B609 Project Report

Implementation of Hungarian Algorithm for Job Assignment

Group members

Giridhar Gomatom (ggomatom@indiana.edu)
Saandeep Jogiparti (saanjogi@indiana.edu)

Under the guidance of
Prof. Haixu Tang

at

Indiana University Bloomington

Table of Contents

Introduction

In this project, we are implementing Hungarian Algorithm for Job Assignment with minimum cost. We implemented Algorithm in JAVA which takes input as cost matrix form in which row elements represents person's costs with various jobs, and jobs are represented in columns. We present our approach in implementing the algorithm for finding minimum cost with the assignment of jobs to each person. We also mention different test-cases which are possible and solutions we obtained to them with our approach in this report.

Implementation procedure:

In our implementation procedure we follow below method for our assignment problem to be done.

1. Reads input from the file where line1 as no. of rows, line2 as no. of columns, and following lines based on the rows and columns as cost matrix.
2. Then it first does row reduction and then column reduction.
3. <u>Assignment Process:</u>
   If we get single zero in each column and each row exactly

   Assigns job to person and print all assignments made and minimum cost for it.
4. **else if**
5. <u>Ticking and Line Drawing Process:</u>
   With the help of flag arrays for each row and column we try to cover zeros present in the obtained cost matrix after row and column reductions. To cover the zeros we follow these steps according to Hungarian algorithm.
   - i) Tick all unassigned rows. Store it in Tick_row flag array as true.
   - ii) If a row is ticked and has a zero, then tick the corresponding column. Store it in Tick_Column Flag Array as true.
   - iii) If a column is ticked and has an assignment, then tick the corresponding row. Store it in Tick_row flag array as true.
   - iv) Repeat Steps ii and iii till no more ticking is possible.
   - v) Now we draw lines on un-ticked rows and ticked columns.

6. <u>Dual Adjustment Process:</u>
   Then we identify the smallest value say X_{ij}, from the remaining uncovered elements of the cost matrix and then do the following operations:
   - i) $C_{ij} = C_{ij}$, if the element is covered once by a lines i.e. where, tick_row array = **false**, and tick_column array = **false**

ii) $C_{ij} = C_{ij}-X_{ij}$, if the element is uncovered by lines drawn i.e. where, tick_row array = **true**, tick_column array = **false**

iii) C_{ij} = Cij+X_{ij}, if the element is covered twice by lines drawn i.e. where, tick_row array = **false** and tick_column array = **true**

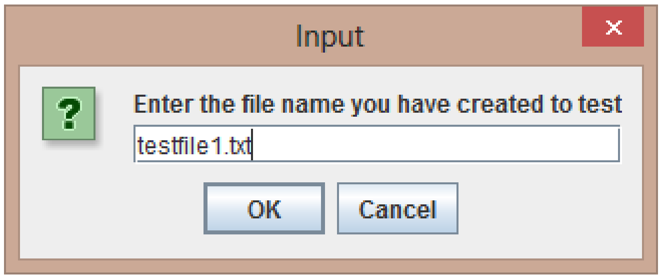
7. Assignment after Dual Adjustment process:
   If we get the exactly assignable zero for each row and column, then we assign them which give the minimum cost.
8. **else if** (the above method goes into infinite loop)
9. Greedy Approach for infinite loop problem caused by dual adjustment step:
   If min element X_{ij} in the dual adjustment is 0 then above dual adjustment step will enter into infinite loop. In this case, we follow arbitrary assignment (Greedy approach) i.e. for each element in row, we take zeros count of that each particular column and store it in an array. Then we find the index of array where the count is minimum. There can be case where count of zeros is same, and then we take minimum index among them.
10. Make assignment for the row and column index returned in step 9.
11. The greedy approach is followed for until all rows of matrix are covered.
12. Then output of job assignment is displayed in the command prompt as well in Message Box. Minimum cost of assignment is displayed in command prompt.

Execution Procedure:

User can create any test file in the folder where the class file is stored. To execute the algorithm on test file, user should give the test file name as input.

Input:

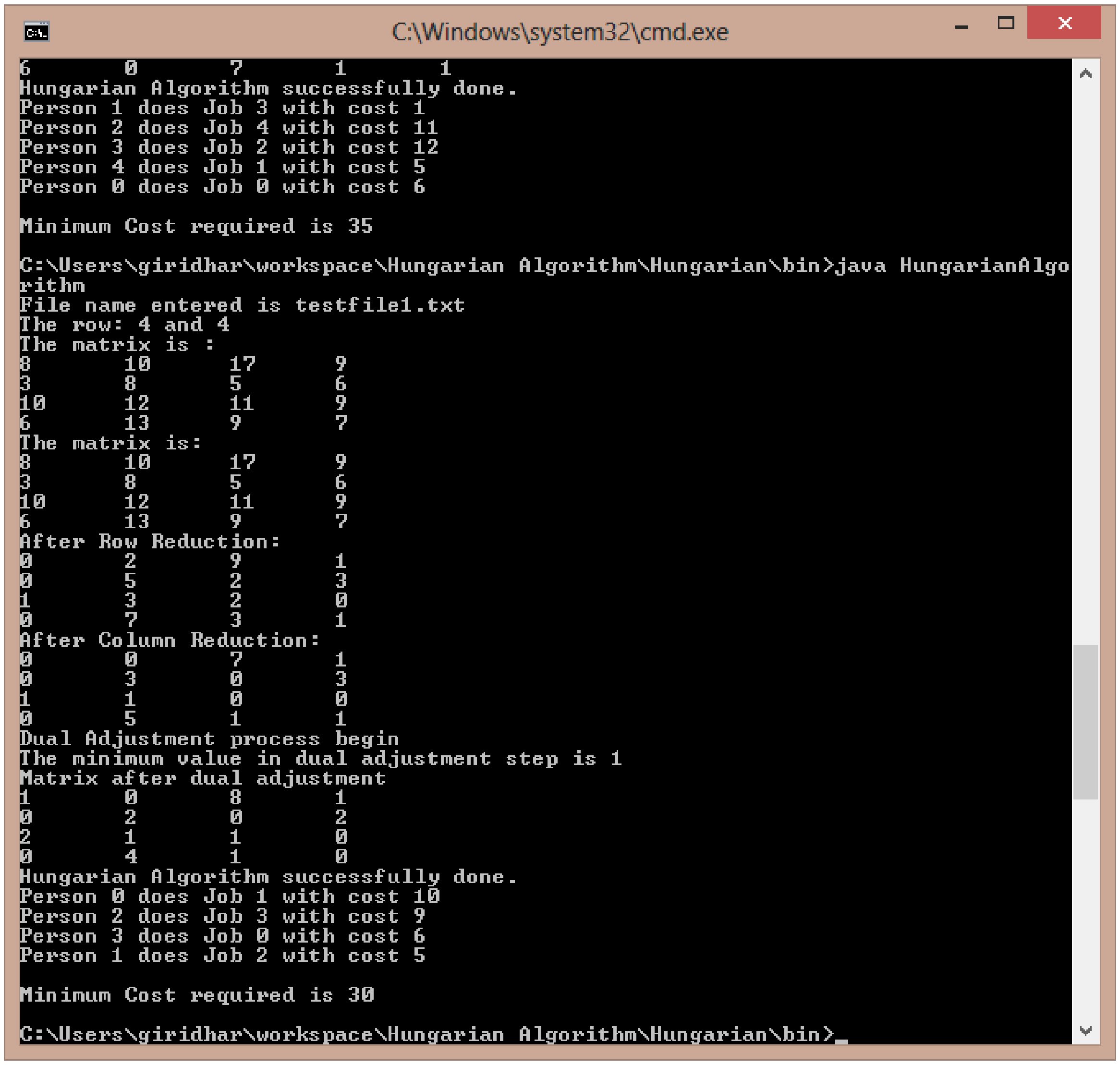
User gives text file name as input for each test-case and our program reads values present in the text file and takes as input for the problem. User gives input as follows:



Output:

The job assignment is displayed in command prompt as well as Message Box.

Minimum cost for the assignment is displayed in command prompt only.



C:\Windows\system32\cmd.exe

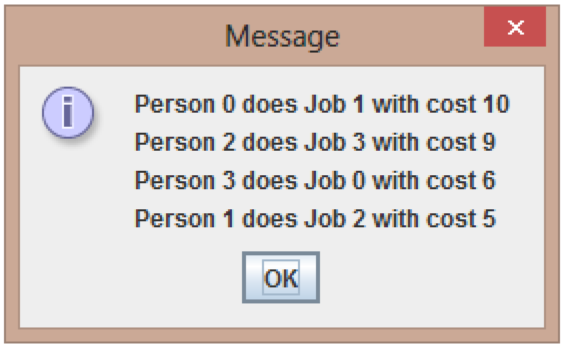
```
6         0         7         1         1
Hungarian Algorithm successfully done.
Person 1 does Job 3 with cost 1
Person 2 does Job 4 with cost 11
Person 3 does Job 2 with cost 12
Person 4 does Job 1 with cost 5
Person 0 does Job 0 with cost 6

Minimum Cost required is 35

C:\Users\giridhar\workspace\Hungarian Algorithm\Hungarian\bin>java HungarianAlgo
rithm
File name entered is testfile1.txt
The row: 4 and 4
The matrix is :
8         10        17        9
3         8         5         6
10        12        11        9
6         13        9         7
The matrix is:
8         10        17        9
3         8         5         6
10        12        11        9
6         13        9         7
After Row Reduction:
0         2         9         1
0         5         2         3
1         3         2         0
0         7         3         1
After Column Reduction:
0         0         7         1
0         3         0         3
1         1         0         0
0         5         1         1
Dual Adjustment process begin
The minimum value in dual adjustment step is 1
Matrix after dual adjustment
1         0         8         1
0         2         0         2
2         1         1         0
0         4         1         0
Hungarian Algorithm successfully done.
Person 0 does Job 1 with cost 10
Person 2 does Job 3 with cost 9
Person 3 does Job 0 with cost 6
Person 1 does Job 2 with cost 5

Minimum Cost required is 30

C:\Users\giridhar\workspace\Hungarian Algorithm\Hungarian\bin>_
```



Message

Person 0 does Job 1 with cost 10
Person 2 does Job 3 with cost 9
Person 3 does Job 0 with cost 6
Person 1 does Job 2 with cost 5

Test-cases:

Test-case 1:

In this test-case we take input from the file “testfile1.txt”, this file contains the following values

4(Number of rows)

4 (Number of columns)

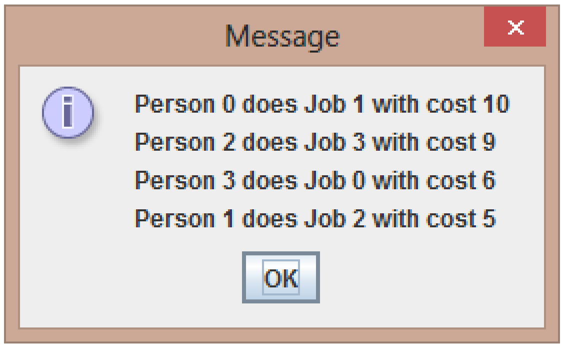
8 10 17 9 (Cost Matrix Elements)

3 8 5 6

10 12 11 9

6 13 9 7

and gives output as



```
Person 0 does Job 1 with cost 10
Person 2 does Job 3 with cost 9
Person 3 does Job 0 with cost 6
Person 1 does Job 2 with cost 5
```

In this case we get assignments as output after row and column reductions.

Test-case 2:

In this test-case we take input from the file “testfile2.txt”, this file contains the following values

5

5

6 12 3 11 15

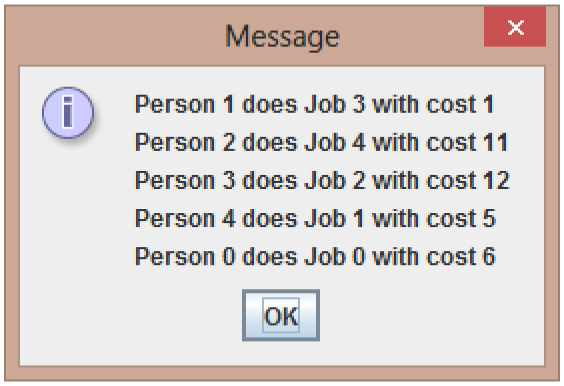
4 2 7 1 10

8 11 10 7 11

16 19 12 23 21

9 5 7 6 10

And gives output as:



```
Person 1 does Job 3 with cost 1
Person 2 does Job 4 with cost 11
Person 3 does Job 2 with cost 12
Person 4 does Job 1 with cost 5
Person 0 does Job 0 with cost 6
```

In this case, we get output after dual adjustment process.

Test-case 3:

In this test-case we take input from the file “testfile3.txt”, this file contains the following values

5

4

10 19 8 15

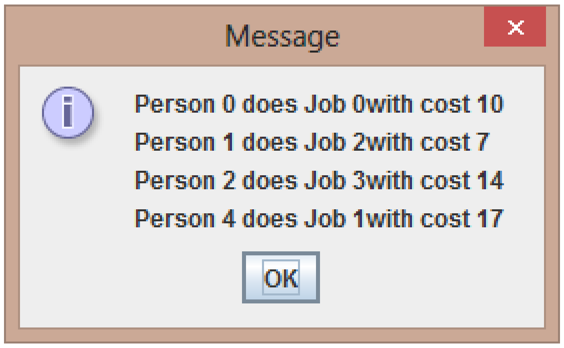
10 18 7 17

13 16 9 14

12 19 8 18

14 17 10 19

and gives output as



```
Person 0 does Job 0with cost 10
Person 1 does Job 2with cost 7
Person 2 does Job 3with cost 14
Person 4 does Job 1with cost 17
```

In this case, we get output by following dual adjustment process but here we have persons more than jobs.

Test-case 4:

In this test-case we take input from the file "testfile4.txt", this file contains the following values

4

4

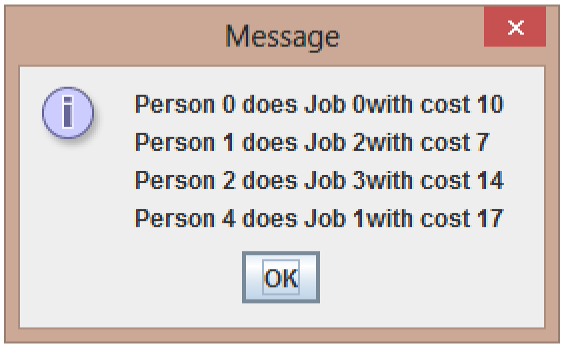
10 11 12 13

11 12 10 9

7 8 6 5

4 3 6 3

and gives output as:



```
Person 0 does Job 0with cost 10
Person 1 does Job 2with cost 7
Person 2 does Job 3with cost 14
Person 4 does Job 1with cost 17
```

In this case, we get output by following greedy method of our algorithm.

References:

1. Lec-16 Assignment Problem - Hungarian Algorithm
http://www.youtube.com/watch?v=BUGIhEecipE
2. Greedy Algorithm http://en.wikipedia.org/wiki/Greedy_algorithm

Acknowledgement:

We thank Professor Haixu Tang and instructor Yu Chuan-Yih for their continued support throughout the course of the project.