**Literature Review**

This section explores literature related to drone delivery scheduling. It will cover the background of drones, and difficulties in implementing a scheduler. Finally, a conclusion will be drawn of the findings.

**Drone Delivery**

A drone is a small, unmanned flying vehicle. In recent years many companies have unveiled plans to begin delivering packages via drone. In 2013 Amazon announced their ‘Prime Air’ service that will deliver packages to customers (BBC). DHL have announced their ‘Parcelcopter’ project, which has successfully delivered medicine to the island of Juist in the North Sea (Hern, 2015). There are many more companies who have unveiled plans to release their own delivery drone solution (Sacramento, 2019). This sudden upsurge in use has been brought on by advancements in technology used in construction of drones. The price of manufacturing carbon fibre has dropped from $25 to $10 per kg over the last 20 years. This combined with improvements in battery technology allows drones to fly faster and further (Dorling et al, 2017).

There is great demand from customers for a faster, more reliable option for delivery. This is a driving factor in the development of drones for last-mile delivery. A study conducted on over 4700 people from China, Germany and the USA showed that 23% of customers are willing to pay extra for the benefit of same-day delivery (McKinsey, 2016)



**Figure 1 – Amazon Prime Air drone**

There are several advantages to using drones for last-mile delivery. The primary advantage of drone vs truck is the speed and timing accuracy as drones are not affected by traffic or road layout of a city. This enables them to offer fast delivery and tell the customer to the minute when the item will arrive (Hau L. Lee et al, 2016).

For the company deploying the delivery solution, they will likely save money. A study performed by ARK Invest suggests that Amazons drone delivery service could be charged at just $1 per delivery and still be profitable. (Keeney, 2015)(Figure 2).

It is also suggested that delivery by drones will be environmentally beneficial. As drones are battery powered, they do not directly produce any diesel pollution. Research shows that carbon dioxide emissions produced by drones is lower than that of trucks, even taking into consideration the energy requirements of drones and the smaller area they can service. (Goodchild and Toy).

The speed, cost and environmental benefit are the key advantages of delivery via drone. They are the driving force between the recent upsurge in usage.

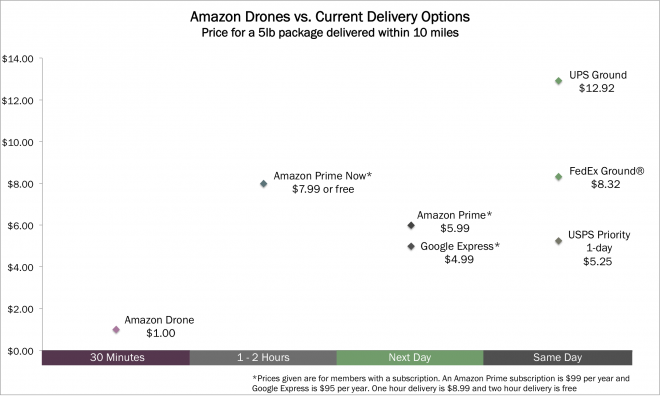


Figure 2 – Comparison of delivery cost and time

**Problem Specification & Boundaries**

The problem to be studied and solved here is how to create optimal delivery paths for drones. We are going to assume that a company of has set up a depot and plan on using that as a base for their drones to work from. They will also use it as a charging station while they are not in use.

Current drone technology makes this a simple scheduling task. Amazon announced on 5th June 2019 that they expect their drones to be able to fly up to 15 miles and deliver packages under 5 pounds (About Amazon Blog). The CEO Jeff Bezos has stated that 86% of items delivered by Amazon weigh 5 pounds or less (CBS News). This shows us that Amazon are aiming to deliver one item per drone per trip.

We will be relaxing some of these restrictions to enable us to look to the future. The Alta 8 from FreeFly Systems can carry up to 18kg (FreeFlySystems.com). It is reasonable to assume that technology will continue to improve to the stage where companies will use one drone to carry a much larger item, or several items at once.

These assumptions of the future of drone tech and their use within industry turn this problem from simple delivery time scheduling, to a travelling salesman problem.

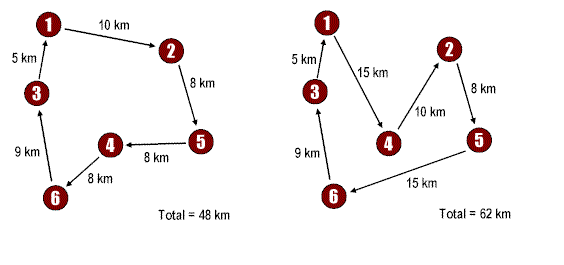
**The Travelling Salesman Problem (TSP)**

The travelling salesman problem is an old one, and variations appear as early as 1759 (P.Larran AGA)In the 1930s it was studied by mathematicians in Vienna and Harvard (math.utwaterloo.ca). The problem described a salesman who must visit multiple cities. He only wants to visit each city once and wants to start back where he started. A perfect solution to the problem finds the shortest route for the salesman to take to complete their journey (Amunar Saiyed).

At a glance the problem seems trivial, it is simple to understand and the method of solving it isn’t difficult either. All we must do is find every route and pick the shortest one.

If we name the number of cities n, the number of possible routes is the factorial of n (Amunar Saiyed). If the salesman must visit 5 cities, there are 120 possible routes. If we increase this to 10 cities, there are 362,800 possible routes, and 15 cities gives us 1.3e12 possible routes. This exponential growth is where the difficulty lies in solving the TSP.

If we were able to find and evaluate 1 million routes per second, it would take over 15 days to find the solution for a 15-point route. If we apply this to the drone delivery domain, we may have thousands of orders a day and the amount of processing time and power to brute force the best route is unrealistic. For this reason, we need to find alternative methods to solve the TSP.

The problem is NP-hard which means that there are no known techniques to solve it in polynomial time. (Kylie Bryant)

**Figure 3 –Two routes in a travelling salesman problem**

**Clustering**

Because of the difficulty of the Travelling salesman Problem, and the limitations of drone technology, it is useful to break down our problem into smaller groups, or clusters. If we take the above example of 15 cities and we split this into 3 sets of 5, our computation time drops dramatically. Assuming the same processing power this would take 0.00036 seconds to find and evaluate the best routes. While one million routes per second is an unrealistic number, it illustrates the potential benefits to computing and time resources needed with clusters. Clustering also allows us to model a delivery system with multiple drones delivering simultaneously, which is more realistic then just creating one route for one drone to carry out. This technique will also improve the speed of delivery.

**K-means**

The k-means technique takes a parameter of k, and randomly selects that many locations to begin. These locations are set as initial centroids, or exemplars. From here the algorithms assigns each location to a cluster depending on which centroid it is closest to. It then recalculates the centroid by taking the mean of all the locations per cluster. Finally, it reassigns locations to their nearest centroid again. This process repeats until no locations change cluster. The model can be adapted slightly to assign the closest location to the mean as the centroid. This is known as k-mediod. (Courtney-Marie Bruggeman)

While this technique is widely used, it a number of issues (dev.google). Primarily we need to define the number of clusters ourselves. This requires extra analysis before deciding what value to use for the final model, testing a few different values of k and comparing against some metric. Additionally, k-means can give poor results if a poor spread of initial locations are selected. Because the locations are randomly selected, they may end up all in proximity. The solution to this is to run the algorithm several times until a good solution is found, but this costs time and resources. (dev.google).

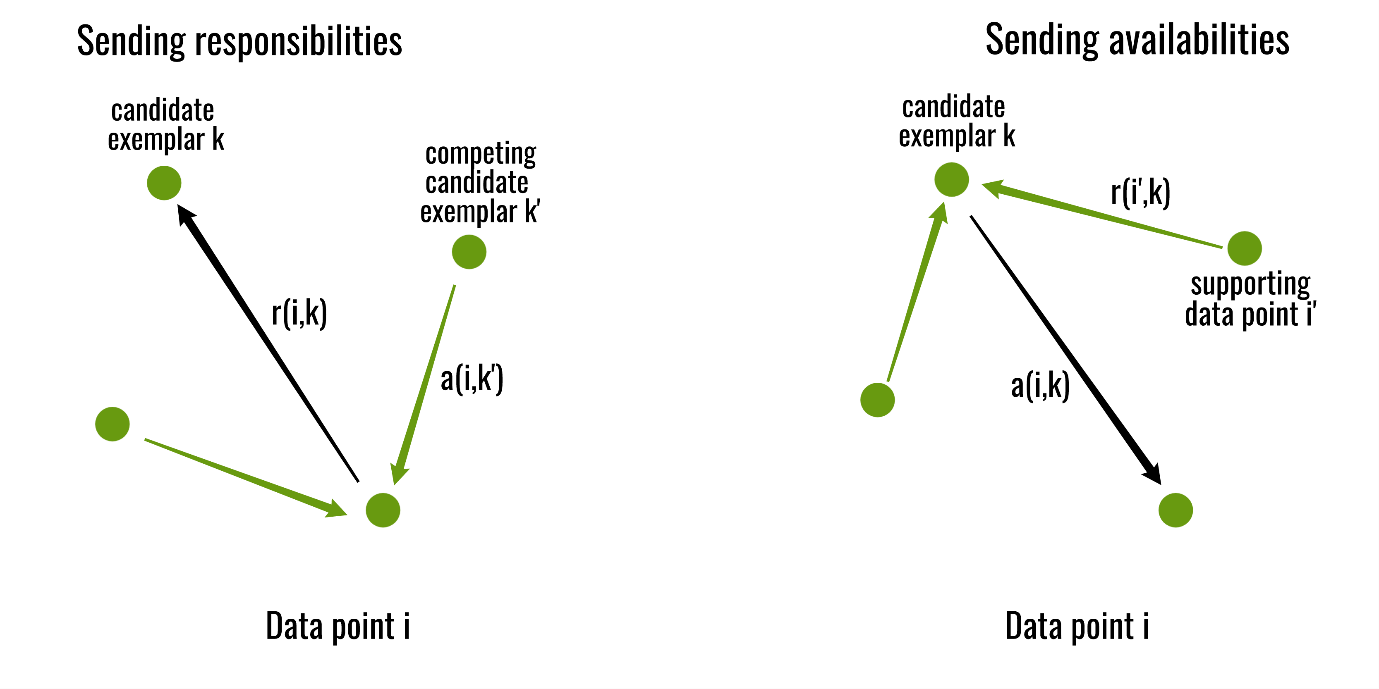
**Affinity Propagation**

Affinity Propagation solves these key issues that k-means faces. It tackles the issue of number of clusters by defining its own, with no input from the user. Because of this is does not fall victim to a poor initial selection of locations.

The process works by alternating between two steps:

* Responsibility. This shows how well fitted a location is to be the exemplar for another, relative to all other locations. Low similarity means a low chance of two points becoming part of the same cluster. This is sent by each location to each other location.
* Availability. This shows how appropriate it would be for a location to pick another location as its exemplar, considering the responsibility score it received from each other location. This is sent back as a reply from each location to each other location.

The process iterates until there is agreement between all locations for where is the best exemplar for where, and no further changes are needed. (Tan L).



(Geeks for geeks) Figure 2 – Affinity propagation steps

Lizhuang Tan et al performed a comparative study of k-means and affinity propagation for clustering with a travelling salesman problem. They surmised that both algorithms showed an improvement in computational cost then when solving the same problem without clustering. Furthermore, they concluded that they prefer affinity propagation because of the sensitivity of k-means to poor initial centroids, as well as a requirement for a pre-set number of clusters.

**Travelling Salesman Problem**

Here we will review several techniques applicable to solving the Travelling salesman problem.

First, we must consider what method we will use to solve the problem. The simplest method of solving the problem is brute force. The method is to run through every possible route and pick the shortest one. This method is possible for a small number of destinations, however as previously mentioned, this is an unrealistic method as the number of possible routes increases so dramatically for each extra item that must be delivered. For drone delivery, this is not a feasible method to use.

An informed search is a type of search that contains an array of knowledge about the search space, such as cost so far and distance from the target. There are a range of algorithms that fall under informed searches, but we will focus on A\* here. A\* has several benefits, but it is mainly used due to being optimal and complete. What this means is that on any given search space, if there is a solution, A\* is guaranteed to find the best one (geeksforgeeks.org).

There are two key issues with using this type of search on this domain. Primarily, we don’t know what the target is for the search to find, other than to have visited every location and have the shortest route possible. If we knew this target, there would be no need to perform a search at all. A\* searches are useful for finding the route to a target and showing how to get there, in our case we don’t care about how to get there, we just want the final route. The second issue we have is hardware limitations. As A\* searches keep track of where they have been until now, if the route we must take is extremely long, it may be unfeasible to run this type of search on our hardware (Stackabuse.com). These limitations add up and make A\* an unsuitable method of solving our problem.

This leads us on to local searches. A local search is unlike an informed search in that it does not keep track of where it has been. Because of this, the issue of hardware limitation is eliminated, as only the current point the search is at is stored in memory. A local search doesn’t know what the final target it is aiming towards looks like. Unlike A\*, local searches carry no guarantee of finding a solution, and if they do find one, it may not be the best solution there is. (Science direct)

**Nearest Neighbour**

Nearest Neighbour is a basic search algorithm where it looks to each of its neighbouring states and selects the one that seems the best. In this domain it would look to every location it hasn’t visited and pick the one that is closest. The search is complete and returns the route that it has found once there are no new places to visit. The issue with this type of search is that it can be stuck in what’s known as a local maximum. This is where the route the algorithm has found cannot be improved upon by the algorithm. It will always find this same route. (Amanur Rahman Saiyed)

**Convex Hull & Cheapest Insertion**

Next there is a combination of techniques that are viable to solve the TSP. The convex hull algorithm is used to create an outside boundary that all locations lie within. It begins the search at an extreme point, such as topmost. Started facing away from the rest of the locations, the search looks clockwise and stops when it finds another point. The process is repeated until it returns to the start point. This gives us our outside boundary. (Engineering book)

From here we use a method called cheapest insertion to visit all the locations that lie within this boundary. This finds every remaining location, and every way to get to this location. It then calculates a ‘penalty’ for travelling to this location. This penalty is the difference between the distance travelled for the new route, and the distance travelled for the old route. The algorithm then selects the route with the lowest penalty and moves on until all locations have been visited. (Engineering book)

**Genetic Algorithms**

Genetic algorithms were first suggested in the 1960s (Bremermann et al). They are intended to simulate evolution as it occurs in nature. They mimic natural selection by selecting only the best individuals to go on to produce more individuals in the next generation. Each individual has a set of characteristics, and they pass this on to their offspring, so strong genes are maintained through generations.

The search begins with a randomly generated set of individuals, or in the case of TSP, routes. These individuals are characterised by a set of parameters and should all be different. From here a fitness score is calculated for each individual. This score determines good the solution is compared to others. Here our fitness score would be length of route. Next, selection occurs. Selection is the method of passing on genes to the next generation. Two pairs of individuals are selected based on their fitness score and move forward.

A process called crossover then occurs. For each pair selected to produce offspring, a crossover point is selected. This is a random point somewhere in the genes. Offspring are generated by exchanging genes within this crossover point. The offspring are added to the population. When offspring are formed, there is a low probability that mutation will occur, meaning that certain properties of the new individual are changed in some way. For TSP this may be a pair of cities switching places randomly within the route.

The process continues until the population has converged. Conversion here means that the offspring being created are not significantly different from the generation creating them. The size of the population does not grow. Once new generations have been formed, individuals that have the lowest score are removed from the search space. (P. Larranaga) (Kylie Bryant)



Fig 3 – Flowchart of a genetic algorithm

**Environments**

Here we will discuss the environment to be used for the development of the system. The three key problems to be solved are:

* Clustering
* Genetic Algorithm
* Graphical User Interface

We will review the suitability of Java and Python for solving these problems. These language have been selected due to familiarity, as

**Java**

Java has retained either 1st or 2nd place for most used programming language for the last 15 years (stackify). It is a very portable language and will run on nearly any system. It

**Summary**

In summary, drones are becoming an increasingly prominent method of delivering items to customers. Due to advances in technology it is becoming more viable to use them. Several large multinational companies have expressed interest in using the technology. There are several issues to be resolved before this becomes reality, one of which is fast and efficient delivery scheduling. Efficient delivery benefits both the company, as they save money, and the customer, as they save time. This project will aim to produce a solution to the scheduling problem.