A note on assignment problems

By E. S. Page*

It is shown that the Bottleneck Assignment problem can be transformed into a special case of the classical Assignment problem. A minor modification of the Hungarian method of the solution for the latter produces an algorithm for the former.

Assignment problems deal with the allocation of items to locations, one to each, in such a way that some optimum return is obtained. Four examples are the following:

1A. The expected value of orders that will be obtained by salesman i $(i = 1, 2, \ldots, n)$ in sales area j $(j = 1, 2, \ldots, n)$ is known to be Γ_{ij}. Assign the salesmen one to each area so that the total expected value is greatest.

1B. The comfort and convenience of class i in room j is assessed by an index Γ_{ij}. It is wished to assign the classes, one to each room, so that the least index assigned is maximized.

2A. In 1A suppose that Γ_{ij} is the distance salesman i has to travel to reach area j. Assign the salesmen, one to each area, so that the total distance travelled is a minimum.

2B. Mechanics have to repair pieces of equipment; the time, Γ_{ij}, for mechanic i to repair equipment j is known. Assign the mechanics one to each equipment to minimize the time by which all equipments are repaired.

Thus, given the $n \times n$ matrix (Γ_{ij}) the four examples require the selection of n elements of the matrix, one from each row and each column $\Gamma_{1j_1}, \Gamma_{2j_2}, \ldots \Gamma_{n,j_n}$, where $(j_1, j_2, \ldots j_n)$ is some permutation, π, of $(1, 2, \ldots n)$ in order to:

1A. $\underset{\pi}{\text{Maximize}} \, (\Gamma_{1j_1} + \Gamma_{2j_2} + \ldots + \Gamma_{nj_n})$

1B. $\underset{\pi}{\text{Maximize}} \, \text{Minimum} \, (\Gamma_{1j_1}, \Gamma_{2j_2}, \ldots \Gamma_{nj_n})$

2A. $\underset{\pi}{\text{Minimize}} \, (\Gamma_{1j_1} + \Gamma_{2j_2} + \ldots + \Gamma_{nj_n})$

2B. $\underset{\pi}{\text{Minimize}} \, \text{Maximum} \, (\Gamma_{1j_1}, \Gamma_{2j_2}, \ldots \Gamma_{nj_n})$.

The usual forms of the assignment problem are 1A and 2A. We can restate these problems as follows. Let $x_{ij} = 1$ if item i is assigned to location j, and 0 otherwise. Then 1A and 2A require the optimization of $\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}\Gamma_{ij}$ subject to the conditions

$$\sum_{i=1}^{n} x_{ij} = 1 \,, \quad j = 1, 2, \ldots n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \,, \quad i = 1, 2, \ldots n$$

$$x_{ij} \geqslant 0 \,, \quad i, j = 1, 2, \ldots n.$$

Accordingly problems 1A and 2A are special cases of the transportation problem which itself is a special case of the more general linear programming problem. While the algorithms for the general problems could be used to solve these assignment problems, special methods have been devised which take advantage of the simplifying features (Kuhn, 1955).

We now show how a simple transformation exhibits 1B and 2B (the Bottleneck problems) as special cases of 1A and 2A.

The transformations

We can suppose that all the $\Gamma_{ij} \geqslant 0$ without loss of generality, since any optimum assignment for the matrix (Γ_{ij}) is also optimum for (Γ'_{ij}) where $\Gamma'_{ij} = \Gamma_{ij} + c$, for any constant c. In 1A, 2A the optimum is increased by nc and in 1B, 2B by c.

Let $M = \max_{i,j} \Gamma_{i,j}$; then problems 1A and 2A are transformed into one another by writing $\Gamma'_{ij} = M - \Gamma_{ij}$. That is, an optimum assignment for problem 1A with cost γ is an optimum assignment for 2A with cost $nM - \gamma$.

For the problems 1B, 2B we can in addition suppose the elements Γ_{ij} to be positive integers, since if they are not we can rank the Γ_{ij} in ascending order of magnitude and make the transformation $\Gamma'_{ij} = \Gamma_{ij}$, where Γ'_{ij} is the rank of Γ_{ij} in the ordering. Any ties, i.e. where $\Gamma_{ij} = \Gamma_{kl}$, will have the same rank, $\Gamma'_{ij} = \Gamma'_{kl}$. For example the matrix

$$\begin{pmatrix} 3\cdot 1 & 1\cdot 8 & 3\cdot 1 \\ \pi & 9 & \sqrt{2} \\ e & 10^5 & \sqrt{2} \end{pmatrix} \quad \text{becomes} \quad \begin{pmatrix} 4 & 2 & 4 \\ 5 & 6 & 1 \\ 3 & 7 & 1 \end{pmatrix}.$$

Thus, we can suppose the Γ_{ij} to be positive integers in the range $(1, n^2)$. Then the transformation $\Gamma'_{ij} = n^2 + 1 - \Gamma_{ij}$ transforms 1B and 2B into one another.

* This research was supported by the Office of Naval Research under contract No. Nonr-855(09) for research in probability and statistics at the University of North Carolina, Chapel Hill, N.C. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Dr. Page's present address is: University Computing Laboratory, Newcastle upon Tyne.

Consider now how to transform problem 2B into 2A. If we can arrange that any sum of elements of the matrix is dominated by the greatest element in the sum (or equal greatest elements) then we shall be able to solve 2B by the methods used for 2A. We therefore need to replace the Γ_{ij} by numbers widely spread out. This task is similar to that of isolating the root of largest modulus of a polynomial equation and can be tackled by methods similar to those of the root squaring technique.

Suppose we form a new matrix $\Gamma'_{ij} = \Gamma^{\alpha}_{ij}$ where α is to be chosen. We require that

$$\Gamma'_{1j_1} + \ldots + \Gamma'_{nj_n} < \Gamma'_{1k_1} + \ldots + \Gamma'_{nk_n} \qquad (1)$$

should imply

$$\mathrm{Max}\,(\Gamma'_{1j_1}, \ldots \Gamma'_{nj_n}) \leqslant \mathrm{Max}\,(\Gamma'_{1k_1}, \ldots \Gamma'_{nk_n}). \qquad (2)$$

Choose α so that

$$s^{\alpha} \geqslant n(s-1)^{\alpha} \text{ for } s = 1, 2, \ldots m \qquad (3)$$

where m is the greatest element of (Γ_{ij}). With this choice of α (1) certainly implies (2).

Alternatively the new matrix can be formed $\Gamma''_{ij} = K^{\Gamma_{ij}}$ for a suitable choice of K. The requirement that (1) implies (2) is met if

$$K^{s} \geqslant nK^{(s-1)} \text{ for } s = 1, 2, \ldots m \qquad (4)$$

i.e. take $K \geqslant n$ where the matrix is $n \times n$.

Thus either of these transformations reduces problem 2B into 2A and so may be solved by known algorithms.

The transformation of problem 1B into 1A now follows easily. We go from 1B→2B→2A→1A. Thus the transformations are

$$\Gamma'_{ij} = K' - (M - \Gamma_{ij})^{\alpha}$$

or.

$$\Gamma'_{ij} = K'' - K^{(M - \Gamma_{ij})}$$

where K', K'' are large enough integers to make $\Gamma'_{ij} > 0$.

Example

Consider the example given by Kuhn to illustrate the Hungarian method of solution to problem 1A. We modify it to get a solution of 1B, i.e. to maximize $\min\,(\Gamma_{1j_1}, \ldots \Gamma_{nj_n})$.

$$(\Gamma_{ij}) = \begin{pmatrix} 8 & 7 & 9 & 9 \\ 5 & 2 & 7 & 8 \\ 6 & 1 & 4 & 9 \\ 2 & 3 & 2 & 6 \end{pmatrix}.$$

The elements Γ_{ij} are already consecutive integers with repetitions beginning at unity, so we have no need to rank them and replace with the ranks.

Problems 2B considers the matrix $(\Gamma^{(1)}_{ij})$ where $\Gamma^{(1)}_{ij} = 9 - \Gamma_{ij}$, and then for problem 2A we can use the matrix $(\Gamma^{(2)}_{ij})$, where $\Gamma^{(2)}_{ij} = 4^{\Gamma^{(1)}_{ij}}$. Finally problem 1A uses $(\Gamma^{(3)}_{ij})$, where $\Gamma^{(3)}_{ij} = 65536 - \Gamma^{(2)}_{ij}$. Thus

$$(\Gamma^{(1)}_{ij}) = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 4 & 7 & 2 & 1 \\ 3 & 8 & 5 & 0 \\ 7 & 6 & 7 & 3 \end{pmatrix}, \quad \Gamma^{(2)}_{ij} = \begin{pmatrix} 4 & 16 & 1 & 1 \\ 256 & 16384 & 16 & 4 \\ 64 & 65536 & 1024 & 1 \\ 16384 & 4096 & 16384 & 64 \end{pmatrix}$$

$$\Gamma^{(3)}_{ij} = \begin{pmatrix} 65532 & 65520 & 65535 & 65535 \\ 65280 & 49152 & 65520 & 65532 \\ 65472 & 0 & 64512 & 65535 \\ 49152 & 61440 & 49152 & 65472 \end{pmatrix}.$$

Accordingly this matrix could form the input to a program for the more general solution to problem 1A, and an optimum assignment so obtained will be optimum for problem 1B on matrix (Γ_{ij}); the optimum cost can then be obtained directly from (Γ_{ij}) or by reversing the transformations.

Modification of the Hungarian method

In this section we assume familiarity with the Hungarian method, and refer to Kuhn's paper (1955) for the details and a flow diagram of the solution.

The disadvantage of trying to solve 1B and 2B by the algorithm of 1A, 2A on the transformed matrix is the very large size of the matrix elements. Even for quite small problems, say 10×10, the transformed numbers have far outstripped single-length working in a computer. Some minor steps can be taken to reduce the size. For example, the element of (Γ_{ij}) giving the optimum for 1B cannot be greater than the nth largest element of the matrix. Accordingly the n largest elements of (Γ_{ij}) can be replaced by their minimum without affecting either an optimum assignment or its value. In the example given this device reduces the numbers by about a factor of 4, and in the $n \times n$ case by a factor of at most n^n. Constants can be subtracted from rows or columns of (Γ_{ij}) changing only the value of an assignment but not its optimality. Even after all these adjustments, large numbers still remain. They can, however, be avoided by changing the direction of attack: instead of transforming the matrix for 1B to fit the Hungarian or another method for 1A we consider transforming (very slightly) the Hungarian method to make it applicable for 1B.

The Hungarian method first marks the positions in the matrix of row (or column) maxima and attempts to complete an assignment with the marked positions. If this fails it gives rules for selecting areas, R, of the matrix in which further items to be marked should be found, together with a rule for picking those to be marked. More positions are marked until finally a complete assignment can be made.

The only section of this algorithm which needs modification to deal with problem 1B is the rule for picking the items to be marked. The appropriate rule is: mark the item or items in R which are the greatest in R. (In the modified algorithm it is not necessary to compute explicitly the cover (u_i, v_j) required for 1A.) We illustrate the procedure on the previous example; marked items are replaced by X, items in the selected area are shown with their numerical values, and other items are replaced by —. We refer to Kuhn for an explanation of how the areas are selected. The stages are as follows:

$$\begin{pmatrix} \textcircled{8} & 7 & X & X \\ 5 & 2 & - & X \\ 6 & 1 & - & X \\ 2 & 3 & - & X \end{pmatrix} \rightarrow \begin{pmatrix} X & - & X & X \\ 5 & 2 & \textcircled{7} & X \\ 6 & 1 & 4 & X \\ 2 & 3 & 2 & X \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} X & \textcircled{7} & X & - \\ - & 2 & X & X \\ - & 1 & - & X \\ - & 3 & - & X \end{pmatrix} \rightarrow \begin{pmatrix} X & X & X & - \\ 5 & 2 & X & X \\ \textcircled{6} & 1 & - & X \\ 2 & 3 & - & X \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} X & \boxed{X} & - & - \\ - & - & \boxed{X} & X \\ \boxed{X} & - & - & X \\ - & - & - & \boxed{X} \end{pmatrix}$$

The greatest elements in the selected areas are enclosed in a circle and the actual assignment is shown by squares, i.e. Γ_{12}, Γ_{23}, Γ_{31}, Γ_{44} with optimum value

$$= \min(7, 7, 6, 6) = 6.$$

We refer to Kuhn's paper for an explanation why the Hungarian method works. In this modification, at each stage the selected areas are those which need a representative item to take the partial assignment nearer a complete one. Naturally the largest item in the area is chosen. An algebraic justification may be written in terms of the "cover" for the transformed matrix.

Comments on the method

The modification above has as its greatest virtue the slight changes it requires to an already established algorithm. If a computer program for the Hungarian method is available a program for the solution of 1B can be derived easily. If, however, only problems of types 1B, 2B are likely to be encountered and a program has to be written afresh the method developed by Porsching (1963) has undoubted advantages as it is much easier to program. This latter method proceeds by marking the elements starting with all the largest, then all the next largest and so on, and providing at each stage an easy, although possibly long, rule for deciding whether a complete assignment can be formed from the marked items. The modified Hungarian method tends only to mark items that may be useful in completing an assignment, and has another way of deciding whether a complete assignment is yet possible.*

An entirely different approach would be necessary to deal with very large problems. The problems considered can easily be formulated in Dynamic Programming terms. Let $C\{i_1, i_2, \ldots i_k; j_1, j_2, \ldots j_k\}$ be the cost or value of an optimum assignment of items $i_1, \ldots i_k$ to places $j_1, \ldots j_k$. Then clearly this cost $C\{i_1, \ldots i_k; j_1, \ldots j_k\}$ can be obtained by examining the costs, and picking the best, of assignments of i_k to $j_1, j_2, \ldots j_k$ in turn together with an optimum assignment of $i_1, \ldots i_{k-1}$ to the $(k-1)$ places not filled. Thus

$$C\{i_1, \ldots i_k; j_1, \ldots j_k\} = \underset{1 \leqslant s \leqslant k}{\text{Optimum}} \left[\Gamma_{i_k j_s} \oplus C\{i_1, \ldots i_{k-1}; j_1, \ldots j_{s-1}, j_{s+1}, \ldots j_k\}\right]$$

where the "Optimum" and $\oplus$ operations depend on which of the four problems is being solved. Thus, to solve an $n \times n$ problem we need to build up successively a table of optimum costs of all the $m \times m$ problems, $m = 1, 2, \ldots n$ that can be formed by selecting items and places. The number of such problems is largest when $m = n/2$ and is $\binom{n}{n/2}^2 \sim 2^{2n+1}/\pi n$, and their costs need to be available for computing the next stage. A practical method is obtained by grouping the items and places into problems of a size that can be tackled and switching the components between groups in some way similar to that of Held and Karp (1962) in another problem in order to improve the solution already found.

Acknowledgements

I wish to thank my friends, W. J. Hall and W. L. Smith, for discussions relevant and irrelevant but always entertaining during my stay at Chapel Hill.

* I am indebted to Dr. S. Vajda for reading this note in draft and for telling me of a modification by O. Gross (1959) of the Ford–Fulkerson method for problems 1A, 2A.

References

GROSS, O. (1959). "The Bottleneck Assignment problem," Rand report P-1630.

HELD, M., and KARP, R. M. (1962). "A Dynamic Programming approach to sequencing problems," *J. Soc. Indust. Appl. Math.*, Vol. 10, p. 196.

KUHN, T. W. (1955). "The Hungarian method for the assignment problem," *Nav. Res. Log. Quart.*, Vol. 2, p. 83.

PORSCHING, T. A. (1963). "Matrix assignments and an associated min-max problem," *Math. Comp.*, Vol. 17, p. 81.

VAJDA, S. (1961). *Mathematical Programming*, Addison-Wesley, Reading, Mass., pp. 141–47.