Week 8

Chinese Postman Problem*

8.1 Chinese Postman Problem

It is named in honor of the Chinese mathematician Guan Meigu. As the name suggests it has post office and there are different streets in the city with some junctions.
To traverse each street, there is some amount of post associated with each as shown in fig.
The problem is that can a postman who starts from the post office visits all the streets and come back to the starting point which is P and the cost should be minimum.
In Graph theory this concept is equivalent to Eulerian circuit.

The name appears from the establishment of a cohesive relationship between the n group of men and the group of n women. If every man fits k women and every woman fits k men, there should be a perfect match. Again edges are allowed, which increases the range of applications.

*Lecturer: Ananad Mishra. Scribe: Hera Ahsan.

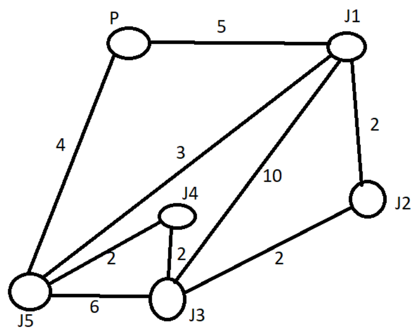

Figure 8.1:

Case 1:

$$\forall \subseteq V(G)$$
$$deg(v_i) is even$$

Case 2:

$$\forall \subseteq V(G)$$
$$deg(v_i) is odd$$

If all the vertex is equal, the graph is Eulerian and the answer is Total hem weights. If not, we have to repeat the edges. All crossing is an Eulerian graphic circuit obtained by repeating the edges. Finding a very short crossing is equivalent to finding the minimum weight of the ends of your repetition will make all vertex degrees equal. We say "repetition" because we do not have to use the edge more than twice. If we use a third or more edge times in making all the vertices equal, then removing two of those copies will go away all vertices are equal. There may be many ways to choose duplicate edges.
Example 1:

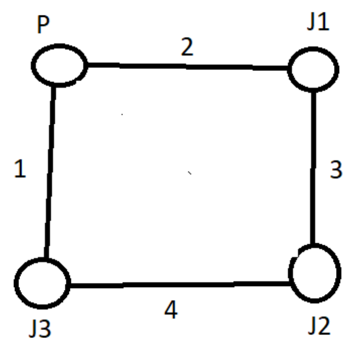

Figure 8.2:

This belongs to Case 1 because every vertex has even degree.
Example 2:

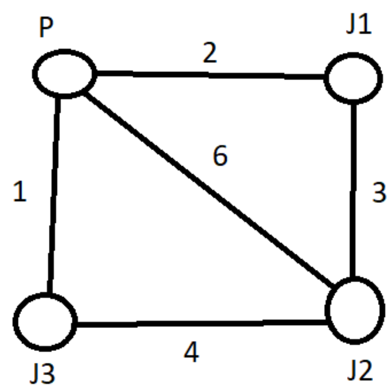

Figure 8.3:

Here P and J has odd degree because edges need to be visited twice.To find minimum cost we need to make the odd degree nodes even by additional edge.
Here are the 3 possibilities but the solution will be the one which will give the minimum cost.

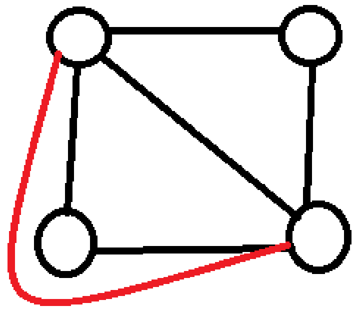

Figure 8.4:

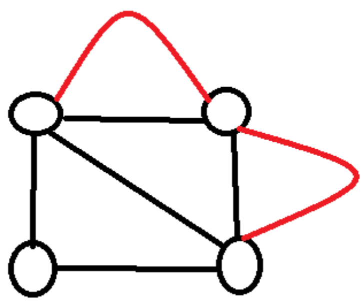

Figure 8.5:

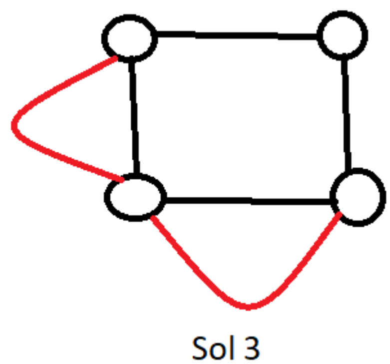

Figure 8.6:

In first one , extra cost is 6.
In second, extra cost is 5.
In third , it is 5.
Best solution is either B or C.
For second it will be P J1 J2 J3 P J2 J1.

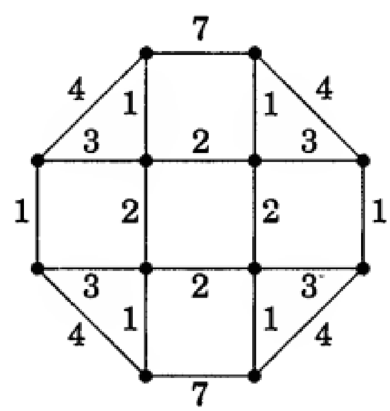

Figure 8.7:

Problem: The outer eight vertices have odd degree and the inside ones are even.
We have to pair them so that it become even degree. Adding an edge from the odd vertex to the corresponding vertex creates an equal vertex strange. We must continue to add edges until we complete the trail to the strange vertex. Duplicate edges should contain a collection of unusual pairing lines vertices.
Best solution will be 4+4+2+2 = 12.

8.2 Weighted Bipartite Matching

Problem: Let's say there are 4 construction sites: S1, S2, S3, S4 and 4 cranes: C1, C2, C3, C4. Distance between sites and cranes are given in matrix form.
The problem is which crane is sent to which site such that the cost is minimum.
There are can be many such problems like this.
These problems are called as Assignment problem and the graph of this is a weighted bipartite graph.

Transversal A transversal of an N X N matrix consists of N positions one in each row and one in each column.

$$\begin{pmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{pmatrix}$$

Finding a transversal with maximum sum is the Assignment Problem. This is the matrix formulation of the maximum weighted matching problem.
The brute force algorithm of the assignment problem is going to take n! time where all possible transversals are retrieved.

$$nx(n-1)x(n-2)x.......$$

But this n! can be a very large number also so this is not a practical solution.
Assignment problem as optimization.
minimize

$$\Sigma_i \Sigma_j C_i j X_i j$$

$$1.\Sigma_i x_i j = 1$$

$$2.\Sigma_j x_i j = 1$$

$$x_i j \subseteq 0, 1 x_i j \geq 0$$

$$\begin{pmatrix} C_0 0 & C_0 1 & C_0 1 & .. \\ .. & .. & 2 & \\ .. & .. & C_n n & \end{pmatrix}$$

To solve these problems a new method was introduced which is Hungarian Algorithm.

8.3 Hungarian Algorithm

The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover.The algorithm begins with a cover. It can terminate only when the equality subgraph has a perfect matching, which guarantees equal value for the current matching and cover.

$$\begin{pmatrix} 40 & 60 & 15 \\ 25 & 30 & 45 \\ 55 & 30 & 25 \end{pmatrix}$$

Step 1: Find row min:

$$15, 25, 25$$

Step 2: Subtract row min with every element.

$$\begin{pmatrix} 25 & 45 & 0 \\ 0 & 5 & 20 \\ 30 & 5 & 0 \end{pmatrix}$$

Step 3: Find column min:

$$0, 5, 0$$

Step 4: subtract column min with every element.

$$\begin{pmatrix} 25 & 40 & 0 \\ 0 & 0 & 20 \\ 30 & 0 & 0 \end{pmatrix}$$

Cover all the zeroes With the minimum number of lines.
3 lines will be required in this case.
If no. of lines required = n then STOP.

$$\begin{pmatrix} 25 & 40 & 0^* \\ 0^* & 0 & 20 \\ 30 & 0^* & 0 \end{pmatrix}$$

$$0^* aretheassignments.$$

$$Totalcostofassignment : 25 + 30 + 15$$

Example 2:

$$\begin{pmatrix} 30 & 25 & 10 \\ 15 & 10 & 20 \\ 25 & 20 & 15 \end{pmatrix}$$

Step 1: Find row min:

$$10, 10, 15$$

Step 2: Subtract row min with every element.

$$\begin{pmatrix} 20 & 15 & 0 \\ 5 & 0 & 10 \\ 10 & 5 & 0 \end{pmatrix}$$

Step 3: Find column min:

$$5, 0, 0$$

Step 4: subtract column min with every element.

$$\begin{pmatrix} 15 & 15 & 0 \\ 0 & 0 & 10 \\ 5 & 5 & 0 \end{pmatrix}$$

Cover all the zeroes With the minimum number of lines.
2 lines will be required in this case.
Since 2 ¡ 3 therefore we need to find the minimum in all the uncovered values and then subtract that with all the uncovered values.

$$\begin{pmatrix} 10 & 10 & 0^* \\ 0^* & 0 & 15 \\ 0 & 0^* & 0 \end{pmatrix}$$

$$0^* aretheassignments.$$

$$Totalcostofassignment : 10 + 10 + 25$$

OR

$$Totalcostofassignment : 10 + 15 + 20$$