THE HUNGARIAN METHOD FOR THE ASSIGNMENT PROBLEM[1]

H. W. Kuhn
Bryn Mawr College

Assuming that numerical scores are available for the perform-
ance of each of n persons on each of n jobs, the "assignment problem"
is the quest for an assignment of persons to jobs so that the sum of the
n scores so obtained is as large as possible. It is shown that ideas latent
in the work of two Hungarian mathematicians may be exploited to yield
a new method of solving this problem.

1. INTRODUCTION

Stated informally, the problem of personnel-assignment asks for the best assignment of
a set of persons to a set of jobs, where the possible assignments are ranked by the total scores
or ratings of the workers in the jobs to which they are assigned. Variations of this problem,
both mathematical and non-mathematical, have a long history (see the Bibliography appended).
However, recent interest in the question, when posed in the terms of linear programming,
seems to stem from the independent work of Flood, J. Robinson, Votaw, and Orden. Flood's
work [12], begun in 1949, regards the problem as the most "degenerate" case of the transpor-
tation problem. Robinson regarded it as a relative of the travelling salesman problem; her
work is available only in the form of RAND Corporation memoranda. The problem was dis-
cussed from various points of view in the work of Votaw and Orden (see [9]) presented to the
SCOOP Symposium on Linear Inequalities and Programming, June 14-16, 1951. The compu-
tational advantages to be gained by considering the problem in combination with the dual linear
program have been stressed by Dantzig, von Neumann and others (see [8], [10], and [12]). The
purpose of this paper is to develop a computational method that uses this duality in a particu-
larly effective manner. One interesting aspect of the algorithm is the fact that it is latent in
work of D. König and E. Egerváry that predates the birth of linear programming by more than
15 years (hence the name, the "Hungarian Method").

The theoretical basis of the algorithm is laid in Sections 2 and 3. Section 2 (which is
derived from the proof of König in "Theorie der Graphen" (1936) Chelsea, 1950, pp. 232-233)
treats the problem of assignment when there are but two ratings, 1 and 0, indicating that a
worker is qualified or not. Section 3 (which is derived from the work of Egerváry in [3]) shows
that the general problem of assignment can be reduced to this special case by a procedure that
is computationally trivial.

The algorithm is given an independent (and self-contained) statement in Section 4 and
Section 5 is devoted to a detailed example to illustrate its application.

2. THE SIMPLE ASSIGNMENT PROBLEM

The problem of Simple Assignment is illustrated by the following miniature example:

Four individuals (denoted by i = 1, 2, 3, 4) are available for four jobs (denoted by j = 1,
2, 3, 4). They qualify as follows:

$$\text{Individual}\begin{cases} 1 \\ 2 \\ 3 \\ 4 \end{cases} \text{qualifies for job(s)} \begin{cases} 1, 2, \text{ and } 3 \\ 3 \text{ and } 4 \\ 4 \\ 4 \end{cases}$$

This information can be presented effectively by a <u>qualification matrix</u>

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

in which horizontal rows stand for individuals and vertical columns for jobs; a qualified individual is marked by a 1 and an unqualified individual by a 0. Then the Simple Assignment Problem asks:

What is the largest number of jobs that can be assigned to qualified individuals (with not more than one job assigned to each individual)?

This may be stated abstractly in terms of the matrix Q:

What is the largest number of 1's that can be chosen from Q with no two chosen from the same row or column?

It is clear that we can start an assignment by placing unassigned individuals in any unassigned jobs for which they qualify. Thus, we might assign individuals 1 and 2 to jobs 3 and 4, respectively; this information is entered in the matrix below by asterisks.

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

Since it is impossible to improve this assignment by placing an unassigned individual in an unassigned job for which he qualifies, this assignment is said to be <u>complete</u>. If an assignment is complete, it is natural to attempt an improvement by means of a <u>transfer</u>. For instance, the transfer:

Move individual 1 from job 3 to job 1
"   2   "   "   4 " " 3 ,

results in the following <u>incomplete</u> assignment:

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

Here we may assign either individual 3 or 4 to job 4 to complete the assignment. Either result, say

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

is optimal, since there all qualified pairs involve either individual 1 or job 3 or job 4, and hence four assignments would involve one of these twice. Thus, although there is a transfer

possible in this optimal assignment (move 1 from job 1 to job 2), it leads to a complete assignment. The discussion to follow establishes that this situation holds in general, namely, that one can always construct an optimal assignment by a successsion of transfers followed by additional assignments until this is no longer possible.

Suppose n <u>individuals</u> (i = 1, ..., n) are available for n <u>jobs</u> (j = 1, ..., n) and that a <u>qualification matrix</u> $Q = (q_{ij})$ is given, where $q_{ij} = 1$ if individual i qualifies for job j and $q_{ij} = 0$ otherwise. If an assignment (not necessarily optimal) of certain qualified individuals to jobs is given, then the easiest way to improve it is to assign any unassigned individual to an unassigned job for which he qualifies. If this is possible, the given assignment is said to be <u>incomplete</u>; otherwise, it is <u>complete</u>. If the assignment is complete, then it is reasonable to attempt an improvement by means of a <u>transfer</u>. A transfer changes the assignment of r distinct individuals $i_1, ..., i_r$ employed in jobs $j_1, ..., j_r$. It moves i_1 into an unassigned job j_0 and i_k into job j_{k-1} for k = 2, ..., r. All of the new assignments (i_k to j_{k-1}) are assumed to be qualified for k = 1, ..., r. It is convenient to call the result of leaving all assignments unchanged a transfer also. A useful notation for transfers that change some assignment is

$$\begin{array}{ccccccc} & i_1 & i_2 & \cdots & i_{r-1} & i_r & \\ j_0 & j_1 & j_2 & \cdots & j_{r-1} & j_r & . \end{array}$$

We shall call every (assigned) individual involved in such a transfer an <u>essential individual</u> and every job assigned to an inessential individual an <u>essential job</u>. Thus:

LEMMA 1. For a given assignment, if an individual is assigned to a job, then either the individual or the job is essential, and <u>not both</u>.

COROLLARY 1. For all assignments, the number of individuals assigned to jobs equals the number of essential individuals and jobs.

The motivation of the definition of essentiality is partially explained by the next two lemmas.

LEMMA 2. For a given assignment, if an individual is assigned to a job and qualifies for another, unassigned, job then the individual is essential.

PROOF: The transfer of the individual to the unassigned job establishes him as essential.

LEMMA 3. For a given assignment, if every transfer leaves a job assigned then the job is essential.

PROOF. Assume the job j to be inessential. Then some individual i_k is assigned to it and involved in a transfer that moves $i_1, i_2, ..., i_k$ in order. Symbolically,

$$\begin{array}{ccccccc} & i_1 & i_2 & \cdots & i_{k-1} & i_k & \\ j_0 & j_1 & j_2 & \cdots & j_{k-1} & j & , \end{array}$$

and j is unassigned. This proves the lemma.

These lemmas, in combination, establish the key result:

THEOREM 1. For a given assignment, if every transfer leads to
a complete assignment then, for every individual qualified for a job, either
the individual or the job is essential, and possibly both.

PROOF. Let individual i be qualified for job j. If i is assigned to j then Lemma 1
asserts that one or the other is essential. If i is assigned to another job then j is unassigned
and Lemma 2 asserts that the individual i is essential. If i is unassigned then every transfer
leaves j assigned (otherwise the assignment is incomplete) and Lemma 3 asserts that j is
essential. This proves the theorem.

Starting with any assignment (say, of one individual to a job for which he is qualified),
either every transfer leads to a complete assignment or at least one more individual can be
assigned after some transfer. Since at most n individuals can be assigned, this proves:

THEOREM 2. There is an assignment which is complete after
every possible transfer.

The problem will now be viewed from another, dual, aspect. Consider a possible budget
to account for the value of an individual assigned to a job for which he is qualifed. Such a
budget will allot either one unit or nothing to each individual and to each job. A budget is said
to be adequate if, for every individual qualified for a job, either the individual or the job is
allotted one unit, and possibly both.

THEOREM 3. The total allotment of any adequate budget is not
less than the largest number of jobs that can be assigned to qualified
individuals.

PROOF. If the part of the adequate budget allotted to jobs assigned in an optimal assign-
ment is counted, it is seen to be not less than the number of jobs assigned because these jobs
are all assigned to qualified individuals. Since the total budget is not less than this amount, this
proves the theorem.

Consider any assignment that is complete after every possible transfer (by Theorem 2,
there are such) and consider the budget that allots one unit to each essential individual or job
and zero otherwise. Theorem 1 asserts that this budget is adequate. Taking account of Corol-
lary 1, we have proved:

THEOREM 4. There is an adequate budget and an assignment
such that the total allotment of the budget equals the number of jobs
assigned to qualified individuals.

Since Theorem 3 implies that the assignment of Theorem 4 is optimal, we have provided
the following answer to the Simple Assignment Problem:

The largest number of jobs that can be assigned to qualified individuals is equal to the smallest total allotment of any adequate budget. Any assignment is optimal if and only if it is complete after every possible transfer.

3. THE GENERAL ASSIGNMENT PROBLEM

Suppose n individuals (i = 1, ..., n) are available for n jobs (j = 1, ..., n) and that a rating matrix $R = (r_{ij})$ is given, where the r_{ij} are positive integers, for all i and j. An assignment consists of the choice of one job j_i for each individual i such that no job is assigned to two different men. Thus, all of the jobs are assigned and an assignment is a permutation

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ j_1 & j_2 & \ldots & j_n \end{pmatrix}$$

of the integers 1, 2, ..., n. The General Assignment Problem asks:

For which assignments is the sum

$$r_{1j_1} + r_{2j_2} + \cdots + r_{nj_n}$$

of the ratings largest?

The dual problem considers adequate budgets, that is, allotments of non-negative integral amounts of u_i to each individual and v_j to each job in such a manner that the sum of the allotments to the ith individual and the jth job is not less than his rating in that job. In symbols,

(1) $$u_i + v_j \geq r_{ij} \qquad (i, j = 1, \ldots, n).$$

The problem dual to the General Assignment Problem is then:

What is the smallest total allotment

$$u_1 + \ldots + u_n + v_1 + \ldots + v_n$$

possible for an adequate budget?

The following analogue of Theorem 3 is immediate.

THEOREM 5. The total allotment of any adequate budget is not less than the rating sum of any assignment.

PROOF. Since each individual and job occurs exactly once in an assignment the sum of the allotments to individuals and jobs in an assignment is exactly the total allotment. However, the budget is adequate and therefore this is not less than the sum of the ratings of the individuals in their assigned jobs. In symbols,

$$u_1 + v_{j_1} \geq r_{1j_1}, \ldots, u_n + v_{j_n} \geq r_{nj_n}$$

by the condition that the budget is adequate. Adding these inequalities, we have

$$u_1 + \cdots + u_n + v_{j_1} + \cdots + v_{j_n} \geqq r_{1j_1} + \cdots + r_{nj_n} .$$

However, the integers $j_1, \ldots, j_n$ appearing in the assignment

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ j_1 & j_2 & \ldots & j_n \end{pmatrix}$$

are merely an arrangement of $1, \ldots, n$ and the theorem is proved.

It is an immediate consequence of this theory that, if an adequate budget and an assignment can be exhibited such that the total allotment equals the rating sum, then they must be simultaneously a solution of the assignment problem and its dual. We shall now show that this is always possible and can be achieved by solving certain, related, Simple Assignment Problems.

Associate with each adequate budget for the rating matrix $R = (r_{ij})$ a Simple Assignment Problem by the following rule:

The individual i is qualified for the job j if $u_i + v_j = r_{ij}$; otherwise, he is not qualified.

We see immediately that:

THEOREM 6. If all n individuals can be assigned to jobs for which they are qualified in the Simple Assignment Problem associated with an adequate budget, then the assignment and the budget solve the given General Assignment Problem and the rating sum equals the total allotment.

PROOF. For the given budget and assignment, we have

$$u_1 + v_{j_1} = r_{1j_1} , \ \cdots , \ u_n + v_{j_n} = r_{nj_n} .$$

Adding these equations,

$$u_1 + \cdots + u_n + v_1 + \cdots + v_n = r_{1j_1} + \cdots + r_{nj_n}$$

and this proves the theorem.

If not all individuals can be assigned to jobs for which they are qualified in the Simple Assignment Problem associated with an adequate budget, then the budget can be improved by a simple procedure. Before this procedure can be described, it must be noted that an adequate budget must allot either a positive amount to every individual or a positive amount to every job since otherwise it would not be enough for the positive rating of some individual in some job. We shall assume, without loss of generality since rows and columns enter symmetrically, that every individual is allotted a positive amount; in symbols

$$u_i > 0 \qquad\qquad (i = 1, \ldots, n) .$$

Assume that the largest number of individuals that can be assigned to jobs for which they are qualified is $m < n$. Choose an optimal assignment and let the essential individuals be $i = 1, \ldots, r$

and the essential jobs be $j = 1, \ldots, s$ (possibly renumbering individuals and jobs). Corollary 1 asserts that

$$r + s = m .$$

Then the rule for changing the budget is:

$$u'_1 = u_1, \ldots, u'_r = u_r, u'_{r+1} = u_{r+1} - 1, \ldots, u'_n = u_n - 1$$

$$v'_1 = v_1 + 1, \ldots, v'_s = v'_s + 1, v'_{s+1} = v_{s+1}, \ldots, v'_n = v_n .$$

(The u'_i are still non-negative because the u_i were <u>positive</u> integers.) We must check that
   (a) the new budget is adequate, and
   (b) the total allotment has been decreased.
The adequacy is checked by inequalities (1) which can only fail where u_i has been decreased and v_j has been left unchanged. But this means that both the individual i and the job j are inessential. Theorem 1 then asserts that individual i is not qualified for job j and hence

$$u_i + v_j > r_{ij}$$

by the rule for constructing the associated Simple Assignment Problem. Since all the numbers involved are integers,

$$u'_i + v'_j = (u_i - 1) + v_j = (u_i + v_j) - 1 \geqq r_{ij}$$

and the new budget is adequate.
   The total allotment has been decreased by n - r and increased by s, thus has been decreased by $n - (r + s) = n - m > 0$. Summarizing:

> THEOREM 7. If at most m < n individuals can be assigned to jobs for which they are qualified in the Simple Assignment Problem associated with an adequate budget, then the total allotment of the budget can be decreased by a positive integral amount.

Starting with any adequate budget (say, that which allots to every individual his highest rating and nothing to the jobs), either it is optimal, and Theorem 6 applies, or it can be decreased by Theorem 7. Since it can be improved at most a finite number of times, we have provided the following answer to the General Assignment Problem:

> The largest possible rating sum for any assignment is equal to the smallest total allotment of any adequate budget. It can be found by solving a finite sequence of associated Simple Assignment Problems.

4. THE HUNGARIAN METHOD

   In this section we shall assemble the results of the two preceding sections, abstracted from the context of actual assignments, and state explicitly the algorithm implicit in the

arguments of those sections. In certain cases where it seems advisable to use a different terminology, the discrepancy will be noted parenthetically.

As considered in this paper, the General Assignment Problems asks: Given an n by n matrix $R = (r_{ij})$ of positive integers, find the permutation $j_1, \ldots, j_n$ of the integers $1, \ldots, n$ that maximizes the sum $r_{1j_1} + \ldots + r_{nj_n}$. It is well known (see references [3] and [10] in the Bibliography) that the linear program dual to this problem can be stated: Find non-negative integers $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ subject to

$$(1) \hspace{3cm} u_i + v_j \geq r_{ij} \hspace{2cm} (i, j = 1, \ldots, n)$$

that minimize the sum $u_1 + \ldots + u_n + v_1 + \ldots + v_n$. A set of non-negative integers satisfying (1) will be called a cover (or an adequate budget) and the positions (i, j) in the matrix for which equality holds are said to be marked (or qualified in the associated Simple Assignment Problem); otherwise (i, j) is said to be blank. A set of marks is called independent if no two marks from the set lie in the same line (the term "line" is used here to denote either a row or column). Then a fundamental result of König says: If the largest number of independent marks that can be chosen is m then m lines can be chosen that contain all of the marked positions. (This is precisely the conclusion of Section 1 with "jobs assigned to qualified individuals" playing the rôle of "independent marks.")

The algorithm to be described in this report is based on these remarks in the following manner. If a cover for R is given, a largest set of independent marks is found; if this set contains n marks then obviously the marked (i, j) constitute the desired assignment (Theorem 6). If the set contains less than n marks then a set of less than n lines containing all of the marked (i, j) is used to improve the cover (Theorem 7).

The construction of an initial cover and an initial set of independent marks can be made quite conveniently as follows:

Let $a_i = \max_j r_{ij}$ for $i = 1, \ldots, n$ and $b_j = \max_i r_{ij}$ for $j = 1, \ldots, n$. Further let $a = \Sigma_i a_i$ and $b = \Sigma_j b_j$.

If $a \leq b$ define $\begin{cases} u_i = a_i & \text{for} \quad i = 1, \ldots, n \\ v_j = 0 & \text{for} \quad j = 1, \ldots, n. \end{cases}$

If $a > b$ define $\begin{cases} u_i = 0 & \text{for} \quad i = 1, \ldots, n \\ v_j = b_j & \text{for} \quad j = 1, \ldots, n. \end{cases}$

At this stage, as at all subsequent stages, there is associated with the matrix R and the cover $\{u_i, v_j\}$ a matrix $Q = (q_{ij})$ where

$$q_{ij} = \begin{cases} 1 & \text{if } u_i + v_j = r_{ij} \\ 0 & \text{otherwise.} \end{cases}$$

At each stage we shall also need a set of independent 1's from Q which will be distinguished by asterisks. To provide such a set at the first stage, in the first case ($a \leq b$) the rows are

examined in order and the first 1 in each row without a 1* in its column is changed to a 1*. In the second case (a > b), the same instructions are followed with rows and columns exchanging rôles.

The two basic routines of the algorithm will be called Routine I and Routine II. A schematic description of the order of their repetition is given in Figure 1.
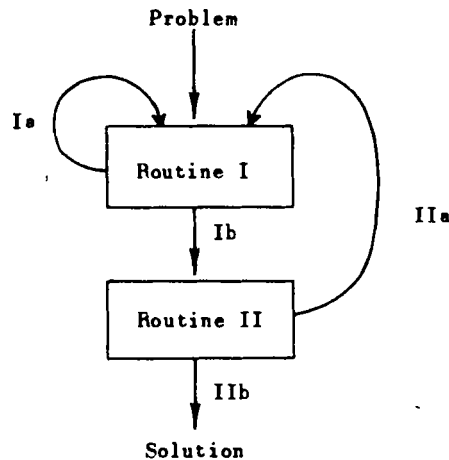

Figure 1

Every occurrence of Ia will increase the number of assignments (i.e., of asterisks in Q) by one and every occurrence of IIa will decrease the current covering sum $(\sum_i u_i + \sum_j v_j)$ by at least one. Since the number of assignments is bounded from above by n and the covering sums are bounded from below by zero, this insures the termination of the combined algorithm.

Routine I

Routine I works with a fixed matrix Q associated with a fixed cover $\{u_i, v_j\}$. The input also includes a certain set of asterisks marking 1's in Q.

The computation begins with the search of each column of Q in turn for a 1*. If a 1* is found, we proceed to the next column (no columns left = Alternative Ib). If a 1* is not found in the column, then the column is called eligible and is searched for a 1. If a 1 is not found, we proceed to the next column (no columns left = Alternative Ib). If a 1 is found in (i_1, j_0), we record i_1 and j_0 and start a process that constructs a sequence of the following form:

$$1 \quad \text{in} \quad (i_1, j_0)$$
$$1^* \quad \text{in} \quad (i_1, j_1)$$
$$1 \quad \text{in} \quad (i_2, j_1)$$
$$\cdots \quad \cdots \quad \cdots .$$

The routine then divides into two cases according to the parity of the number of terms currently in the sequence. In Case 1, we have just found a 1 in (i_k, j_{k-1}) and have recorded i_k and j_{k-1}. We then search the row i_k for a 1*. If a 1* is not found then we change each 1 in the sequence to 1* and each 1* in the sequence (if any) to a 1. This is Alternative Ia and means that we start

Routine I again. In Case 2, we have just found a 1^* in (i_k, j_k). We then search column j_k for a 1. If a 1 is not found, then row i_k is recorded as essential, i_k and j_{k-1} are deleted from the record and we go back to Case 2 with the last two terms of the sequence deleted and searching for a 1 in column j_{k-1} from row $i_k + 1$ on. Note that, if $k = 1$, then we go back to our preliminary search for a 1 in the eligible column j_0 from row $i_1 + 1$ on. Completing Case 2, if a 1 is found in (i_{k+1}, j_k) we test whether i_{k+1} is distinct from $i_1, \ldots, i_k$. If it is distinct then we record i_{k+1} and j_k and are back in Case 1. If it is not distinct, we go on searching for a 1 in column j_k from row $i_{k+1} + 1$ on.

(This routine is connected with Section 2 in the following way. Given an assignment, we enumerate all possible transfers. Such a transfer starts at an eligible column. If there are no eligible columns, there are no transfers and the given assignment is complete. The occurrence of Alternative Ia means that we have found a transfer that frees a column that contains a 1 that is unassigned. In this event, we carry out the transfer:

$$
\begin{array}{cccccc}
i_1 & i_2 & \cdots & i_{k-2} & i_{k-1} & \\
j_0 & j_1 & j_2 & \cdots & j_{k-2} & j_{k-1}
\end{array}
$$
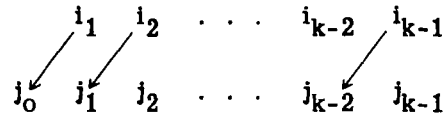
and assign (i_k, j_{k-1}). If a transfer is developed that cannot be continued and which yields a complete assignment, the last row involved is recorded as essential, following which the enumeration of the transfers is continued. If the enumeration of the transfers is completed without the occurrence of Alternative Ia, this is Alternative Ib and we have an assignment in which all transfers yield complete assignments.)

The output of Routine I in Alternative Ib is an optimal assignment for Q and a set of essential rows. Every 1 lies either in an essential row or in the column of a 1^* in an essential row (Theorem 1).

A tentative flow diagram for Routine I is given in Figure 2. For this arrangement of the routine, we use the following notation:

Symbol	Use in Routine
i	Index of rows of Q.
j	Index of columns of Q.
k	$\begin{cases} \text{Tally of length of sequence of 1's and } 1^*\text{'s.} \\ \text{Tally to clear essential rows in Alternative Ia.} \end{cases}$
ℓ	Tally to test distinctness of i_{k+1} from $i_1, \ldots, i_k$.
$i_1, i_2, \ldots, i_n$	Record of rows in sequence of 1's and 1^*'s.
$j_0, j_1, \ldots, j_{n-1}$	Record of columns in sequence of 1's and 1^*'s.
$\epsilon_1, \epsilon_2, \ldots, \epsilon_n$	Record of essential rows.

The values of these quantities for the input of Routine I are:

$$ i = j = k = \ell = 1, \quad i_\nu = \epsilon_\nu = 0 \text{ for } \nu = 1, \ldots, n. $$

The values of these quantities for the output of Alternative Ib are:

$$i = j = k = \ell = 1, \qquad i_\nu = j_{\nu-1} = 0 \ \text{ for } \ \nu = 1, \ldots, n \ .$$

and

$$\epsilon_i = \begin{cases} 1 \\ 0 \end{cases} \ \text{ if row i is } \ \begin{cases} \text{essential} \\ \text{inessential.} \end{cases}$$

The symbol "$A \longrightarrow B$" is to be read "replace the value of A by the value of B".
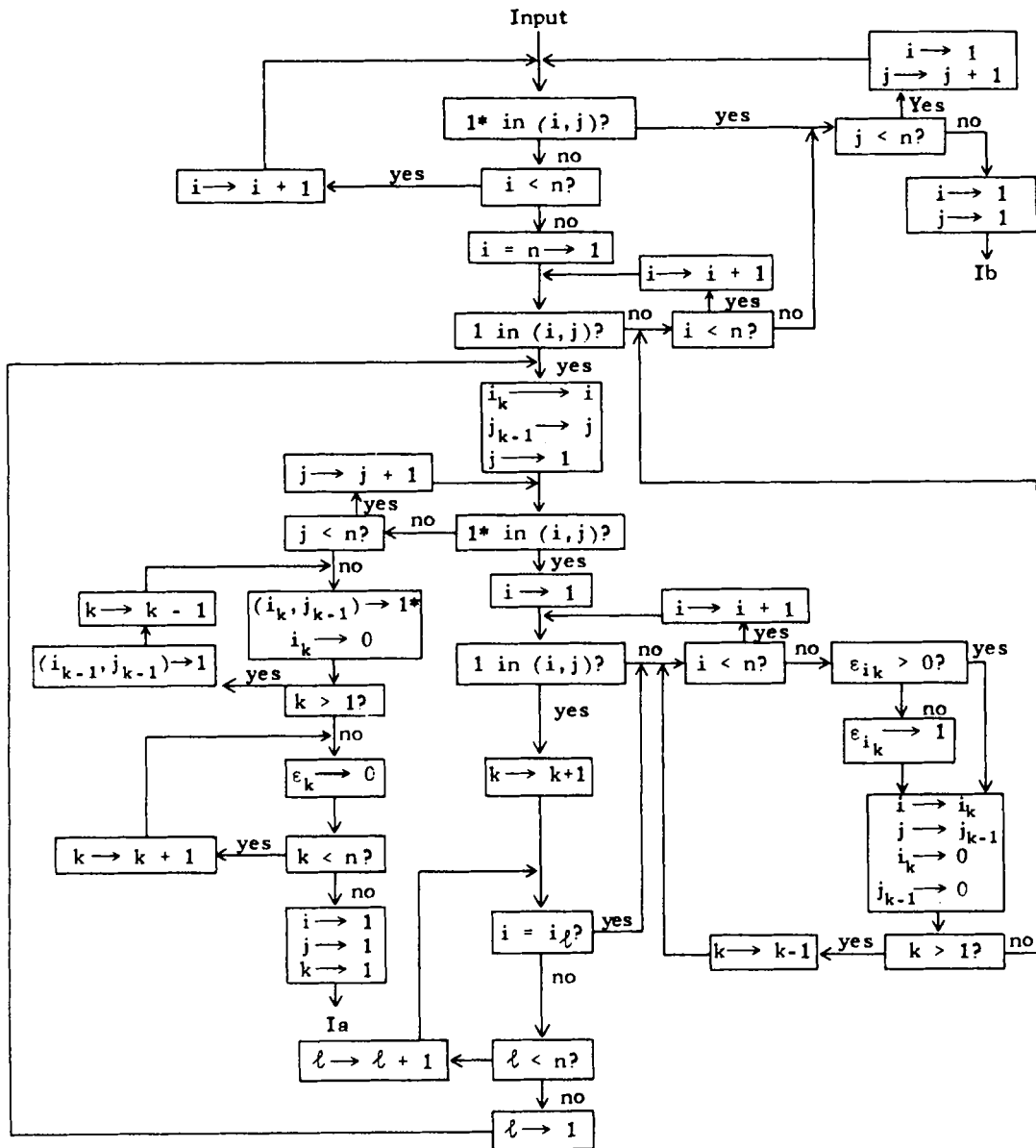

Figure 2

Routine II

The input of Routine II consists of a cover $\{u_i, v_j\}$ and a set of essential rows and columns (a column is essential if it contains a 1* in an inessential row). We first compute d, the minimum of $u_i + v_j - r_{ij}$ taken over all inessential rows i and columns j. If there are no such (i, j) then the set of 1* in Q constitutes a solution to the General Assignment Problem (Theorem 6). Otherwise, d > 0 and there are two mutually exclusive cases to be considered.

Case 1. For all inessential rows i, $u_i > 0$. Compute m, the minimum of d and u_i taken over all inessential i. Then

$$u_i \longrightarrow u_i - m \text{ for all inessential rows i, and}$$

$$v_j \longrightarrow v_j + m \text{ for all essential columns j.}$$

Case 2. For some inessential row i, $u_i = 0$. Compute m, the minimum of d and v_j taken over all inessential j. Then

$$u_i \longrightarrow u_i + m \text{ for all essential rows i, and}$$

$$v_j \longrightarrow v_j - m \text{ for all inessential columns j.}$$

After these changes have been made in the cover, we are in Alternative IIa and should return to Routine I.

5. AN EXAMPLE

The following example, although small in size, illustrates all of the possibilities of the routines (except Case 2 of Routine II):

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

Sum of row maxima   = 9 + 8 + 9 + 6 = 32.
Sum of column maxima = 8 + 7 + 9 + 9 = 33.
Hence, the initial cover is provided by the row maxima. The next table shows the successive covers obtained from the algorithm (reading out from the matrix):

The following tables explain the construction of the successive covers and of the corresponding assignments:

Stage 1.   Remarks

<table>
<tr><td></td><td>1</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
</table>

This matrix marks (with 1) those positions for which $u_i + v_j = r_{ij}$ in the first cover.

<table>
<tr><td></td><td>1*</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1*</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
</table>

Assign in each row the first 1, if any, not in the column of a previous assignment. Assignments are marked by asterisks. No transfers are possible and hence all assigned columns and no assigned rows are essential.

	0	0	
9	8	7	⧅
8	5	2	⧅
9	6	1	⧅
6	2	3	⧅

Thus, the algorithm decreases <u>all</u> u_i and increases v_3 and v_4 by the minimum of $u_i + v_j - r_{ij}$ on the part of the matrix shown at left. The second cover is:
$$u_1 = 8, u_2 = 7, u_3 = 8, u_4 = 5 \text{ and } v_1 = v_2 = 0, v_3 = v_4 = 1.$$

Stage 2.

<table>
<tr><td>1←</td><td>—1*</td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1*</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
</table>

The change in the cover has introduced a new 1 at (1,1) and there is one possible transfer, indicated by an arrow. Thus, row 1 and column 4 are essential.

	0	0	1	
⧅	⧅	⧅	⧅	⧅
7	5	2	7	⧅
8	6	1	4	⧅
5	2	3	2	⧅

Thus, the algorithm decreases u_2, u_3, and u_4 and increases v_4 by the minimum of $u_i + v_j - r_{ij}$ on the part of the matrix shown at left. The third cover is:
$$u_1 = 8, u_2 = 6, u_3 = 7, u_4 = 4 \text{ and } v_1 = v_2 = 0, v_3 = 1, v_4 = 2.$$

Stage 3.

<table>
<tr><td>1←</td><td>—1*</td><td></td><td></td></tr>
<tr><td></td><td>1←</td><td>—1*</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
</table>

The change in the cover has introduced a new 1 at (2,3) and eliminated the 1 at (1,4). The possible transfers are indicated by arrows.

The transfer $\begin{smallmatrix}1 \swarrow 2\\ 1 \swarrow 3 \swarrow 4\end{smallmatrix}$ leads to an incomplete assignment (column 4

<table>
<tr><td>1*</td><td></td><td>1</td><td></td></tr>
<tr><td></td><td></td><td>1*</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td>1*</td></tr>
<tr><td></td><td></td><td></td><td>1</td></tr>
</table>

is unassigned and (3,4) is qualified). The matrix at left completes it. All assigned columns and no assigned rows are essential because there are no transfers.

0

8	7	
6	2	
7	1	
4	3	

Thus, the algorithm decreases all u_i and increases v_1, v_3, and v_4 by the minimum of $u_i + v_j - r_{ij}$ on the part of the matrix shown at left. The fourth cover is:

$$u_1 = 7, u_2 = 5, u_3 = 6, u_4 = 3 \text{ and } v_1 = 1, v_2 = 0, v_3 = 2, v_4 = 3.$$

Stage 4.

1*	1	1	
		1*	1
			1*
	1		1

The change in the cover has introduced new 1's at (1,2) and (4,2). Thus the assignment is incomplete and is completed by assigning (4,2).

1*	1	1	
		1*	1
			1*
	1*		1

The assignment shown is optimal.

Check: $u_i + v_j \geq r_{ij}$ for all i, j.

$$r_{11} + r_{23} + r_{34} + r_{42} = 8 + 7 + 9 + 3 = 27.$$

$$u_1 + \ldots + u_4 + v_1 + \ldots + v_4 = 7 + 5 + 6 + 3 + 1 + 0 + 2 + 3 = 27.$$

BIBLIOGRAPHY

[1] König, D., "Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre." Math. Ann. 77 (1916) 453-465.

[2] Frobenius, G., "Über zerlegbare Determinanten," Sitzungsber., Preuss. Akad. Wiss. (1917) 274-277.

[3] Egerváry, J., "Matrixok kombinatorius tulajdonságairól." Mat. Fiz. Lapok (1931) 16-28 (translated as "Combinatorial Properties of Matrices" by H. W. Kuhn, ONR Logistics Project, Princeton (1953), mimeographed).

[4] Hall, P., "On Representatives of Subsets," J. London Math. Soc. 10 (1935) 26-30.

[5] Easterfield, T. E., "A Combinatorial Algorithm," J. London Math. Soc. 21 (1946) 219-226.

[6] Birkhoff, Garrett, "Tres Observaciones sobre el Algebra Lineal," Univ. Nac. Tucumán. Revista A5 (1946) 147-151.

[7] Thorndike, R. L., "The Problem of Classification of Personnel," Psychometrika 15 (1950) 215-235.

[8] Dantzig, G. B., "Application of the Simplex Method to a Transportation Problem," Chapter XXIII in Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13, ed. T. C. Koopmans, New York, 1951.

[9] Votaw, D. F. and Orden, A., "The Personnel Assignment Problem," Symposium on Linear Inequalities and Programming, SCOOP 10, USAF (1952) 155-163.

[10] Neumann, J. von., "A Certain Zero-sum Two-Person Game Equivalent to the Optimal Assignment Problem," Contributions to the Theory of Games II, Ann. Math. Study 28 (1953) 5-12.

[11] Dwyer, P. S., "Solution of the Personnel Classification Problem with the Method of Optimal Regions," Psychometrika 19 (1954) 11-26.

[12] Flood, M. M., "On the Hitchcock Distribution Problem," Pacific J. Math. 3 (1953) 369-386.

[13] Motzkin, T. S., "The Assignment Problem" in Proc. of Sixth Symposium on Applied Mathematics, to appear.

*   *   *