UNITED AIRLINES

Marc Bitar, Suren Dharmasena, Austin Douthit, Chris Lee, Jack Casey

Building a Foundation

Airline Industry Outlook:

- This is an industry that has been undergoing many changes following the shift in demand that the COVID-19 Pandemic brought as well as the shift in oil supply that the war in Ukraine has created

There are 3 considerations an airline needs to consider if it wants to establish itself at an airport

1. Locations to consider
2. Fixed hangar costs
3. Routes to reach destination

As the world continues to open up following the endemic, airline companies are required to navigate through this shifting market to meet the demand of the consumers

Context of Our Application:

The application involves helping an airline determine where to set up hangars at several different airports in order to minimize the cost of connecting flights between California (LAX and SFO) and New York (LGA and EWR).

The model involves us crafting a network flow model: airports as the nodes and flights connecting the hubs as arcs. Further characterizing this application, the problem will involve the following constraints: hangar costs, allocating resources, and fulfilling node demand – all of which are further detailed below.

Demand: LAG - 516 incoming flights, EWR - 356 incoming flights

Supply: SFO - 467 outgoing flights, LAX - 437 outgoing flights

Network Model Key:

'ATL' : Atlanta, Georgia

'DEN': Denver, Colorado,

'DFW': Dallas Fort Worth, Texas

'DTW': Detroit, Michigan

'EWR': Newark ,New Jersey

'LAX': Los Angeles, California

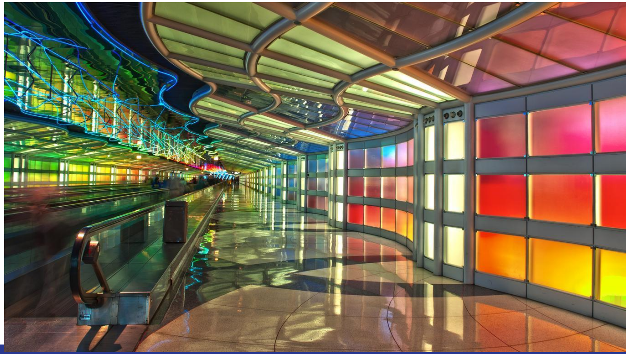
'LGA': Queens, New York

'ORD': Chicago O'Hare,

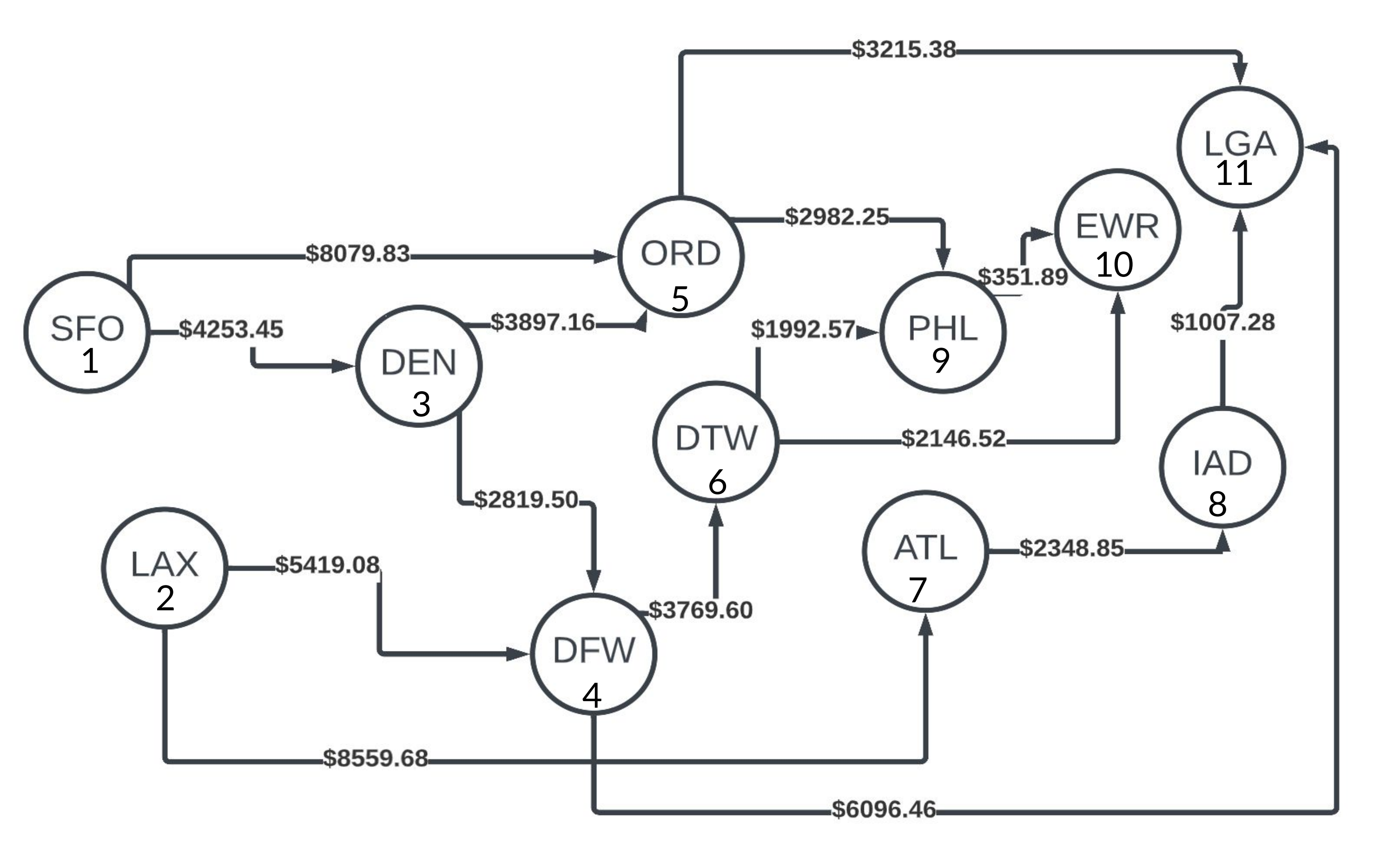
'PHL': Philadelphia, Pennsylvania

'SFO': San Francisco

'IAD': Dulles, Virginia



Network Model Diagram



High-Level Objective and Constraints:

Objective: Minimize Cost

Subject to:

- Node Requirements
- Capacity/Linking Constraints
- Mixed Integer:
  - Pay the one-time hangar fee at an airport (Binary)
  - How many flights fly through each airport (Integer)

Dataset

Fuel Cost	TO:								
FROM:	DEN	DFW	ORD	DTW	ATL	PHL	IAD	EWR	LGA
SFO	$4,253.45		$8,097.83						
LAX		$5,419.08			$8,559.68				
DEN		$2,819.50	$3,897.16						
DFW				$3,769.60					$6,096.46
ORD						$2,982.25			$3,215.38
DTW						$1,992.57		$2,146.52	
ATL							$2,348.85		
PHL								$351.89	
IAD									$1,007.28

Fixed Costs	
FROM:	Cost per Airport
DEN	$100 Million
DFW	$130 Million
ORD	$215 Million
DTW	$80 Million
ATL	$150 Million
PHL	$90 Million
IAD	$105 Million

Supply (Daily Flights)	
FROM:	Flights
SFO	467.00
LAX	537.00

Demand (Daily Flights)	
TO:	Flights
EWR	356.00
LGA	416.00

Assumptions

Hangar Cost: An estimation determined by state tax rate and demand for airport

Fuel Cost: It costs $4 per mile flown

Big M: Determined from public information shared of by the airport

Demand: Based off the average number of daily flights flown out of demand nodes from June-August

Supply: Based off the average number of daily flights flown out of supply nodes from June-August

Defining Dictionaries

Supply and Demand Node Dictionaries:

- Key: The name of the node
- Value: The corresponding supply/demand

Multidictionary:

- List: A tuple that outlines a flight location
- Dictionary:
  - Key: Tuple that outlines a flight starting/ending location
  - Value: The fuel cost associated with route

Multidictionary:

- Key: name of the node (transhipment)
- Pair: cost to purchase hangar
- Pair: maximum flights allowed to land at a node

```python
supply = dict({'SFO': 467,
               'LAX': 437})

demand = dict({'EWR': 356,
               'LGA': 516})
```

```python
arcs, fuelCost = gp.multidict({
    ('SFO', 'ORD'): 8097.83,
    ('SFO', 'DEN'): 4253.45,
    ('LAX', 'DFW'): 5419.08,
    ('DFW', 'LGA'): 6096.46,
    ('LAX', 'ATL'): 8559.68,
    ('DEN', 'ORD'): 3897.16,
    ('DFW', 'DTW'): 3769.60,
    ('DEN', 'DFW'): 2819.50,
    ('ATL', 'IAD'): 2348.85,
    ('ORD', 'PHL'): 2982.25,
    ('DTW', 'PHL'): 1992.57,
    ('DTW', 'EWR'): 2146.52,
    ('PHL', 'EWR'): 351.89,
    ('ORD', 'LGA'): 3215.38,
    ('IAD', 'LGA'): 1007.28
})

# maxFlights is bigM
airport, hangerCost, maxFlights = gp.multidict({
    'DEN':	[100_000_000, 448],
    'DFW':	[130_000_000, 460],
    'ORD':	[215_000_000, 513],
    'DTW':	[80_000_000, 429],
    'ATL':	[150_000_000, 492],
    'PHL':	[90_000_000, 426],
    'IAD':	[105_000_000, 413]
})
```

Defining Model and Decision Variables

Model Created for application named: “United”

Decision Variables:

Flow[$_{i,j}$]: Number of flights departing from location i and arriving to location j

- Integer Value: The number of flights taken from airport I to airport J

Hangar[$_j$]: Paying the start-up fee for having access to hangar in airport J

- Binary value: Don’t pay the start-up cost, 1 pay the start-up cost

```
model = gp.Model('United')
flow = model.addVars(arcs, vtype='I', name="flow")
hangar = model.addVars(airport, vtype='B', name='hangar')
```

Defining Objectives

Objective: Minimize the total cost

Minimizing total costs ((Fuel Cost (Ci) * Flow[$_{i,j}$])+ (Hangar Cost (HC$_j$) * Hangar[$_j$]):

- MIN (Cij * Xij) + (HCj * Hj)

```
model.setObjective(gp.quicksum((fuelCost[arc]*flow[arc] for arc in arcs))
                   + gp.quicksum((hangerCost[a]*hangar[a] for a in airport)))
```

Defining Constraints - Supply and Demand Node

Variable Names:

- Spots: A list that contains the name of the supply Node
- Planes: A list that contains the name of demand Node

Constraint Codes:

- Inflow - Outflow <= Supply
  - Since total Supply exceed > Total Demand

```
spots = supply.keys()
spots_flow = model.addConstrs((gp.quicksum(flow.select(spot, '*'))
                               <= supply[spot] for spot in spots), name="supply constraint")
```

```
planes = demand.keys()
planes_flow = model.addConstrs((gp.quicksum(flow.select('*', plane)) -
                                gp.quicksum(flow.select(plane, '*')) >= demand[plane]
                                for plane in planes), name="demand constraint")
```

Defining Constraints - Linking, Capacity, & Transshipment Constraints

Transhipment Constraint: The difference from total inflow from any airport to airport J to the outflow from airport J to any airport must be greater than zero

```
model.addConstrs(gp.quicksum(flow.select('*',j)) - gp.quicksum(flow.select(j,'*')) >= 0 for j in airport)
#transhipment Constraint
```

Linking/Capacity Constraints: If the start-up binary hangar cost is paid, than United is able to fly into that airport

- If not paid, then the flow into the airport is zero
- Also includes capacity constraints, flow into the airport cannot exceed the maximum capacity at said airport

```
model.addConstr((flow[('SFO', 'DEN')] - maxFlights['DEN']*hangar['DEN'] <= 0), name = 'linking')
model.addConstr((flow[('LAX', 'DFW')] + flow[('DEN', 'DFW')] - maxFlights['DFW']*hangar['DFW'] <= 0), name = 'linking')
model.addConstr((flow[('SFO', 'ORD')] + flow[('DEN', 'ORD')] - maxFlights['ORD']*hangar['ORD'] <= 0), name = 'linking')
model.addConstr((flow[('DFW', 'DTW')] - maxFlights['DTW']*hangar['DTW'] <= 0), name = 'linking')
model.addConstr((flow[('DTW', 'PHL')] + flow[('ORD', 'PHL')] - maxFlights['PHL']*hangar['PHL'] <= 0), name = 'linking')
model.addConstr((flow[('LAX', 'ATL')] - maxFlights['ATL']*hangar['ATL'] <= 0), name = 'linking')
model.addConstr((flow[('ATL', 'IAD')] - maxFlights['IAD']*hangar['IAD'] <= 0), name = 'linking')
```

LP Model Formulation

```
\ Model United
\ LP format - for model browsing. Use MPS format to capture full model detail.
Minimize
  8097.83 flow[SFO,ORD] + 4253.45 flow[SFO,DEN] + 5419.08 flow[LAX,DFW]
   + 6096.46 flow[DFW,LGA] + 8559.68 flow[LAX,ATL] + 3897.16 flow[DEN,ORD]
   + 3769.6 flow[DFW,DTW] + 2819.5 flow[DEN,DFW] + 2348.85 flow[ATL,IAD]
   + 2982.25 flow[ORD,PHL] + 1992.57 flow[DTW,PHL] + 2146.52 flow[DTW,EWR]
   + 351.89 flow[PHL,EWR] + 3215.38 flow[ORD,LGA] + 1007.28 flow[IAD,LGA]
   + 1e+08 hangar[DEN] + 1.3e+08 hangar[DFW] + 2.15e+08 hangar[ORD]
   + 8e+07 hangar[DTW] + 1.5e+08 hangar[ATL] + 9e+07 hangar[PHL]
   + 1.05e+08 hangar[IAD]
Subject To
 supply_constraint[SFO]: flow[SFO,ORD] + flow[SFO,DEN] <= 467
 supply_constraint[LAX]: flow[LAX,DFW] + flow[LAX,ATL] <= 437
 demand_constraint[EWR]: flow[DTW,EWR] + flow[PHL,EWR] >= 356
 demand_constraint[LGA]: flow[DFW,LGA] + flow[ORD,LGA] + flow[IAD,LGA]
   >= 516
 R4: flow[SFO,DEN] - flow[DEN,ORD] - flow[DEN,DFW] >= 0
 R5: flow[LAX,DFW] - flow[DFW,LGA] - flow[DFW,DTW] + flow[DEN,DFW] >= 0
 R6: flow[SFO,ORD] + flow[DEN,ORD] - flow[ORD,PHL] - flow[ORD,LGA] >= 0
 R7: flow[DFW,DTW] - flow[DTW,PHL] - flow[DTW,EWR] >= 0
 R8: flow[LAX,ATL] - flow[ATL,IAD] >= 0
 R9: flow[ORD,PHL] + flow[DTW,PHL] - flow[PHL,EWR] >= 0
 R10: flow[ATL,IAD] - flow[IAD,LGA] >= 0
 linking: flow[SFO,DEN] - 448 hangar[DEN] <= 0
 linking: flow[LAX,DFW] + flow[DEN,DFW] - 460 hangar[DFW] <= 0
 linking: flow[SFO,ORD] + flow[DEN,ORD] - 513 hangar[ORD] <= 0
 linking: flow[DFW,DTW] - 429 hangar[DTW] <= 0
 linking: flow[ORD,PHL] + flow[DTW,PHL] - 426 hangar[PHL] <= 0
 linking: flow[LAX,ATL] - 492 hangar[ATL] <= 0
 linking: flow[ATL,IAD] - 413 hangar[IAD] <= 0
Bounds
Binaries
 hangar[DEN] hangar[DFW] hangar[ORD] hangar[DTW] hangar[ATL] hangar[PHL]
 hangar[IAD]
Generals
 flow[SFO,ORD] flow[SFO,DEN] flow[LAX,DFW] flow[DFW,LGA] flow[LAX,ATL]
 flow[DEN,ORD] flow[DFW,DTW] flow[DEN,DFW] flow[ATL,IAD] flow[ORD,PHL]
 flow[DTW,PHL] flow[DTW,EWR] flow[PHL,EWR] flow[ORD,LGA] flow[IAD,LGA]
End
```

What the Solution Means

Flow[i,j] = The number of flights going through path i to J

hangar[j] = If the one-time fee was paid to establish a hangar

Optimal Value: Is the total minimized cost that fulfills all the constraints

Variable	X
flow[SFO,ORD]	436
flow[LAX,DFW]	436
flow[DFW,LGA]	80
flow[DFW,DTW]	356
flow[DTW,EWR]	356
flow[ORD,LGA]	436
hangar[DFW]	1
hangar[ORD]	1
hangar[DTW]	1

Optimal Value: 434889133.96

Solution outlined within the network model