THE PERSONNEL ASSIGNMENT PROBLEM

D. F. Votaw, Jr. and A. Orden*

Summary

A basic personnel-classification problem and variants of it are stated (section 1), and methods for obtaining exact or approximate solutions are indicated (section 2). The capabilities of a high speed computer for carrying out the simplex method of solution are discussed (section 3).

§1. Introduction

Suppose that with regard to N persons there are m mutually exclusive personnel categories and that with regard to N jobs there are n mutually exclusive job categories. Let a_i and b_j be, respectively, the number of persons in the ith category and the number of jobs in the jth category $(\sum_i a_i = \sum_j b_j = N;\ a_i, b_j > 0;\ i = 1, \ldots, m;\ j = 1, \ldots, n)$. Assume that any two persons in the ith category are essentially identical and that any two jobs in the jth category are essentially identical. Let c_{ij} be the productivity of any person in the ith category on any job in the jth category (c_{ij} is a real number, called a <u>score</u>; the larger the score, the greater the productivity); also, let x_{ij} be any possible number of persons in category i placed on jobs in category j. The personnel-classification (assignment) problem (see [5, p. 217]) is to find values of $x_{11}, \ldots, x_{mn}$ such that $\sum_{i,j} x_{ij}c_{ij}$ assumes its <u>maximum</u> value, where

$$\sum_i x_{ij} = b_j,$$

$$(1{:}1) \qquad \sum_j x_{ij} = a_i,$$

$$x_{ij} \geq 0,$$

$(x_{ij}, a_i, b_j$ integers; $i = 1, \ldots, m;\ j = 1, \ldots, n)$.

The problem is equivalent to the Hitchcock-Koopmans transportation problem[1] (see [1], [3], [4]).

---

* Yale University and Department of the Air Force, respectively.

1 c_{ij} could be interpreted as negative cost.

THE PERSONNEL ASSIGNMENT PROBLEM

D. F. Votaw, Jr. and A. Orden*

Summary

A basic personnel-classification problem and variants of it are stated (section 1), and methods for obtaining exact or approximate solutions are indicated (section 2). The capabilities of a high speed computer for carrying out the simplex method of solution are discussed (section 3).

§1. Introduction

Suppose that with regard to N persons there are m mutually exclusive personnel categories and that with regard to N jobs there are n mutually exclusive job categories. Let a_i and b_j be, respectively, the number of persons in the ith category and the number of jobs in the jth category $(\sum_i a_i = \sum_j b_j = N;\ a_i, b_j > 0;\ i = 1, \ldots, m;\ j = 1, \ldots, n)$. Assume that any two persons in the ith category are essentially identical and that any two jobs in the jth category are essentially identical. Let c_{ij} be the productivity of any person in the ith category on any job in the jth category (c_{ij} is a real number, called a <u>score</u>; the larger the score, the greater the productivity); also, let x_{ij} be any possible number of persons in category i placed on jobs in category j. The personnel-classification (assignment) problem (see [5, p. 217]) is to find values of $x_{11}, \ldots, x_{mn}$ such that $\sum_{i,j} x_{ij}c_{ij}$ assumes its <u>maximum</u> value, where

$$\sum_i x_{ij} = b_j,$$

$$(1:1) \qquad \sum_j x_{ij} = a_i,$$

$$x_{ij} \geq 0,$$

$(x_{ij}, a_i, b_j$ integers; $i = 1, \ldots, m;\ j = 1, \ldots, n)$.

The problem is equivalent to the Hitchcock-Koopmans transportation problem[1] (see [1], [3], [4]).

---

* Yale University and Department of the Air Force, respectively.

1 c_{ij} could be interpreted as negative cost.

A variant of this problem is to find a largest-minimum-score assignment (i.e., to find values of $x_{11}, \ldots, x_{mn}$ such that the minimum element of the set of c_{ij}'s associated with positive x_{ij}'s assumes its maximum value).

An important special case of the general problem is that in which each c_{ij} equals 0 or 1. This arises, for example, when each person is scored on each job simply as qualified or not qualified. A variant of this special case can be stated as follows: (a) Is N the maximum of $\sum_{i,j} x_{ij}c_{ij}$? (b) If so, find values of $x_{11}, \ldots, x_{mn}$ for which the sum equals N. The variant arises when the assigning agency wishes to have each person qualified on the job to which he's assigned. When m or n is "small", this variant is termed the "quota problem". By repeated solution of the variant one can obtain a largest-minimum-score assignment.

Application of the simplex method (or the method discussed in [4]) to personnel-classification problems yields mathematically satisfactory solutions; but whether these methods are fully satisfactory in practice is perhaps an open question. The purposes of this paper are: (i) to present several methods (of dealing with personnel-classification problems) which, though not necessarily optimal, are simpler and more rapid than the simplex method (see section 2)[1]; (ii) to discuss application of the simplex method, involving use of a high-speed computing machine (see section 3). Even when the simplex method is used, the methods discussed in section 2 might be helpful in selecting a "good" assignment for the first step of the method.

§2. Discussion of the General Problem and Special Cases.

Throughout this section we shall assume, except in 2B(3), that $a_1 = \cdots = a_m = b_1 = \cdots = b_n = 1$ (then $N = m = n$). This assumption involves no loss of generality. The $n \times n$ array, (c_{ij}), will be represented by C. The set of cells $(1,j_1), \ldots, (n,j_n)$ of C, where $j_1, \ldots, j_n$ is a permutation of $1, \ldots, n$, is termed a <u>two-way permutation</u>. The set $(1,j_1), \ldots, (n,j_n)$ is also termed an <u>assignment</u>; it means placement of person 1 on job $j_1, \ldots,$ placement of person n on job j_n. The set $c_{1j_1}, \ldots, c_{nj_n}$ and the sum $\sum_i c_{ij_i}$ are termed, respectively, a <u>permutation set</u> and a <u>permutation sum.</u>

---

[1]The method described in 2B(3) cannot be carried out quickly except when m or n is small.

2A. Rapid methods for the general problem, not necessarily optimal

(1) Properties of the permutation sum of a random assignment. Select $(j_1, \ldots, j_n)$ at random (the probability that any given assignment is selected is $1/n!$). The expected value and variance of the random sum $S = \sum_i c_{ij_i}$ are, respectively:

$$E(S) = \sum_{i,j} c_{ij}/n,$$

(2:1)

$$\sigma_S^2 = (n-1)^{-1}[\sum_{i,j} c_{ij}^2 + (T/n)^2 - n^{-1}(\sum_i r_i^2 + \sum_j z_j^2)], \; (n>1),$$

where $T = \sum_{i,j} c_{ij}$, $r_i = \sum_j c_{ij}$, $z_j = \sum_i c_{ij}$.

(2) The n-step method. Select n two-way permutations such that no two have an element in common. The average of the n permutation sums is $\sum_{i,j} c_{ij}/n = E(S)$ (see (2:1)); thus at least one of these n sums is at least as large as $E(S)$.

(3) The column (row) maximal method. In C choose a column, j_1, arbitrarily and select a largest element in it, say $c_{i_1j_1}$. In the $(n-1) \times (n-1)$ minor associated with $c_{i_1j_1}$ choose a column arbitrarily, say j_2, and select a largest element in it, say $c_{i_2j_2}$. Continuing this procedure until all columns have been selected, we have a permutation set $c_{i_1j_1}, \ldots, c_{i_nj_n}$, termed a column maximal permutation set. A row maximal permutation set is defined in an entirely similar way. There are $n!$ formally distinct column (row) maximal sets; their mean is at least as large as $E(S)$.

(4) The cyclic column (row) maximal method. When $j_1, \ldots, j_n$ in (3) form a cyclic permutation of $1, \ldots, n$, the column maximal set is said to be cyclic. A cyclic row maximal set is defined similarly. There are n formally distinct cyclic column (row) maximal sets; the mean of the corresponding n permutation sums is at least as large as $E(S)$ (see (2:1)); thus, as in the case of the n-step method, a permutation sum at least as large as $E(S)$ can be found by inspecting at most n permutation sets.

(5) Switching pairs. Given a two-way permutation $(i_1, j_1), \ldots, (i_n, j_n)$, choose two c_{ij}'s associated with it, say $c_{i_1j_1}$ and $c_{i_2j_2}$, and note whether

$$c_{i_1 j_1} + c_{i_2 j_2} < c_{i_2 j_1} + c_{i_1 j_2}.$$

If so, the sum associated with $(i_1, j_1), \ldots, (i_n, j_n)$ is smaller than the sum associated with $(i_2, j_1), (i_1, j_2), \ldots, (i_n, j_n)$, and thus the second two-way permutation yields a better assignment than the first.

(6) An elimination method. It is generally possible to determine some elements in C that are not in a maximum-sum permutation set. Consider any element, c_{ij}. If for some $w \neq j$,

$$c_{ij} + c_{i'w} < c_{iw} + c_{i'j} \quad \text{(for all } i' \neq i\text{)},$$

then c_{ij} is not in a maximum-sum permutation set. Similarly, c_{ij} is not in a maximum-sum set if for some $u \neq i$,

$$c_{ij} + c_{uj'} < c_{ij'} + c_{uj} \quad \text{(for all } j' \neq j\text{)}.$$

(7) A use of bounding-sets. A set of $2n$ numbers $d_1, \ldots, d_n$, $e_1, \ldots, e_n$ is termed a bounding-set, with regard to C, if for all (i, j)

$$c_{ij} \leq d_i + e_j. \tag{2:2}$$

Such a set can be constructed easily. A bounding-set may supply useful information in an assignment problem since for any such set the maximum permutation sum, say $\overline{s}$, is not greater than $\sum_i d_i + \sum_j e_j$. There exists a minimal bounding set (i.e., a set $d'_1, \ldots, d'_n$, $e'_1, \ldots, e'_n$ such that $\sum_i d'_i + \sum_j e'_j = \overline{s}$ and such that for each c_{ij} in some permutation set equality holds in (2:2)). The final solution in the simplex method yields a minimal bounding-set.

2B. Methods relevant to the case where each $c_{ij} = 0$ or 1.

(1) Inspection of subsets of rows and columns. Note whether there is at least one row (column) devoid of 1's. If so, obviously the maximum sum is less than N. More generally, note whether there is a set of h rows (columns) having 1's in at most $(h - 1)$ columns (rows); if so, the maximum sum is less than N $(h = 1, \ldots, n)$. The procedure is easier to carry out after the rows (also columns) are ordered so that the row sums (column sums) are in, say, non-decreasing order. This method is usually prohibitively laborious except when n is small. However, when the answer to question (a) in section 1 is No, that fact may happen to be determined quickly by use of this method, even when n is not small.

(2) Deletion of rows and columns. Choose any of the 1's in C and delete the rest of its row and column. If there are other 1's left, choose one of them and delete its row and column. Continue until all 1's have been chosen or deleted. Let r be the number of 1's chosen. Clearly these r 1's belong to a permutation set. If $r = n$, a solution of the problem has been obtained. In any event, we have that $r \leq \bar{s} \leq 2r$, where $\bar{s}$ is the maximum permutation sum; thus, if $2r < n$, the answer to (a) is No.

(3) Filling quotas. We assume that each $c_{ij} = 0$ or 1 but not that $m = n = N$ (see section 1). Without loss of generality we assume that $m \geq n$. Question (a) of the variant problem (see section 1) has the answer Yes or No according as it is or is not true that every one of the following $2^n - 1$ inequalities holds (see [2]):

$$(2:3) \quad \begin{aligned} & g_1 \geq b_1, \quad g_2 \geq b_2, \; \ldots, \; g_n \geq b_n, \\ & g_{12} \geq b_1 + b_2, \; \ldots, \; g_{n-1\,n} \geq b_{n-1} + b_n, \\ & \vdots \\ & g_{12\ldots n} \geq b_1 + \cdots + b_n, \end{aligned}$$

where $g_{h_1 \ldots h_p}$ is the number of persons qualified for at least one of the job categories $h_1, \ldots, h_p$ $(1 \leq p \leq n)$. Suppose the answer to question (a) is Yes. By repeatedly placing a person on a job tentatively and determining from (2:3) whether the remaining persons can fill the remaining quotas, we can answer question (b).

§3. Machine Computation Using the Simplex Method

The personnel-classification problem being considered here requires solution of the algebraic system:

$$(3:1) \quad \begin{aligned} & \sum_{j=1}^{n} x_{ij} = a_i, \\ & \sum_{i=1}^{m} x_{ij} = b_j, \\ & \text{all } x_{ij} \geq 0; \end{aligned}$$

Find the maximum of $\sum_{i,j} c_{ij} x_{ij}$ where a_i, b_j, and c_{ij} are given.

This algebraic problem has a variety of possible applications. In the present discussion we are considering x_{ij} as the number of men of type i who are to be assigned to job type j, with c_{ij} as scores based on psychological testing. The same algebraic problem was formulated a few years ago as a "transportation problem" in which x_{ij} are the quantities of a certain item which should be shipped from origins, i, to destinations, j; a_i and b_j are the amounts specified to be shipped and received; and c_{ij} are the shipping costs per unit of the item in going from i to j. The objective is to minimize the total shipping cost. Another problem, already discussed at this symposium by Dr. Holley, which leads to the same mathematical formulation is that of optimum aircraft allocation.

In view of the various possible applications, a code has been set up for solution of (3:1) on the Bureau of Standards automatic electronic computer (the SEAC). The code is based on the simplex method, described in references [1] and [3]. Since an explanation of the method is available, it will not be described here. Questions as to the size of problems which can be carried out on the automatic computer and the computation time required will now be considered. Background considerations as well as results will be presented.

In the SEAC, and in similar electronic computers, moderate size matrix problems can be handled by feeding all the data and computing instructions into the <u>high speed memory</u> of the machine at the beginning of the problem, and reading the solution out of the high speed memory at the end of the computation. On the other hand in problems involving very large matrices the high speed memory is not large enough to hold the entire matrix, and it becomes necessary to hold initial data and intermediate results in <u>auxiliary memory</u>, namely, on magnetic tapes or on a magnetic drum. The present SEAC code is for moderate size problems which can be put completely in the high speed memory. We can, if it becomes desirable, revise the code to handle larger problems, using magnetic tape for auxiliary memory.

Let m be the number of job types and n be the number of personnel types. The present SEAC code permits solution of problems with $m \times n$ up to 150. Accordingly, we recently ran a simple calculation in which 18 men were to be assigned to 8 job types.

The limitation on the size of the problem comes about as follows: The SEAC has 512 "registers" of mercury delay line memory (high speed memory). About 250 registers are required for the computation instructions, about 75 for storing intermediate and final results of the computation, about 30 for the quantities a_i and b_j of (3:1), and this leaves about 150 for the table of c_{ij}'s of (3:1). The space available for storing c_{ij} values limits the size of problem which can be handled.

An increase in the capacity of the high speed memory would permit more than a corresponding increase in the size of problem since the instruction code would remain practically unchanged and the additional capacity could be used entirely for data. Recently a bank of Williams tube memory has been added to the SEAC. The number of registers of additional memory available is not yet certain as the engineers are still working on the problem of how closely distinct spots can be spaced on the surface of an oscilloscope tube. They are certain of 256 additional registers and hope they can go up to 512. With additional capacity of 256 we will be able to run problems with $m \times n$ up to about 300, and with 512 additional registers $m \times n$ can go up to about 500. (Minor changes would be needed in the code in order to make use of the Williams tube memory. These changes have not yet been made.)

As to computation time, it should be noted that for moderate size problems, say $m \times n$ up to 500, the time of computation is of the same order of magnitude as the time required to type the initial data. The computation time on a sample computation in which m and n were both 10 was 3 minutes. The time of computation can be shown by study of the computing method and the code to be proportional to $(m+n)^3$. Thus a 10 x 50 problem would require 27 times as long as a 10 x 10 problem, or about 80 minutes. The data of a 10 x 50 problem consist of 500 values of c_{ij}, 10 values of a_i and 50 values of b_j. Assuming 5 seconds to type and check each data value, the time to type the data for feeding the 10 x 50 problem into the machine would be 47 minutes. Thus as already noted the computing time for moderate size problems is comparable to the data typing time.. For very large problems the computing time would become dominant since it rises in proportion to $(m+n)^3$, while the typing time rises in proportion to mn.

A further remark on computing time may be of interest. The time for an individual operation on the SEAC (addition, comparison, multiplication, etc.) is in the range 1 to 3 milliseconds. Typical elementary computations which occur in the simplex method (and in many other computing machine codes) consist of a series of a few individual operations, e.g. one or two additions, multiplications, and comparisons. Such a series of elementary operations takes on the order of 20 milliseconds. Since these typical series of operations are repeated about $(m+n)^3$ times, multiplying 20 milliseconds by $(m+n)^3$ gives a fair approximation to the computing time.

To deal with problems with $m \times n$ greater than 500 it would be necessary to use magnetic tape for storage of data. On the SEAC this would considerably increase the computation time. The SEAC is limited here by the fact that its design does not provide for reading or writing on magnetic tape simultaneously with computation. As a result the time estimate given earlier [proportional to $(m+n)^3$] should be increased by a factor of 2 for large problems. Other computing machines, in particular the UNIVAC, are designed for handling magnetic tape simultaneously with computation. In such machines the use of magnetic tapes for auxiliary memory would have little effect on computing time.

In solution of (3:1) on a computing machine by the simplex method, we can speak with impunity of very large problems. This arises from the fact that the simplex process, applied to (3:1), is a problem in which there are no round-off errors. This follows from reference [1] where it is shown that the essential arithmetic of the process can be set up in terms of additions and subtractions alone, that is, without multiplication or division. (Multiplication and division appear in the SEAC code, but are not intrinsic to the computation, e.g. they are used for shifting digits to the right or left.)

With regard to application of the computing machine to personnel classification, some psychometricians have indicated that problems in which there are 8 types of job and 50 types of man are likely to be of useful size. This, as has been indicated earlier, would be quite feasible for the machine -- each problem would require about $1\frac{1}{2}$ hours on the basis of the present method of coding. For routine application it would probably be desirable to introduce checking features into the code, which would somewhat lengthen the computing time. On the other hand, this might be compensated by careful review to find ways of saving steps in the code.

§4. Acknowledgements

The author of sections 1 and 2 (DFV) wishes to acknowledge valuable discussions with J. W. Tukey, M. A. Woodbury, and J. T. Dailey regarding formulation of the main problem stated in section 1 and with S. Kakutani, W. G. Lister, and John Williams regarding some of the methods described in section 2.

References

[1] George B. Dantzig, "Application of the Simplex Method to a Transportation Problem" (Chapter XXIII in Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13, edited by T. C. Koopmans, John Wiley and Sons, New York, 1951).

[2] Philip Hall, "On Representatives of Subsets", Journal of the London Mathematical Society, Vol. 10 (1935), pp. 26-30.

[3] Alex Orden, "A Procedure for Handling Degeneracy in the Transportation Problem", Planning Research Division, Program Standards and Cost Control, DCS/Comptroller, Headquarters USAF, April, 1951.

[4] Julia Robinson, "A Note on the Hitchcock-Koopmans Problem", The RAND Corporation, June 15, 1950.

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