The University Of Sheffield.

Aerospace Engineering

Final Project Report Cover Sheet

Surname: Mythen Forenames: Alice

Registration number 1701554415

Module code: AER429

Module name: Aerospace Individual Project

Assignment title: Knowledge Graph Analysis for Improving Sustainability in Engineering Applications

Supervisor: Prof David Wagg

Date submitted: 10/05/21

Student Declaration

I confirm that the work I have submitted is my own original work and that I have made appropriate references to any sources used.
I agree to any submitted materials being used by the Department and/or University to detect and take action on any unfair means.
I confirm that the printed version of this assessment and any electronic copy I am required to submit to TurnItIn are identical.
I understand that failure to meet these guidelines may result in one or more of the following penalties being imposed:

1. A reduced mark (including a mark of zero) being recorded for this work.
2. A reduced mark (including a mark of zero) being recorded for the unit/module.
3. A formal reprimand (warning) being placed on my student record at Departmental and/or University level which may be taken into account whenever references are written.
4. Reporting of the incident to the University Disciplinary Committee.
5. Suspension or expulsion from the University.

Student's signature: A.Mythen Date: 10/05/21

Aerospace
Engineering

UNIVERSITY OF SHEFFIELD

AEROSPACE ENGINEERING
FINAL YEAR PROJECT

KNOWLEDGE GRAPH ANALYSIS FOR IMPROVING SUSTAINABILITY IN ENGINEERING APPLICATIONS

Author
A.V. MYTHEN

Registration Number
170154415

May, 2021

Supervisor: Prof David Wagg

Dissertation submitted to the University of Sheffield in partial fulfilment of the requirements for the degree of
Master of Engineering

Abstract

With the aim of this project being to develop an idea involving knowledge graphs and digital twins to improve environmental sustainability in a chosen engineering application, literature around these subjects were reviewed. It was then decided to construct and use a knowledge graph of air traffic coming in and out of Heathrow airport to analyse the data in attempt to discover a way to reduce CO_2 emissions. It was also decided to explore the idea of having a digital twin of an airport, and possibly combining this with the knowledge graph. However, due to time constraints this concept did not get explored further but is discussed in the conclusions and future work Section. The method for collecting the flight data and constructing the knowledge graph are described and from the analysis of the data an observation was made that some flights were scheduled to leave at the same time, to the same destination. This led to the exploration of a concept proposed by Airbus, that an aircraft could use the upwash coming off a leading aircraft's wing for free lift and save between 5-10% of its fuel consumption. A method for finding flights leaving Heathrow that could utilise this technique is described, along with the calculations for the estimated fuel savings.

Contents

Nomenclature

ACM	Airport Capacity Management
CO^2	Carbon Dioxide
CFD	Computational Fluid Dynamics
FC	Fuel Consumption
FCFS	first-come-first-serve
FCO	Fuel Consumption Optimization
FT	Flight Time
KGs	Knowledge Graphs

Acknowledgement

I would like to thank my project supervisor, Professor David Wagg, for providing brilliant support throughout the course of the project. My thanks also to two of his PhD students who kindly took the time to meet with me to give their guidance and advice. And finally, I would like to thank the Director of Aerospace Engineering, Dr Robert Howell, for his helpful video updates throughout the year and continuous words of encouragement.

1. INTRODUCTION

Over the years there has been a rapid development of knowledge graphs being used to model unstructured and time-evolving network information. A knowledge graph simply describes objects of interest and connections between them [1] and over recent years seem to be used for increasingly complex applications. A current example being the way in which virtual assistants provide answers to users' questions [2]. There have been some discussions, for example in a paper discussed later on by Agniva Banaerjee et al. [2], of how knowledge graphs might be used in the context of digital twins in engineering applications. A digital twin is a virtual model of a process, product or service [3] and like knowledge graphs, digital twins have been gaining more attention over the years. With environmental sustainability being one of the biggest issues faced by mankind at present [4], the following project aim was decided:

Project Aim: To develop an idea involving knowledge graphs and digital twins to improve environmental sustainability in a chosen engineering application.

In order to achieve this aim the following list of objectives was made:

- Explain the motivation behind trying to improve sustainability, drawing upon the current facts and figures i.e., carbon emissions.
- Understand and explain what knowledge graphs and digital twins are and find examples of how they are currently being used for engineering applications.
- Find examples or discussions of how knowledge graphs might be used in the context of digital twins in engineering applications.
- Explore ideas in which knowledge graphs and digital twins can be used individually or together to improve sustainability in engineering applications and choose one idea to develop.
- Find data and carry out calculations to support the developed idea i.e., prove it will improve sustainability.
- Discuss the cost and benefits of producing and using the chosen idea.

From these objectives a Gantt chart was created with more specific tasks and can be found in Appendix 1. Once the idea was chosen a new more specific list of objectives was created and added to the updated Gannt chart which can be found in Section 9, figure 9.1. The rest of the report is organised as – In Section 2 gives some background information on knowledge graphs and digital twins. In Section 3 relevant literature and related work has been researched and reviewed. Section 4 describes the journey of how the knowledge graphs were constructed. In Section 5, the developed idea will be discussed, and calculations carried out proving whether it will improve environmental sustainability or not. Section 6 will discuss all the work that has been carried out and which objectives were not achieved due to time constraints. The report will be summarised in Section 7, with suggestions for future work. Sections 9 and 10 are project management and self-review respectively, evaluating the projects progress and development.

2. BACKGROUND

2.1 Knowledge Graphs

Hogan, A. et al. [5] adopted the definition of a knowledge graph to be "a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities". Entities can be any real-world object or event, or abstract concept (e.g., documents) [6]. One of the most well-known graphs is the Google Knowledge Graph, which was announced in 2012 [7]. It is used to provide users with more relevant search results [8], as it can make connections between related entities and uses the searches from previous users to provide information from what they found useful. Since Googles, many further announcements of the development of knowledge graphs have been made, including Amazons, who have said they improve search, recommend products, and infer missing information from information extracted from KGs in the form of embeddings [9]. Knowledge graph embeddings can be used to infer relations between entities that are not explicitly given by the data, using score functions to measure the distance of two entities to its relation type.

2.2 Digital Twins

Although the term "Digital Twin" didn't gain recognition until 2002 [10], Stephen Ferguson, along with many others, believes 'one of the best real-life examples of a digital twin in action' was the Apollo 13 mission, 32 years prior [11]. When the oxygen tanks exploded, NASA's engineers were able to test possible solutions using the digital twin model of Apollo 13 on earth. Nowadays digital twins are a staple in modern engineering and can be used across many different industries for various applications. As well as testing proposed solutions, like during Apollo 13, they can be used for digitally testing proposed products before they are physically produced and are very commonly used for optimising the performance of real assets. Real-time data is collected from sensors on the physical asset. The data is then evaluated and simulated in the virtual copy to find optimization strategies which can then be applied to the real asset. Digital twins are becoming increasingly complex, with the integration of the internet of things, artificial intelligence and software analytics [12].

3. LITERATURE REVIEW

3.1 Introduction

The following Section will cover the literature which has been reviewed in relation to the subjects covered in this project. Subsection 3.2 highlights the need for improving environmental sustainability and the literature was used to determine which area to focus on. Subsection 3.3 reviews work that has already been done with regards to knowledge graphs and digital twins, especially in relation to engineering applications and environmental sustainability, to identify a gap in knowledge to be explored. The final subsection covers the literature which was reviewed once the direction of the project had been determined, therefore related more towards Heathrow airport, air traffic and fuel saving techniques.

3.2 Motivation behind the aim of the project

In 2018, Arora, N.K. [4] published an article called ‘Environmental Sustainability – necessary for survival’. If the title alone doesn’t indicate the importance of the issue, the contents does, where they discuss the devastating impacts of increasing greenhouse gas emissions. In 2019 the government reported 80% of greenhouse gas emissions in the UK was carbon dioxide [12]. 73.2% of that CO_2 emitted is said to come from the energy sector which consists of electricity, heat and transport [14]. Within transport 1.9% of emissions come from aviation and in 2019, Schafer et al. [15] said the aviation sector has the largest growth rates for greenhouse gas emissions within the transportation sector, within the developed countries. In 2019, Terrenoire, E. et al. [16] assessed the climate impact of civil aviation CO_2 emissions and found that these emissions were only one aspect on the possible impact on the climate and other agents. For example, NO_X emitted by aircrafts also impact the climate. For these reasons this project will focus towards reducing CO_2 emissions in the aviation sector.

3.3 Previous work relating to knowledge graphs and digital twins

The idea of knowledge graphs being used in the context of digital twins appears to be a very recent concept with limited literature on the subject. In 2017, Agniva Banarjee et al. [2] introduced a way of formalizing knowledge as digital twin models coming from sensors in industrial production lines. In July 2020, an enterprise called Stardog [17] put out a blog post explaining the critical need for moving businesses fully "online" by creating a digital twin with an enterprise knowledge graph. Although the motivation behind the post seemed to be more to advertise their product, it does support the concept of knowledge graphs being used to create digital twins.

Kaewunruen. S. et al. [18] evaluated the technical and financial viability of turning *'existing'* buildings into Net Zero Buildings, using digital twins. Articles from TWI Ltd [19] and Recharge [20] discuss the idea of using digital twin technology for real-time assessments of the structural conditions of wind turbines in order to extend their lives by employing the appropriate maintenance when needed. In 2019, both O'Sullivan, K. from SITA [21] and Ørsted, M from Copenhagen Airport [22] put out separate blog posts describing their concepts to predict and control the events in an airport using digital twin technology.

Qian Zhao et al. [23] put forward a knowledge graph construction method in the field of aviation risk in order to achieve the efficient organization and knowledge sharing of the historical cases of aviation. Although this paper does not relate to improving environmental sustainability, it very clearly goes through the steps they follow to construct their knowledge graph which could be applied to constructing one's own knowledge graph. Richard M.Keller [24] developed a knowledge graph of air traffic across an area in the US which could then be queried to help answer active research questions. Again, this is not directly related to environmental sustainability, however, as said in the paper the availability of the data provides the possibility of analysing the data to discover new ways to improve air transportation efficiency, in turn reducing fuel consumption.

In July 2020, Kaltenböck, M, et al. [25] published a paper explaining how creating a global climate action knowledge graph can better connect people, organisations and businesses in the global quest to tackle major climate-related issues by sharing data so they can learn with and from one another. The paper is well related towards the aim of this project, although not in an engineering application, gives good background information about knowledge graphs and examples of how organisations are using them today. It also provides steps to develop the climate action knowledge graph, however not in a way that someone could follow to construct their own graph. In an online article published in 2020 [26], Thorton, C. discusses their method for constructing a knowledge graph, the different tools used and provides their own python code, making it easier to follow. Although the knowledge graph they constructed has no relevance to the aim of this project, it does prove Python to be a good programming language to use to construct a knowledge graph, along with several of the packages within Python, like Pandas and NetworkX.

After considering all the papers and articles presented above it can be concluded that both knowledge graphs and digital twins are being used more and more in recent years. However, the literature about knowledge graphs being used in the context of digital twins is still limited and no literature on them being used together in the context of environmental sustainability was found. Taking inspiration from Richard M. Keller [24] this project will explore the idea of using a knowledge graph to analyse the air traffic coming in and out of Heathrow airport in an attempt to discover a way to reduce fuel consumption and therefore CO_2 emissions. Similar to SITA [21] and Copenhagen Airport [22] the concept of having a digital twin of an airport will also be further explored, and the possibility of combining the data from the knowledge graph with the digital twin.

3.4 Literature relating to air traffic and fuel saving techniques

Heathrow is currently using an air traffic control service called NATs. From an article on Heathrow's Strategic Airport Capacity Management [27], NATs is said to be 'a leading air navigation services specialist' and has a reputation 'for operational excellence and innovation'. It has benefited Heathrow with its strategic ACM (Airport Capacity Management) tool by using simulations and data analysis to support decision-making and was able to create a new early morning runway slot, which is extremely rare and hadn't been done since 1996. In this article, however, and on the NATs Heathrow Airport Factsheet [28] there is no mention of environmental impacts or a desire to reduce fuel consumption. These articles are also extremely biased as they are written by NATs, and therefore do not include information regarding issues Heathrow still has with air traffic. Other articles online [29] show, regardless of this innovative air traffic control service, Heathrow does experience delays in its flight schedule. In

2018, Heathrow released its Carbon Footprint Report [30] detailing a breakdown of its current carbon emissions and its strategies for reducing them in the future, from changing the way people arrive at the airport to encouraging cleaner and quieter aircraft. The report shows a wide range of strategies for various aspects of the airport, not just emissions from the aircrafts, however, none of which focus on air traffic, leaving a gap to be explored.

In 1991, Neuman, F. et al. [31] analysed delay reducing and fuel saving sequencing and spacing algorithms for arrival traffic. They express the importance of scheduling to bring order to arriving aircraft so sequencing and spacing algorithms can be applied. They conclude that the first-come-first-serve (FCFS) scheduling method is fair, determines proper separations and the order remains preserved when strategies are put in place to reduce delays. In 2008, Rathinam, S. et al. [32] presented an optimization model for reducing aircraft taxi times, to compete with the FCFS algorithm, and demonstrated with preliminary simulation results that approximately six minutes of taxi time per aircraft could be saved at Dallas Fort Worth International Airport. Although the 1991 article was published by NASA and therefore has more accreditation, it is from 30 years ago and therefore could be seen as outdated.

Singh, V. et al. [33] reviewed 277 articles on fuel consumption optimization (FCO) in air transport published between 1973 and 2014. From their review they found that 'the aviation sector's fuel efficiency improvements have slowed down since the 1970s-1980s due to the slower pace of technological development in engine and aerodynamic designs and airframe materials', however the trend in research of the FCO is increasing, showing the interest is there. In March 2021, Tasca, A.L. et al. [34] published an article promoting their PARSIFAL project involving an innovative box-wing aircraft which, through CFD simulations, they have estimated will reduce fuel consumption by 20% and believe will bring a considerable reduction of climate change. As impressive as these savings sound, this project seems to be in the very early stages, with only CFD simulations so far.

In 2019, Airbus [35] released an article introducing the concept of aircraft adopting the flight patterns of migrating birds, in order to save fuel, known as wake-energy retrieval. The theory is for a leader aircraft to provide free lift to a following aircraft, from its upwash. Although at first this technique seems dangerous, as for years aircraft have tried to avoid flying in one's wake, test flights have been demonstrated and adopting this technique is expected to produce fuel savings of between 5-10% per trip. As Airbus is a huge, well known company and flight tests have been demonstrated, this project seems to have a legitimate future. Another recently proposed technique for reducing flight emissions was one published by Wells, C.A. et al. [36] who demonstrates that altering flight routes to exploit wind fields when flying from London to New York, can substantially reduce air distance travelled and therefore fuel consumption. A very good point made in the article is that this technique is possible in the short term, whereas technological advances for improving fuel-efficiency tend to take many years to implement.

From the literature reviewed, it is apparent that reducing fuel consumption in the aviation sector is a popular goal and has been growing more and more over the years. New innovative ideas are being published and some are even in the testing phases. Having this understanding of what ideas are out there already will be useful when analysing Heathrow's flight data, as some of these concepts and techniques might be able to be applied or adapted to reduce the fuel consumption of flights leaving and arriving at Heathrow airport.

4. THE KNOWLEDGE GRAPH

4.1 Constructing the knowledge graph

As mentioned in the previous Section, this project aims to explore the idea of using a knowledge graph to analyse the air traffic coming in and out of Heathrow airport in an attempt to discover a way to reduce fuel consumption and therefore CO_2 emissions. In order to do so, a knowledge graph must first be constructed. The next 2 subsections will go through the method which was followed to collect the necessary data and format it into a way it could be used, and then how that data was used to construct the knowledge graphs.

4.1.1 Data collection and formatting for the knowledge graph

The knowledge in the graph that was to be produced was flights coming in and out of Heathrow airport, therefore this data had to be collected. Before doing so an ontology was defined in order to determine what flight information would need to be found. The ontology involved only the classes and relations deemed relevant to the task and can be seen in table 4.1. Departures and arrivals can be found on the Heathrow website for flights on that day. However, it is not presented in a way that can be easily transferred into an excel spreadsheet and does not include the aircraft types for each flight, which as shown in table 4.1 was a desired class.

***Table 4.1** Ontology*

Classes	Relations
Departure Date/Time	Departs on
Aircraft	Aircraft flown
Country/Code	Departing to
Airline Name	Operated by
Terminal Number	Depart from
Flight Code	

For these reasons, a different live air traffic website [37] was used. The data of one day's worth of flights, arrivals and departures, was copy and pasted into an excel spreadsheet. The first 5 lines of the raw data of departures can be seen in table 4.2. Some formatting was done, and further information added in order to get it into the desired format to construct the knowledge graph.

Table 4.2 *Raw data from live air traffic website [37] accessed 5/11/2020*

Dep Time:	Flight:	Destination:	Airline:	Aircraft:
6:25	AF1081	Paris (CDG)	Air France	A319 (F-GRHY)
6:25	BA1414	Belfast (BHD)	British Airways	A319 (G-DBCJ)
6:30	KL1000	Amsterdam (AMS)	KLM	B738 (PH-BGA)
6:30	LH921	Frankfurt (FRA)	Lufthansa	A20N (D-AINF)
7:10	BA2770	Jersey (JER)	British Airways	A319 (G-EUPD)

It can be seen that 5 out of the 6 desired classes are included in the table. The terminal number of each flight, however, is missing. This was easily added, as certain airlines are associated with particular terminals and can be found on the Heathrow website [38]. Also, due to travel restrictions because of the COVID pandemic, only terminals 2 and 5 were in operation. By using the filter function on excel, the terminals associated with terminal 5 (British Airways, China Southern, Iberia, Iberia Express, Japan Airlines and Qatar Airways) were selected and a sixth column was added for terminal number. Then terminal 2 was assigned to all the other airlines.

As you can see the fifth column shows the aircraft code followed by the registration number unique to that flight. To make it clearer to the user what aircraft make was being flown, it was decided to replace the aircraft code with the actual make i.e., Airbus A319-100 instead of just A319. In order to do so, firstly the registration numbers in brackets had to be removed. This was done using an excel code which can be found in Appendix 2, along with all other excel codes mentioned in this report. Then a list was created with all the aircraft codes in one column and their respective aircraft types in the column beside them. This information was again found on the live air traffic website [37] and can be seen in Appendix 5. Using the VLOOKUP function in excel, the aircraft types were assigned to the list of the flights and were moved to the aircraft column. The first 5 lines of the formatted data of departures can be seen in table 4.3. The text colour and font were changed, and hyperlinks removed for aesthetics. Now the flight data was in the desired format with all the relevant information, the knowledge graph could begin to be constructed.

Table 4.3 *Formatted data from live air traffic website [37] accessed 5/11/2020*

Dep Time:	Flight Code:	Destination:	Airline:	Aircraft:	Terminal:
6:25	AF1081	Paris (CDG)	Air France	Airbus A319-100	Terminal 2
6:25	BA1414	Belfast (BHD)	British Airways	Airbus A319-100	Terminal 5
6:30	KL1000	Amsterdam (AMS)	KLM	Boeing B737-800	Terminal 2
6:30	LH921	Frankfurt (FRA)	Lufthansa	Airbus A320	Terminal 2
7:10	BA2770	Jersey (JER)	British Airways	Airbus A319-100	Terminal 5

4.1.2 Coding the knowledge graph

As shown in the Gannt chart in Section 9, figure 9.1, an objective was to first produce a knowledge graph of 50 departure flights leaving Heathrow. This number was chosen to begin with so the knowledge graph was not too crowded on screen, meaning it could still be visually analysed. To first get a sense of what the knowledge graph would look like, software called protégé was used to visualise the ontology defined in table 4.1. This graphical representation can be seen in figure 4.1. An initial idea was to import the data from the excel spreadsheet into protégé to create the knowledge graph, however a method was not found. In one of the articles reviewed in Section 3.3 [26] the author details their method of constructing a knowledge graph using Python, so it was decided this approach would be explored.

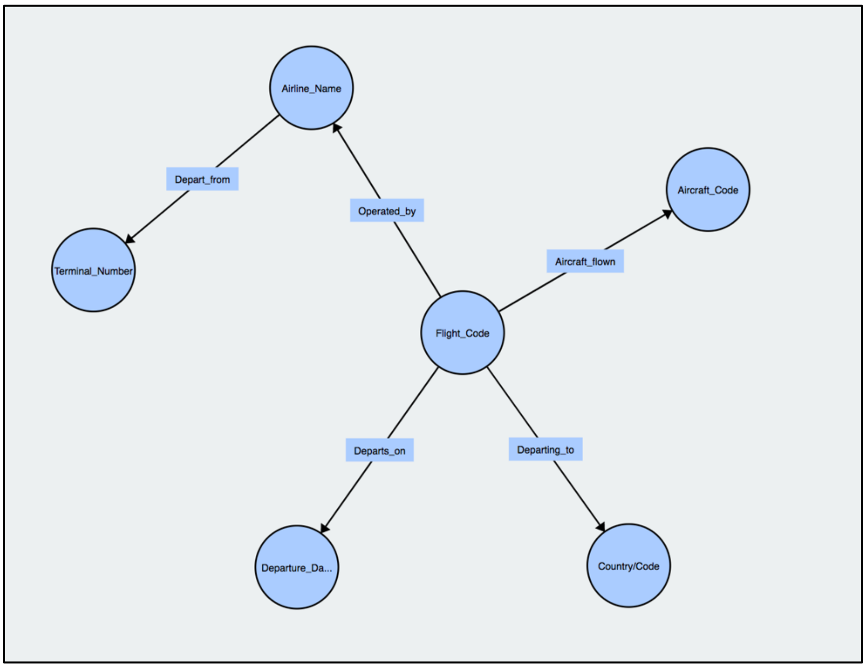


Figure 4.1 *Ontology produced using protégé*

A Python package called NetworkX allows the construction of complex networks made up of nodes and edges [39]. Using this package, a simple knowledge graph was constructed, very similar to the one produced in Protégé. The classes of the ontology were represented by nodes and the relations were edges. A package called matplotlib.pyplot was used to plot the graph, which can be seen in figure 4.2. The code for this can be found in Appendix 2.1.

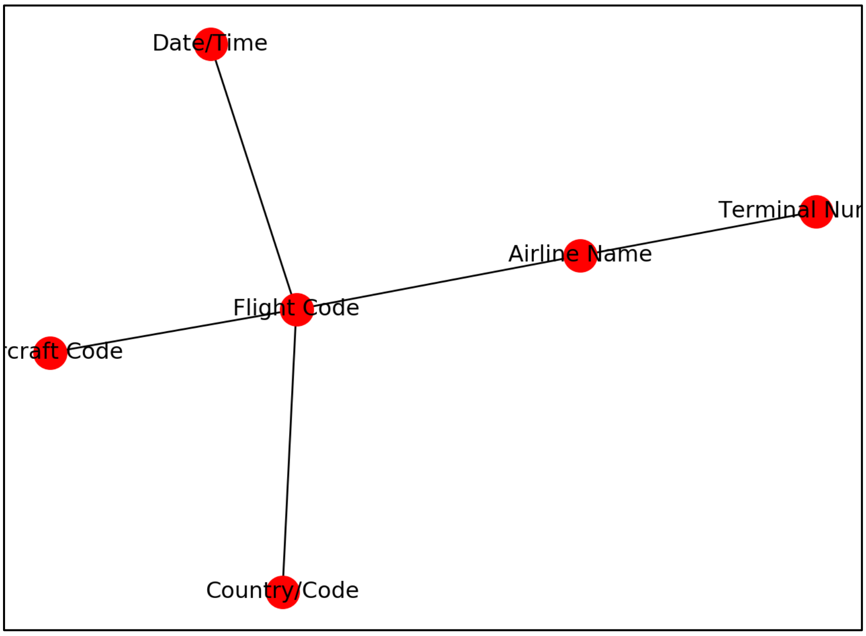


Figure 4.2 *Graph of ontology produced using Python*

Next a graph, shown in figure 4.3, was created with 5 flights, by manually inputting each individual flight into the code. This code can be found in Appendix 3.2. This would obviously be an incredibly inefficient use of time if it were to be done with 50 flights, so a way of importing the data from the excel file into Python was required. This was done using the Python package xlrd. First, just the same 5 flights were imported, to try to recreate the same graph as shown in figure 4.3. This was successfully done with the help of a for loop. The code for these can be found in Appendix 3.3. After visually checking both graphs matched, the same code was used with 50 flights to produce the graph shown in figure 4.4.

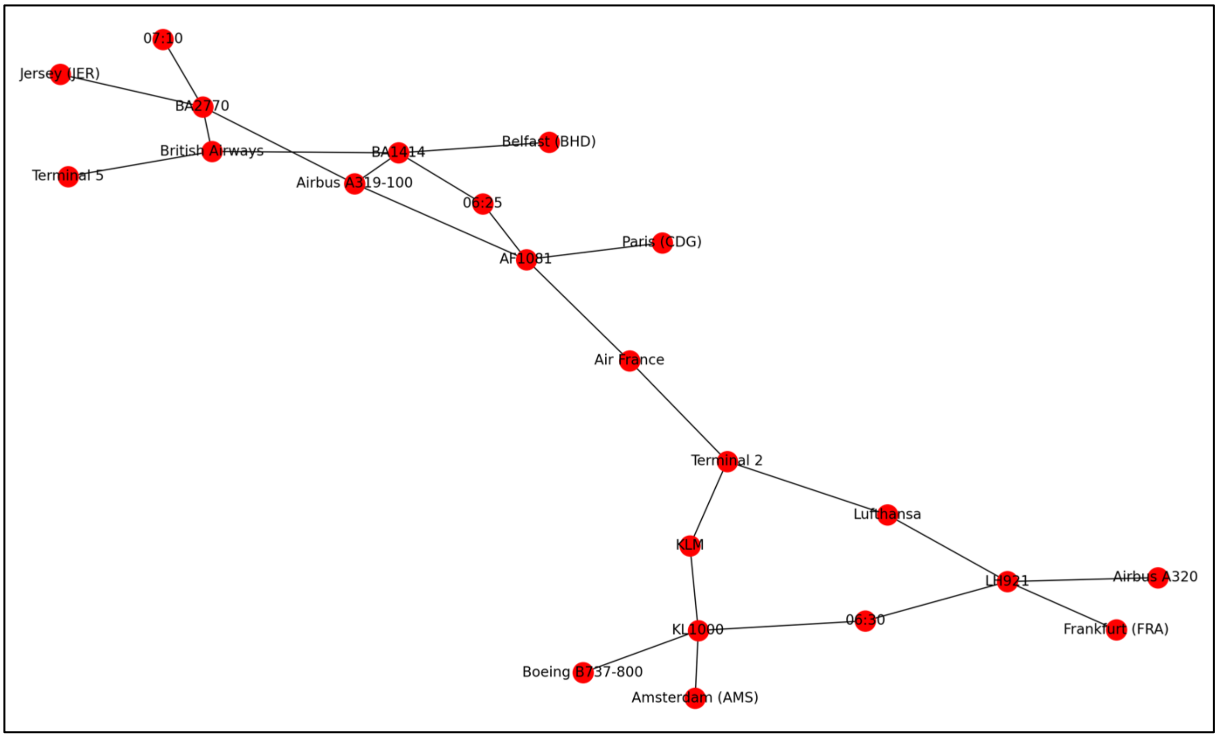


Figure 4.3 *Graph of 5 flights produced using python*

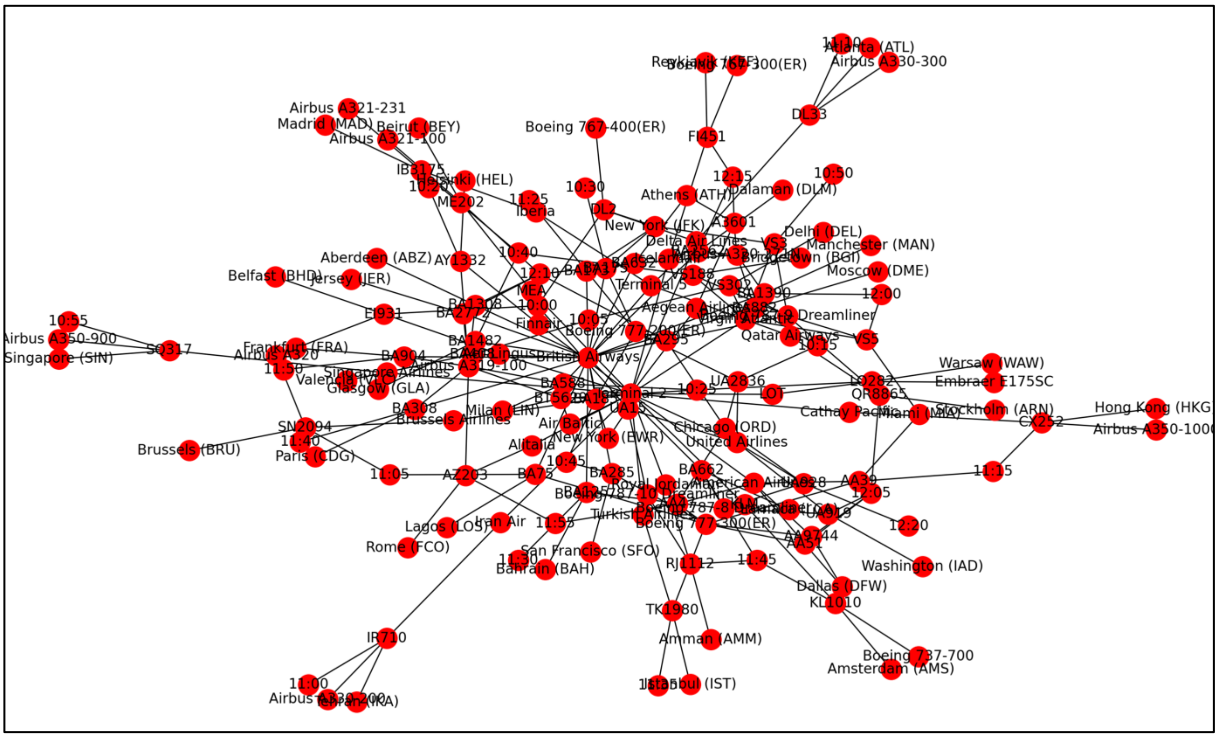


Figure 4.4 *Graph of 50 flights produced using python*

Next the arrival flights were added to the graph. This was done by adding the arrival flights to the excel sheet below the departure flights and adding a column for origin, referring to the airport that flight has come from. In the destination column 'Heathrow (LHR)' was added to each flight and for the departure flights 'Heathrow (LHR)' was added in the origin column. The code was adjusted slightly to include the arrivals, and in order to make the graph easier to read and therefore analyse the data, a slight change to the size and colour of the nodes, edges and font were made. The graph produced is shown in figure 4.5. As you can see the graph is very crowded making it difficult to analyse the data. It is also impossible to determine which flights are departing and which are arriving. An idea to overcome this was to colour code the nodes by classes. Unfortunately, a quick method as to do this was not found and it was concluded that it was not an efficient use of time. Instead, it was decided to make two separate graphs. One for departures and one for arrivals. This would also help with the overcrowding issue meaning each graph would be easier to analyse. The code for these can be found in Appendix 3.5 and the graphs in figures 4.6 and 4.7.

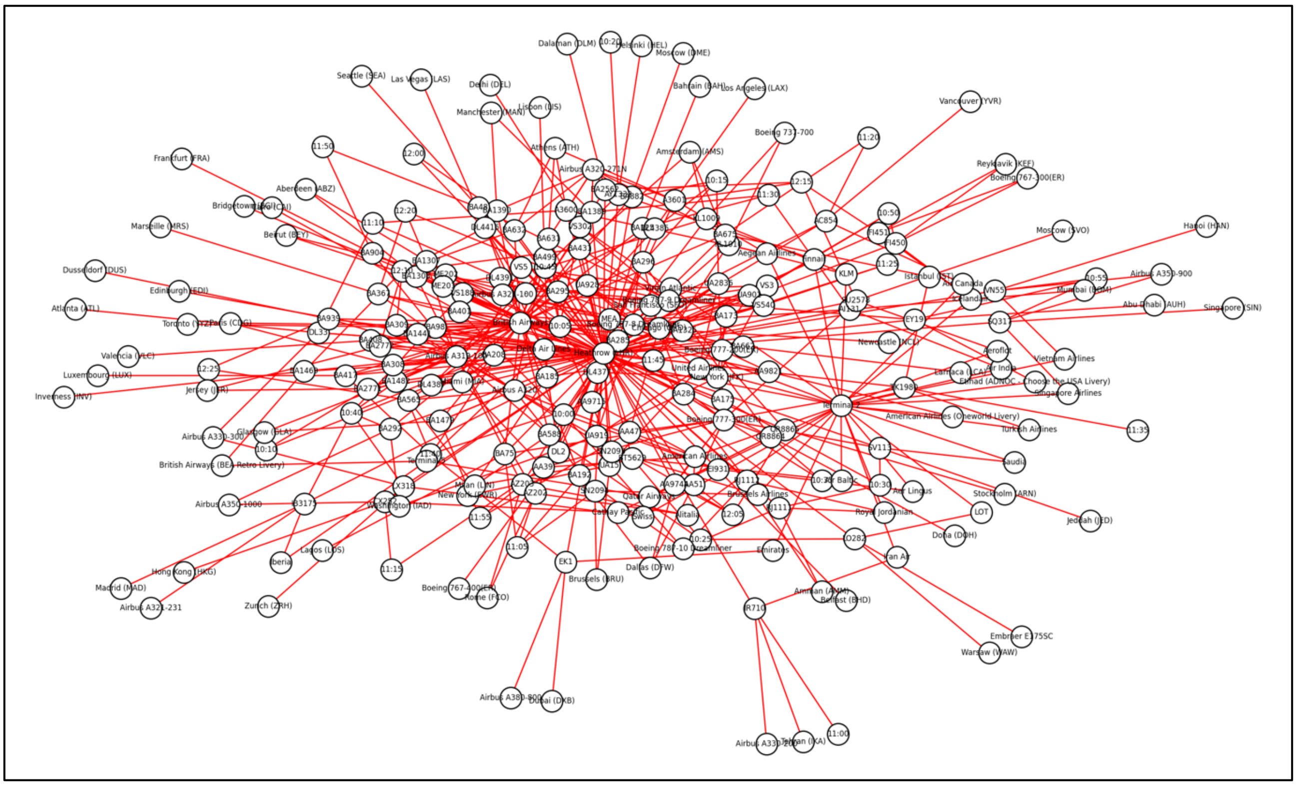


Figure 4.5 *Graph of 50 departures and 50 arrivals produced using python*

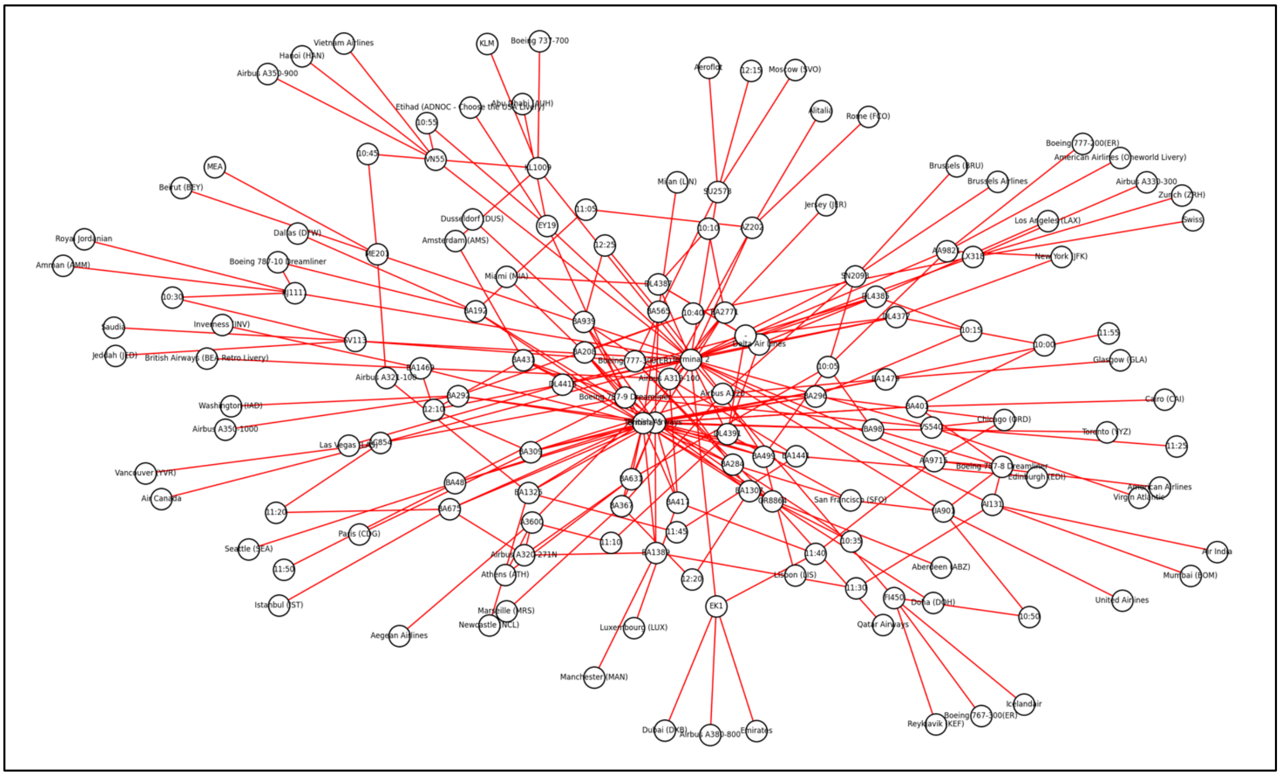


Figure 4.6 *Graph of 50 arrival flights produced using python*

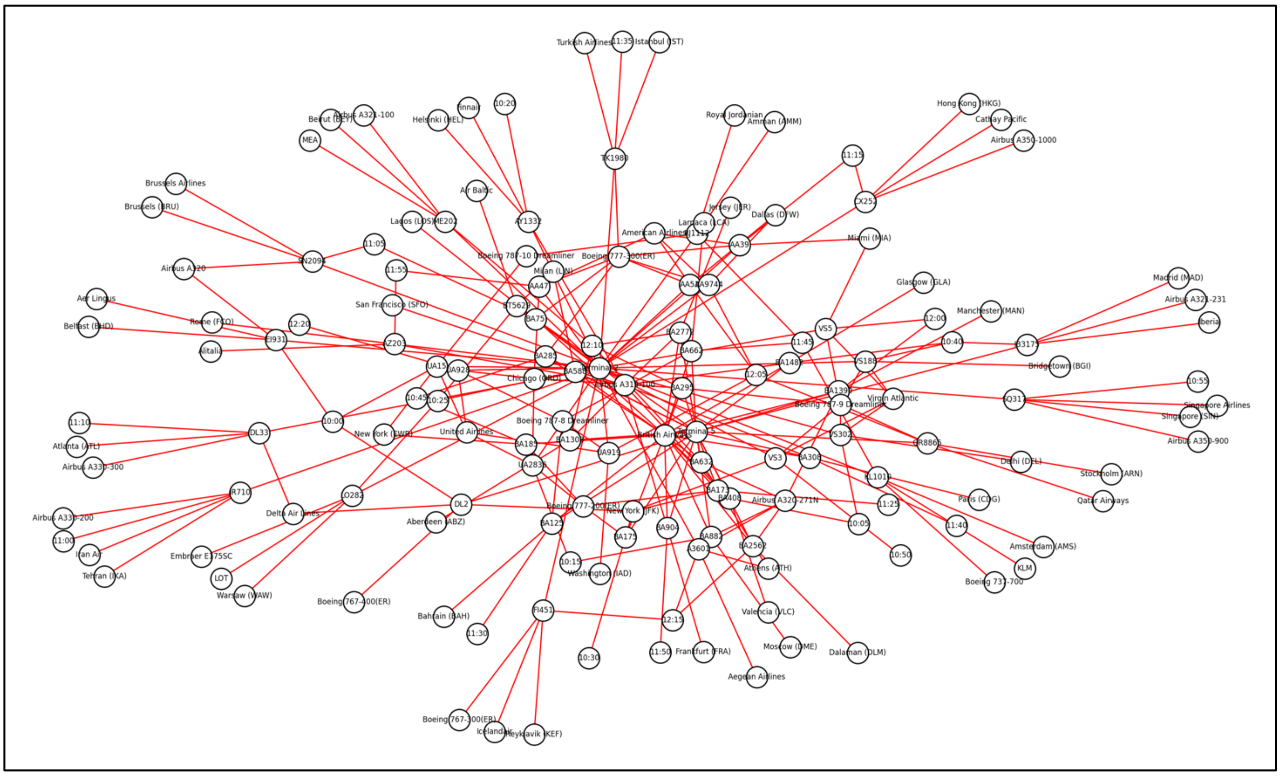


Figure 4.7 *Graph of 50 departures produced using python*

4.2 Knowledge graph analysis/Ideas for reducing CO_2 emissions

Once the knowledge graphs had been constructed, they were used to analyse the flight data in an attempt to discover a way of decreasing fuel consumption. The observations found from visually analysing the knowledge graphs will be described in this subsection and the ideas which were inspired by the observations will be explained. Although only 50 departure flights and 50 arrival flights were shown in each graph from one day, it should still show overall trends of Heathrow's flight data, as many flights are scheduled for the same time on more than one day of a week, and it would appear that the general schedule repeats itself weekly.

Observation 1: From the joint knowledge graphs, arrivals and departures, it can be seen that many airlines use the same aircraft types. In order to show this more clearly, several of the nodes have been hidden and the knowledge graph has been zoomed in to show an example. This can be seen in figure 4.8. As you can see British Airways, Virgin Atlantic, Qatar Airways, Air Canada etc are all using Boeing 787-9 Dreamliner’s and British Airways, Air Baltic and Alitalia are using Airbus A319-100’s.

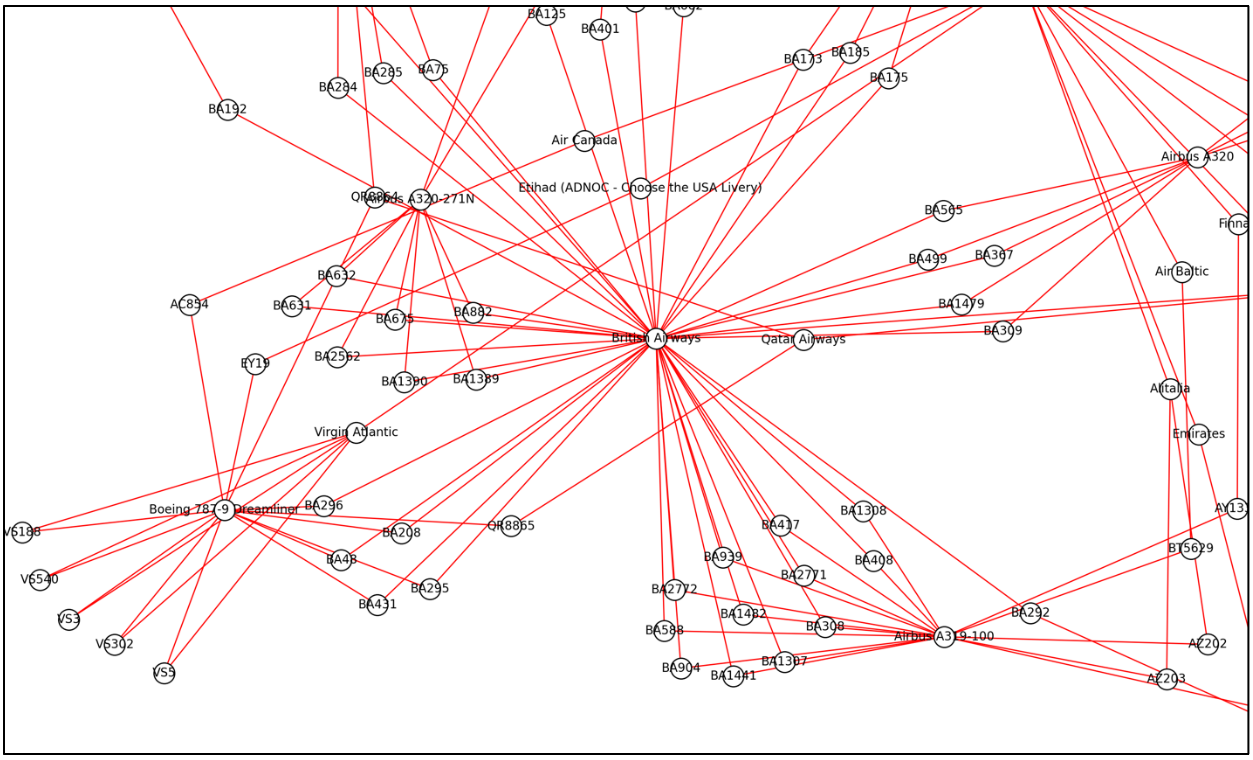


Figure 4.8 Zoomed in section of the arrivals and departures knowledge graph

Initial Idea 1: From this observation an idea did occur that instead of each having their own fleet, airlines could share aircraft. The logic behind this is that it could reduce the amount of taxiing between gates, which uses very inefficient amounts of fuel. For example, if an Airbus 320 operated by Air France was to arrive at a gate, 45 minutes before (the usual turnaround time for a short-haul flight [40]) an Airbus 320 operated by Brussels Airlines was scheduled to depart from the same gate, instead of taxiing the Air France aircraft out the way to allow the Brussels Airlines aircraft to get into the gate to start boarding passengers, the Brussels Airline could use the Airbus 320 that is already there.

An obvious complication from this however, is that all airlines have branding on their aircraft these days. This would therefore mean removing the branding which probably would not go down well with the airlines as it is less advertisement for them. It also brings in lots of other legal issues to do with insurance and therefore would require a lot of collaboration between airlines. For these reasons, this idea was not chosen to go forward with.

Observation 2: Next just the arrival flights were analysed. An observation made here was that many flights were scheduled to arrive at the same time. Again, several of the nodes were hidden in order to visualise this clearer and sections were zoomed in on to show examples. It can be seen in figure 4.9 that two flights are scheduled to arrive at Heathrow at 11:30 and three at 11:40.

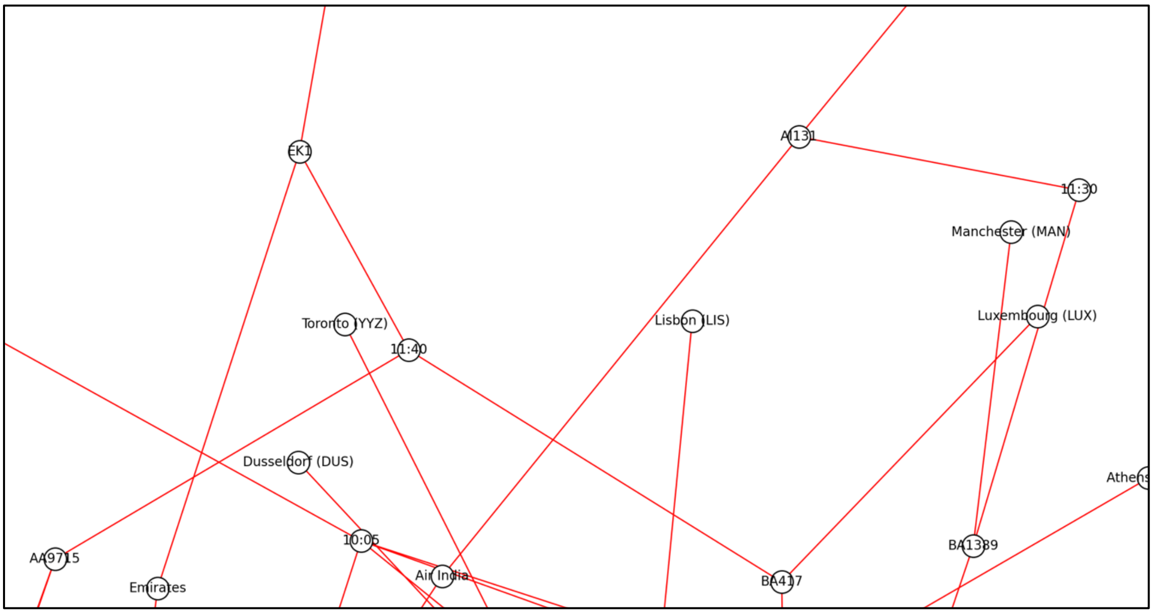


Figure 4.9 *Zoomed in section of the arrival's knowledge graph*

Initial idea 2: With many flights all scheduled to arrive at the same time, aircraft must loiter above Heathrow airport until there is a slot for them to land. This led to the questioning of which order the aircraft are allocated to land in. From a fuel reducing point of view, if two aircraft were loitering above Heathrow airport, it would be logical to let the one with the greater fuel consumption land first, therefore less fuel would be used. This was the concept behind the second idea. However, the big issue here is that it shows favouritism to airlines using less fuel-efficient aircraft. This is obviously not something that should be encouraged if trying to reduce fuel consumption and therefore was not chosen to develop further.

Observation 3: Finally, just the departure flights were analysed. An observation made here was that on several occasions' flights were leaving at the same time and going to the same destination. Again, by hiding some nodes and edges (i.e., aircraft and airline) these flights can be spotted more easily. Two examples are shown in the figures below with two flights both travelling to New York at 10:00 and two flights both travelling to Dallas at 12:05.

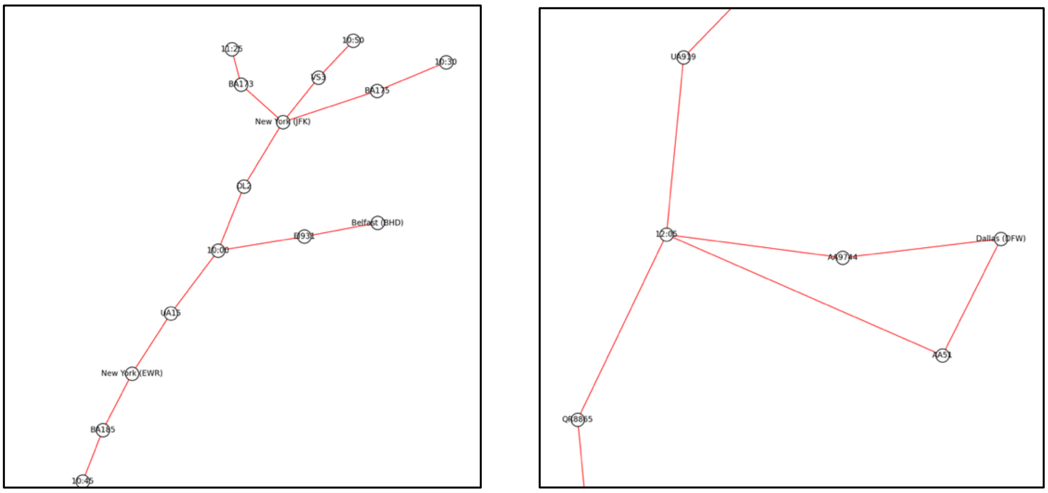


Figure 4.10 and 4.11 *Zoomed in section of the departure's knowledge graph*

Initial Idea 3: Observation 3 gave inspiration to utilise the wake-energy retrieval concept mentioned in Section 3.4 of the literature review. Aircraft scheduled to depart at the same time, heading in the same direction could fly in a formation to utilise the energy from the wake of the lead aircraft in order to reduce fuel consumption. This idea, of the three mentioned, seems to have the greatest potential, as Airbus have done flight tests and the issues mentioned toward the other ones wouldn't be an issue here. The idea will be discussed further in the next subsection and initial calculations made to see if it is worth developing further.

5. WAKE-ENERGY RETRIEVAL

5.1 Introduction to wake-energy retrieval

As mentioned in the literature review, Airbus are hoping to adopt the phenomenon observed in migrating birds such as geese, known as wake-energy retrieval. This is where the birds fly in a "V" shape to save energy. The wake coming from the leader creates a smooth current through the surrounding air, known as upwash, which allows the following bird to benefit from free lift, therefore expending less energy. Airbus are hoping that because aircraft wings also create wake, the same principle would apply to an aircraft following another aircraft, meaning the engine could reduce thrust and therefore fuel consumption [35]. The figures below are from the Airbus article, showing the wake from the leading bird and aircraft, and where the one following would be positioned to utilise the upwash.

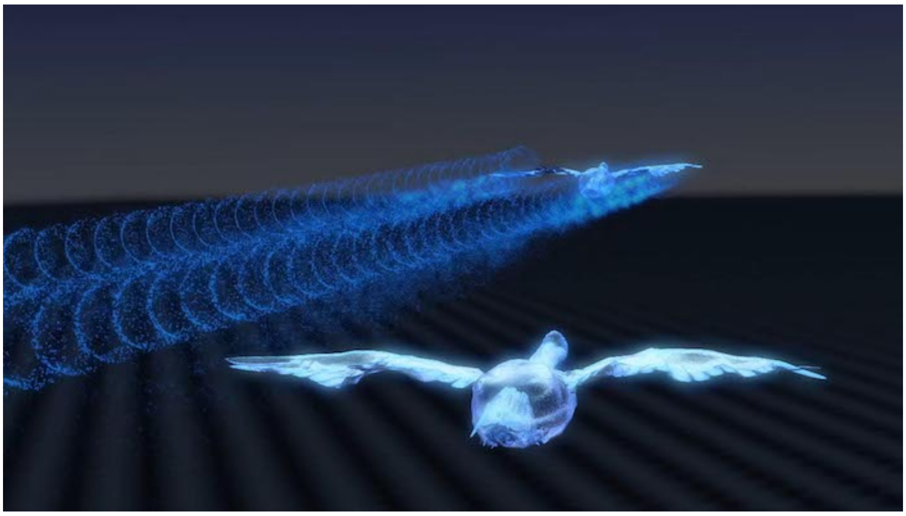


Figure 5.1 *Digital visualisation of a bird flying with its wing in a leading birds upwash* ***[35]***

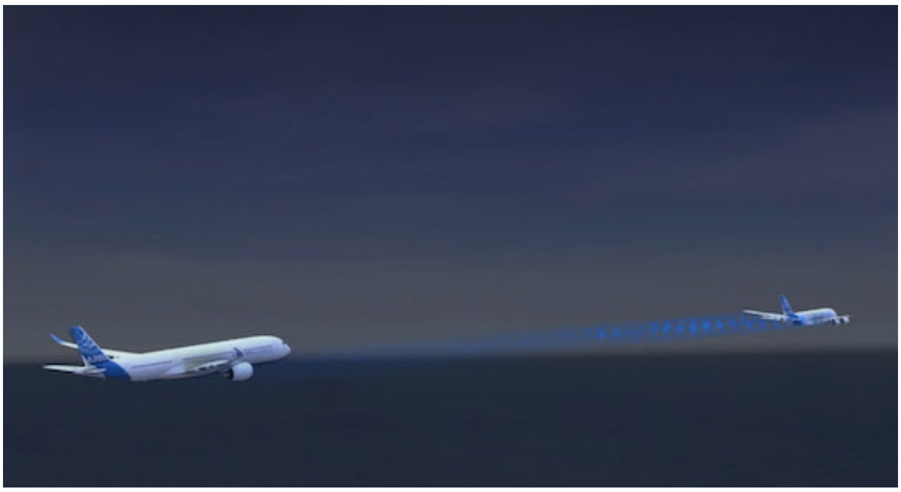


Figure 5.2 *Digital visualisation of an aircraft flying with its wing in a leading aircrafts upwash [35]*

The Airbus fello'fly project aims to bring two aircraft together from different start points, where flight plans share a common waypoint for entry to oceanic airspace and then closer to the end of their journey would separate to head to their destinations. As discussed in the previous Section, observation 3 shows several flights are heading to the same destination, at the same time from Heathrow airport. If this wake-energy retrieval technique works, these flights could be benefiting more from it as they can utilise it for longer, not just where they come together at a common waypoint. It would also require less efforts to try to rendezvous at the exact same time and place.

5.2 Initial Fuel saving calculations

Airbus has said that if this technique works it could reduce the fuel consumption of aircraft on long-haul flights by 5-10%. Using the midpoint of these values, 7.5%, some initial calculations were made to see what impact wake-energy retrieval could have on flights departing from Heathrow airport. Using the same method described in Section 4.1.1 a full day's worth of flight data was collected and formatted into an excel sheet. As the data was collected on a different day to that collected for the knowledge graphs, the data used is different.

The next step was to find all the pairs of flights leaving at the same time, going to the same destination. At this stage as it was only initial calculations, involving one day's worth of flights, this was done checking row by row on the excel sheet by eye. By doing so it was found that there were 3 pairs of flights with the same destination and departure time and 1 pair which were both going to New York, but different airports within New York. It was decided that this would have very little effect on the calculations as the route would be the same for most of the journey and therefore this pair was included in the calculations. The following equation was used to calculate the fuel saved of the following aircraft:

$$Flight\ Time\ (h) x\ Fuel\ Consumption\ of\ the\ following\ aircraft\ \left(\frac{kg}{h}\right) x\ 0.075 \quad (5.1)$$

The two unknown values in the equation above (flight time and fuel consumption) were not included in the data that has been collected thus far and therefore needed to be found in order to carry out these calculations. As only 4 pairs of flights were being dealt with at this point, these values were simply found online and manually inputted into the spreadsheet. The data and results of the calculations can be seen in table 5.1

Table 5.1 *Hand calculations of initial fuel savings*

	DepTime:	Airport:	Destination:	FC (kg/h):	FT (h):	Fuel Saved (kg):
1	10:00	New York (JFK)	New York	5100	8.00	3780
	10:00	New York (EWR)	New York	6300	8.00	
2	12:05	Dallas (DFW)	Dallas	7500	10.50	5906.25
	12:05	Dallas (DFW)	Dallas	7500	10.50	
3	13:30	New York (JFK)	New York	7500	8.00	3978
	13:30	New York (JFK)	New York	6630	8.00	
4	15:30	Los Angeles (LAX)	Los Angeles	7500	11.08	6234.375
	15:30	Los Angeles (LAX)	Los Angeles	7500	11.08	
Total Fuel Saved (kg):						**19896.625**

From these rough initial calculations, it can be seen that in one day 19, 896.63 kg of fuel could be saved by using this technique. If this were to be the average amount saved each day it can be calculated that in a year a possible 7, 262, 269.95 kg of fuel could be saved. Although these are rough calculations and other days may not save as much as this one does, it does show a significant amount of fuel can be saved. Because of this, further calculations were made with a week's worth of flights and a system of automating these calculations was made using excel and Python.

5.3 Further data collection and formatting for coded fuel saving calculations

Although the method used for the initial calculations was not too time consuming, it was only for one day's worth of flights. For anything more than this, it would be too time consuming to do them using this method, therefore a code was written to do these calculations much faster using python. This also removes the element of human error, as the pairs of flights were found by eye and some could have been missed. As with the method described in Section 5.2, some extra bits of information needed to be added to each flight in the excel sheet. These are flight time (in hours) for each flight and fuel consumption (in kg/h) for each aircraft. As the previous calculations only involved 4 pairs of flights these values were found online and manually inputted into the excel sheet, however now with a full day's worth of flights this would not be an efficient use of time.

This issue was overcome by finding a list of destinations from Heathrow airport with their flight times. This was found on another live air traffic website [41] and along with the flight times, it also included which countries each airport was in, which came to be useful later on. When the list was pasted into an excel sheet, it wasn't in the exact desired format, however it didn't take long to fix using the transpose function in excel to change the airport, city, country and flight time of each place from a vertical list to a horizontal list to form columns. The first 5 lines from the list can be seen in table 5.2.

Table 5.2 *List of airports with their associated cities, countries and flight times from Heathrow airport*

Airport:	**City:**	**Country:**	**Time:**
Istanbul IST	Istanbul	Turkey	4h 0m
Frankfurt FRA	Frankfurt	Germany	1h 40m
Paris CDG	Paris	France	1h 20m
Chicago ORD	Chicago	United States	8h 55m
Dallas DFW	Dallas	United States	10h 30m

As you can see the airports listed here include the airport codes without brackets, whereas the flight data includes the airport codes with brackets meaning if we used the VLOOKUP function to assign flight times to each flight they would not match up. This was overcome by removing the country codes and having a column of just the destination. It can also be seen in the table above that the flight time is in the form of a string not a value. Through a series of excel codes, the hour and the minutes values were able to be singled out and turned into values and then the flight time value to be used in equation 5.1 was calculated using the equation $hour + (minutes/60)$. Now the VLOOKUP function can be used to assign the flight times and countries to the list of flights. Again, all these excel codes can be found in Appendix 2.

Next the same thing needs to be done for the aircraft fuel consumption values. In Section 4.1.1 a list of aircraft types and their codes had already been formed. Now their fuel consumption must be added to this list in another column. The majority of these values were found on one website [42] and therefore were easy to copy and paste into excel, while some others had to be individually found online and manually added. Once again when this list was complete, the VLOOKUP function was used to assign all the fuel consumption values to each of the flights based on what aircraft type was being flown. Now all the relevant information was in one excel sheet a code could be written to follow out the same calculations as the initial ones shown in the Section above.

For this bit of coding the Python package Pandas was used to read the excel file instead of xlrd, as it was found to be easier to use to select certain rows and values, which was more necessary for these calculations. The code was used to iterate down each row and find matches if the departure time and destination were the same. If this was the case it would then select the fuel consumption and flight time of the second flight of the pair and use these values and equation 4.1 to calculate the fuel saved. The destination and fuel saved of each pair would then be added to a list and the list would be printed, along with the sum of the fuel saved. The code for this can be found in Appendix 4.1. The code was run using the flight data from the previous Sections initial calculations to see if the final values matched. The following line of code was returned showing that the values do match.

[('New York', 3780.0), ('Dallas', 5906.25), ('New York', 3978.0), ('Los Angeles', 6234.375)]

19896.625

With the excel functions and the python code it should then be a fast process to repeat these calculations on more days of flight data to find a more accurate value for how much fuel could be saved in a week and therefore on average in a year. This will be covered in the next subsection.

5.4 Finding flight pairs

Over the course of a week flight data from each day was copied into individual sheets of an excel spreadsheet and formatted using the methods described in Sections 4.1.1 and 5.3 The code used in the previous Section was then run using Monday's flight data to show how many pairs of flights there were leaving at the same time, going to the same destination and then returning the value of fuel saved if these flights adopted the wake-energy retrieval technique. The code returned 0 flight pairs and therefore 0 fuel saved. At this point it became apparent that the day that had been previously tested may have been an anomaly and not as many flights did leave at the same time, going to the same destination as was previously predicted. A simplified piece of code was therefore used to find all the flight pairs from each day and can be found in Appendix 4.2.

Only 5 pairs of flights were found over the whole week. These results were very disappointing as the flight data used in the initial calculations had 4 pairs of flights in just one day. It was also found that one of the flight pairs was a short haul flight. As in the Airbus article [35] they say 10% could be saved on long haul flights, only flights greater than 6 hours were of interest [43]. This therefore led to the change of code, to see how many flights were leaving at the same time and going to the same country, not just the same airport or city, and who's flight time was greater than or equal to 6 hours. This would hopefully show more pairs, but the wake-energy retrieval technique could still be adopted for most of the journey. This code can be found in Appendix 4.3

This time 25 pairs of flights were returned, showing there are many more flights going to the same country than there are just to the same airport or city, which is promising for the fuel saving calculations. However, when looking at the countries that had been paired, the majority of them were going to the USA. Due to the size of the USA, it did give rise to a concern that the individual destinations could be on different sides of the country and therefore would take different flights paths across the Atlantic, meaning the fuel calculations would be inaccurate, as the aircraft would have to take a detour and may use more fuel than is saved. This therefore gave inspiration to use the distance between final destinations instead of countries or airports/cities to pair the flights. This would also mean destinations in different countries, but still very near to each other, could be paired together.

A package on python called Geopy can be used to calculate the distance between two locations if the latitude and longitude are known. In order to use this to find countries within a certain distance of each other, the latitude and longitude of each destination must be found and assigned to each flight. An excel spreadsheet of a World Cities Database was found online [44] and downloaded with a list of all world cities and their corresponding latitude and longitude. The VLOOKUP function could have been used here again to go through each city assigning the latitude and longitude, however due to the size of the file, it ran the risk of crashing. Therefore, the latitude and longitude were manually copied over to the list previously made of all Heathrow destinations. Then from that list the VLOOKUP function was used to assign the latitude and longitude to each flight across the weeks' worth of data.

The code was then altered, to now iterate through the rows of flights and calculate the distance between the destinations using the latitude and longitude. It would then add to a list a tuple of the 2 destinations and the distance between them, if the departure time was the same, the distance between them was less than 1000 miles and the flight time was greater than or equal to 6 hours. 1000 miles was chosen as the distance as it would allow destinations on the east coast of the US to be paired together and destinations on the west coast to be paired but it would prevent two destinations being paired that were on different coasts. As mentioned earlier they take different flight paths across the Atlantic. This bit of code can be found Appendix 4.4

The number of flight pairs that were found using this technique was 22, which is slightly less pairs than there were when matching countries, as the pairs from the USA which were over 1000 miles apart have been removed. One pair had been added which are within 1000 miles of each other but in different countries. This therefore would appear to be a more accurate way of finding pairs than just matching them on countries but gives many more results than just matching cities. The pairs found using this final method were then used to calculate the fuel which could have been saved that week if the wake-energy retrieval technique works and was adopted.

5.5 Final calculations and results

Again, by rewriting the code slightly, as shown in Appendix 4.5, it was possible to return the flight pairs, along with the necessary information for the fuel saving calculations. These were then inputted into a table on excel. The Airbus article does not specify whether the fuel saving was 5-10% of the total fuel consumed in a journey or 5-10% of the fuel consumed while in the formation for wake-energy retrieval, therefore calculations were done to make best estimates for both scenarios. The value acquired for fuel consumption is assumed to be during cruise, meaning the fuel consumption for the whole journey had to be calculated. As roughly 85% of fuel is consumed during cruise, the fuel consumed with take-off and landing was calculated using the equation below:

5.2 $$Fuel\ Consumed\ over\ whole\ journey\ = \frac{Cruise\ Time\ (h)\ x\ Fuel\ Consumption\ (kg/h)}{0.85}$$

Using this equation, the fuel consumed for each individual flight was calculated. Then taking the smaller value from each pair, to account for the 'worst-case scenario', 7.5% of the fuel consumption was calculated and summed up to find total fuel saved over the week. This came to 68, 351.14 kg. If this was taken as an average weeks' worth of savings, over the course of a whole year a possible 3, 554, 259.34 kg of fuel could be saved.

Next, the second scenario, that Airbus were referring to 5-10% of fuel saved while the aircraft were in formation, was considered. This would therefore require further calculations and thinking. On average it is said to take roughly 10 minutes for take-off and 30 minutes for descent and landing, therefore, to calculate cruise time 40 minutes was taken off the total time of the flights. For flights going to the same destination the fuel saved was calculated with equation 5.3, again using the fuel consumption of the aircraft with the lower value.

5.3 $$Fuel\ Saved\ = Cruise\ Time\ x\ Fuel\ Consumption\ x\ 0.075$$

For flights which were not going to the same destination, other factors needed to be considered. The destinations of each of the individual flights in each pair were put into maps to evaluate how to make the most accurate calculations. Due to the great-circle route [] flights traveling to the east coast of the USA would be taking roughly the same route across the Atlantic and then splitting to reach their own destination. In some cases, it would appear that some flights would pass over the destination of the other flight in its pair. In this situation it was assumed that the flights would stay in formation and utilise the wake-energy retrieval technique up until the first destination was reached. This scenario can be seen in the diagram below and has been named the 'Drop-off' scenario. The calculations for these scenarios were the same as in equation 5.3 using the cruise time of the first destination i.e., Halifax in the diagram below.

Figure 5.3. *Diagram showing 'Drop-off' scenario with fuel calculations*

For flights which were not on the exact same flight path, there appeared to be two options. The first option would be for the aircraft to stay in formation all the way to one of the airports and one of the aircraft would divert to reach their destination. Obviously, this would add on extra flight time for the second aircraft, meaning more fuel was used. So, this must be compared with the fuel saved to see if it is worth it. The second option is to only stay in formation across the Atlantic and then split to go their separate directions. As this is the path most flights take anyway, it would not require any divergence and therefore wouldn't add any extra fuel, however the time utilising the wake-energy retrieval is decreased. These two options are shown below with calculations to see which would be more effective.

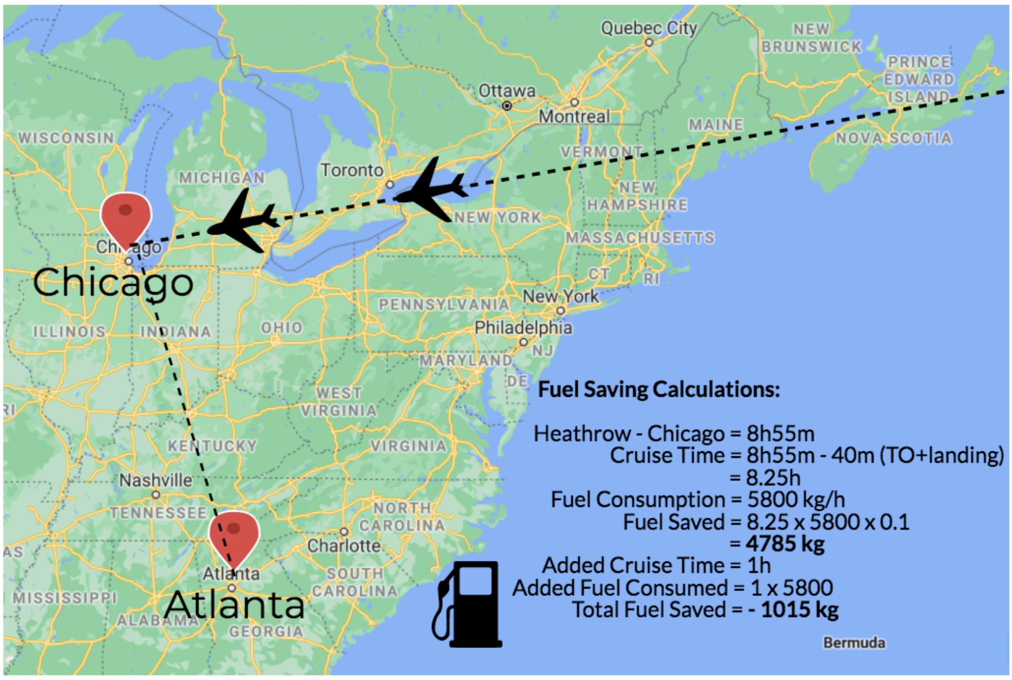


Figure 5.4. *Diagram showing 'Diversion' scenario with fuel calculations*

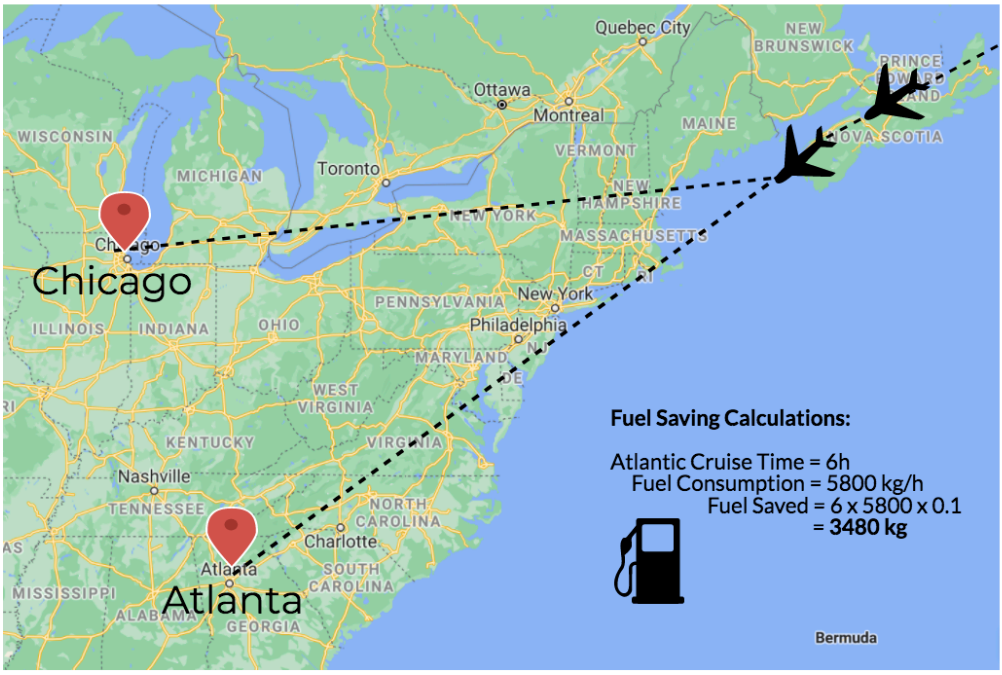


Figure 5.5. *Diagram showing 'Split' scenario with fuel calculations*

It can be seen that the 'Diversion' scenario would actually use more fuel for the added flight time than is saved. Therefore, the 'Split' scenario is used to calculate the other flights which do not follow the same flight path. It was estimated that the cruise flight time from Heathrow across the Atlantic was 6 hours, as this was the cruise time to Halifax which appeared to be on the general flight path across to the east coast of the US, therefore this value was used for the calculations. The table below shows all the pairs of flights and the fuel savings. They have been colour coded to show which ones used which calculations. As you can see this table shows only 19 flight pairs, when the code in the previous Section returned 22. This was because the code doesn't take into account if the flight before has already been paired. Therefore, there were some overlaps. Once these were identified, the scenario which saved more fuel was used. For example, 10:05 New York and 10:05 Chicago were paired up and the same 10:05 Chicago was paired with a different 10:05 New York. In this scenario it would save more fuel if both New York flights were used as they can utilise the wake-energy retrieval technique for longer.

***Table 5.3** Flight pairings from one week's worth of flight data with fuel saving calculations*

Day:	Pair:	Destination:	FT (h):	Aircraft:	FC (kg/h):	Fuel Consumed (kg):	Fuel Saved – a (kg):	Time in Formation (h):	Fuel Saved – b (kg):
Mon	1	New York	8.00	B77W	7500	64676.47	2963.06	6.00	2520.00
		Halifax	6.67	B789	5600	39507.45			
	2	Chicago	8.92	A35K	5800	56271.37	4220.35	6.00	2610.00
		Atlanta	9.75	A35K	5800	61957.65			
Tues	3	Chicago	8.92	B789	5600	54330.98	4074.82	8.25	3463.60
		Chicago	8.92	B772	6630	64324.00			
	4	Chicago	8.92	A35K	5800	56271.37	4220.35	6.00	2610.00
		Atlanta	9.75	A35K	5800	61957.65			
Wed	5	Islamabad	7.75	B772	6630	55224.00	3560.82	7.08	3026.70
		Islamabad	7.75	A333	5700	47477.65			
	6	Atlanta	9.75	A35K	5800	61957.65	4220.35	6.00	2610.00
		Chicago	8.92	A35K	5800	56271.37			
Thurs	7	New York	8.00	B77W	7500	64676.47	4486.59	7.33	3078.60
		Atlanta	9.75	B789	5600	59821.18			
	9	Atlanta	9.75	A35K	5800	61957.65	3896.12	7.75	3311.70
		Washington	8.42	A333	5700	51948.24			
	10	Chicago	8.92	B789	5600	54330.98	3751.24	6.00	2520.00
		New York	8.00	A35K	5800	50016.47			
Fri	11	Miami	10.00	B772	6630	72774.00	3376.64	7.75	2870.14
		Washington	8.42	B763	4940	45021.80			
	12	Chicago	8.92	A35K	5800	56271.37	4220.35	6.00	2610.00
		Atlanta	9.75	A35K	5800	61957.65			
	13	New York	8.00	B77L	6800	58640.00	3751.24	7.33	3188.55
		New York	8.00	A35K	5800	50016.47			
Sat	14	Miami	10.00	B77W	7500	82323.53	3376.64	7.75	2870.14
		Washington	8.42	B763	4940	45021.80			
	15	Chicago	8.92	A35K	5800	56271.37	4220.35	6.00	2610.00
		Atlanta	9.75	A35K	5800	61957.65			
	16	New York	8.00	B772	6630	57174.00	3751.24	7.33	3188.55
		New York	8.00	A35K	5800	50016.47			
Sun	17	Miami	10.00	B77W	7500	82323.53	3376.64	7.75	2870.14
		Washington	8.42	B763	4940	45021.80			
	18	Philadelphia	7.92	B788	4900	41774.90	3133.12	7.25	2663.15
		Atlanta	9.75	A35K	5800	61957.65			
	19	New York	8.00	B77L	6800	58640.00	3751.24	7.33	3188.55
		New York	8.00	A35K	5800	50016.47			
						Total Saved – a (kg):	68351.14	Total Saved – b (kg):	51809.82

	Drop off
	Split
	Same Destination

Due to the travel restrictions resulting from the ongoing global pandemic, flights coming in and out of Heathrow airport, and all airports around the world, have decreased massively. From the week the data used above was collected, the daily average number of departures flights was calculated to be 154. From Heathrow's traffic statistics [46], in 2018 there was a daily number of 1,300 flights. Assuming half of these were departures, roughly an average of 650 flights were leaving Heathrow airport pre-covid. Using these figures, the fuel savings were scaled accordingly to estimate how much fuel potentially could have been saved pre-covid and therefore could be saved in the future when travel restrictions are fully lifted. The weekly savings were then multiplied by 52 to estimate a yearly average for each scenario. These results can be seen in table 5.4 and the yearly average savings visualised graphically in figure 5.6.

Table 5.4 *Estimated fuel savings for four possible scenarios at Heathrow airport*

Scenario	Departures Per Day (Average)	Estimated Fuel Saved Per Week (Average) (kg)	Estimated Fuel Saved Per Year (Average) (kg)
Covid - a During covid, assuming 10% fuel saved of overall fuel consumption	**154**	**68,351.14**	**3,554,259.34**
Covid - b During covid, assuming 10% fuel saved of fuel consumed while in formation	**154**	**51,809.82**	**2,694,110.64**
Pre-Covid - a Pre-covid, assuming 10% fuel saved of overall fuel consumption	**650**	**287,428.66**	**14,946,284.66**
Pre-Covid - b Pre-covid, assuming 10% fuel saved of fuel consumed while in formation	**650**	**217,869.39**	**11,329,208.33**

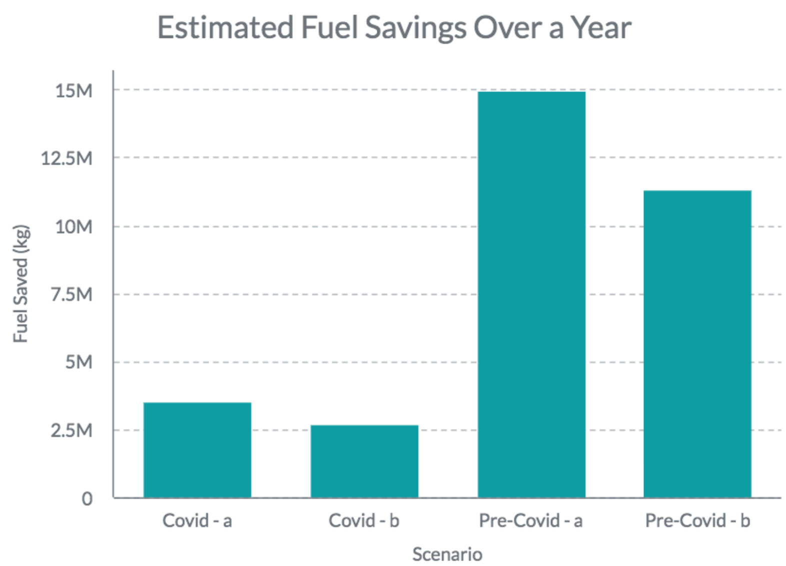


Figure 5.6 *Estimated fuel savings over a year for four possible scenarios at Heathrow airport*

The values for the yearly savings were then converted into money and CO_2 savings. As fuel prices do vary, an initial idea was to use the current fuel price for the covid scenarios and the fuel price in 2018 for the pre-covid scenarios, as that was the year the daily departures average was taken from. However, it was decided that a better comparison could be made between scenarios if fuel price remained constant. As of January 2021, the price of jet fuel was said to be £0.33 per kg [47]. This value was used to calculate money saved by multiplying it by the yearly savings values. The same process was used for calculating the amount of CO_2 that will not have been emitted due to the fuel not being burnt. It has been calculated that 3.15kg of CO_2 is emitted from 1kg of jet fuel [48]. The total savings of fuel, money and CO_2 for all four scenarios can be seen in table 5.5.

Table 5.5. *Estimated savings of fuel, money and CO^2 emissions for four possible scenarios at Heathrow airport*

	Covid - a Savings per year	Covid - b Savings per year	Pre-Covid - a Savings per year	Pre-Covid - b Savings per year
	3, 554, 259.34 kg	2, 694, 110.64 kg	14, 946, 284.66 kg	11, 329, 208.33 kg
	£1, 172, 905.60	£889, 056.51	£4, 932, 273.90	£3, 738, 638.70
CO2	11, 195, 916.92 kg	8, 486, 448.52 kg	47, 080, 796.25 kg	35, 687, 006.24 kg

6. DISCUSSION

The final results and outcomes from this project can be found in Sections 4 and 5. These include the knowledge graphs consisting of 50 departure flights and 50 arrival flights, and estimated fuel savings which are shown in tables 5.3, 5.4, 5.5 and figure 5.6. This Section will discuss these results and how much these outcomes support the original objectives laid out for the project. As specified in Section 1 of this report, an original objective was to explore ideas in which knowledge graphs and digital twins can be used individually or together to improve sustainability in engineering applications and choose one idea to develop. After reviewing some literature on these subjects, it was decided that a knowledge graph would be constructed and used to analyse the air traffic coming in and out of Heathrow airport in an attempt to discover a way to decrease fuel consumption.

As shown in Section 4.1.2, figure 4.5, a knowledge graph was successfully constructed with 50 departure flights and 50 arrival flights. In order to determine the difference between the departures and arrivals it had been intended to change the colours of the nodes, however after failing several times to find a method do so it was concluded that it was not an efficient use of time and having separate graphs for departures and arrivals was a sufficient alternative, as the data was still able to be analysed by looking at the two graphs alongside each other and individually. If more time was provided it would be beneficial to find a method of changing the node colours and also adding labels to the edges to show the relations between the nodes, however the knowledge graphs were still able to be analysed and from the analysis several ideas were proposed and a final one was further explored. Therefore, the objective mentioned above was achieved.

The developed idea, as mentioned previously, was for aircraft scheduled to depart from Heathrow at the same time, heading in the same direction to adopt the wake-energy retrieval technique, proposed by Airbus, in order to reduce fuel consumption. An objective was to find data and carry out calculations to support this developed idea. The necessary data was successfully found online and formatted in excel in a way that it could be used more easily. Calculations were done, as described in Section 5 and the final results can be seen in figure 5.6 and tables 5.4 and 5.5. The results show a significant savings in fuel, money and CO_2, showing this technique would improve the environmental sustainability of Heathrow airport. However, these results do have a large margin of error. Firstly, the flight data was only taken for one week. Although flights do tend to repeat themselves weekly, some days have many more pairs than others, as shown in the initial fuel saving calculations. Also, the time of the year was not taken into consideration. It could be assumed that there will be more flights leaving and arriving Heathrow in the summer months, as people go on summer holidays. If given more time, further calculations could be carried out to account for this.

The results shown have all been calculated under the assumption that the pair of flights will be leaving one after another. This is so the aircraft will already be close to each other and once they are at cruise altitude, they will be able to manoeuvre into formation for wake-energy retrieval. If they do not leave one after another, it will require more manoeuvring for either the following plane to catch up or the lead plane to slow down, which may result in extra fuel consumption and/or break safety regulations. This therefore brings up the issue of delays. If one flight of the pair were delayed, what would happen? A flight is considered on time if it is no more than 15 minutes late departing. Therefore, it would be recommended that within this window flights hold off and depart together even if one is ready to depart before the other, as they will not be considered a delay and they can still save fuel using the wake-energy retrieval technique. If one of the pairs were to be delayed longer than 15 minutes while the other was ready to leave, further discussions would need to be had. An article in 2020 [49] estimated that each minute of a delay costs the airline $74.20 USD. Using these figures rough calculations could be made to determine at which point the delays would cost more than the fuel saved. However, it is not only the money that needs to be considered but the passenger's satisfaction, and the airlines' reputations. Unfortunately, due to time constraints, these calculations and considerations were not able to be explored further.

As shown in table 5.4, due to travel restrictions daily flights are less than a quarter of what they used to be, with only 2 terminals in operation. Therefore, estimated savings were calculated by scaling the values calculated from the current daily flight figures. While analysing the data it could be seen that there were very limited flights travelling to Asia. Although it is hard to be sure because no old flight data was found, it could be predicted that many of the flights travelling to East Asia could adopt the wake-energy retrieval technique, as many of the destinations are close together and the flight times are generally greater than 10 hours. This means even if they would have to 'Split' off to reach their individual destinations they could still stay in formation for the majority of the journey, therefore utilising the free lift from the leading aircraft for a long time.

In order for aircraft to fly close enough to retrieve the energy lost by the wake, the required air traffic management technology is necessary. Airbus is working on a technical solution that involves developing pilot assistance functions in order to safely fly in the correct position for the following aircraft to remain in the updraft of air of the leading aircraft. These developments will cost money and although not specified in their articles, it can be assumed they will not be cheap. However, as shown in table 5.5 the fuel saved by using this technique will result in saving money from the cost of fuel. This is therefore an incentive for airlines to invest and adopt the technique. This does, however, lead to the issue of sharing fuel savings. As only the following aircraft utilise the wake energy, that aircraft will save on fuel and money, even though the leading aircraft had to put in just as much work. A sharing scheme would have to be put in place so each aircraft benefits equally.

As one of the original aims involved both knowledge graphs and digital twins, it was intended to explore the idea of having a digital twin of an airport and possibly combining this with the knowledge graph. Due to time constraints, it was decided that it would be better to explore one idea in full and calculate results, than to look into these two ideas and possibly run out of time before getting any results. It also appeared that the idea of the digital twin would be much more conceptual involving more literature reviews as opposed to being able to quantify results. As one of the aims was to find data and carry out calculations, this aim may not have been achieved if the digital twin path was chosen. This idea will however be discussed further in the final Section about future work.

7. CONCLUSIONS AND FUTURE WORK

Having determined the aim of this project involved improving environmental sustainability in a chosen engineering application, literature around this subject was reviewed and it was concluded that the focus would be reducing CO_2 emissions in the aviation sector. From reviewing literature on both knowledge graphs and digital twins it was decided that a knowledge graph of flights coming in and out of Heathrow airport would be constructed in order to analyse the data to find a way to reduce fuel consumption and therefore CO_2 emissions. It had also been planned to find a way to use a digital twin to improve the environmental sustainability of the airport and potentially use the knowledge graph and digital twin together. Unfortunately, due to time constraints the digital twin concept did not take shape, and it was decided it would be better to continue with the fuel savings idea in order to get quantifiable results which could determine whether it would reduce CO_2 emissions. Knowledge graphs of the flight data were successfully constructed and analysed, and from these an idea was developed, and calculations were made to determine the fuel savings, meaning many of the original objectives of the project were successfully carried out.

Through the analysis of the flight data and the fuel saving calculations the following conclusions were made:

- As many flights get scheduled to leave Heathrow at the same time, heading in the same direction, Airbus' proposed wake-energy retrieval strategy could be adopted by these flights. Calculations from one week's worth of flights, while COVID travel restrictions were in place, estimated that between 51,809.82 and 68, 351,14 kg of fuel could have been saved.
- The daily average number of flights from the time the flight data was collected was calculated to be less than four times what it was in 2018, before COVID travel restrictions. It was concluded that without travel restrictions the number of flight pairs and therefore fuel saved through wake-energy retrieval would be much greater. Estimated fuel savings were between 217, 869.39 – 287, 428.66 kg.
- It was observed that there was a limited number of flights to Asia, another result of the travel restrictions and was concluded that pre-covid, many flights to South

East Asia could adopt the wake-energy retrieval strategy, as lots of the destinations are close together and the flight times are generally 10hours + meaning the aircraft can be in formation for majority of the journey. This would therefore result in further fuel savings.

- Finally, it was concluded that although the results calculated in the report do have a large margin of error from estimated fuel consumption values, lack of consideration for delays and only using one week's worth of data, the potential savings of fuel and therefore reduction in CO_2 emissions are significant if this wake-energy retrieval strategy is successful. This is on top of the savings from the routes Airbus have already planned, which involve flights meeting at a common waypoint.

As mentioned throughout the report, due to time constraints, several of the objectives were not achieved. However, if future work were to come from this project, some of the unachieved objectives would be explored, as well as new developments which have been prompted from the work already done. One of the objectives to be explored would be the concept of the digital twin of Heathrow airport. Having sensors all around the airport collecting data would mean a digital representation of the airport could be visualised, showing real time events. This digital twin could also be used to control various things throughout the airport, as well as run simulations of different scenarios in order to prepare for future events and/or find optimum solutions for problems that might occur. For example, the digital twin could be used to measure and control the temperature of the airport. It could determine, depending on how many and where people are situated around the airport, how best to regulate the temperature in an energy efficient way. It could potentially run scenarios of different taxi routes for aircraft to find the most fuel-efficient journey. The data collected from the sensors could be stored in a knowledge graph to then be analysed for improvements. Much more research would need to be done to explore this idea further.

As mentioned in Section 6, delays were not originally considered in the code or calculations and unfortunately due to time constraints further work was not able to be done in order to account for these. If future work were to be done on this project the previously mentioned calculations could be carried out to determine if the cost of one aircraft waiting for a delayed aircraft would be more or less than that of the fuel saved using wake-energy retrieval. Improvements could also be carried out to the code. Currently the code only pairs up flights scheduled to leave at the exact same time, however flights rarely leave at this scheduled time as they are given a 15-minute window in which to depart, before being considered delayed. By adjusting the code, it could be used to find flights heading to destinations within 1000 miles of each other, who's departure times are within 15 minutes of each other. This would increase the probability of finding flights pairs in a day, but also allow the flights to leave within the 15-minute window and not count as a delayed flight. Using the same code but changing the time window to 30 minutes, further pairs might also be found. These would be useful to know if a flight was to get delayed. For example, if a flight had been paired with a flight within their 15-minute window, however it was now delayed more than 15 minutes, it would already know if any flights in the next 15 minutes window were a match, and therefore instead of the first flight delaying on purpose this flight could pair with a later flight.

The work carried out in this project has prompted several ideas new for future developments. Two of which were briefly discussed in Section 4.2, however, decided against. The idea of allowing less fuel-efficient aircraft to land first, therefore consuming less fuel, was rejected as it unfairly rewards these aircraft, when more fuel-efficient aircraft should be benefitting. This did however give rise to the thought that more fuel-efficient aircraft should be rewarded therefore encouraging more airlines to use them. From the Heathrow Carbon Footprint Report [30] that was reviewed in Section 3.4, it does mention that 'Airlines can benefit from landing fees up to 11 times cheaper by utilising cleaner and quieter aircraft', showing that these sorts of benefits schemes are already in place, however further ones could be applied. The other idea mentioned in Section 4.2 was for airlines to share aircraft in order to reduce fuel consumed taxiing between gates. This was rejected as it would require a lot of cooperation with the airlines, including insurance and branding issues. However, if the wake-energy retrieval method could be applied, it could prove that airlines are willing to work together and this method could one day be adopted.

Another idea for future developments is taking inspiration from Airbus' wake-energy retrieval technique to save fuel by flying in formation. Although the physics is slightly different, a similar concept is used by cyclists and cars, known as drafting or slipstreaming. This is where one moving object follows another closely behind to utilise the low-pressure zone generated by the downforce of the leading moving object. This means the following object works less hard to overcome the air resistance [50]. This concept combined with the code written to pair up the flights could be used for lorries leaving depots or ferries. Lorries taking similar routes across the country could be paired up, or even grouped together, and could travel in each other's slipstreams. Robert Mathews from Science Focus [51] reported that 'According to research published last year by the University of Aachen, Germany, lorries can save around 17 per cent in fuel by tailgating each other.' Similar calculations to the ones done in this project could be carried out to estimate fuel savings. However, safety tests would need to be carried out, as the distance required to utilise the slipstream may not be a safe stopping distance, and therefore would be too dangerous.

8. REFERENCES

[1] Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A. and Taylor, J., 2019. Industry-scale knowledge graphs: lessons and challenges. *Queue*, *17*(2), pp.48-75.

[2] Banerjee, A., Dalal, R., Mittal, S. and Joshi, K.P., 2017. Generating digital twin models using knowledge graphs for industrial production lines. *UMBC Information Systems Department*.

[3] Marr, B., 2020. *What Is Digital Twin Technology - And Why Is It So Important?*. [online] Forbes. Available at: <https://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twin-technology-and-why-is-it-so-important/?sh=3fcdb012e2a7> [Accessed 15 November 2020].

[4] Arora, N.K., 2018. Environmental Sustainability—necessary for survival.

[5] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A. and Navigli, R., 2020. Knowledge graphs. *arXiv preprint arXiv:2003.02320*.

[6] Ontotext. 2021. *What is a Knowledge Graph? | Ontotext Fundamentals*. [online] Available at: <https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/> [Accessed 4 February 2021].

[7] The Alan Turing Institute. 2021. *Knowledge graphs*. [online] Available at: <https://www.turing.ac.uk/research/interest-groups/knowledge-graphs#:~:text=Some%20knowledge%20graphs%20are%20used,Wikidata%2C%20WordNet%2C%20Geonames%20etc> [Accessed 6 February 2021].

[8] Pecánek, M., 2020. *Google's Knowledge Graph Explained: How It Influences SEO*. [online] Ahrefs.com. Available at: <https://ahrefs.com/blog/google-knowledge-graph/> [Accessed 6 February 2021].

[9] Kamakoti, B., 2020. *Introduction to Knowledge Graph Embedding with DGL-KE*. [online] Medium. Available at: <https://towardsdatascience.com/introduction-to-knowledge-graph-embedding-with-dgl-ke-77ace6fb60ef#:~:text=Knowledge%20graph%20embeddings%20(KGEs)%20are,be%20used%20to%20infer%20relations.> [Accessed 6 February 2020].

[10] Miskinis, C., 2019. *The Mysterious History of Digital Twin Technology and Who Created It*. [online] Challenge Advisory. Available at: <https://www.challenge.org/insights/digital-twin-history/> [Accessed 7 February 2021].

[11] Ferguson, S., 2020. *Apollo 13: The First Digital Twin | Simcenter*. [online] Simcenter. Available at: <https://blogs.sw.siemens.com/simcenter/apollo-13-the-first-digital-twin/> [Accessed 10 February 2021].

[12] Twi-global.com. 2021. *What is Digital Twin Technology and How Does it Work?*. [online] Available at: <https://www.twi-global.com/technical-knowledge/faqs/what-is-digital-twin> [Accessed 7 February 2021].

[12] Department for Business, Energy and Industrial Strategy, 2019. 2019 UK greenhouse gas emissions, final figures.

[14] Ritchie, H. and Roser, M., 2021. *Emissions by sector*. [online] Our World in Data. Available at: <https://ourworldindata.org/emissions-by-sector> [Accessed 8 November 2020].

[15] A. Schafer, J.B. Heywood, H.D. Jacoby, I.A. Waitz, 2009. Transportation in a Climate-Constrained World. *MIT Press, Massachusetts.*

[16] Terrenoire, E., Hauglustaine, D.A., Gasser, T. and Penanhoat, O., 2019. The contribution of carbon dioxide emissions from the aviation sector to future climate change. *Environmental research letters*, *14*(8), p.084019.

[17] Union, S., 2020. *Create Your Digital Twin With An Enterprise Knowledge Graph | Stardog*. [online] Stardog Union. Available at: <https://www.stardog.com/blog/create-your-digital-twin-with-an-enterprise-knowledge-graph/> [Accessed 25 November 2020].

[18] Kaewunruen, S., Rungskunroch, P. and Welsh, J., 2019. A digital-twin evaluation of net zero energy building for existing buildings. *Sustainability*, *11*(1), p.159.

[19] Twi-global.com. 2020. *Digital Twin Technology To Monitor Offshore Assets*. [online] Available at: <https://www.twi-global.com/media-and-events/press-releases/2017-03-twi-embarks-on-lifecycle-engineering-asset-management-through-digital-twin-technology> [Accessed 17 November 2020].

[20] Ballani, J., 2020. *'Digital Twinning Could Stretch Wind Turbine Life By 25 Years' | Recharge*. [online] Recharge | Latest renewable energy news. Available at: <https://www.rechargenews.com/technology/digital-twinning-could-stretch-wind-turbine-life-by-25-years/2-1-777656> [Accessed 17 November 2020].

[21] O'Sullivan, K., 2020. *Digital Twins, The Airport Operations Control Interface Of The Future*. [online] Sita.aero. Available at: <https://www.sita.aero/pressroom/blog/digital-twins-the-airport-operations-control-interface-of-the-future/#:~:text=Digital%20twins%2C%20the%20airport%20operations%20control%20interface%20of%20the%20future,-Published%20on%2011&text=A%20digital%20twin%20is%20the,it%20goes%20around%20the%20circuit> [Accessed 21 November 2020].

[22] Ørsted, M., 2020. *Improving Airport Decision Making With The Digital Twin Concept*. [online] International Airport Review. Available at: <https://www.internationalairportreview.com/article/101388/digital-twin-concept-data-airport-

operations/#:~:text=The%20Digital%20Twin%20concept%20gives,outcome%20and%20optimise%20the%20result.&text=This%20innovative%20use%20of%20data,steadily%20move%20towards%20digital%20solutions.> [Accessed 21 November 2020].

[23] Zhao, Q., Li, Q. and Wen, J., 2018. Construction and application research of knowledge graph in aviation risk field. In *MATEC Web of Conferences* (Vol. 151, p. 05003). EDP Sciences.

[24] M. Keller, R., 2019, May. Building a knowledge graph for the air traffic management community. In *Companion Proceedings of The 2019 World Wide Web Conference* (pp. 700-704).

[25] Kaltenböck, M., Recheis, D., Bharwani, S. and Barrott, J., 2020. *Creating a global Climate Action Knowledge Graph to leverage artificial intelligence and speed progress on the climate agenda*. [online] Available at: <https://www.poolparty.biz/wp-content/uploads/2020/07/Climate-Change-Positioning-Paper_SWC_REEEP_SEI.pdf> [Accessed 13 January 2021].

[26] Thornton, C., 2020. *Auto-Generated Knowledge Graphs*. [online] Medium. Available at: <https://towardsdatascience.com/auto-generated-knowledge-graphs-92ca99a81121> [Accessed 10 November 2020].

[27] Martin, S., 2016. Strategic Airport Capacity Management at Heathrow. *Impact*, *2*(1), pp.7-12.

[28] NATS. 2021. *Heathrow Airport - NATS*. [online] Available at: <https://www.nats.aero/news/factsheets/heathrow-airport/> [Accessed 8 March 2021].

[29] Mazareanu, E., 2020. *UK: average flight punctuality at Heathrow London 2008-2019 | Statista*. [online] Statista. Available at: <https://www.statista.com/statistics/303697/average-flight-punctuality-at-heathrow-london/> [Accessed 8 March 2021].

[30] Iansen-Rogers, J., 2018. *Heathrow Carbon Footprint 2018*. [online] Heathrow.com. Available at: <https://www.heathrow.com/content/dam/heathrow/web/common/documents/company/heathrow-2-0-sustainability/futher-reading/Heathrow-Carbon-Footprint-report_2018.pdf> [Accessed 10 March 2021].

[31] Neuman, F. and Erzberger, H., 1991. *Analysis of delay reducing and fuel saving sequencing and spacing algorithms for arrival traffic*. National Aeronautics and Space Administration, Ames Research Center.

[32] Rathinam, S., Montoya, J. and Jung, Y., 2008, September. An optimization model for reducing aircraft taxi times at the Dallas Fort Worth International Airport. In *26th International Congress of the Aeronautical Sciences (ICAS)* (pp. 14-19).

[33] Singh, V. and Sharma, S.K., 2015. Fuel consumption optimization in air transport: a review, classification, critique, simple meta-analysis, and future research implications. *European Transport Research Review*, *7*(2), pp.1-24.

[34] Tasca, A.L., Cipolla, V., Abu Salem, K. and Puccini, M., 2021. Innovative Box-Wing Aircraft: Emissions and Climate Change. Sustainability 2021, 13, 3282.

[35] Airbus. 2019. *Airlines are looking to reduce fuel consumption. Wake-energy retrieval could help*. [online] Available at: <https://www.airbus.com/newsroom/stories/airlines-are-looking-to-reduce-fuel-consumption-wake-energy-retrieval-could-help.html> [Accessed 7 January 2021].

[36] Wells, C.A., Williams, P.D., Nichols, N.K., Kalise, D. and Poll, I., 2021. Reducing transatlantic flight emissions by fuel-optimised routing. *Environmental Research Letters*, *16*(2), p.025002.

[37] Flightradar24. 2020. *Live Flight Tracker - Real-Time Flight Tracker Map | Flightradar24*. [online] Available at: <https://www.flightradar24.com/data/airports/lhr> [Accessed 30 November 2020].

[38] Heathrow Airport. 2021. *Which terminal | Heathrow*. [online] Available at: <https://www.heathrow.com/at-the-airport/terminal-guides/which-terminal> [Accessed 30 November 2020].

[39] Networkx.org. 2020. *NetworkX — NetworkX documentation*. [online] Available at: <https://networkx.org/> [Accessed 10 November 2020].

[40] Godfrey, K., 2019. *BA crew have just 20 minutes to get cabin ready for take off between flights*. [online] The Sun. Available at: <https://www.thesun.co.uk/travel/9437437/british-airways-turnaround-cleaning-delayed-flights/> [Accessed 19 April 2021].

[41] Flightsfrom.com. 2021. *List of destinations & airlines from Heathrow, London - FlightsFrom.com*. [online] Available at: <https://www.flightsfrom.com/LHR/destinations> [Accessed 15 March 2021].

[42] Saurabh, 2019. *Fuel Consumption of Popular Aircraft*. [online] Alliknowaviation.com. Available at: <https://alliknowaviation.com/2019/12/14/fuel-consumption-aircraft/> [Accessed 22 March 2021].

[43] Moffitt, M., 2020. *The difference between short-haul, medium-haul and long-haul flights*. [online] Point Hacks. Available at: <https://www.pointhacks.com.au/differences-short-medium-long-haul-flights/?mgn> [Accessed 22 March 2021].

[44] Simplemaps.com. 2020. *World Cities Database | Simplemaps.com*. [online] Available at: <https://simplemaps.com/data/world-cities> [Accessed 22 March 2021].

[45] Greatcirclemap.com. 2021. *Great Circle Map*. [online] Available at: <https://www.greatcirclemap.com/?routes=LHR-JFK%2C%20LHR-

PHL%2C%20LHR-YHZ%2C%20LHR-ORD%2C%20LHR-ATL%2C%20LHR-IAD%2C%20LHR-MIA> [Accessed 12 April 2021].

[46] Heathrow Airport. 2019. *Traffic statistics | Heathrow*. [online] Available at: <https://www.heathrow.com/company/about-heathrow/performance/airport-operations/traffic-statistics> [Accessed 15 April 2021].

[47] FlightDeckFriend.com | Pilot Jobs | Flight Training | Aspiring Pilots. 2021. *How Much Does Jet Fuel Cost? | Jet A1 | FlightDeckFriend.com*. [online] Available at: <https://www.flightdeckfriend.com/ask-a-pilot/how-much-does-jet-fuel-cost/#:~:text=Due%20to%20collapse%20in%20oil,to%20around%20%240.20%20per%20KG.%20> [Accessed 15 April 2021].

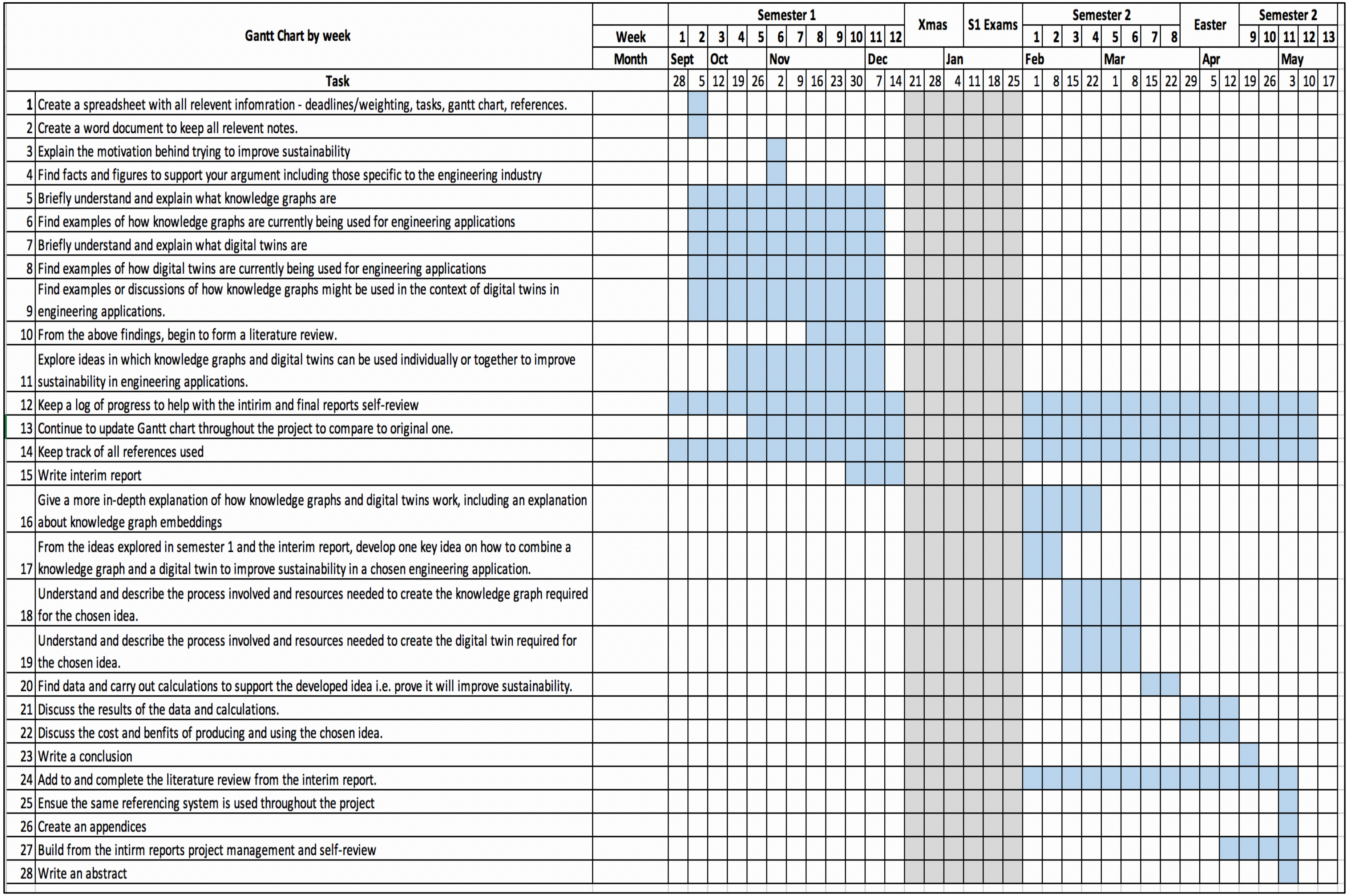
[48] Verifavia.com. 2021. *How are aircraft CO2 emissions calculated?*. [online] Available at: <https://www.verifavia.com/greenhouse-gas-verification/fq-how-are-aircraft-co2-emissions-calculated-11.php> [Accessed 15 April 2021].

[49] All Things On Time Performance. 2020. *Flight Delays in Numbers – Not Only Painful For Passengers*. [online] Available at: <https://www.allthingsontimeperformance.com/flight-delays-in-numbers-not-only-painful-for-passengers/#:~:text=The%20average%20delay%20on%20departing,a%20flight's%20scheduled%20departure%20time> [Accessed 17 April 2021].

[50] Racecar Engineering. 2021. *Slipstream and 'dirty air' explained - Racecar Engineering*. [online] Available at: <https://www.racecar-engineering.com/tech-explained/slipstream-and-dirty-air-explained/> [Accessed 26 April 2021].

[51] Matthews, R., 2021. *Can you save fuel by slipstreaming other vehicles?*. [online] Science Focus. Available at: <https://www.sciencefocus.com/science/can-you-save-fuel-by-slipstreaming-other-vehicles/> [Accessed 26 April 2021].

Appendix 1: *Original Gantt Chart*



Appendix 2: *Excel code/functions used for formatting flight data*

1-A	B	C	D	E
2	**Command**	**Code**	**Before**	**After**
3	Removing reg number	=LEFT(D3,(FIND("(",D3))-2)	A319 (F-GRHY)	A319
4	Assigning aircraft	=VLOOKUP(D4,Aircrafts!A2:$B$35,2,0)	A319	Airbus A319-100
5		=VLOOKUP(D5,Aircrafts!A3:$B$35,2,0)	B789	Boeing 787-9 Dreamliner
6		=VLOOKUP(D6,Aircrafts!A4:$B$35,2,0)	A332	Airbus A330-200
7	Removing airport code	=LEFT(D7,(FIND("(",D7))-2)	Paris (CDG)	Paris
8	Changing a string into hours	=LEFT(D8,(FIND("h",D8))-1)	1h 40m	1
9		=MID(D8,FIND(" ",D8)+1,15)		40m
10		=VALUE(LEFT(E9,(FIND("m",E9))-1))		40
11		=VALUE(RIGHT(E8,LEN(E8)-1))		1
12		=E11+(E10/60)		1.666666667
13	Assigning Flight Time	=VLOOKUP(D13,Countries!$A$2:$B$260,2,0)	Belfast	1.41666667
14		=VLOOKUP(D14,Countries!$A$2:$B$260,2,0)	Amsterdam	1.33333333
15		=VLOOKUP(D15,Countries!$A$2:$B$260,2,0)	Frankfurt	1.66666667
16	Assigning Fuel Consumption	=VLOOKUP(D16,Aircrafts!$F$2:$G$35,2,0)	A333	5700
17		=VLOOKUP(D17,Aircrafts!$F$2:$G$35,2,0)	B748	9600
18		=VLOOKUP(D18,Aircrafts!$F$2:$G$35,2,0)	A35K	5800
19	Assigning Latitude	=VLOOKUP(D19,Countries!$D$2:$E$260,2,0)	Dallas	32.7936
20		=VLOOKUP(D20,Countries!$D$2:$E$260,2,0)	Istanbul	41.01
21		=VLOOKUP(D21,Countries!$D$2:$E$260,2,0)	Shanghai	31.1667
22	Assigning Longitude	=VLOOKUP(D22,Countries!$G$2:$H$260,2,0)	Dallas	-96.7662
23		=VLOOKUP(D23,Countries!$G$2:$H$260,2,0)	Istanbul	28.9603
24		=VLOOKUP(D24,Countries!$G$2:$H$260,2,0)	Shanghai	121.4667

Appendix 3: *Python code used to produce the knowledge graphs*

3.1 Python code used to produce Figure 4.2

```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph

G = nx.Graph()  #creates a graph called G

G.add_edge('Flight Code', 'Date/Time')  #add the nodes 'Flight Code' and 'Date/Time' to graph G, connecting them with an edge
G.add_edge('Flight Code', 'Airline Name')
G.add_edge('Flight Code', 'Aircraft Code')
G.add_edge('Flight Code', 'Country/Code')
G.add_edge('Airline Name', 'Terminal Number')

nx.draw(G, with_labels=True)     #draws graph G with labels
plt.show()  #shows the graph plotted
```

3.2 Python code used to produce Figure 4.3

```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph

G = nx.Graph()  #creates a graph called G

#Flight 1
G.add_edge('AF1081','6:25')  #adds the nodes 'AF1081' and '6:25' to graph G, connecting them with an edge
G.add_edge('AF1081','Paris (CDG)')
G.add_edge('AF1081','Air France')
G.add_edge('AF1081','Airbus A319-100')
G.add_edge('Air France','Terminal 2')

#Flight 2
G.add_edge('BA1414','6:25')
G.add_edge('BA1414','Belfast (BHD)')
G.add_edge('BA1414','British Airways')
G.add_edge('BA1414','Airbus A319-100')
G.add_edge('British Airways','Terminal 5')

#Flight 3
G.add_edge('KL1000','6:30')
G.add_edge('KL1000','Amsterdam (AMS)')
G.add_edge('KL1000','KLM')
G.add_edge('KL1000','Boeing B737-800')
G.add_edge('KLM','Terminal 2')

#Flight 4
G.add_edge('LH921','6:30')
G.add_edge('LH921','Frankfurt (FRA)')
G.add_edge('LH921','Lufthansa')
G.add_edge('LH921','Airbus A320')
G.add_edge('Lufthansa','Terminal 2')

#Flight 5
G.add_edge('BA2770','7:10')
G.add_edge('BA2770','Jersey (JER)')
G.add_edge('BA2770','British Airways')
G.add_edge('BA2770','Airbus A319-100')
G.add_edge('British Airways','Terminal 5')


nx.draw(G, with_labels=True)     #draws graph G with labels
plt.show()  #shows the graph plotted
```

3.3 Python code used to produce Figure 4.4

```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph
import xlrd #a library for reading data and formatting information from Excel files

file = "./Desktop/FYP Master File.xls"  #excel file stored as a string

G = nx.Graph()  #creates a graph called G
node0_1 = [] #creates and stores a list to be added to
node0_2 = []
node0_3 = []
node0_4 = []
node3_5 = []

book = xlrd.open_workbook(file) #opens workbook file = "./Desktop/FYP Master File.xls" and stores it as an object called book

sheet = book.sheet_by_index(0) #specifies which sheet in the excel file to use (0 = sheet 1)

for row in range(sheet.nrows):  #iterate strictly to the number of rows of the excel speadsheet
  data = sheet.row_slice(row) #breaks up the rows and creates an object called data which is all of the information on a given row
  Flight = data[1].value  #creates an object called Flight from the value in the second column, row by row
  Time = data[0].value
  Destination = data[2].value
  Airline = data[3].value
  Aircraft = data[4].value
  Terminal = data[5].value
  node0_1.append((Flight, Time)); #creates a tuple and adds it to the list node0_1
  node0_2.append((Flight, Destination));
  node0_3.append((Flight, Airline));
  node0_4.append((Flight, Aircraft));
  node3_5.append((Airline, Terminal));



G.add_edges_from(node0_1) #creates edges between the tuples in the list and adds them to graph G
G.add_edges_from(node0_2)
G.add_edges_from(node0_3)
G.add_edges_from(node0_4)
G.add_edges_from(node3_5)



nx.draw(G, with_labels=True) #draws graph G with labels
plt.show()  #shows the graph plotted
```

3.4 Python code used to produce Figure 4.5

```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph
import xlrd #a library for reading data and formatting information from Excel files

file = "./Desktop/FYP Master File.xls"  #excel file stored as a string

G = nx.Graph()  #creates a graph called G
node0_1 = [] #creates and stores a list to be added to
node0_2 = []
node0_3 = []
node0_4 = []
node3_5 = []
node0_6 = []

book = xlrd.open_workbook(file) #opens workbook file = "./Desktop/FYP Master File.xls" and stores it as an object called book

sheet = book.sheet_by_index(0) #specifies which sheet in the excel file to use (0 = sheet 1)

for row in range(sheet.nrows):  #iterate strictly to the number of rows of the excel speadsheet
  data = sheet.row_slice(row) #breaks up the rows and creates an object called data which is all of the information on a given row
  Flight = data[1].value  #creates an object called Flight from the value in the second column, row by row
  Time = data[0].value
  Destination = data[2].value
  Airline = data[3].value
  Aircraft = data[4].value
  Terminal = data[5].value
  Arrival = data[6].value #arrival flights added
  node0_1.append((Flight, Time)); #creates a tuple and adds it to the list node0_1
  node0_2.append((Flight, Destination));
  node0_3.append((Flight, Airline));
  node0_4.append((Flight, Aircraft));
  node3_5.append((Airline, Terminal));
  node0_6.append((Flight, Arrival));


G.add_edges_from(node0_1) #creates edges between the tuples in the list and adds them to graph G
G.add_edges_from(node0_2)
G.add_edges_from(node0_3)
G.add_edges_from(node0_4)
G.add_edges_from(node3_5)
G.add_edges_from(node0_6)


nx.draw(G, with_labels=True, font_size=10, edge_color='red', edgecolors='black', node_color='white',
        #draws graph G with labels, font and colour adjustments

plt.show()  #shows the graph plotted
```

3.5 Python code used to produce Figures 4.6 and 4.7

```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph
import xlrd #a library for reading data and formatting information from Excel files

file = "./Desktop/FYP Master File.xls"  #excel file stored as a string


book = xlrd.open_workbook(file) #opens workbook file = "./Desktop/FYP Master File.xls" and stores it as an object called book


for i in range(0,2):    #iterates through sheet 1 and then sheet 2
    sheet = book.sheet_by_index(i)
    G = nx.Graph()  #creates a graph called G
    node0_1 = []    #creates and stores a list to be added to
    node0_2 = []
    node0_3 = []
    node0_4 = []
    node3_5 = []
    for row in range(sheet.nrows):  #iterate strictly to the number of rows of the excel speadsheet
        data = sheet.row_slice(row) #breaks up the rows and creates an object called data which is all of the information on a given row
        Flight = data[1].value #creates an object called Flight from the value in the second column, row by row
        Time = data[0].value
        Destination_Origin = data[2].value
        Airline = data[3].value
        Aircraft = data[4].value
        Terminal = data[5].value
        node0_1.append((Flight, Time));  #creates a tuple and adds it to the list node0_1
        node0_2.append((Flight, Destination_Origin));
        node0_3.append((Flight, Airline));
        node0_4.append((Flight, Aircraft));
        node3_5.append((Flight, Terminal));
    G.add_edges_from(node0_1)  #creates edges between the tuples in the list and adds them to graph G
    G.add_edges_from(node0_2)
    G.add_edges_from(node0_3)
    G.add_edges_from(node0_4)
    G.add_edges_from(node3_5)

    plt.figure()
    nx.draw(G, with_labels=True, font_size=6, edge_color='red', edgecolors='black', node_color='white',)
    #draws graph G with labels, font and colour adjustments

plt.show()#shows the graphs plotted
```

Appendix 4: *Python code for fuel saving calculations/flight pairing*

4.1 Python code used for initial fuel savings calculations

```
import pandas as pd #used to read and format information form the Excel file

file = pd.read_excel('./Desktop/FYP Master File.xls') #stores the whole excel file as an object called file

Pairs = []  #creates and stores a list for the pairs of flights to be added to
Fuel_Saved = [] #creates and stores a list of the fuel saveds in each pair to be added to


for i in range(0,file.shape[0]-1):  #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0]  #creates an object called DepTime1 from the value in the first column, row by row
    DepTime2 = file.iloc[i+1,0]    #creates an object called DepTime2 from the value in the first column below the row of DepTime1
    Dest1 = file.iloc[i,3]
    Dest2 = file.iloc[i+1,3]
    FC1 = file.iloc[i,6]    #FC = fuel consumption
    FC2 = file.iloc[i+1,6]
    FT1 = file.iloc[i,7]    #FT = flight time
    FT2 = file.iloc[i+1,7]

    if DepTime1 == DepTime2 and Dest1 == Dest2:
        FS = FC2*FT2*0.075  #Fuel Saved = fuel consumption x flight time x 0.075 (eq 4.1)
        Pairs.append((Dest1,FS))    #creates a tuple of the shared destination and the fuel saved and adds to the pairs list
        Fuel_Saved.append(FS)   #adds the fuel saved per flight pair to the Fuel_Saved list


print(Pairs) #prints the list of tuples from the pairs list
print(sum(Fuel_Saved))  #prints the sum of the fuel saved per flight pair
```

4.2 Python code used for pairing flights based on destination

```
import pandas as pd #used to read and format information form the Excel file

file = pd.read_excel('./Desktop/FYP Master File.xls') #stores the whole excel file as an object called file

Pairs = []  #creates and stores a list for the pairs of flights to be added to


for i in range(0,file.shape[0]-1):  #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0]   #creates an object called DepTime1 from the value in the first column, row by row
    DepTime2 = file.iloc[i+1,0] #creates an object called DepTime2 from the value in the first column below the row of DepTime1
    Dest1 = file.iloc[i,3]
    Dest2 = file.iloc[i+1,3]

    if DepTime1 == DepTime2 and Dest1 == Dest2:
        Pairs.append((Dest1,Dest2)) #creates a tuple of the shared destination and adds to the pairs list


print(Pairs)
print(len(Pairs))
```

4.3 Python code used for pairing flights based on countries

```
import pandas as pd #used to read and format information form the Excel file

file = pd.read_excel('./Desktop/FYP Master File.xls') #stores the whole excel file as an object called file

Pairs = []  #creates and stores a list for the pairs of flights to be added to


for i in range(0,file.shape[0]-1):  #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0]   #creates an object called DepTime1 from the value in the first column, row by row
    DepTime2 = file.iloc[i+1,0] #creates an object called DepTime2 from the value in the first column below the row of DepTime1
    Country1 = file.iloc[i,4]
    Country2 = file.iloc[i+1,4]

    if DepTime1 == DepTime2 and Country1 == Country2:
        Pairs.append((Country1,Country2)) #creates a tuple of the shared destination and adds to the pairs list


print(Pairs)
print(len(Pairs))
```

4.4 Python code used for pairing flights based on distance

```
import pandas as pd #used to read and format information form the Excel file

from geopy.distance import geodesic #used to measure distance between locations using Lat and Lng

file = pd.read_excel('./Desktop/FYP Master File.xls') #stores the whole excel file as an object called file

Pairs = []  #creates and stores a list for the pairs of flights to be added to


for i in range(0,file.shape[0]-1): #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0] #creates an object called DepTime1 from the value in the first column, row by row
    Dest1 = file.iloc[i,2]
    FT1 = file.iloc[i,8]
    Lat1 = file.iloc[i,6]
    Lng1 = file.iloc[i,7]

    DepTime2 = file.iloc[i+1,0]
    Dest2 = file.iloc[i+1,2]
    Lat2 = file.iloc[i+1,6]
    Lng2 = file.iloc[i+1,7]

    Dist = geodesic((Lat1, Lng1), (Lat2, Lng2)).miles   #distance in miles between Dest1 and Dest2

    if DepTime1 == DepTime2 and Dist < 1000 and FT1 >= 6:   #FT = flight time
        Pairs.append((Dest1, Dest2, Dist))  #adds destinations and distance between them to the pairs list


print(Pairs)
print(len(Pairs))
```

4.5 Python code used for pairing flights based on distance and printing all necessary information needed for fuel saving calculations

```
import pandas as pd #used to read and format information form the Excel file

from geopy.distance import geodesic #used to measure distance between locations using Lat and Lng

file = pd.read_excel('./Desktop/FYP Master File.xls')   #excel file storedd as a string

Pairs = []  #creates and stores a list for the pairs of flights to be added to
LH = []

for i in range(0,file.shape[0]-1):  #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0]   #creates an object called DepTime1 from the value in the first column, row by row
    Dest1 = file.iloc[i,2]
    Lat1 = file.iloc[i,6]
    Lng1 = file.iloc[i,7]
    FT1 = file.iloc[i,8]     #FT = Flight Time
    AC1 = file.iloc[i,9]     #AC = Aircraft
    FC1 = file.iloc[i,10]    #FC = Fuel Consumption

    DepTime2 = file.iloc[i+1,0]
    Dest2 = file.iloc[i+1,2]
    Lat2 = file.iloc[i+1,6]
    Lng2 = file.iloc[i+1,7]
    FT2 = file.iloc[i+1,8]
    AC2 = file.iloc[i+1,9]
    FC2 = file.iloc[i+1,10]

    Dist = geodesic((Lat1, Lng1), (Lat2, Lng2)).miles   #distance in miles between Dest1 and Dest2

    if DepTime1 == DepTime2 and Dist < 1000 and FT1 >= 6:
        LH.append((Dest1, Dest2, Dist)) #adds destinations 1 and 2 and the distance between them to the list LH (long haul)
        Pairs.append((Dest1, FT1, AC1, FC1, Dest2, FT2, AC2, FC2))  #adds all relavent info needed for fuel calculations to list pairs


print(Pairs)
print('Long Haul Flights =',len(LH))
```

Appendix 5: *List of aircraft with their associated codes and fuel consumption*

Aircraft Codes:	Aircraft:	Fuel Consumption (kg/h):
A319	Airbus A319-100	2374
A320	Airbus A320	2430
A321	Airbus A321-231	2740
A332	Airbus A330-200	5590
A333	Airbus A330-300	5700
A359	Airbus A350-900	5800
A35K	Airbus A350-1000	5800
A388	Airbus A380-800	11500
B748	Boeing 747-8	9600
B77L	Boeing 777-200(LR)	6800
B77W	Boeing 777-300(ER)	7500
B788	Boeing 787-8 Dreamliner	4900
B789	Boeing 787-9 Dreamliner	5600
A20N	Airbus A320-271N	2430
B772	Boeing 777-200(ER)	6630
A21N	Airbus A321-100	2740
B764	Boeing 767-400(ER)	5100
B78X	Boeing 787-10 Dreamliner	6300
B763	Boeing 767-300(ER)	4940
B738	Boeing 737-800	2530
B744	Boeing 747-400	10230
E145	Embraer ERJ-145EP	1120
E190	Embraer E190LR	1970
E195	Embraer E195LR	1950
A318	Airbus A318	1720
E75S	Embraer E175SC	1650
B737	Boeing 737-700	2420
B787	Boeing 787-10 Dreamliner	6300
B773	Boeing 777-300(ER)	7500
E145	Embraer ERJ-145EP	1120
ER4	Embraer ERJ-145EP	1120
E90	Embraer E190LR	1970
E95	Embraer E195LR	1950

9. PROJECT MANAGEMENT

The Gantt chart that was devised during the Interim report stage can be seen in figure 9.1. As shown, detailed tasks were listed with predicted time frames and many of them were completed or at least started at the Interim report stage, showing good planning and progress. The revised Gantt chart of how the project actually went can be seen in figure 9.2, with an added column to right showing where in the report the tasks outcome can be found. Overall, the two Gantt charts are very similar with only a few changes to when tasks were completed and as discussed previously, the digital twin concept was not explored further meaning tasks 17, 24, 25 and 26 were incomplete. However, as you can see in the week starting May 3rd, tasks 25 and 26 do have a blue square in them meaning they were started, as the digital twin concept was discussed in the conclusions and future work Section. Although the time frames given for most of the tasks were reasonable and were followed in terms of weeks, some tasks required many more hours in a day than was expected. This was because, as described in Sections 4 and 5, lots of Excel and Python code was required, both of which had to be learnt throughout the project.

Gantt Chart by week

	Task	Status
1	Create a spreadsheet with all relevent infomration - deadlines/weighting, tasks, gantt chart, references.	Completed
2	Create a word document to keep all relevent notes.	Completed
3	Explain the motivation behind trying to improve sustainability	Started
4	Find facts and figures to support your argument including those specific to the engineering industry	Started
5	Briefly understand and explain what knowledge graphs are	Completed
6	Find examples of how knowledge graphs are currently being used for engineering applications	Completed
7	Briefly understand and explain what digital twins are	Completed
8	Find examples of how digital twins are currently being used for engineering applications	Completed
9	Find examples or discussions of how knowledge graphs might be used in the context of digital twins in engineering applications.	Completed
10	From the above findings, begin to form a literature review.	Completed
11	Explore ideas in which knowledge graphs and digital twins can be used individually or together to improve sustainability in engineering applications.	Completed
12	Keep a log of progress to help with the intirim and final reports self-review	Ongoing
13	Continue to update Gantt chart throughout the project to compare to original one.	Ongoing
14	Keep track of all references used	Ongoing
15	Write interim report	Completed
16	Give a more in-depth explanation of how knowledge graphs and digital twins work, including an explanation about knowledge graph embeddings	
17	From the ideas explored in semester 1 and the interim report, develop one key idea on how to combine a knowledge graph and a digital twin to improve sustainability in a chosen engineering application.	Ongoing
18	Understand and describe the process involved and resources needed to create the knowledge graph required for the chosen idea.	Ongoing
19	Build apon the literature review, include a critical analysis of Heathrows current air traffic management system	
20	Create an ontology for a knowledge graph of Heathrows flights	Completed
21	Create a knowledge graph of 50 departure flights	
22	Incorporate arrival flights into the knowledge graph	
23	Use the knowledge graph to explore ways in which to improve traffic efficiency/environmental sustainability	
24	Understand and describe the process involved and resources needed to create the digital twin required for the chosen idea.	
25	Explore how the digital twin and knowledge graph can work together	
26	Explore ways in which a digital twin of Heathrow could be used to improve environmental sustaiability	
27	Produce and employ a method for measuring and evaluating the effects of the propsed stategies on the environmental sustainability of Heathrow	
28	Discuss the cost and benfits of producing and using the chosen idea.	
29	Write a conclusion	
30	Ensure the same referencing system is used throughout the project	
31	Create an appendices	
32	Build from the intirm reports project management and self-review	
33	Write an abstract	

Figure 9.1. *Gantt chart devised during the Interim report stage*

	Gantt Chart by week — Task	Status	Section in Report
1	Create a spreadsheet with all relevent infomraition - deadlines/weighting, tasks, gantt chart, references.	Complete	N/A
2	Create a word document to keep all relevent notes.	Complete	N/A
3	Explain the motivation behind trying to improve sustainability	Complete	3.2
4	Find facts and figures to support your argument including those specific to the engineering industry	Complete	3.2
5	Briefly understand and explain what knowledge graphs are	Complete	1
6	Find examples of how knowledge graphs are currently being used for engineering applications	Complete	3.3
7	Briefly understand and explain what digital twins are	Complete	1
8	Find examples of how digital twins are currently being used for engineering applications	Complete	3.3
9	Find examples or discussions of how knowledge graphs might be used in the context of digital twins in engineering applications.	Complete	3.3
10	From the above findings, begin to form a literature review.	Complete	3.3
11	Explore ideas in which knowledge graphs and digital twins can be used individually or together to improve sustainability in engineering applications.	Complete	3.3
12	Keep a log of progress to help with the intirim and final reports self-review	Complete	N/A
13	Continue to update Gantt chart throughout the project to compare to original one.	Complete	9
14	Keep track of all references used	Complete	8
15	Write interim report	Complete	N/A
16	Give a more in-depth explanation of how knowledge graphs and digital twins work, including an explanation about knowledge graph embeddings	Complete	2.2
17	From the ideas explored in semester 1 and the interim report, develop one key idea on how to combine a knowledge graph and a digital twin to improve sustainability in a chosen engineering application.	Incomplete	7
18	Understand and describe the process involved and resources needed to create the knowledge graph required for the chosen idea.	Complete	4.1.1\4.1.2
19	Build apon the literature review, include a critical analysis of Heathrows current air traffic management system	Complete	3.4
20	Create an ontology for a knowledge graph of Heathrows flights	Complete	4.1.1
21	Create a knowledge graph of 50 departure flights	Complete	4.1.2
22	Incorporate arrival flights into the knowledge graph	Complete	3.1.2
23	Use the knowledge graph to explore ways in which to improve traffic efficiency/environmental sustainability	Complete	4.2
24	Understand and describe the process involved and resources needed to create the digital twin required for the chosen idea.	Incomplete	N/A
25	Explore how the digital twin and knowledge graph can work together	Incomplete	7
26	Explore ways in which a digital twin of Heathrow could be used to improve environmental sustainability	Incomplete	7
27	Produce and employ a method for measuring and evaluating the effects of the propsed stategies on the environmental sustainability of Heathrow	Complete	5
28	Discuss the cost and benfits of producing and using the chosen idea.	Complete	5\6
29	Write a conclusion	Complete	7
30	Ensure the same referencing system is used throughout the project	Complete	8
31	Create an appendices	Complete	Appendices
32	Build from the intirm reports project management and self-review	Complete	9\10
33	Write an abstract	Complete	Abstract

Figure 9.2. *Revised Gantt chart showing timeline of when tasks were actioned*

10. SELF REVIEW

Having felt overwhelmed with the open-ended nature of the project title, I chose to add the 'environmental sustainability' aspect, therefore, adding a clear aim and tailoring the project towards something I feel passionate about. At first, in an attempt to avoid coding, something I did not see as a strength of mine, I spent time with other pieces of software i.e., protégé, upon reflection, this was not an efficient use of my time. When I did come to the decision to use Python, I was pleasantly surprised that I managed to produce the initial graph I was hoping to in much less time than I thought. At that point I was happy with the progress I had made with regards to the knowledge graph aspect of the project and had planned to make improvements to the graphs. As mentioned previously it was intended to change the colours of the codes and add labels to the edges to make it clear which flights were arrivals, and which were departures. I am disappointed I did not manage to achieve this, as I do believe if given more time, I would be able to. However, I do believe I made the right decision to focus on the next aspect of the project to find quantifiable results.

Overall, I am very pleased with the new skills I have learnt throughout this project. Not only did I improve my coding skills within Python, with Networkx and Geopy, I have also learnt many new skills within Excel. Although I have used Excel throughout the years in both my job and at university, I had only used some of the basic functions. This project has led me to use Excel in a new way, learning more complex functions which I can now apply when using Excel in the future. I have also learnt how to read and format data from Excel files in Python using the packages xlrd and Pandas. I believe all of these skills are valuable and may be considered desirable to a future employer.

Upon reflection, I have deviated slightly from the original topic, which was knowledge graphs. However, I feel this shows initiative as I have directed the project towards something I feel very passionate about, environmental sustainability, and exploring the topic more throughout this project has confirmed that this is the area of work I would like to go into after university. Although I did not achieve the full aim, as the digital twin concept was not explored further, the majority of the objectives were achieved, and I do believe the correct decision was made to focus on the fuel saving aspect of the project in order to get quantifiable results.