MSIS 638

Case 2.3d

Jia Liang Ma

1. Search for “assignment problem” on the internet. Describe what the assignment problem is in your own words.

The assignment problem is a combinatorial optimization problem. A solution to perform as many tasks as possible by distributing the tasks that every agent has one tasks at least and at most. It looks a bit similar as the TSP problem in the previse class. For assignment problem, the difference is the methodology consists of finding, in a weighted bipartite graph, a matching of a given size, in which the sum of weights of the edges is a minimum. In other words, the fundamental of this problem is trying to find the margin effect of the case we research. The objective is to minimize the total relocation cost.

Reference:

<https://en.wikipedia.org/wiki/Assignment_problem>

<https://www.youtube.com/watch?v=I_1qPoD67bA>

1. Watch the following video describing the assignment problem and an efficient algorithm to solve it.

Hungarian algorithm

Step 1: In the m x m cost matrix, find the minimum cost in each row. Subtract each cost by the minimum cost in its row. In the new matrix, find the minimum cost in each column. Subtract each cost by the minimum cost in its column.

Step 2: Draw the min number of (row and column) lines to cover all 0s. If number of lines = m, an optimal solution is available among the covered zeros. If number of lines < m, go to step 3.

Step 3: Find the smallest nonzero element (k) that is not covered by the lines drawn in step 2. Subtract k from each element that is not covered and add k to each element that is covered by two lines. Return to step 2.

1. Apply the Hungarian algorithm explained in the video to the following assignment problem example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **OR1** | **OR2** | **OR3** | **OR4** | Row min |
| Minus row min | **Patient 1** | 4 | 6 | 2 | 8 | 2 |
|  | **Patient 2** | 3 | 5 | 7 | 3 | 3 |
|  | **Patient 3** | 6 | 2 | 5 | 4 | 2 |
|  | **Patient 4** | 4 | 7 | 3 | 6 | 3 |

Subtracting the minimum numbers in each row, and minus to all matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Minus col min |  |  |  |  |  |
|  |  | **OR1** | **OR2** | **OR3** | **OR4** |
|  | **Patient 1** | 2 | 4 | 0 | 6 |
|  | **Patient 2** | 0 | 2 | 4 | 0 |
|  | **Patient 3** | 4 | 0 | 3 | 2 |
|  | **Patient 4** | 1 | 4 | 0 | 3 |
|  | Col Min | 0 | 0 | 0 | 0 |

Subtracting the minimum numbers in each column, and minus to all matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** | 2 | 4 | 0 | 6 |
| **Patient 2** | 0 | 2 | 4 | 0 |
| **Patient 3** | 4 | 0 | 3 | 2 |
| **Patient 4** | 1 | 4 | 0 | 3 |

Finding the lines to cover all the 0s. In this case, the total lines are smaller than m, then we go to step 3. (minus the matrix with the minimum num: 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** | 1 | 3 | 0 | 5 |
| **Patient 2** | 0 | 2 | 5 | 0 |
| **Patient 3** | 4 | 0 | 4 | 2 |
| **Patient 4** | 0 | 3 | 0 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** |  |  | 0 |  |
| **Patient 2** | 0 |  |  | 0 |
| **Patient 3** |  | 0 |  |  |
| **Patient 4** | 0 |  | 0 |  |

After step 3, pointing all the 0s’ out.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** |  |  | 1 |  |
| **Patient 2** | 1 |  |  | 1 |
| **Patient 3** |  | 1 |  |  |
| **Patient 4** | 1 |  | 1 |  |

K=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** |  |  | 1 |  |
| **Patient 2** | 0 |  |  | 1 |
| **Patient 3** |  | 1 |  |  |
| **Patient 4** | 1 |  | 0 |  |

Turning the repeated zone of 1s into 0.

The optimal solution for this case is x13+x24+x32+x41= z

Z= 2+3+2+4 (compared the chart above with original chart’s spot) = 11

11\*1000(per OR) = 11,000

1. Conduct sensitivity analysis on at least two parameters of the model and investigate its impact on the optimal solution (Hint: increase and decrease two parameters and resolve the model to see if it impacts the optimal assignment. Remember to make *one* change at a time).

Parameters: Patients, Operation rooms ($1000 / per room)

Let took the first column as the variation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **OR1** | **OR2** | **OR3** | **OR4** | Row min |
| Minus row min | **Patient 1** | 7 | 6 | 2 | 8 | 2 |
|  | **Patient 2** | 3 | 5 | 7 | 3 | 3 |
|  | **Patient 3** | 1 | 2 | 5 | 4 | 1 |
|  | **Patient 4** | 5 | 7 | 3 | 6 | 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Minus col min |  |  |  |  |  |
|  |  | **OR1** | **OR2** | **OR3** | **OR4** |
|  | **Patient 1** | 5 | 4 | 0 | 6 |
|  | **Patient 2** | 0 | 2 | 4 | 0 |
|  | **Patient 3** | 0 | 1 | 4 | 3 |
|  | **Patient 4** | 2 | 4 | 0 | 3 |
|  | Col Min | 0 | 1 | 0 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | **OR1** | **OR2** | **OR3** | **OR4** |
|  | **Patient 1** | 5 | 3 | 0 | 6 |
|  | **Patient 2** | 0 | 1 | 4 | 0 |
|  | **Patient 3** | 0 | 0 | 4 | 3 |
|  | **Patient 4** | 2 | 3 | 0 | 3 |

Lines>m

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OR1** | **OR2** | **OR3** | **OR4** |
| **Patient 1** |  |  | 0 |  |
| **Patient 2** | 0 |  |  | 3 |
| **Patient 3** | 3 | 0 |  |  |
| **Patient 4** |  |  | 3 |  |

According to the analysis, z= 6+3+3 =12

OR fee will increase for $1000. With different array, the optimal solution might be different in this case.

A hospital needs to assign four patients to four operating rooms. Depending on the equipment in each operating room, the cost of assigning each patient to each operating room (in $1000) is given in the following table. Use the Hungarian algorithm to find the optimal assignment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | OR1 | OR2 | OR3 | OR4 |
| Patient 1 | 4 | 6 | 2 | 8 |
| Patient 2 | 3 | 5 | 7 | 3 |
| Patient 3 | 6 | 2 | 5 | 4 |
| Patient 4 | 4 | 7 | 3 | 6 |