

# A study on Euler Graph and it's applications

Ashish Kumar

M.Sc. Mathematics  
Department of Mathematics and Statistics, SHUATS  
Allahabad, U.P., India

**Abstract:-** Main objective of this paper to study Euler graph and it's various aspects in our real world. Now a day's Euler graph got height of achievement in many situations that occur in computer science, physical science, communication science, economics and many other areas can be analysed by using techniques found in a relatively new area of mathematics. The graphs concerns relationship with lines and points (nodes). The Euler graph can be used to represent almost any problem involving discrete arrangements of objects where concern is not with the internal properties of these objects but with relationship among them.

To achieve objective I first study basic concepts of graph theory, after that I summarizes the methods that are adopted to find Euler path and Euler cycle.

**Keywords:-** graph theory, Konigsberg bridge problem, Eulerian circuit.

## Introduction

A graph  $G$  consists of a set  $V$  called the set of points (nodes, vertices) of the graph and a set of edges such that each edge  $e \in E$  is associated with ordered or unordered pair of elements of  $V$ , i.e., there is a mapping from set of edges  $E$  to set of ordered or unordered pairs of elements of  $V$ . The set  $V(G)$  is called the vertex set of  $G$  and  $E(G)$  is the edge set. The graph  $G$  with vertices

$V$  and edges  $E$  is written as  $G = (V, E)$  or  $G(V, E)$ . Types of graphs are simple graph, multi graph, pseudo graph. An undirected graph  $G$  consists of sets of vertices  $V$  and a set of edges  $E$  such that each edges is associated with an unordered pair of vertices. A directed graph  $G$  consists of a set of vertices  $V$  and a set of edges  $E$  such that each edges is associated with an ordered pair of vertices then the graph is called directed graph or digraph.

The degree of a vertex of an graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex  $v$  in a graph  $G$  denoted by  $\deg(v)$ .

If  $G = (V, E)$  be an undirected graph with  $e$  edges. Then

$$\sum_{v \in V} \deg(v) = 2e$$

i.e., the sum of degrees of the vertices in an undirected graph is twice the number of edges (**Handshaking theorem**).

If  $G = (V, E)$  be a directed graph with  $e$  edges, then

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = e$$

i.e., the sum of the out degrees of the vertices of a digraph  $G$  equals the sum of in degrees of the vertices which equals the number of edges in  $G$ .

## 1) Origin and Importance of Eulerian graph (Konigsberg problem)

The river called Pregel flows through the city Konigsberg (located in Russia) dividing the city into four land regions, two are river banks and two are islands or delta formation. The four land regions were connected by 7 bridges. There was an entertaining or interesting exercise for the citizens of Konigsberg. Start from any land regions and come back to the starting point after crossing each of the seven bridges exactly once without repeating same path.

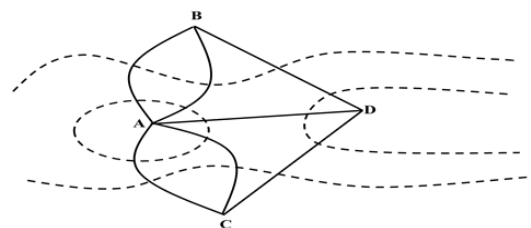


Fig. 1: The bridges of Konigsberg problem

Euler suggested and further explained that it is impossible to do so by using the terminology of points (representing the land regions) and lines (representing the bridges). Hence, he titled his paper as "Solutions to a problem relating to the geometry of positions." Through this explanation, he laid the foundation for Graph Theory. Graph theory started from this problem.

He abstracted the case of Königsberg by eliminating all unnecessary features. He drew a picture consisting of “dots” that represented the landmasses and the line-segments representing the bridges that connected those land masses. The resulting picture looked somewhat similar to the figure shown below.

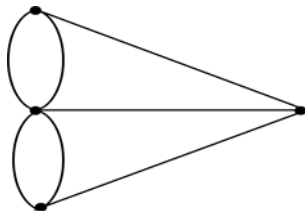
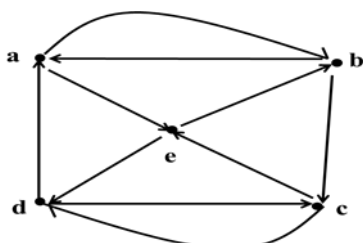


Fig. 2

This simplifies the problem to great extent. He came out with the then new concept of degree of nodes. The Degree of Node can be defined as the number of edges touching a given node. Euler proposed that any given graph can be traversed with each edge traversed exactly once if and only if it had, zero or exactly two nodes with odd degrees. The graph following this condition is called Eulerian circuit or path.

Using Euler’s theorem we need to introduce a path to make the degree of two nodes even. And other two nodes can be of odd degree out of which one has to be starting and other at another the end point. Suppose we want to start our journey from node. So, the two nodes can have odd edges. But somehow we need to edit the actual graph by adding another edge to the graph such that the two other nodes have even degree.

Trail that visits every edge of the graph once and only once is called **Eulerian trail**. Starting and ending vertices are different from the one on which it began. A graph of this kind is said to be traversable. An Eulerian circuit is an Eulerian trail that is a circuit i.e., it begins and ends on the same vertex. A graph is called Eulerian when it contains an **Eulerian circuit**. A digraph in which the in-degree equals the out-degree at each vertex.



A vertex is odd if its degree is odd and even if its degree is even.

## 2) Existence of an Euler path

There are several ways to find the existence of Euler path. Considering the existence of an Euler path in a graph is directly related to the degree of vertices in a graph. Euler formulated the theorems for which we have the sufficient and necessary condition for the existence of an Euler circuit or path in a graph respectively.

**Theorem:** An undirected graph has at least one Euler path if and only if it is connected and has two or zero vertices of odd degree.

**Theorem:** An undirected graph has an Euler circuit if and only if it is connected and has zero vertices of odd degree.

Let us take a graph having no odd vertices, the path can begin at any vertex and will end there; vice-versa in the case of two odd vertices, the path must begin at one odd vertex and end at the other. Any finite connected graph with two odd vertices is traversable. A traversable trail may begin at either odd vertex and will end at the other odd vertex.

Finding an Euler path is a relatively simple problem it can be solve by keeping few guidelines in our mind:

- Always leave one edge available to get back to the starting vertex (for circuits) or to the other odd vertex (for paths) as the last step.
- Don’t use an edge to go to a vertex unless there is another edge available to leave that vertex (except for the last step)

## 3) Constructive Algorithm

Constructive algorithm used to the prove Euler’s theorem and to find an Euler cycle or path in an Eulerian graph. A graph with two vertices of odd degree. The graph with its edges labelled according to their order of appearance in the path found. Steps that kept in mind while traversing Euler graph are first to choose any vertex  $u$  of  $G$ . Start traversing through edges that not visited yet until a cycle is formed. Record the cycle and remove the edges it consists of. If there is an unvisited edge’s then start the first step until whole graph is traversed.

Two cycles emerged by traversing one of them and insert other when common vertex found. The result is a new cycle. For an Eulerian graph that must contain two vertices with odd degree, otherwise no Euler path can be found. Start from a vertex of odd degree  $u$ . Then add or remove edge between the vertices of odd degree and thus ensure that every vertex has an even degree

**Example:** Illustrations of Constructive algorithm to find Euler cycle, consider the graph

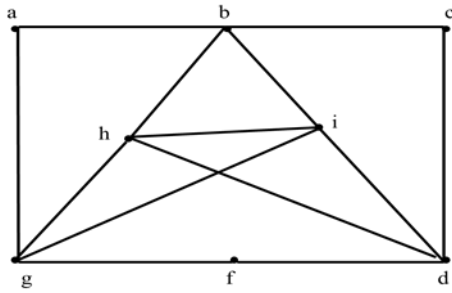


Fig. 4(a)

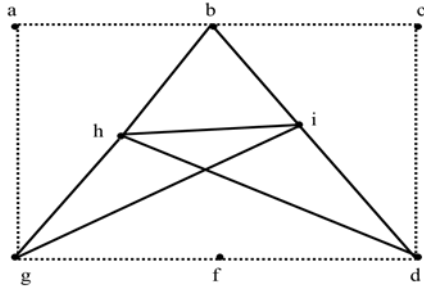


Fig. 4(b)

(a) Check to start that the graph is connected and all vertices are of even degree, find cycle

Cycle of length 6 such as  $((g, a), (a, b), (b, c), (c, d), (d, f), (f, g))$

(b) Remove the edges involved in this cycle from the graph:

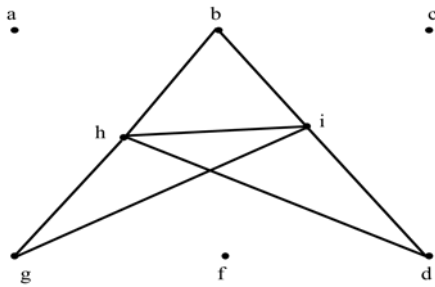


Fig. 4(c)

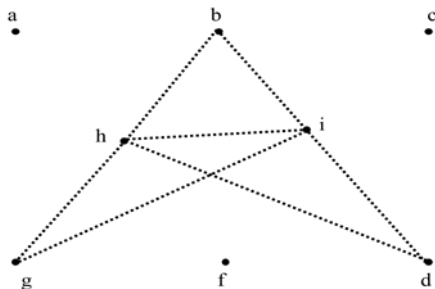


Fig. 4(d)

In the 'smaller' graph that remains, the vertices must still all be of even degree.

Cycle of length 7  $((g, h), (h, b), (b, i), (i, d), (d, h), (h, i), (i, g))$ .

(c) Merge the cycles from step 2 into the cycle in step 1 at appropriate points:

Cycle of length 13

$((g, a), (a, b), (b, c), (c, d), (d, f), (f, g), (g, h), (h, b), (b, i), (i, d), (d, h), (h, i), (i, g))$ .

**Theorem:** A connected graph has an Euler trail, but not an Euler cycle, if and only if it has exactly two edges of odd degree.

**Examples:** An Euler trail exists for the graph in fig. 5

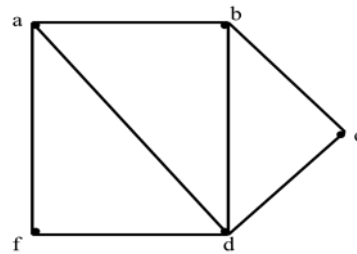


Fig. 5

Euler trail

$((a, f), (f, d), (d, c), (c, a), (b, d), (d, a), (a, b))$

An algorithm for constructing an Euler trail is the following:

1. If the graph is connected and all its vertices are of even degree then construct an Euler cycle (necessarily it's an Euler trail). Otherwise, if the graph is connected and has exactly two vertices are of odd degree, identify those vertices as the initial  $v_i$  vertices and end  $v_e$  vertices of the Euler trail to be constructed and remove the edges along a trail joining them. Find an Euler cycle in what remains.

2. If the cycle obtained is written using  $v_i$  as its initial vertices, then the edges of the trail can simply be added to the end of the cycle to give an Euler trail for the original graph.

**Example:** An Euler cycle for the graph same as in fig.5 can be found by Constructive algorithm:

Noting: that  $\deg(a) = 3$  and  $\deg(b) = 3$  identifies these as the start and end of the trail.

The edge  $(a, b)$  (which acts as a trail from  $a$  to  $b$ ) is thus removed from the graph to start leaving the 'new' graph shown below:

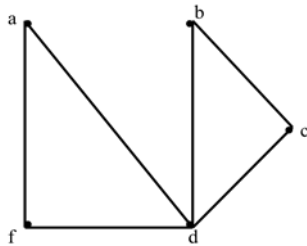


Fig.6(a)

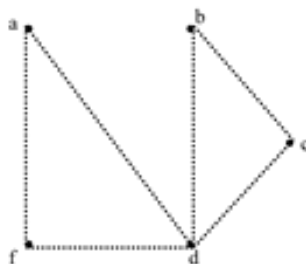


Fig. 6(b)

Cycle found, Cycle of length 6  
 $((a, f), (f, d), (d, c), (c, b), (b, d), (d, a))$

Attaching the edge  $(a, b)$  to the end of this cycle gives an Euler trail for the original graph:

$((a, f), (f, d), (d, c), (c, a), (b, d), (d, a), (a, b))$ .

**Eulerization and Semi- Eulerization**

In case Eulerian circuit or path does not exist for the graph. There is a simple process of Eulerization which provides a solution for this type of problem. Eulerization nothing but it is the process of adding phantom (or duplicate) edges to the graph so that the resulting graph has not any vertex of odd degree (and thus contains an Euler circuit). A similar problem rises for obtaining a graph that has an Euler path. The process in this case is called Semi-Eulerization and ends with the creation of a graph that has exactly two vertices of odd degree.

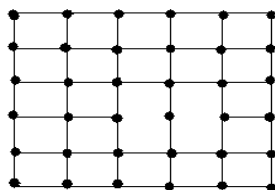


Fig. 7: Eulerization

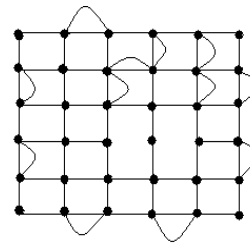


Fig. 8: Semi- Eulerization

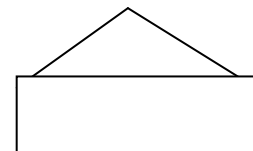
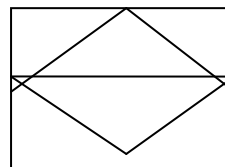
Eulerization can be achieved in many ways by selecting a different set of edges to duplicate.

**4) Application:**

**Line drawings**

A graph has a **unicursal tracing** if it can be traced without lifting the pencil or retracing any line. Obviously, a closed unicursal tracing of a line drawing is equivalent to an Euler circuit in the corresponding graph. Similarly, an open unicursal tracing is equal to an Euler path.

- A line drawing has a closed unicursal tracing iff it has no points of intersection of odd degree.
- A line drawing has an open unicursal tracing iff it has exactly two points of intersection of odd degree.



**Chinese Postman Problem**

A part of his duties, a postman starts from his office, visits every street at least once, delivers the mail and comes back to the office. Suggest a route of minimum distance. The Chinese Postman Problem (CPP) is a close cousin to TSP (Travelling Salesman Problem). In this routing problem the traveller (Postman or Salesman) must traverse every arc (i.e. road or street link) in the network. The name comes from the fact that a Chinese mathematician, Mei-Ko Kwan (1962), developed the first algorithm to solve this problem for a rural postman. It is an extension to one of the earliest graph theory questions, the Königsberg Bridge Problem, which was studied by Euler (1736). Operations researchers are interested in finding the shortest route in any type of network.

Algorithm was designed to give the shortest route that was required for any network or road map, including those that had more than two points (nodes) with odd numbers of lines (edges) emerging from them. If a road map or network had more than two nodes that were of odd order, the odd ordered nodes had to be paired together, and the shortest distance found between the pairs. If the network was traversed, these edges would have to be repeated to give the shortest distance to travel all edges and return to the starting point.

The algorithm is as follows:

Step 1: Find all nodes with odd numbers of edges emerging from them

Step 2: These nodes should be listed in all the possible pairings

Step 3: Find the shortest distance between each node of the pair, then add up the total for each set of pairs.

Step 4: Choose the set of pairs with the minimum total distance. This is added on to the sum of all the edges to give the shortest distance to traverse the network and return to the original point of entry.

A postman has to go around the following route starting and finishing at *A*, his postal depot. He has to go along each road, shown as lines, once and only once in graph  $A = \text{depot}$ ,

*B* = Basant Vihar, *C* = Charm Villa,  
*D* = Don House, *E* = Anjali Market,  
*F* = Fishries Shop

Fig. 4.2 is a weighted graph. Where weight defined between two node/vertices.

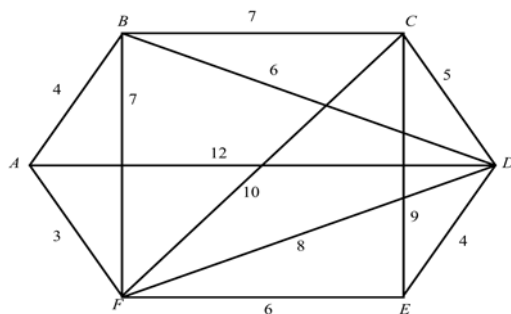


Fig. 9

Following the steps of the route inspection algorithm:

**Step 1**

Odd Vertices	Degree
<i>A</i>	3
<i>D</i>	5
<i>E</i>	3
<i>F</i>	5

**Step 2 & 3**

Possible pairings of odd vertices *AD* and *EF* shortest route *AD* and *EF* distance  $12+6=18$

Possible pairings of odd vertices *AE* and *DF* shortest route *AFE* and *DF* distance  $3+6+8=17$

Possible pairings of odd vertices *AF* and *DE* shortest route *AF* and *DE* distance  $3+4=7$

**Step 4:** From the above calculations it is clear that to create an Eulerian graph that is the shortest possible edges *AF* and *DE* must be repeated.

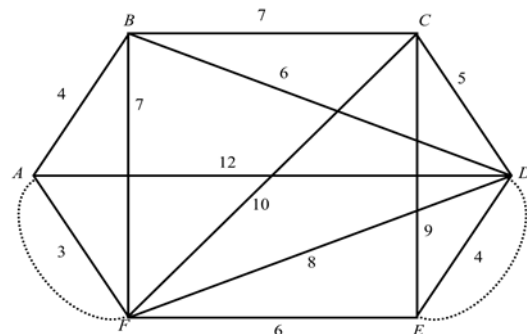


Fig. 10

A possible Euler cycle that starting and finishing at *A*:

$A-F-E-D-C-B-A-D-E-C-F-D-B-F-A$

The total length of this route is:

$$4 + 7 + 5 + 6 + 7 + 12 + 10 + 8 + 9 + 6 + 4 + 3 + \text{Repeated edges } (4 + 3) = 78$$



**City Street Map:** 17 Streets to sweep with Sweeper Truck. Pretend for simplicity, that he can sweep whole street in one pass.

**Job:** Start at Garage, sweep all 17 streets once, and return back into Garage. If travel some streets twice, ensure that is minimum number streets.

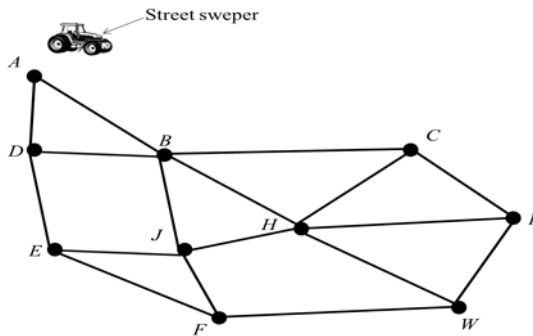


Fig. 11

Optimizing by Constructive algorithm

**Step 1:** Determine degree of each vertex-

$A = 2, B = 5, C = 3, D = 3, E = 3, F = 3, J = 4, H = 5, I = 3, W = 3$

**Step 2:** Identify all odd degree vertices- There are 8 odd vertices

**Step 3:** Change all odd degree vertices to even by removing fewest number edges shown in figures.

Removing an edge between vertices *D* and *E*.  
 Now again removing edge between *B* and *H*. Now again removing edge between *C* and *I*. Now removing edge between *F* and *W*. Now that the street map has been optimally Eulerized.

Graph will now have an Euler Circuit since each vertex is even degree.

Determine the Fig. 11. It's not an Eulerian circuit most of vertices are of odd degree. After optimizing the application of Eulerian circuit we get Fig. 11(e) by removing edges to make even degree

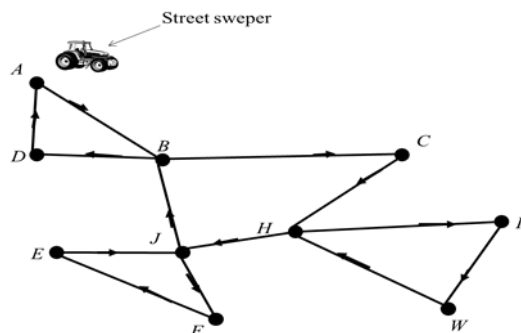


Fig. 11(e)

Euler cycle by Constructive algorithm

$A - B - C - H - I - W - H - J - F - E - J - B - D - A$

Starting at *A* (garage), sweep all roads once, only once.

**Conclusion**

Eulerian graph is applicable in many real situations. Applications of Eulerian circuits abound. For example, Eulerian circuits are obviously desirable in the deployment of street sweepers, snowplows, buses and mail carriers. In these applications, traversing a street more than once is a waste of resources. Thus, Eulerian circuits represent optimal solutions in terms of conserving resources. Methods such as these have been implemented and have resulted in considerable cost savings to the municipalities involved.

**References**

- [1] Bhatt S., Even S., Greenberg D., and Tayar R. [2002]: "Traversing directed Eulerian mazes" Journal of Graph Algorithms and Applications, Vol. 6, Series 2, pp. 157-173.
- [2] Chwe Byoung-Song [1994]: "A proof of some Schutzenberger type results for Eulerian paths and circuits on digraphs" International Journal of Mathematics and Mathematical Science, Vol. 17 No. 3, pp. 497-502.
- [3] Dvorak Tomas, Havel Ivan and Liebl Petr [1997]: "Euler cycles in the complete graph  $K_{2m+1}$ " Discrete Mathematics, Vol. 171, pp. 89-102.
- [4] Isaev M. I. [2012]: "Asymptotic behaviour of the number of Eulerian circuits" Mathematics Company, arXiv:1104:3046v3
- [5] Jonsson Jakob [2002]: "On the number of Euler trails in directed graphs" Mathematica Scandinavica, Vol. 90, pp. 191-214.
- [6] Jordon Heather, Chartrand Gary and Zhang Ping [2014]: "A cycle decomposition conjecture for Eulerian graphs" Australian Journal of Combinatorics, Vol. 58, pp. 48-59.
- [7] Khade R. H. and Chaudhari D. S. [2012]: "An approach for minimizing CMOS layout by applying Euler's path rule" Proceedings published in International Journal of Computer Applications, Vol. 10, pp. 18-21
- [8] Kotzig Anton and Turgeon M. Jean [1982]: "Quasi-groups defining Eulerian paths in Complete Graphs" Journal of Combinatorial Theory, Series B32, pp. 45-56.
- [9] Mcdiarmid Lolin and Michael Molloy [2002]: "Edge-disjoint cycles in regular directed graph" Journal of Graph Theory, Vol. 22, Issue 3, pp. 231-237.
- [10] Sarma Samar Sen, Basuli Krishnendu and Naskar Saptarshi [2008]: "Determination of Hamiltonian circuit and Euler trail from a given graphic degree sequence" Journal of Physical Sciences, Vol. 12, pp. 249-252.
- [11] Shapira Asaf, Huang Hao, Ma Jie, Sudakov Benny and Yuster Raphael [2012]: "Large feedback arc sets,

*high minimum degree subgraphs, and long cycles in Eulerian digraphs*” *Combinatorics, Probability and Computing*, Vol. 22, Issue 06, pp. 859-873.

- [12] Shen Jain and Brualdi A. Richard [2002]: “*Disjoint cycles in Eulerian digraphs and the diameter of interchange graphs*” *Journal of combinatorial theory, Series B* 85, pp. 189–196.
- [13] Subramanian K. G., Hasni Roslan and Ismail Abdul Samad [2009]: “*Some applications of Eulerian graphs*” *International Journal of Mathematical Science Education*, Vol. 2, Series 2, pp. 1 – 10.