8 The Chinese postman problem

8.1 Definition

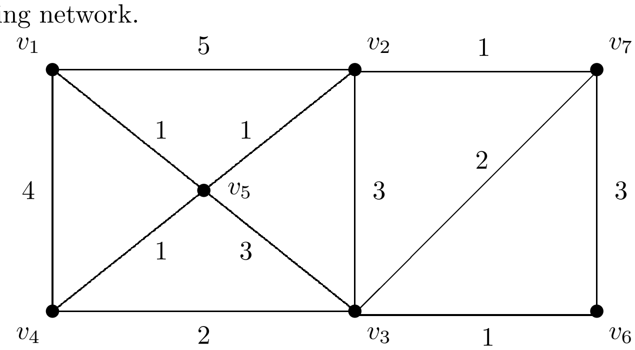
Let N be a network and W be a walk in N. Then the *length* of W, $length(W)$, is the sum of the weights of the edges of W, counting each edge the same number of times as it appears in W.

8.2 Problem

Let N be a connected network in which each edge has a positive integer weight. Find a closed walk in N which contains each edge of N at least once and is as short as possible. This problem is called the Chinese Postman Problem after a Chinese graph theorist, Guan, who gave a characterisation for a shortest closed walk which contains all edges of N in 1960. Note that if W is a closed walk which contains all edges of N then we must have $length(W) \geq w(N)$. Furthermore, if equality holds, then W contains every edge of N exactly once, and hence W is an Euler tour of N.

8.3 Example

Let N be the following network.



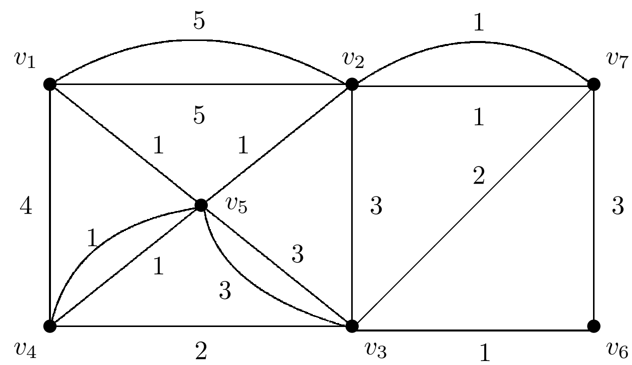
N has no Euler tour since it has four vertices of odd degree. Hence, if W is a shortest closed walk which contains all edges of N, then $length(W) > w(N)$.

8.4 Definitions

We shall call a closed walk in N, which contains all edges of N a *postman walk* for N. By an *extension* of N we shall mean an network N^* obtained from N by replacing each edge e of N by one or more *parallel edges*, i.e. edges with the same end vertices as e and the same weight as e. An *Eulerian extension* of N is an extension N^* such that each vertex of N^* has even degree.

8.5 Example

The following network N^* is an Eulerian extension of the network N in Example 8.3.



Note that an Euler tour R of N^* corresponds to a postman walk W in N which uses the edges $v_1v_2, v_2v_7, v_4v_5, v_5v_3$ twice and all other edges once. Thus

$$length(W) = length(R) = w(N^*) = w(N) + 5 + 1 + 1 + 3 = w(N) + 10.$$

8.6 Lemma

Let N be a network in which each edge has a positive integer weight, and let k be an integer. Then N has a postman walk of length k if and only if N has an Eulerian extension of weight k.

Proof (a) Suppose N has a postman walk W of length k. Construct an extension N^* of N by replacing each edge of N by $p(e)$ parallel edges, where $p(e)$ is the number of times e is contained in W. Then N^* is Eulerian since W corresponds to an Euler tour R of N^*. Furthermore $length(W) = length(R) = w(N^*)$. Hence N has an Eulerian extension of weight k.

(b) Suppose N has an Eulerian extension N^* of weight k. Let R be an Euler tour of N^*. Then R corresponds to a postman walk W for N with $length(W) = length(R) = w(N^*)$. Hence N has a postman walk of length k.

8.7 Corollary

The minimum length of a postman walk of N is equal to the minimum weight of an Eulerian extension of N.
Proof Take k as small as possible in Lemma 8.6

We will solve the Chinese Postman problem for a network N by finding a minimum weight Eulerian extension of N. We need the following elementary lemma.

8.8 Lemma

Let G be a graph and let S be the set of vertices of odd degree in G. Then $|S|$ is even.

Proof We have $\sum_{v \in V(G)} d_G(v) = 2|E(G)|$. Reducing both sides modulo two gives $|S| \equiv 0 \pmod 2$. Thus $|S|$ is even.

8.9 Edmond's Algorithm

The following strongly polynomial algorithm for solving the Chinese Postman Problem on a network was given by Edmonds in 1965. It uses his algorithm for finding a minimum weight perfect matching in a weighted complete graph, see Remark 6.15, and also Dijkrstra's algorithm for finding shortest paths, in order to find a minimum weight Eulerian extension of the network.

Let N be a network in which each edge has a positive integer weight.

Step 1 Let $S = \{v_1, v_2, \ldots, v_{2m}\}$ be the set of vertices of odd degree in N. For each $v_i, v_j \in S$ construct a shortest path $P_{i,j}$ from v_i to v_j in N.

Step 2 Construct a weighted complete graph K_{2m} with $V(K_{2m}) = S$ and $w(v_i v_j) = \mathit{length}(P_{i,j})$ for all $v_i, v_j \in S$. Find a minimum weight perfect matching M in K_{2m}.

Step 3 Construct an Eulerian extension N^* of N by replacing each edge $e \in E(P_{i,j})$ by two parallel edges, for all $v_i v_j \in M$.

Step 4 Construct an Euler tour T of N^*. Then T corresponds to a shortest postman walk W for N. Furthermore

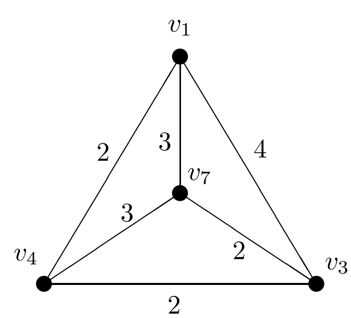
$$\mathit{length}(W) = \mathit{length}(T) = w(N^*) = w(N) + w(M).$$

8.10 Example

Let N be the network given in Example 8.3.

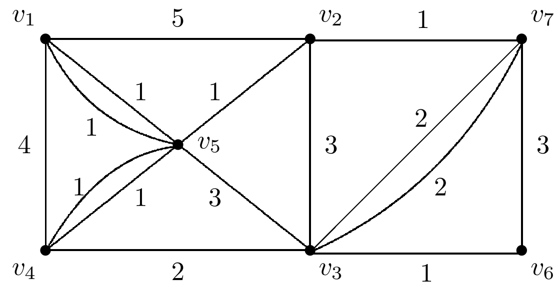
Step 1 The set of vertices of odd degree in N is $S = \{v_1, v_3, v_4, v_7\}$. The shortest paths in N joining the vertices of S are: $P_{1,3} = v_1 v_5 v_3$, $P_{1,4} = v_1 v_5 v_4$, $P_{1,7} = v_1 v_5 v_2 v_7$, $P_{3,4} = v_3 v_4$, $P_{3,7} = v_3 v_7$, $P_{4,7} = v_4 v_5 v_2 v_7$.

Step 2 Construct the weighted K_4 below:



A Minimum weight perfect matching is $M = \{v_1v_4, v_3v_7\}$ and $w(M) = 2 + 2 = 4$.

Step 3 Construct N^* by 'doubling' edges along $P_{1,4}$ and $P_{3,7}$.



Step 4 Construct an Euler tour R of N^*. For example

$$R = v_1v_2v_3v_4v_1v_5v_2v_7v_3v_7v_6v_3v_5v_4v_5v_1.$$

Put $W = R$. Then W is a shortest postman walk for N and $length(W) = w(N) + w(M) = w(N) + 4$.

We next show that the closed walk constructed by Edmond's algorithm is a shortest postman walk for N. We need the following lemma.

8.11 Lemma

Let G be a graph and let $S = \{x_1, x_2, \ldots, x_{2m}\}$ be the set of vertices of odd degree in G. Then G has a set of m pairwise edge-disjoint paths $\mathcal{P}$ such that each vertex in S is an end vertex of exactly one path in $\mathcal{P}$.

Proof We use induction on $|S|$. If $S = \emptyset$ then the lemma is trivially true, we just take $\mathcal{P} = \emptyset$. Hence we may suppose that $S \neq \emptyset$. Let H be a component of G which contains at least one vertex of odd degree. Applying Lemma 8.8 to H, we must have $|S \cap V(H)| \geq 2$. Choose $x_i, x_j \in S \cap V(H)$, and let $P_{i,j}$ be an x_ix_j-path in H. Let $G' = G - E(P_{i,j})$ and $S' = S - \{x_i, x_j\}$. Then S' is the set of vertices of odd degree in G'. By induction G' has a set of $(m-1)$ pairwise edge-disjoint paths $\mathcal{P}'$ such that each vertex in S' is an end vertex of exactly one path in $\mathcal{P}'$. Then $\mathcal{P} = \mathcal{P}' \cup \{P_{i,j}\}$ is the required set of m paths in G.

8.12 Theorem

Let N be a network in which each edge has a positive integer weight. Then Edmond's Algorithm constructs a shortest postman walk for N.

Proof By Corollary 8.7, it suffices to show that the extension N^* constructed by Edmond's algorithm is a minimum weight Eulerian extension of N.

Let $S = \{x_1, x_2, \ldots, x_{2m}\}$ be the set of vertices of odd degree in N. Let N^*_{min} be a minimum weight Eulerian extension of N and put $G = N^* - E(N)$. For each vertex v of G, we have $d_G(v) = d_{N^*_{min}}(v) - d_N(v)$. Since N^*_{min} is

Eulerian, all vertices of N^*_{min} have even degree. Thus $d_G(v) \equiv d_N(v) \pmod 2$ for all $v \in V(G)$. Hence the set of vertices of odd degree in G is again S.

By Lemma 8.11, G has a set of m pairwise edge-disjoint paths $\mathcal{P}$ such that each vertex in S is an end vertex of exactly one path in $\mathcal{P}$. Let N' be obtained from N by doubling edges along each path in $\mathcal{P}$. Then N' has no vertices of odd degree. Hence N' is an Eulerian extension of N which is contained in N^*_{min}. By the minimality of N^*_{min}, we must have $N^*_{min} = N'$.

Thus N^*_{min} can be obtained by choosing a set of m pairwise edge-disjoint paths such that each vertex in S is an end vertex of exactly one of the paths, and then 'doubling edges' along the paths. Furthermore, the sum of the lengths in N of the paths must be as small as possible. This corresponds to choosing a minimum weight perfect matching in the weighted K_{2m} constructed by Edmond's algorithm.

8.13 Note

We have not covered Edmond's algorithm for constructing a minimum weight perfect matching in K_{2m} in this course. Thus when asking you to apply Algorithm 8.9 to specific examples, I will ensure that the set S is quite small ($|S| = 2m \leq 4$), so that you can find a minimum weight perfect matching of the weighted K_{2m} by exhaustive search. Similarly, for small examples, you may be able to find shortest paths by inspection, rather than applying Dijkstra's algorithm.

8.14 Remark

It is straightforward to modify Algorithm 8.9 to:

- find a shortest trail which joins two specified vertices of a network N and contains all edges of N;
- find a shortest directed closed walk which contains all edges of directed network.