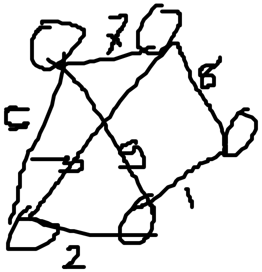
TSP



Traveling Salesman Problem

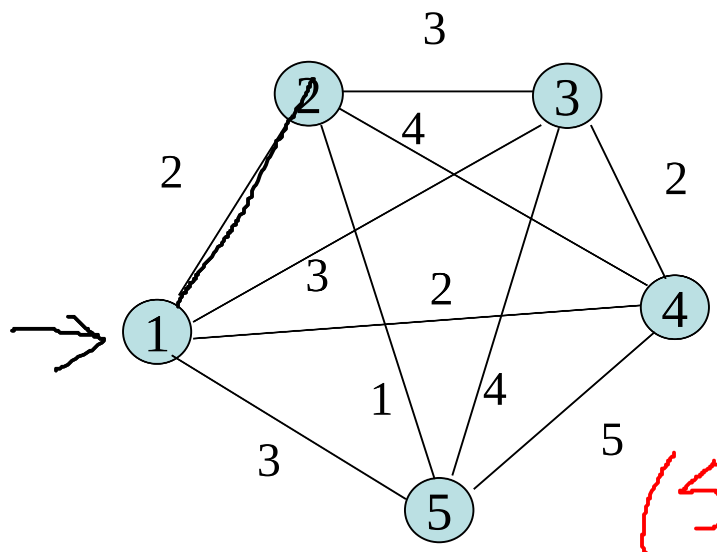
In this problem, a traveling salesman wishes to visit several given cities and return to his starting point, covering the least possible total distance.

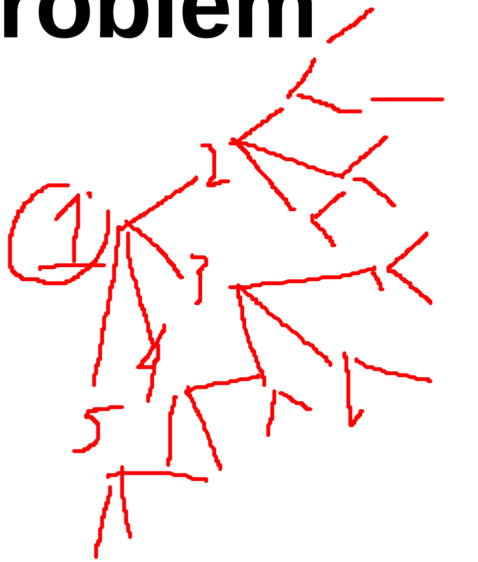
The TSP involves finding the minimum traveling cost for visiting a fixed set of customers.

The vehicle must visit each customer exactly once and return to its point of origin also called depot.

The objective function is the total cost of the tour.

Traveling Salesman Problem





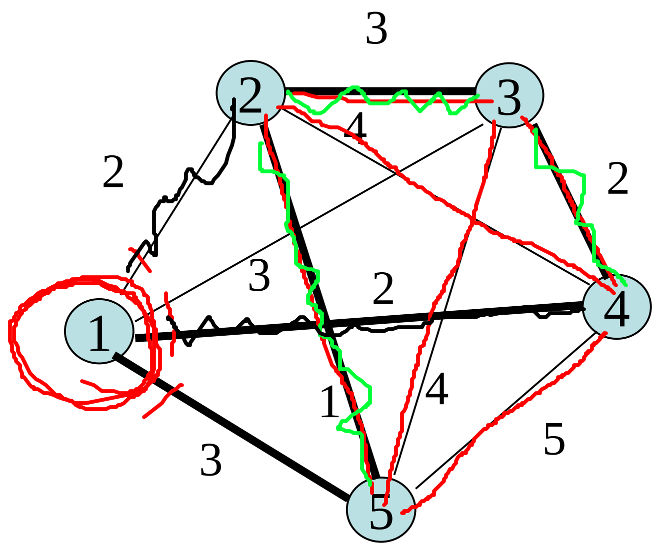
$(5-1)\times 3\times 2\times 1$

The total number of solutions is (n-1)! /2 if the distances are symmetric.

For example, if there are 50 customers to visit, the total number of solutions is 49!/2=3.04x10^{62}.

Traveling Salesman Problem

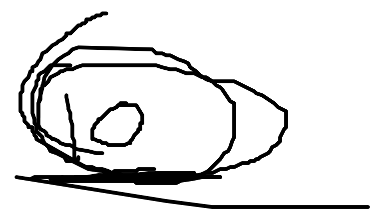
Solution



$1+2+3=6$

$w_1 + w_2$

$= 2 + 2 = 4$



If the depot is located at node 1, then the optimal tour is 1-5-2-3-4-1 with total cost equal to 11.

- Various algorithms have been proposed but they take too long to apply. There are several heuristic algorithms that quickly tell us approximately what the shortest distance is.

The travelling salesperson problem can then be more mathematically formally stated as:

Given a weighted, complete graph, find a minimum-weight Hamiltonian cycle.

The Travelling Salesperson Problem: Complexity

The travelling salesperson problem is quite complex due to the fact that the numbers of cycles to be considered *grows rapidly* with each **extra city** (vertex).

If for example the salesperson wanto to visit 100 cities, then

$$\frac{1}{2}(99!) = 4.65 * 10^{155}$$

cycles need to be considered.

This leads to the following problems:

- There is **no known efficient algorithm** for the travelling salesperson problem.
- Thus we just look for a good (!) **lower and/or upper bounds** for the length of a *minimum-weight Hamiltonian cycle.*

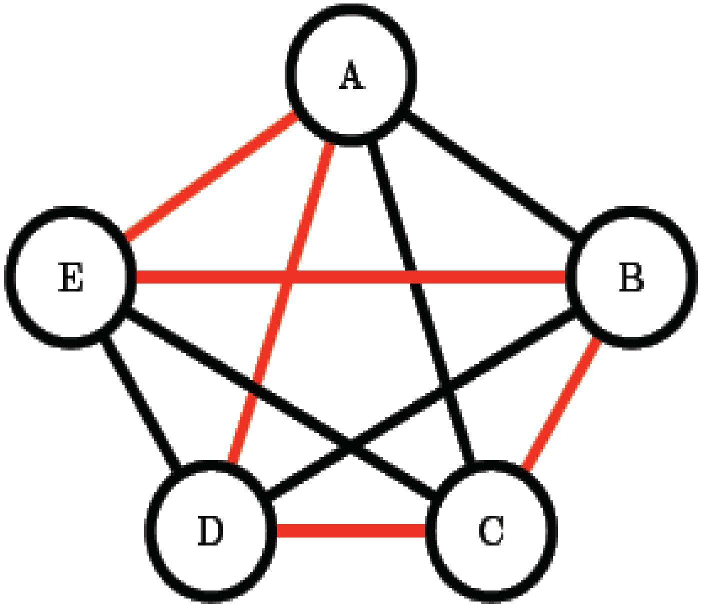
Lower bound for the travelling salesperson problem

To get a better approximation for the actual solution of the travelling salesperson:

it is useful to get *not only* an **upper bound** for the solution **but also a lower bound**.

The following example outlines a simple method of how to obtain such a lower bound.

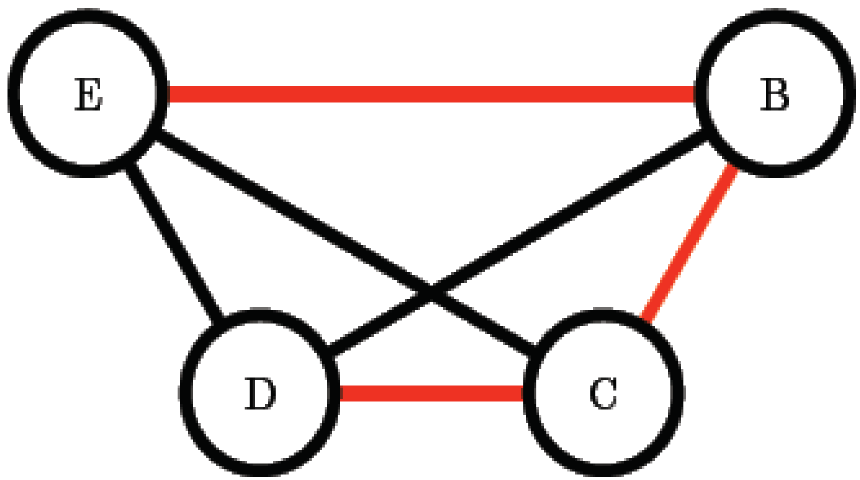
Consider the graph below:



Assume that $ADCBEA$ is a minimum weight Hamiltonian cycle.

Lower bound for the travelling salesperson problem example cont.

If we remove the vertex A from this graph and its incident edges: **we get a path passing through the remaining vertices.**

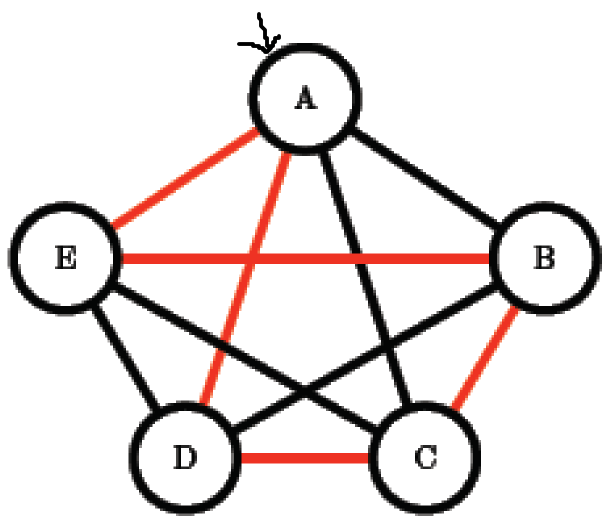


Such a path is certainly a **spanning tree** of the complete graph formed by these **remaining** vertices.

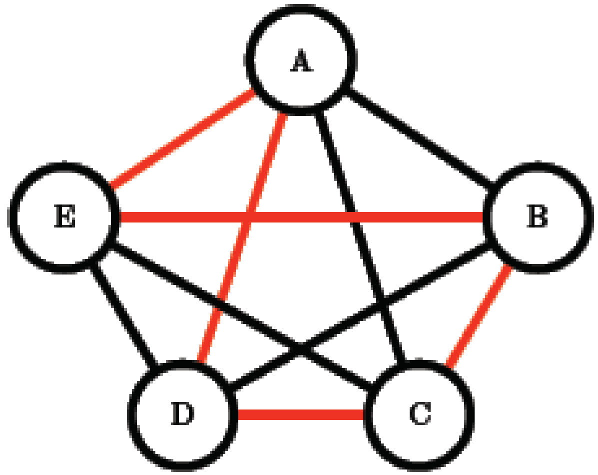
Lower bound for the travelling salesperson problem example cont.

Therefore the weight of the Hamiltonian cycle $ADCBEA$ is given by:

$$\begin{aligned}&\texttt{total weight of Hamiltonian cycle} = \\ &\texttt{total weight of spanning tree connecting } B, C, D, E \\ &+ \texttt{weights of two edges incident with } A\end{aligned}$$



Lower bound for the travelling salesperson problem example cont.



and thus:

`total weight of Hamiltonian cycle` $\geq$
`total weight of spanning tree connecting` B, C, D, E
$+$ `weights of the` **two smallest** `edges incident with` A

The right hand side is therefore a lower bound for the solution of the travelling salesperson problem in this case.

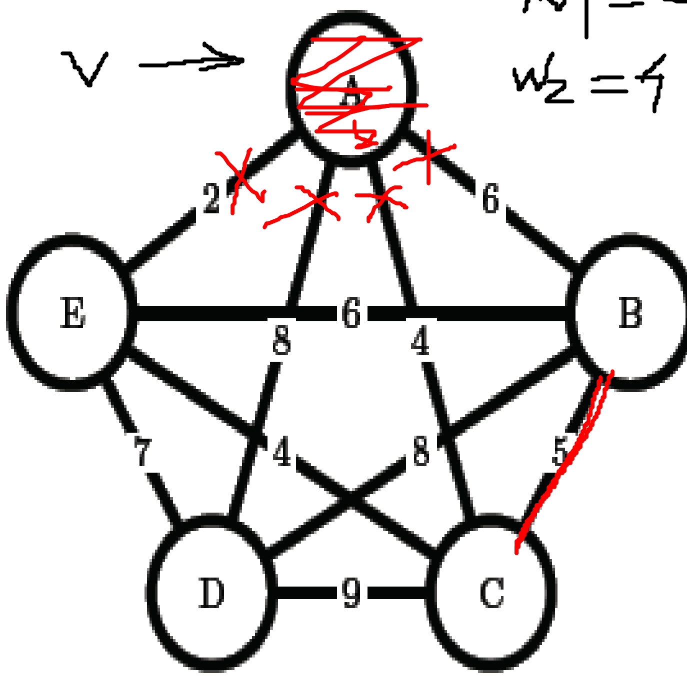
Algorithm 2.36 (Lower bound for the travelling salesperson problem).

- *Step 1: Choose a vertex V and remove it from the graph.*
- *Step 2: Find a minimum spanning tree connecting the remaining vertices, and calculate its total weight w.*
- *Step 3: Find the two smallest weights, w_1 and w_2, of edges incident with V.*
- *Step 4: Calculate the lower bound $w + w_1 + w_2$.*

Note: Different choices of the initial vertex V give different lower bounds.

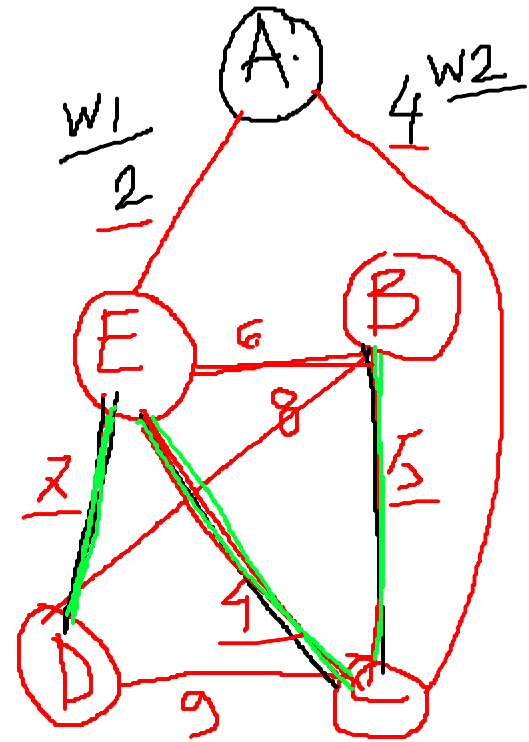
Use the Lower bound algorithm to find a lower bound for the Travelling salesperson problem for the following graph:

$V \rightarrow$



$W_1 = 2$

$w_2 = 4$



$\rightarrow w = 5 + 4 + 7 = 16$

MST $\rightarrow$

MST $16 + 2 + 4 = 22$

Lower B. $= W + W_1 + w_2 = 22$

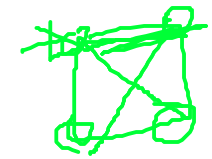
$\rightarrow$ U.B $=$ L.B $\times 2 = 44$

Self study

Upper bound of the travelling salesperson problem

To get an upper bound we use the following algorithm:

Algorithm 2.35 (The heuristic algorithm). *The idea for the heuristic algorithm is similar to the idea of Prim's algorithm, except that we build up a cycle rather than a tree.*

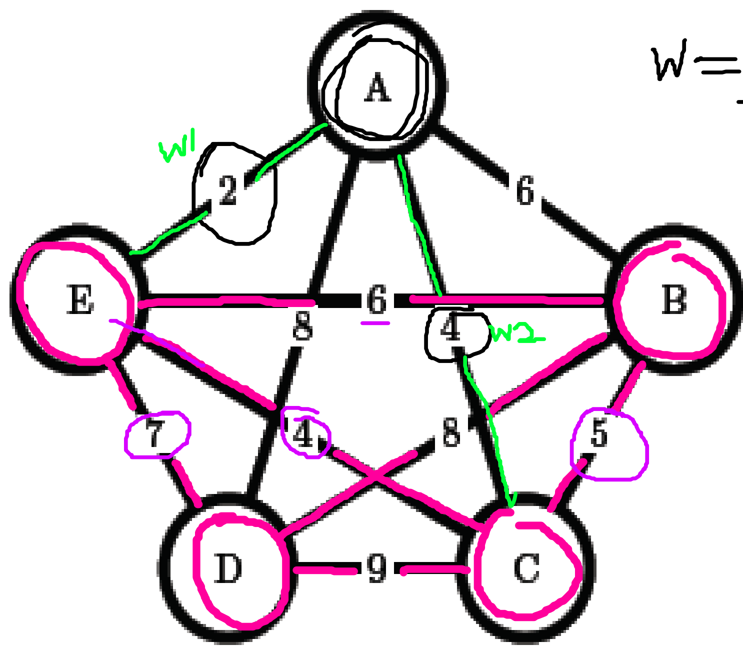


- *START with all the vertices of a complete weighted graph.*
- *Step 1: Choose any vertex and find a vertex joined to it by an edge of minimum weight. Draw these two vertices and join them with two edges to form a cycle. Give the cycle a clockwise rotation.*
- *Step 2: Find a vertex not currently drawn, joined by an edge of least weight to a vertex already drawn. Insert this new vertex into the cycle in front of the 'nearest' already connected vertex.*
- *REPEAT Step 2 until all the vertices are joined by a cycle, then STOP.*

The total weight of the resulting Hamiltonian cycle is then an upper bound for the solution to the travelling salesperson problem.

Be aware that the upper bound depends upon the city we start with.

Use the heuristic algorithm to find an upper bound for the Travelling salesperson problem for the following graph:



$W = 5 + 4 + 7 = 16$

$L.B = W + W1 + W2$

$= 16 + 2 + 4$

$= 22$.

$U.B = 22 \times 2 = 44$

For example, if we take the weighted graph of Fig. 11.1 and remove the vertex C, then the remaining weighted graph has the four vertices A, B, D and E. The minimum-weight spanning tree joining these four vertices is the tree whose edges are AB, BD and DE, with total weight 10, and the two edges of minimum weight incident with C are CB and CA (or CE) with total weight 15 (see Fig. 11.3). Thus, the required lower bound for the travelling salesman problem is 25. Since the correct answer is 26, we

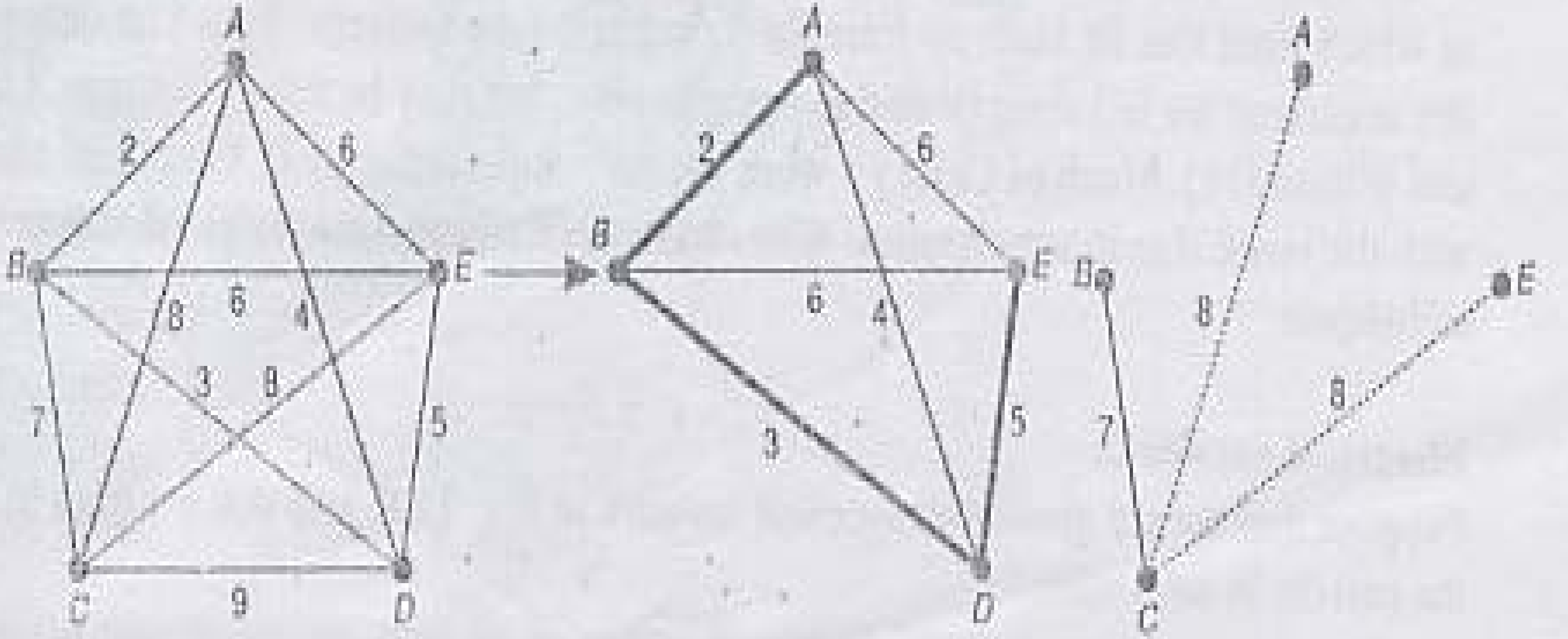


Fig. 11.3