COMP3888 - Phase 1

September 2020

Description

This document describes the first step of converting a drone delivery problem into computable algorithm.

Assumptions

The assumptions are listed below.

- No factors will be taken into consideration at this stage.
- The starting point, destination as well as charging stations will be provided as three dimensional point like (x, y, z).
- The location data will be in floating-point numbers.
- The hardware will have sufficient capacity and computational power to run algorithm. Space and time complexity analyze will be provided.
- The algorithm will not be required to control the drones. Only giving an optimal path.
- Currently "Optimal" is defined to be the minimum distance from starting point to destination. However, this goal might be later changed to the least amount of time.

Reduction

Steps

In order to run Dijkstra's algorithm, a directed graph will be required. To construct a graph from given points, connect the following

- Start to destination (one way)
- Start to all stations (one way)

- Stations to stations (bidirectional)
- Stations to destination (one way)

Edge weights: direct distance between three dimensional coordinates.

Complexity

Let the number of charging station equals n.

$$O(1) + O(n) + O(n^2) + O(n) = O(n)$$

Complexity of step 1,2,4 are trivial. The space complexity of connecting all stations with each other is equivalent to hand-shake problem. Requires

$$1 + 2 + ... + (n-1) = \frac{n \times (1 + n - 1)}{2} = O(n^2)$$

number of edges.
Adding the number of nodes, the total space complexity would be

$$O(n^2)$$

. Running time shall be the same assuming computing distance takes $O(c)$.

Main: Dijkstra's

Graph is constructed. The implementation of Dijkstra's will not be expanded here. The running time of Dijkstra's would be

$$\Theta(E)$$

where E is the number of edges. Together with the space complexity mentioned above, the running time shall be

$$\Theta(n^2)$$