Statement of Work & Deliverables: Amazon Prime Air Traffic Control

Group 12

Behkish Nassirzadeh
Amish Patel
Chang Bok Lee
David Hong

1. Problem Definition

Amazon Prime Air uses multiple quadrotors to deliver packages. The goal is to find the shortest path to its destination, while not interfering with other quadrotors. Unlike the traditional surface logistic services, the flying quadrotors operates in 3-dimensional space which allow the quadrotors to be highly efficient. One of the main challenges for operating multiple quadrotors is to avoid collisions between each other and other obstacles. The path planning requires metaheuristic data that can estimate the distance between a start point and an endpoint, the average altitude, the average fuel consumption, etc.

For the path planner to start computing, it needs a digitized representation of different environments the quadrotors are flying in. The digitization of the quadrotor environment will include the topographical data of the city. This will limit the minimum height the quadrotor must fly in. The current location of the quadrotor will be recorded using 3 coordinate system, (x_i, y_i, z_i). Each coordinate set can then be used to track the path the quadrotor will take to deliver its package. Also, this allows computing the trajectory of quadrotors. The trajectories from different quadrotors can then be compared for possible collision. If any possible collision is detected, then the trajectory of the quadrotors need to be altered to avoid such collision.

2. Problem Formulation

State

The state represents the path of each quadrotor, given in $(x_i, y_i, z_i)_d$ coordinate system. The path will contain all the coordinates in an incremental step that will take the quadrotor from the start position to the goal position. The subscript *i* in the coordinate system represents the current step for the given quadrotor, *d*. Figure 1 defines the matrix representation of the state and the coordinate taken by the given quadrotor.

$$S = \begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_n \end{bmatrix} \qquad \mathrm{d}_i = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_k & y_k & z_k \end{bmatrix}$$

Figure 1. Matrix representation of states

Initial State

The initial coordinates are generated to randomly without checking for their feasibility.

Goal :

All quadrotors have paths from a start point to an end point assigned, with no collision and minimum cost.

Actions:

To move from one state to another, we move from one coordinate to another via crossover and mutation. Each transition between the grid boxes can only be made to a grid box that is 1 box away from the current grid box.

1. Addition
   - Another coordinate can be added to generate a new solution space
2. Subtraction
   - A coordinate can be removed to generate a shorter path
3. Replacement
   - A coordinate within the path can be replaced with another coordinate that generates a solution with a lower cost
4. Crossover
   - A new child path can be formed by combining two parent paths

Cost

We define the cost function as follows:

The optimized path will use the minimum cost to deliver a package. The cost function includes: C_{Length} *representing the distance travelled by a quadrotor,* $C_{Altitude}$ which penalizes paths that include a higher altitude, C_{Fuel} to represent a path that uses more fuel than previously calculated.

$Cost = C_{Length} + C_{Altitude} + C_{Fuel}$
C_{Length} is defined as the total distance a quadrotor must travel.
$C_{Altitude}$ is defined as the average altitude a quadrotor is flying in.
C_{Fuel} is defined as the amount of fuel expected to be consumed while making a delivery.

3. Proposed Optimization Algorithm

3.1. Genetic Algorithm

The Genetic Algorithm is a population based non-deterministic optimization method. The GA simulates how pathing of quadrotors can be improved by applying a series steps similar to biological evolutionary steps taken in nature.

The initial set of solutions is generated randomly and will be tested for its fitness to find the best solutions and will be improved using proper techniques. The generated solution is first tested for its fitness. The fitness test includes collision test to make sure two or more quadrotors are not sharing the same 3D space at the same time. Once the solution passes the fitness test, the solution path goes through either crossover, mutation or replacement in attempt to generate a better solution. The better solution is defined as the solution that uses less cost compared to the original cost while also passing the fitness test.

3.2. Justification

Due to the nature of the problem, the genetic algorithm fits well when generating potential solutions. The number of quadrotors make the pathing problem non-deterministic. While the path taken by a quadrotor can be pre-calculated based on computed heuristic data, there are many factors that make the result invalid: wind velocity, weather, obstacles, etc. The GA works very well with such non-deterministic factors in place. The code can mutate, crossover or replace to generate an optimal solution. In addition, since the problem includes population based solutions, many solutions are possible. A population based algorithm like the GA is a better fit than a trajectory algorithm.