Aerospace Engineering Final Year Project

Knowledge Graph Analysis For Improving Sustainability in Engineering Applications

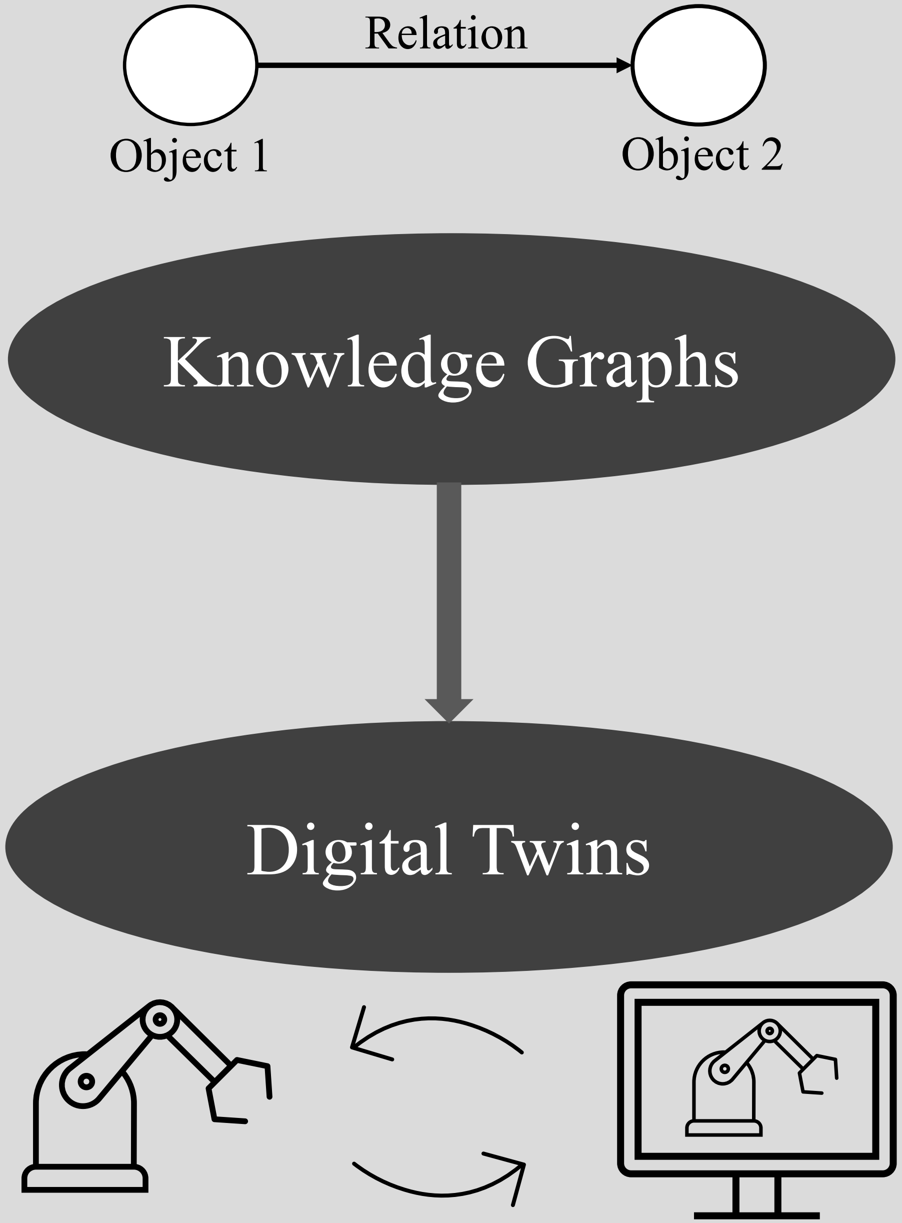
Presenter

A.V.MYTHEN

Outline

- Introduction
- Literature Review
- Constructing the Knowledge Graph
- Knowledge Graph on 50 arrival flights and 50 departure flights
- Separated Knowledge Graphs
- Knowledge Graph Analysis
- Wake-Energy Retrieval
- Initial Fuel Saving Calculations
- Finding Flight Pairs
- Final Calculations
- Results and Discussions
- Conclusions
- Future Work
- References

Introduction



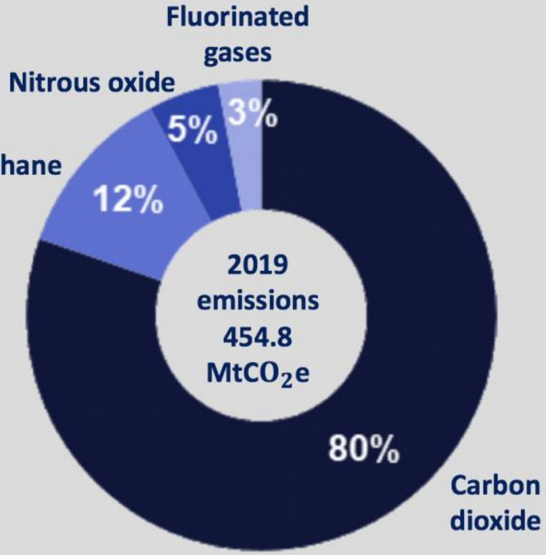
Project Aim: To develop an idea involving knowledge graphs and digital twins to improve environmental sustainability in a chosen engineering application.

1. Explain the motivation behind trying to improve sustainability, drawing upon the current facts and figures i.e., carbon emissions.
2. Understand and explain what knowledge graphs and digital twins are and find examples of how they are currently being used for engineering applications.
3. Find examples or discussions of how knowledge graphs might be used in the context of digital twins in engineering applications.
4. 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.
5. Find data and carry out calculations to support the developed idea i.e., prove it will improve sustainability.
6. Discuss the cost and benefits of producing and using the chosen idea.

Literature Review

Environmental Sustainability

- "...necessary for survival." [1]
- 80% of greenhouse gas emissions in the UK was carbon dioxide [2].
- The **aviation sector** has the largest growth rates for greenhouse gas emissions within the transportation sector, within developed countries [3].

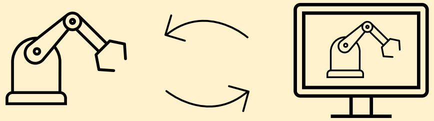


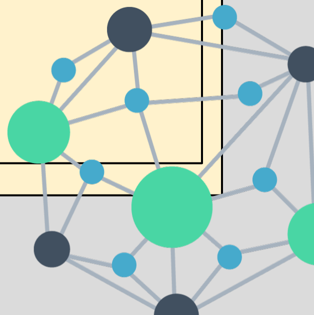
Air Traffic and Fuel Saving Techniques

- Heathrow's current air traffic control service, NATs, do not prioritise environmental sustainability [4] [5].
- Fuel consumption optimization in air transport is increasing in popularity [6].
- PARSIFAL project – box-wing aircraft – 20% fuel consumption reduction [7].
- Airbus – wake-energy retrieval – 5-10% fuel consumption reduction [8].

Knowledge Graphs and Digital Twins

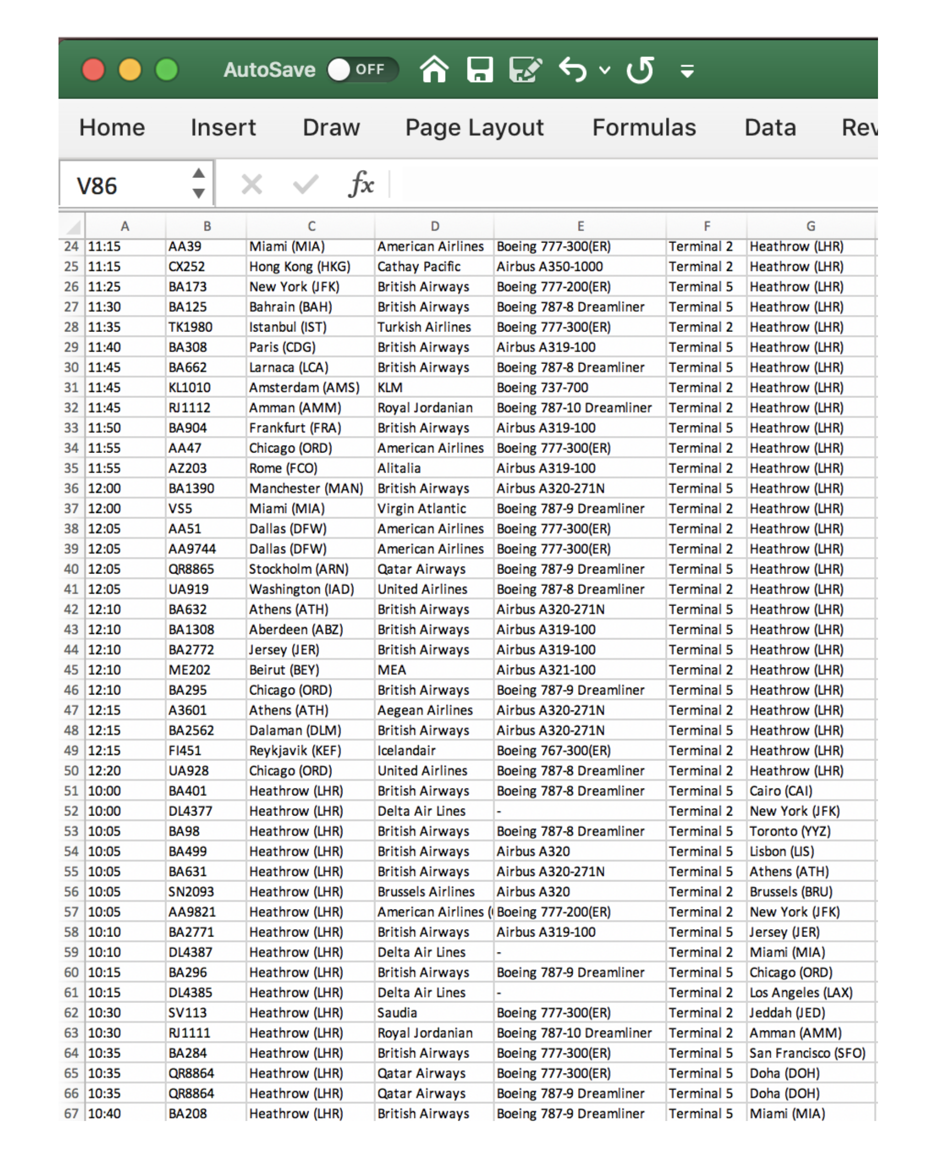
- Lots of literature was reviewed on the topics of knowledge graphs and digital twins separately.
- Limited literature about knowledge graphs being used in the context of digital twins.
- No literature found on them being used together in the context of environmental sustainability.





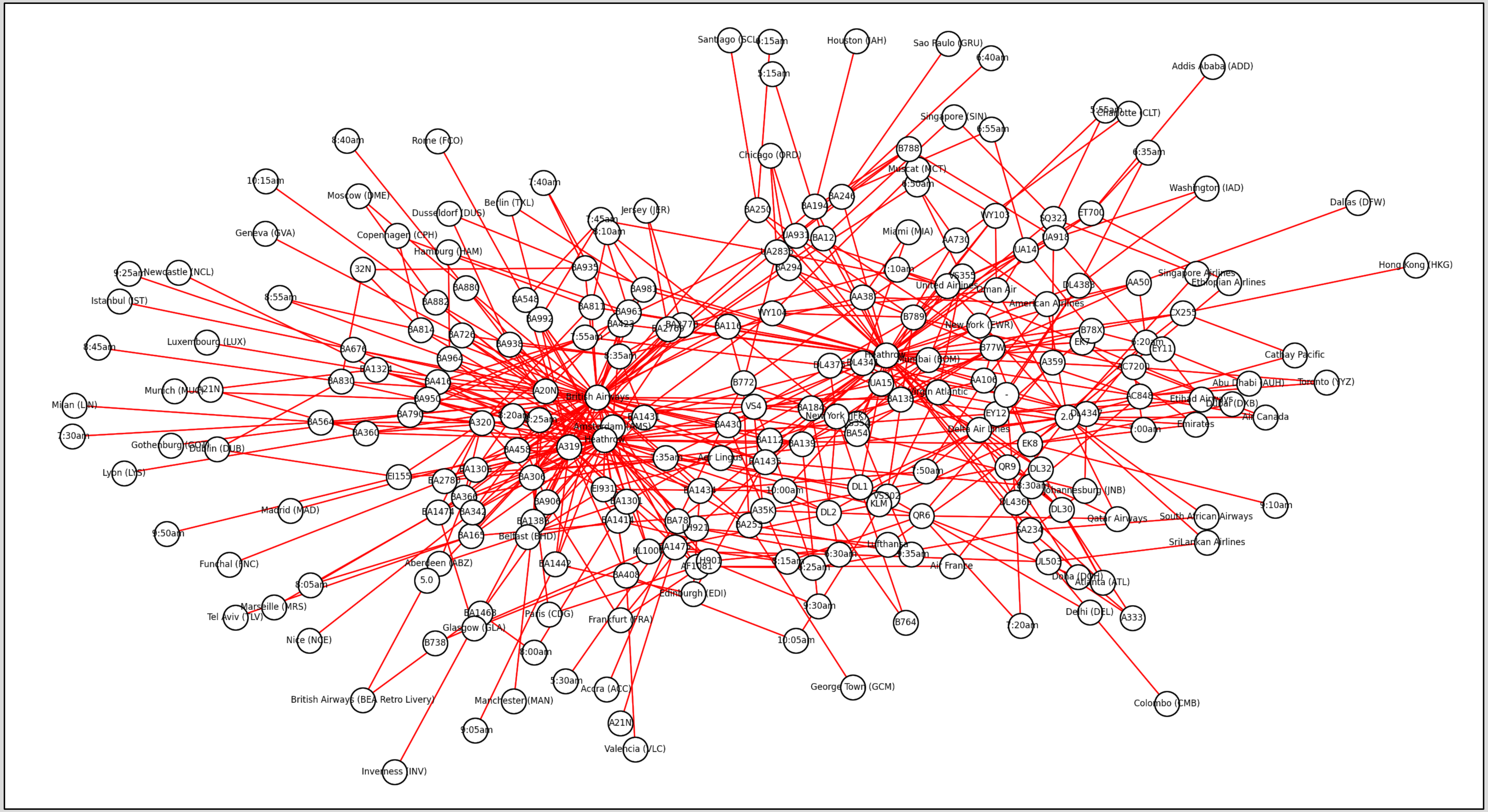
Constructing the Knowledge Graph

```python
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
```



	A	B	C	D	E	F	G
24	11:15	AA39	Miami (MIA)	American Airlines	Boeing 777-300(ER)	Terminal 2	Heathrow (LHR)
25	11:15	CX252	Hong Kong (HKG)	Cathay Pacific	Airbus A350-1000	Terminal 2	Heathrow (LHR)
26	11:25	BA173	New York (JFK)	British Airways	Boeing 777-200(ER)	Terminal 5	Heathrow (LHR)
27	11:30	BA125	Bahrain (BAH)	British Airways	Boeing 787-8 Dreamliner	Terminal 5	Heathrow (LHR)
28	11:35	TK1980	Istanbul (IST)	Turkish Airlines	Boeing 777-300(ER)	Terminal 2	Heathrow (LHR)
29	11:40	BA308	Paris (CDG)	British Airways	Airbus A319-100	Terminal 5	Heathrow (LHR)
30	11:45	BA662	Larnaca (LCA)	British Airways	Boeing 787-8 Dreamliner	Terminal 5	Heathrow (LHR)
31	11:45	KL1010	Amsterdam (AMS)	KLM	Boeing 737-700	Terminal 2	Heathrow (LHR)
32	11:45	RJ1112	Amman (AMM)	Royal Jordanian	Boeing 787-10 Dreamliner	Terminal 2	Heathrow (LHR)
33	11:50	BA904	Frankfurt (FRA)	British Airways	Airbus A319-100	Terminal 5	Heathrow (LHR)
34	11:55	AA47	Chicago (ORD)	American Airlines	Boeing 777-300(ER)	Terminal 2	Heathrow (LHR)
35	11:55	AZ203	Rome (FCO)	Alitalia	Airbus A319-100	Terminal 2	Heathrow (LHR)
36	12:00	BA1390	Manchester (MAN)	British Airways	Airbus A320-271N	Terminal 5	Heathrow (LHR)
37	12:00	VS5	Miami (MIA)	Virgin Atlantic	Boeing 787-9 Dreamliner	Terminal 2	Heathrow (LHR)
38	12:05	AA51	Dallas (DFW)	American Airlines	Boeing 777-300(ER)	Terminal 2	Heathrow (LHR)
39	12:05	AA9744	Dallas (DFW)	American Airlines	Boeing 777-300(ER)	Terminal 2	Heathrow (LHR)
40	12:05	QR8865	Stockholm (ARN)	Qatar Airways	Boeing 787-9 Dreamliner	Terminal 5	Heathrow (LHR)
41	12:05	UA919	Washington (IAD)	United Airlines	Boeing 787-8 Dreamliner	Terminal 2	Heathrow (LHR)
42	12:10	BA632	Athens (ATH)	British Airways	Airbus A320-271N	Terminal 5	Heathrow (LHR)
43	12:10	BA1308	Aberdeen (ABZ)	British Airways	Airbus A319-100	Terminal 5	Heathrow (LHR)
44	12:10	BA2772	Jersey (JER)	British Airways	Airbus A319-100	Terminal 5	Heathrow (LHR)
45	12:10	ME202	Beirut (BEY)	MEA	Airbus A321-100	Terminal 2	Heathrow (LHR)
46	12:10	BA295	Chicago (ORD)	British Airways	Boeing 787-9 Dreamliner	Terminal 5	Heathrow (LHR)
47	12:15	A3601	Athens (ATH)	Aegean Airlines	Airbus A320-271N	Terminal 2	Heathrow (LHR)
48	12:15	BA2562	Dalaman (DLM)	British Airways	Airbus A320-271N	Terminal 5	Heathrow (LHR)
49	12:15	FI451	Reykjavik (KEF)	Icelandair	Boeing 767-300(ER)	Terminal 2	Heathrow (LHR)
50	12:20	UA928	Chicago (ORD)	United Airlines	Boeing 787-8 Dreamliner	Terminal 2	Heathrow (LHR)
51	10:00	BA401	Heathrow (LHR)	British Airways	Boeing 787-8 Dreamliner	Terminal 5	Cairo (CAI)
52	10:00	DL4377	Heathrow (LHR)	Delta Air Lines	-	Terminal 2	New York (JFK)
53	10:05	BA98	Heathrow (LHR)	British Airways	Boeing 787-8 Dreamliner	Terminal 5	Toronto (YYZ)
54	10:05	BA499	Heathrow (LHR)	British Airways	Airbus A320	Terminal 5	Lisbon (LIS)
55	10:05	BA631	Heathrow (LHR)	British Airways	Airbus A320-271N	Terminal 5	Athens (ATH)
56	10:05	SN2093	Heathrow (LHR)	Brussels Airlines	Airbus A320	Terminal 2	Brussels (BRU)
57	10:05	AA9821	Heathrow (LHR)	American Airlines (	Boeing 777-200(ER)	Terminal 2	New York (JFK)
58	10:10	BA2771	Heathrow (LHR)	British Airways	Airbus A319-100	Terminal 5	Jersey (JER)
59	10:10	DL4387	Heathrow (LHR)	Delta Air Lines	-	Terminal 2	Miami (MIA)
60	10:15	BA296	Heathrow (LHR)	British Airways	Boeing 787-9 Dreamliner	Terminal 5	Chicago (ORD)
61	10:15	DL4385	Heathrow (LHR)	Delta Air Lines	-	Terminal 2	Los Angeles (LAX)
62	10:30	SV113	Heathrow (LHR)	Saudia	Boeing 777-300(ER)	Terminal 2	Jeddah (JED)
63	10:30	RJ1111	Heathrow (LHR)	Royal Jordanian	Boeing 787-10 Dreamliner	Terminal 2	Amman (AMM)
64	10:35	BA284	Heathrow (LHR)	British Airways	Boeing 777-300(ER)	Terminal 5	San Francisco (SFO)
65	10:35	QR8864	Heathrow (LHR)	Qatar Airways	Boeing 777-300(ER)	Terminal 5	Doha (DOH)
66	10:35	QR8864	Heathrow (LHR)	Qatar Airways	Boeing 787-9 Dreamliner	Terminal 5	Doha (DOH)
67	10:40	BA208	Heathrow (LHR)	British Airways	Boeing 787-9 Dreamliner	Terminal 5	Miami (MIA)

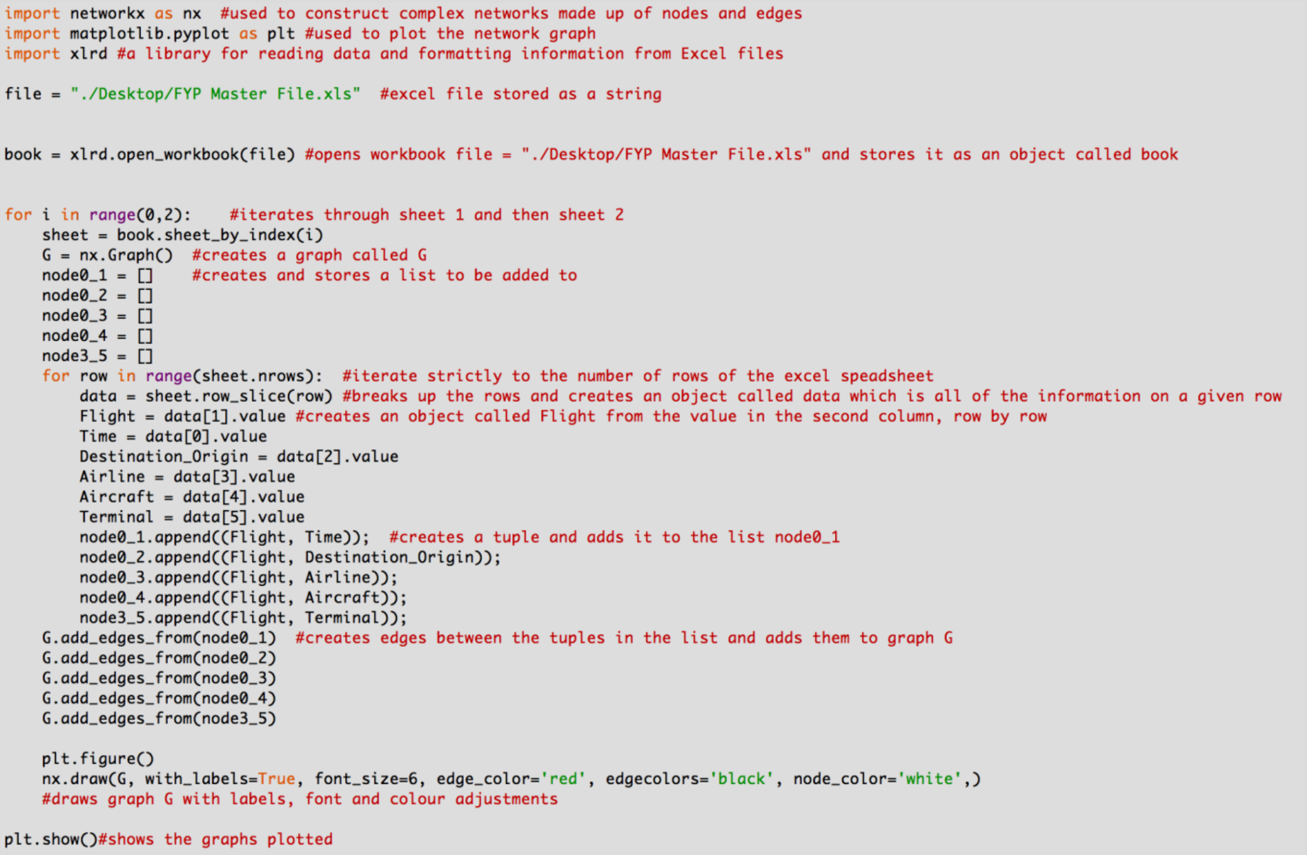
Knowledge Graph of 50 arrival flights and 50 departure flights



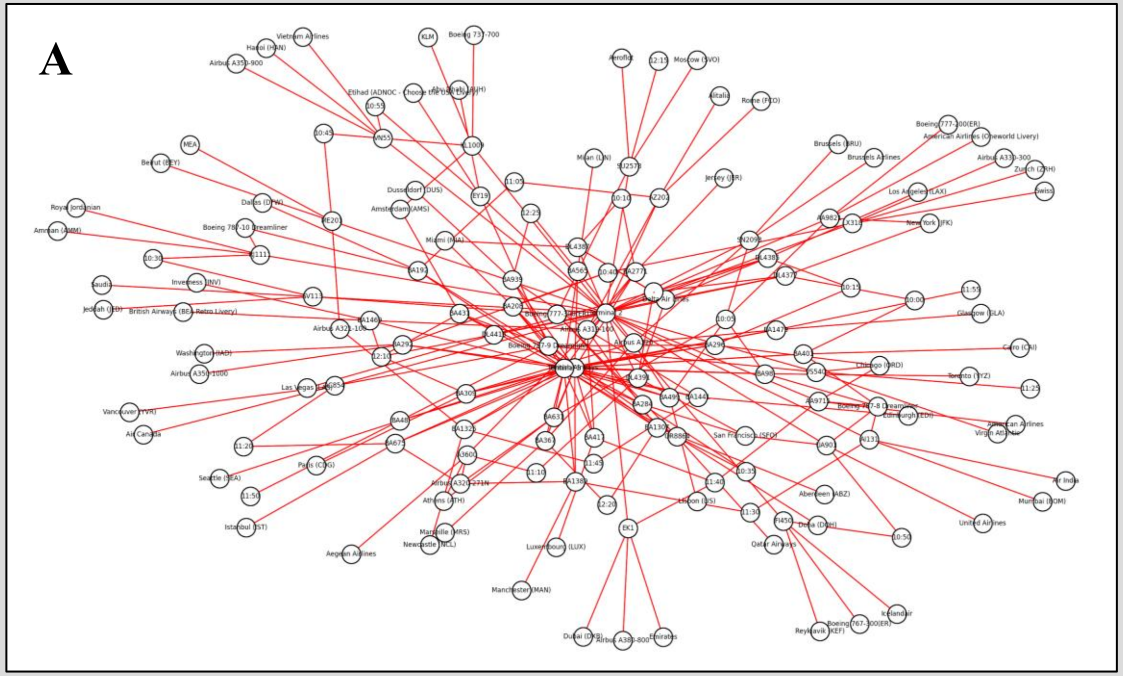
Separated Knowledge Graphs

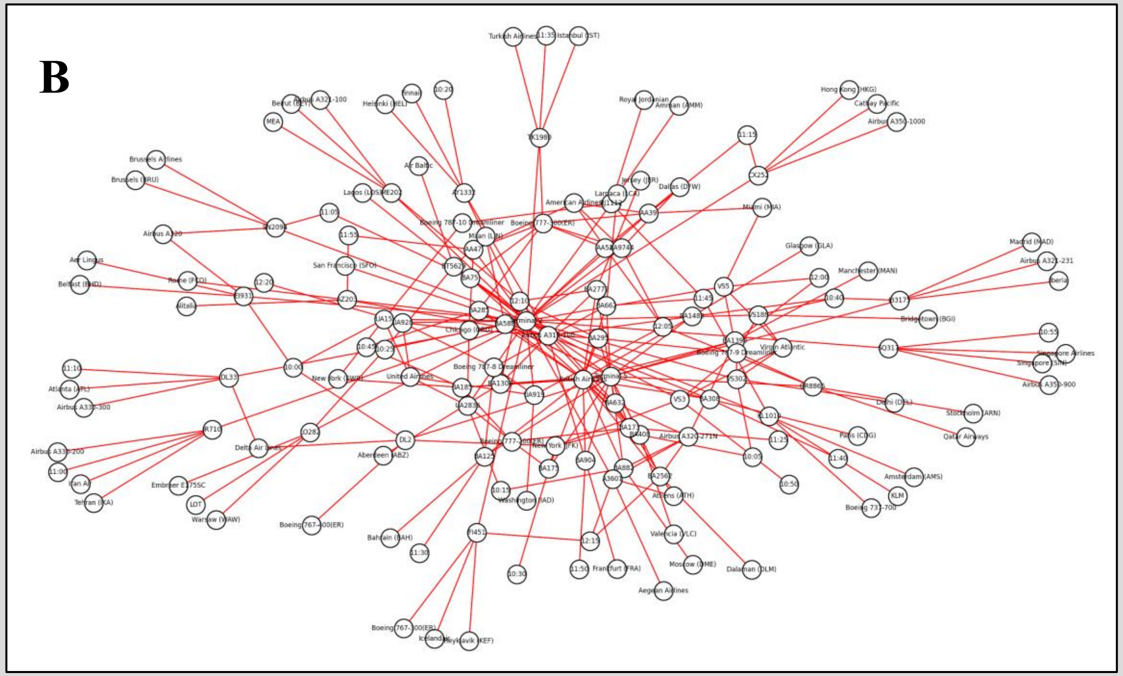
A - 50 arrival flights

B - 50 departure flights

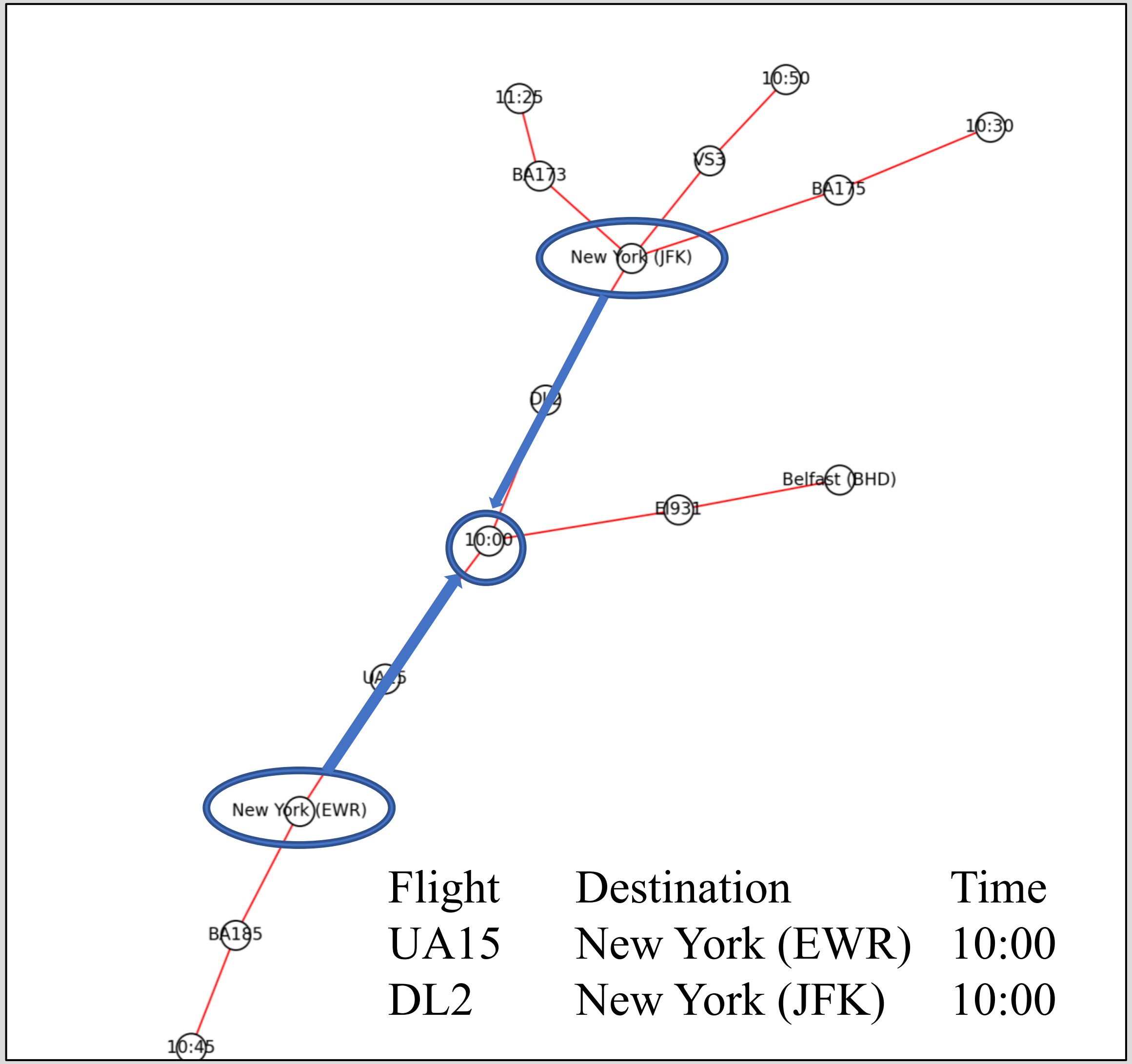


```
import networkx as nx  #used to construct complex networks made up of nodes and edges
import matplotlib.pyplot as plt #used to plot the network graph
import xlrd #a library for reading data and formatting information from Excel files

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
```

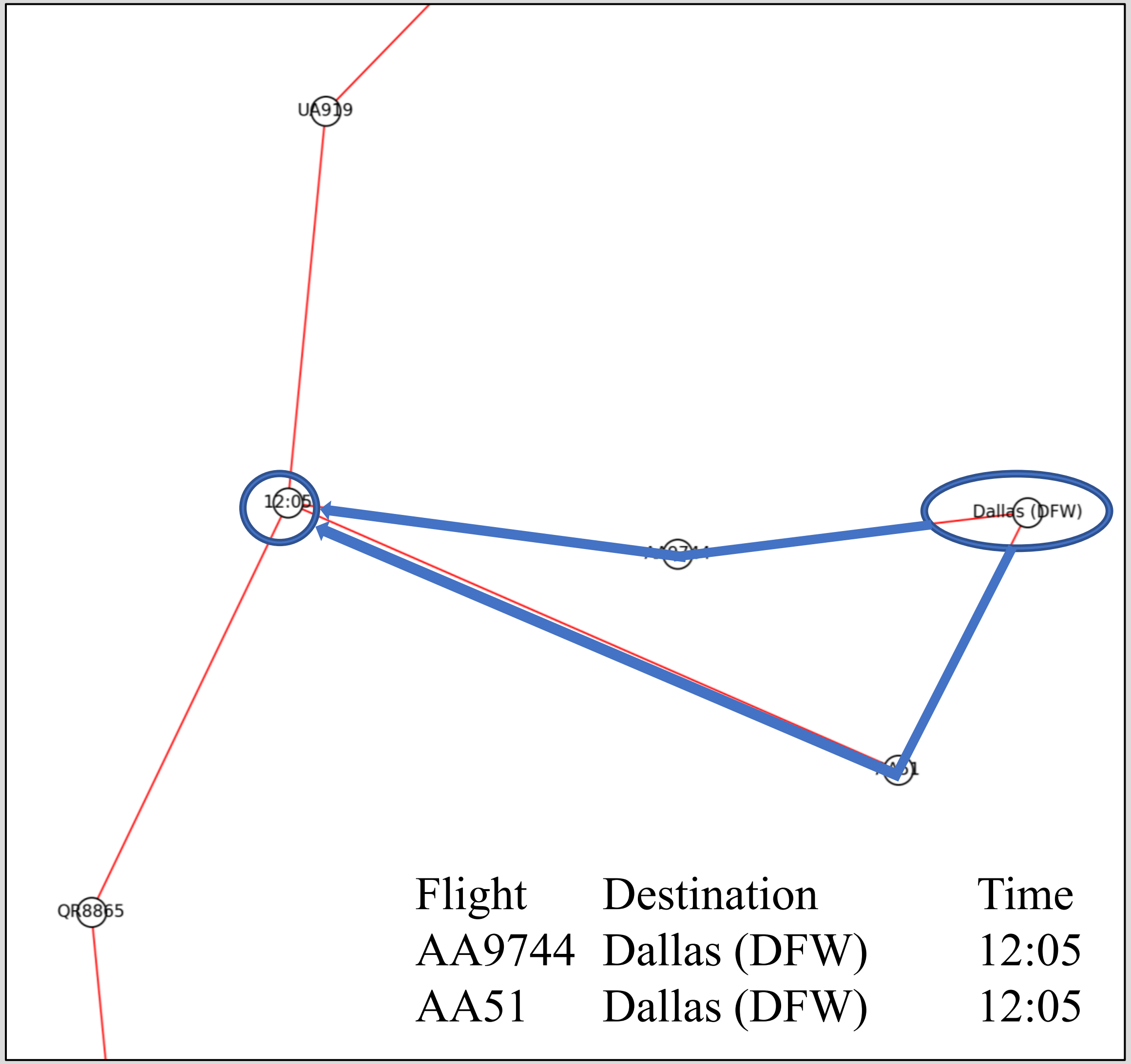




Knowledge Graph Analysis



Flight	Destination	Time
UA15	New York (EWR)	10:00
DL2	New York (JFK)	10:00

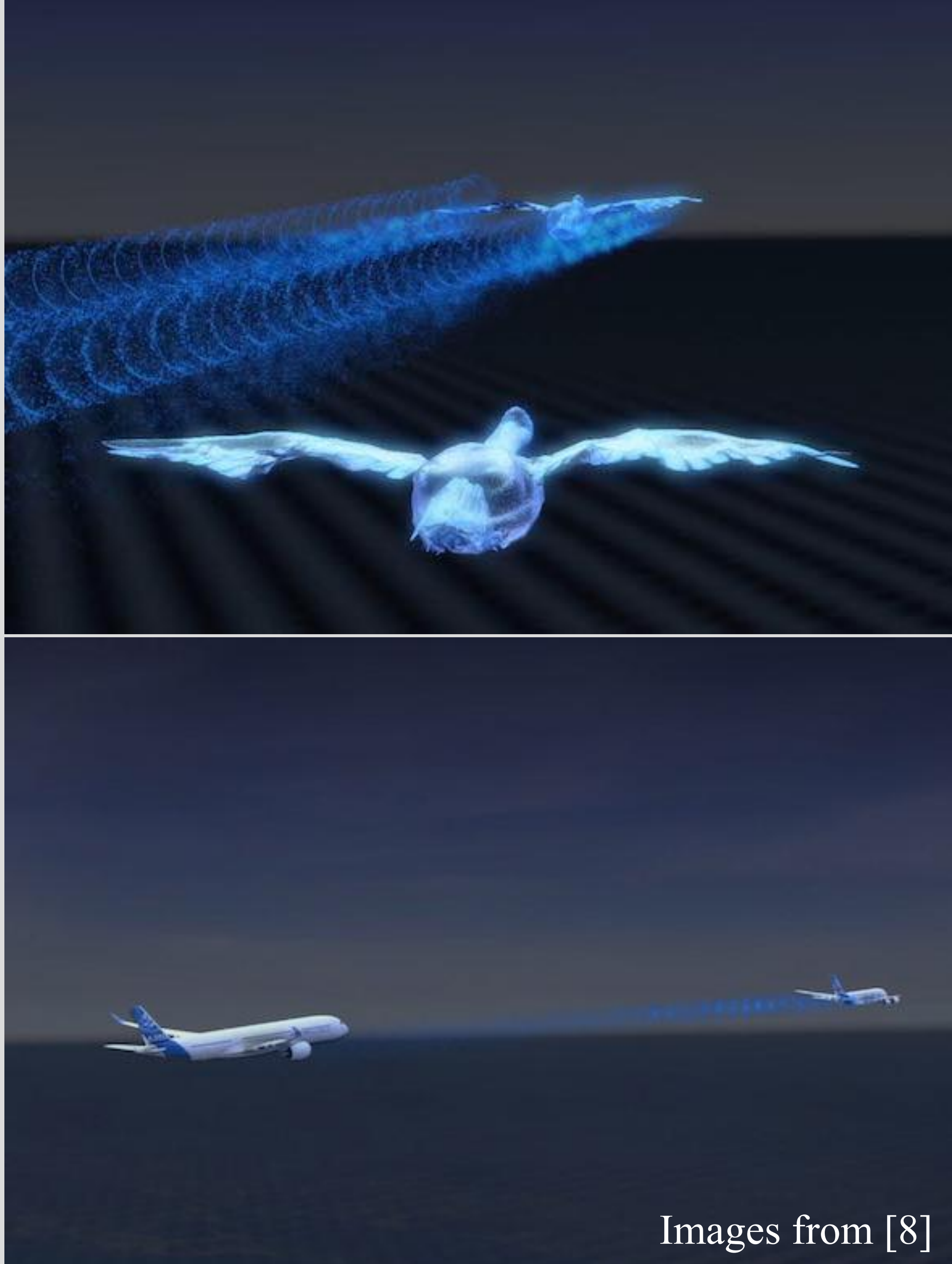


Flight	Destination	Time
AA9744	Dallas (DFW)	12:05
AA51	Dallas (DFW)	12:05

Wake-Energy Retrieval

- Birds fly in a “V” shape to save energy.
- Wake coming from the leader creates a smooth current – upwash.
- Allows the following bird to benefit from free lift, expending less energy.
- Airbus are hoping the same principle would apply to aircraft.
- Reduce fuel consumption by 5-10% per trip.

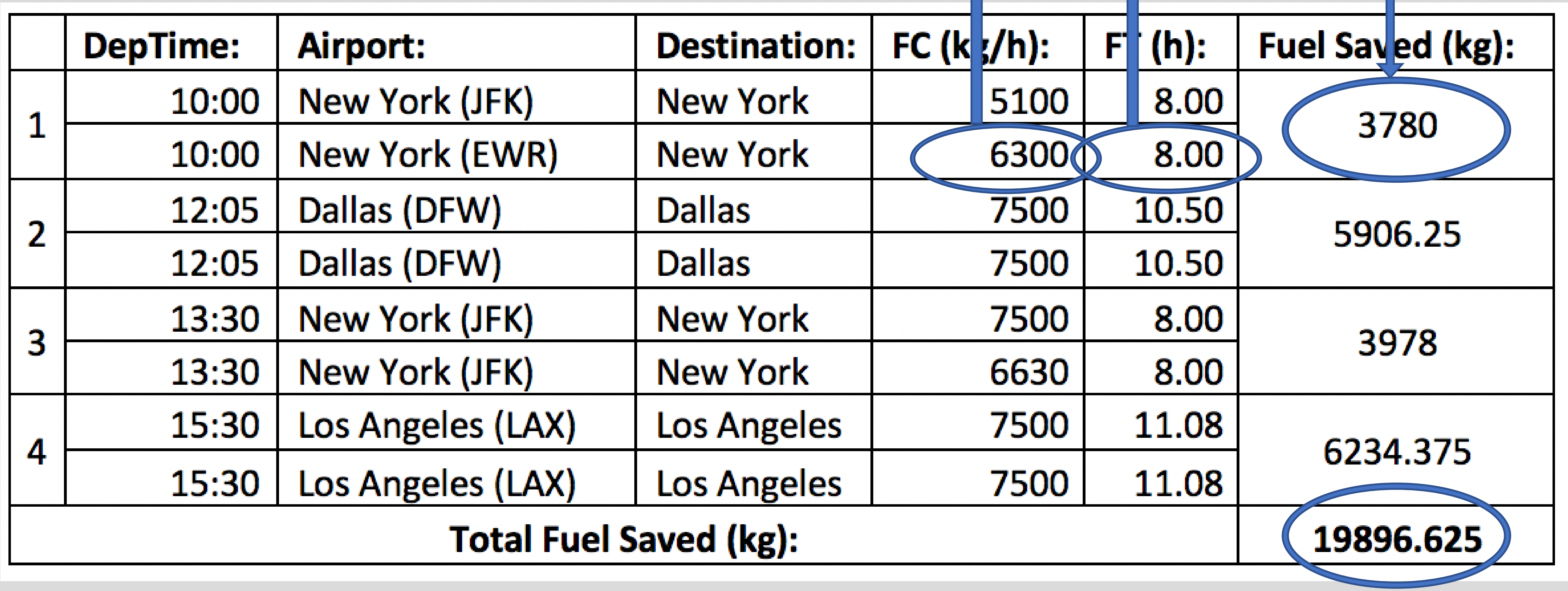
AIRBUS



Images from [8]

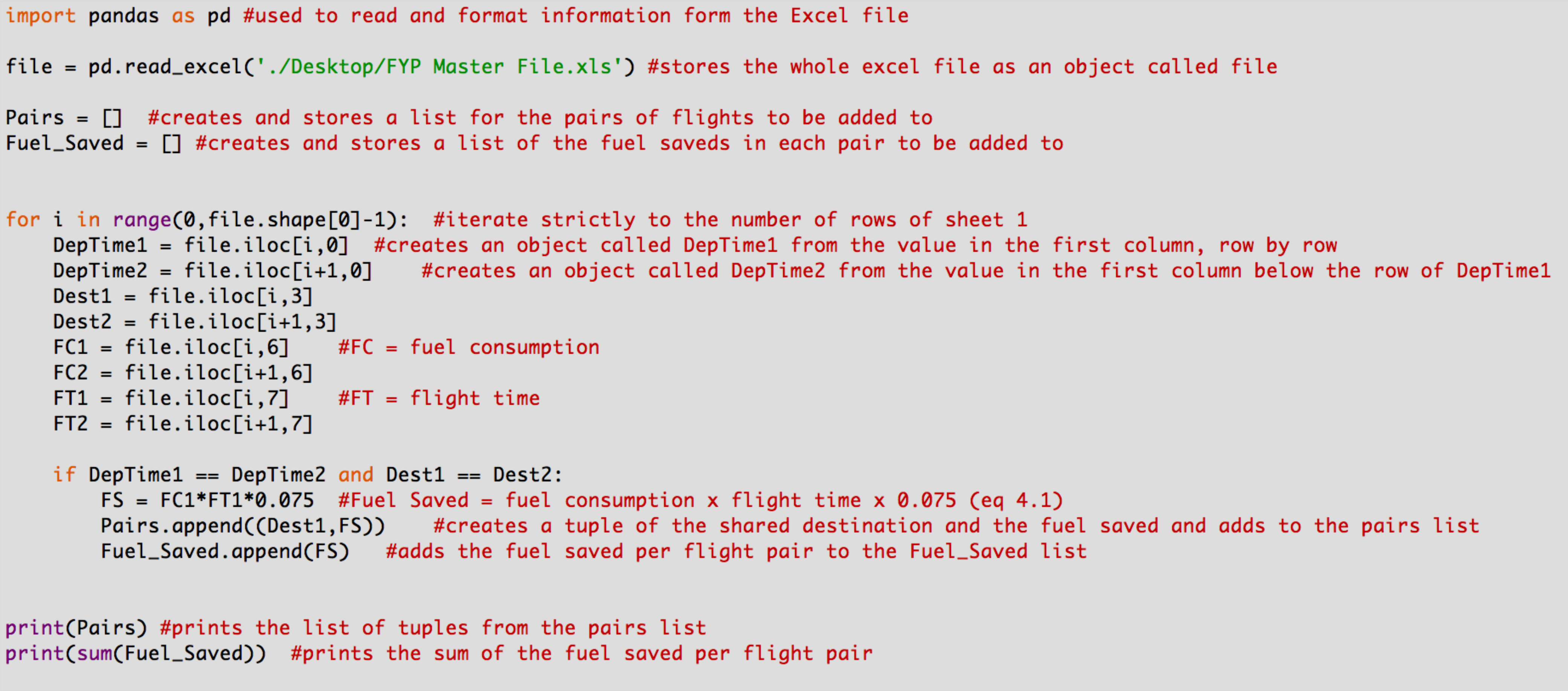
Initial Fuel Saving Calculations

$6300 \times 8.00 \times 0.075$



	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**

Initial Fuel Saving Calculations



```python
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 = FC1*FT1*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
```

Returned:

```
[('New York', 3780.0), ('Dallas', 5906.25), ('New York', 3978.0), ('Los Angeles', 6234.375)]
19896.625
```

Finding Flight Pairs

Cities/Airports → 5 pairs

- Minor fuel saving

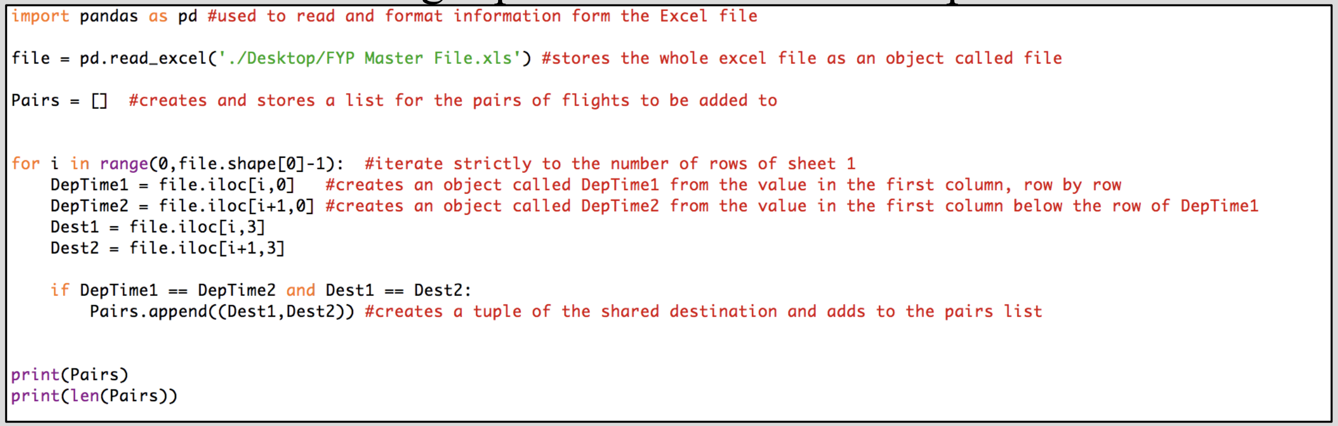
Countries→ 25 pairs

- USA destinations on different coasts would take different routes across the Atlantic

Within 1000 miles → 22 pairs

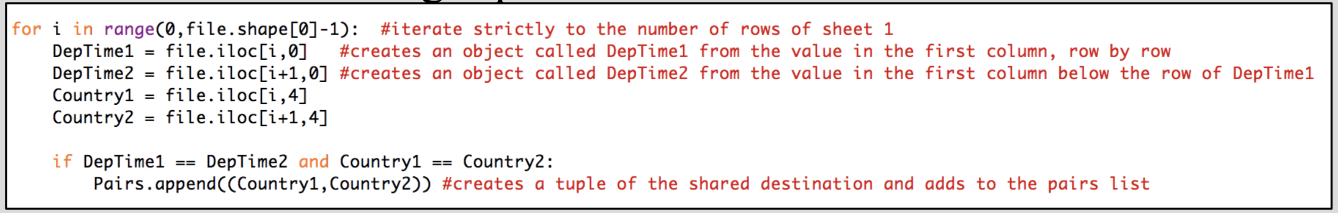
- Avoids the USA destinations on different coasts
- Allows destinations from different countries to pair

Code used to find flight pairs based on cities/airports



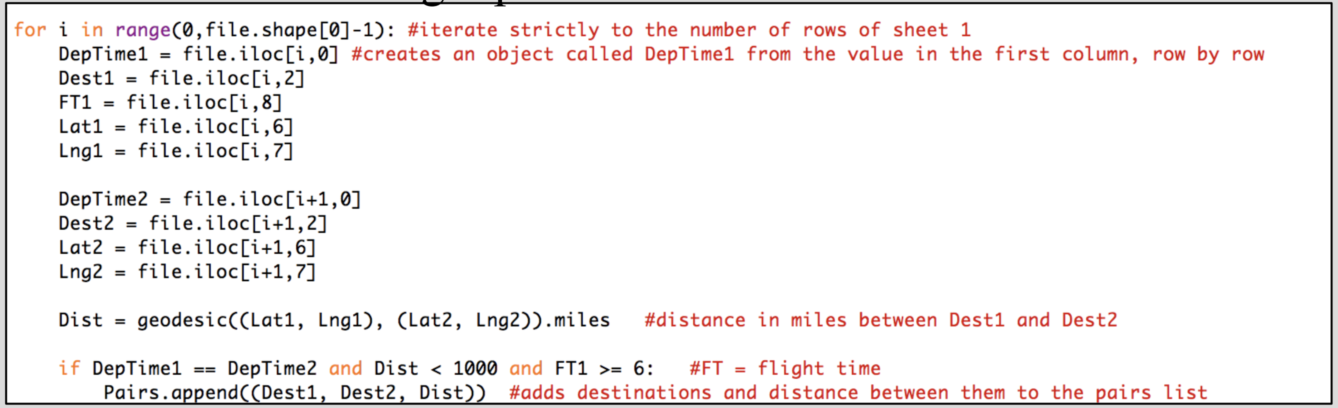
```
import pandas as pd #used to read and format information form the Excel file

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))
```

Code used to find flight pairs based on countries



```
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
```

Code used to find flight pairs within 1000 miles



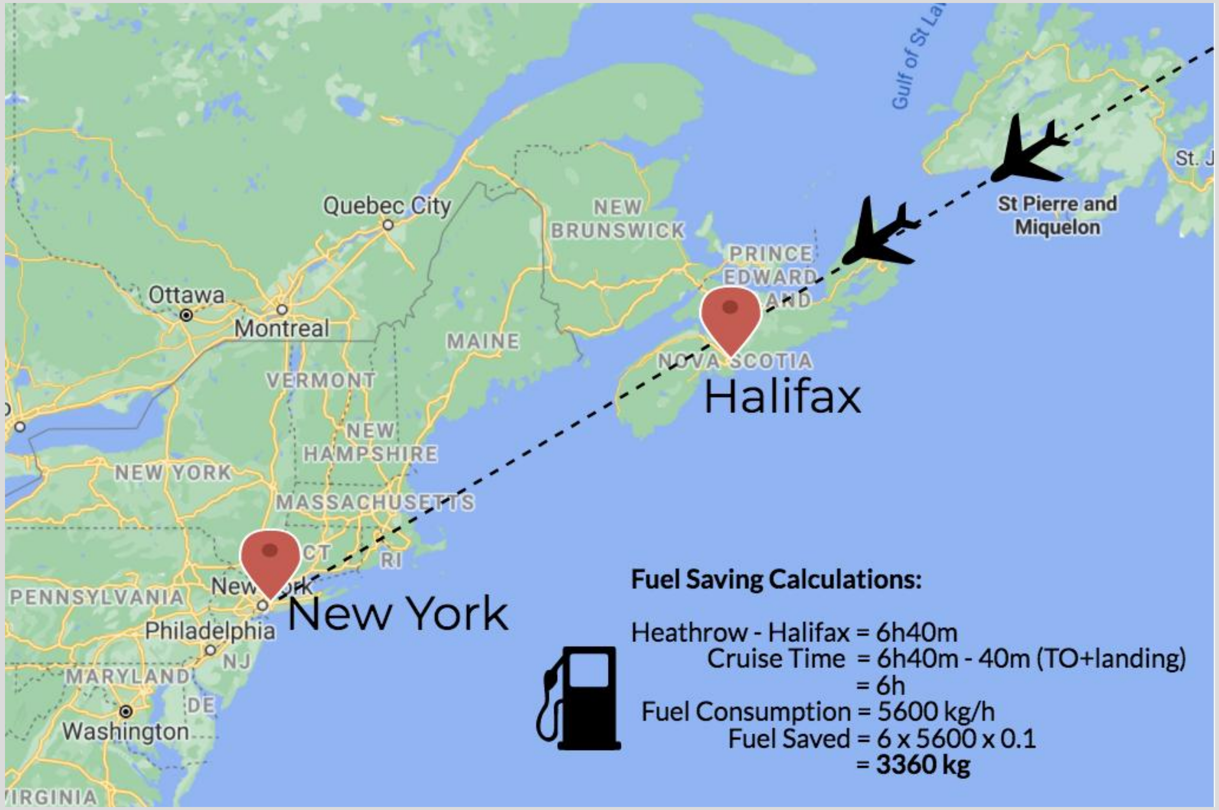
```
for i in range(0,file.shape[0]-1): #iterate strictly to the number of rows of sheet 1
    DepTime1 = file.iloc[i,0] #creates an object called DepTime1 from the value in the first column, row by row
    Dest1 = file.iloc[i,2]
    FT1 = file.iloc[i,8]
    Lat1 = file.iloc[i,6]
    Lng1 = file.iloc[i,7]

    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
```

Final Calculations

Scenario A → 5-10% of the overall fuel consumed in the journey → **68, 351.14kg**

Scenario B → 5-10% of the fuel consumed while the aircraft are in formation

‘Drop Off’



Fuel Saving Calculations:

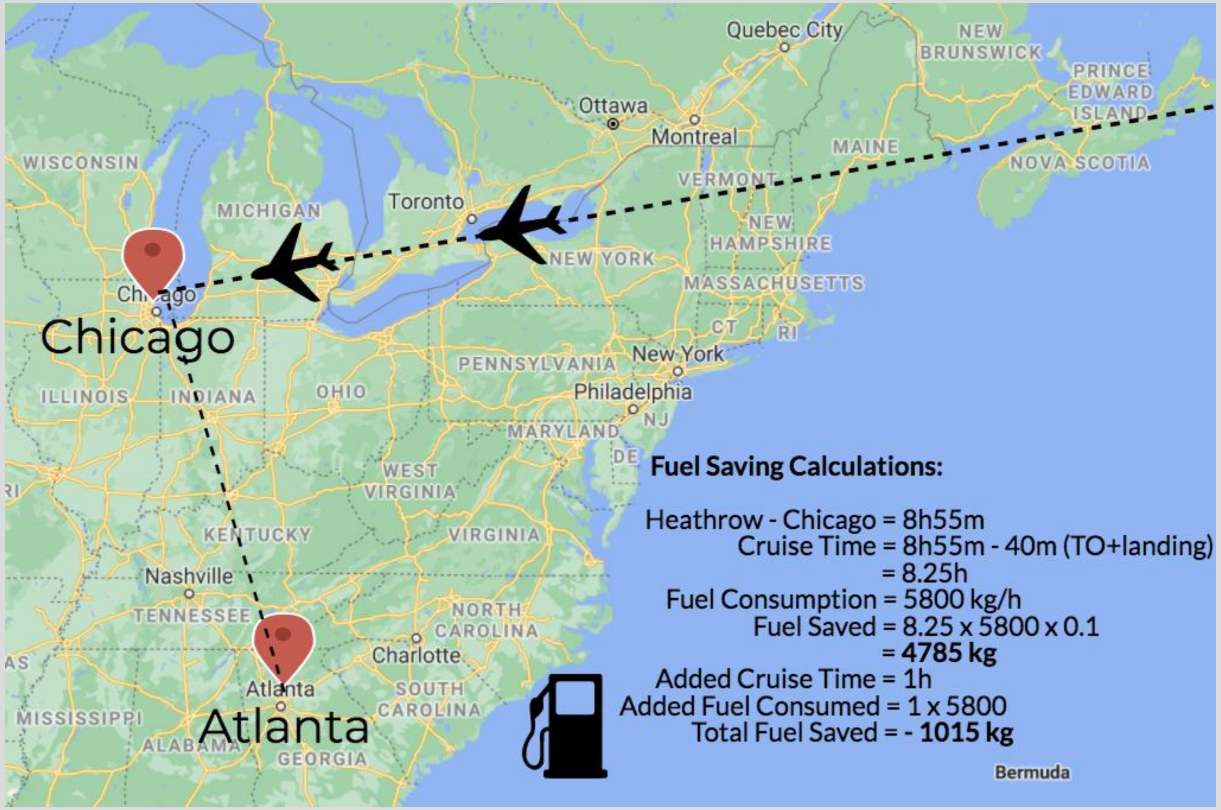
Heathrow - Halifax = 6h40m
Cruise Time = 6h40m - 40m (TO+landing)
= 6h
Fuel Consumption = 5600 kg/h
Fuel Saved = 6 x 5600 x 0.1
= **3360 kg**

Final Calculations

Scenario A → 5-10% of the overall fuel consumed in the journey → **68, 351.14kg**

Scenario B → 5-10% of the fuel consumed while the aircraft are in formation → **51, 809.82kg**

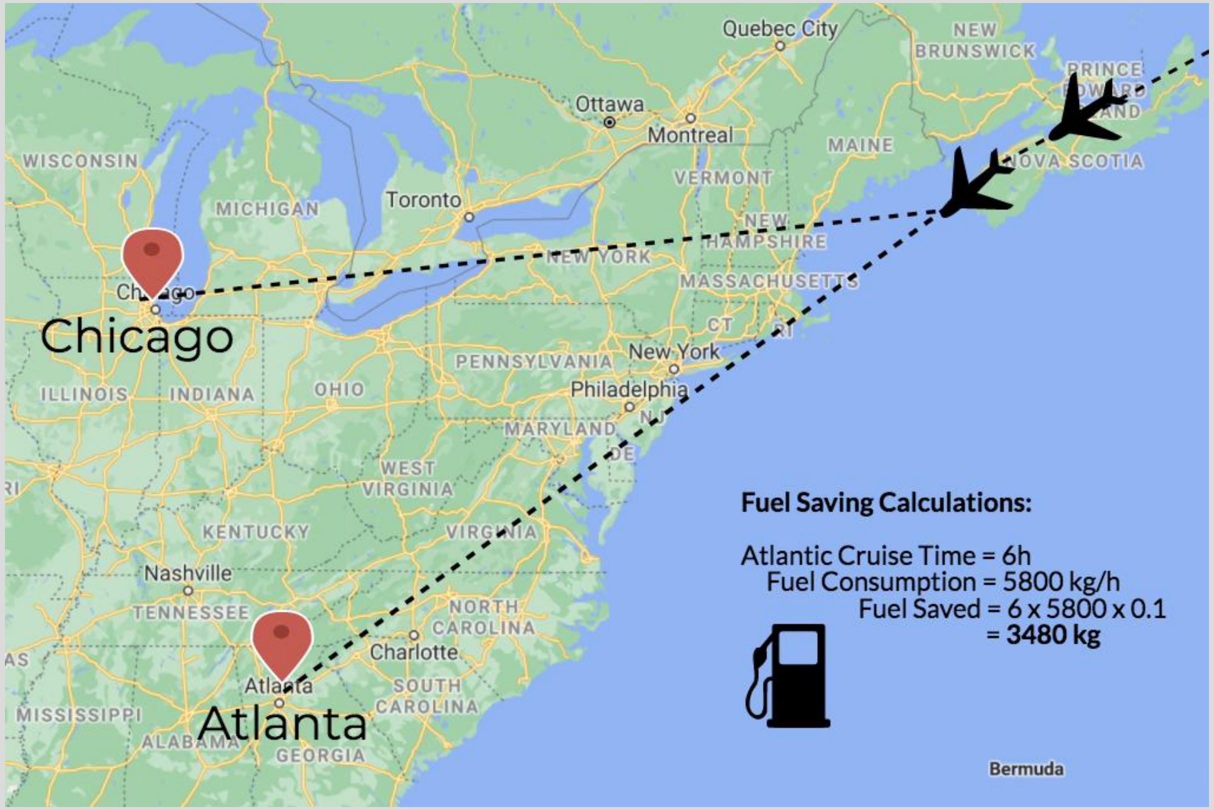
‘Diversion’



Fuel Saving Calculations:

Heathrow - Chicago = 8h55m
Cruise Time = 8h55m - 40m (TO+landing)
= 8.25h
Fuel Consumption = 5800 kg/h
Fuel Saved = 8.25 x 5800 x 0.1
= **4785 kg**
Added Cruise Time = 1h
Added Fuel Consumed = 1 x 5800
Total Fuel Saved = **- 1015 kg**

‘Split’



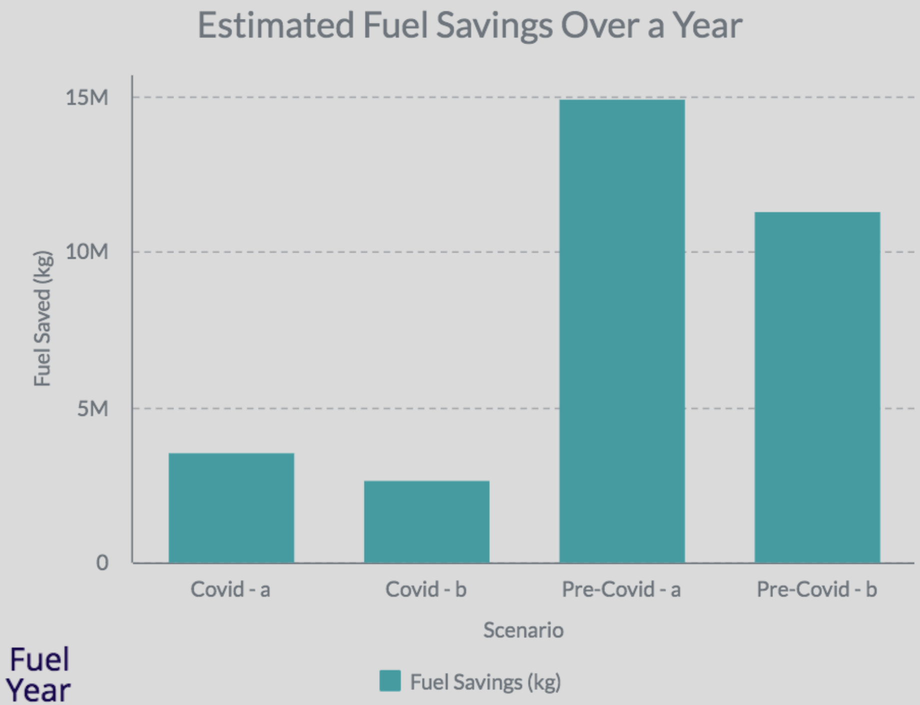
Fuel Saving Calculations:

Atlantic Cruise Time = 6h
Fuel Consumption = 5800 kg/h
Fuel Saved = 6 x 5800 x 0.1
= **3480 kg**

Results and Discussion

Travel restrictions due to the COVID-19 pandemic resulted in:

- Daily departures are less than a quarter of what they were pre-covid
- Limited flights to Asia


Estimated Fuel Savings Over a Year

Scenario	Departures Per Day (Average)	Estimated Fuel Saved Per Week (Average) (kg)	Estimated Fuel Saved Per Year (Average) (kg)
Covid - a During covid, assuming 7.5% fuel saved of overall fuel consumption	154	68,351.14	3,554,259.34
Covid - b During covid, assuming 7.5% fuel saved of fuel consumed while in formation	154	51,809.82	2,694,110.64
Pre-Covid - a Pre-covid, assuming 7.5% fuel saved of overall fuel consumption	650	287,428.66	14,946,284.66
Pre-Covid - b Pre-covid, assuming 7.5% fuel saved of fuel consumed while in formation	650	217,869.39	11,329,208.33

Delays:

- 15 minute window
- $74.20 USD per minute

Results and Discussion

	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

Conclusions

- Flight leaving Heathrow at the same time, heading in the same direction could adopt Airbus' proposed wake-energy retrieval strategy and roughly save between 51, 809.82 - 68, 351.14 kg of fuel in one week while travel restrictions were in place.

- Without travel restrictions fuel savings would be much greater. Estimated fuel savings using the daily average number of flights pre-covid were between 217, 869.39-287, 428.66kg.

- Without travel restrictions further fuel savings could be made as there would be more flights travelling to south east Asia, where many destinations are close together and flight times are greater than 10 hours so wake-energy retrieval could be utilised for a long time.

- Results calculated in the report do have a large margin of error, however the potential savings of fuel and therefore reduction in CO_2 emissions are significant.

Future Work

Account for Delays

- Further calculations to determine if the cost of a delay is more or less the that of the fuel saved.
- Change code to find flight pairs leaving within a 15 minute window.
- Change code to find flights leaving in the next 15 minute window in case of delays.

Airlines Sharing Aircraft

- Reduced fuel consumption from less taxiing

Image from [10]

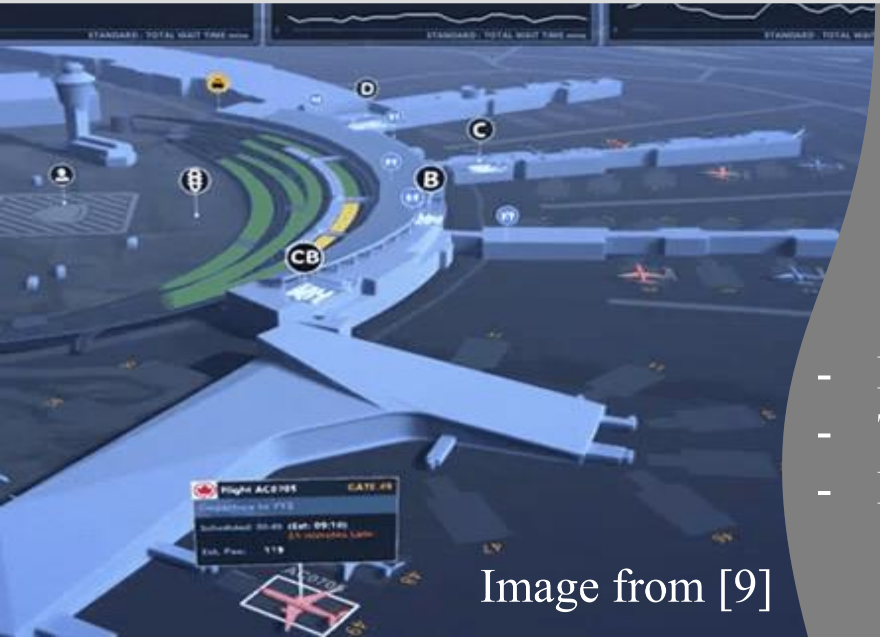


Image from [9]

Digital Twin of Heathrow Airport

- Run simulations
- Temperature control
- Efficient taxi routes

Lorries to Utilise Slipstream

“According to research published last year by the University of Aachen, Germany, **lorries** can save around **17 per sent in fuel** by **tailgating** each other.” [11]

References

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

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

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

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

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

[6] 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.

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

[8] 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].

[9] 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].

[10] Drum, B., 2016. *Airline Tail Logos*. [online] Airline Tail Logos. Available at: <https://airlinetaillogos.wordpress.com/> [Accessed 20 May 2021].

[11] 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].