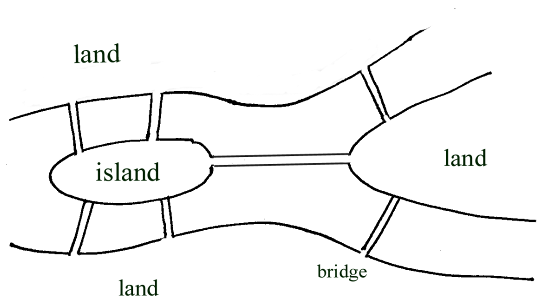
8. The Postman Problems

The Chinese postman problem (CPP) A postal carrier must pick up the mail at the post office, deliver the mail along blocks on the route, and finally return to the post office. To make the job easier and more productive, every postal carrier would like to cover the route with as little travelling as possible.

The problem is to determine how to cover all the streets assigned and return to the starting point in the shortest distance. If we construct a network $G = (V, E)$ in which each edge represents a street on which the mail must be be delivered and each vertex represents an intersection, this problem is equivalent to finding a cycle in G which travels each edge at least once in minimum total distance.

This problem was first considered by a Chinese mathematician, **Kwan Mei-Ko**, who has visited Singapore in the eighties. Examples of CPP include routing street sweepers, snowplows, interstate lawn movers, police patrol cars, electric line inspectors and automated guided vehicles in a factory or warehouse.

The Königsberg bridge problem The Königsberg bridges were represented by the following map. The problem was to determine whether one could traverse each bridge exactly once and return to the starting point.



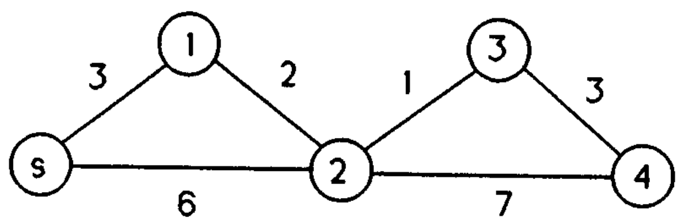
Representing the map as a graph (with possible multiple edges), **Leonard Euler** showed that this was impossible.

A cycle in a graph that crosses each edge exactly once is called an **Euler tour**. An graph that possesses an Euler tour is called an **Euler graph**. Note that an Euler tour exists in a postman problem if and only if the postman arrives at a vertex by an edge must be able to leave the vertex through another edge. This implies that each vertex has even number of edges incident to it (even degree).

Theorem An undirected graph is an Euler graph if and only if all vertices have even degree.

The graph for the Kőnigsberg bridge problem has four vertices of odd degrees, and consequently no Euler tour exists.

An example of the Euler graph is given in the following.



The graph has four Euler tours:

$$
\begin{aligned}
&s - 1 - 2 - 3 - 4 - 2 - s \\
&s - 1 - 2 - 4 - 3 - 2 - s \\
&s - 2 - 3 - 4 - 2 - 1 - s \\
&s - 2 - 4 - 3 - 2 - 1 - s
\end{aligned}
$$

An algorithm for finding Euler tours

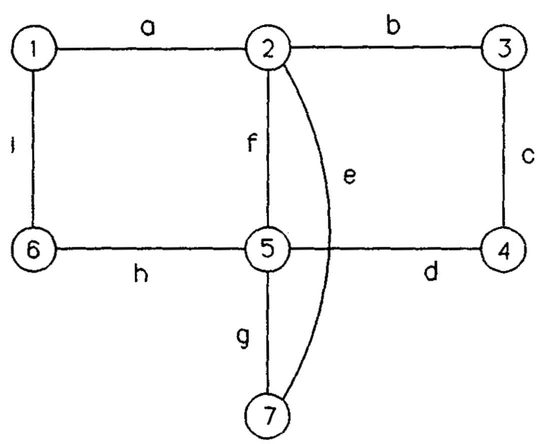
Step 1 Begin at any vertex s and construct a cycle C. Traverse to any edge (s,x) incident to s and mark the edge (s,x). Next traverse to any unmarked edge incident to x. Repeat this process of traversing unmarked edges until return to vertex s.

Step 2 If C contains all the edges of G, then C is an Euler tour. If not, by removing all the edges of C from G, we get a smaller Euler graph G'. Since G is connected, G' has at least one common vertex v with C.

Step 3 Starting at v, construct a cycle C' in G'.

Step 4 Splice together the cycles C and C' to obtain a new cycle C. Return to Step 2.

Example Apply the algorithm to the following network.



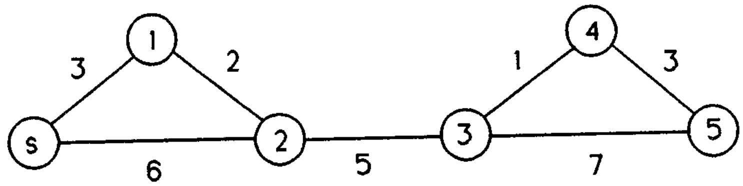
We may have C : a, f, h, l and C' : b, c, d, g, e. Both are cycles. We may splice them together to form a new cycle.

If a network has an Euler tour, then the Chinese postman problem is solved.

If a network has no Euler tour, then at least one edge must be crossed more than once. In a vehicle routing context, we call this

deadheading, since the vehicle is not performing any productive work.

The following network does not have any Euler tour.



A. The postman problem for undirected network

Suppose $G = (V, E)$ is not an Euler graph. Let $a(i, j)$ be the length of the edge (i, j) in G. Let $f(i, j)$ be the number of times that the edge (i, j) needs to be **repeated** by the postman. Construct a new graph G^* that contains $f(i, j) + 1$ copies of (i, j) in G. Then G^* is an Euler graph.

The postman wishes to select the repeated edges (namely, determine values of $f(i, j)$ for all (i, j)) so that

- G^* is an Euler graph.
- The total length $\sum_{(i,j)\in E} a(i, j)f(i, j)$ of repeated edges is minimum.

On the repeated edges

- If vertex x is an odd-degree (respectively even degree) vertex in G, an odd (respectively even) number of edges incident to

x must be repeated by the postman, so that the x has even degree in G^*.

- If we trace out from an odd-degree vertex by a path of repeated edges as far as possible, then the path must end at another odd-degree vertex.

The following theorem is proposed by Kwan Mei-Ko, which suggests a method for solving the postman problem.

Theorem A Euler tour of the postman problem is optimal if and only if

(i) no more than one duplicate edge is added to any edge in G.

(ii) the length of the added edges in any cycle does not exceed half of the length of the cycle.

An undirected postman algorithm

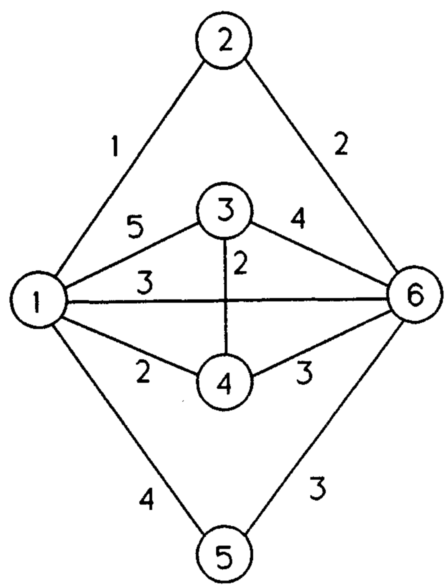
1. Find a shortest path between every pair of odd-degree vertices in G.

2. Construct a network G' whose vertex set consists of all odd-degree vertices in G and whose edge set consists of all edges joining each pair of vertices with shortest distances as their weights.

3. Determine a minimum-weight matching of G'.

4. The edges in a shortest path joining a matched pair of odd-degree vertices will be repeated by the postman.

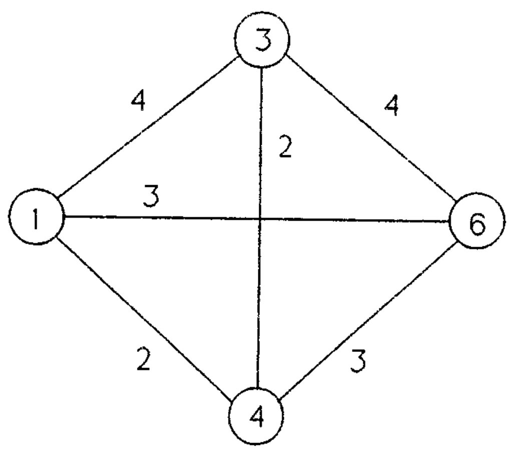
Example Determine an optimal postman route in the following network.



Applying the Floyd-Warshall algorithm (or by intuition), we have the shortest distance matrix:

	1	2	3	4	5	6
1	0	1	4	2	4	3
2	1	0	5	3	5	2
3	4	5	0	2	7	4
4	2	3	2	0	6	3
5	4	5	7	6	0	3
6	3	2	4	3	3	0

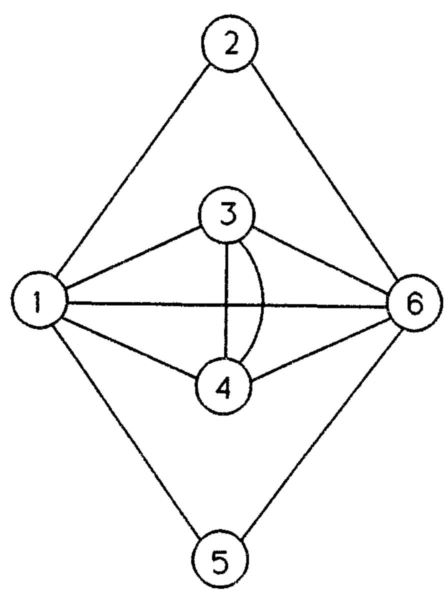
The vertices $1, 3, 4, 6$ are odd-degree vertices. Using the shortest distances as weights, we obtain the following network G'.



By observation we have the following matching for vertices $1, 3, 4, 6$.

Matching	Weight
$1-3,\ 4-6$	$4+3=7$
$1-4,\ 3-6$	$2+4=6$
$1-6,\ 3-4$	$3+2=5$

Note that $1-6,\ 3-4$ is an optimal matching. By adding the edges along the paths connecting 1 and 6, 3 and 4, we get the Euler network G^*.



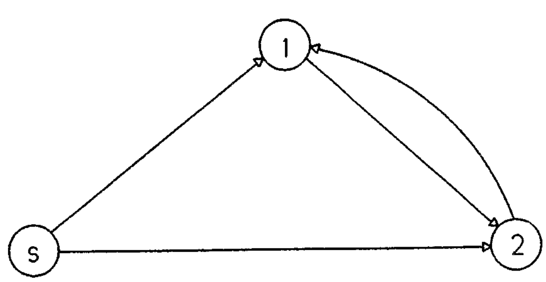
Then by applying the algorithm for finding the Euler tours in G^*, we have an optimal route $1-2-6-5-1-3-6-4-3-4-1-6-1$, which traverses each edge in G^* exactly once and traverses each edge in G at least once. Only edges $(1,6)$ and $(3,4)$ are repeated in G.

B. The postman problem for directed network

Let $G = (V, E)$ be a directed network. For each vertex x, the number of arcs with x as end vertex is called the **in-degree** of x, and is denoted as $d^-(x)$. Similarly, we denote $d^+(x)$, the **out-degree** of x, as the number of arcs with x as the starting vertex. If $d^-(x) = d^+(x)$ for all $x \in V$ then G is called symmetric.

Unlike the postman problem for undirected networks, the postman problem may have no solution for a directed network.

Remark: A postman problem has no solution if and only if there exists a nonempty proper subset S of V such that the cut $(S, \bar{S})$ is empty.



The number of times a postman enters a vertex must equal the number of times the postman leaves the vertex. Therefore,

- a postman must repeat some of the arcs leaving the vertex x if $d^-(x) > d^+(x)$.
- a postman must repeat some of the arcs entering the vertex x if $d^-(x) < d^+(x)$.

We consider the following two cases:

Case 1: The network G is symmetric
An Euler tour in G can be found by using the techniques similar to the one used for the undirected graph.

Case 2: The network is not symmetric
Let $f(i, j)$ be the number of times that the postman repeats arc (i, j). Since the postman enters and leaves each vertex x the same number of times, we have

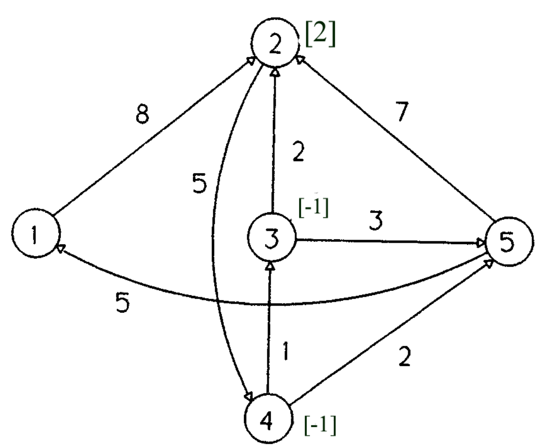
$$d^-(i) + \sum_j f(j, i) = d^+(i) + \sum_j f(i, j)$$

for all vertex $i \in V$. Let $D(i) = d^-(i) - d^+(i)$. Thus the postman wants to find nonnegative integers $f(i, j)$ solving the following problem:

$$\begin{array}{ll} \min & \Sigma_{(i,j)}\, a(i, j) f(i, j) \\ \text{subject to} & \Sigma_j\, f(i, j) - \Sigma_j\, f(j, i) = D(i) \quad \text{for all } i \in E \end{array}$$

The minimization problem is merely a minimum cost flow problem. Let f^* be an optimal solution for the problem. Then $f^*(i,j)$ is integer for all (i,j). Hence create a graph G^* with $f^*(i,j)+1$ copies of arc (i,j) for all $(i,j) \in E$. The graph G^* is symmetric. Hence as in Case A we can find an Euler tour for G^*, which gives an optimal postman route for G.

Example Find an optimal postman route of the following network.



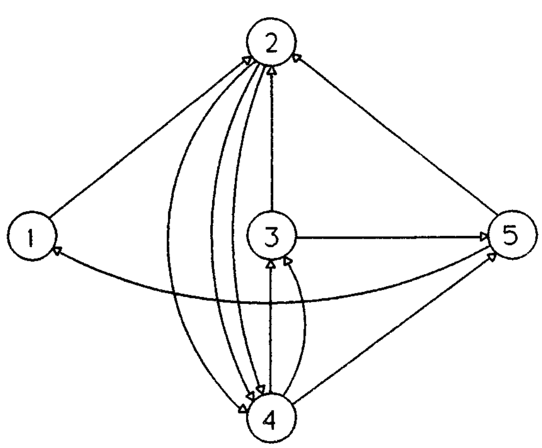
By computing the in-degree and out-degree of each vertex, we have

$Vertex$ i	$In - degree$ $d^-(i)$	$Out - degree$ $d^+(i)$	$Net\ Supply$ $D(i)$
1	1	1	0
2	3	1	2
3	1	2	-1
4	1	2	-1
5	2	2	0

Solving the minimum cost flow problem, we have the optimal solution:

$f(4,3) = 1$; $f(2,4) = 2$ and $f(i,j) = 0$, otherwise.

Thus we add one arc $(4,3)$ and two arcs $(2,4)$ to the network to obtain a symmetric graph G^* as below.



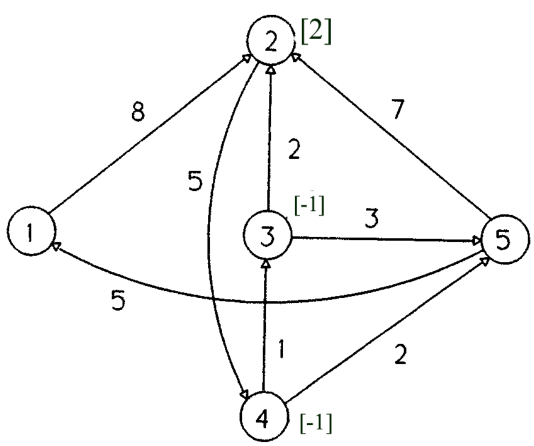
There exists an Euler tour for G^* and hence an optimal postman route for G.

$$1-2-4-3-2-4-3-5-2-4-5-1$$

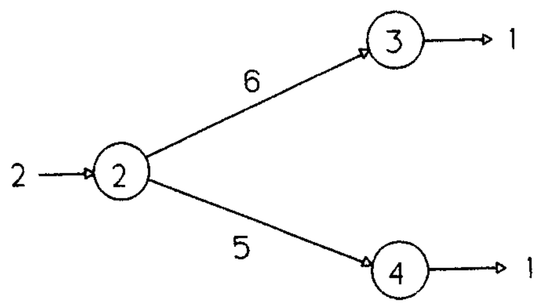
Formulate the postman problem as a transportation problem

let $S = \{x \in V | D(x) > 0\}$ and $T = \{x \in V | D(x) < 0\}$. Consider each vertex in S as a source and each vertex in T as a destination. For each $i \in S$ and $j \in T$, determine the shortest distance $c(i,j)$ from i to j. Then the postman problem is converted into the following transportation problem.

min	$\Sigma_i \Sigma_j c(i,j) g(i,j)$	
subject to	$\Sigma_j g(i,j) = D(i)$	for all $i \in S$
	$\Sigma_i g(i,j) = D(j)$	for all $j \in T$
	$g(i,j) \geq 0$	for all i, j



In the network, vertex 2 is a source and vertices 3 and 4 are destinations. The shortest distances $c(2,4) = a(2,4) = 5$ and $c(2,3) = a(2,4) + a(4,3) = 5 + 1 = 6$. We have a simple transportation problem:



The obvious solution gives $g(2,4) = g(2,3) = 1$. Hence we add one copy of arc $(4,3)$ and two copies of arcs $(2,4)$ to G to obtain a postman route for G.

C. The postman problem for mixed network

Let G be a network with both directed arcs and undirected edges. G is called **even** if every vertex has even degree (ignoring directions). Consider the following three cases:

Case 1: G is even and symmetric.

Case 2: G is even but not symmetric.

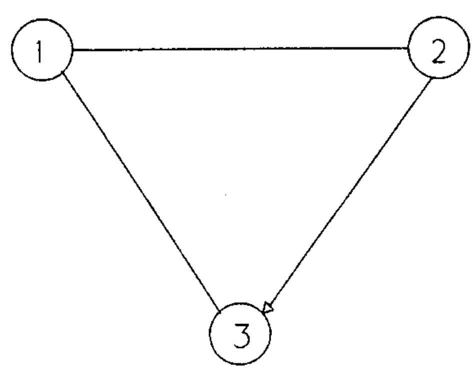
Case 3: G is not even.

Case 1. Since G is symmetric, generate a cycle of directed arcs of G. Repeat this procedure until all directed arcs have been used.

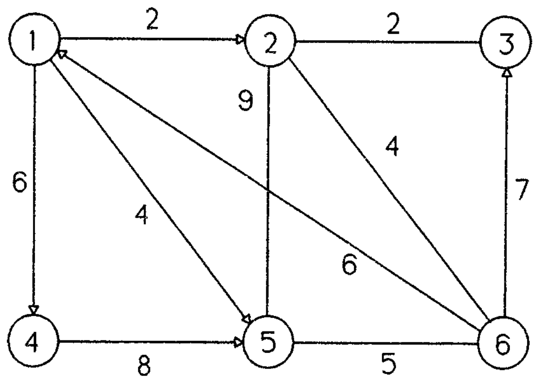
Next, generate a cycle using only the undirected arcs in G. Repeat this procedure until all arcs in G have been used.

Splice together all the cycles generated above into one cycle C. Then C forms a Euler tour of G and hence obtain an optimal solution to the postman problem.

Case 2. It is not easy to know in advance if the postman must repeat any arc or edge. For example, there exists an Euler tour for the following network.



On the other hand, the following network is an even, non-symmetric network. It has no Euler tour, since the arc $(6, 1)$ must be repeated twice so that the postman can exit vertex 1 along arcs $(1, 2), (1, 4)$ and $(1, 5)$.



A mixed postman algorithm

Stage 1

Let U and W be the set of all undirected and directed edges in G respectively. Select a tentative direction for each edge in U. Denote the resulting directed graph by G_D. For each vertex i, define

$$D(i) = d^{-}(i) - d^{+}(i).$$

If $D(i) = 0$ for all $i \in G_D$, then G_D is symmetric. Therefore there is an Euler tour for G_D and hence obtain an optimal postman route for G. If G_D is not symmetric, go to stage 2.

Stage 2

Construct a directed network $G' = (V, E')$ as follows:

(a) For each arc $(i, j) \in W$, set $(i, j) \in E'$ with the same weight and infinite capacity.

(b) For each edge $\{i, j\} \in U$, create two directed arcs $(i, j), (j, i)$ in E' with the same weight and infinite capacity.

(c) For each arc (i, j) in G_D, which is obtained from U by a tentative assignment of direction, we assign an artificial arc $(j, i)'$ in E' with weight zero and capacity 2 .

This define a minimum cost flow problem on G'. If no optimal flow can be found in G', then no postman route exists. If an optimal flow f in G' exists, then go to stage 3.

Stage 3

Construct a network G^* as follows:

(i) For each non-artificial arc (i, j) in G', make $f(i, j) + 1$ copies of arc (i, j) in G^*.

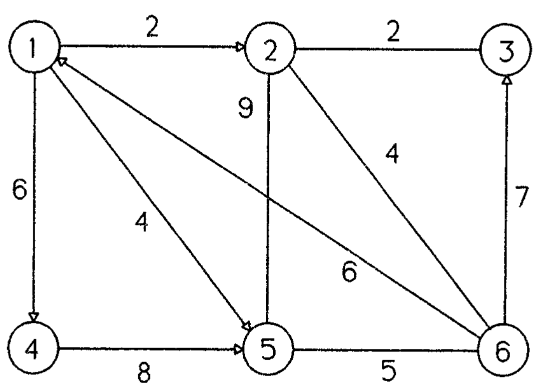
(ii) If $f(i, j)' = 2$ for artificial arc $(i, j)'$, make one copy of (i, j) in G^*. (*This means that if two flow units traverse an artifi-*

cial arc, the tentative direction assigned to this arc in G_D is reversed)

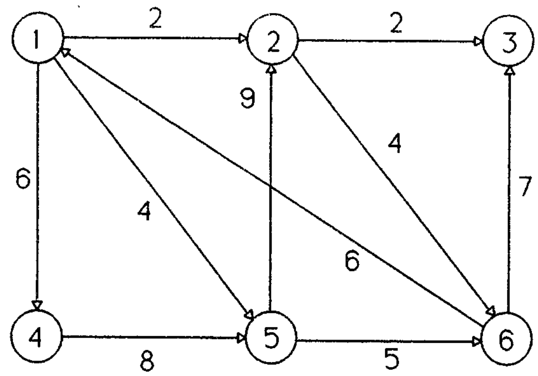
(iii) If $f(i,j)' = 0$ for artificial arc $(i,j)'$, make one copy of (j,i) in G^*. (*This means that if no units traverse an artificial arc, the tentative direction assigned to this arc in G_D is retained*)

G^* is a symmetric directed network. Then we can find an Euler tour for G^* and hence obtain an optimal postman route for G.

Example Consider the network:



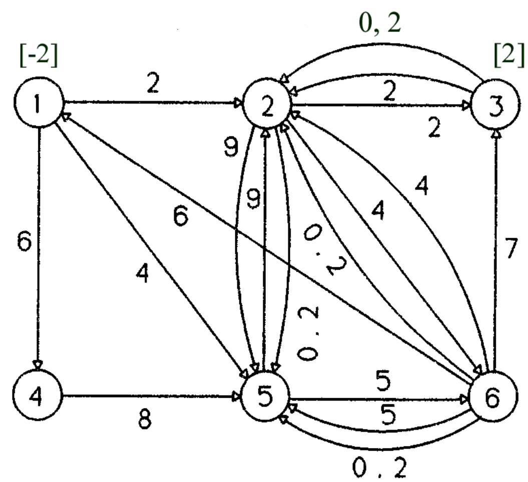
Assign a tentative direction to all undirected edges. We have the resulting network G_D



Compute the net supply (or demand) for each vertex i.

i	$In - degree$ $d^-(i)$	$Out - degree$ $d^+(i)$	$Net\ \ Supply$ $D(i)$
1	1	3	-2
2	2	2	0
3	2	0	2
4	1	1	0
5	2	2	0
6	2	2	0

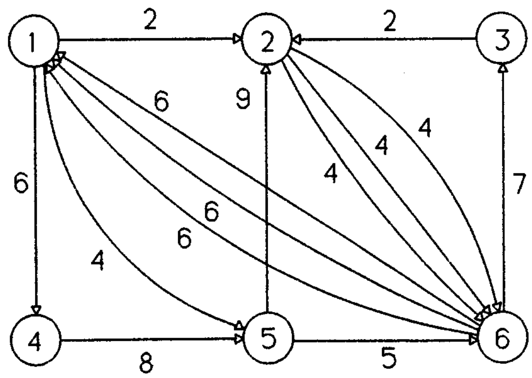
We construct the network G' according to procedures as described in (a), (b) and (c) above.



It is now required to find a minimum-cost way of sending 2 flow units from supply vertex 3 to demand vertex 1. By inspection, the minimum-cost path from vertex 3 to vertex 1 is the shortest path $3 \to 2 \to 6 \to 1$. Therefore the flow is given by,

$$f(3,2)' = f(2,6) = f(6,1) = 2,$$
$$f(i,j) = 0, \quad otherwise.$$

Thus the arcs $(2,6)$ and $(6,1)$ must be repeated twice since $f(2,6) = f(6,1) = 2$. On the other hand, $(3,2)'$ is an artificial arc. Hence $f(3,2)' = 2$ implies that the tentative direction in G_D of the undirected arc $(2,3)$ must be reversed to $(3,2)$. The resulting network G^* is given below.



G^* is symmetric and hence there exists an Euler tour for G^*. This induces an optimal postman route for G.

Case 3. No efficient algorithm is available. The general approach is to make the graph even in an optimal way. Then make the network symmetric and hence solve the problem by finding an Euler tour.