Solving TSP with Christofides' Algorithm

Clément Mommessin

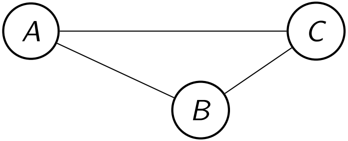
November 2024

Traveling Salesperson Problem

We consider a salesperson who wants to visit all n cities. We are given a **complete graph** G with weight of edges representing the cost of travelling from one city to another (the distance).

The goal is to find a **tour of minimum cost** traversing all cities once and only once.

Euclidian metric:
We assume the *triangle inequality* for the distance between cities.
For any 3 vertices A, B, C, we have $c(AC) \leq c(AB) + c(BC)$.

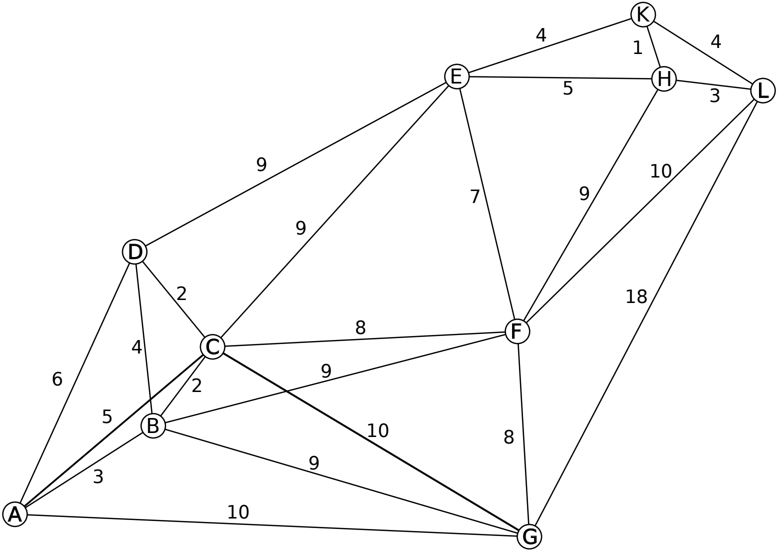


Christofides' Algorithm on graph G:

1. Create a Minimum Spanning Tree T^* of G.
2. Let V_{odd} be the set of vertices with odd degree in T^*.
   Find a Minimum-weight Perfect Matching M^* in the subgraph induced in G by V_{odd}.
3. Combine the edges of T^* and M^* to form a connected multigraph H.
4. Form a Eulerian circuit in H (traverse all edges exactly once).
5. Transform the Eulerian circuit in H into an Hamiltonian circuit in G by shortcutting repeated vertices.

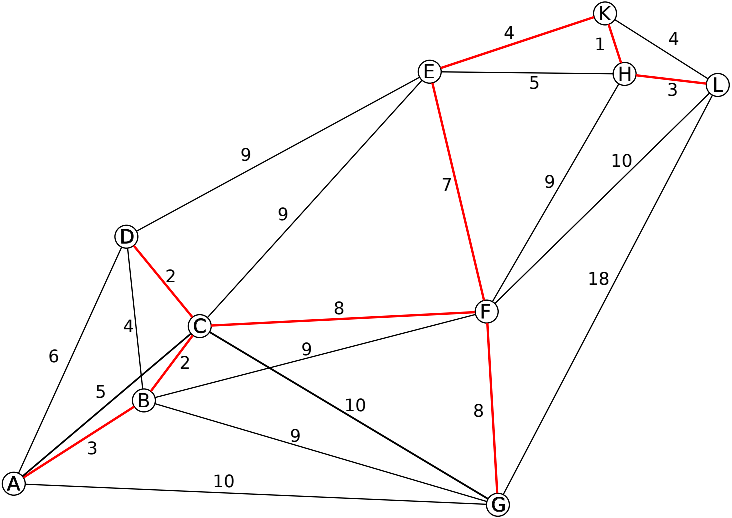
Example Graph

WARNING: This is not a complete graph, but it works for this example



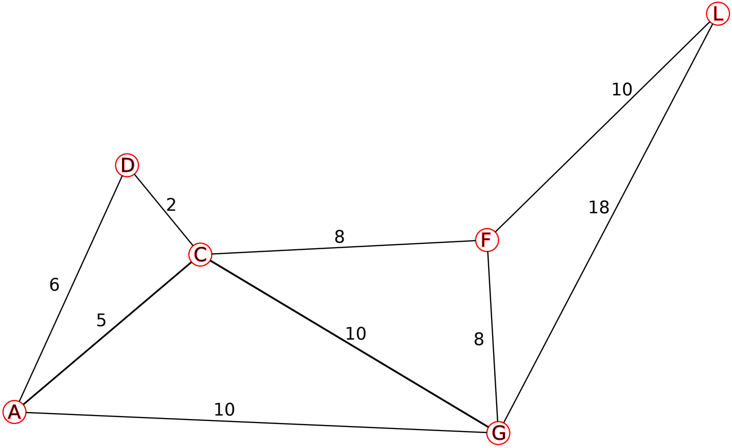
Step 1: Create a Minimum Spanning Tree T^*

Minimum Spanning Tree



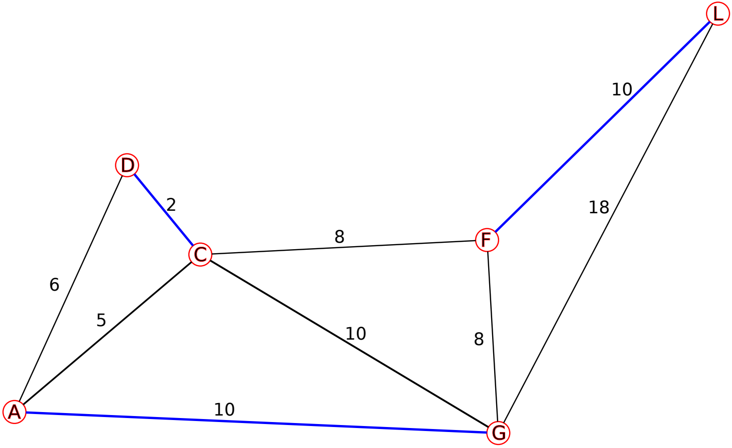
Let V_{odd} be the set of vertices with odd degree in T^*.
Prove that V_{odd} has an even number of vertices.

Vertices with Odd Degree



Step 2: Find a Minimum-weight Perfect Matching M^* in the subgraph induced in G by V_{odd}.

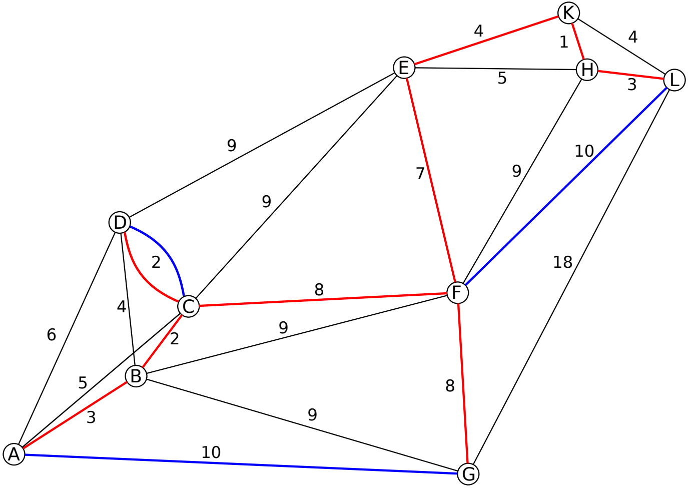
Minimum Perfect Matching



Step 3: Combine the edges of T^* and M^* to form a connected multigraph H.

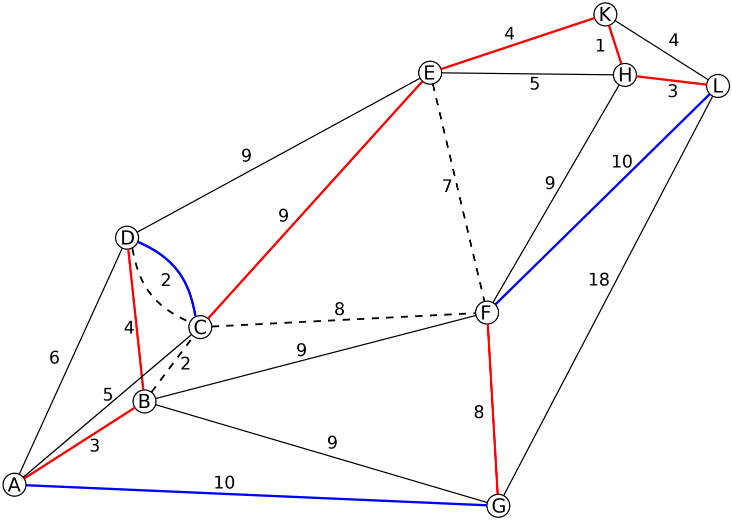
<u>Prove that each vertex of H has an even degree</u>.

Union of Graphs



Step 4: Form a Eulerian circuit in H (traverse all edges exactly once).
Step 5: Transform the Eulerian circuit in H into an Hamiltonian circuit in G by shortcutting repeated vertices.

Shortcutting



Et voilà ! (Bonus question: is it optimal?)

Approximation Proof

Prove that the tour formed by the algorithm is a 2-approximation, i.e., the tour has a cost no greater that 2 times the cost of an optimal tour.

Hints:

1. What is the relation between the cost of the tour and the weights of the minimum spanning tree and the minimum perfect matching?
2. Show that the shortcutting process does not increase the total weight of the tour.

Even better: prove it is a 3/2-approximation.

Approximation Proof

Prove that the tour formed by the algorithm is a 2-approximation, i.e., the tour has a cost no greater that 2 times the cost of an optimal tour.

Hints:

1. What is the relation between the cost of the tour and the weights of the minimum spanning tree and the minimum perfect matching?
2. Show that the shortcutting process does not increase the total weight of the tour.
3. Prove that the weight of the minimum spanning tree (ω_{T^*}) is a lower bound of the value of the optimal tour (ω_{OPT}).
4. Prove that the weight of the minimum perfect matching (ω_{M^*}) is a lower bound of the value of the optimal tour (ω_{OPT}).

Even better: prove it is a 3/2-approximation.