Algorithm Write-Up

Overview

The chosen algorithm to traverse the weighted graph was a Monte Carlo Search Tree (MCTS), which starts from a designated start node and aims to minimize the total cost over "k" traversals. The algorithm consists of four main steps: Selection, Expansion, Simulation, and Backpropagation.

1.) **Selection**
   - Navigates down the tree, selecting the best node based on the UCB1 formula, which aims to balance exploration and exploitation

2.) **Expansion**
   - If the selected node isn't fully expanded, it adds a new child node to the tree representing the unexplored neighbor

3.) **Simulation**
   - Simulates a random path from the expanded node to the end node, keeping track of the accumulated costs along the way

4.) **Backpropagation**
   - The results of the simulation are backpropagated up the tree, updating both the total cost and the visit count for each node along the path

Exploration vs. Exploitation

MCTS balances exploration and exploitation using the UCB1 formula within the Selection step of the process. The UCB1 formula has two main components:

1.) **Exploitation Term**
   - Driven by minimizing the average cost

2.) **Exploration Term**
   - Driven by nodes with a low visit count, increasing the weight, which encourages exploration. The exploration constant is modified to weigh exploration larger during simulations and to weigh exploitation more during traversals to select the best path.

Uncertainty

The algorithm handles uncertainty by sampling the edge weights only when traversed for the first time, storing the results, and using those discovered weights in future simulations. This process allows the algorithm to iteratively learn the graph structure without needing prior knowledge of the weights. The balance between exploration and exploitation enables the algorithm to explore new paths while exploiting known low-cost paths, thus handling uncertainty in unknown weights.

Different Values of K

The algorithm is flexible for different values of k. Each traversal is independent of the others, with the algorithm running a full set of simulations for each traversal. As k increases, the number of total simulations increases, and the results are further backpropagated and updated in the tree. Because of this MCTS doesn't need to be adjusted for different k.

Evaluation

The algorithm was evaluated by testing it on random graphs with varying weight distributions and varying values of k and simulations. Measuring the performance was accomplished by looking at the total cost of traversals and the consistency of selected paths.