The Assignment Problem – AJ's Method

Suppose there are n jobs and n persons to fill these jobs. People fit some jobs better than other jobs. Thus, each person has a rating for each jobs. In other words, Person A has a rating to fill Job 1, and a rating to fill Job 2. Person B has a rating to fill Job 1, and a rating to fill Job 2. If there are three people and three jobs, there would be nine ratings.

The *lower* the rating, the better it is. These ratings are **relative** – they provide an ordering of preference. The numerical values themselves indicate nothing, and a rating of "2" is *not* twice as bad as a rating of "1."

Our goal: assign each person to exactly one job in such a way as to minimize the total of ratings.

The n x n **ratings matrix** $C = (c_{ij})$ is where c_{ij} is the rating of the person i for the job j.

The n x n **assignment matrix** $X = (x_{ij})$ is such that

$$x_{ij} = \begin{cases} 1, \text{if person } i \text{ is assigned to job } j \\ 0, \text{otherwise} \end{cases}$$

The assignment problem is

$$(A) = \begin{cases} \text{minimize} \quad \propto (\mathrm{X}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \textit{subject to} \quad \sum_{j=1}^{n} x_{ij} = 1, i = 1, \dots, n \\ \qquad \sum_{i=1}^{n} x_{ij} = 1, i = j, \dots, n \\ \qquad x_{ij} \geq 0, \forall i, j \\ \qquad x_{ij} = 0 \text{ or } 1 \; \forall i, j \end{cases}$$

The dual of Problem (B) is

$$(B) = \begin{cases} \text{maximize} \quad \beta(\mathrm{U}, \mathrm{V}) = \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j \\ \textit{subject to} \qquad u_i \text{ free}, i = 1, \dots, n \\ \qquad v_j \text{ free}, j = 1, \dots, n \\ \qquad u_j + v_j \leq c_{ij}, i, j = 1, \dots, n \end{cases}$$

It follows from the complementary slackness theorem (as applied to problems of the forms *A* and *B*) that X^* is optimal for *A* if and only if there exist U^* and V^* such that X^*, U^*, V^* satisfy

1. $X^*$is feasible for (A).
2. U^*, V^* are feasible for (B). That is, $\overline{c_{ij}} \geq 0 \; \forall i, j$
3. $\overline{c_{ij}} x_{ij}^* = 0 \; \forall i, j$ where $\overline{c_{ij}} = c_{ij} - u_i^* - v_j^*$

I will present an alternate to the Hungarian Method to solve this problem.

AJ's Method - Given C, an n x n matrix:

2	8	5	1	10
5	9	9	3	5
6	6	2	8	2
2	6	3	8	7
2	5	3	4	3

1. Row by row, from left to right, circle the minimum value. This represents an "assignment." There should only be one circled value for each row. In case of a tie, choose and circle the left most value.

2	8	5	1	10
5	9	9	3	5
6	6	2	8	2
2	6	3	8	7
2	5	3	4	3

The orange cells will represent the minimum values for each row.

2. A) If all columns have a circled value, we have an optimal solution.
B) If there are columns without a circled value, label those columns with a "u." These are **unsaturated** columns.

	u			u
2	8	5	1	10
5	9	9	3	5
6	6	2	8	2
2	6	3	8	7
2	5	3	4	3

3. Make a new table with as many columns as there are unsaturated columns. The cells in this new table will be determined by the absolute value difference between the row minimum and the respective unsaturated column value. The new table is shown in blue.

	u			u		
2	8	5	1	10	|1-8|=7	|1-10|=9
5	9	9	3	5	|3-9|=6	|3-5|=2
6	6	2	8	2	|2-6|=4	|2-2|=0
2	6	3	8	7	|2-6|=4	|2-7|=5
2	5	3	4	3	|2-5|=3	|2-3|=1

4. In our original table, find all columns that have only one assignment. This signifies an assignment that is optimal. We may exclude the row in which the optimal assignment is located from our table of differences.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

The excluded row is now shown in grey. We will not consider this row in further steps.

5. Go column by column through the difference table and find the minimum value.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

This minimum value represents the penalty we must "pay" to reassign our selection in row 5 from column 1 to column 2. This reassignment will result in an optimal assignment in column 1, as the only assignment for column 1 will be in row 4. We can then exclude row 4 from our difference table for this reason.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

For the same reasoning, we can exclude the rest of row 5 from our difference table.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

Now we can proceed to the next column in our difference table and find the minimum value there.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

This step is to be repeated for as many unsaturated columns your table may have.

6. Using the minimum values, proper reassignments may be made. The initial assignment in row 2 will change from column 4 to column 5, since |3-5|=2.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

Likewise, the initial assignment in row 5 will change from column 1 to column 2 since |2-5|=3.

	u			u		
2	8	5	1	10	7	9
5	9	9	3	5	6	2
6	6	2	8	2	4	0
2	6	3	8	7	4	5
2	5	3	4	3	3	1

An optimal, non-unique solution should now result.

2	8	5	1	10
5	9	9	3	5
6	6	2	8	2
2	6	3	8	7
2	5	3	4	3

2+5+2+1+5=15

The same solution was achieved using the brute-force CheckAssignmentSolution program.