The Long Road: Comparison of Approximation Heuristics for the Traveling Salesperson Problem

Miles Hoene-Langdon, Daniel Chen, Artem Yushko, Arthur Viegas Eguia

Introduction

Traveling Salesperson Problem:

Input: A complete graph G.

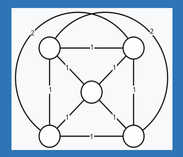
Output: A Hamiltonian cycle of G with minimum total edge weight.

Christofides Algorithm:

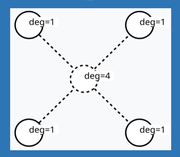
1. Create a minimum-weight spanning tree T of G.
2. Find the set O of vertices in G with odd degree.
3. Create a subgraph M of G induced by O.
4. Find the minimum-weight perfect matching for M.
5. Combine T and minimum-weight perfect matching of M to form a Eulerian graph H.
6. Find a Eulerian cycle for H.
7. Convert the Eulerian cycle found in the previous step into a Hamiltonian cycle.

Example of Christofides Algorithm [5]:

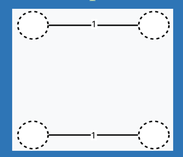
Input G:



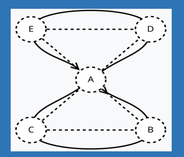
Step 1:



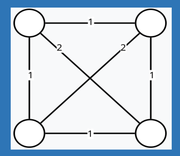
Step 4:



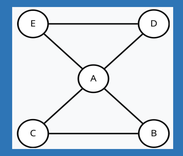
Step 6:



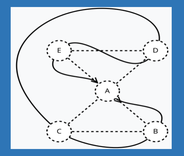
Step 2 & 3:



Step 5:



Step 7:



Other Heuristics Used:

- **Smallest Insertion**: arbitrarily pick two vertices to start the tour and loop through all unvisited vertices to find the one which, when added between two vertices of the current tour, increase the tour length the least, then continue this process until all vertices are visited and the tour is complete.
- **Nearest Neighbor**: start at an arbitrary vertex and go to the closest unvisited vertex repeatedly until all vertices are visited, then return to the start vertex.

Datasets:

- TSP95 Country Datasets: vertices are cities and edges are distance between cities [4].
- Electrical Grid Dataset: vertices are locations in need of power and edges are distances between them [1].
- Yeast Protein Dataset: vertices are proteins and edges are measures of interactions between them [3].

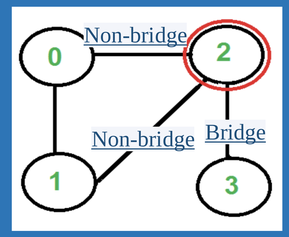
Methods

Implementation of Christofides Algorithm:

- Minimum-weight spanning tree computed with Kruskal's Algorithm.
- Minimum-weight perfect matching computed with Edmonds' Blossom Algorithm through Networkx.
- Eulerian cycle computed using Fleury's Algorithm.

Fleury's Algorithm [2]:

1. Check whether the input graph G has 2 or 0 vertices with odd degree.
2. If 0 vertices of odd degree, then start anywhere. If 2 vertices of odd degree, then start at one of them.
3. Follow each edge one at a time. If there is a choice between a bridge and a non-bridge edge, then choose the non-bridge.
4. Stop when all edges reached



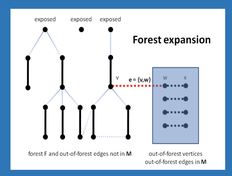
- Bridges are edges that once crossed, cut off access to unvisited nodes of the graph
- Edge 2-3 above is a bridge.
- Edges 2-0 and 2-1 are not bridges, so both are valid.

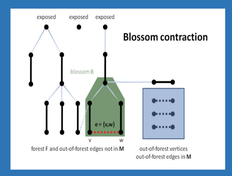
Edmonds' Blossom Algorithm [6]:

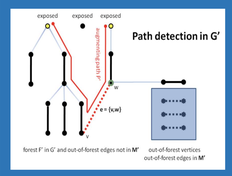
We implemented a simpler version of this algorithm that doesn't factor in edge weights. This simpler version produces a perfect matching, not necessarily of minimum weight. Note, forests are disjoint trees created from a matching with unmatched vertices as roots. The steps of this algorithm are as follows on a single iteration.

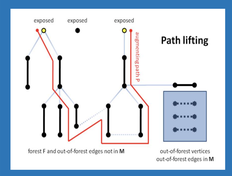
1. Process an out of forest edge that causes an expansion of the forest.
2. Detects a blossom (odd length cycle) and contracts it.
3. Locates an augmenting path P' in the contracted graph.
4. Lifts the augmenting path P' to an augmenting path P in the original graph and updates accordingly.

This algorithm continues iterating the steps above until no augmenting path is found and the resulting matching is perfect as a result.





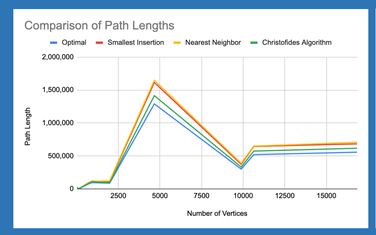


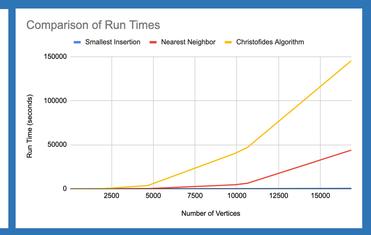


Processing datasets:

- Algorithms ran on cluster, through tmux, measuring running time and length of paths.
- Each heuristic algorithm was run three times and numbers of three runs were averaged for 10 country datasets, the electrical grid dataset and the yeast protein dataset.

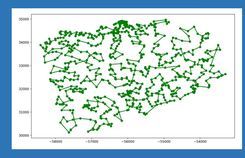
Results



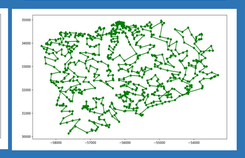


Example: Tours of Cities in Uruguay [4]

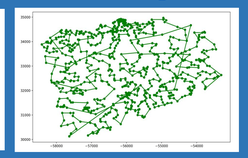
Christofides:



Smallest Insertion:



Nearest Neighbor:



- Christofides algorithm yields the shortest tour, but with significantly longer run time.
- Nearest neighbor heuristic is fastest by far with decent path lengths.

Conclusion: Christofides is the most desirable approximation algorithm when running time and computational power are not constraints. However, in situations where precision is not as key, nearest neighbor is desirable due to its efficiency.

References

[1] Denilson F. Barbosa, Carlos N. Silla, and Andre Y. Kashiwabara. 2015. Applying a variation of the ant colony optimization algorithm to solve the multiple traveling salesmen problem to route the teams of the electric power distribution companies. In Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective - Volume 1 (SBSI '15), Vol. 1. Brazilian Computer Society, Porto Alegre, BRA, 23–30.

[2] GeeksforGeeks. (2023, June 6). *Fleury's algorithm for printing Eulerian Path or circuit*. https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/

[3] Johnson, O., Liu, J. A traveling salesman approach for predicting protein functions. *Source Code Biol Med* 1, 3 (2006). https://doi.org/10.1186/1751-0473-1-3

[4] UWaterloo. (2022, February 8). *National traveling salesman problems*. UWaterloo. https://www.math.uwaterloo.ca/tsp/world/countries.html

[5] Wikimedia Foundation. (2024, December 29). *Travelling salesman problem*. Wikipedia. https://en.wikipedia.org/wiki/Travelling_salesman_problem

[6] Wikimedia Foundation. (2024, October 12). *Blossom algorithm*. Wikipedia. https://en.wikipedia.org/wiki/Blossom_algorithm