, is an essential row. This row is termed essential because the individual assigned to this job can not be transferred to a different job.

Once entering Routine I, a check is made to determine if each column has a 1^*. If a column is found that does not have a 1^* in it, then a search for a 1 in that column is made, so a transfer can be attempted. If a 1 is not found a transfer can not be attempted and that column is passed over.

Recall that no column or row can have more than one 1^*. So if a 1 is found in the column, say in row k, before it can be shifted to a 1^* a check of row k is done to determine if there is already a 1^* in the k^{th} row. If not, the shift of that 1 to a 1^* is done. A transfer has been completed and outcome IA has occurred. Now that Q has been altered, Routine I must be performed again to check for any other transfers.

If a 1^* is found in row k, then a search is done to determine if that 1^* can be shifted to a 1 in its column. If so, a transfer is made and outcome IA has occurred again. If the 1^* can not be shifted to a 1 in its column and no transfer is made, then the row in which the 1^* is in becomes essential and the algorithm proceeds to the next column.

Once Routine I has passed through all the columns of Q, with no transfers being made, Q is ready to be sent to Routine II. An example of Routine I follows.

Recall the Q from the previous section,

$$Q = \begin{bmatrix} 1^* & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1^* & 1 \\ 0 & 1^* & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Given this Q a search is made for 1^* in each column. Proceeding down each column 1^*'s are found in columns one and two. Column three does not have a 1^* so a search for a 1 in column three is done in order to attempt a transfer. A 1 is found in q_{33}. Since no row or column can have more than one 1^*, a search is made of row three to determine if a 1^* is already in that row. A 1^* is found in q_{32}. Next a search of column two is made for a 1 to transfer the 1^* to, from the q_{32} position. A 1 is found in q_{12}, but once again, a search for a 1^* in that row must be done and a 1^* is found in q_{11}. A search for a 1 in column one is now done to attempt to reassign the 1^* in q_{11}. A 1 is not found so no transfer can be made and row one is designated essential. Since the first 1 found in column two could not be assigned the 1^*, a search of column two is made attempting to locate another 1. No other 1 is found, no transfer is made and row three is labeled essential. The 1 we were attempting to convert to a 1^* was in q_{33}, so column three is searched for a different 1 and none is found, hence we proceed to the next column. Proceeding down column four a 1^* is found; go to next column. Proceeding down column

five, no 1^* is found, the search for a 1 is done. A 1 is found in q_{25}. Row two must then be searched for a 1^* and a 1^* is found in q_{24}. Then a 1 in column four is needed to attempt to transfer the 1^* from q_{24}. A 1 is found in q_{14}, but the same result that occurred when the first transfer was attempted will also occur here. That is, the 1^* in q_{11} can not be reassigned so no transfer is done. If row one had not already been labeled essential it would now be labeled as such.

Column four is now searched for another 1 to attempt to reassign the 1^* that is in q_{24}. A 1 is found in q_{44}. A search is made of row four for a 1^*. Since one is not found, the 1 in q_{44} is assigned the 1^*. The 1^* in q_{24} becomes a 1 and the 1 in q_{25} becomes a 1^*. The transfer has now been completed. Once this has been done all previously essential rows become inessential and the altered Q is sent to Routine I again. The altered Q is as follows.

$$Q = \begin{bmatrix} 1^* & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1^* \\ 0 & 1^* & 1 & 0 & 0 \\ 0 & 0 & 0 & 1^* & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

When entering Routine I a transfer will again be attempted using the 1 in q_{33}. When the transfer still can not be completed, rows one and three will once again be designated as essential. All the other columns of Q have 1^*'s in them so no other transfers are attempted. Since no transfers were made Q is now ready for Routine II.

ROUTINE II

Upon entering Routine II all possible transfers have been made. The following definitions will be useful in discussing Routine II.

An essential column is a column with a 1^* in an inessential row. Since all essential rows also have 1^* in them the number of essential rows and columns equals the number of 1^*'s in Q.

In this method of solving the General Assignment Problem, the only acceptable pairings of individuals to jobs are those of individuals qualified to jobs, that is $u_i + v_j = r_{ij}$. An optimal solution is an assignment of n pairings (n 1^*'s in Q). Note an assignment is a set of k pairings such that $k \leq n$, assumptions A1 - A3 hold, and all r_{ij} are positive integers. Recall that in solving the dual we want to minimize $\sum_{i=1}^{n} u_i + v_j$ which we will refer to as the budget.

Routine II is used to determine whether an optimal solution has been found. If an optimal solution has not been found, a reduction in the budget will be made.

Before the algorithm for Routine II can be discussed, the relationship between the number of essential rows, ER, inessential rows, IER, essential columns, EC, and inessential columns, IEC, must be understood. Note that EC + IEC = n and ER + IER = n. As stated before ER + EC = the number of 1^* in Q. Hence, ER + EC $\leq$ n. The only time ER + EC = n is when ER = 0 and EC = n. The reason for this is that if ER + EC = n, then there are n 1^*'s in Q. This would mean that every column had a 1^* in it, so it is impossible that a transfer is attempted, but not completed. Therefore ER would be zero.

The next relationship that is needed is between ER and IEC. By the trichotomy principle there are three cases.

CASE I — IEC < ER, adding EC to both sides of this equation leads to the following:

IEC + EC < ER + EC. Since IEC + EC = n and ER + EC $\leq$ n, we have this equation: n < ER + EC $\leq$ n.
This is a contradiction.

CASE II — IEC = ER, adding EC to both sides of this equation then leads to IEC + EC = ER + EC. Which leads to n = ER + EC $\leq$ n. This implies that ER + EC = n, therefore ER = 0 and EC = n. This is the trivial case.

CASE III — IEC > ER, which follows directly.

The last relationship that needs to be understood is between EC and IER. Since IEC = n - EC and ER = n - IER, then the above yields the following relationship between EC and IER. Either EC = IER which implies ER = 0, EC = n or EC < IER. With these relationships clarified, the algorithm for Routine II is discussed.

Upon entering Routine II the essential rows have been recorded, and next the essential columns must be recorded. Once this is done, if there are no inessential columns, then there are n essential columns and the assignment of 1^* is an optimal solution. Thus the General Assignment Problem and its dual have been solved.

If there are inessential columns, then a reduction in the budget can be made. To determine the amount of the reduction, first compute d, the minimum of $u_i + v_j - r_{ij}$ taken over all inessential i's and j's. Once d has been computed there are two mutually exclusive cases.

CASE I — $u_i > 0$ for all inessential rows. If this occurs, compute m, the minimum of d and u_i taken over all inessential i. Then the reduction in the budget is done as follows:

Replace u_i by $u_i - m$ for all inessential i.
Replace v_j by $v_j + m$ for all essential j.

Recall that EC < IER so this will always produce a budget reduction.

CASE II $u_i = 0$ for some inessential row. If this occurs, let m = d. Then the reduction in the budget is done as follows:

Replace u_i by $u_i + m$ for all essential rows i.
Replace v_j by $v_j - m$ for all inessential columns j.

Note that in Case II d will always be less than or equal to the minimum of all v_j taken over all inessential j.

Proof: Let v be the minimum of all v_j taken over all inessential j. Let d be the minimum $u_i + v_j - r_{ij}$ taken over all inessential i and j.

In Case II recall that $u_i = 0$ for some inessential i, therefore $u_i + v_j - r_{ij}$ will be the smallest when $u_i = 0$ and $v_j = v$. So $0 + v - r_{ij} = d \leq v$ for all inessential i, j.

Recall that IEC > ER, so this too will always produce a budget reduction.

Two questions must be addressed at this point. Does the condition $u_i + v_j \geq r_{ij}$ still hold after a budget reduction? Has the number of 1's in Q increased, thus providing more qualified individuals?

Consider the first question. When entering Routine II there are two possible cases: $u_i > 0$ for all inessential rows i or $u_i = 0$ for some inessential row i. Under either case there are two alternatives, m = d, or m = u.

It must be shown that with any combination of these that $u_i + v_j \geq r_{ij}$ still holds. Note that in both cases, m = minimum (d, u) where d = minimum of $u_i + v_j - r_{ij}$ taken over all inessential i, j and u = minimum u_i taken over all inessential i.

Let v_j* and u_i* denote the values of v_j and u_i after a budget reduction.

CASE 1 $u_i > 0$ for all inessential rows i.

Budget reduction: $u_i* = u_i - m$ for all inessential rows i,

$u_i* = u_i$ for all essential rows i,

$v_j^* = v_j + m$ for all essential columns j,

$v_j^* = v_j$ for all inessential colulmns j.

A. Let m = d. Then the budget reduction causes the following changes.

i. If u_i is in an essential row, and v_j is in an essential column, then

$u_i^* + v_j^* = u_i + v_j + m \geq r_{ij}$.

ii. If u_i is in an essential row, and v_j is in an inessential column, then

$u_i^* + v_j^* = u_i + v_j \geq r_{ij}$.

iii. If u_i is in an inessential row, and v_j is in an essential column, then

$u_i^* + v_j^* = u_i - m + v_j + m = u_i + v_j \geq r_{ij}$.

iv. If u_i is in an inessential row, and v_j is in an inessential column, then

$u_i^* + v_j^* = u_i - m + v_j \geq r_{ij}$ since m = d is the minimum of

$u_i + v_j - r_{ij}$ taken over all inessential i,j.

B. Let m = u. Then the budget reduction causes the following changes.

i. If u_i is in an essential row, and v_j is in an essential column, then

$u_i^* + v_j^* = u_i + v_j + m \geq r_{ij}$.

ii. If u_i is in an essential row, and v_j is in an inessential column, then

$u_i^* + v_j^* = u_i + v_j \geq r_{ij}$.

iii. If u_i is in an inessential row, and v_j is in an essential column, then

$u_i^* + v_j^* = u_i + m + v_j - m = u_i + v_j \geq r_{ij}$.

iv. If u_i is in an inessential row, and v_j is in an inessential column, then

$u_i^* + v_j^* = u_i - m + v_j \geq r_{ij}$ since $m = u \leq d$ = minimum of

$u_i + v_j - r_{ij}$ over all inessential i,j.

Therefore if $u_i > 0$ for all inessential i, then after the budget reduction is made, the condition that $u_i + v_j \geq r_{ij}$ still holds.

CASE 2 $u_i = 0$ for some inessential row i.

Budget reduction: $u_{i*} = u_i + m$ for all essential rows i,

$u_{i*} = u_i$ for all inessential rows i,

$v_{j*} = v_j - m$ for all inessential columns j,

$v_{j*} = v_j$ for all essential columns j.

Recall that in Case II, $u_i = 0$ for some inessential row i, that m = d. Then the budget reduction causes the following changes

i. If u_i is in an essential row, and v_j is in an essential column, then

$$u_{i*} + v_{j*} = u_i + m + v_j \geq r_{ij}.$$

ii. If u_i is in an essential row, and v_j is in an inessential column, then

$$u_{i*} + v_{j*} = u_i + m + v_j - m = u_i + v_j \geq r_{ij}.$$

iii. If u_i is in an inessential row, and v_j is in an essential column, then

$$u_{i*} + v_{j*} = u_i + v_j \geq r_{ij}.$$

iv. If u_i is in an inessential row, and v_j is in an inessential column, then

$u_{i*} + v_{j*} = u_i + v_j - m \geq r_{ij}$ since m = d is minimum of

$u_i + v_j - r_{ij}$ taken over inessential i,j.

Next, consider the second question: Will the number of 1's in Q increase after a budget reduction?

Since d is the minimum of $u_i + v_j - r_{ij}$, if m = d then at least one new 1 will

definitely be introduced. If m = u then a new 1 may not be introduced after the first budget reduction. But when Routine II is applied again, then $u_i = 0$ for some inessential i, so Case II will occur, m will equal d, thus a new 1 will be introduced. So even if we go through Routine II and a new 1 is not introduced, then when we go through it the next time a new 1 will definitely be introduced.

Once Q has been altered, the new Q is sent to Routine I again. An example of Routine II follows.

Recall the Q used in the previous section. Once Q has been sent to Routine I twice, and it has been determined that no further transfers are possible, the algorithm proceeds to Routine II. At this point recall Q has the following form:

$$Q = \begin{array}{c} \begin{matrix} & & & E & E \end{matrix} \\ \begin{bmatrix} 1^* & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1^* \\ 0 & 1^* & 1 & 0 & 0 \\ 0 & 0 & 0 & 1^* & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} E \\ \\ E \\ \\ \\ \end{matrix}$$

Note that rows one and three are essential. Then by definition, columns four and five are essential. A search needs to be done to determine if the given assignment is an optimal solution. Since there are inessential columns, the assignment is not an optimal solution, and a budget reduction must be done.

In order to do a budget reduction, R must be used, so we return to the ratings matrix R.

Recall the matrix R used in the first section.

$$R = \begin{bmatrix} 9 & 9 & 8 & 9 & 7 \\ 1 & 5 & 3 & 6 & 6 \\ 3 & 4 & 4 & 3 & 2 \\ 4 & 4 & 4 & 5 & 3 \\ 3 & 3 & 2 & 8 & 7 \end{bmatrix}$$

Compute d, the minimum of $u_i + v_j - r_{ij}$ taken over inessential i, j. The rows and columns that are essential in Q are also considered essential in R.

Recall $u_1 = 9$, $u_2 = 6$, $u_3 = 4$, $u_4 = 5$, $u_5 = 8$, and all the v_j's = 0. Since rows 2, 4, and 5 are inessential and columns 1, 2, and 3 are inessential we have:

$u_2 + v_1 - r_{21} = 6 + 0 - 1 = 5,$

$u_2 + v_2 - r_{22} = 6 + 0 - 5 = 1,$

$u_2 + v_3 - r_{23} = 6 + 0 - 3 = 3,$

$u_4 + v_1 - r_{41} = 5 + 0 - 4 = 1,$

$u_4 + v_2 - r_{42} = 5 + 0 - 4 = 1,$

$u_4 + v_3 - r_{43} = 5 + 0 - 4 = 1,$

$u_5 + v_1 - r_{51} = 8 + 0 - 3 = 5,$

$u_5 + v_2 - r_{52} = 8 + 0 - 3 = 5,$

$u_5 + v_3 - r_{53} = 8 + 0 - 2 = 6.$

Given these values, d = 1. Once d has been determined, it can be seen that we have CASE I, $u_i > 0$ for all inessential rows i. Thus, m, the minimum of d and u_i (inessential i) is determined. In our case, m = d = 1. Given m = 1 and Case I, the reduction in the budget is as follows.

$u_1 = 9$, $u_2 = 6 - 1 = 5$, $u_3 = 4$, $u_4 = 5 - 1 = 4$, $u_5 = 8 - 1 = 7$ and

$v_1 = 0$, $v_2 = 0$, $v_3 = 0$, $v_4 = 0 + 1 = 1$, $v_5 = 0 + 1 = 1$

With the new values for u's and v's, Q is now altered in the following way:

$$Q = \begin{bmatrix} 1^* & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1^* \\ 0 & 1^* & 1 & 0 & 0 \\ 1 & 1 & 1 & 1^* & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Note that when the altered Q is sent to Routine I again, column three will be searched for a 1^*, but a 1^* will not be found. This will cause a transfer to be attempted with the 1^* in q_{32}, the 1 in q_{12}, the 1^* in q_{11}, the 1 in q_{41}, the 1^* in q_{44}, the 1 in q_{24}, and the 1^* in q_{25}. This will not be possible. Thus row one will be labeled essential. Next colulmn four will be searched for a different 1 with which to attempt a transfer. A transfer will then be attempted with the 1^* in q_{32}, the 1 in q_{12}, the 1^* in q_{11}, the 1 in q_{41}, the 1^* in q_{44}, and the 1 in q_{54}. This time the transfer is possible and the altered Q is as follows.

$$Q = \begin{bmatrix} 1 & 1^* & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1^* \\ 0 & 1 & 1^* & 0 & 0 \\ 1^* & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1^* & 0 \end{bmatrix}$$

Once Q has been altered in this way, and sent back to Routine II it will be determined that an optimal assignment can be found. This will be due to having a 1^* in each column, so an optimal assignment of 1^*'s has been found.

CHAPTER 3

Hungarian Method

STATEMENT OF THE PROBLEM

The second method of solving the General Assignment Problem is the Hungarian Method. Recall assumptions A1 - A3 made in Chapter I. For the Hungarian Method, we also assume that the problem is a minimization problem, but given a maximization problem it can easily be converted to a minimization problem by multiplying all entries in the rating matrix R by -1. Before continuing with the discussion of the Hungarian Method, some notation and definitions are needed.

Since the Hungarian method is one of minimization, the matrix which is used to determine an optimal assignment is referred to as C, the cost matrix. The entry c_{ij} in C represents the cost of individual i doing job j.

As before, an assignment is a set of n entries with no two entries from the same row or column. The sum of the entries of an assignment is referred to as the cost of the assignment.

The last definition needed is that of an optimal assignment, which is the assignment with the smallest possible cost.

The basis for the Hungarian method is the following theorem.

Theorem - If a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.

Proof:

Let $\sum_{i,j=1}^{n} e_{ij_i}$ where $(j_1, j_2, j_3, \ldots j_n)$ is a permutation of $(1, 2, 3, \ldots n)$

be the optimal assignment.

Let $\sum_{i,j=1}^{n} d_{ik_i}$ where $(j_1, j_2, j_3, \ldots j_n)$ is a permutation of $(1, 2, 3, \ldots n)$

be any other assignment.

Note $\sum_{i,j=1}^{n} e_{ij_i} \leq \sum_{i,j=1}^{n} d_{ik_i}$.

Let s be a constant added to some row t. There is only one entry modified in any given assignment, since an assignment can not have more than one entry from any row or column. Therefore

$$\left(\sum_{i,j=1}^{n} e_{ij_i}\right) + s \leq \left(\sum_{i,j=1}^{n} d_{ik_i}\right) + s$$

Thus, the optimal assignment is preserved.

Given this theorem, the Hungarian method attempts to go through a process that will reduce the entries in the matrix and eventually produce a zero in each row and column so that an assignment can be chosen from the given zeros. Thus we are seeking an assignment with zero cost.

OUTLINE OF THE ALGORITHM

The procedure is as follows:

Step 1 - Subtract the smallest entry in each row from all the entries in that row.

Step 2 - Subtract the smallest entry in each column from all the entries in that column.

Step 3 - Draw the minimum number of lines through appropriate rows and columns so all zero entries of the cost matrix are covered.

Step 4 - If the minimum number of lines is equal to n, then an optimal assignment of zeros is possible and we are finished. If not proceed to Step 5.

Step 5 - Determine the smallest entry not covered by any line, subtract that entry from all uncovered entries. Then add it to all entries covered by a both a horizontal and a vertical line. Return to Step 3.

Step 1 introduces at least one zero entry in each row.

Step 2 ensures that each column has a zero entry. Note that because of Step 1, some columns already have zeros.

Step 3 is done using a Maximum Flow Algorithm from Network Theory [1, p. 571]

Step 4 is often referred to as the optimality test. To verify that it is a test for optimality it must be shown that it takes a minimum of n lines to cover an optimal assignment. If it takes a minimum of n lines to cover all zeros then this implies that there are n zeros with no two in the same column or row. Since an optimal assignment has n zeros, no two coming from the same column or row, then an optimal assignment exists. [4, p. 623]

Step 5 is equivalent to finding the smallest uncovered entry, say c_o, and subtracting c_o from all entries in the matrix. Next add c_o to all entries in the rows and columns covered by a line. Any entry covered by only one line will have no change, but any entry covered by two lines will increase by c_o so that an adjustment needs to be made. The reason for adding back c_o to covered rows and columns is that these rows and columns have zeros in them which would become negative values when c_o is subtracted from them. Since the Hungarian method desires an optimal assignment of zeros, negative values are avoided. Note by Theorem 1, this adding and subtracting of c_o will not change the optimal assignment in the matrix.

Using these steps until step 4 yields n lines will give an optimal assignment.

The following proof shows that an optimal assignment will be attained after a

OUTLINE OF THE ALGORITHM

The procedure is as follows:

Step 1 - Subtract the smallest entry in each row from all the entries in that row.

Step 2 - Subtract the smallest entry in each column from all the entries in that column.

Step 3 - Draw the minimum number of lines through appropriate rows and columns so all zero entries of the cost matrix are covered.

Step 4 - If the minimum number of lines is equal to n, then an optimal assignment of zeros is possible and we are finished. If not proceed to Step 5.

Step 5 - Determine the smallest entry not covered by any line, subtract that entry from all uncovered entries. Then add it to all entries covered by a both a horizontal and a vertical line. Return to Step 3.

Step 1 introduces at least one zero entry in each row.

Step 2 ensures that each column has a zero entry. Note that because of Step 1, some columns already have zeros.

Step 3 is done using a Maximum Flow Algorithm from Network Theory [1, p. 571]

Step 4 is often referred to as the optimality test. To verify that it is a test for optimality it must be shown that it takes a minimum of n lines to cover an optimal assignment. If it takes a minimum of n lines to cover all zeros then this implies that there are n zeros with no two in the same column or row. Since an optimal assignment has n zeros, no two coming from the same column or row, then an optimal assignment exists. [4, p. 623]

Step 5 is equivalent to finding the smallest uncovered entry, say c_o, and subtracting c_o from all entries in the matrix. Next add c_o to all entries in the rows and columns covered by a line. Any entry covered by only one line will have no change, but any entry covered by two lines will increase by c_o so that an adjustment needs to be made. The reason for adding back c_o to covered rows and columns is that these rows and columns have zeros in them which would become negative values when c_o is subtracted from them. Since the Hungarian method desires an optimal assignment of zeros, negative values are avoided. Note by Theorem 1, this adding and subtracting of c_o will not change the optimal assignment in the matrix.

Using these steps until step 4 yields n lines will give an optimal assignment.

The following proof shows that an optimal assignment will be attained after a

finite number of iterations of Step 1 through Step 5.

To show finiteness we need to show that the reduced costs are always positive.

Let $S_r = \{i_1, i_2, \ldots i_p\}$ a set of uncovered rows in C.

Let $\bar{S}_r$ = set of covered rows in C. Note that $\bar{S}_r$ has n-p elements.

Let $S_c = \{j_1, j_2, \ldots j_q\}$ set of uncovered columns in C.

Let $\bar{S}_c$ = set of covered columns in C. Note that $\bar{S}_c$ has n - q elements.

So p = number of elements in S_r and q = number of elements in S_c.

Let $\hat{c}_{ij}$ = the entries in C after Steps 1 and 2.

Let $\bar{c}_{ij}$ = the entries in C after a reduction from Step 5 has been made.

Let n = number of rows or columns in C.

Let c_o = smallest uncovered entry found in Step 5.

Let k = maximum number of zeros in an assignment = minimum number of lines needed to cover all zeros in C. [4, 623]

Proof: $\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{c}_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\bar{c}_{ij}$ = *reduced cost*

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{c}_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\bar{c}_{ij} =$$

$$\sum_{S_r x S_c} c_o + \sum_{S_r x \bar{S}_c} 0 + \sum_{\bar{S}_r x S_c} 0 + \sum_{\bar{S}_r x \bar{S}_c} -c_o$$

$= pqc_o - (n - p)(n - q)c_o = n(p + q - n)c_o$, but p + q = number of uncovered rows and columns = 2n - k. So $n(p + q - n)c_o = n(2n - k - n)c_o = n(n - k)c_o$. Since $c_o > 0$, n > 0 and n > k then $n(n - k)c_o > 0$. Therefore, the sum of all costs is being reduced by a positive integer each time step 5 is performed, so the process in the Hungarian method is finite.

EXAMPLE OF HUNGARIAN METHOD

Recall the rating matrix R used in Kuhn's method.

$$R = \begin{bmatrix} 9 & 9 & 8 & 9 & 7 \\ 1 & 5 & 3 & 6 & 6 \\ 3 & 4 & 4 & 3 & 2 \\ 4 & 4 & 4 & 5 & 3 \\ 3 & 3 & 2 & 8 & 7 \end{bmatrix}$$

Since our R is a rating matrix and we want to maximize our rating, we need to multiply the matrix by -1. We will refer to the matrix -R as C, the cost matrix.

$$C = \begin{bmatrix} -9 & -9 & -8 & -9 & -7 \\ -1 & -5 & -3 & -6 & -6 \\ -3 & -4 & -4 & -3 & -2 \\ -4 & -4 & -4 & -5 & -3 \\ -3 & -3 & -2 & -8 & -7 \end{bmatrix}$$

Applying Step 1, the smallest entry in each row is subtracted from all entries in its row, producing the following matrix:

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 5 & 1 & 3 & 0 & 0 \\ 1 & 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 & 2 \\ 5 & 5 & 6 & 0 & 1 \end{bmatrix}$$

Since the smallest entry in each column is zero, Step 2 makes no changes.

Step 3 leads to covering C in the following manner.

$$C = \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{2} \\ \cancel{5} & \cancel{1} & \cancel{3} & \cancel{0} & \cancel{0} \\ \cancel{1} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{2} \\ 1 & 1 & 1 & \cancel{0} & 2 \\ 5 & 5 & 6 & \cancel{0} & 1 \end{bmatrix}$$

Therefore the minimum number of lines needed is 4 which is less than n = 5. So by Step 4, an optimal assignment can not yet be obtained. Hence, we apply Step 5. In applying Step 5, the smallest uncovered entry, 1, is subtracted from all uncovered entries and added to all entries covered twice leading to the following C.

$$C = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 \\ 5 & 1 & 3 & 1 & 0 \\ 1 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 4 & 4 & 5 & 0 & 0 \end{bmatrix}$$

After Step 5, we return to Step 3. This time C is covered in this way.

$$C = \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{2} \\ \cancel{5} & \cancel{1} & \cancel{3} & \cancel{1} & \cancel{0} \\ \cancel{1} & \cancel{0} & \cancel{0} & \cancel{2} & \cancel{2} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} \\ \cancel{4} & \cancel{4} & \cancel{5} & \cancel{0} & \cancel{0} \end{bmatrix}$$

Step 4's test for optimality holds since the minimum number of lines needed was 5, which is equal to n. Therefore, an optimal assignment is found, and we are done.

CHAPTER 4

Determination of Optimal Assignments.

After using either Kuhn's method, or the Hungarian Algorithm to determine that an optimal solution can be found, the solution needs to be determined. Note that it only has been determined that an optimal solution can be found, it doesn't determine what that optimal solution is.

If using Kuhn's method, an optimal solution has been found if there are n essential rows, which means there were n 1^*'s in the Q matrix. Once it has been determined that an optimal solution can be found, the actual assignment is the assignment of individual i to job j where there is a 1^* in q_{ij}.

Recall the Q matrix in Kuhn's algorithm after it had been determined that an optimal assignment was possible.

$$Q = \begin{bmatrix} 1 & 1^* & 0 & 0 & 0 \\ 0 & 1 & 0 & 1^* & 1 \\ 0 & 1 & 1^* & 0 & 0 \\ 1^* & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1^* \end{bmatrix}$$

Then the assignment, with the highest possible rating is q_{12}, q_{24}, q_{33}, q_{41}, q_{55}. Recall the rating matrix R in Kuhn's algorithm.

$$R = \begin{bmatrix} 9 & 9 & 8 & 9 & 7 \\ 1 & 5 & 3 & 6 & 6 \\ 3 & 4 & 4 & 3 & 2 \\ 4 & 4 & 4 & 5 & 3 \\ 3 & 3 & 2 & 8 & 7 \end{bmatrix}$$

The rating associated with q_{12} is 9, q_{24} is 6, q_{33} is 4, q_{41} is 5 and q_{55} is 7. Then the total rating for the optimal assignment is 31.

When using the Hungarian method, an optimal assignment of zeros can be found when the minimum number of lines needed to cover all zeros is equal to n. Recall that to find the minimum number of lines a maximum flow algorithm is used. Once that algorithm is done, the path of the flow determines which zeros to choose.

Both methods produce only one optimal assignment, but other optimal assignments may exist. One way to find other optimal assignments is to rearrange the order of the columns so that in Kuhn's method the initial assignment differs. Also this will give a different maximum flow result in the Hungarian method. Note, however, that the sum of the r_{ij} will have the same value for all optimal assignments for a given matrix R. Thus it will not improve the solution, but just produce an equivalent one. Also, note, that there may be only one optimal solution. So rearranging columns may not produce a different solution.

CHAPTER 5

Summary and Conclusion

While both the Hungarian method and Kuhn's algorithm are effective techniques for solving the assignment problem, neither is trivial.

The Hungarian method requires finding the minimum number of lines needed to cover all zero entries. If the matrix is small this may be an easy task, but relatively speaking in any n x n matrix with $n > 5$, the task becomes difficult. In Kuhn's method the task of checking the 1's and 1*'s needs to be performed efficiently. Much has been done in the computer programming field to produce programs to alleviate the complexity of each of these methods. [8, p.793]. Others have produced methods for speeding the process in which an optimal solution can be found. [9, p. 194] The use of computer programs and acceleration methods can greatly improve both the efficiency and the effectiveness of the two methods.

Both of the methods require certain assumptions made. Of the assumptions A1 - A3, the only one that can be modified is A1, the number of jobs must equal the number of individuals. If this assumption doesn't hold, for example, n jobs and n + 1 individuals, a dummy job can be introduced into the matrix with a rating of 1 in Kuhn's method and a cost larger than all other costs in the Hungarian method. then whichever individual was assigned that job would not be assigned to any job.

The information on this topic is vast. The assignment problem has applications in many other areas. In most introductory texts on linear programming the assignment problem is addressed as a specific case of either the transportation problem, [6 p.490] or the traveling salesman problem. [7, p.259] Usually when it is discussed, it is solved using the Hungarian method instead of Kuhn's method. In the context of the traveling salesman

problem, usually found in introductory network theory texts, the covering of all zeros is discussed as a maximum flow problem [7, p.323]. However, when the assignment problem is presented in the context of the transportation problem the covering of zeros is not thoroughly investigated. [6, p.491] The transportation problem is consistently found in basic linear programming texts and the traveling salesman problem is usually found in network Theory texts. In either case, the covering of zeros is never presumed to be a trivial matter.

The assignment problem also is also treated in textbooks on operations research eg. [7, p.112] and in systems analysis eg. [7, p.114]. In order to study this subject further and determine a more efficient algorithm for the assignment problem, further research in the areas of network and graph theory would be essential.

Bibliography

1. Bazaraa, Jarvis, and Sherali, *Linear Programming and Network Flows*, New York, Wiley and Sons, 1990.

2. H.W. Kuhn, *The Hungarian Method for the Assignment Problem*, Naval Reasearch Logistics Quarterly 2, March-June(1956)712-726.

3. Rorres, Anton, *Applications of Linear Algebra*, Wiley and Sons, 1984.

4. G. Strang, *Introduction to Applied Mathematics* , Wellesly and Cambridge, 1986.

5. C. van de Panne, *Linear programming and Related Techniques*, North-Holland,1971.

6. E. Kaplan, *Mathematical Programming and Games*, New York, Wiley, 1982.

7. K. Mital, *Optimization Methods in Operations Research and Systems Analysis*, New York, Wiley, 1976

8. Glover, Karney, Klingman, and Napier, *A Computation Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems*, Management Science 20(1974),793-813.

9. V. Srinivasan, G.L. Thompson, *Accelerated Algorithms for Laabeling and Relabeling of Trees, with Applications to Distribution Problems*, Journal of the Assosciation for Computing Machinery, 19(1972), 712-726

I, Laura J. Oliver, hereby submit this thesis/report to Emporia State University as partial fulfillment of the requirements for an advanced degree. I agree that the Library of the university may make it available for use in accordance with its regulations governing materials of this type. I further agree that quoting, photocopying, or other reproduction of this document is allowed for private study, scholarship (including teaching) and research purposes of a nonprofit nature. No copying which involves potential financial gain will be allowed without written permission of the author.

Signature of Author

8-3-92
Date

The Optimal Assignment Problem
Title of Thesis/Research Project

Douglas K. Cooper
Signature of Graduate Office Staff Member

August 10, 1992
Date Received

Distribution: Director, William Allen White Library
Graduate School Office
Author