Module 5 Eulerian Graphs

Contents

5.1 Motivation and origin

The existence of closed trails containing all the edges in a graph is the subject of first paper in graph theory (1736). It was written by Leonhard Euler (1707−1783)[1], thus initiating the theory of graphs. Like many combinatorial problems, Euler's paper has its motivation in a problem that can be easily stated.

Königsberg-7-bridge-problem: The river Pregel flows through the city of Königsberg (located in Russia) dividing the city into four land regions of which, two are banks and two are islands. During the time of Euler, the four land regions were connected by 7 bridges as shown below.

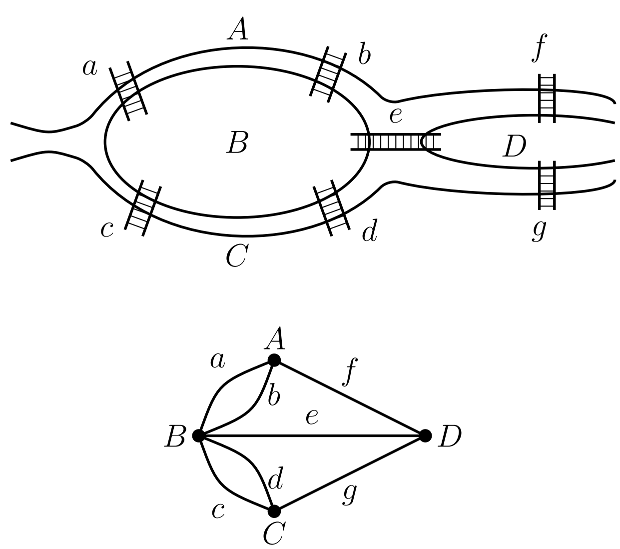


Figure 5.1: Pregel river in the city of Königsberg with 7 bridges and its representation as visualized by Euler.

The citizens of Königsberg had an entertaining exercise. Start from any land region and come back to the starting point after crossing each of the seven bridges

[1]L. Euler: Solutio Problematis ad Geometriam Situs Pertinents [Traslation: The solution of a problem relating to the geometry of position], *Commentar: Academiae scientiarum Imperialis Petropolitanae*, 8(1736), p 128-140.

exactly once. Euler 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.

Definitions.

- *A trail in a graph which contains all its edges is called an **Eulerian trail**. It can be open or closed.*
- *A graph is called an **Eulerian graph** if it contains a **closed** Eulerian trail.*

Note that an Eulerian graph is necessarily connected. Moreover, if G is Eulerian then one can choose an Eulerian trail starting and ending from any given vertex.

Theorem 5.1 (Euler, 1736)**.** *A connected graph G is Eulerian iff every vertex has even degree.*

Proof. (1) G is Eulerian $\Rightarrow$ Every vertex has even degree.

Let $W = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{t-1}, e_t, v_t(= v_0))$ be a closed Eulerian trail in G. If v_i is an internal vertex ($\neq v_0$) of W appearing k times, then $deg_G(v_i) = 2k$. If v_0 appears r times internally, then $deg_G(v_i) = 2r + 2$.

(2) (Proof due to Fowler, 1988) Every vertex is of even degree $\Rightarrow$ G is Eulerian.

This implication is proved by induction on m. If $m \leq 2$, then G is one of the following four graphs. Clearly, each of them is Eulerian.

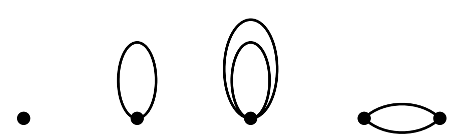


Figure 5.2: All connected graphs with at most 2 edges and every vertex of even degree.

We proceed to prove the induction step assuming that the implication holds for all connected graphs with at most $m-1$ edges and that G has m edges. Let x be any vertex of G, and (x, w_1), (x, w_2) be two of the edges incident with x; w_1, w_2 need not be distinct. Consider the graph H obtained from G by deleting (x, w_1), (x, w_2) and adding a new edge $f = (w_1, w_2)$; see Figure 5.3. The graph H has $m-1$ edges and its every vertex has even degree. However, H may be connected or disconnected. So, we consider two cases.

Case 1: H is connected.

By induction hypothesis, H contains a closed Eulerian trail, say

$$W = (v_0, e_1, v_1, \ldots, w_1, f, w_2, \ldots, v_0).$$

Then the trail

$$W^* = (v_0, e_1, v_1, \ldots, w_1, \underbrace{(w_1, x), x, (x, w_2)}, w_2, \ldots, v_0)$$

is a closed trail in G; see the figure below.

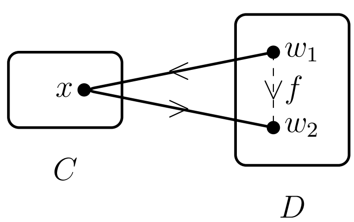


Figure 5.3: Construction of a new graph and the extension of a trail.

Case 2: H is disconnected.

In this case, H contains two components, say C and D such that $x \in C$ and $f \in D$. Both the graphs C and D have less than m edges and every vertex in $V(C) \cup V(D)$ is even. Hence, by induction hypothesis, C and D contain Eulerian trails, say $W_1(x,x)$ and $W_2 = (v_0, e_1, v_1, \ldots, w_1, f, w_2, \ldots, v_0)$. Then

$$(v_0, e_1, v_1, \ldots, w_1, \underbrace{(w_1, x), W_1(x,x), (x, w_2)}, w_2, \ldots, v_0)$$

is an Eulerian trail in G. □

Using the above theorem it is easy to conclude that the graph shown in Figure 5.1 is non-Eulerian.

Corollary. *If a connected graph G contains exactly two vertices of odd degree say x and y, then it contains a (x,y)-Eulerian trail.*

Proof. Let G^* be a new graph obtained by adding a new vertex z and joining it to x and y. Clearly, G^* is an Eulerian graph in which z has degree 2. Without loss of generality, let

$$W = (z, (z,x), x, \ldots, y, (y,z), z)$$

be a closed Eulerian trail in G^*. But then the sub-trail $W' = (x, \ldots, y)$ of W is a required (x,y)-trail in G. □

Corollary. *If a connected graph G contains $2k$ (≥ 2) vertices of odd degree, then $E(G)$ can be partitioned into k sets $E_1, E_2, \ldots, E_k$ such that each E_i induces a trail.*

Proof. Apply the above proof technique. □

5.2 Fleury's algorithm to generate a closed Eulerian trail

Though Euler's theorem neatly characterizes Eulerian graphs, its proof is existential in nature. Fleury(1983) described an algorithm to generate a closed Eulerian trail in a given connected graph in which every vertex has even degree. In the following we describe this algorithm and prove that it indeed generates an Eulerian trail.

Fleury's algorithm:

Input: A weighted connected graph $(G, \mathcal{W})$ in which every vertex has even degree.

Output: A closed Eulerian trail W.

Step 1: Choose a vertex v_0 (arbitrarily) and define the trail $W_0 := v_0$.

Step 2: After selecting a trail, say $W_k = (v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k)$, form the graph $G_k = G_{k-1} - e_k$. (That is, $G_k = G - \{e_1, e_2, \dots, e_k\}$, where $G_0 = G$).

Step 3:

(i) If there is no edge incident with v_k in G_k, then **stop**. Declare W_k is a closed Eulerian trail of G.

(ii) If there is an edge incident with v_k in G_k, select an edge say $e_{k+1} = (v_k, v_{k+1})$, giving preference to a non-cut-edge of G_k. Define $W_{k+1} = (v_0, e_1, v_1, e_2, \dots, e_k, v_k, e_{k+1}, v_{k+1})$. Goto Step 2 with W_{k+1}.

An illustration: The trails $W_1, \dots, W_{10}$ and the graphs $G_1, \dots, G_{10}$ generated by the algorithm are shown below. W_{10} is a closed Eulerian trail. For simplicity, we have shown the trails with edges alone.

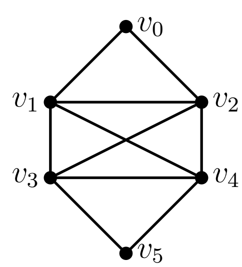
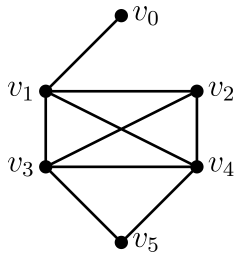
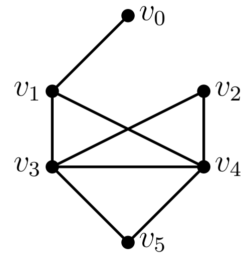


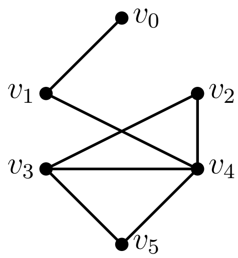
Figure 5.4: Input graph with $W = v_0$ and $G = G_0$.



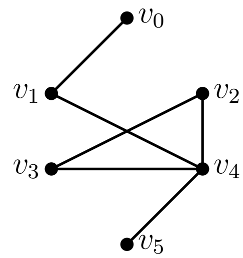
(a) $W_1 = (v_0, v_2)$;
$G_1 = G - \{(v_0, v_2)\}$



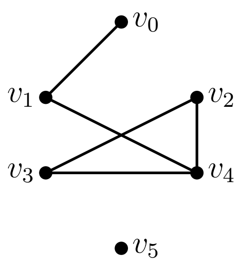
(b) $W_2 = (v_0, v_2)(v_2, v_1)$;
$G_2 = G - \{(v_0, v_2)(v_2, v_1)\}$.



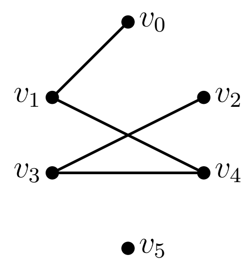
(c) $W_3 = (v_0, v_2)(v_2, v_1)(v_1, v_3)$
$G_3 = G - \{(v_0, v_2)(v_2, v_1)(v_1, v_3)\}$



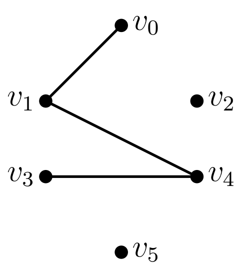
(d) $W_4 = W_3 \to (v_3, v_5)$;
$G_4 = G - E(W_4)$



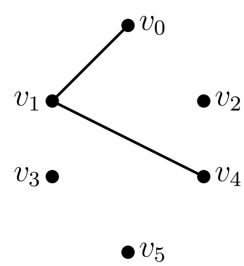
(e) $W_5 = W_4 \to (v_5, v_4)$;
$G_5 = G - E(W_5)$



(f) $W_6 = W_5 \to (v_4, v_2)$;
$G_6 = G - E(W_6)$



(g) $W_7 = W_6 \to (v_2, v_3)$;
$G_7 = G - E(W_7)$



(h) $W_8 = W_7 \to (v_3, v_4)$;
$G_8 = G - E(W_8)$

(i) $W_9 = W_8 \to (v_4, v_1)$; $G_9 = G - E(W_9)$

(j) $W_{10} = W_9 \to (v_1, v_0)$; $G_{10} = G - E(W_{10})$

Figure 5.5: Iterations of Fleury's algorithm.

- Look at the graph G_2 of Figure 5.5b. There are 3 edges incident with v_1. To proceed further, we cannot select the edge (v_1, v_0) (see Step 3 (ii)). We can select (v_1, v_3) or (v_1, v_4). We have decided to select (v_1, v_3) arbitrarily.

Theorem 5.2 (Correctness of the algorithm)**.** *Every trail constructed by Fleury's algorithm is a closed Eulerian trail.*

Proof. Let $W_p = (v_0, e_1, v_2, e_2, \ldots, e_p, v_p)$ be a trail generated by the algorithm.
Claim 1: W_p is a closed trail.

That W_p is a trail is obvious by Step 2, since in every iteration we select an edge which has not been selected in the earlier iterations. Moreover, by (Step 3 (ii)), there are no more edges incident with v_p. Therefore, if $v_0 \neq v_p$, then $deg_G(v_p) = 2k + 1$, where k is the number of times v_p appears internally in W_p, which is a contradiction. Hence, we conclude that $v_0 = v_p$.
Claim 2: W_p contains all the edges of G.

Assume the contrary and let $G_p = G - E(W_p)$. So, $E(G_p) \neq \emptyset$. Hence, there are vertices of positive degree in G_p. Moreover, every vertex in G_p has even degree, since G_p is obtained from G by deleting the edges of a closed trail (see exercise 7). Let

$S = \{v \in V(G_p) : deg_{G_p}(v) > 0\}$. Let $H = [S]$. Then $v_p \in V - S$, since $deg_{G_p}(v_p) = 0$, by Step 3 (i).

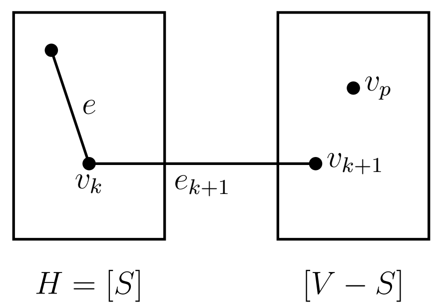


Figure 5.6: A step in the proof of Theorem 5.2.

Let v_k be the last vertex in W_p such that $v_k \in S$. Then $v_{k+1} \in V - S$ and $e_{k+1} = (v_k, v_{k+1})$ is the only edge joining S and $V - S$ in G_p. Therefore we conclude that

(1) e_{k+1} is a cut-edge of G_p.

Next, since every vertex of H has positive degree, there exists an edge e incident with v_k in H. It is not a cut-edge of H, since every vertex in H has even degree (see Exercise 7). e is not a cut-edge of G_p too, since $H \subseteq G_p$. While executing the $(k+1)$-th iteration, we have preferred to select e_{k+1} rather than e. Hence,

(2) e_{k+1} is not a cut-edge of G_p, by Fleury's rule Step 3(ii).

The conclusions (1) and (2) contradict each other. Hence, Claim 2 holds, and W_p is a closed Eulerian trail. $\square$

5.3 An application of Eulerian graphs: Chinese postman problem (Optional)

Problem: As 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.

This optimization problem was first discussed in a paper by Chinese mathematician Mei-Ku Kuan (1962) and hence the problem is named "Chinese Postman Problem".

Graph Theory Model: Given a connected weighted graph $(G, \mathcal{W})$, design an algorithm to find a closed walk of minimum weight containing every edge of G at least once.

It is obvious that if G is Eulerian then one can apply Fleury's algorithm and the resulting closed Eulerian trail is an optimal trail, since every edge appears exactly once. However, if G is non-Eulerian we can construct a super graph G^* which is Eulerian by duplicating certain edges. Note that by duplicating every edge of G, we get a super-graph of G which is Eulerian. So, the problem is to find an optimal set of edges in G whose duplication yields an Eulerian graph. The following algorithm is a solution to Chinese postman problem.

Algorithm:

Input: A connected weighted graph G.

Output: An optimal closed walk of G, containing every edge at least once.

Steps:

(1) If G is Eulerian, apply Fleury's algorithm.

(2) If G is not Eulerian, then identify all the vertices of odd degree, say $v_1, v_2, \ldots, v_{2k}$.

(3) Find a shortest (v_i, v_j)-path P_{ij} for every pair of vertices v_i and v_j by applying Dijkstra's algorithm or Floyd-Warshall algorithm. Let the weight of P_{ij} be $\mathcal{W}_{ij}$.

(4) Construct a complete graph G^* on vertices $z_1, z_2, \ldots, z_{2k}$ ($v_i \leftrightarrow z_i$) by joining z_i and z_j with an edge of weight $\mathcal{W}_{ij}$.

(5) (Matching problem) Find a set M of k edges say $\{(z_1, z_1^{'}), \ldots, (z_k, z_k^{'})\}$ in G^* such that

(i) no two edges are adjacent;

(ii) subject to (i), M has the minimum weight among all such sets of edges.

(6) In G, duplicate the edges of P_{ij} joining v_i and v_j if $(z_i, z_j) \in M$, to obtain an Eulerian super-graph G^e of G.

(7) Apply Fleury's algorithm to G^e. The resultant closed Eulerian trail is an optimal closed walk of G.

An illustration: Consider the road map G shown in Figure 5.7 with post office located at v_1.

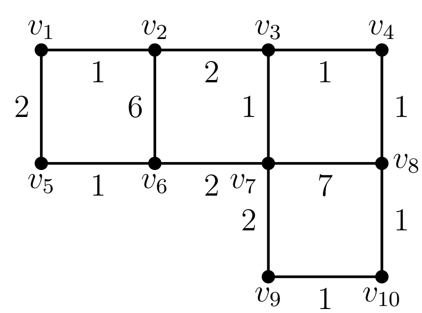


Figure 5.7: An input graph for illustration.

We apply the above seven steps to find an optimal walk containing each edge at least once.

(1) G is not Eulerian.

(2) v_2, v_3, v_6, v_8 are the vertices of odd degrees.

(3) (a) A shortest (v_2, v_3)-path is (v_2, v_3); its weight is 2.
(b) A shortest (v_2, v_6)-path is (v_2, v_1, v_5, v_6); its weight is 4.
(c) A shortest (v_2, v_8)-path is (v_2, v_3, v_4, v_8); its weight is 4.
(d) A shortest (v_3, v_6)-path is (v_3, v_7, v_6); its weight is 3.
(e) A shortest (v_3, v_8)-path is (v_3, v_4, v_8); its weight is 2.
(f) A shortest (v_6, v_8)-path is $(v_6, v_7, v_3, v_4, v_8)$; its weight is 5.

(4) The complete weighted graph G^* constructed by following the steps (3) and (4) is shown in Figure 5.8.

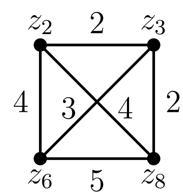


Figure 5.8: A complete weighted graph G^* .

(5) The following are the three sets of two non-adjacent edges in G*.

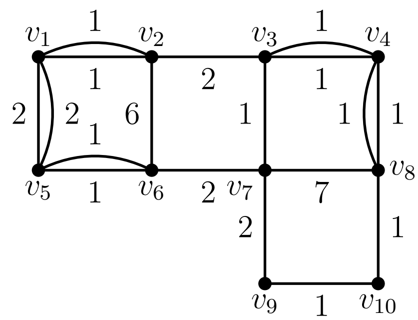
(a) $M_1 = \{(z_2, z_3), (z_6, z_8)\}$; its weight is 7.
(b) $M_2 = \{(z_2, z_6), (z_3, z_8)\}$; its weight is 6.
(c) $M_3 = \{(z_2, z_8), (z_3, z_6)\}$; its weight is 7.

M_2 has the minimum weight.

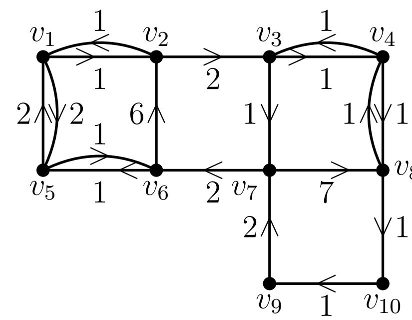
(6) We duplicate the edges of paths $P(v_2, v_6)$ and $P(v_3, v_8)$ in G and obtain the Eulerian graph G^e shown in Figure 5.9a.

(7) We apply Fleury's algorithm to obtain the following optimal closed Eulerian trail, which is a solution to Chinese Postman Problem.

(v_1, v_2) (v_2, v_3) (v_3, v_4) (v_4, v_8) (v_8, v_4) (v_4, v_3) (v_3, v_7) (v_7, v_8) (v_8, v_{10}) (v_{10}, v_9) (v_9, v_7) (v_7, v_6) (v_6, v_5) (v_5, v_1) (v_1, v_5) (v_5, v_6) (v_6, v_2) (v_2, v_1).



(a) G^e



(b) $W(v_1, v_1)$

Figure 5.9: The super-graph G^e of G constructed by following Step (6) and an Eulerian trail $W(v_1, v_1)$.

Exercises

1. Show that a connected graph G is Eulerian iff there exists a partition $(E_1, \ldots, E_k)$ of $E(G)$ such that each $[E_i]$ is a cycle.

2. (a) If in a graph G, every vertex has even degree, then show that G contains no cut-edges.

   (b) If G is Eulerian, show that $k_1(G)$ is even.

3. Draw a simple Eulerian graph G for some n $(4 \leq n \leq 10)$ with $k_0(G) = 1$ and $k_1(G) = 4$.

4. Draw a simple graph with $n = 7$, $\delta(G) \geq 3$ and containing no closed Eulerian trail but containing an open Eulerian trail.

5. Does there exists a simple Eulerian graph on even number of vertices and odd number of edges? Justify your answer.

6. Find the minimum number of edge disjoint trails together containing all the edges of G shown in Figure 5.10.

7. If W is a closed trail in a graph G, then show that $deg_G(x) \equiv deg_H(x) \pmod 2$, for every $x \in V(G)$, where $H = G - E(W)$.

8. Prove or disprove:

   (a) If G is Eulerian, then $L(G)$ is Eulerian.

   (b) If $L(G)$ is Eulerian, then G is Eulerian.

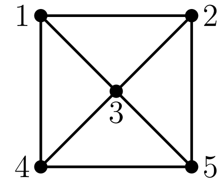


Figure 5.10: A graph G

9. If G is a connected graph, then show that $L(G)$ is Eulerian iff either every vertex in G has even degree or every vertex has odd degree.

10. Given a simple graph G, the p^{th} iterated line graph $L^p(G)$ is defined recursively as follows.
    (i) $L^1(G) = L(G)$,
    (ii) $L^p(G) = L(L^{p-1}(G)$; $p \geq 2$.
    (a) If G a is connected graph $(n \geq 5)$, then show that $L^3(G)$ is Eulerian implies $L^2(G)$ is Eulerian.
    (b) Prove or disprove: $L^2(G)$ is Eulerian $\Rightarrow L(G)$ is Eulerian.

11. If T is a tree, find the minimum number of edges to be added to T to obtain a spanning Eulerian supergraph G^e of T.

12. In the Fleury's algorithm, a graph G^e is constructed by adding the paths P_{ij} in Step (6). Show that G^e is Eulerian.

13. (a) Find conditions on r and s, for $K_r \square K_s$ to be Eulerian.
    (b) Find necessary and sufficient conditions that G and H should satisfy for $G \square H$ to be Eulerian.

14. Let k, n, p be integers ≥ 3. Find necessary and sufficient conditions for $(C_k + K_n) + K_p^c$ to be Eulerian.

15. (a) For which integers $a, b, c, d (\geq 1)$, $K_{a,b,c,d}$ is Eulerian.
    (b) For which values of n, $K_{1,3,5,\ldots,2n-1}$ is Eulerian? Justify.

16. Let G be a connected simple r-regular graph on even number of vertices such that its complement G^c is also connected. Prove or disprove: G or G^c is Eulerian.

17. Show that every connected graph contains a closed walk which contains every edge such that any edge appears at most twice.

18. Solve the Chinese postman problem for the street network shown in Figure 5.11.

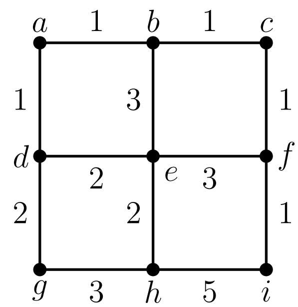


Figure 5.11: A street network.

19. A road map is shown in Figure 5.12. The central strip along each of these roads is to be painted white to have a smooth two-way traffic. The first coordinate in the ordered pair shown along an edge denotes the time for traveling and painting, and the second coordinate denotes the time for traveling without painting. Describe a shortest route if one starts painting from A and comes back to A after finishing the job. What is the total time required for such a route.

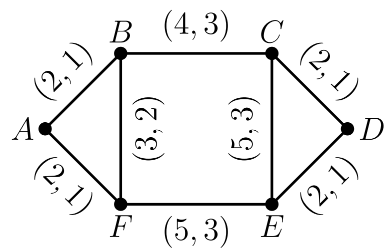


Figure 5.12: A road map.