Demo Words:

Here we have the software loaded and with a pre-existing database loaded.

I created the sample customers shown for the purpose of this video.

Here we have the map of Aberdeen and the depot is marked here

The number of drones selected is 5, maximum flight time is 600 seconds and flight speed is 15 metres per second.

The search algorithm we have selected for this run is the Greedy Best First algorithm

So I will go ahead and run the program. After a short delay we will see our clusters and routes marked on.

We can see the results here – each coloured line represents the route between the customers in that cluster.

We can also see that there are 6 routes. What has happened is the software has first created 5 clusters, one per drone. The routes are calculated and it is discovered that one of the routes takes longer than 600 seconds to complete. This cluster is then split in half and routes calculated for the two new clusters. These two are possible, thus all routes are possible and they are drawn on the map.

I have now loaded the output HTML file.

It shows us the parameters of the last run, such as total number of customers, number of drones and the resulting number of routes.

It also gives us the wind speed and bearing at the time.

Finally we see the route detail table and the map. The table helps to give an idea of which drone will be flying which route. As we can see, Drone 1 has been assigned the 6th route.

Hello and welcome to my presentation. My project has been creating a Drone Delivery Scheduler using clustering and path finding algorithms.

NEXT SLIDE

My motivations behind this are simple;

I am a regular online shopper, and one who hates waiting several days for my order to arrive. Naturally when I heard that companies had drone delivery services in the works I was very excited.

In addition to this.

I find drones to be an exciting new technology with a lot of potential applications.

I also believe that they will help reduce the carbon footprint we have on the planet, helping to save the environment.

NEXT SLIDE

The aims of the project were to create a software application that helps plan drone delivery. It’s intended to create routes for drones to take that are a fast as possible. I also wanted to compare the effectiveness of different algorithms for this task.

NEXT SLIDE

A short introduction. A drone is a small unmanned flying vehicle. Many companies are looking into using them for delivering items, such as Amazon’s “Prime Air” and DHL’s “Parcelcopter”. They intend to use drones to deliver from a city depot to people houses.

NEXT SLIDE

Some of the advantages of using drones for delivery are:

Speed – Drones are not hindered by busy roads and rush hour traffic. They can fly a direct route from where they are to the customer, ignoring any confusing road layouts or small lanes that a van may not fit down.

Cost – The cost of flying a drone is much lower than that of driving a van around a busy city.

Environment – Research shows that co2 emissions produced by drones are lower than that of trucks when used for delivery close to the depot.

There is also high demand for such a service. A study conducted by McKinsey and Co revealed that 23% of customers surveyed would be happy to pay extra for same-day delivery.

NEXT SLIDE

While these advantages that drones hold over traditional delivery methods are strong, drones do have issues.

Their delivery range is limited by battery life, weight capacity and adverse weather. Drones currently have a flight time typically ranging from 15 minutes to 30 minutes for consumer drones on the market. They are not able to carry much weight without further reducing this flight time. If we then consider that the drone might have to fly into a strong headwind, it may not be possible to delivery items at all.

Additionally, the laws around drone delivery are still restrictive. In the UK, drones are not allowed to fly within 150 metres of any congested area or organised open-air assembly of 1000 people or more. They also cannot legally fly within 50 metres of a person, property, vehicle or structure that is not owned by the person in charge of the aircraft.

These restrictions severely hinder the viability of drones for delivery at this time.

NEXT SLIDE

In 2019, Amazon predicted that their drones would have a 5 mile flight range and a 5 pound weight capacity. This restricts how many customers could be served at once.

For the purpose of this project, I have assumed that drone tech and laws will advance to improve these limits.

The assumption is that one drone will legally be allowed to carry multiple items to multiple customers in one flight.

NEXT SLIDE

With this in mind, there is a clear need for an efficient schedule. A good schedule will save time of the company, meaning that their invested money in the drones goes further.

It will also create greater customer satisfaction, as customers will receive their items faster and be given an accurate estimated delivery time.

NEXT SLIDE

The difficulty in created an efficient schedule, or route for the drone to take is known as the travelling salesman problem. This problem dates back to 1579.

It traditionally describes a salesman who wants to visit a number of cities. He only wants to visit each city once and wants the fastest route between the cities.

While this sounds like a simple task, we just need to figure out all the routes and pick the shortest one. This is manageable for a small number of customers, but the number of potential routes exponentially increases.

If we have 5 customers, there are 120 possible routes.

If we have 10 customers, there are 362,800 routes

For 15 customers, the number jumps to one trillion three hundred and seven billion, six hundred and seventy-four million, three hundred and sixty-eight thousand possible routes.

Obviously this is not viable to calculate all of these routes in any reasonable time frame, and if we assume that a company like Amazon may have hundreds or thousands of orders a day, it is clear that we cannot use this method to find our routes.

There is no known algorithm to solve the problem.

NEXT SLIDE

Because of this issue, we need to break customers down into smaller, manageable groups, and then find routes for each of these groups. We can’t expect to find a good route for 15 customers, but if we break the 15 customers into three groups of 5, we have a good chance at finding a good or even the best route between them.

For this we use a clustering algorithm to break the customers into smaller groups, and a path-finding algorithm to find the routes for each group.

NEXT SLIDE

Clustering allows us to group customers together based on a measure of similarity. In our case, we will use geographic location.

NEXT SLIDE

The algorithm I opted to use is called “kMeans”. All we need to give the algorithm is the number of clusters we want, and it does the rest.

It first randomly selects “k” number of centres for the clusters.

Each customer is then assigned to a cluster based on which centre they are closest to.

The algorithm then recalculates the centres by finding the mean distance to each customer in each cluster, and assigns that customer as the centre.

This process repeats until no changes are made to the clusters.

NEXT SLIDE

Now that we have our clusters we can move on to finding a route between the customers. There are a range of possible algorithms to complete this task, but we will focus on two: Greedy Best First and Genetic Algorithm.

NEXT SLIDE

The Greedy Best First algorithm is simple to understand and should provide reasonably good routes. It calculates the flight time from the drones current location to each other unvisited location.

It then selects the shortest one and repeats until it’s back at the depot.

This algorithm is much faster to execute than brute forcing, however is not guaranteed to give us the fastest route.

NEXT SLIDE

Our other algorithm selected is the Genetic Algorithm. This algorithm is designed to emulate Darwin’s Theory of Evolution.

It starts with a population of randomly generated routes.

A score is calculated for each route, which in our case is the length of the route.

High scoring routes are selected to breed with one another, which means they swap some characteristics.

The routes have the potential to mutate at this stage, meaning that some features of the route will be changed at random.

The lowest scoring routes are removed from the search.

This process repeats for a set number of generations.

NEXT SLIDE

The genetic algorithm is good for this type of problem as it allows a large number of potential solutions to be tested at once. It is able to iteratively improve the route with each generation and the parameters of the search can be modified to increase population size and the number of generations, allowing however many candidate routes we want to be tested.

However, if a bad set of candidates are generated at the beginning, it can sometimes give bad results.

NEXT SLIDE

The software was written in Python. It provides a simple Graphic User Interface for users to interact with.

Data that is entered by the user is saved to a database so it can be used later.

It allows the user to easily adjust the number of drones, flight speed and maximum flight time parameters.

For clustering, the “kMeans” algorithm was implemented, and for path finding there is the choice of Greedy Best First or Genetic Algorithm.

Once the routes have been found, there is a visual output of the clusters and associated routes.

Finally, an HTML file is generated which gives an overview of the last run of the application.

NEXT SLIDE

Here we can see the result of the clustering. Customers are split into 5 groups by location. The marker in the centre that is not within a cluster is the depot.

NEXT SLIDE

I then used the Genetic Algorithm to find routes for each cluster. Each different coloured line represents each route.

NEXT SLIDE

Here is a sample HTML output file. It gives a brief overview of the parameters and resulting routes.

NEXT SLIDE

A requirements specification was created early on in the project. This document outlines features that must be present in the software, features that should be present, and features that could be present.

It outlines the scope of of the project and exactly what is expected to have been produced.

NEXT SLIDE

All of the Must have functionality of the application has been implemented. These were bear-minimum functions that the application needs to create a schedule.

NEXT SLIDE

The should have functionality list is mostly complete. The only objective here that is not met is that the software is not delivered as a .exe file. It was discovered after development that this is not achievable due to the way the software has been built.

NEXT SLIDE

The could haves are nice to have but not essential functions. Due to time constraints of the project, neither of these were completed.

NEXT SLIDE

Testing is not yet completed, however I have conducted some preliminary tests. I set up mock customers and input parameters to allow for 5 routes to be created. I then compared the effectiveness of the Genetic Algorithm and Greedy Best First Search algorithms.

I expected that the genetic algorithm would produce shorter routes.

NEXT SLIDE

The results show that my expectations were correct. Across all 5 routes, the genetic algorithm out-performed the greedy best first algorithm. Greedy best first at best produced a route 3.76% longer than the genetic algorithm, and at worst 51.89% longer.

NEXT SLIDE

From these results it is fair to say that the genetic algorithm is better suited to the task. It is better able to optimise the routes by exploring more possible solutions than greedy best first.

There are instances where the genetic algorithm produces a longer route by distance than but manages to fly with a tail-wind for longer, and thus the route actually takes less time to complete.

However, there are occasions where the genetic algorithm can be out-performed by greedy best first. Particularly in cases where there are a large amount of possible routes, the genetic algorithm can generate a bad set of initial routes and not have enough time to improve them.

NEXT SLIDE

I have had many successes in this project.

A software solution has been provided, which allows users to quickly and easily input data from the GUI.

It accepts a range of user inputs, and provides different path-finding algorithms.

The program executes quickly, which in a real-world scenario would be essential.

It also proves that AI is an effective tool for scheduling drone delivery.

NEXT SLIDE

However, it has not all been positive.

There was little thought put into design, which led to a very basic looking program. Additionally some of the code is messy, which makes maintaining the program harder down the line.

There is no guarantee that the solutions found are optimal, or indeed that they are good at all.

Input to the software is not a simple as it could be

And finally the output HTML file is basic and not interactive in any way