Analyst's Traveling Salesman Problem Algorithm Implementation

Gabriel Johnson and Robby Lawrence

October 17, 2024

What is the Analyst's Traveling Salesman Problem (ATSP)?

- The ASTP is a problem related to the more well-known Traveling Salesman Problem (TSP).
- While the TSP aims to find the shortest path touring a finite set of points, the ATSP asks if there exists a curve of finite length (rectifiable) that contains a given set of points in $\mathbb{R}^n$.
- We follow the algorithm laid out in Time Complexity of the Analyst Traveling Salesman written by Anthony Ramirez and Vyron Vellis.

Language and Libraries

- Our code is written in C++
- We use igraph for efficient graph manipulation
- Building is handled with CMake

Goals

1 Develop a subprogram to output a 2-to-1 Euler tour of a graph.

- Takes as input a number of vertices, and an edge set between those vertices.
- Outputs an ordering of the vertices that begins at the initial vertex, visits each edge twice, and ends at the initial vertex.

Goals

2 Develop a subprogram to output a sequence of vertex sets.

- Inputs a set of points in the plane
- Outputs an increasing sequence vertex sets $(X_0 \subseteq X_1 \subseteq X_2 \subseteq \cdots)$, given criteria determined in the paper.

Goals

3 Develop a subprogram to calculate “local flatness”

- This program outputs the local flatness of the vertex set V around v.
- Inputs a vertex set V, a vertex $v \in V$.
- Inspect a ball around v, find a “line of best fit” for the points in the ball, and output a value corresponding to how well the line fits the points

Progress

Organization and build system

- igraph included in as a submodule to the repository
- build script to build igraph on new system
- CMake to build the project
- build script to handle calling cmake for the project with the correct library dictories

Progress

Goal 1

- Used standard template library since igraph was not needed for this subprogram
- Class structure consists of an unordered map with an integer representing the vertex number as the key and a vector of pairs of integers for the edges. Each pair in the vector has two integers: the first represents the other vertex the vertex is connected to, and the second is how many times this edge has been traversed.
- Function that searches for the vertex numbered one and the helper function starting there.
- The helper function recursively calls itself after iterating through the adjacency vector for a given node, checking each edge to see if it's been traversed less than twice. At the first available node, it increments the edge both ways, then breaks off and recurses.

Progress

Goal 3

- The flatness function takes in a set of points in $\mathbb{R}^2$, a vertex v as an index into the set of points, a value for ϵ, and a value k, from which the radius of the ball is computed. Returns the flatness.
- Point set class which handles finding intersections between the ball and the set of points
- Needs testing