1. **Introduction about linear programming concept:**

**Linear programming is the most common decision-making method, the scientist George Dantzig was found it at 1947 and insert it as a new solution method for the problems that face the decision maker.**

**Accordingly, Linear programming can be considered as a mathematical method used in the propose of Planning and making the optimal decision Among a group of alternatives offered to maximize profits and minimize the cost as possible.**

**The main role that linear programming plays is solving the problems that lead to maximize profits and minimize costs. Linear programming is used in different fields Especially in the field of marketing and production management. Linear programming has a lot of methods and each one dedicated to solving a type of problems. This research discusses the Hungarian method which is one of the methods that used to solve assignment problems.**

1. **What is Assignment problem:**

**Assignment problem is considered as one of linear programming problems which is an determining the most efficient assignment of people to projects ,sales people to territories, auditors to companies for audits, contracts to bidders, jobs to machines, heavy equipment (such as cranes) to construction jobs, and so on. Our target is to minimize cost or time given a Specific data Provided that one person is assigned to one specific project.**

**A close up of a map

Description automatically generated**

**This is the Assignment problem network; this network is very similar to the network for the transportation problem. we can say that Assignment problem is a Special case of transportation problem in which the supply at each source and the demand at each destination must equal one. It should that one person assigned to one project, and each job only needs one person.**

**We can solve this problem by experimenting with all possible possibilities (permutations and combinations) possible and choosing the most appropriate solution, but this can only be when the problem is small and has a small number of possibilities , we cannot do that when the numbers are many and consequently there is a large number of possibilities, In this case we can use the Hungarian method Which will be explained in the second section.**

1. **About Hungarian method:**

**Hungarian method is involved to solve assignment problems in in polynomial time  with time complexity O(n4), it was developed and published in 1955 by Harold Kuhn, He has named the method by Hungarian as its algorithm was strongly based the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry ,Then it was reviewed by Munkres  in 1957 and observed that it is strongly polynomial with time complexity O(n3),.**

1. **Hungarian Method algorithm:**

* **For normal balanced minimization problems:**

1. **Matrix reduction.**

* **For each row, find the minimum number and subtract it from each element in that row.**
* **Similarly, for each column, find the minimum number and subtract it from each element in that column.**

1. **Cover all zeros with a minimum number of vertical and horizontal lines.**

* **In this step cover all zeros with vertical and horizontal lines to reach the fewest possible lines.**
* **If lines are less than the number of row and columns so we need to continue to step 3, else we can find optimal solution now .**

1. **Create additional zeros.**

**Revise the present opportunity cost table. This is done by subtracting the smallest number not covered by a line from every uncovered number. This same smallest number is also added to any number(s) lying at the intersection of horizontal and vertical lines. We then return to step 2 and continue the cycle until an optimal assignment is possible.**

* **For balanced maximization problems:**

1. **Turn the maximization problem to minimization problem.**

* **Find the maximum element in all matrix element and subtract each element in that matrix from it.**

1. **Now you can solve problem like normal balanced minimization problem.**

* **For unbalanced problems:**

1. **Create a dummy column or row.**

* **If you have a non-square matrix (number of rows not equal number of columns) you may need to have a dummy columns or dummy rows that contain a zeros to turn matrix to be squared.**

1. **Project run and working steps for my part:**

**Note: My part is the part that has code attached to.**

1. **Start frame:**

**A screenshot of a cell phone

Description automatically generated`**

**This is the start frame of the task you should fill the text fields by the required information and choose the type of problem (Maximization or Minimization) And click ok**

**A screenshot of a cell phone

Description automatically generated**

**If you don’t enter all required information (number of row or columns number) or did not choose the problem type you will have this error message**

**Note: You should enter the numbers in a digital image not in writing using letters because the use of letters leads to a n malfunction in the program**

1. **Frame with the panel that contain table and result :**

**A screenshot of a social media post

Description automatically generated**

**This panel that appeared contain the table which is have the number of columns and rows that entered Now if you clicked submit without fill table by matrix element you will see the following error message.**

**A screenshot of a cell phone

Description automatically generated**

**Else if you entered all matrix elements and click you will see the problem result in result window as following.**

**A screenshot of a cell phone

Description automatically generated  
the following code is the action performed by clicking on submit button the other actions code It will be discussed both in project documentation and in research by the rest of the team members on its parts**

A screenshot of a social media post

Description automatically generated

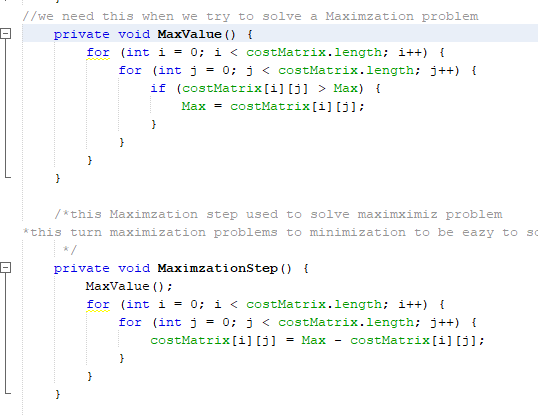
**After clicking on submit button, object of Hungrian\_method class will be created. the following constructor called and the following code will be executed. And execute function will be called also.**

**A screenshot of a social media post

Description automatically generated  
in this part of code the entered matrix elements and the number of rows and columns are set, if the matrix is unbalanced the cost matrix set with size of the dummy column and row . other arrays are set, and other variables will be discussed later in its functions .After this execute function with the following code will executed.**

****

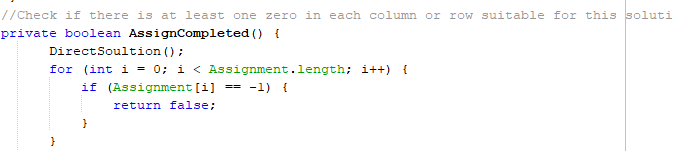
**First I check if it’s a maximization problem if true Maximization step will be called.**

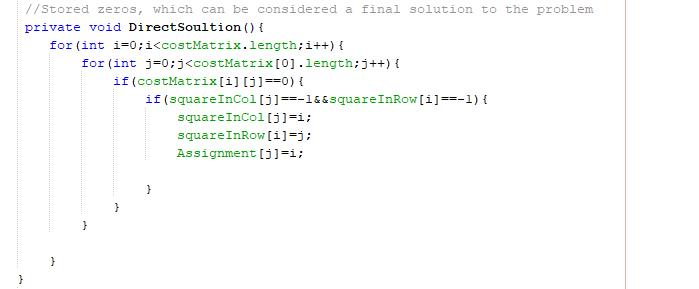
****

* **MaximaizationStep (): calls MaxValue() which in turn searches for the largest number in the matrix and assign its value in the Max variable that has appeared in the constructor before, MaximaizationStep () subtract each element in the costMatrix from the max value .**

**step 1 ,2 ,3,4 and 5 the other team members will discuss it in their parts the following explanation will be just brief to make the scenario coherent**

1. **step1(): is the reduction step (reduction of rows and columns as shown in the method algorithm.**
2. **Step2(): cover columns that contain zeroes.**
3. **Step3(): check if all columns covered or not if yes now you can now chick if AssignCompleted () or not else step4() called.**
4. **Step4(): cover rows that contain zeroes.**
5. **AssignCompleted ()**

****

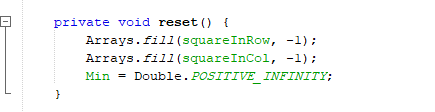
**It calls DirectSoultion() function** 

**Which stores the zeroes which consider as a final solution, then AssignCompleted () check if the assignment completed or not it yes return true else return false. If AssignCompleted () return true the execute() function return the Solution array else continue to next step**

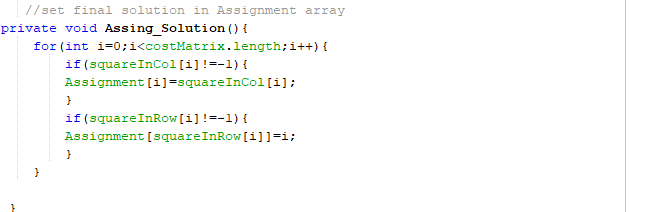
1. **While starts and it will still running till the covered lines be greater than or equal number of row and columns ( CoveredLines(): return the number of row and columns covered).**

**If number of rows and columns covered not greater than or equal number of row and columns now move to step7**

1. **Step5() : subtract Minimum uncovered number from all other uncovered CostMatrix elements, add Minimum uncovered number to the intersected numbers.**
2. **Repeat all Previous steps tell the covered lines be greater than or equal number of rows and columns or Assignment competed. And every turn arrays that used in the previous turn reset using reset () function.**

****

**If Assignment competed it will direct return the solution array else if the covered Lines became greater than or equal number of rows and columns the Assign\_solution () will be called**

 This function just places the solution in the Assignment array to return this solution array.

1. **How you can run the project?**
2. **You must have any java IDE Like “NetBeans”.**
3. **Extract the project file.**
4. **Open NetBeans or any Java IDE.**
5. **Choose file from menu bar.**

**A screenshot of a cell phone

Description automatically generated**

1. **Select open project.**

**A screenshot of a cell phone

Description automatically generated**

1. **Choose the extracted file and click open project.**

**A screenshot of a social media post

Description automatically generated**

1. **Click on run.**
2. **References:**

**Book: Quantitative Analysis for Management.**

<https://www.geeksforgeeks.org/hungarian-algorithm-assignment-problem-set-1-introduction/>

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

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

<http://www.hungarianalgorithm.com/hungarianalgorithm.php>

<https://www.youtube.com/watch?v=rrfFTdO2Z7I&t=473s>

<https://www.youtube.com/watch?v=bqX11QhMEfM&t=2s>