How to Find Optimal Solutions to Assignment Problems 1/ 2/

Leo Törnqvist

August 3, 1953

Assignment problems have been studied in a number of papers by Von Neumann [1], Koopmans and Beckmann [2], Dantzig [3], Motzkin and others. The problems studied are all special cases of the general problem of how to select an admissible subset of a given set of real numbers so as to minimize (or maximize) the sum of the numbers included in the subset selected.

We consider a finite set x of labelled real numbers x_k, where the label (location index) k ranges over some index set K, and a class P of admissible subsets p of K. The sets p are thus elements of the class P and the elements of the set p are indices $k \in K$. For each set p we study the sum

$$S_p(x) = \sum_{k \in p} x_k ,$$

1/ Research undertaken under contract between the Cowles Commission for Research in Economics and The RAND Corporation.

2/ This note was prepared under pressure of time just before Mr. Törnqvist's return to Finland. I have done some putting together and a limited amount of editing which, I hope has not distorted the author's intentions. T. C. Koopmans.

We are interested in simple methods, using only additions, (subtractions), sorting and tabulations, for the purpose of determining the extremal subclasses $\check{P} = \check{P}(x)$ $\hat{P} = \hat{P}(x)$, respectively, of sets $p \in P$, which are such that

$$S_{\check{p}}(x) = \underset{p \in P}{\text{Min}}\, S_p(x) \qquad \text{if and only if } \check{p} \in \check{P}(x)$$

and

$$S_{\hat{p}}(x) = \underset{p \in P}{\text{Max}}\, S_p(x) \qquad \text{if and only if } \hat{p} \in \hat{P}(x).$$

Problems of finding $\check{P}(x)$ or $\hat{P}(x)$ shall be called assignment problems. This name could perhaps be given to a still broader class of problems where the task is to <u>assign</u> weights w_k to the given numbers x_k such that the weighted sum

$$S_w(x) = \sum_{k \in K} w_k x_k$$

will have some desired property, when it is required that the set $w = \left\{ w_k \right\}_{k \in K}$ of weights belong to a certain class W of admissible w.
In the special cases when the task is to find $\check{P}(x)$ the set w of admissible weights are sets of numbers $w_k = \begin{cases} 0 \\ 1 \end{cases}$ such that $w_k = 1$ for $k \in p \in P$ $w_k = 0$ for $k \notin p$. The problem to find $\check{P}(x)$ or $\hat{P}(x)$ could more precisely be called a problem of finding optimal admissible sums. These problems can however be used as keys for solving more general assignment problems. If for instance the class w of admissible weights w_k is such that every admissible $w_k = \varepsilon n_k$ where n_k is an integer this general assignment problem can be reformulated as an admissible sum problem by means of such a change of the index set K and starting table

x that to every k correspond a sufficiently large number of sub-locations (k,h) where numbers $x_{k,h} = x_k$ are located. When the solution to the problem has to be described it is however of no interest to know the sublocation numbers h, it is enough to know the number $\frac{w_k}{\varepsilon} = n_k$ of numbers x_{kh} located at some sublocation k.

As said already, we are interested in simple methods for finding $\check{P}(x)$ or $\hat{P}(x)$. Such methods have to avoid the use of more complicated operations than 1) sorting, classification, 2) tabulations, 3) summations including subtraction. It is desirable to come to the solution with a minimum number of such operations.

The basic idea is to make use of such simple decompositions [3/]

$$x_k = x_k' + x_k'', \quad K = K' \cup K'', \quad P = P' \cup P''$$

as repeatedly, with the purpose of partitioning the problem of finding $\check{P}(x)$ into a chain of simpler problems to find $\check{P}_1(x')$ for a similar assignment problem, where the number of sets $p \in P_1$ is relatively small and the table x' contains only a part of the numbers x_k in x, the remaining x_k' being zero.

In many problems K contains a number of subsets K' which are such that a fixed number of terms x_k are located in any given K' for every $p \in P$. In this case, the first operations used in practice to find $\check{P}(x)$ have the purpose of transforming the set x into a new "suitable" initial set x^* with the properties

---

3/ For an example see page 7, second paragraph below.

$$\check{P}(x^*) = \check{P}(x)$$

$$\min_{k \in K'} x_k^* = 0 \qquad \text{for every such subset } K' \text{ of } K$$

In the following we assume that x contains only non-negative numbers x_k and is a suitable initial set for finding $\check{P}(x)$.

In the cases when the number N_p of admissible sets p in P is small there is no need for an elaborate technique to find $P(x)$. We can make use of "brute force" methods of calculating $S_p(x)$ directly for all $p \in P$ and thereafter order the numbers $S_p(x)$ according to size.

In cases when the number N_p is large, for instance > 10 we can save many additions and much sorting time by means of the "refined brute force" method expressed in theorem [4] 2.

---

[4] Footnote added by T. C. Koopmans: Theorem numbers follow those of an earlier paper. For later reference, I quote here Theorem 1 from the earlier paper:

Theorem 1. If for every $p \in P$ a prescribed number N_n of terms in $S_p(x)$ must be located in the subset $K_n \subset K$, then the table $x' = x + z$, where $z_k = z_n$ for all $k \in K_n$, will have the same optimal sets as the table x. Furthermore if $x' = -x$, $\hat{p}' = \check{p}$, $\check{p}' = \hat{p}$.

Proof: This holds because $\sum_{k \in p} z_k$ is constant for every $p \in P$.

<u>Theorem 2.</u> If $S_p(x) = S_p(x') + S_p(x'')$, $S_p(x'') \geq 0$ for all $p \in P$

(note that this is satisfied if $x_k = x_k' + x_k''$ and $x_k'' \geq 0$)

and the different numbers $S_p(x')$ are ordered according to size

$$S_p(x') = S_{P_i}(x') \qquad \text{for all } p \in P_i \subset P$$

and

$$S_{P_o}(x') < S_{P_1}(x') < \dots < S_{P_i}(x') < \dots,$$

if further we denote

$$\underset{p \in P_o}{\text{Min}} S_p(x) = S_{P_o}(x') + \underset{p \in P_o}{\text{Min}} S_p(x'') = S_{\check{p}_o}(x)$$

and if i_o is the largest integer such that

$$S_{P_i}(x') \leq S_{\check{p}_o}(x) \qquad \text{for } i \leq i_o$$

then

$$\underset{p \in P}{\text{Min}} S_p(x) = \underset{i \leq i_o}{\text{Min}} (S_{P_i}(x) + \underset{p \in P_i}{\text{Min}} S_p(x'')) = \underset{p \in \bigcup_{i \leq i_o} P_i}{\text{Min}} S_p(x)$$

Proof: If p is an element of $(P - \bigcup_{i \leq i_o} P_i)$,

$$S_p(x) = S_{P_j}(x') + S_p(x'') \quad \text{for some } j > i_o \text{ and hence}$$

$$S_p(x) > S_{\check{p}_o}(x) \geq \underset{p \in P}{\text{Min}} S_p(x).$$

Hence the optimal set $\check{p} \in \check{P}(x)$ must be a subclass of $\bigcup_{i \leq i_o} P_i$, and $\check{p}$ itself an element of some P_i for $i \leq i_o$.

<u>Remark</u> No element $\check{p} \in \check{P}$ can be identical with any $p \in P_i$ for which

$$S_p(x'') > S_{\check{p}_o}(x) - S_{P_i}(x'')$$

or even more strongly with any element $p \in P_i$ such that

$$S_p(x'') > \underset{j < i}{\text{Min}} S_{\check{p}_j}(x) - S_{P_i}(x').$$

Hence, when we have to find $\check{P}(x)$ in the case when $S_{p_i}(x') \geqq 0$ and $S_{\check{p}_o}(x) \geqq 0$ are known it is thus unnecessary to calculate the numbers $S_p(x'')$ for those p for which already the term $S_{p_i}(x')$ is larger than $S_{\check{p}_o}(x)$. Thereafter we can eliminate all such $p \in P_i$ for which it can be proved that

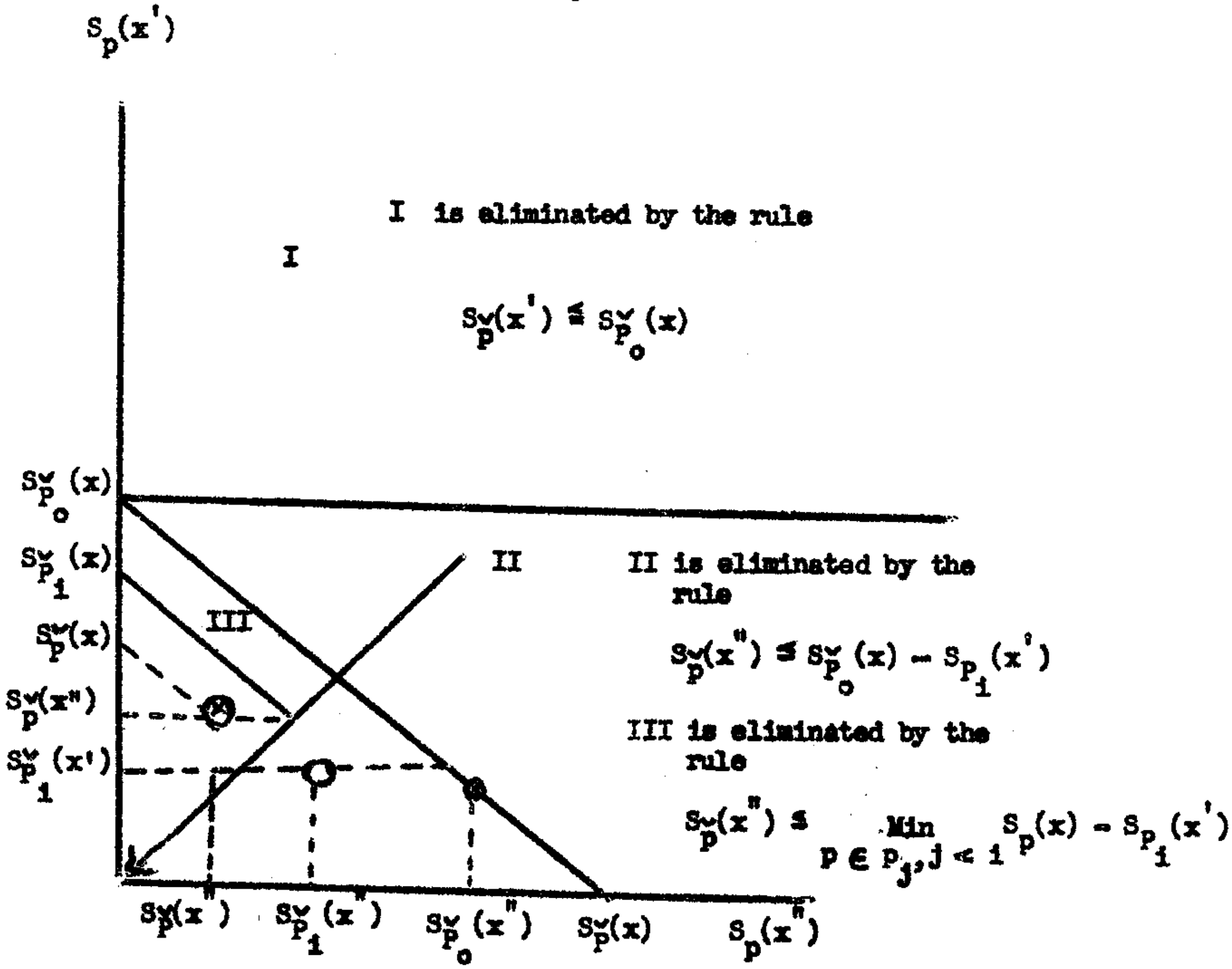
$$S_p(x'') > S_{\check{p}_o}(x) - S_{p_i}(x) .$$

If the smallest sum $S_p(x)$ in the union of the sets P_j for $j < i$ is known we can be sure that no $p \in P_i$ for which

$$S_p(x'') > \min_{j < i} S_{\check{p}_i}(x) - S_{p_i}(x')$$

is an element of $\check{P}(x)$.

Graphical representation of the procedure.



The same point can correspond to a whole set of elements $p \subset P$.

In the case when the numbers $S_{P_i}(x') + \underset{p \in P_i}{\text{Min}} S_p(x'')$ are pairwise different, $\check{P}(x) = \check{P}_i(x'')$ for some $i \leq i_o$.

If $\underset{i_o \geq i \geq o}{\text{Min}} S_{P_i}(x') + \underset{p \in P_i}{\text{Min}} S_p(x'')$ is attained for all $i \in I$, then the optimal set $\check{P}(x) = \bigcup_{i \in I} \check{P}_i(x'')$.

An especially good split $x = x' + x''$ for finding $P(x)$ with only a small amount of work to be done is the split defined by means of the formula

$$(1) \qquad x'' = 0, \quad x'_k = x_k \qquad \text{if} \quad x_k \geq x_{k_o}$$

$$(2) \qquad x''_k = x_k, \quad x' = 0 \qquad \text{if} \quad x_k < x_{k_o}$$

where x_{k_o} is chosen so large that it happens that $S_{p_o}(x') = 0$, but if x_{k_o} were to be reduced $S_p(x')$ would become > 0. All subsets P_i for which $S_p(x') \leq \underset{p \in p}{\text{Min}} S_p(x'')$ are then usually easily found. After the subsets P which can be seen to give a $\underset{p \in p}{\text{Min}} S_p(x'') > \underset{j < i}{\text{Min}} (\underset{p \in P_j}{\text{Min}} S_p(x'') - S_{P_i}(x') + S_{P_j}(x'))$ are excluded the optimal sets $\check{P}_i(x'')$ for the other P_i can be found for instance by "brute force."

If some class P_i contains a large number of admissible sets p the above technique can be used repeatedly for finding $\check{P}_i(x'')$.

Because the following simple mathematical facts about sums of two numbers sometimes can be useful for solving assignment problems we state them in theorems 3 and 4.

<u>Theorem 3.</u> If $P_o \subset P$ and $S_p(x) = S_p(x') + S_p(x'')$ for every $p \in P$

Then $S_{\check{p}_o}(x) = \underset{p \in P_o}{\text{Min}} S_p(x) \geqq \underset{p \in P}{\text{Min}} S_p(x) = S_{\check{p}}(x') + S_{\check{p}}(x'')$ and thus

$$S_{\check{p}}(x'') \leqq S_{\check{p}_o}(x) - S_{\check{p}}(x') \leqq S_{\check{p}_o}(x) - n_{k_o} x_{k_o}$$

where n_{k_o} denotes the number of terms in the sum $S_{\check{p}}(x')$, which are larger than equal to x_{k_o}.

<u>Theorem 4.</u> If $S_{\check{p}}(x) = S_{\check{p}}(x') + S_{\check{p}}(x'') \leqq S_{p_1}(x') + S_{p_1}(x'')$

then

either $S_{\check{p}}(x') \leqq S_{p_1}(x')$

or $S_{\check{p}}(x'') \leqq S_{p_1}(x'')$

The correctness of these theorems can be seen without formal proofs The usefulness of theorem 3 follows from the fact that we can be sure, that if $S_{\check{p}}(x')$ contains n_{k_o} or more numbers larger than x_{k_o} then $\check{p}$ is not an element of any subclass $P'' \subset P$ for which

$$\underset{p \in P''}{\text{Min}} S_p(x'') > S_{\check{p}_o}(x) - n_{k_o} x_{k_o}.$$

The fact expressed in theorem 4 can be utilised for instance for formulating the following efficient rule for finding the set $\check{P}(x)$ when the numbers $S_p(x')$ and $S_p(x'')$ are known but we do not like to make all the additions necessary for determining $s_p(x)$ for all $p \in P$. For the case when the numbers $S_p(x')$ and $S_p(x'')$ are punched on the same card the operations can be performed in light steps as follows:

1. Sort the cards according to increasing size of $S_p(x')$.

2. Take out the first group P_o of cards for which $S_p(x')$ is smallest.

3. Sort the cards in the group P_o according to increasing $S_p(x'')$. Denote the group of cards for which $S_p(x'')$ in this group are smallest by $\check{P}_o$, and the corresponding sum $S_p(x)$ by $S_{\check{P}_o}(x)$.

4. Calculate the differences $S_{\check{P}_o}(x) - S_{P_i}(x')$ for all different $S_p(x')$ for which this number is positive.

5. Punch red cards with the pair of numbers $S_{P_i}(x')$, $S_{\check{P}_o}(x) - S_{P_i}(x')$ and put them in the groups of cards for which $S_p(x')$ is the same. Take away all cards for which $S_p(x') > S_{P_o}(x)$.

6. Sort the groups P_i of cards according to increasing size of $S_p(x'')$. Take out the groups for which $S_p(x'')$ is at most equal to $S_p(x'')$ on corresponding red cards. From each subgroup with different $S_p(x'')$ take out the cards $\check{P}_i$ with the smallest $S_p(x')$.

7. Calculate the sums $S_{\check{P}_i}(x) = S_{\check{P}_i}(x') + S_{\check{P}_i}(x'')$ for these cards and take out the cards $\check{P}_{i>o}$ for which this sum is smallest.

8. The group of cards for which $S_p(x)$ is the smallest in the whole set of cards is then either equal to the group $\check{P}_o$ or $\check{P}_{i>o}$ depending

on which group has the smaller $S_p(x)$. If both groups $\check{P}_o$ and $\check{P}_{i>o}$ give the same sum $S_p(x)$ the union of $\check{P}_o$ and $\check{P}_{i>o}$ represent the optimal set $\check{P}(x)$. In some cases it happens that a red card is among $\check{P}_{i>o}$ and its original card in $\check{P}_o$. In this case the reproduced red card has to be removed from $\check{P}_o \cup \check{P}_{i>o}$.

This rule can naturally be used repeatedly if we like to avoid calculating for instance $\underset{p \in P_i}{\text{Min}}\ S_p(x'')$ without having first calculated $S_p(x'')$ for every p in P_i.

If we like to get mechanical rules for finding $\check{P}(x)$ such as this rule for finding $\check{P}(x)$ when $S_p(x) = S_p(x') + S_p(x'')$ in the case when $S_p(x')$ and $S_p(x'')$ are numerically known it seems to be necessary to specify the class of admissible sets P and thereby restrict the study to specific assignment problems, for instance, to the linear or quadratic assignment problems studied earlier by Beckmann and Koopmans.

One would hope that choices from among the methods based on the mathematical facts expressed in the theorems 1-4 could be made to construct mechanically applicable rules for different subclasses of assignment problems. However, before this can be done in a way which does not considerably reduce the possibility of using such short cuts that can be used in special cases, it would be very useful to have a set of practical examples solved by mathematicians able to make use of those "degrees of freedom," which are not yet fixed uniquely.

The following examples show that linear and quadratic assignment problems can easily be solved in practice by means of the methods discussed.

Example 1. A linear assignment problem.

Find the permutation $\check{p} = (\check{p}(1), \check{p}(2) \ldots \check{p}(n))$ such that

$$S_{\check{p}} = \text{Min} \sum_{i=1}^{n} x_{ip(i)}.$$

In this case we first subtract from the numbers $\{x_{ij}\}$ the smallest number $\underset{j}{\mathrm{Min}}\ x_{ij}$ in each row and if the table $\left\{x'_{ij} = x_{ij} - \underset{j}{\mathrm{Min}}\ x_{ij}\right\}$ does not already have as the smallest number in each column the number 0 we subtract $\underset{i}{\mathrm{Min}}\ x'_{ij}$ from the numbers x'_{ij}. In this way we get a table x''_{ij} with at least one zero in every column and in every row. If there exists one and only one zero in every row and column these zeros are located at the optimal permutation $\check{p} = \overset{n}{\underset{i=1}{\{}} i, p(i) \}$. If not we let K_o denote the set of locations for which $x''_{ij} < x''_{i_o j_o}$ and chose $x''_{i_o j_o}$ so large that from K_o can be found at least one permutation. Only in more rare cases $S_{\check{p}}$ contains terms from $K - K_o$. Sometimes it happens that a term of the same size as $x''_{i_o j_o}$ is included for some $\check{p} \in \check{P}(x)$. It is already an extremely difficult task to find a matrix x'' which is such that more than one term in $S_{\check{p}}(x'')$ is larger than $x''_{i_o j_o}$.

If we like we can try to increase the probability that the starting table will be such that $S_p(x^*) = 0$ by adding to a row i_1 in x'' which contains more than one zero the "largest" next smallest number in the columns j for which $x''_{i_1 j} = 0$. Thereafter we can subtract the smallest number in each column j for which $x_{i_1 j} = 0$ from all other numbers in these columns. By increasing the difference between the sums of numbers subtracted from all elements of a row or column and the sum of numbers added to each row or column we can decrease the sums $S_p(x^*)$ until we get the $\underset{p \in P}{\mathrm{Min}}\ S_p(x^*) = 0$. It is however uncertain if it is worth while to use such transformations before the problem is solved by "refined brute force."

Example 2.

Numerical example of a linear assignment problem.

$x =$

							Min
30	15	21	29	27	26	11	11
28	20	11	19	21	27	16	11
18	25	13	18	6	15	14	6
11	19	16	7	14	13	1	1
10	15	7	8	11	9	6	6
7	18	14	13	6	5	4	4
5	14	22	4	9	10	7	4

$x' =$

							Min.
19	4	10	18	16	15	0	0
17	9	0	8	10	16	5	0
12	19	7	12	0	9	8	0
10	18	15	6	13	12	0	0
4	9	1	7	5	3	0	0
3	14	10	9	2	1	0	0
1	10	18	0	5	0	3	0

$x'' =$

18	**0**	10	18	16	14	0	0
16	5	**0**	8	10	15	5	0
11	15	7	12	**0**	8	8	0
9	14	15	**6**	13	11	0	0
3	5	1	7	5	2	**0**	0
2	10	10	9	2	**0**	0	0
0	6	18	0	5	5	3	0
Min: 0	0	0	0	0	0	0	

x^{III}

						add 6 ↓	Min
18	0	10	18	16	14	6	0
16	5	0	8	10	15	11	0
11	15	7	12	0	8	13	0
9	14	15	6	13	11	6	0
3	5	1	7	5	2	6	1
2	10	10	9	2	0	6	0
0	6	18	0	5	5	9	0
0	0	0	0	0	0	6	

x^{IV}

18	[0]	10	18	16	14	6	0
16	5	0	8	10	15	11	0
11	15	7	12	0	8	13	0
3	8	9	[0]	7	5	0	0
2	4	0	6	4	1	5	0
2	10	10	9	2	[0]	6	0
[0]	6	18	0	5	5	9	0
0	0	0	0	0	0	0	

x^{V}

18	0	15	18	16	14	6	0
16	5	5	8	10	15	11	5
11	15	12	12	0	8	13	0
3	8	14	0	7	5	0	0
2	4	5	6	4	1	5	1
2	10	15	9	2	0	6	0
0	6	23	0	5	5	9	0
0	0	5	0	0	0	0	

x^{VI}

					add 2 ↓		
18	0	15	18	16	14	6	0
11	0	0	3	5	10	6	0
11	15	12	12	0	8	13	0
3	8	14	0	7	5	0	0
1	3	4	5	3	0	4	0 ← subtract 1
2	10	15	9	2	0	6	0 ← subtract 2
0	6	23	0	5	5	9	0
0	0	0	0	0	0	0	

x^{VI}

18	[0]	15	18	16	16	6
11	0	[0]	3	5	12	6
11	15	12	12	[0]	10	13
3	8	14	0	7	7	[0]
[0]	2	3	4	3	[0]	3
[0]	8	13	7	2	[0]	4
0	6	23	[0]	5	7	9

there exist two solutions

Solved by using only theorem 1 repeatedly.

Example 3[5/]

Application to Dantzig's numerical example of a transportation problem in which the requirements are integral numbers.

Reference: "Activity Analysis ... "Chapter XXIII, p. 367

5/ Reconstructed from my recollection of Mr. Törnqvist's oral presentation. T.C.K.

Figures in parenthesis express the number of times each column or row is thought to be written out, so that the matrix is always square.

x	(13)	(3)	(3)	(3)	(2)	(2)
	(1)	3	2	1	2	3
	(5)	5	4	3	-1	1
	(7)	0	2	3	4	5

x'		(3)	(3)	(3)	(2)	(2)
	(1)	3	0	0	3	2
	(5)	5	2	2	0	0
	(7)	0	0	2	5	4

x' written in full (this need not be done in actual application)

						subtract 2							
3	3	3	0	0	0	0	0	0	3	3	2	2	add 2
5	5	5	2	2	2	2	2	2	0	0	0	0	
5	5	5	2	2	2	2	2	2	0	0	0	0	
5	5	5	2	2	2	2	2	2	0	0	0	0	
5	5	5	2	2	2	2	2	2	0	0	0	0	
5	5	5	2	2	2	2	2	2	0	0	0	0	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	
0	0	0	0	0	0	2	2	2	5	5	4	4	

x''		(3)	(3)	(1)	(2)	(2)	(2)
	(1)	5	2	0	[0]	5	3
	(1)	5	2	0	[0]	0	0
	(4)	5	2	0	0	[0]	[0]
	(1)	0	0	[0]	0	5	4
	(6)	[0]	[0]	0	0	5	4

Any permutation taken from the boxed in squares represents the solution. Other possible permutations containing only zeros present the same incidence over groups of identical rows or columns, and thus represent the same transportation program.

We will numerically solve a quadratic assignment problem.

<u>Example 4.</u> The table x we assume to be as follows

$x =$

i \ j	λ \ ρ	1				2				3				4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	1	1	-	-	-	·	·	·	·	-	·	·	·	-	·	·	·
	2	-	0	-	-	·	-	·	·	·	-	·	·	·	-	·	·
	3	-	-	2	-	·	·	-	·	·	·	-	·	·	·	-	·
	4	-	-	-	[3]	·	·	·	-	·	·	·	-	·	·	·	-
2	1	-	5	3	[2]	[0]	-	-	-	-	·	·	·	-	·	·	·
	2	3	-	4	1	-	6	·	-	·	-	·	-	·	-	·	·
	3	2	0	-	7	-	-	7	-	·	·	-	·	·	·	-	·
	4	1	9	6	-	-	-	-	8	·	·	·	-	·	·	·	-
3	1	-	7	8	9	-	0	1	4	9	-	-	-	-	·	·	·
	2	3	-	6	3	6	-	5	3	-	6	-	-	·	-	·	·
	3	1	6	-	[4]	[1]	6	-	7	-	-	[0]	·	·	·	-	·
	4	0	5	7	-	4	3	1	-	-	-	-	3	·	·		-
4	1	-	7	8	1	-	1	8	7	-	6	1	0	4	-	-	-
	2	0	-	6	[3]	[3]	-	3	5	7	-	[1]	5	-	[5]	-	-
	3	1	5	-	4	5	6	-	2	8	1	-	3	-	-	0	-
	4	7	8	9	-	7	0	1	-	6	7	8	-	-	-	-	2

$= \{x_{ij\lambda\rho}\}$

The problem is to find that permutation $p = (1 \longrightarrow p(1),\ 2 \longrightarrow p(2),\ 3 \longrightarrow p(3),\ 4 \longrightarrow p(4))$ which gives the minimisum.

$$\sum_{i=1}^{4} \sum_{j=1}^{4} x_{ij\,p(i)\,p(j)} = S_p(x); \quad \sum_{j=1}^{4} \sum_{i=1}^{4} x_{ij\check{p}(i)\check{p}(j)} = S_{\check{p}}(x)$$

The "forbidden" numbers are denoted by $-$. To simplify the problem we assume that the numbers $x_{ij\lambda\rho} = x_{ij\lambda\rho} + x_{ji\rho\lambda}$, for $j < i$, and $x_{ij\lambda\rho} = 0$ for $j > i$. This simplification can be done because $x_{ij\lambda\rho}$ and $x_{ji\rho\lambda}$ will either not at all or both be included in $S_p(x)$, so we can assume that this preliminary modification of the table x is already done. We also have subtracted the smallest term in each subtable x_{ij} from $x_{ij\lambda\rho}$, giving the zeros in the table (Theorem 1). Because $x_{iip(i)p(i)}$ and $x_{jjp(j)p(j)}$ will be included in $S_p(x)$ if $x_{ijp(i)p(j)}$ is included we can further reduce the table by adding these terms to the corresponding terms in the subtables $(i = 2,\ j = 1)$ and $(i = 4,\ j = 3)$. The relevant information is now summarized in table x'.

$x' =$

i													
1	1												
	2												
	3												
2	1	-	5	5	5								
	2	10	-	12	10								
	3	10	7	-	17								
	4	10	17	16	-								
3	1	-	7	8	9	-	0	1	4				
	2	3	-	6	3	6	-	5	3				
	3	1	6	-	4	1	5	-	7				
	4	0	5	7	-	4	3	1	-				
4	1	-	7	8	1	-	1	8	7	-	16	5	7
	2	0	-	6	3	3	-	3	5	15	-	6	13
	3	1	5	-	4	5	6	-	2	17	7	-	6
	4	7	8	9	-	7	0	7	-	17	15	10	-

Now we can again subtract the smallest term (5) in subtable (i=2, j=1) and the smallest term (5) in the subtable (i=4, j=3) from the other numbers in these subtables (Theorem 1). We get then the table x''.

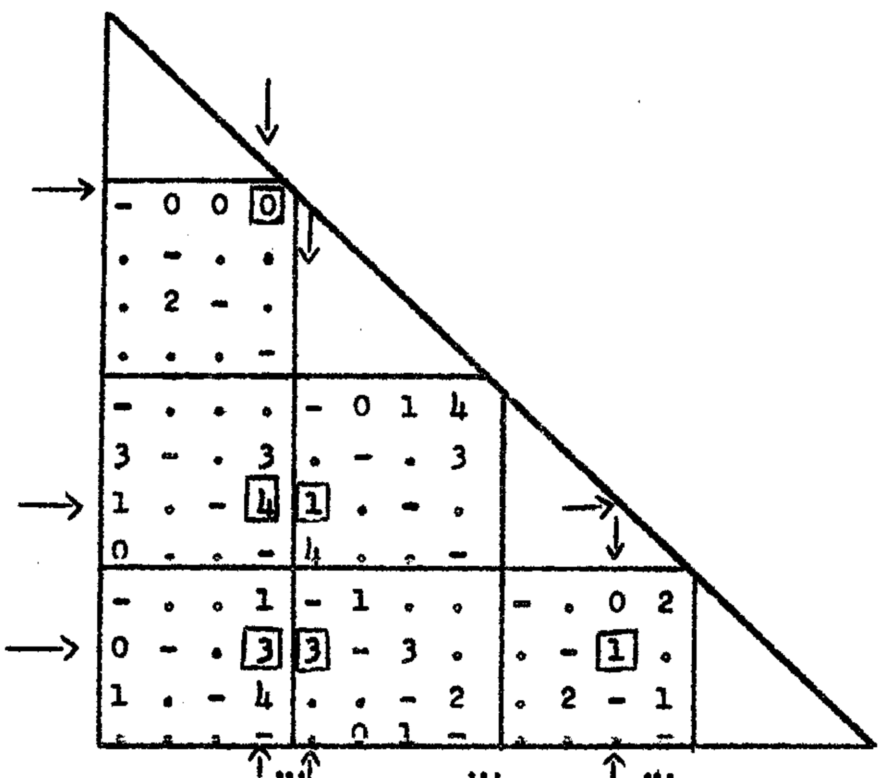
$x'' =$

i													
1	1												
	2												
	3												
	4												
2	1	-	0	0	0								
	2	5	-	7	5								
	3	5	2	-	12								
	4	5	12	11	-								
3	1	-	7	8	9	-	0	1	4				
	2	3	-	6	3	6	-	5	3				
	3	1	6	-	4	1	5	-	7				
	4	0	5	7	-	4	3	1	-				
4	1	-	7	8	1	-	1	8	7	-	11	0	2
	2	0	-	6	3	3	-	3	5	10	-	1	8
	3	1	5	-	4	5	6	-	2	12	2	-	1
	4	7	8	9	-	7	0	1	-	12	10	5	-

The terms in the sum x_p which are admissible are six numbers from this table one from every subtable (2,1), (3,1), (4,1) and one from each of the subtables (3,2), (4,2), and one from (4,3). The six numbers are located in the intersections of lines reflected at the main diagonal. All intersection points must be admissible, that is, they cannot lie on a diagonal where the terms are (-).

If we now fill out a new table by taking the numbers from x'' in order of size and stop the process when we find an admissible summation way[5] we get the table:

$x''' =$



In this table the numbers $x'''_{2114} = 0$, $x'''_{3134} = 4$, $x'''_{4124} = 3$, $x'''_{3231} = 1$,

---

5/ To see that the process cannot be stopped before the numbers equal to 4 are entered, we can try to construct a similar network of lines reflected at the main diagonal. If the numbers 4 are forbidden no such network with already entered numbers as intersection points can be found.

$x'''_{4221} = 3$, $x'''_{4323} = 1$ give a sum = 12 corresponding to the permutation 1 →4, 2 →1, 3 →3, 4 →2. This minimisum is so small that it hardly can be beaten if we let also terms larger than 4 be included in x_p.

To see if any other promising ways exist (Theorem 3) we can study which is the smallest admissible partial sum of at most five terms that can be found from x'''. These sum must be smaller or equal to 12 - 5 - 7 if it contains 5 terms and $\leq 12 - 10 = 2$ if it contains 4 admissible terms if the corresponding sum shall be ≤ 12. We see without difficulty that no such partial sums exist. The permutation p = (1 →4, 2 →1, 3 →3, 4 →2) gives thus the minimum. The corresponding minimum in the table x is <u>12 + 10 = 22</u>. In this case we did not need theorem 2 to solve the problem. By means of theorem 2 we can find the minimum sum, for instance, as follows:

Taking as the first term in x''_p a term from the subtable (i = 2, j = 1) we get:

$$x''_{p\nu} = \text{Min} \left\{ s_0,\ 2 + s_2,\ 5 + s_5,\ 7 + s_7,\ 11 + s_{11},\ 12 + s_{12} \right. = 12$$

$$s_0 = \text{Min} \begin{Bmatrix} \beta_{01} = \text{Min}\ (\underline{4 + 1 + 3 + 3 + 1},\ 3 + 6 + 4 + 5 + 2) = 12 \\ \beta_{02} = \text{Min}\ (6 + 6 + 9 + \ldots,\ 7 + 4 + 6 + \ldots) > 17 \\ \beta_{03} = \text{Min}\ (6 + 1 + 8 + 7 + 5,\ 5 + 4 + 5 + 5 + 1) > 17 \end{Bmatrix} = 12$$

$$2 + s_2 = +2 + \beta_{21} = 2 + \text{Min}\ (7 + 1 + 8 + \ldots,\ 5 + 1 + 7 + 8) > 17$$

$$5 + s_5 = 5 + \text{Min} \begin{cases} \beta_{51} = \text{Min}\ (1 + 5 + 7 + \ldots,\ \underline{0 + 3 + 1 + 6 + 1}) = 16 \\ \beta_{52} = \text{Min}\ (3 + 5 + 7 + \ldots,\ 0 + 1 + 0 + 3 + 8) = 17 \\ \beta_{53} = \text{Min}\ (\underline{3 + 3 + 1 + 2 + 2},\ 1 + 7 + 0 + 5 + 1) = 16 \\ \beta_{54} = \text{Min}\ (9 + 0 + 4 + 6 + \ldots,\ \underline{4 + 5 + 1 + 1 + 0}) = 16 \end{cases}$$

$$7 + s_7 = 7 + \beta_{71} = 7 + \text{Min}\,(8 + 0 + 9 + 0 + 12,\ 7 + 3 + 8 + 1 + 2) = 28$$

$$11 + s_{11} = 11 + s_{11,1} = 11 + \text{Min}\,(8 + 4 + \quad\quad ,\ 6 + 3 \quad\quad\quad) > 17$$

$$12 + s_{12} = 12 + \text{Min}\left\{\begin{matrix} \beta_{12,1} = (7 + \ldots \quad\quad , 6 + \quad\quad\quad) \\ \beta_{12,2} = (9 + \ldots \quad\quad , 3 + 3 + \quad\quad) \end{matrix}\right\} > 17$$

The minimum sum is thus 12 = 0 + 4 + 1 + 3 + 3 + 1. There are in all three next smallest sums = 16.

REFERENCES

[1] von Neumann, J., "A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem," Contributions to the Theory of Games, II, 1953, H. W. Kuhn and A. W. Tucker, eds., pp. 5-12.

[2] Beckmann, M. and Koopmans, T. C., "A Note on the Optimal Assignment Problem," Cowles Commission Discussion Paper, Economics 2053.

[2] Beckmann, M. and Koopmans, T. C., "On Some Assignment Problems," Cowles Commission Discussion Paper, Economics 2071

[3] Motzkin, T.,In a communication to T. C. Koopmans.