Practices 3 and 4: Paths, connectivity and Eulerian graphs

(Degree in Computer Science Engineering)
Departamento de Matemática Aplicada
ETS Ingeniería Informática
Universidad Politécnica de Valencia

Contents

Paths

Definition (Path)

A **path** (of length n) in a graph G is an ordered sequence $v_0 e_1 v_1 e_2 \ldots e_n v_n$ such that:

- $v_0, v_1, \ldots, v_n$ are vertices of G,
- $e_1, e_2, \ldots, e_n$ are edges of G,
- v_{i-1} and v_i are the endpoints of e_i for all $i = 1, 2, \ldots, n$.

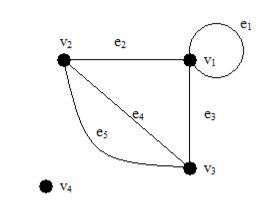
We say that v_0 is the **initial vertex** and v_n is the **final vertex** of the path.

Definition

- A path is **closed** if the initial and final vertices coincide.
- A path is said to be a **simple** if all its edges are different.

Example

Example



1. $v_1 e_1 v_1 e_3 v_3 e_4 v_2 e_5 v_3 e_3 v_1$ is a closed path. It is not simple because an edge is repeated.
2. $v_2 e_4 v_3 e_3 v_1$ is a simple path with v_2 as initial vertex and v_1 as final vertex.

Properties of paths

The following properties are satisfied for paths:

Properties

Let $G = (V, E, \Psi)$ be a graph. Suppose that $u, v, w \in V$:

- A vertex is always connected with itself by a path of length 0.
- If there is a path from u to v, and there is path from v to w, then there is a path from u to w.
- For non-directed graphs, if there is a path from u to v, then there is a path from v to u.

Connected components

Definition (Connected graph)

We say that two vertex v and w of a graph G are **connected** if there exists a path from v (initial vertex) to w (final vertex). If every pair of vertices of G is connected, then we say that G is a **connected graph**.

Definition (Connected component)

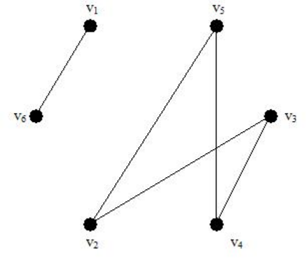
Given any vertex v of a graph G, the subgraph determined by all the vertices which are connected with v (and the edges that are incident with them) is a connected graph called **connected component** of G.

Therefore a graph is connected if and only if it has only one connected component.

Examples

Example

- The previous graph is not connected since, for example, v_4 and v_1 are not connected. It has two connected components. One is given by the vertex v_4, and the other by the other vertices and edges.
- The following graph has 2 connected components:



- One of the components is given by the vertex v_1 and v_6, and by the edge that joins them. The other component is given by the rest of the graph.

Eulerian graphs

Definition (Eulerian graph)

Let $G = (V, E, \Psi)$ be a graph. A path in G is said to be **Eulerian** if it simple and it contains all the edges of G (that is, if every edge of G appears exactly once). The graph G is said to be **Eulerian** if it contains an Eulerian **closed** path.

It is clear that an Eulerian graph must be connected.

Remark

The problem of the Königsberg bridges, written in these terms, consists on deciding if the corresponding graph is Eulerian or not.

The following theorem, due to Euler, provides a characterization of the Eulerian graphs and, therefore, gives an answer to the problem of the Königsberg bridges. (Which is the answer?)

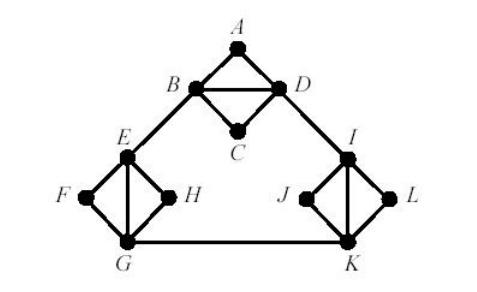
Eulerian graphs

Theorem (Euler Theorem (1))

Let $G = (V, E, \Psi)$ *be a connected graph. Then* G *is* **Eulerian** *if and only if* **all its vertices have even degree**.

Example

Example (Consider the following graph:)



Applying Euler's theorem we can deduce that it is Eulerian (since it is connected and all their vertices have even degree).

Hierholzer's algorithm

To find a **closed** Eulerian path we can apply Hierholzer's algorithm:

Suppose that we already know that the graph G has an Eulerian path.

Hierholzer's algorithm

1. Start at some vertex $v \in V$.
2. Add edges to the path until returning to v.
3. If we are not done, there exists a vertex $v' \in V$ that belongs to the current path but that has adjacent edges not in the path. Start another path from $v' \in V$, following unused edges until returning to v', and join the tour formed in this way to the previous tour.
4. Repeat the last step until you get the desired path.

Existence of an Eulerian paths

We can also pose a more general question:

Given a graph G, is there an Eulerian path (not necessarily closed) in G?

The Euler Theorem has a version for paths that answers this question:

Theorem (Euler Theorem (2))

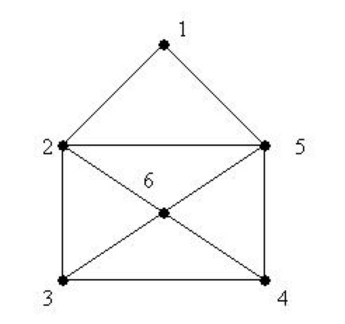
Let $G = (V, E, \Psi)$ be a **connected** *graph which is not Eulerian. Then G contains an* **Eulerian path (which is not closed)** *if and only if there exists* **exactly** *two vertices of odd degree.*

In this case, any Eulerian path has its endpoints at the vertices of odd degree.

Example

Example (Consider the following graph:)

This graph is not Eulerian. However, it admits an Eulerian path which is not closed (by the Euler's theorem because it has exactly two vertex of odd degree).



An example of an Eulerian path is the following:

32634652154

Algorithm to obtain a non-closed Eulerian path

If a connected graph has exactly 2 vertices of odd degree, we can obtain a non-closed Eulerian path using the following algorithm:

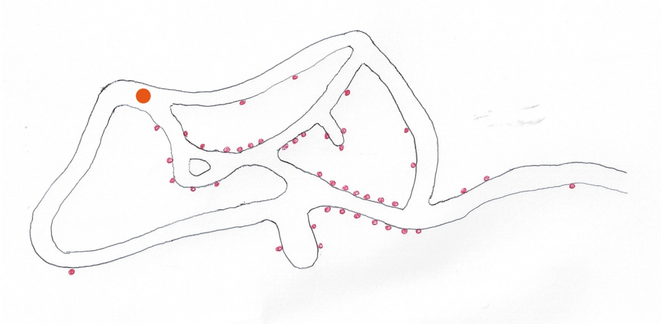
1. Add a new edge joining the two vertices of odd degree
2. The obtained graph is Eulerian (because now the degrees of all vertices are even). Hence, we can find a closed Eulerian path applying the Hierholzer algorithm.
3. Represent "circularly" the obtained closed Eulerian path.
4. Removing the initially added edge we will obtain a non-closed Eulerian path.

EXERCISE: Apply this algorithm to the previous example.

Chinese Postman Problem: statement

Suppose there is a mailman who needs to deliver mail to a certain village. The mailman is unwilling to walk far, so he wants to find the shortest route through the village, that meets the following criteria:

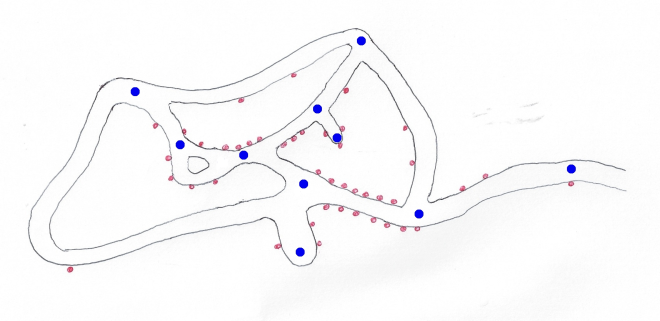
- It is a closed path (it ends at the same point it starts).
- He needs to go through every street at least once.



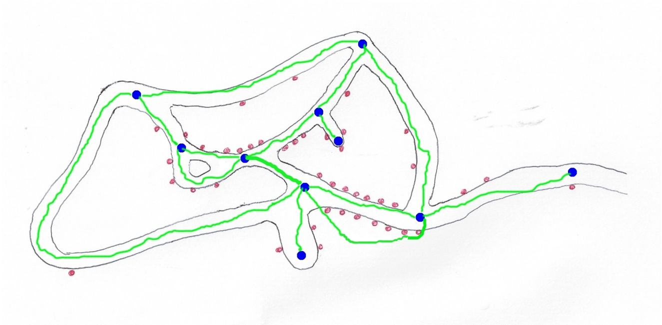
We have represented all the houses by small red circles. The post office is located near the orange circle.

Modeling the problem

The objective consists of modeling our problem by means of Graph Theory. We can consider the graph whose vertices represent the intersections of streets:



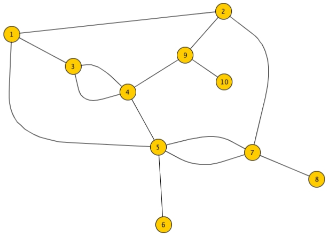
And the edges represent the streets:



OBSERVATION:

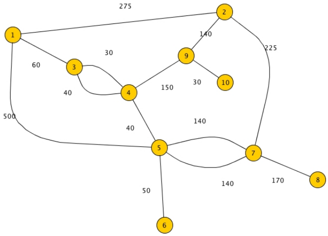
If a street has houses on both sides, it is enough for the postman to travel once down that street. The same occurs if there are houses on both sides but the street is short (the postman may deliver the mail walking in zig-zag pattern). However, if the street is long and it has houses at both sides, it is better to travel two times along the street (one for each side); we have represented this situation using two edges instead of one.

We have the following graph:



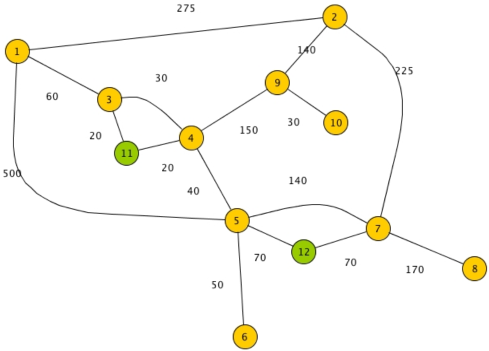
If the graph were Eulerian, then a closed Eulerian path would be the solution to our problem. However it is not Eulerian (there are vertices of odd degree). Therefore the mail carrier must retrace at least one street in order to cover the entire route. The problem can be solved by adding some parallel edges (the retraced streets) to the graph in such a way that the total length of the associated streets is the minimum possible and the obtained graph is Eulerian; then apply Hierholzer Algotihm. So we need to know the lengths of the streets.

Assigning to each edge the length of the corresponding street we obtain the following weighted graph:



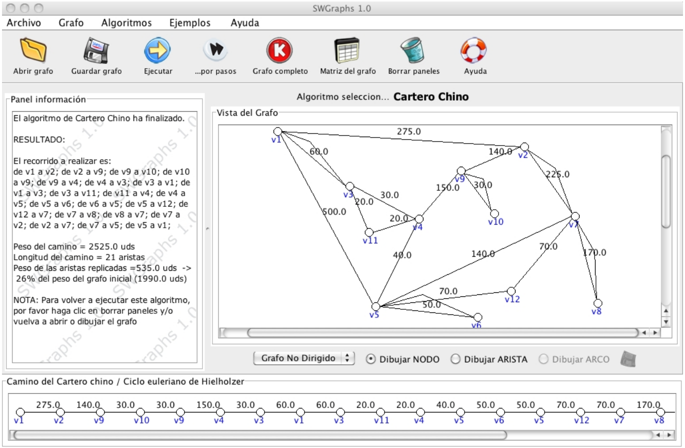
We will not study the details of the concrete algorithm to solve the Chinese Postman Problem. We leave to the computer (and the program SWGraphs) the "hard work".

SWGraphs does not admit parallel edges and we have parallel edges in our graph. This problem can be solved easily: if two vertices are joined by two parallel edges, we will add a **new** vertex in "the middle" of one of them (and the weight is transformed into two equal weights):



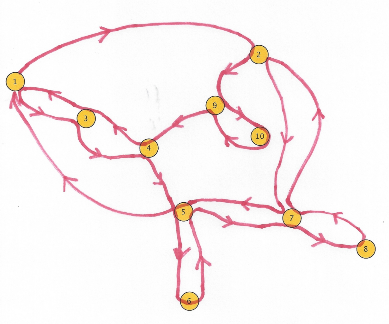
Resolution

Introduce the weighted graph in SWGraphs and apply the Chinese Postman Algorithm ("Cartero chino"):



Solution

More explicitly, the trail is the following one:



Observe that there are streets that are travelled twice.

Summary

The **Chinese Postman Problem** consists of, given a connected weighted graph, to find a closed path of minimum weight that contains all the edges of the graph.

If the graph is Eulerian, any closed Eulerian path is a solution. Otherwise some of the edges will be repeated.

This problem was studied by the chinese mathematician Meigu Guan in 1962. The name (Chinese Postman Problem) comes from this fact.