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, as well as suggested techniques to solve the problem.

# Drone Delivery

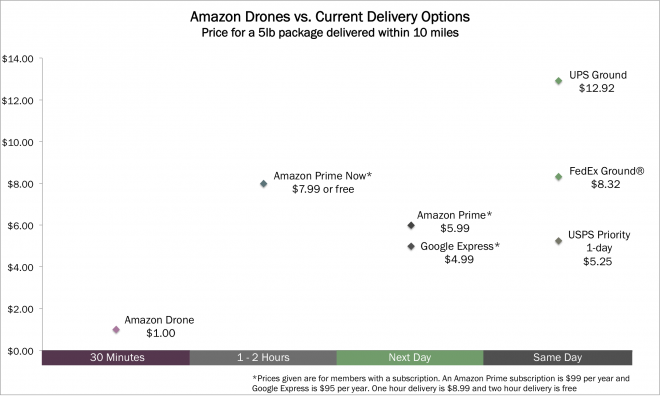
A drone is a small, unmanned flying vehicle. In recent years many companies have unveiled plans to begin delivering packages via drone. In 2013 Amazon announced its ‘Prime Air’ service that will deliver packages to customers (Dorling et al., 2017). DHL has announced its ‘Parcelcopter’ project, which has successfully delivered medicine to the island of Juist in the North Sea (Dorling et al., 2017). Many more companies are doing the same, such as Google and Swiss Post (Dorling et al., 2017). This sudden upsurge in use has been brought on by advancements in technology used in the construction of drones. Improvements in battery technology allow drones to fly faster and further then ever before (Dorling et al., 2017).

There is a great demand from customers for a faster, more reliable option for delivery. This demand 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 & Company, 2016).

**Figure 1 – Amazon Prime Air drone (Amazon)**

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

Research suggests 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 are lower than that of trucks when used for locations close to the depot, or for small numbers of recipients, or both. (Goodchild and Toy, 2018). The speed, cost, and environmental benefits are the critical advantages of delivery via drone. They are the driving force between the recent upsurge in usage.

**Figure 2 – Comparison of delivery cost and time across several mediums and companies (Keeney, 2015)**

# Problem Specification & Boundaries

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

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 (Wilke, 2019). Jeff Bezos has stated that 86% of items delivered by Amazon weigh 5 pounds or less (Guglielmo, 2013). This statistic shows us that Amazon is aiming to deliver one item per drone per trip. Because of this limitation, the current scheduling task is simple. If a drone can only carry one item at a time, a fair solution would be to send items in chronological order. The orders would enter a queue and wait their turn for a drone to be available. If drones were able to carry more weight while retaining a 15-mile range, we can look to solve a more complex problem. The Alta 8 from FreeFly Systems can carry up to 18kg (Freefly Systems, 2019). 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.

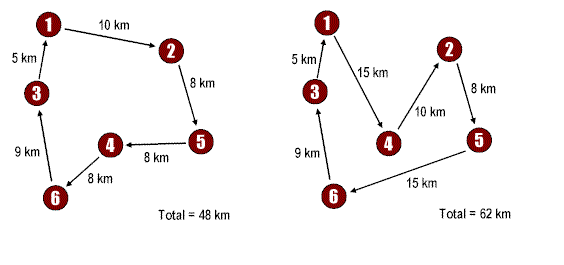
If drones can carry multiple items per flight, it allows the company to deliver to many more customers concurrently, without having to scale up the number of drones they own. With this model, we need to calculate optimal routes and group customer locations so that one drone can serve many people. While we would still have to consider when an order is placed, it would no longer be the only parameter for our scheduling. Someone may move from position one in the queue to position five if we find that a more efficient route is to send a single drone to 4 other people first. With this system, we can still provide accurate delivery estimations while also improving the efficiency of our system. 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 by Euler (Larrañaga et al., 1999). In the 1930s, mathematicians in Vienna and Harvard studied the problem(math.utwaterloo.ca). The problem describes 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 (Saiyed, 2012). The problem of drone scheduling is an example of a TSP. It is scaled down, so instead of a salesman travelling between cities, we have a drone travelling between people’s homes.

At a glance, the problem seems trivial, it is simple to understand, and the method of solving it is not complicated. 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’ (Saiyed, 2012). If the salesman must visit five cities, there are 120 possible routes. If we increase this to 10 cities, there are 362,800 possible routes, and 15 cities give 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. (Bryant, 2000) 

**Figure 3 –Two routes in a travelling salesman problem (essaycorp)**

We will now review several techniques to solve the travelling salesman problem.

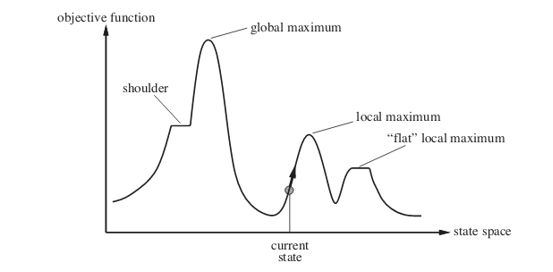
First, we must consider what method we will use to solve the problem. The simplest method of solving the problem is brute force. The method is to run through every possible route and pick the shortest one. This method is possible for a small number of destinations; however, as previously mentioned, this 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 is a range of algorithms that fall under informed searches, but we will focus on A\*. A\* has several benefits but it mainly sees use due to being an optimal and complete algorithm. What this means is that on any given search space, if there is a solution, A\* is guaranteed to find the best one (Nosrati, Karimi, Hasanvand, 2012).

There are two critical issues with using this type of search on this domain. Primarily, we do not 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 do not care about how to get there; we only want the final route. The second issue we have is hardware limitations. As A\* will only end when the best solution is found, the time taken to complete the search can be extremely long (Nosrati, Karimi, Hasanvand, 2012). These limitations add up and make A\* an unsuitable method of solving our problem.

These issues lead us 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, hardware limitations become less of a factor, due to lower memory usage. A local search does not know what the final target it is aiming towards is. 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. (Al-Betar, 2016)

### Hill Climbing

Hill Climbing 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 has not visited and select the closest one. The search is complete and returns the route that it has found once there are no better choices immediately surrounding it. The issue with this type of search is that it can become stuck in a local maximum, where there are no better places for the search to go locally, but there are elsewhere within the domain. (Saiyed, 2012)

**Figure 4 – Hill Climbing algorithm (Geeksforgeeks)**

### Convex Hull & Cheapest Insertion

These are 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 repeats until it returns to the start point. This process gives us our outside boundary. (Goetschalckx, 2011)

From here, we use a method called cheapest insertion to visit all the locations that lie within this boundary. This algorithm 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 are visited. (Goetschalckx, 2011)

### Genetic Algorithm

Bremermann et al first proposed genetic algorithms in 1965 (Larrañaga et al., 1999). 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 the algorithm maintains healthy genes through generations.

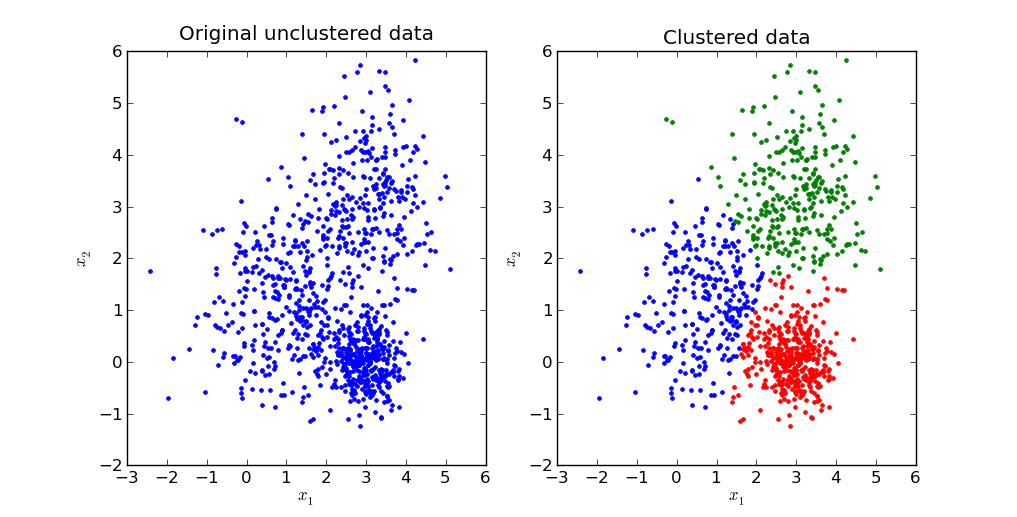
The search begins with a randomly generated set of individuals, or in the case of TSP, routes. These individuals are characterised by a set of parameters, and should all be different. From here, a fitness score is calculated for each individual. This score determines how good the solution is compared to others. Here our fitness score would be the length of the 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.

**Figure 5 – Genetic Algorithm flow chart (Apache ignite)**

A process called crossover then occurs. For each pair selected to produce offspring, a crossover point is selected. The crossover point is a random point somewhere in the genes. Offspring are generated by exchanging genes within this crossover point, and the offspring are added to the population. When offspring are formed, there is a low probability that mutation will occur, meaning that specific properties of the new individual change 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 that created 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. (Larrañaga et al., 1999) (Bryant, 2000)

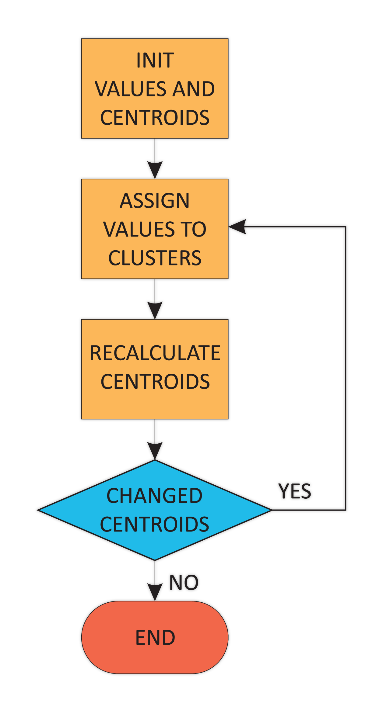
## Clustering

Because of both 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 three sets of five, 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. 

**Figure 6 – Clustering data into 3 clusters (mubaris)**

Clustering also allows us to model a delivery system with multiple drones delivering simultaneously, which is more realistic than just creating one route for one drone to carry out. If we assume that drones will be able to carry more weight but fly the same distance as they currently can, we still need to create clusters. We do not want drones to be flying randomly from one edge of their range to the other, but instead to deliver to a few tightly grouped locations and return to the depot. These two factors show the need for clustering on our problem.

### 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 algorithm assigns each location to a cluster depending on which centroid is nearest. 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 adjustment is known as k-mediod. (Bruggeman et al., 2010)

**Figure 7 – K-means clustering (oreilly)**

While this technique sees wide use, it has several issues (Google Developers, 2019). Primarily we need to define the number of clusters ourselves. To do this requires additional 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 is selected. Because the locations are randomly selected, they may end up all in proximity. The solution is to run the algorithm several times until it finds a good solution; however, this costs time and resources. (Google Developers, 2019).

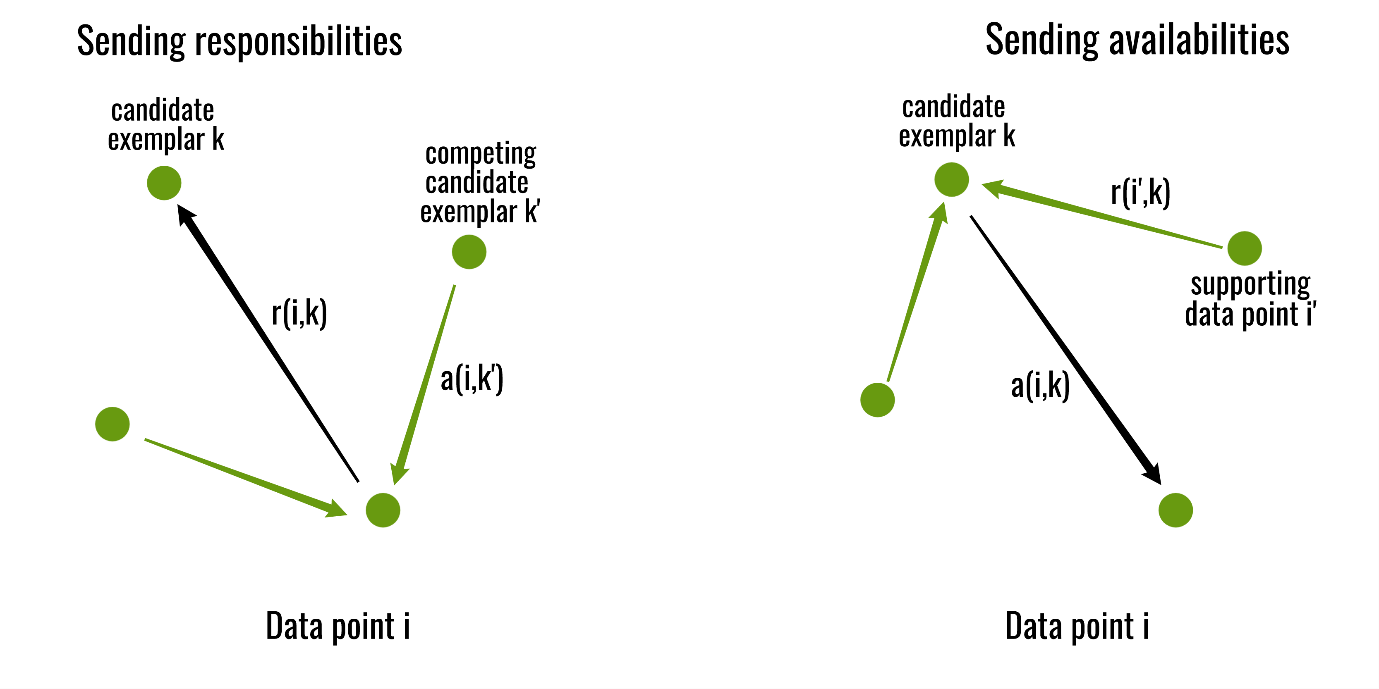
### Affinity Propagation

Affinity Propagation solves these key issues that k-means faces. It tackles the issue of determining how many clusters to have; it defines its own, with no input from the user. Because of this, it does not fall victim to a poor initial selection of locations.

The process works by alternating between two steps:

* Responsibility. This step 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. Each location sends this similarity to each other location.
* Availability. This step 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 score is sent back as a reply from each location to each other location.

The process iterates until there is an agreement between all locations for where is the best exemplar for where, and no further changes are needed. The process iterates until there is an agreement for which nodes are the best exemplars. (Tan et al., 2016).



**Figure 8 – 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 than 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.

# 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 a 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 solve the scheduling problem.

# Conclusion

Based on findings in the literature above, we can select several techniques to create a solution for drone delivery scheduling. Some constraints of drones will be relaxed, and we will assume they can carry more weight and, thus, multiple items at once. Research suggests that breaking customer locations into clusters is essential so that the solution is likely to be relevant in the near future. For this, we will use the affinity propagation technique outlined in section 2.2.2, as our research shows that this method is more fitting to our domain then k-Means or k-Mediods. Once the problem is broken into clusters, we are left with multiple travelling salesman problems to solve. We will use a genetic algorithm to solve these. The genetic algorithm has been chosen as it is a stimulating technique to learn and has seen much application on solving travelling salesman problems. This leads to the conclusion that it is a very fitting solution to this task.

# References

Al-Betar, M. (2016). β-Hill climbing: an exploratory local search. Neural Computing and Applications, 28(S1), pp.153-168.

Bruggeman, C., Liu, W., Mitchell, M. and You, E. (2010). Creating Clusters: An Analysis of Traveling Salesman as a Clustering Mechanism.

Bryant, K. (2000). Genetic Algorithms and the Traveling Salesman Problem.

Dorling, K., Heinrichs, J., Messier, G. and Magierowski, S. (2017). Vehicle Routing Problems for Drone Delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(1), pp.70-71.

Freefly Systems. (2019). Freefly ALTA 8 Specifications - Dimensions, Weight & Payload. [online] Available at: https://freeflysystems.com/alta-8/specs [Accessed 6 Nov. 2019].

Goetschalckx, M. (2011). Supply Chain Engineering. Boston, MA: Springer US, pp.229-240.

Goodchild, A. and Toy, J. (2018). Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO 2 emissions in the delivery service industry. Transportation Research Part D: Transport and Environment, 61, p.66.

Google Developers. (2019). Interpret Results and Adjust Clustering. [online] Available at: https://developers.google.com/machine-learning/clustering/interpret [Accessed 6 Nov. 2019].

Google Developers. (2019). k-Means Advantages and Disadvantages | Clustering in Machine Learning. [online] Available at: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages [Accessed 6 Nov. 2019].

Guglielmo, C. (2013). Turns Out Amazon, Touting Drone Delivery, Does Sell Lots of Products That Weigh Less Than 5 Pounds. [online] Forbes.com. Available at: https://www.forbes.com/sites/connieguglielmo/2013/12/02/turns-out-amazon-touting-drone-delivery-does-sell-lots-of-products-that-weigh-less-than-5-pounds/#4a56fbe6455e [Accessed 6 Nov. 2019].

Keeney, T. (2015). Drone Delivery: How Can Amazon Charge $1 for Drone Delivery?. [online] ARK Investment Management. Available at: https://ark-invest.com/research/drone-delivery-amazon [Accessed 6 Nov. 2019].

Larrañaga, P., Kuijpers, C., Murga, R., Inza, I. and Dizdarevic, S. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review, 13(2), pp.129-133.

Lee, H. (2016). TECHNOLOGICAL DISRUPTION AND INNOVATION IN LAST-MILE DELIVERY. [online] p.11. Available at: https://www.gsb.stanford.edu/sites/gsb/files/publication-pdf/vcii-publication-technological-disruption-innovation-last-mile-delivery.pdf [Accessed 6 Nov. 2019].

McKinsey & Company (2016). Parcel delivery. The future of of last mile. [online] pp.9-9. Available at: https://www.mckinsey.com/~/media/mckinsey/industries/travel%20transport%20and%20logistics/our%20insights/how%20customer%20demands%20are%20reshaping%20last%20mile%20delivery/parcel\_delivery\_the\_future\_of\_last\_mile.ashx [Accessed 6 Nov. 2019].

Nosrati, M., Karimi, R. and Hasanvand, H. (2012). Investigation of the \* (Star) Search Algorithms: Characteristics, Methods and Approaches. World Applied Programming, p.251.

Saiyed, A. (2012). The Traveling Salesman Problem. p.2.

Tan, L., Tan, Y., Yun, G. and Wu, Y. (2016). Genetic Algorithms Based on Clustering for Traveling Salesman Problems. pp.103-107.

Wilke, J. (2019). A drone program taking flight. [online] US Day One Blog. Available at: https://blog.aboutamazon.com/transportation/a-drone-program-taking-flight [Accessed 6 Nov. 2019].

Figures

Fig 1 - Amazon <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>

Fig 2 - Keeney, T. (2015). Drone Delivery: How Can Amazon Charge $1 for Drone Delivery?. [online] ARK Investment Management. Available at: https://ark-invest.com/research/drone-delivery-amazon [Accessed 6 Nov. 2019].

Fig 3 – Essaycorp <https://blog.essaycorp.com/travelling-salesman-problem/>

Fig 4 – Geeksforgeeks <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/>

Fig 5 – ApacheIgnite <https://apacheignite.readme.io/docs/genetic-algorithms>

Fig 6 – Mubaris <https://mubaris.com/posts/kmeans-clustering/>

Fig 7 – Oreilly <https://www.oreilly.com/library/view/machine-learning-for/9781786469878/a01f88c8-59a6-4ba4-9c3d-f00b8f309b70.xhtml>

Fig 8 – Geeksforgeeks <https://www.geeksforgeeks.org/affinity-propagation-in-ml-to-find-the-number-of-clusters/>