The Assignment Problem

We have a table (an A matrix), e.g. the following (we have n rows and n columns, now n=6):

1	3	5	7	9	2
2	3	4	1	5	8
1	2	2	4	5	6
1	1	3	1	1	1
1	4	5	8	7	9
1	2	3	5	5	5

The (linear) assignment problem is as follows: *Choose exactly one number in each row and in each column (i.e., and assignment), so that their sum is as small as possible. (This is the **minimum version** of the assignment problem.)*

Definition

- **assignment**: collection of numbers, so that there is exactly one number in each row and each column
- **value of the assignment**: the sum of the chosen numbers

<u>Note</u>: in the **maximum** version we look for an assignment with maximum sum of the numbers. This can be derived from the minimum version and vice versa. But we will deal only with the minimum version. (As the other one is similar.) There are many applications.

We solve the assignment problem with the Hungarian Method. But first, some preprocessing comes:

Preprocessing for the <u>rows</u>: Let us reduce the numbers in each row, by the minimum of the row. (After this preprocessing step, we will have at least on zero in each row.)

From this:

1	3	5	7	9	2	1
2	3	4	1	5	8	1
1	2	2	4	5	6	1
1	1	3	1	1	1	1
1	4	5	8	7	9	1
1	2	3	5	5	5	1

We get this one:

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

Claim: Performing the previous preprocessing step, the value of any assignment decreases by the same amount.

<u>Proof</u>: Let the row-minimums be denoted as u(i) for i=1,…,n. Then considering any assignment, if the value of the assignment (i.e. the sum of the numbers in the assignment) is S *before* the preprocessing step, then *after* the preprocessing step it is S-(u(1)+u(2)+...+u(n)). So the difference is u=u(1)+u(2)+...+u(n) for each assignment between the "old" value and the "new" value. (Thus, if an assignment is better than another assignment before the reduction, it will remain better also after the reduction.)

Corollary: The above preprocessing step changes the A matrix, let us suppose that the new matrix is A'. Then if an assignment is optimal in case of the A matrix, then it is also optimal in case of the A' matrix.

After this, let us make a similar **preprocessing for the columns**, i.e. let us reduce each column with the column-minimum.

From this:

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

We will get:

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

Now we have a transformed matrix (we can denote it again by A for simplicity). Here we already do have at least one zero in each row and each column. The original numbers are changed. But if we find an optimal assignment in the present matrix, then the same assignment (i.e. what numbers should be chosen) will be optimal also in the original one.

Of course, if we could find a zero in each row and each column, we would be done (as their sum is as small as possible). Can we do that? (Sometimes yes and sometimes not.)

Let us see:

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

- There is only a unique zero in the first row (denoted by bold number).
- Also there is only one zero in the last row. (And also in the 5-th row.)

Unfortunately, these two aforementioned zeros are in the same column.
So: we are not able to choose a zero from each row. But, it is not a problem at all, as we will do some further transformations.

Some definitions:

- two zeros are independent if they are in different rows and columns
- several zeros are independent, if any two of them are independent
- a line that goes through some row or column, is called a “covering line”

Theorem (Kőnig Dénes - Egerváry Jenő). The maximum number of independent zeros equals to the minimum number of covering lines, that cover all zeros in the table.

Proof: $\leq$: trivial, by definition
$\geq$: later.

Let us find the maximum number of independent zeros in the table.

- From the first row, there is only one option to choose. Let us choose this zero. (Then its row and column excluded from consideration, i.e. we make a reduction).
- Then there is only one option in the second row, let us choose this one.
- Remained only one choice in the third row, let us choose this one.
- There is one choice in the second column
- No more choices. We have found 4 independent zeros. Can we cover **all** zeros by 4 lines? (If yes, then 4 is the maximum number of independent zeros.)

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

- For the zero in the first row, let us cover its column.
- For the zero in the second row, let us cover its column
- For the zero in the third row, let us cover its column (or row)
- For the zero in the fourth row, let us cover its row

We get:

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

Now, we can see that

- there are 4 independent zeros, and not more, because
- all zeros can be covered by 4 lines

After this: **Hungarian Method**

1. Let us cover all zeros with (possibly) minimum number of lines. If the number of lines is n (the number of rows and columns), then we do have n independent zeros, STOP. Go back to the first tableau, choose the same assignment, which is optimal.
2. (Otherwise, i.e. we have less than n covering lines), let epsilon be the smallest uncovered number. Let us perform the next Transformation:
   - We reduce each uncovered number by epsilon
   - We add epsilon to each number that is covered twice
   - (nothing is made with the other numbers, that are covered only once).
3. Go to Step 1.

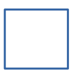
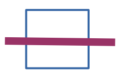
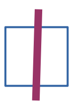
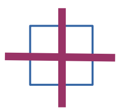
<u>Claim</u>: The optimal solutions remain optimal during the transformation.
Proof: the transformation can be made by whole rows and columns as follows:

Transformation':

- in each **not covered** row we reduce each number by epsilon, and
- in each **covered** column we add epsilon to each number

<u>Claim</u>: the results of Transformation and Transformation' are the same.

Proof: Let us see:

	Transformation	Transformation'
	-eps	-eps
	----	-----
	----	-eps + eps
	+ eps	+ eps

Let us see the transformation. Now **eps=1.**
From this:

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

We get this:

0	1	2	6	7	**0**
1	1	1	**0**	3	6
1	1	**0**	4	4	5
1	0	1	1	**0**	0
0	2	2	7	5	7
0	**0**	0	4	3	3

We already do have n independent zeros, we are (almost done). We go back to the original tableau and choose the same assignment:

1	3	5	7	9	**2**
2	3	4	**1**	5	8
1	2	**2**	4	5	6
1	1	3	1	**1**	1
1	4	5	8	7	9
1	**2**	3	5	5	5

Thus, **min= 9.**

Reading: Harold W. Kuhn, A tale of three eras: The discovery and rediscovery of the Hungarian Method, European Journal of Operational Research, Volume 219, Issue 3, 2012, Pages 641-651, https://doi.org/10.1016/j.ejor.2011.11.008.
(https://www.sciencedirect.com/science/article/pii/S0377221711009957)

Abstract: In the Fall of 1953, a translation of a paper of Jenő Egerváry from Hungarian into English combined with a result of Dénes Kőnig provided the basis of a good algorithm for the Linear

Assignment Problem. To honor the Hungarian mathematicians whose ideas had been used, it was called the Hungarian Method. In 2005, Francois Ollivier discovered that the posthumous papers of Jacobi contain an algorithm that, when examined carefully, is essentially identical to the Hungarian Method. Since Jacobi died in 1851, this work was done over a hundred years prior to the publication of the Hungarian Method in 1955. This paper will provide an account for the mathematical, academic, social and political worlds of Jacobi, Kőnig, Egerváry, and Kuhn. As sharply different as they were (Prussian monarchy, Hungary under the Nazis and the Communists, and the post-war USA), they produced the same mathematical result. The paper is self-contained, assuming little beyond the duality theory of linear programing. The Hungarian Method and Jacobi's algorithm will be explained at an elementary level and will be illustrated by an example, solved both by the Hungarian Method and by Jacobi's Method.
Keywords: Assignment; Combinatorial optimization; Graph theory; Linear programing; Hungarian Method