K-Shortest Paths Algorithm

Tomas Sheiles-Hutton s5132012

# Overview

Finding the shortest route between two locations through a particular set of nodes is a common computer science problem. While finding an arbitrary route between two nodes is a relatively simple problem; without using a heuristic to determine an efficient route, routes between nodes in a large graph may become needlessly long. In order to plot an efficient route between two nodes, some heuristic is required. For this project Dijkstra’s algorithm was used to find the shortest route between two nodes of a loop-less, directional graph. This algorithm calculates the distance from beginning of the graph to all adjacent neighbours, if the distance from one node to its neighbour is lower than the distance to the same node from another route, the distance is updated. Each node with the next shortest distance is then taken as the next node in the route. This ensures that the final route to each node in the graph is the shortest possible.

Finding the next shortest paths from the source to the goal node is similarly challenging. Using a brute force approach to determine these paths would be impractical for larger graphs. Because of this, a method must be determined to efficiently calculate similar divergent paths from the original shortest route.

# Algorithm Description

# Pseudo Code

# Results and analysis