Lab Assignment 11 - Routing

Jens Clausen, David Pisinger,

Section for OR, DTU Management,

Francesc/François Antón Castro ([fa@space.dtu.dk](mailto:fa@space.dtu.dk)), DTU Space

The text “GIS i Danmark 2”, pages 125 - 134 can be downloaded from

Campusnet (file “Ruteoptimering.pdf”).

Exercise 1 - Shortest Path Trees

In “GIS i Danmark 2”, page 126 you will find a map and a distance table for Bornholm. On page 130 there is a description of Dijkstra’s Single Source Shortest Path algorithm - this is also given in the slides for the lecture. Find with Ronne as source (origin) the shortest path to each of the other locations on Bornholm.

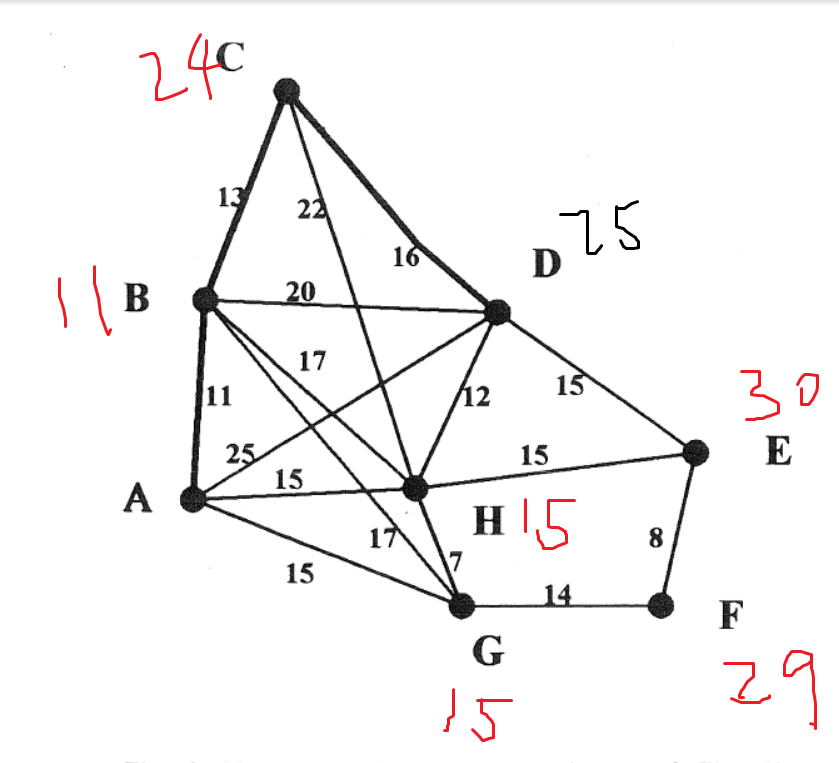
Find the shortest path from Ronne (A) to each of the other locations on Bornholm following Dijkstra’s algorithm, we should first set up the conditions:

1. all edge costs are non-negative
2. only consider each vertex once
3. select vertices that is not part of solution and with the shortest distance
4. add the vertices to set S (solution) when considered
5. check “shortcuts” for all outgoing edges from a vertices

According to the algorithm, the initial values are: ds🡸0, di🡸∞, where ds is the distance at starting vertex and di is all other vertices. Regarding No.3 above, we have to select all the vertices (as no vertices is in S) and find the shortest distance from A to one of them, while always bear in mind No.5, so we get:

AB=11 AD=25 AC=24 AE=30 AF=29 AH=15 AG=15

AB has the shortest path, we will update dB 🡸 ds+csB =11, and take the first vertex into S, so that we don’t use this vertex in next for loop.



Exercise 2 - Minimum Spanning Trees

A tree is a set of edges in a graph containing no circuit. A spanning tree includes all vertices. A minimum spanning tree is a tree with minimum weight among all spanning trees. Minimum spanning trees are interesting e.g. in connection with high-capacity backbone communication networks. Find a minimum spanning tree for Bornholm with the given locations and edges. Describe your method.

There are a couple of algorithms for finding Minimum Spanning Trees (MST), namely Kruskal’s and Prim’s. We introduce Prim’s algorithm in Bornholm’s case here. It starts out on a single –vertex tree, then add V-1 edges to it, find the shortest edge that connects a vertex on the tree to a vertex not yet on the tree. Then, for each iteration of the step, the tree will grow one more vertex. Remember that we are finding a tree, so the edges that form a circuit with the found edges should be eliminated.

For instance, we start at A, there are four vertices connect directly with A:

AB=11 AD=25 AH=15 AG=15

We see AB has the shortest path, we add B to the tree. Now we see there are seven vertices connect with A and B:

CB=13 DB=20 HB=17 GB=17 DA=25 HA=15 GA=15

The shortest path is CB, so we add C to the tree, There are 8 vertices connect with A,B and C:

DC=16 HC=22 DB=20 HB= 17 GB=17 DA=25 HA=15 GA=15

HA and GA has equal length, we can choose one of them. We iterate the step until all the vertices are included, finally we end up with a tree (see figure )

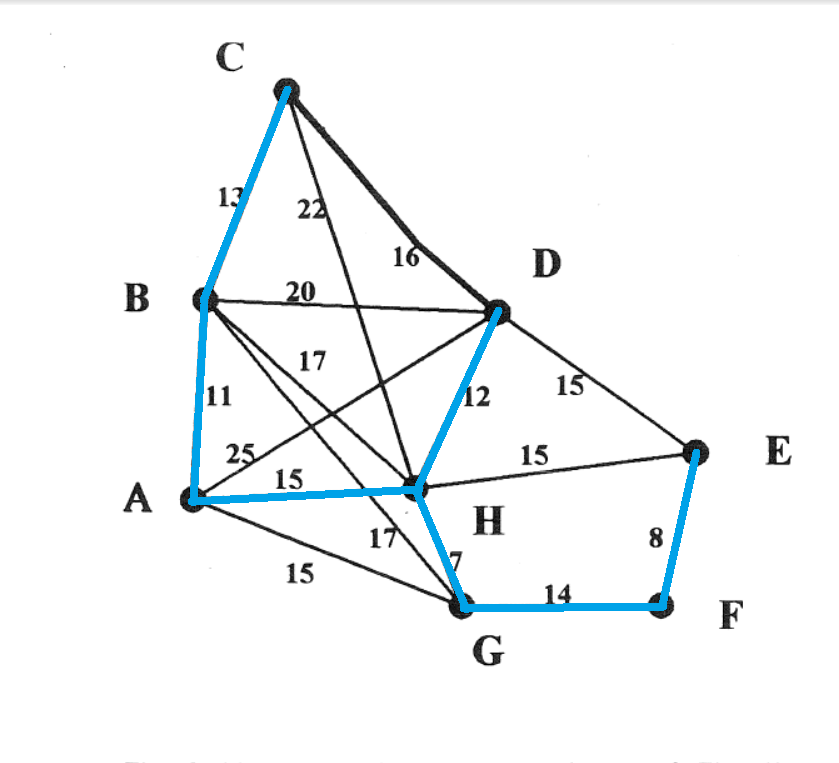


Figure 1 Minimum spanning tree

Exercise 3 - the Travelling Salesman

Find the shortest Hamiltonean circuit on Bornholm, i.e. the shortest circuit visiting each city at least once. Argue that your solution is actually the shortest circuit.

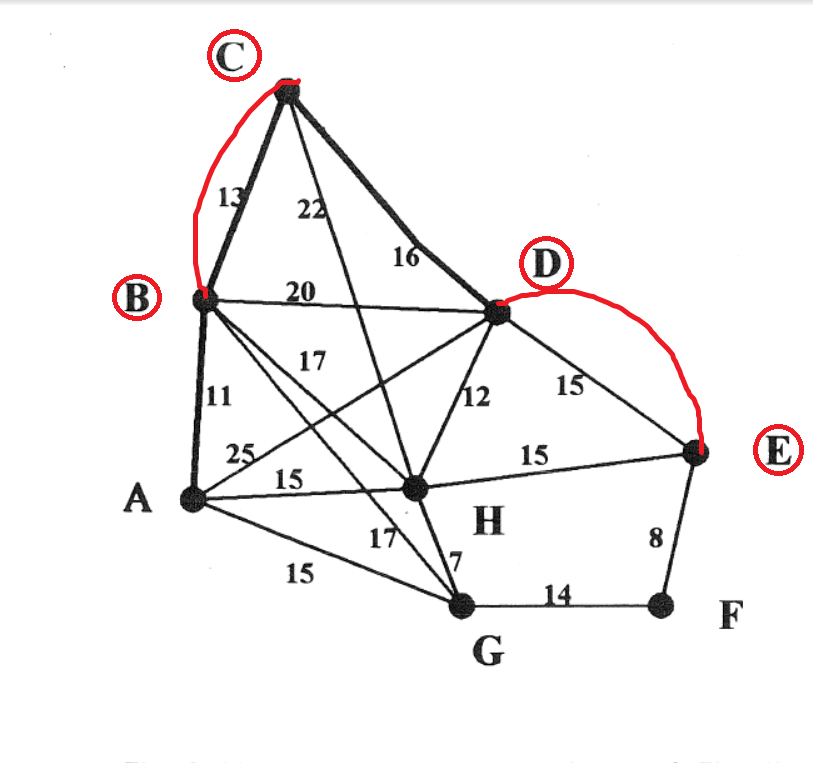
Exercise 4 - the Chinese Postman

The shortest postman tour in the Bornholm network can be found as follows:

* Find all vertices of odd degree, i.e. with an odd number of incident edges.
* Construct a complete network of these vertices - the distance between each pair of vertices must be the one given in the distance table of Figure 1.
* Match the vertices in pairs, such that the sum of the distances between matched vertices is as small as possible.
* Identify the paths corresponding to the edges in the matching. Add each of these paths to the network, even if it is already present.
* Now each vertex is of even degree - find a circuit in the network visiting each edge exactly once (a Eulerian tour).

Carry through this construction.

Deliverables: A report containing the answers 1-4.



First of all, we find that B, C, D, E has odd degree.

Then we construct a complete network of these vertices by pairing each of them, we get six pairs in total:

BC=13 BD=20 BE=32 CD=16 CE=31 DE=15

There is no direct path between B and E, We have to take one of the two options namely BDE or BHE. Since the later one gives shorter path, we chose it.

The six pairs of vertices can make 3 pairs of sum:

BC+DE=28 BD+CE=51 BE+CD=48

We see that BC and DE gives the smallest value. In order to make an Eulerian trail, all the vertices must be of even order, and all the edges must be used. Thus, we have to add the pair of sum that gives the smallest value to the original graph, which is BC and DE.

The length of an optimal Chinese postman route is the total length of edges plus BC+DE, which is 170+28=198.

Now each vertex is of even degree, the number of times each edge will appear in a Chinese postman route is half the order of its vertex, with A exceptional as the route ends at A, so A will appear on one extra occasion. The orders look like:

|  |  |  |
| --- | --- | --- |
| vertex | Order | Times appear in route |
| A | 4 | 2+1=3 |
| B | 6 | 3 |
| C | 4 | 2 |
| D | 6 | 3 |
| E | 4 | 2 |
| F | 2 | 1 |
| G | 4 | 2 |
| H | 6 | 3 |

The total number that the vertices will appear in the route is 19, the route is ABCDBCHADHBGHEDEFGA.

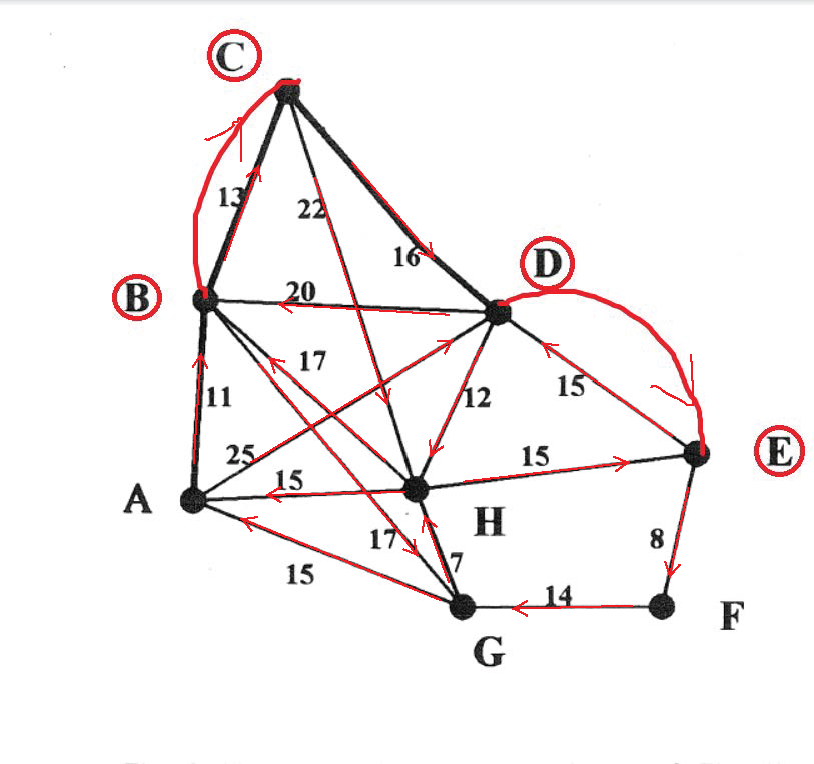


Figure 2 Chinese postman route