LECTURE NOTES
ON

FLIGHT SCHEDULING AND OPERATIONS

IV B. Tech I semester (JNTUH-R15)

PREPARED BY
Ms. M.SNIGDHA
Assistant Professor

AERONAUTICAL ENGINEERING
INSTITUTE OF AERONAUTICALENGINEERING
(Autonomous)
DUNDIGAL, HYDERABAD - 500043

UNIT I
NETWORK FLOWS AND INTEGER PROGRAMMING MODELS

Computational Complexity, Case Studies of Airline Operations and Scheduling and Simulation

Introduction:

The case studies and examples presented in the previous chapters where integer linear programming models were adopted could be easily solved using optimization software. This was mainly due to a relatively small number of decision variables and/or constraints. The objective of the previous cases and examples was to introduce the development of mathematical models rather than solve the problems. We assumed that once the problem is formulated, we could use software to obtain the optimal solution. Unfortunately, the problems that many airlines face involve millions or even billions of decision variables. These huge models cannot be solved using the standard software package. Consider the following example. A cargo airline serves 30 cities within its network with a single fleet-type aircraft. Each route on average consists of 7 flights per day. (Aircraft Routing), in order to find the optimum solution we need to list all possible routes. Even for the small Ultimate Air case we had more than6, 000 decision variables for aircraft routing for the 737-800 fleet.

Complexity Theory:

- Since the emergence of real-world problems and their solution methodologies, there has been a constant need to classify and compare them on the basis of the computational tractability.
- Specifically, we are interested in knowing how the computational times increase as the size of the problem grows. As an example, i fit takes 5 seconds to solve a linear program with 100 variables and 50 constraints using a specific software package, how long would it take to solve a similar model with 1,000 variables and 500 constraints on the same computer and using the same software? Will it take ten times as much time to solve the problem or more?
- Therefore, instead of making computational time dependent on the computer we would like to use other algorithms that present a general overview of the complexity of the problem, as the size of the problem grows.
- Assume that you have a sequence of *n* numbers to sort from smallest to largest. One method (algorithm) for sorting is by comparing two neighboring numbers if they are in the wrong order, they are reordered small to large.
- This process is repeated until all the numbers are in sequence. This method is called bubble sort. The best scenario is that all the numbers are already in the right order.
- In this case after *n* comparisons (steps) the algorithm tells us that the numbers are in order. Where all numbers are sorted from largest to smallest, we have to

repeat this comparison $n2$ times since every number is in the wrong order.

- We are normally interested in the worst-case scenario since it will serve as the upper bound for the number of steps to generate the solution. According to this example the bubble sort has a complexity of order of $n2$. It is shown as O ($n2$) where O stands for the order of time.
- They classified the algorithms based on their computation tractability into two groups, polynomial (P) and non-deterministic polynomial (NP) time algorithms. For polynomial time algorithms, the solution times are bounded by a polynomial order.
- As an example, our bubble sort algorithm is classified as a polynomial time of order of $n2$. Polynomial time algorithms are typically considered 'good'. This is because these algorithms can solve large instances of the problem in reasonable steps and time.
- For non- deterministic polynomial (NP) time algorithms on the other hand, as the name implies, the number of steps to solve the problem grows exponentially with the problem size.
- As an example, consider that an algorithm is classified as non- deterministic polynomial O ($2n$) order. If the problem size doubles, then the steps (times) that it takes to solve this problem will be the number of steps for the original problem to power two or O ($22n$) = O ($2n$) 2.
- These algorithms are considered 'bad' in the sense that as the problem size increases, the computational times grows very large. Many algorithms that are adopted to solve the combinatorial-type problems such as the traveling salesman have an order of complexity of O (n!).
- These problems represent the most challenging in terms of computational complexity. As the order of time implies, in these algorithms the computational complexity grows exponentially with the size of the problem.

Heuristic Procedures:

- Not being able to obtain the optimum solutions in a timely fashion for NP type algorithms prompted the researchers to develop other alternatives. These alternative solution methods are generally classified as heuristic methods.
- Heuristic methods are techniques that do not guarantee or promise the optimum solutions but attempt to provide a 'good' and sometimes 'near optimum' solution in a minimal amount of time.
- As for the airlines, they either develop their in-house customized heuristics to solve their mathematical models in their operations research departments or outsource the service

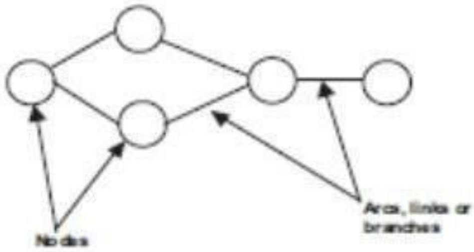
Introduction:

- A large part of the problems that airlines face can be translated into network and integer programming models.
- It should be represent as a small selection of models from the vast area of network and integer programming techniques. For a complete discussion of

various network models.

Networks:

- A network is defined as a collection of points and lines joining these points.
- There is normally some flow along these lines, going from one point to another.



2.1 Basic elements of a network

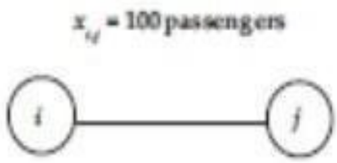
Network Terminology:

Nodes and Arcs:

- In a network, the points (circles) are called nodes and the lines are referred to as arcs, links or arrows.

Flow:

- The amount of goods, vehicles, flights, passengers and so on that move from one node to another.



Flow between two nodes.

Directed Arc:

- If the flow through an arc is allowed only in one direction, then the arc is said to be a directed arc.
- Directed arcs are graphically represented with arrows in the direction of the flow.

Directed flow

Undirected Arc:

- When the flow on an arc (between two nodes) can move in either direction, it is called an undirected arc.
- Undirected arcs are graphically represented by a single line (without arrows) connecting the two nodes.

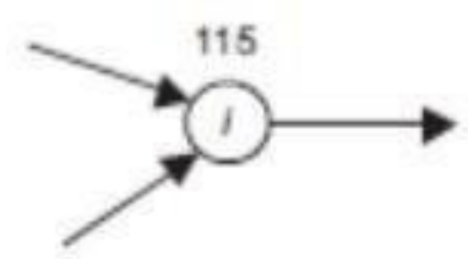
Undirected flow

Arc Capacity:

- The maximum amount of flow that can be sent through an arc.
- Examples include restrictions on the number of flights between two cities.

Supply Nodes:

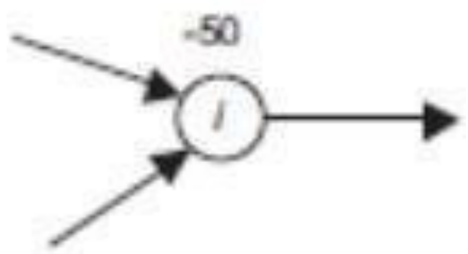
- Nodes with the amount of flow coming to them greater than the amount of flow leaving them – or nodes with positive net flow.



Supply node:

Demand Nodes:

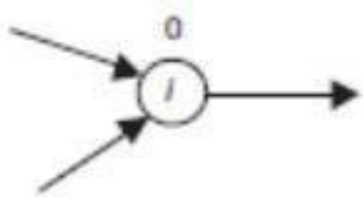
- Nodes with negative net flow or outflow greater than inflow.



Demand node:

Transshipment Nodes:

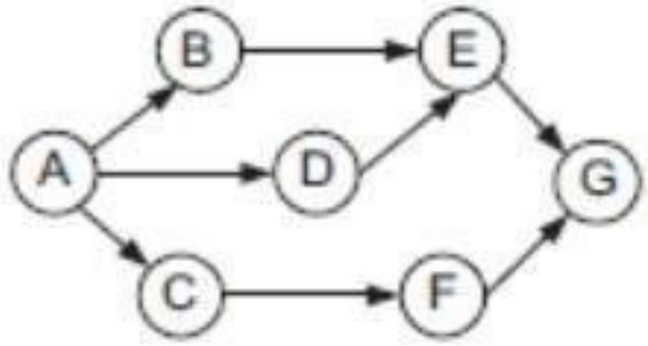
- Nodes with the same amount of flow arriving and leaving – or nodes with zero net flow.



Transshipment node:

Path:

- Sometimes two nodes are not connected by an arc, but could be connected by a sequence of arcs.
- A path is a sequence of distinct arcs that connect two nodes in this fashion.
- Airliners utilize hubs to provide connections between city pairs in their network.



A network showing three paths from Ato G

Source:

- Starting node in the path.

Destination:

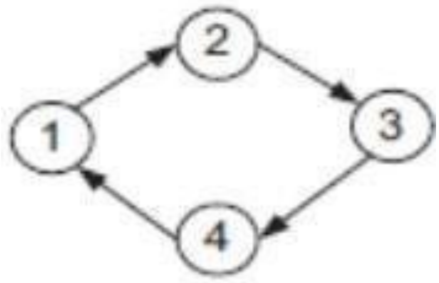
- Last node in the path.

Cycle:

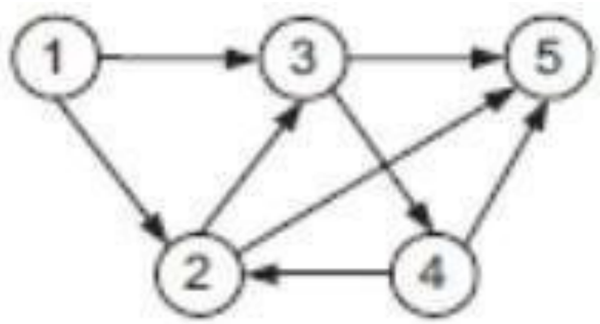
- A sequence of directed arcs that begins and ends at the same node.
- Examples include aircraft that start from an airport which is a maintenance base and, after flying to several destinations, end up at the same airport from which they departed.

Connected Network:

- A network in which every two nodes are linked by at least one path.



Cycle



Connected network

Network Flow Models:
Shortest Path (Route) Problem:

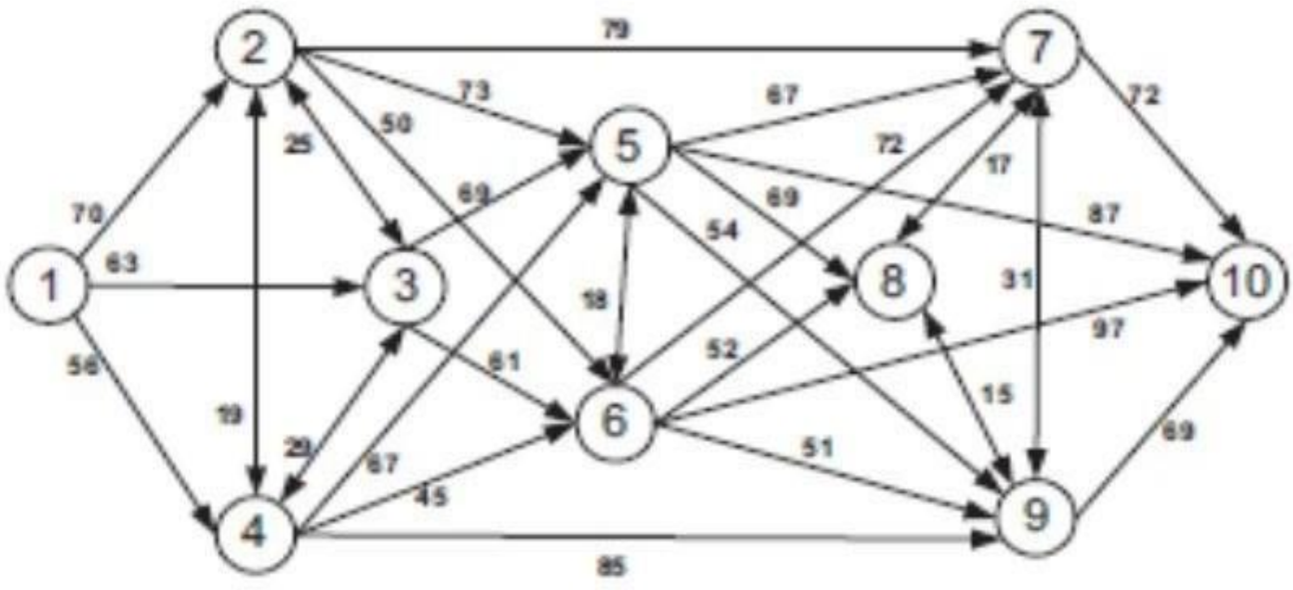
- This problem attempts to identify a path, from source to destination, within the network, that results in minimum transport time/cost.
- This particular problem should be especially attractive to cargo handlers and origin/destination.
- The problem consists of a connected network with known costs for each arc in the network.
- The objective is to identify the path with the minimum cost between two desired nodes.

Example:

- We want to determine the best route that results in the shortest flying time from node.
- The nodes represent the cities, and the arcs are the flights.
- The numbers on the arcs represent the flight time in minutes between the city pairs. We want to determine the best route that results in the shortest flying time from node. 1 (source) to node 10 (destination).
- We assume the following binary (0–1) decision variable.

Network with flight times between city pairs:

- Then the objective function is to minimize the total flying cost (time) as follows:
- We have three sets of constraints as follows:



Source node:

- The flow must originate from node 1.
- To make sure that the flow (in this case our starting flight) leaves the source we must have:

Transshipment nodes:

- Every other node (except source and destination) is a transshipment node.
- That is the net flow in these nodes should be zero.

Destination node:

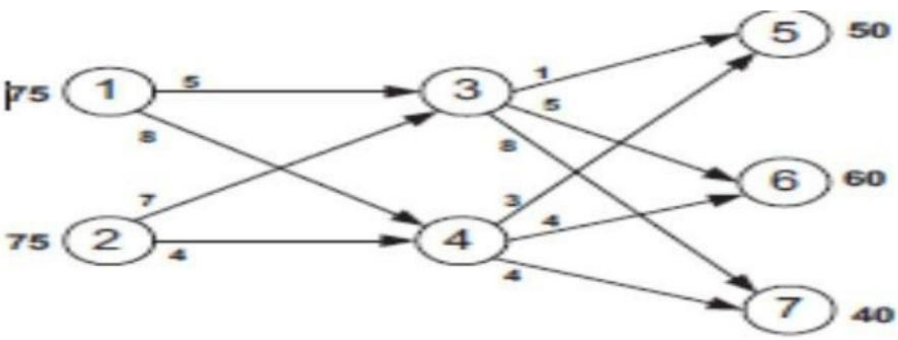
- The flow must end up at the destination node (node 10).
- The objective function attempts to minimize the total cost.
- Constrains ensures that the flow is shipped from the source (supply) node.
- The set of constraints impose that all other nodes (except the source and the destination node) are transshipment nodes.
- Finally, constraints ensure that the flow is received at the destination (demand) node.

Minimum Cost Flow Problem:

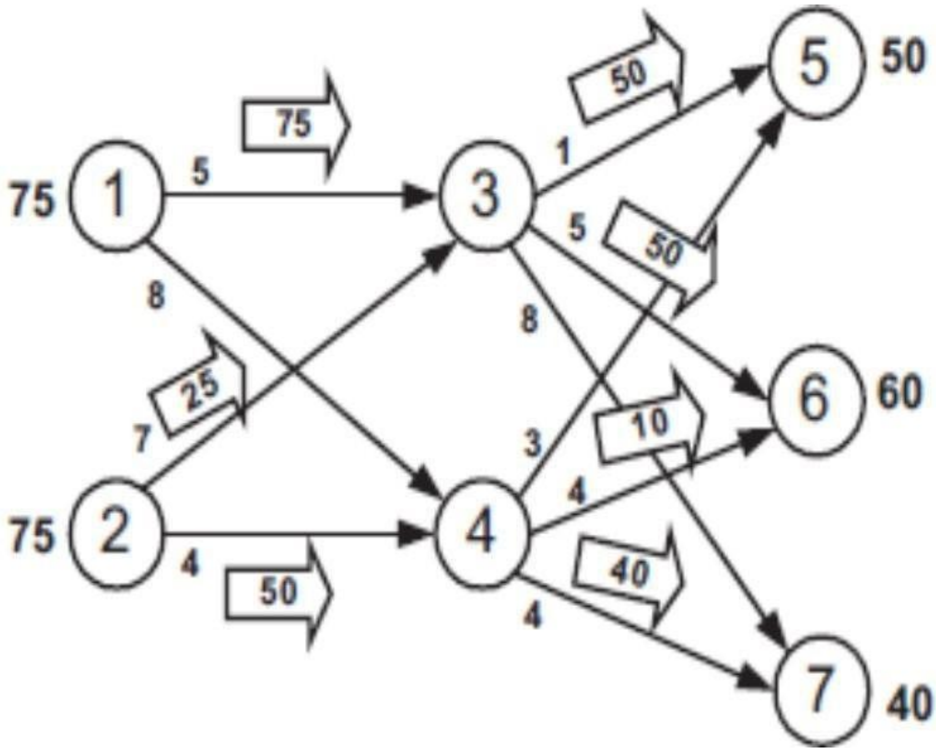
- The minimum cost flow network problem seeks to satisfy the requirements of nodes at minimum cost.
- This is a generalized form of transportation, transshipment, and shortest path problems.
- This problem assumes that we know the cost per unit of flow and capacities associated with each arc.

Example:

- An airline is tasked with transporting goods from nodes 1 and 2 to nodes 5, 6 and 7.
- The airline does not have direct flights from the source nodes to the destination nodes.
- Instead, they are connected through its hubs in nodes 3 and 4.
- The numbers next to the nodes represent the demand/supply in tons.
- The numbers on the arcs represent the unit cost of transportation per ton.
- We want to determine the best way to transport the goods from sources to destinations so that the total cost is minimized.
- The aircraft flying to and from node 4 can carry a maximum of 50 tons of cargo.



- Similarly, we write constraints for the other six nodes.
- Note that the net flow for nodes 3 and 4 should be zero as these are transshipment nodes.
- All the flights to and from node 4 can carry a maximum of 50 tons.
- Therefore, all the flow to and from this node must be limited to 50 as follows:



Solution to minimum cost flow:

- The objective function attempts to minimize the total cost of the network.
- Constraints satisfy the requirements of each node by determining the amount of inflow and outflow from that node.
- The set of constraints impose the lower and upper-bound restrictions along the arcs.

Maximum Flow Problem:

- The Maximum Flow problem is a special case of the Minimum Cost flow problem.
- It attempts to find the maximum amount of flow that can be sent from one node (source node) to another (destination node) when the network is capacitated.

Example

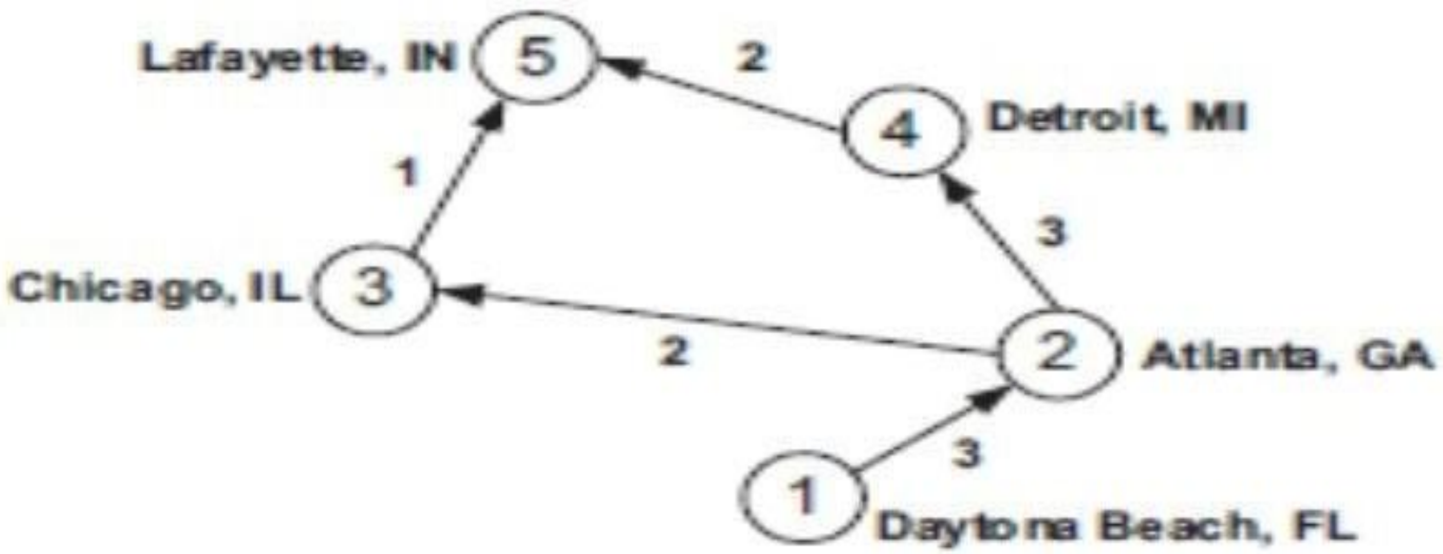
- This example is adapted from Winston and Venkataramanan (2003).
- An airline must determine the number of daily connecting flights that can be arranged between Daytona Beach (DAB), Florida, and Lafayette (LAF), Indiana.

- Connecting flights must stop in Atlanta (ATL), Georgia, and then make one more stop in Chicago (ORD), Illinois, or Detroit (DTW), Michigan. Owing to its current policies with these airports, the airline has a maximum number of daily flights which it can operate between the city pairs.

Maximum number of flights per city-pair for Shuttle Hopper Airways

City-Pairs	Maximum number of daily flights
DAB - ATL	3
ATL - ORD	2
ATL - DTW	3
ORD - LAF	1
DTW - LAF	2

- The airline wants to determine how to maximize the number of connecting flights daily from Daytona Beach, FL, to Lafayette, IN, respecting the current restrictions.
- The following network represents this problem with arcs showing maximum daily flights along the city pairs.



Network presentations from source to destination

- To formulate the problem, let us assume the following decision variables:
- $x_{i,j}$= Number of flights (integer) from node *i* to node *j*
- *f* = Number of daily flights from DAB to LAF
- In this problem, the objective is to maximize the daily flights between DAB and LAF. Therefore:
- Maximize *f*
- Similar to the Shortest Path Problem, we have a set of constraints for source, transshipment and destination nodes:
- **Source node**: DAB is our source node. *f* is the total flow leaving DAB, therefore:
  $X1, 2 = f$
- **Transshipment nodes**: We write one constraint for each transshipment node. For example, for node 2 (ATL) we have:
  $x1, 2 - x2, 3 - x2, 4 = 0$
- Similarly we write transshipment constraints for other nodes 3 and4. **Destination node**: The same number of daily flights *f* departing from DAB should now arrive at destination node LAF.
  $x\ 4{,}5 + x3{,}5 = f$
- **Arc capacity**: The last set of constraints address the capacity of arcs as follows:

Solving this problem generates a maximum flow of three daily flights between DAB and LAF as follows:

1 Flight assigned to the DAB-ATL-ORD-LAF route and;

2 Flights assigned to the DAB-ATL-DTW-LAF route.

The general model is mathematically expressed as follows (Ahuja et al. 1993):

Sets

M = Set of nodes

Index

i,j,k= Index for nodes

Parameters

$L_{i,j}$= Lower bound on flow through arc (*i,j*) $U_{i,j}$= Upper bound on flow through arc (*i,j*) *m* = Destination node

Decision Variables:

$x_{i,j}$= Amount of flow from node *i* to node *j*

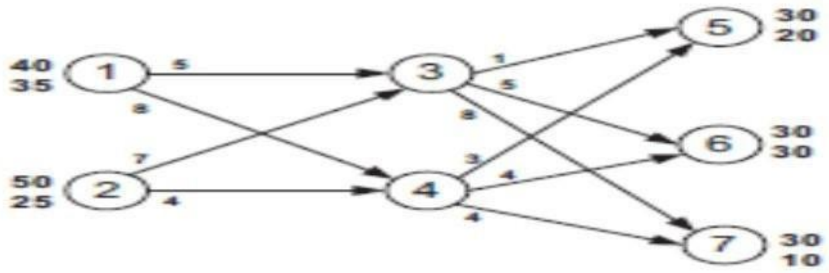
f = Amount of flow from source node to destination node

Objective Function Maximize f Subject to

- The objective function attempts to maximize flow from the source node (node 1) to the destination node (node *m*).
- The set of constraints impose the outflow and inflow restrictions on the source and destination nodes. All other nodes are transshipment nodes.
- The set of constraints imposes this restriction.
- Finally, constraints restrict the flow along the arcs based on the imposed capacity.

Multi-Commodity Problem:

- All the network models explained so far assume that a single commodity or type of entity is sent through a network. Sometimes a network can transport different types of commodities.
- The multi-commodity problem seeks to minimize the total cost when different types of goods are sent through the same network.
- The commodities may either be differentiated by their physical characteristics, or simply by certain attributes.
- The multi-commodity problem is extensively used in transportation industry. In the airline industry, the multi-commodity model is adopted to formulate crew pairing and fleet assignment models.



Network presentation for multi-commodity problem

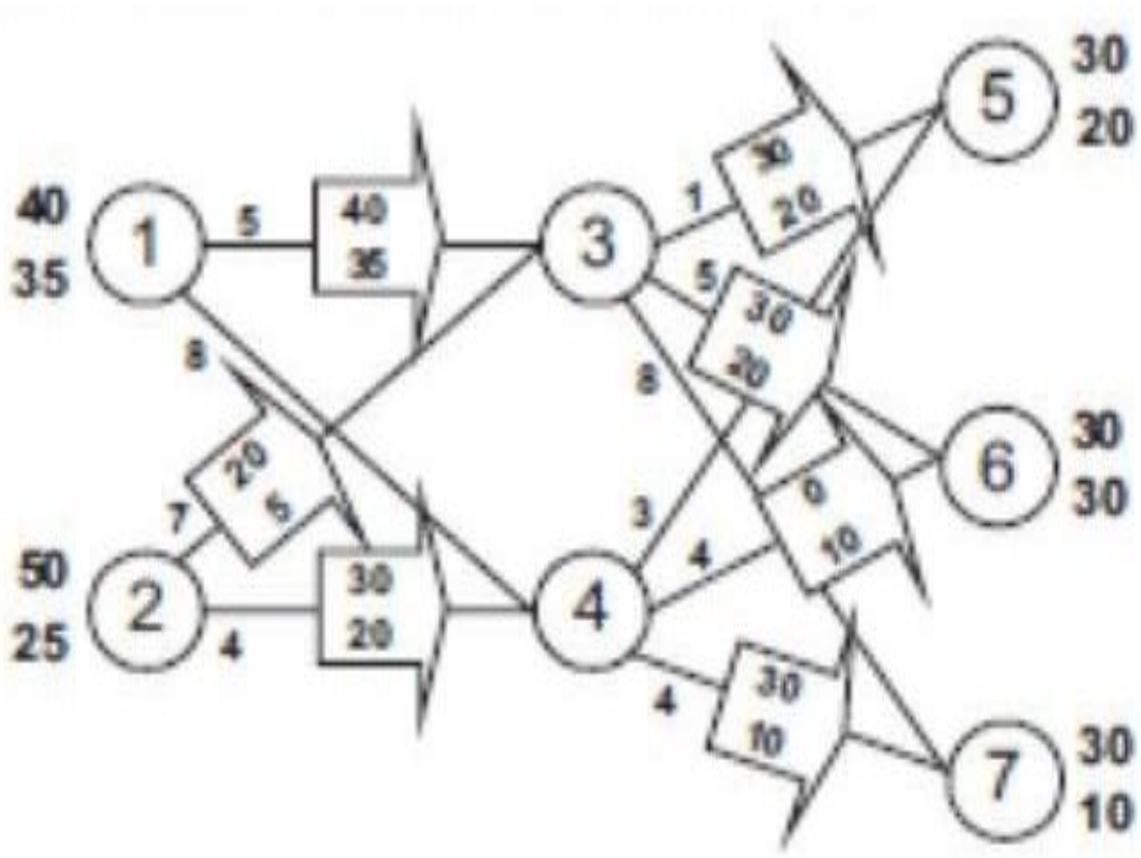
Example

- We modify the example that was presented for the Minimum Cost Flow problem discussed earlier to address the multi-commodity model formulation.
- The only difference is that instead of having only one type of cargo, in this case we have two types (two commodities).
- The numbers next to each node represent the supply/demand for each cargo at that node.
- As an example, node 1 supplies 40 and 35 tons of cargo 1 and 2 respectively.
- The transportation costs per ton are also similar.
- We want to determine how much from each cargo should be routed on each arc so that the total transportation cost is minimized.

- To formulate this problem we assume the following decision variable:
- $x_{i,j,k}$= Amount of flow from node *i* to node *j* for commodity *k* In this decision variable the indices *i* and *j* represent the nodes (*i*,*j*= 1,.,7) and represents the type of commodity (*k* = 1,2).
- The objective function is therefore:
- *Minimize*5*x*1,3,1
- + 5*x*1, 3, 2 + 8*x*1, 4, 1 + 8*x*1, 4, 2 + ...We need to write one constraint for each node. For example, for node 1 we have:
- We write similar constraint for the other six nodes.
- Recall that all the flights to and from node 4 can carry a maximum of 50 tons.

Therefore:

- Solving this problem using software generates a total minimum cost of$1,150.



Solution to multi-commodity problem:

The general model is mathematically expressed.

Sets

M = Set of nodes

K = Set of commodities

Indices

i ,j = Index for nodes

k = Index for commodities

Parameters

$c_{i,j,k}$= Unit cost of flow from node i to node j for commodity k, $b_{i,k}$= Amount of supply/demand at node i for commodity k $U_{i,j}$= Flow capacity on arc (i,j)

Decision Variable

$x_{i,j,k}$= Amount of flow from node i to node j for commodity k

- In this model, the objective function seeks to minimize the total network cost over all nodes and all commodities.
- The set of constraints satisfies the supply/demand of the node and imposes capacity constraints on the arc.

Integer Programming Models:

- Integer programming models relate to certain types of linear programming in which all of the decision variables are required to be non-negative integers.

Set-Covering/Partitioning Problems

- Set-covering problems relate to cases where each member of one set should be assigned/ matched to member(s) of another set.
- Examples include the assignment of crew members to flights, aircraft to routes, and so on.
- The objective in a set covering problem is to minimize the total cost of this assignment.

Example

- The following is an example of set-covering adapted and modified from Winston and Venkataramanan (2003).
- An airline wants to design its 'hub' system (hub-and-spoke systems.
- Each hub will be used for connecting flights to and from cities within 1,000 miles of the hub.
- The airline wants to serve the following cities:
- Atlanta, Boston, Chicago, Denver, Houston, Los Angeles, New Orleans, New York, Pittsburgh, Salt Lake City, San Francisco, and Seattle.
- The airline wants to determine the smallest number of hubs it will need in order to cover all of these cities.
- By cover, we mean each city should be within 1,000 miles of at least one hub.

Distance-matrix between cities

		1	2	3	4	5	6	7	8	9	10	11	12
		AT	BO	CH	DE	HO	LA	NO	NY	PI	SL	SF	SE
1	AT	0	1037	674	1398	789	2182	479	841	687	1878	2496	2618
2	BO	1037	0	1005	1949	1804	2979	1507	222	574	2343	3095	2976
3	CH	674	1005	0	1008	1067	2054	912	802	452	1390	2142	2013
4	DE	1398	1949	1008	0	1019	1059	1273	1771	1411	504	1235	1307
5	HO	789	1804	1067	1019	0	1538	356	1608	1313	1438	1912	2274
6	LA	2182	2979	2054	1059	1538	0	1883	2786	2426	715	379	1131
7	NO	479	1507	912	1273	356	1883	0	1311	1070	1738	2249	2574
8	NY	841	222	802	1771	1608	2786	1311	0	368	2182	2934	2815
9	PI	687	574	452	1411	1313	2426	1070	368	0	1826	2578	2465
10	SL	1878	2343	1390	504	1438	715	1738	2182	1826	0	752	836
11	SF	2496	3095	2142	1235	1912	379	2249	2934	2578	752	0	808
12	SE	2618	2976	2013	1307	2274	1131	2574	2815	2465	836	808	0

- To identify which cities are covered by each hub.
- Simply replace all the distance is less than 1,000 miles (covered) and 0 otherwise..
- To formulate this problem, we define the following binary decision variable:
- We want to minimize the number of hubs, therefore the objective function is:
- Each city must be covered by at least one hub.

Binary-matrix showing cities covered by each hub:

		1	2	3	4	5	6	7	8	9	10	11	12
		AT	BO	CH	DE	HO	LA	NO	NY	PI	SL	SF	SE
1	AT	1	0	1	0	1	0	1	1	1	0	0	0
2	BO	0	1	0	0	0	0	0	1	1	0	0	0
3	CH	1	0	1	0	0	0	1	1	1	0	0	0
4	DE	0	0	0	1	0	0	0	0	0	1	0	0
5	HO	1	0	0	0	1	0	1	0	0	0	0	0
6	LA	0	0	0	0	0	1	0	0	0	1	1	0
7	NO	1	0	1	0	1	0	1	0	0	0	0	0
8	NY	1	1	1	0	0	0	0	1	1	0	0	0
9	PI	1	1	1	0	0	0	0	1	1	0	0	0
10	SL	0	0	0	1	0	1	0	0	0	1	1	1
11	SF	0	0	0	0	0	1	0	0	0	1	1	1
12	SE	0	0	0	0	0	0	0	0	0	1	1	1

- Note that we use the greater than or equal-to sign because a city can be covered by more than one hub. Similarly for Boston (Index 2), we have:
- Solving this binary integer program using software generates three hubs as follows: *Atlanta* covers Chicago, Houston, New Orleans, New York, and Pittsburgh; Pittsburgh covers Atlanta, Chicago, Boston, and New York; Salt Lake City covers Denver, Los Angeles, San Francisco, and Seattle.
- We see that some cities are covered by more than one hub.
- As an example, Chicago is covered by both Atlanta and Pittsburgh hubs.
- In the case where we want to cover each city by exactly one hub, all the inequalities in the above model become equal signs.
- This special case where each member of one set is covered exactly once is called set-partitioning.
- If we run the above program with this restriction, that is, changing all greater than or equal to signs with strictly equal to signs, we find that the minimum number of hubs to cover all cities exactly once is also three.
- Hubs are: *Boston* covers New York and Pittsburgh *New Orleans* covers Atlanta, Chicago, and Houston Salt Lake City covers Denver, Los Angeles, San Francisco, and Seattle Therefore, as the name implies, set- partitioning attempts to make disjoint sets such that no member appears in two sets.

The integer binary programming model is as follows:

- In this model, the objective function seeks to minimize the total covering cost.
- The set of constraints imposes that each member of set 1 is covered by at

least one member of set 2

- Both set-covering and set-partitioning models are used extensively in these models.
- By a set-covering or set-partitioning matrix, we mean a matrix of $a_{i,j}$ parameters where the members of one set (index i) are represented by rows and members of the other set (index j)are represented by columns
- In this matrix a value of 1 means that the specific member in set 1is covered by the specific member in set 2.
- A value of means that this coverage does not exist.

Traveling Salesman Problem:

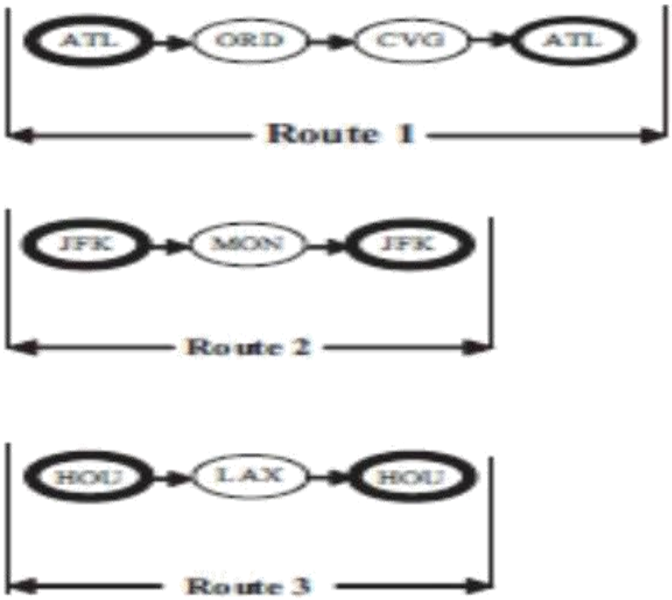
- The Traveling Salesman problem is a classical problem in operations research.
- It has vast applications in sequencing series of jobs or routes.
- The Traveling Salesman problem is as follows: Starting from his hometown, a traveling salesman wants to visit a series of cities just once, and finally return to his hometown.
- The problem is to determine the best sequence for visiting these cities so that the total cost (total distance or total time traveled) is minimized.
- Despite the simplicity of the problem's scope, the solution to this problem is very challenging and falls among one of the most computationally intensive combinatorial problems.
- To clarify this problem, consider the following example.

Example:

- A cargo airline based in Atlanta (ATL) wants to determine the sequence of flights to cities in its network such that the total distance flown in its cycle is minimized.
- A restriction to this operation is that the flight sequences must start and end in Atlanta.

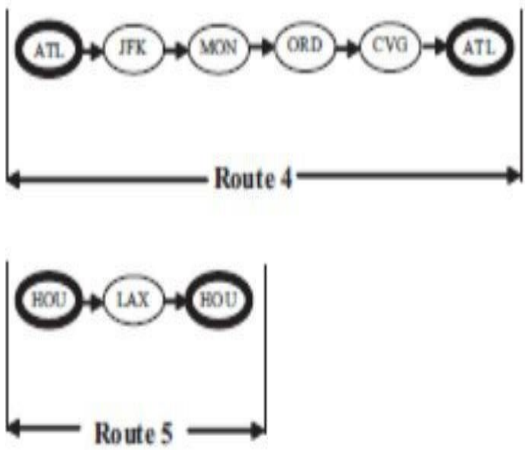
		1	2	3	4	5	6	7
		ATL	ORD	CVG	HOU	LAX	MON	JFK
1	ATL	-	702	454	842	2396	1196	864
2	ORD		-	324	1093	2136	764	845
3	CVG			-	1137	2180	798	664
4	HOU				-	1616	1857	1706
5	LAX					-	2900	2844
6	MON						-	396
7	JFK							-

- The objective function is therefore to minimize the total distances flown:
- The first set of constraints is to make sure that each city is visited only once. For example, for Atlanta we have:
- We write similar constraints for the other six cities.
- The second set of constraints must route the aircraft after visiting a city.
- Without these constraints, the aircraft will be stuck in one city.
- The constraint to route the aircraft after visiting Atlanta.
- We write similar constraints for the other six cities as well.



Solution showing three disjoint sequences or sub-tours:

- This solution shows three disjointed sequences. It does not offer the expected complete tour sequence among all of the seven cities. This is a common difficulty with the Traveling Salesman problem. Instead of one tour of all the cities, the solution generates sub-tours.
- To address this difficulty, the common approach is to prevent the formation of sub-tours. First, the problem is solved, and then we add additional constraints to break these sub-tours, if they are formed.
- As an example, we have the JFK-MON-JFK sub-tour in our solution. To break this sub-tour we add the following constraint in our model:
- We solve the problem once again, adding this new constraint.



Solution showing two sub-tours after adding first breaking Constraint

- We now have two sub-tours. Hence, we add the following constraint to break the second sub-tour:
- Adding this constraint results in the following complete tour solution.

Final tour sequence of flights with distances:

Origin	Destination	Miles
ATL	CVG	454
CVG	JFK	664
JFK	MON	396
MON	ORD	764
ORD	LAX	2136
LAX	HOU	1616
HOU	ATL	842
TOTAL		6872

Sets

- N = Number of cities

Index

- i,j= Index for cities

Parameters

- $c_{i,j}$= Cost of traveling from city i to city

Decision Variable

- The integer programming model is as follows:

Objective Function

- In this model, the objective function seeks to minimize the total travelling cost.
- The set of constraints ensure that each city i is followed by exactly one city j. Similarly, the set of constraints ensures that each city j is visited exactly once.
- The set of constraints imposes the restriction on the sub-tours.
- Variables ti and tj are arbitrary fixed numbers used for breaking the sub-tours.

UNIT II

AIRCRAFT ROUTING & MANAGEMENT OF IRREGULAR OPERATIONS

Introduction

- The solution obtained from the fleet assignment in the previous chapter identifies the flow of fleet through the network.
- However, it does not identify which specific aircraft from that fleet is assigned to each flight leg. Aircraft routing is the process of assigning each individual aircraft (referred to as tail number) within each fleet to flight legs.
- The aircraft routing is also referred to as aircraft rotation, aircraft assignment or tail assignment.
- Major goal of this assignment problem is to maximize the revenue or minimize operating cost with the following considerations (Clarke et al. 1997, Gopalan and Talluri1998, Papadakos 2009).
- Flight coverage: each flight leg must be covered by only one aircraft.
- Aircraft load balance: the aircraft must have balanced utilization loads.
- Maintenance requirements: not all the airports that an airline flies to have the capability to perform maintenance checks on all fleet types.
- The airlines normally have maintenance bases, typically at their hubs, for different fleet types.
- The maintenance consideration is to ensure that the aircraft are flown through the network in a manner that allows them to receive the required maintenance checks at the right time and at the right base.

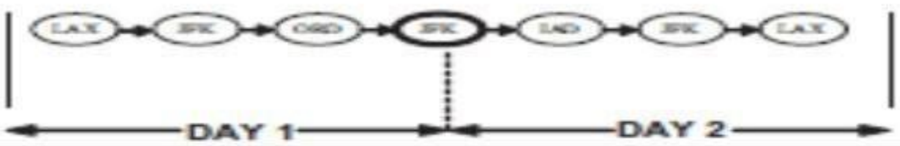
Maintenance Routing

- The mathematical approaches to the aircraft-routing problem typically assume that the same schedule is repeated daily over a period of time.
- A similar approach is adopted for the weekends, when the frequency of flights is lower.
- In our Ultimate Air example, we assume that we have the maintenance facilities for the two fleet types only at our hub, that is, JFK. Each aircraft must be routed so that it stays overnight at JFK, at most after three days of operation.
- Valid Routings for a routing to be valid, it needs to incorporate the turn-around time. Turn-around time is the minimum time needed for an aircraft from the time it lands until it is ready to depart again.
- This time includes the taxi into the gate, unloading passengers and baggage, cleaning, inspections, boarding new passengers, loading new baggage, and so on. The turnaround time varies from 20 minutes to 1 hour among airlines.
- In the Ultimate Air example, we assume that the turn-around time is 45minutes. According to this turn-around time, a valid routing cannot include flight113 followed by flight 138 in our 757 fleet.
- The turn-around time is 20 minutes, which is less than our minimum of minutes.

Routing Cycles:

- For Ultimate Air, we assume that only routes with three-day closed cycles are valid. A closed cycle is when an aircraft starts from a city, and at the end of the three-day cycle, ends up at that same city to start another cycle.
- This requirement is included to better present the process of aircraft routing by reducing the number of potential routings.
- It should be noted that closed cycles are not typically a requirement for airlines.
- The airlines usually develope monthly aircraft routing with no closed cycles.
- That is, an aircraft has the potential to have a totally different routing every day with no pattern or cycles as long as it receives the required maintenance checks.
- The aircraft stays at JFK every night and repeats the cycle every day.
- This routing provides a maintenance opportunity for the aircraft every night.
- It should be noted that each time an aircraft is at a maintenance station, it does not necessarily mean that maintenance is performed on the aircraft.

Flight No.	Origin	Departure Time	Destination	Arrival Time	Flight Hrs	Fleet Type
			DAY 1			
101	LAX	5:00	JFK	13:30	5.5	737-800
129	JFK	15:05	ORD	16:05	2	737-800
109	ORD	17:10	JFK	20:10	2	737-800
			DAY 2			
140	JFK	6:20	IAD	7:20	1	737-800
120	IAD	14:25	JFK	15:25	1	737-800
127	JFK	19:00	LAX	21:30	5.5	737-800



B737-800 two-day routing:

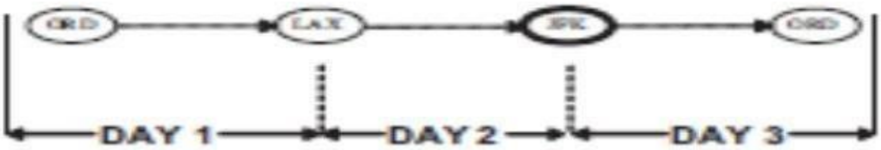
- As it does not provide a maintenance opportunity at JFK after three days of operations.
- Note that in our Ultimate Air example, we only selected one, two and three-day routing cycles.
- The airlines may extend these routings to weekly routings, and soon with a maintenance opportunity every three days.

Route Generators:

- For the proposed set-portioning mathematical model, we begin by generating all possible valid aircraft routings.

- It may seem that generating these routes is a very difficult and tedious task.
- This is certainly the case if we want to enumerate all possible routes manually. Automated systems are used extensively to generate and filter these routes for the airlines in a relatively short time.

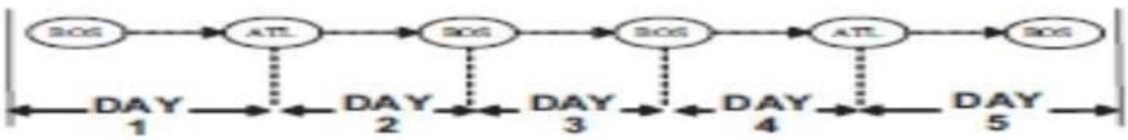
Flight No.	Origin	Departure Time	Destination	Arrival Time	Flight Hrs	Fleet Type
			DAY 1			
107	ORD	7:30	JFK	10:30	2	737-800
141	JFK	12:00	IAD	13:00	1	737-800
120	IAD	14:25	JFK	15:25	1	737-800
124	JFK	19:00	LAX	21:30	5.5	737-800
			DAY 2			
101	LAX	5:00	JFK	13:30	5.5	737-800
129	JFK	15:05	ORD	16:05	2	737-800
109	ORD	17:10	JFK	20:10	2	737-800
			DAY 3			
140	JFK	6:20	IAD	7:20	1	737-800
119	IAD	8:15	JFK	9:15	1	737-800
141	JFK	12:00	IAD	13:00	1	737-800
120	IAD	14:25	JFK	15:25	1	737-800
130	JFK	21:00	ORD	22:00	2	737-800



B737-800 three-day routing:

- Recall that in our Ultimate Air example, we are only interested in three-day cycle aircraft routings.
- That is, after three days the aircraft ends up at the same airport from which it started out on the first day of its cycle, only to repeat another cycle.
- To provide the maintenance opportunity for the aircraft, the routing must include at least one overnight stay at JFK.
- A computer program was developed to generate three-day-cycle aircraft routes. These aircraft are routed through a series of feasible flights.
- This route then selects at least one overnight stay in JFK at the end of the first and/or second day for maintenance.
- At the end of the third day, the aircraft is routed back to the airport as follows: Read the flight numbers, departure and arrival cities, as well as departure and arrival times for a set of flights assigned to a specific fleet (identified by fleet routing).
- Create all possible valid one-day routings incorporating turn-around times– place in a file.
- Attach each feasible one day routing of this file to all other one-day routings in this file.
- Does this step twice to create three-day routings – place in a file.
- Examine each element of this three- day file according to the following criteria:– I t starts and ends at the same city.– Each day, flights start at the city where the aircraft ended the day before.– An overnight stay at JFK occurs at least once.

Flight No.	Origin	Departure Time	Destination	Arrival Time	Flight Hrs	Fleet Type
			DAY 1			
116	BOS	6:15	JFK	7:45	1.5	757-200
131	JFK	9:30	ATL	12:00	2.5	757-200
111	ATL	13:10	JFK	15:40	2.5	757-200
133	JFK	18:05	ATL	20:35	2.5	757-200
			DAY 2			
110	ATL	8:10	JFK	10:40	2.5	757-200
138	JFK	12:30	BOS	14:00	1.5	757-200
118	BOS	15:00	JFK	16:30	1.5	757-200
139	JFK	21:30	BOS	23:00	1.5	757-200
			DAY 3			
116	BOS	6:15	JFK	7:45	1.5	757-200
131	JFK	9:30	ATL	12:00	2.5	757-200
111	ATL	13:10	JFK	15:40	2.5	757-200
139	JFK	21:30	BOS	23:00	1.5	757-200
			DAY 4			
117	BOS	10:00	JFK	11:30	1.5	757-200
138	JFK	12:30	BOS	14:00	1.5	757-200
118	BOS	15:00	JFK	16:30	1.5	757-200
133	JFK	18:05	ATL	20:35	2.5	757-200
			DAY 5			
110	ATL	8:10	JFK	10:40	2.5	757-200
138	JFK	12:30	BOS	14:00	1.5	757-200
118	BOS	15:00	JFK	16:30	1.5	757-200
136	JFK	21:30	BOS	23:00	1.5	757-200



B757-200 five-day routing with no opportunity for overnight maintenance at the JFK hub

- Add each element that satisfies all the above conditions to a file of potential valid three-day routing candidates.
- This program also generates the mathematical model suitable for linear programming software.
- Running this program generated a total of 6,221 and455 valid three-day routings for the 737-800 and 757-200 fleet types respectively.

Mathematical Model for 757-200 Fleet:

- Since the 757-200 fleet has a lower number of flights and routing candidates, we start by developing the mathematical model for this fleet.
- The mathematical model for the 737-700 fleet.
- **Decision Variable***:* The goal of the aircraft-routing problem is to assign routes to individual aircraft within a specific fleet type.
- We generated all possible valid routings.
- Each of these routings qualifies as a candidate to be assigned to an aircraft.

- Among all these candidates, we need to identify those routings that optimize the objective function and satisfy the constraints.
- We define the following binary decision variable to find such routings for the 757-200fleet.

Objective Function:

- We have 455 valid routings for the flights assigned to 757-200 fleet.
- Constraints for 757-200 Fleet
- There are two sets of constraints for our aircraft-routing problem:
- Flight coverage and the number of available aircraft.

Flight Coverage:

- Each routing candidate covers a certain number of flights in its three-day cycle. Each flight must be covered every day.
- For example, sample 1 routing candidate for the 757-200 fleet covers flights 131,111 and 133 in day one.
- In day two it covers flights 110,138, 118 and 133. This routing covers flight 131 in its first day but does not fly this flight in the available mathematical models use different measures for the objective function (see list of references).
- Some of these measures include:

Maximizing through values:

- Non-stop flights are the first choice for passengers. In the absence of such point-to-point flights, passengers must take connecting flights.
- A through flight is a type of connection that uses the same aircraft for the flights involved.
- This enables the passengers to remain onboard rather than deplaning, searching for and walking to their connecting flight-gate.
- Through flights are especially attractive in very busy airports.

Minimizing cost:

- Airlines may assign pseudo-costs to penalize routings which they consider to be unfavorable.
- These unfavorable routes may include bad connection times and circular routings where aircraft are isolated by flying between a small number of spokes, and so on.

Maximizing maintenance opportunities:

- Those routings that provide multiple maintenance opportunities for the aircraft are given higher weights.
- Assume that in our Ultimate Air case, the objective is to select those routings that maximize maintenance opportunities.
- To clarify this, let us return to the five sample routings for the 757-200 fleet
- The first three samples have only one overnight stay at JFK in their three-day

cycles. Accordingly, the coefficients of these variables in the objective function are one.

- For sample routings four and five, this coefficient is two since they have two overnight stays at JFK in their three-day cycles.
- We determine these coefficients for every routing candidate for this fleet.
- Again, a simple computer programe can easily generate these coefficients.
- Thus, the objective function for our 757-200 fleet is as follows: other two days of its cycle.
- Accordingly, other routings with flight 131 in their second and third day of cycles must be selected to cover flight 131 in all three days.
- To cover all flights, we need one constraint for each flight for each day of the three-day cycle.
- As an example, searching through all the 455 routing candidates, only six candidates actually cover flight 125 in different days as shown.

Routing Candidate Variable	Day 1	Day 2	Day 3
x_1	125	105	131-111
x_2	125	105	138-118
x_3	131-111	125	105
x_4	138-118	125	105
x_5	105	131-111	125
x_6	105	138-118	125

- According to the above variable notations, to cover flight 125 in day one, we write the following constraint:
- This is because flight 125 in day one only appears in *x*1 and *x*2. Similarly, to cover this flight in the second and third day of the cycle
- The flight 125 in the second day of the cycle
- The flight 125 in the third day of the cycle
- Similarly, we write the constraints for the other 11 flights.
- Total number of constraints required to cover all daily flights for the 757-200 fleet is 36 (12 flights× 3-daycycles).

Number of Available Aircraft:

- Each routing candidate is a three-day cycle assigned to one aircraft.
- Accordingly, the number of selected routes should not exceed the available number of aircraft tin the fleet.
- We assumed that we have six 757- 200 aircraft.

- The following constraint ensures that the number of selected routes is limited to the number of aircraft.

Solution for 757-200 Fleet:

- We used optimization software to solve this problem.
- The program reported that there is no feasible solution to this problem.
- That is, with six aircraft, it is not possible to cover all the flights assigned to the 757-200 fleet.
- However, our fleet routing of these six aircraft are capable of flying all our757-200 flights through the network.
- So, why do we not get a feasible solution to our aircraft-routing problem? The answer is that the fleet-routing problem does not consider the following constraints that we have imposed on our aircraft routings.
- A 45-minute turn-around time. Three-day closed cycles, starting and ending at the same city.
- This requirement eliminates a large number of potential routes that are perfectly acceptable to the airlines.
- Note that we introduced this arbitrary requirement to reduce the problem size.
- At least one overnight stay at JFK for maintenance in a three-day period.
- These additional constraints in the aircraft-routing problem result in an infeasible solution for our problem.
- To search for solutions, we eliminated the constraint on the number of available aircraft to see how many aircraft would be needed to fly the proposed daily schedule of flights assigned to the 757-200 fleet.
- We ran this model, and the feasible solution now required eight aircraft.
- The solution for this model with eight aircraft

Feasible eight aircraft solution for the 757-200 fleet:

Routing	DAY 1	DAY 2	DAY 3
1	125	105	138-118
2	110	131-111	131-111-133
3	113-135	114	136
4	131-111-136	113-136	114
5	105	138-118	125
6	114	135	113-135
7	138-118	125	105
8	133	110-133	110

- It should be noted that the airlines frequently face this problem where the existing aircraft are not enough to fly the proposed schedule.
- The main reason is that the arriving and departing flights in the proposed schedule are not synchronized.

- Let us look at our Ultimate Air schedule and set of constraints.
- We see that the two flights, 125 and 105, have only two routing candidates each day while other flights have many possibilities (see the constraints for flight 125 in the previous section).

Flights 105 and 125

Flight no.	Origin	Departure time	Destination	Arrival time	(hrs)	Fleet type
125	JFK	7:25	SFO	9:55	5.5	757-200
105	SFO	9:50	JFK	18:20	5.5	757-200

- We see that flight 125 arrives at SFO at 9:55.
- The aircraft flying this flight cannot fly flight 105 because it departs at 9:50.
- Therefore, the aircraft flying flight 125 to SFO is stranded for the entire day, as there are no other flights from SFO for it to connect with.
- So, one possibility that the operations team at Ultimate Air may consider is to synchronize these two flights.
- To do this, we need to delay the departure time for flight 105 (or fly flight 125 earlier).
- If we delay flight 105 by one hour to incorporate our 45- minute turn-around time, then these two flights can be paired.
- The revised schedule for these two flights is with this revised schedule, the two flights, 125 and 105, can be paired.
- This change is incorporated into the route generator program and the revised three-day valid routes are generated.
- The process for developing the linear integer model is repeated, as described earlier. Solving the new model generates multiple optimal solutions with six available aircraft.
- As we see, the same routings are repeated every day of the three-day cycle, but indifferent sequences, which result in multiple optimum solutions.

One of the optimal solutions with six aircraft

Routing	DAY 1	DAY 2	DAY 3
1	125-105	135	114
2	110-138-118-136	113	131-111-133
3	113	131-111-133	110-138-118-136
4	131-111-133	110-138-118-136	113
5	114	125-105	135
6	135	114	125-105

- As this process has shown, changing the departure time for one flight results in a solution with two less aircraft.
- Furthermore, examining this solution more closely, we notice that the aircraft flying flights 113, 114 and 135 are also stranded at their respective destinations, away from the JFK hub, at the end of the day.
- Despite the fact that we are covering all our flights with the available six aircraft of the 757-200 fleet type, it is possible to add more flights without needing more aircraft. Further synchronizing the arrival and departure times for these flights will further reduce number of aircraft needed.
- The value of the objective function for this solution is nine.
- This represents the total number of aircraft grounded over nigh at JFK over the three-day cycle.
- According to this solution, each night, three 757-200 aircraft stay at JFK for maintenance. Routes 1, 5, and 6 provide two maintenance opportunities each during their three-day cycles.
- This process of changing arrival/departure times is very common among airlines.
- The initial schedule proposed by the marketing department and schedule-builders provides feedback to the schedule-builders on operational feasibility and possible changes to the schedule.
- This feedback process continues until all parties are satisfied with the schedule.
- Once the airline finds its routings to be feasible and satisfactory, it then assigns each route to a particular aircraft tail number.
- Note that in the above aircraft routing process, we are indifferent to the method used for assigning tail numbers to the selected routes.
- If, however, there are such influencing factors as aircraft age within the fleet, then the airline may use some rule/criteria for assigning specific tail numbers to routes.

Solution for 737-800 Fleet:

- The same mathematical model approach as described earlier for the 757-200 fleet is adopted for aircraft routing of the 737-800 fleet.
- Recall that we have nine aircraft in this fleet.
- Again, there are no feasible solutions to this aircraft- routing problem with only nine aircraft.
- Similarly, examining this solution, we notice that flights 102, 106, 126, 112and 123 are all stranded at their respective destinations at the end of the day.
- We need one aircraft each day just to fly these flights.
- It shows the detailed schedule for these five flights.
- In an effort to pair the above flights, considering our 45-minute turn-around time, In corporating these changes, and running the program with this revised schedule, still results in no feasible solution.

- That is, even with these changes, it is still not possible to fly all flights with nine aircraft in a three-day cyclic routing.

Solution for aircraft routing of 737-800 fleet with 12 aircraft

Routing	Day 1	Day 2	Day 3
1	101-142-121-139	116-134-115	140-119-128-108-124
2	116-134-115	126	104-142-121-139
3	104-126	106	126
4	140-119-128-108-127	104-132	112
5	102	122-103	123
6	107-141-120-124	102	137-117-129-109-130
7	132	112	122-103
8	106	137-117-142-121-130	107-141-120-127
9	122-103	123	102
10	123	101-129-109-139	116-134-115
11	137-117-129-109-130	107-141-120-127	106
12	112	140-119-128-108-124	101-132

Flight schedule for B737-800 stranded flights

Flight no.	Origin	Departure time	Destination	Arrival time	(hrs)
102	LAX	09:45	JFK	18:15	5.5
106	SFO	15:25	JFK	23:55	5.5
126	JFK	15:30	SFO	18:00	5.5
123	JFK	16:00	LAX	18:30	5.5
112	ATL	18:00	JFK	20:30	2.5

Revised flight schedule for B737-800 stranded flights

Flight no.	Origin	Departure time	Destination	Arrival time	(hrs)
102	LAX	07:45	JFK	16:15	5.5
106	SFO	10:25	JFK	18:55	5.5
126	JFK	18:30	SFO	21:00	5.5
123	JFK	19:00	LAX	21:30	5.5
112	ATL	19:00	JFK	21:30	2.5

- By relaxing the constraint on the number of aircraft, we see that the minimum number of aircraft required to fly the daily schedule of 737-800 flights is 10.
- The changes made to the flight schedule for the stranded flights reduced the number of aircraft needed from 12 to 10.
- It shows the routings for this10-aircraft solution.
- Other minor changes to the flight schedule also failed to generate a solution that flies all the above flights with nine aircraft.
- Again, our routing problem here is more restricted than a typical airline-routing problem because of our closed- cycle requirement.
- It is, of course, possible to manually make major changes to the schedule by pairing the flights such that a feasible solution is obtained with nine aircraft.
- (Schedule) and (routings) represent such a solution with a totally modified schedule. However, it is not clear if this operationally feasible solution is also attractive to the marketing department and passengers.

Mathematical Models:

Sets:

- F = Set of flights
- R= Set of feasible routings

Indices:

- j = Route index
- i = Flight index *Parameters*
- c_j= Cost of route j
- $a_{i,j}$= 1 if flight i is covered by route j, and 0 otherwise
- N = Total number of aircraft in the fleet

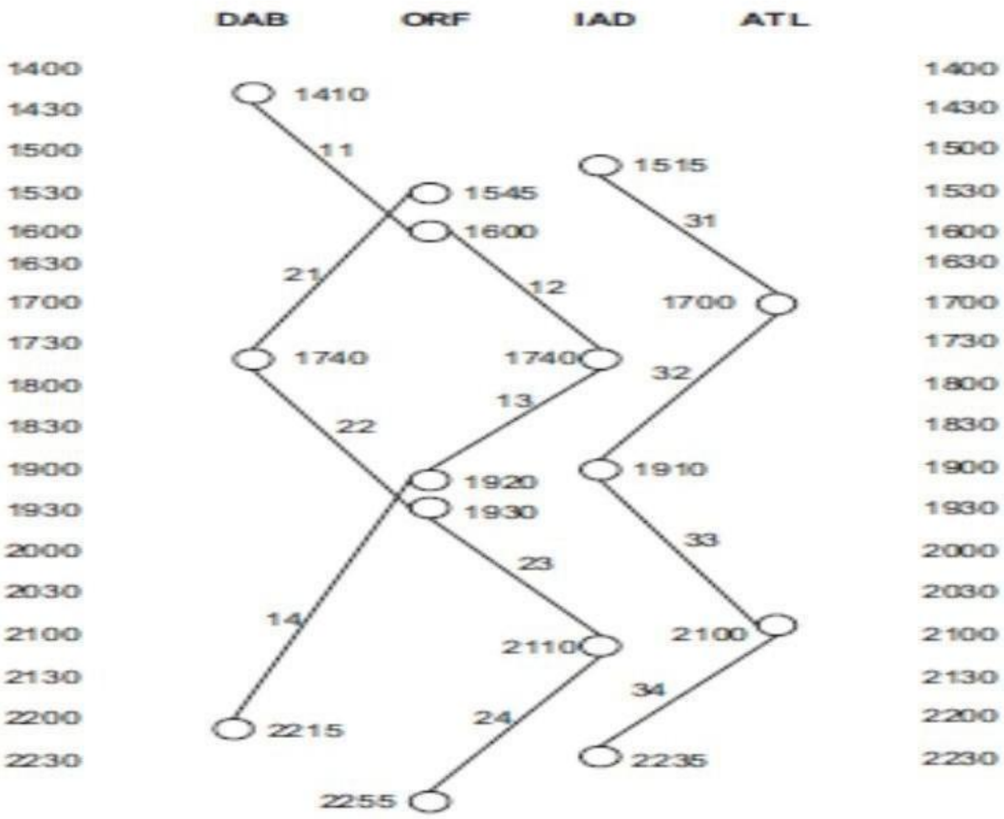
Aircraft routing for B737-800 with nine aircraft

Routing	Day 1	Day 2	Day 3
1	122-101	126-104	123-102
2	126-104	123-102	124-103
3	123-102	124-103	127-106
4	124-103	127-106	122-101
5	127-106	122-101	126-104
6	140-119-141-120-142-121	137-116-139-117	128-107-129-108-130-109
7	137-116-139-117	128-107-129-108-130-109	134-115-132-112
8	128-107-129-108-130-109	134-115-132-112	140-119-141-120-142-121
9	134-115-132-112	140-119-141-120-142-121	137-116-139-117

Indices

Time-band Approximation Model:

- The network structure is similar to the time-space network.
- The time-space representation of the above case study without any disruption.



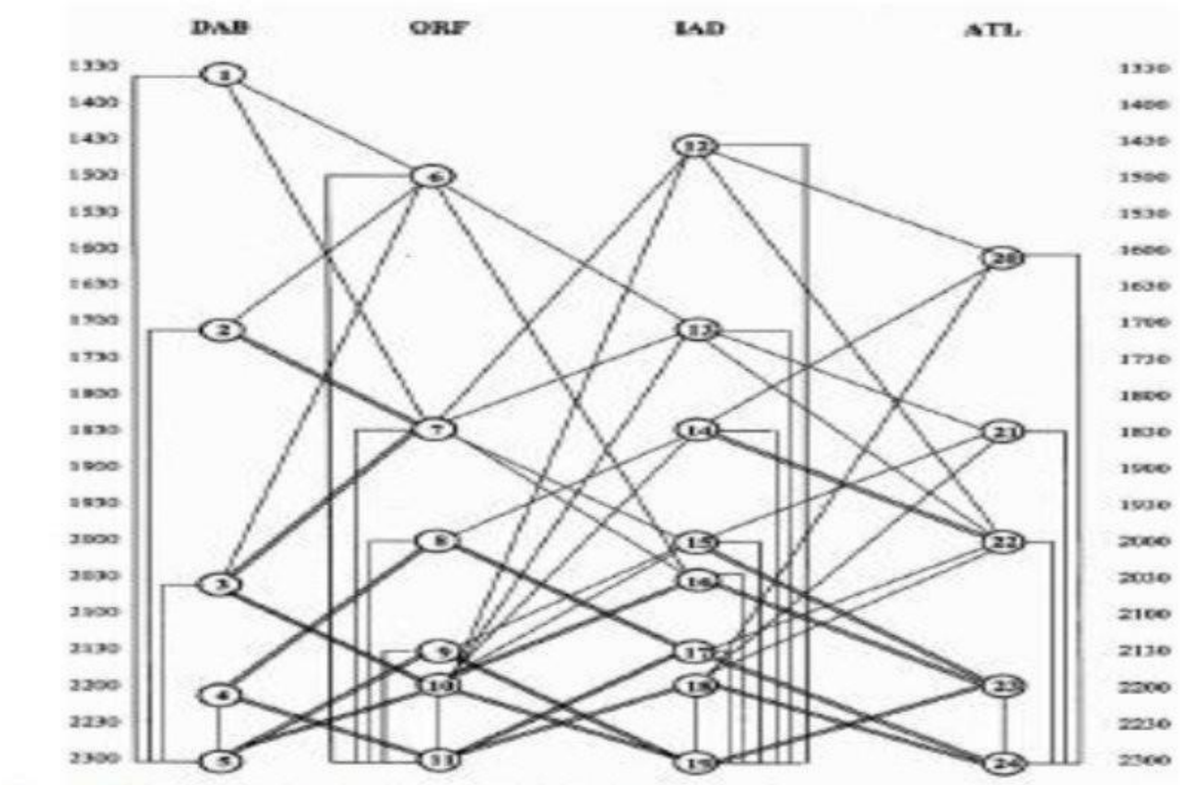
Time band network for the case study:

- The flight numbers are shown on the flight arcs.
- The nodes represent an arrival and departure at a specific time.
- In this network, it is similar to the cities and times are represented horizontally and vertically respectively.
- In this model the time horizon is partitioned into time bands or discrete intervals of fixed length.
- By partitioning the time horizon into time bands, station activity is aggregated into that time-band node.
- We also have the following assumptions for our case study: each station requires a minimum of 40 minutes turnaround time; midnight arrival/departure curfew (no arrival or departure after midnight);each minute of delay on any flight costs the airline $20;cancellation cost for each flight leg is as follows.

Cancellation cost for flight legs:

Aircraft ID	Flight ID	Origin	Destination	Cancellation cost
Aircraft 1	11	DAB	ORF	$7,350
	12	ORF	IAD	$10,231
	13	IAD	ORF	$7,434
	14	ORF	DAB	$14,191
Aircraft 2	21	ORF	DAB	$11,189
	22	DAB	ORF	$12,985
	23	ORF	IAD	$11,491
	24	IAD	ORF	$9,581
Aircraft 3	31	IAD	ATL	$9,996
	32	ATL	IAD	$15,180
	33	IAD	ATL	$17,375
	34	ATL	IAD	$15,624

- ➢ A major assumption and rule in this model is that during the recovery period, any flight arc from any airport (node) can be made available to other feasible airports (nodes).
- ➢ Considering the time-band intervals, the flight paths, and the fact that flight from every node is available to every other feasible node, results in the following time band network.



Time band approximation network

- ➢ All 30-minute activities within an airport are aggregated in a single node. As an example, node 1 represents all activities from 1:30 p.m. through1:59 p.m.

- Argüelles et al. (1998) classify the nodes in two groups: transshipment and sink nodes.
- Transshipment nodes, also referred to as station- time nodes, are those nodes that the aircraft arrives into and leaves.
- These nodes include 1, 2, 3, 4, 6, 7, and so on. Sink nodes, also referred to as station sink nodes, represent those nodes that the aircraft arrives into but does not leave until the end of recovery time.
- These nodes are similar to starting nodes for wrap-around arcs.
- The nodes 5, 11, 19, and 24 are station sink nodes.
- For example, two arcs are drawn from node 2 to node7.
- These two arcs represent flights 11 and 22. For the arc representing flight 11, we have a delay of 210 minutes.
- This is because flight 11 was scheduled to leave DAB at 1410.
- If this flight occurs in node 2, we have the departure time of 1700.
- Considering the nodes are on 30-minute time-bands, this delay spans from 14:00 to17:30, a total of 210 minutes.
- Each minute of delay costs the airline $20. So, flight11 has a delay cost of $4,200 if it departs from node 2.
- The other arc connecting nodes 2 to 7 represents flight 22.
- By looking at the departure and arrival times of this flight, there is no delay (within a 30 minute time-band) for this flight.
- The non-zero delay costs for all flight arcs.

Scenario 1:

- Let us assume that aircraft 2 in airport ORF becomes grounded owing to some mechanical failure at 1400 and is unavailable for the rest of the day.
- The obvious solution without permitting any rerouting of other aircraft is to cancel flights 21, 22, 23, and 24 which are conducted by this grounded aircraft for the day. These cancellations cost the airline a total of $45,246 (the sum of all cancellation costs for these four cancelled flights).
- Let us see how this problem is solved through a series of aircraft rerouting and cancellations in an effort to minimize the total cost to the airline.

Non-zero delay costs:

Flight number	Origin node	Destination node	Delay cost
11	2	7	4,200
11	3	10	8,500
11	4	11	10,300
12	7	15	3,900
12	8	17	5,700
12	9	19	7,800
12	10	19	8,100
13	14	8	1,800
13	15	9	3,900
13	16	10	4,200
13	17	11	5,700
13	18	11	6,100
14	8	4	1,800
14	9	5	3,900
14	10	5	4,200

Non-zero delay costs

Flight number	Origin node	Destination node	Delay cost
21	7	3	4,300
21	8	4	6,100
21	9	5	8,200
21	10	5	8,500
22	3	10	4,300
22	4	11	6,100
23	8	17	1,600
23	9	19	3,700
23	10	19	4,000
24	17	11	1,400
24	18	11	1,800
31	13	21	2,900
31	14	22	4,700
31	15	23	6,800
31	16	23	7,100
31	17	24	8,600
31	18	24	9,000
32	21	15	2,300
32	22	17	4,100
32	23	19	6,200
33	15	23	2,100
33	16	23	2,400
33	17	24	3,900
33	18	24	4,300
34	23	19	2,000

Decision Variables:

- We define the following decision variables.
- z_i= Number of aircraft (integer) terminated at station node *i* (node *i* being a station sink node).
- The binary Variable, x_{kij} is used to identify which flights should be conducted along which routes.
- For example, *x*11 1, 6 represents the variable for flight number 11 through nodes 1 to 6.
- The binary variable y_k is adopted to identify which flight(s) should be cancelled.
- For example, *y*11 represents the decision variable for canceling flight number 11.
- A value of 1 for this variable means that the flight should be cancelled.
- The integer variable z_i is used to keep track of aircraft balance and to have aircraft available at the end of the day for the next day's flight schedule.
- For example *z* 1 is the number of aircraft in node 1 (DAB) which is not flown through the day and is carried to the station-sink5.

Objective Function:

- The objective function consists of two terms, the delayed cost and the cancellation cost for each flight.
- Constraints for this mathematical model, we have three sets of constraints as follows:

Set 1 – Flight Coverage:

- Each flight must either be flown or be cancelled. As an example to express flight.
- We write similar equations for every available flight a total of 12 constraints for this set.

Set 2 – Station Time-Node Flow:

- For this set we need to write the flow of aircraft at each node.
- There are some nodes that have aircraft available (supply nodes) to start the flow within the network (such as nodes 1, 6, and 12).
- Most of the station-time nodes are transshipment nodes signifying that the net flow in these nodes is zero.
- The net flow for a node is determined as follows:
- The number of aircraft in a node = number of outgoing aircraft from the node
- (Minus) incoming aircraft into the node + (plus) the number of aircraft carried over from this node to sink node (same city) for the next day's operation.
- For example, for node 1 we have: two outgoing flights from node 1.
- These are flights 11 represented by arc 1, 6 and flight 22, represented by arc 1, 7.
- There are no incoming flights to node $z1$ represents the number of aircraft that are carried over to node 5 which is a sink station node.
- The purpose of z variables is to allow the flexibility to the model to save the aircraft at some specific cities for the next day's operation.
- The right hand side of the above equation is 1.
- This is because at node 1, (DAB) we have one aircraft available to start the flights from DAB.
- Similarly the flow balance for node 2 (transshipment node) is as follows
- We have two outgoing (flights 11 and 22) and one incoming flow (flight 21)in this node $z2$ is the number of aircraft that are grounded in DAB and are carried over to station-sink 5.
- The right hand side of this constraint is zero since node 2is a transshipment node.
- As the aircraft at node 6 is grounded and not available the right hand side for this constraint is also zero.
- In this case we have 20 station-time nodes resulting in 20 constraints for this set.

Set 3 – Station Sink-Node Flow:

- We include this set of constraints to ensure that there are aircraft available in the designated airports at the end of the day to fly the flights for the next day according to the published schedule.
- Basically the following rule applies for these sink nodes: Required number of aircraft at any sink node = Total incoming flight terminating at this sink node + (plus) number of carried over aircraft from previous transshipment nodes at

this airport.

- In this case study, to be able to fly the published schedule for the next day, we must have one aircraft available in DAB, ORF, and IAD each.
- Therefore we must ensure that the net flow in station sink nodes for these cities is one.
- Without this set of constraints, the aircraft may end up at the wrong airports at the end of the day.
- The following constraint represents the net flow for DAB for station sink node 5
- There are four arcs coming from other cities to node 5.
- The first four terms of the above equation represent these four arcs (flights).
- The other four terms represent the number of aircraft from previous nodes in the same city (DAB) carried over to this sink node.
- It should be noted that for ORF, we assumed that the aircraft is grounded and not available for the rest of the day.
- It is assumed, however, that it will be available for the next day.
- We have four sink nodes which will result in four constraints for this set.

Solution:

- The above case study has 64 flight arcs (*x* variables), 12 flight cancellation (*y* variables) and 20 termination nodes (*z* variables), a total of 96 binary/integer variables has 36 constraints.

Solution for Scenario 1:

Aircraft ID	Flight	Origin	Destination	Origin node	Destination node	Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	1	6	-	-
	21	ORF	DAB	6	2	-	-
	22	DAB	ORF	2	7	-	-
	23	ORF	IAD	7	16	-	-
	24	IAD	ORF	16	10	-	-
	14	ORF	DAB	10	5	4,200	-
Cancel	12	ORF	IAD	-	-	-	10,231
	13	IAD	ORF	-	-	-	7,434
Aircraft 3	31	IAD	ATL	12	20	-	-
	32	ATL	IAD	20	14	-	-
	33	IAD	ATL	14	22	-	-
	34	ATL	IAD	22	18	-	-
Total cost						4,200	17,665

- The minimum cost solution for this scenario is two cancellations and one delayed flight at a total cost of $21,865 ($4,200+$17,665).
- Compare this cost with the trivial solution of $45,246 resulting from canceling all flights operated by aircraft 2.
- Note that this model was based on aggregating the activities within an airport in a 30-minute time-band into one single node.

- The above solution is utilized to fine-tune and determine the actual departure/ arrival times and the actual cost for each flight.
- The detailed solution for each flight with its revised arrival/departures.

Detailed and final solution for Scenario1:

Aircraft ID	Flight	Origin	Destination	Departure time	Arrival time	Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	1410	1520	-	-
	21	ORF	DAB	1600	1715	300	-
	22	DAB	ORF	1755	1905	300	-
	23	ORF	IAD	1945	2045	300	-
	24	IAD	ORF	2125	2225	200	-
	14	ORF	DAB	2305	0020	4,500	-
Cancel	12	ORF	IAD	-	-	-	10,231
	13	IAD	ORF	-	-	-	7,434
Aircraft 3	31	IAD	ATL	1515	1620	-	-
	32	ATL	IAD	1730	1830	-	-
	33	IAD	ATL	1910	2020	-	-
	34	ATL	IAD	2100	2205	-	-
Total cost						5,600	17,665

- The above departure/arrival times accommodate for 40-minute aircraft turnaround times.
- This is because of the 30-minute aggregation in a single node.
- The actual total cost for the above feasible solution is $23,265, which is still significantly lower than the trivial cost.

Scenario 2:

- In scenario 1, we assumed that aircraft 2 was grounded. In scenario 2, we assume that both aircraft 1 and 2 are operational for the day but aircraft 3 is grounded in IAD at 14:00 and will be unavailable for the rest of the day.
- The trivial solution is to cancel all flights conducted by this aircraft, that is, flights 31, 32, 33, and 34 at a total cost of $58,175.
- The mathematical model is basically very similar to scenario 1, with the following minor changes: In set 2 of the constraints, for station-time node 6, the right-hand side becomes one since aircraft 2 is available in city ORF.
- The right-hand side for node 12, however, becomes zero because aircraft 3 is grounded in IAD and is unavailable.
- Similarly, in set 3 of the constraints, the right-hand sides for nodes 11 and

19become one and zero respectively.

Solution for Scenario 2

Aircraft ID	Flight	Origin	Destination	Origin node	Destination node	Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	1	6	-	-
	12	ORF	IAD	6	13	-	-
	33	IAD	ATL	13	22	-	-
	34	ORF	IAD	22	18	-	-
	24	IAD	ORF	18	11	1,800	-
Aircraft 2	21	ORF	DAB	6	2	-	-
	22	DAB	ORF	2	7	-	-
	23	ORF	IAD	7	16	-	-
	13	IAD	ORF	16	10	4,200	-
	14	ORF	DAB	10	5	4,200	
Cancel	31	IAD	ATL	-	-	-	9,996
	32	ATL	IAD	-	-	-	15,180
Total cost						10,200	25,176

- The total cost for this solution is $35,376.
- The overleaf, shows the conversion of this solution to actual departure and arrival times.
- The total cost for this actual flight schedule is also $35,376 which is similar to the approximation time-node solution.

Scenario 3:

- In this scenario, we assume that aircraft 2 and 3 in cities ORF and IAD are operational all day.
- Aircraft 1 in DAB, however, must be grounded at 13:00 for four hours.
- That is, aircraft 1 is unavailable from 13:00 to 17:00.
- The trivial solution is to cancel flights 11 and 12 which are flown by aircraft 1 during 13:00 to 17:00.
- The total cost associated with these two cancelled flightsis$17,581.
- In set 2 of the constraints, for station-time node 6 and 12 the right hand side becomes one.
- The right hand side for node 1 becomes zero because aircraft 1 is grounded in DAB. This aircraft returns back to service after four hours at 17:00.

Detailed and final solution for Scenario 2:

Aircraft ID	Flight	Origin	Destination	Departure time	Arrival time	Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	1410	1520	-	-
	12	ORF	IAD	1605	1700	-	-
	33	IAD	ATL	1910	2020	-	-
	34	ORF	IAD	2100	2205	-	-
	24	IAD	ORF	2245	2345	1,800	-
Aircraft 2	21	ORF	DAB	1545	1700	-	-
	22	DAB	ORF	1740	1850	-	-
	23	ORF	IAD	1930	2030	-	-
	13	IAD	ORF	2110	2210	4,200	-
	14	ORF	DAB	2250	0005	4,200	
Cancel	31	IAD	ATL	-	-	-	9,996
	32	ATL	IAD	-	-	-	15,180
Total cost						10,200	25,176

- Therefore, the right hand side value for node 2 (representing DAB at 17:00) becomes.
- In set 3 of the constraints, since all the three aircraft are available to station sink nodes, we set the right-hand side values for nodes 5, 11, and 19 equal to 1.
- The solution to this linear integer programming model overleaf.
- Based on the above solution no flight is cancelled and the total cost to the airline is $15,800.
- The actual departure and arrival for each flight derived.
- The total actual cost for the above feasible schedule is $13,500.

Mathematical Model

- This section formally introduces the integer linear programming model adapted for the case study.

- This approach is based on the Time-Band Approximation Model by Argüelles et al. 1998.

Table 10.8 Solution for Scenario 3

Aircraft ID	Flight	Origin	Destination	Origin node	Destination node	Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	2	7	4,200	-
	12	ORF	IAD	7	15	3,900	-
	33	IAD	ATL	15	23	2,100	-
	34	ATL	IAD	23	19	2,000	-
Aircraft 2	21	ORF	DAB	6	2	-	-
	22	DAB	ORF	2	7	-	-
	23	ORF	IAD	7	16	-	-
	24	IAD	ORF	16	10	-	-
Aircraft 3	31	IAD	ATL	12	20	-	-
	32	ATL	IAD	20	14	-	-
	13	IAD	ORF	14	8	1800	
	14	ORF	DAB	8	4	1800	
Total cost						15,800	-

Table 10.9 Detailed and final solution for Scenario 3

Aircraft ID	Flight	Origin	Destination	Departure time	Arrival time	Actual Delay cost	Cancellation cost
Aircraft 1	11	DAB	ORF	1700	1810	3,400	-
	12	ORF	IAD	1850	1955	3,300	-
	33	IAD	ATL	2035	2145	1,700	-
	34	ATL	IAD	2225	2330	1,700	-
Aircraft 2	21	ORF	DAB	1545	1700	-	-
	22	DAB	ORF	1740	1850	-	-
	23	ORF	IAD	1930	2030	-	-
	24	IAD	ORF	2115	2215	-	-
Aircraft 3	31	IAD	ATL	1515	1620	-	-
	32	ATL	IAD	1730	1830	-	-
	13	IAD	ORF	1910	2010	1,800	
	14	ORF	DAB	2050	2205	1,600	
Total cost						13,500	-

UNIT III
FLIGHT SCHEDULING

Introduction:

- Flight scheduling is the starting point for all other airline planning and operations.
- The flight schedule is a timetable consisting of what cities to fly to and at what times.
- An airline's decision to offer certain flights will mainly depend on market demand forecasts, available aircraft operating characteristics, available manpower, regulations, and the behavior of competing airlines.
- The number of airports and flight frequencies served by an airline usually expresses and measures the physical size of the airline network.
- For large air carriers, the flight- scheduling group and route development may contain more than 30 employees.
- It shows a small portion of the daily flight schedule for Delta Air Lines.
- The level of detail in constructing the flight schedule varies among the airlines, but it will be a complete schedule for a full cycle.
- A cycle is normally one day for domestic and one week for international services.
- The schedule construction phase begins with the route system.
- The cities in the airline network determine the route system.
- The economics of an air carrier are driven by its route system.
- All the short- and long-term costs attributed to fleet, avionics, labor contracts, and operations are tied to the route systems of an airline.
- The marketing department plays an important role in the construction of this schedule.
- Before the 1978 Airline Deregulation Act, airlines had to fly routes as assigned by the Civil Aeronautics Board (CAB) regardless of the demand for the service! During this period, most airlines emphasized long point-to-point routes.
- Since deregulation, airlines have gained the freedom to choose which markets to serve and how often to serve them.
- This change has led to a fundamental shift in most airlines' routing strategies from point-to-point flights to hub-and-spoke oriented networks.
- The schedule construction phase is a rough first schedule, which requires extensive modification to be both operationally feasible and economically viable.

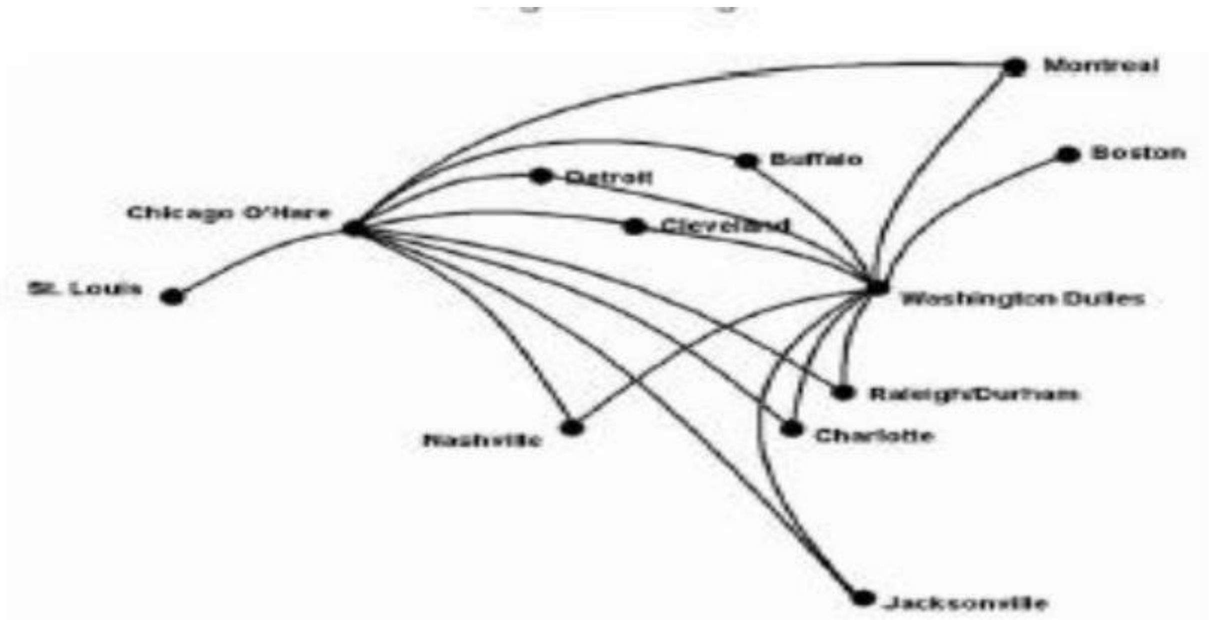
Hub-and-Spoke:

- Most airlines adopt some variation of a hub-and-spoke system.
- Major carriers operate up to five hubs, while smaller ones typically have one hub located at the center of the region they serve.

➢ Each hub has a set of cities that it serves, normally referred to as spokes.
➢ An airline network with Chicago O'Hare and Washington Dulles as hubs.
➢ Air carriers normally assign large capacity non-stop flights between their hubs.
➢ Smaller airplanes are assigned to hub-and-spoke flights.
➢ Major advantages for the airlines adopting hub-and-spoke operations include higher revenues, higher efficiency, and lower number of aircraft needed as compared with point-to-point operations.
➢ Disadvantages of these operations include discomfort to the passengers, as they may require multiple connecting flights at different hubs, congestions and delays at hub airports, and higher personnel and operational costs for the airlines.

Route Development and Flight-Scheduling Process:

➢ There are two types of route development activities: strategic and tactical.
Strategic development: focuses on future schedules which may range from a few months to ten years depending on the air carriers' policies.
➢ Strategic developments respond to major changes in both business and operational environments.
Tactical strategies: focus on short-term changes to the schedule and routes, sometimes on a daily basis.
➢ This is done by constantly monitoring markets, competitors, and operations.



A sample airline network with two hubs and nine spokes

➢ The tactical strategy includes adding, dropping flights, and making changes to city-pair markets and their frequencies.
➢ The following section briefly describes the phases of developing a flight

schedule and the decisions made at each phase.

60+ months	36-12 months	12-3 months	4-1 months
Long range Planning	Market Evaluations	Schedule Optimization	Schedule issues

Long-Range Schedule Planning:

- Fleet diversity Manpower planning Protecting hubs
- Adding or changing hubs adequate facilities at airports.

Market Evaluations:

- Frequency and time of service to each market Adding new and dropping existing markets

Pricing policies:

- Predicting competitors' behaviors
- Code-sharing agreements and alliances.

Schedule Optimization:

- Developing initial schedule based on available fleet Assigning aircraft to flights
- Evaluating facilities and manpower capabilities.

Schedule Issues:

- Crew issues
- Arrival departure times Maintenance issues.
- As described earlier, flight schedule construction is the basis for all other operations. It is therefore important to include detailed airline operations in the process of flight scheduling.
- This, however, creates a complex system with a large number of variables in the model.
- Owing to its complexity it is almost impossible to formulate the complete scheduling construction problem as a mathematical model.
- As a result, the schedule construction process is performed through a structured planning process involving various parts of the airline.
- This planning process is decomposed into sub-problems with less complexity,

which are solved sequentially.

- One of the major drawbacks of this approach is that an individual sub-problem's solution might not be good for the overall airline operations.
- To overcome this difficulty, the process of flight scheduling is performed on a feedback system.
- That is, if the solutions to some sub-problems are not desirable, the flight schedules are altered to see the impact of such changes.
- The process of flight schedule development and the hierarchy of various phases of airline planning.

Load Factor and Frequency:

- Average load-factor plays an important role in determining the frequency of flights between city pairs.
- Load factor is the average percentage of aircraft seats which are filled with passengers.
- The parameters affecting load factors include flight times, frequency, type of service and, of course, fare levels.
- It should be noted that a higher load-factor does not necessarily translate into higher revenues for the airlines.
- As an example it shows the fares, expected demands and load factors for a 150-seat Airbus A-320.
- An 85% load factor generates higher revenues of more than 100% for the airline.
- The load factor is utilized to determine the frequency between city pairs.
- Let the forecasted daily number of passengers between two cities be *PAX* and the airline's policy on average load-factor be *LF*.
- Further, let us assume the average aircraft capacity is *CAP*.
- Then the frequency (*FREQ*) of flights between these two cities is determined by:
- The formulae for load factor is FREQ PAX CAP LF= × (3.1)
- As implied by the above equation, the load factor and frequency have an inverse relationship.
- It is up to the marketing and scheduling departments to actually assign these frequencies between city pairs to different times of the day/week.

Case Study:

- Ultimate Air is a new airline that provides service to the most important domestic business destinations within the United States from its hub at JFK in newyork.
- The cities serviced from JFK are Boston (BOS), Los Angeles (LAX), San Francisco (SFO), Miami (MIA), Atlanta (ATL), Washington D.C. (IAD), and Chicago (ORD).

Ultimate Air route network:

- Based on forecasts, the airline's load-factor policy, and marketing analysis, the airline has proposed providing three daily round-trip flights from JFK to each city in the network.
- It has also developed a first draft of its schedule for the next quarter.
- The complete flight schedule route, incorporating the 42 flights per day.
- The arrival and departure times are local times.
- Presents the demand distribution for each flight as well as distances between cities.
- It is assumed that demand for each flight is normally distributed with the given means and standard deviations.

Destination in miles, demand means and standard deviations for Ultimate Air network

Flight no.	Origin	Destination	Distance (miles)	Demand	Standard deviation
101	LAX	JFK	2475	175	35
102	LAX	JFK	2475	182	36
103	LAX	JFK	2475	145	29
104	SFO	JFK	2586	178	35
105	SFO	JFK	2586	195	39
106	SFO	JFK	2586	162	32
107	ORD	JFK	740	165	33

UNIT IV
FLEET ASSIGNMENT & CREW AND MANPOWER SCHEDULING

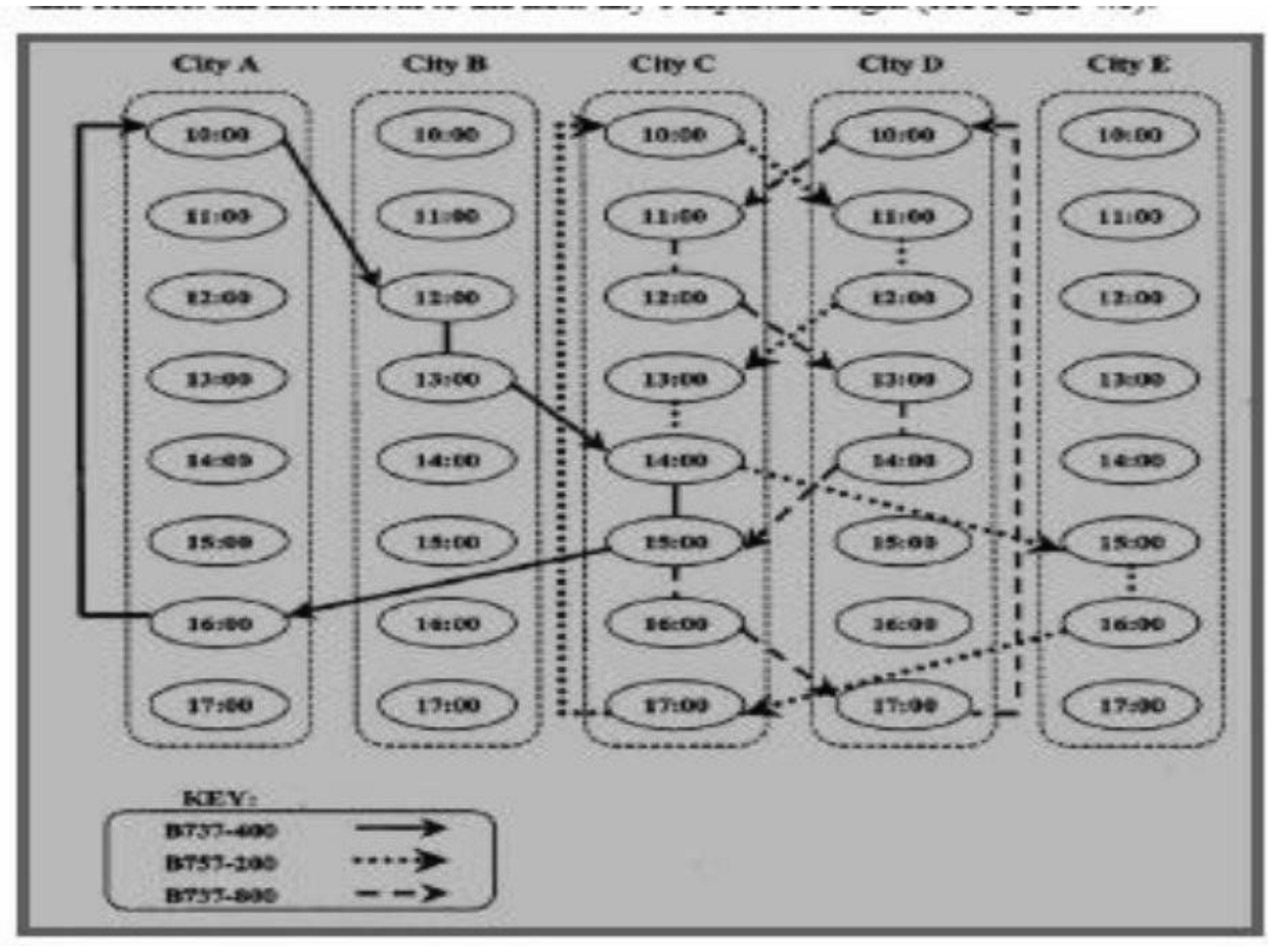
Introduction:

- Following the construction of a flight schedule and its corresponding network, the next step is to assign the right fleet type to each flight in the schedule.
- The task of fleet assignment is to match each aircraft type in the fleet with a particular route in the schedule.
- It should be noted that this phase of planning concerns only fleet type and not a particular aircraft.
- The goal of fleet assignment is to assign as many flight segments as possible in a schedule to one or more fleet types, while optimizing some objective function and meeting various operational constraints.
- Fleet assignment should not be confused with fleet planning.
- Fleet planning is a strategic decision normally undertaken when an airline is conceived, and concerns the number and type of aircraft needed for operation.
- Itentails the process of acquiring the appropriate aircraft-types in order to serve the anticipated markets based on the airline's strategic plan.
- Fleet planning addresses fleet size and fleet mix. In fleet assignment, however, we assume that the airline is operational with the existing aircraft in its fleet, and the problem is to assign a fleet type to each flight leg.
- Airlines typically operate a number of different fleet types. Each fleet type has different characteristics and costs, such as seating capacity, landing weights, crew, maintenance, and fuel.
- It presents the fleet diversity for select airlines. Maintenance cost is a major factor that persuades airlines to be less diverse when planning for their fleet.
- Fleet diversity requires the airlines to have skilled crew and personnel for each fleet type, plan for different maintenance checks, and have less flexibility in replacing an aircraft when a failure occurs.

Mathematical Model:

- A major concern in formulating the fleet assignment problem is keeping track of the fleet at different stations (airports) at any given point in time.
- Fortunately, researchers have developed an ingenious method of adopting a time-space network to formulate this problem.
- This approach facilitates the process of modeling the fleet assignment problem.
- The above time-space network presents the airports as columns, and times of the day as rows.
- In this network, the arcs (arrows) are the flights, and nodes represent the arrival/departure of a flight segment at a specific airport, at a specific time of the day. A wrap-around arc is a ground arc which connects the last node to the first node in a given city.

➢ These arcs normally represent the aircraft that stay overnight in an airport, and connect the last arrival to the next day's departure flight.



An example of a time-space network:

➢ The fleet assignment problem is basically formulated as a multi-commodity network problem.

➢ Each node represents supply/demand, which can be satisfied through a diverse fleet. The model seeks to minimize the total cost or maximize the net profit by assigning the most appropriate fleet type to each flight leg.

➢ The constraints ensure that each flight is assigned to a particular fleet type, and that the number of aircraft for each fleet does not exceed the number of available aircraft.

➢ Other side-constraints may include curfew, range, noise, forced turns, maintenance, and user-specific restrictions.

➢ In the mathematical model presented here, the objective function represents the total cost of the network, which we seek to minimize.

➢ These costs include two parts: operating costs and spill costs.

Constraints:

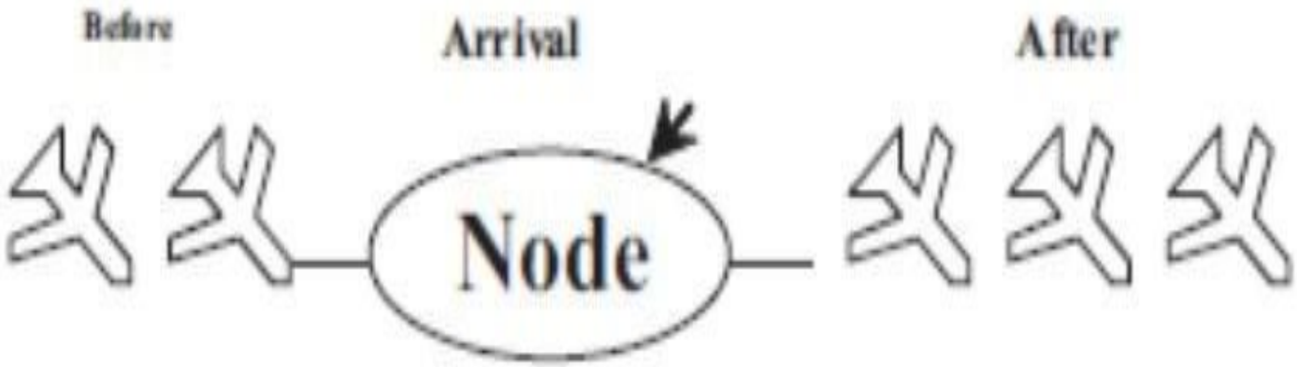
➢ There are three main sets of constraints in the fleet assignment model.

Flight Cover:

- The first set of constraints is what is typically known as flight cover.
- Flight cove implies that each flight must be flown.
- To cover a flight, the sum of all the decision variables representing that flight must add up to 1.
- This constraint ensures that flight 101 is covered.
- Furthermore, the flight will be covered by only one type of fleet since the sum of binary decision variables adds up to 1.
- Only one of the two binary decision variables in this constraint will take a value of 1, forcing the other variable to be zero.
- We write similar constraints for all other 41 flights in our case study.

Aircraft Balance:

- The next set of constraints concerns the aircraft balance or equipment continuity within the fleets.
- This set of constraints ensures that an aircraft of the right fleet type will be available at the right place at the right time.
- Each node represents an arrival or departure. Recall that each node represents a specific time at a specific airport.
- So, the number of aircraft at any node changes with respect to an instant before that node.
- Just before this node, there were two aircraft (of the same fleet type) at the airport. After this arrival, we now have another aircraft (of the same fleet type again) added to those already at this airport.
- The set of constraints for aircraft balance or equipment continuity states that: Number of aircraft of a particular fleet type on the ground at a node = Number of aircraft in that fleet on the ground an instant before that node + arrival of aircraft of the same fleet type at that node – (minus) departures of aircraft of the same fleet type from that node.
- Number of aircraft at this node = 2 (number of aircraft before this node) + 1 (one arrival) – 0 (no departure from this node) = 3



Example of aircraft balance:

- Adopting this approach, we can now write the constraints for balance for each airport in our Ultimate Air case study.
- Let us consider LAX. The flights in and out of LAX we have two types of fleet.
- We use the decision variable $G_{k,j}$ to write the constraints for aircraft balance for each fleet type.
- Let us first consider the B737– 800 fleet type, the first node at LAX is at L1.
- The number of B737–800 aircraft at this node, based on the rule for balance, is basically the number of aircraft carried over from the previous day (wrap-around arc from node L6) minus one departure (flight 101), so:

Arrival/departure flights for LAX

Flight no.	Origin	Departure time	Destination	Arrival time	Duration of flight (hrs)
101	LAX	05:00	JFK	13:30	5.5
102	LAX	09:45	JFK	18:15	5.5
122	JFK	07:35	LAX	10:05	5.5
103	LAX	15:20	JFK	23:50	5.5
123	JFK	16:00	LAX	18:30	5.5
124	JFK	19:00	LAX	21:30	5.5

- Similarly, we write the balance constraints for all other airports in the schedule. There are 42 flights in our Ultimate Air case study.
- Each flight has a departure and an arrival. We have two fleet types. Therefore, the total number of constraints for aircraft balance is 168 (42 × 2 × 2).

Fleet Size:

- This set of constraints is adopted to ensure that the number of aircraft within each fleet does not exceed the available fleet size.
- To address this, we must count the number of aircraft that are grounded overnight for that fleet type at different airports.
- The last node, L6 (originating node for wrap around arc), represents the total

number of aircraft in LAX at the end of the day.

- For this airport, *GL6*, 1represents the total number of grounded B737– 800 aircraft in LAX overnight. Similarly, the number of grounded B757–200 aircraft at the last node in LAX is *GL6*, *2*.
- The total number of B737–800 aircraft in our network is therefore: In the above expression, the integer variables represent the number of aircraft at the last nodes at LAX, SFO, BOS, ORD, ATL, IAD, MIA, and JFK respectively.
- Note that at JFK, we have 42 daily flights arriving at or departing from this airport. Therefore, the last node is represented as *J*42.
- Similarly, the total number of B757–200 aircraft in our network.
- In our case study, Ultimate Air, assume that we have nine and six aircraft in our B737–800 and B757–200 fleets, respectively.
- We can now incorporate these constraints into our model as follows: Since there are only two fleet types, there are only two constraints in this set.

Solution to Fleet Assignment Problem:

- The linear integer program for fleet assignment for Ultimate Air has 252 (84 binary and 168 integer) variables and 212 constraints.
- Using optimization software, the solution to this problem generates a minimum daily cost of fleet assignment of$410,612.57.
- The following table shows the number of aircraft for each fleet type staying overnight at each airport.
- These numbers represent the right number of aircraft for each fleet type at the right airport at the right time.

Scenario Analysis:

- In this section we address some questions pertaining to the number of aircraft and different fleet combinations.

Case 1:

- It may be of interest to us to see what the minimum number of aircraft to cover all flights is.
- In this case, the objective function is modified to minimize the total number of aircraft.
- Therefore, the fleet size constraints are deleted from the set of constraints and become the objective function as follows:
- Running this integer/linear program results in 13 aircraft of which 9 are 737–800 and 4 are 757–200.
- According to this result, the minimum number of aircraft that are needed to fly the published Ultimate Air flights is 13.

Case 2:

- In this case, we evaluate various combinations of the two fleets.
- In our Ultimate Air example, we assumed that we have nine 737 and six 757 aircraft.
- The different costs associated with different number of aircraft combinations between the two fleet types.

Number of B737-800 aircraft	Number of B757-200 aircraft	Total daily cost
8	7	\$411,890
6	9	\$416,116
11	4	\$409,362
15	0	\$413,970
0	15	\$446,364

Total various costs for aircraft combinations

Fleet Assignment Model (FAM):

We now formally present the general mathematical model for the fleet assignment problem.

Sets:

F = Set of flights
K = Set of fleet types
C = Set of last-nodes, representing all nodes with aircraft grounded overnight at an airport in the network
M = Number of nodes in the network

Index:

i = Flight Index
j = Index for fleet
k = Index for nodes

- In the above model, the objective function in seeks to minimize the total cost of assigning the various fleet types to all the flights in the schedule.
- Constraints are the flight-cover constraints to ensure that each flight is flown by one type of fleet Constraints) are the aircraft balance constraints.
- The number of aircraft for any fleet type at any node.
- Before that node (represented in the model by $Gk–1,j$) plus the

arrivals(represented by $S_{i,k}$ taking a value +1) minus the departures (represented by $S_{i,k}$ taking a value of -1) Set of constraint represents the fleet size.

- The number of aircraft in fleet type *j*, should not exceed the available number of aircraft in that fleet (N_j).
- Constraints represent the binary and integer status of the decision variables. *Z+* is the set of positive integer numbers.

Crew and Manpower Scheduling Introduction:

- Crew scheduling involves the process of identifying sequences of flight legs and assigning both the cockpit and cabin crews to these sequences.
- Crew scheduling, like aircraft routing is normally performed after the fleet-assignment process.
- Total crew cost, including salaries, benefits, and expenses, is the second largest cost figure, after the cost of fuel, for airlines.
- The total number of crew, annual crew salaries and benefits, and flight-crew expenses for select US airlines.
- The third column in this table represents regular flight-crew salaries and benefits.
- The fourth column, flight- crew expenses, includes per diems and other expenses incurred for hotels, parking, meals, taxi-cabs, among others, in order for an airline to maintain its crew at a city other than their home base.
- Note that this cost is in addition to the salaries and benefits that the airlines pay to their flight crew.
- The last column shows flight-crew expenses as a percentage of salaries and benefits (column 4 divided by column 3).

Crew cost for US major carriers

Carrier	Number of flight crew[1]	Flight crew expenses[2] (000)	Crew expense/ operating expense[2] (%)
Alaska	1,455	180,845,000	5.57%
AirTran	1,632	157,383,851	6.00%
American	11,166	1,152,808,000	4.48%
Continental	4,867	623,767,000	4.05%
Delta	12,299	802,811,000	3.84%
Southwest	5,915	965,329,000	9.13%
United	6,478	757,020,000	3.44%
US Airways	5,275	482,044,882	3.39%

Source: Airline Pilot Central[1]; BackAviation Form 41 iNET.[2]

- Unlike the fuel cost, a large portion of flight-crew expenses are controllable.
- Even a small percentage of savings in flight crew expenses through better

scheduling translates into millions of dollars, which ultimately can determine the survival or demise of an airline.

- Because of such large anticipated savings, the crew scheduling problem has received considerable attention from both academia and industry.
- Crew scheduling is one of the most computationally intensive combinatorial problems.
- The crew scheduling problem is typically solved in two phases, crew pairing and crew rostering.

Crew Pairing:

- The first phase in the crew scheduling is to develop crew pairing. Crew pairing is a sequence of flight legs, within the same fleet, that starts and ends at the same crew base.
- A crew base is the home station or city in which the crew actually lives. Large airlines typically have several crew bases.
- The sequence of crew pairing must satisfy many constraints such as union, government, and contractual regulations.
- A crew pairing sequence may typically span from one to five days, depending on the airline.
- The objective of crew pairing is to find a set of pairings that covers all flights and minimizes the total crew cost.
- This assumption may be true for the week-day schedules, but for the weekends, the airlines normally have a lower frequency of flights.
- The adopted approach is normally to solve the crew pairing problem for a typical weekday, and then make modifications and adjustments for the week ends.
- Note that in this phase of crew pairing, we generate pairings of flight legs that are feasible and satisfy the regulations.
- In this phase, we do not address individual crew members. This phase is also referred to as an impersonal phase.
- The assignment of each specific crew member to these pairings will be discussed in the second phase, that is, crew rostering
- The following definitions are used in addressing the crew-pairing problem:

Duty:

- A working day of a crew may consist of several flight segments. The length of a duty is determined by Federal Aviation Regulations (FAR) in the United States, as well as by individual airline rules.
- Under the Federal law, airline pilots cannot fly more than 8 hours in a 24-hour period. They also must be able to rest for 8 hours in that same time span.

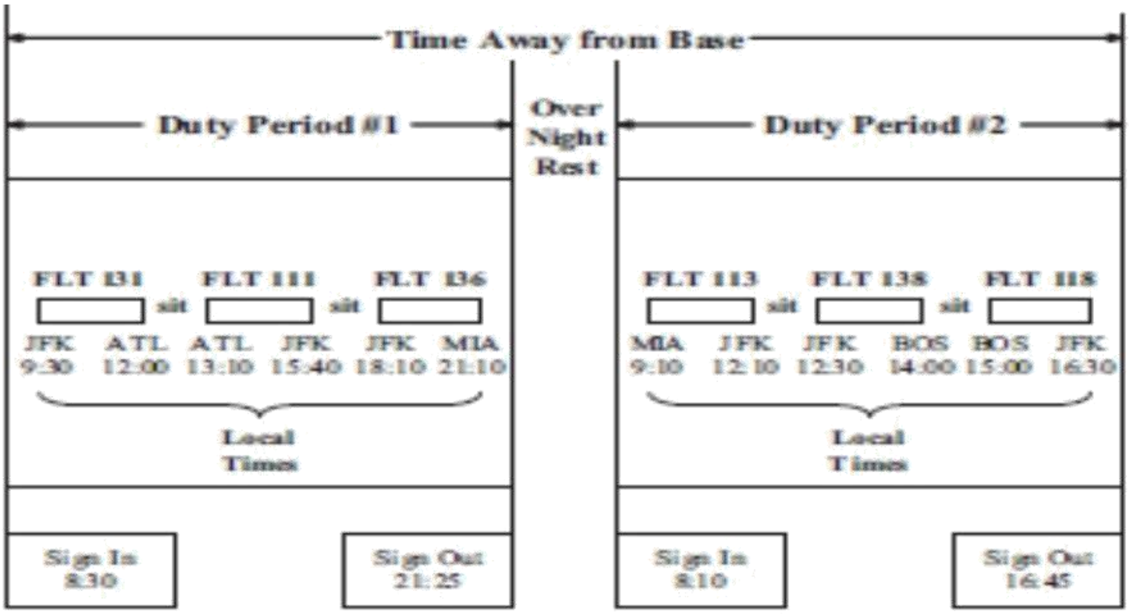
Sit connection:

- A connection during duty is called a sit connection. This involves the waiting times, on the part of the crew, for changing their next leg of duty.

- Normally, airlines impose minimum and maximum sit connection times, typically between 10 minutes and 3 hours

Rest:

- A connection between two duties is referred to as rest, overnight connection or layover.
- illustrates a sample from Ultimate Air's B757-200 fleet's two-day crew pairing, showing duty periods, sits within duty periods, overnight rests, and sign-in and sign-out times, assuming the crew home-base is at JFK.
- For our two-day pairing suggests, the crew is staying overnight, away from their home base, and therefore, the airline has to pay for their perdiems, transportation, accommodation, food, and so on.



A typical pairing with duty periods, sits within duty periods, overnight rests, and sign-in and sign-out times:

- The objective of the crew pairing problem is to minimize the total cost of assigning crews to flight legs, such that every flight is covered, and making sure that union, government, and airline rules are satisfied.
- Furthermore, the constraints should also consider the number of available crews at each base.
- This problem usually seeks pairings that translate into a high utilization of crew flying time, and minimum sit connection times.
- The airlines normally attempt to keep the crew with the same aircraft (tail number) on multiple flight legs as much as possible.
- This way, crew-related problems, such as delays and cancelled connecting flights, will be reduced Delayed, cancelled connecting flights, or other

difficulties in flight pairings result in deadheading.

- Deadheading happens when the crew is transported as non-revenue passengers.
- It should be noted that the solutions for the aircraft routing and crew pairing cannot be the same.
- First, crew members need more rest. An aircraft can be utilized for 14 hours in one day, but the crew can stay with the aircraft only8 hours.
- Second, crew pairing identifies flight legs that start and end at the same crew base (i.e., only JFK to JFK in our case).

Pairings Generators:

- The pairings are generated based on rules and regulations. It starts with a crew base and adds all the feasible flight legs according to the specified rules. It finally ends up at the same crew base from which it started.
- A pairing satisfying all the rules and regulations is called a legal pairing. The length of a pairing depends on the airline and union regulations. A pairing may span from one to five days.
- Some of the rules in generating the feasible pairings include the total daily flight time, and minimum and maximum sit-connection times.
- All possible feasible pairings are generated during this phase. For large airlines with many daily flights, the number of pairings generated becomes very large (billions of legal pairings!).
- This is especially very applicable to airlines with large hubs. Each flight leg at this hub can be potentially paired with many departing flights.
- This combination is compounded if the aircraft is rerouted to the hub several times in a day. In such cases, the generators are normally equipped with some extra rules and filters to identify and select good potential pairings.
- Barnhart (2008) and K labjan (2003) provide an overview of these rules and filters to reduce the number of pairings.
- The following represents the crew pairing requirements for Ultimate Air: Each duty should not exceed 8 hours of flight time.
- A maximum length of two days is allowed for a routing (i.e., two-day pairings).The home base for the crew is JFK. The minimum and maximum sit-connection times are 10 minutes and 3hours respectively.
- A similar program to route generators was developed to generate the potential crew pairings.
- steps for this program are as follows: Read the flight numbers, along with their departure and arrival cities and times, for a set of flights assigned to a specific fleet type (as identified by fleet-routing module)
- Create all possible one and two-day pairings – place in a file. Examine each pairing in this file so that: the pairing ends up at JFK over the routing cycle; for two-day pairing, the first flight of the second day starts out at the city where it ended up the night before; the duty does not exceed eight hours of

flight time in any given day; the sit-connection times are between the allowable minimum and maximum times.

- If a pairing satisfies all of the above conditions, it is added to a file of potential valid pairing candidates.
- This program generated a total of 28 and 314 legal pairings for the 757-200 and737-800 fleet types respectively.
- The main reason is that for crew pairing we generated only one- and two-day pairings as opposed to three-day routings.

Sample one-day crew pairing for B737-800 fleet

SAMPLE	DAY 1				Crew utilization (hrs)
High utilization					
Pairing #1	FLT 140	FLT 119	FLT 128	FLT 108	6
City-pairs	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	
Dept-Arr times	6:20-7:20	8:15-9:15	10:05-11:05	12:20-15:20	
Low utilization					
Pairing #2	FLT 140	FLT 119			2
City-pairs	JFK-IAD	IAD-JFK			
Dept-Arr times	6:20-7:20	8:15-9:15			

Sample two-day crew pairing for B737-800 fleet

SAMPLE	DAY 1			DAY 2			Crew utilization (hrs)
High utilization							
Pairing #3	FLT 142	FLT 121	FLT 127	FLT 104	FLT 142	FLT 121	15
City pairs	JFK-IAD	IAD-JFK	JFK-SFO	SFO-JFK	JKF-IAD	IAD-JFK	
Dept-Arr times	15:15-16:15	18:30-19:30	20:00-22:30	5:05-13:35	15:15-16:15	18:30-19:30	
Medium utilization							
Pairing #4	FLT 132	FLT 112	FLT 130	FLT 107	FLT 141	FLT 120	11
City pairs	JFK-ATL	ATL-JFK	JFK-ORD	ORD-JFK	JFK-IAD	IAD-JFK	
Dept-Arr times	14:35-17:35	18:00-20:30	21:00-22:00	7:30-10:30	12:00-13:00	14:25-15:25	
Low utilization							
Pairing #5	FLT 140			FLT 119			2
City pairs	JFK-IAD			IAD-JFK			
Dept-Arr times	6:20-7:20			8:15-9:15			

Objective Function:

- The determination of cost for crew pairings is a complex process. It is based on the sum of all duty cost in the pairing, cost of time away from the base, and minimum guaranteed pay multiplied by the number of duties.

- The maximum of these three costs determines the above cost function for each pairing. In our Ultimate Air example, we assume two-day pairings to be three times as costly as one-day pairings.
- This is because, in two-day pairings, the crew stays away from home base for one night, and hence the airline is responsible for the incurring costs.
- The cost coefficient is one for pairings 15 and 22, and three for all other pairings.

Flight-Coverage Constraints for B757 Fleet:

- Each pairing candidate covers a certain number of flights. We must ensure that the crew covers each flight exactly once. To write the coverage constraint for flight 125.
- This is because flight 125 only appears in crew pairing 1. Again, flight 114 appears in crew pairings 9, 12, 20, and 27.
- Therefore to cover this flight we have: we can write the flight coverage constraints for the other 10 flights with this fleet type.
- Note that unlike the three-day aircraft routing in which we had a constraint for each flight for each day, in crew pairing we address each flight only once.
- This is because we are interested in knowing which flights should be paired rather than the actual assignment of flights to days.
- A two-day pairing requires two sets of all flights are covered. We will discuss the assignment of pairings to days in the second phase, crew rostering.

Crew-Pairing Solution for B757-200 Fleet:

- We used optimization software to solve the above integer linear program model. Four two-day pairings were selected.
-

Solution to crew pairing for B757-200 Fleet:
Crew Pairing Solution for B737-800 Fleet:

- Similarly, we develop the mathematical model for crew pairing of the 737-800 fleets. Solving this mathematical model generates.

Crew Rostering:

- Once the crew pairing problem is solved, the second phase is crew rostering.
- Crew rostering is the process of assigning individual crew members to crew pairings, usually on a monthly basis. Some airlines, mainly European, allow their crews to select a number of pairings as identified in the first phase, together with rest periods on specific days to construct their monthly personalized schedule
- The airline then attempts to grant these schedules if possible. Crew training days, seniority, and other internal regulations are some of the factors that

influence the assignment of these schedules to crews.

- US Airlines, however, develop their monthly crew schedules based on the solutions generated in the crew-pairing phase, independent of crew desires.
- This approach is then used to construct the monthly schedule by incorporating employee time off, training, union rules, and other contractual obligations.
- The airlines then assign crews to these schedules based on their in-house priority system.
- In both rostering systems, the objective is to maximize crew utilization, evenly distributing individual crew workload and rest times.
- Since the rules and regulations vary among the airlines, the crew rostering process, and the available literature on this topic, is also diverse.
- Some of these methods include assigning high priority employees to high priority pairings; developing monthly rosters for individual crew members based on their requests; developing monthly rosters for each day of the month without considering the crew requests.
- It should be noted that the processes of assigning cockpit-aircrew members (captain and first officer) and cabin-aircrew members (flight attendants) are typically different.
- The cockpit aircrew members usually have the required licenses/type ratings to fly only a specific fleet of aircraft, while cabin aircrew members can be assigned to multiple fleet types.

Ultimate Air Rosters:

- As explained earlier, a roster is a series of crew pairings separated by rest periods and days off. For Ultimate Air, we attempt to develop anonymous rosters on which its employees can bid.
- For presentation purposes, and in an effort to keep the rostering problem to a manageable size, we will develop the rosters on a weekly basis, instead of monthly rosters which are more common among airlines.
- The process of developing monthly rosters is basically the same as that of one done weekly.
- The assumptions for the Ultimate Air crew rosters are as follows: at least one day off between pairings; two pairings per week; balanced workload among all rosters – a work week of 20 flight hours is desirable.
- Each ( ) symbol represents a pairing. Note that each pairing spans a two-day period. Therefore, if a crew is assigned to a pairing on Monday, then this crew member will be flying both on Monday and Tuesday.
- Since we require at least one day rest between pairings, this crew member cannot fly on Wednesday, but can fly on Thursday, Friday, Saturday or Sunday.

Possible weekly crew roster combinations for Ultimate Air

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
✓			✓			
✓				✓		
	✓			✓		
	✓				✓	
		✓			✓	
		✓				✓
			✓			✓

- We assume that the assignment of crew to rosters, in each week, takes into consideration their previous week's rosters.
- That is to say, if a crew member is assigned to a pairing which starts on Saturday of this week, this crew member cannot be assigned to a roster which starts on Monday, and so on. Similar to the crew pairing mathematical model in the previous section, a series of set- partitioning approaches is adopted to assign rosters to individual crew members.
- We use a set-partitioning approach first to identify the anonymous rosters.
- In this approach, the rows of the set-partitioning matrix represent the valid roster combinations, and the columns are the daily pairings, which span the entire week rosters for this fleet first.

Crew Rosters for B757-200 Fleet:

- Four pairings were identified, which covered all the scheduled 757-200 flights in a day.
- Let us call these four pairings P1, P2, P3, and P4. Considering these pairing combinations, and assigning these pairings to days we get112 possible valid rosters.
- This presents three sample valid rosters with corresponding total weekly flight hours.

Three sample rosters for B757-200 fleet

Sample Rosters	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Flight hrs
1	P1			P2				25
2			P3			P1		17
3		P4			P2			19

Decision Variable:

- Similar to crew pairing, the task of this mathematical model is to identify which rosters, among the 112 potential candidates, should be selected.

Objective Function:

- The objective function is therefore constructed in an attempt to minimize the total deviations of the rosters' weekly flight hours from the target of 20 flight weekly hours where:
  h_j= the total weekly flight hours for roster j.
- We use absolute values because the term h_j– 20 may be positive, zero, or negative depending on the roster. In this manner, a negative deviation (low weekly flight hours) is treated as bad as a positive deviation (high weekly flight hours).
- The coefficient for the variable representing sample 1 is |25-20|=5. Similarly, the objective function coefficients for the other two samples are |17-20|=3 and |19-20|=1 respectively.

Pairing Coverage Constraints for B757-200 Fleet:

- Each roster candidate covers a certain number of pairings in each day. We must ensure that the rosters cover each pairing every day, exactly once.
- As an example, sample 1 covers pairings 1 and 2 on Monday and Thursday respectively.
- So this sample is a candidate to cover P1 on Monday and P2 on Thursday.
- We have four pairings that need to fly every day of the week, which makes a total of (4 × 7) 28 constraints as follows: In this set of constraints, index i represent a specific pairing in a given day.

- As an example, the number 1 represents P1 on Monday, while 2 stands for P2 on Monday and 28 is P4 on Sunday. The parameter $a_{i,j}$is defined as follows:

Rostering Solution for B757-200 Fleet:

- The above integer linear program with 112binary decision variables and 28constraints was solved using optimization software.
- The solution for the objective function is 28 hours, which represents the sum of deviations of all rosters from our target of 20 flight hours.
- There are 14 disjointed (non-overlapping) rosters, each covering two pairings per day.
- As we can see from the solution, each pairing is covered exactly once every day.
- In order to keep the flight hours more balanced, one approach is to rotate the rosters every week among the crew members.
- This rotation of weekly rosters not only provides a fair and balanced number of flight hours over the whole month for a particular crew member, but is also very desirable for the airlines and crew to stay current with their network of airports.

Solution to crew rosters for B757-200 fleet

Rosters	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Flight Hours
1	0	P1	0	0	P3	0	0	17
2	0	0	P1	0	0	P3	0	17
3	P1	0	0	P4	0	0	0	16
4	0	0	0	P2	0	0	P3	20
5	P2	0	0	0	P4	0	0	19
6	0	P2	0	0	0	P4	0	19
7	0	0	P2	0	0	0	P4	19
8	0	P3	0	0	P1	0	0	17
9	0	0	P3	0	0	0	P1	17
10	P3	0	0	0	P2	0	0	20
11	0	0	0	P3	0	0	P2	20
12	P4	0	0	P1	0	0	0	16
13	0	P4	0	0	0	P1	0	16
14	0	0	P4	0	0	P2	0	19

- We need at least 14 captains and 14 first officers for our757-200 fleet. The airlines normally have a number of reserve captains and first officers to accommodate unforeseen circumstances.
- As explained earlier, these are anonymous rosters and can be assigned to any crew member.

- Once these rosters are constructed, the airline, based on its rules and regulations, assigns them to each individual crew member.

Rostering Solution for B737-800 Fleet:

- A similar approach is adopted for deriving the solution for the 737-800 fleet.
- We have nine pairings for this fleet. There are a total of 567 roster candidates and 63(9 pairings × 7 days/week) constraints.

Solution to crew rosters for B737-800 fleet

Rosters	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Flight Hours
1	0	0	P1	0	0	0	P7	19
2	P1	0	0	0	P9	0	0	19
3	0	P1	0	0	0	P9	0	19
4	0	0	0	P1	0	0	P9	19
5	P2	0	0	P2	0	0	0	22
6	0	P2	0	0	0	P5	0	20
7	0	0	P2	0	0	0	P5	20
8	P3	0	0	P4	0	0	0	18
9	0	P3	0	0	P4	0	0	18
10	0	0	P3	0	0	P4	0	18
11	0	0	0	P3	0	0	P4	18
12	P4	0	0	0	P3	0	0	18
13	0	P4	0	0	0	P3	0	18
14	0	0	P4	0	0	0	P3	18
15	P5	0	0	0	P6	0	0	20
16	0	0	P5	0	0	0	P6	20
17	0	P5	0	0	0	P8	0	20
18	0	0	0	P5	0	0	P8	20
19	0	0	0	P6	0	0	P1	19
20	P6	0	0	P8	0	0	0	22
	0	P6	0	0	P8	0	0	22
22	0	0	P6	0	0	0	P9	22
23	0	0	0	P7	0	0	P2	22
24	0	P7	0	0	0	P6	0	22
25	0	0	P7	0	0	P7	0	22
26	P7	0	0	P9	0	0	0	22
27	P8	0	0	0	P1	0	0	19
28	0	P8	0	0	P2	0	0	22
29	0	0	P8	0	0	P2	0	22
30	0	0	P9	0	0	P1	0	19
31	0	P9	0	0	P5	0	0	20
32	P9	0	0	0	P7	0	0	22

Crew-Rostering Mathematical Model:

- The mathematical model for crew rostering depends on how we choose to construct the rosters, that is, either individualized or anonymous rosters.
- The approach that was presented in this chapter was based on developing anonymous rosters.

Sets*:*

P= Set of all pairings over all days of the roster period

R = Set of valid rosters

Indices:

j = Roster index *i* = Pairing index *Parameters:*

cj= Deviation of roster *j* flight time from a target value

- In this model, the objective function attempts to minimize the total sum of deviations. Constraint guarantees that each flight pairing in each day is covered only once.

MANPOWER SCHEDULING:

Introduction:

- An airline's product is measured by its timeliness, accuracy, functionality, quality, and price. The airline employees and equipment are the factors that determine such measures.
- Manpower planning for airlines represents one of the most important and challenging tasks, covering a wide range spanning from hiring, training, to scheduling of human resources.
- The concepts of hiring and training are normally very much dependant on the airline strategic plans.
- Manpower scheduling refers to the actual work plan including working, non-working days, times, shifts, locations, and leave periods. Scheduling the employees for an airline is an enormous task.
- There are pilots, flight attendants, ground crew, baggage handlers, reservationists, cooks, janitors, mechanics, administrators, and so on.
- The main purpose of manpower scheduling is to derive a cyclic (normally weekly) plan for each employee so that the total manpower costs are minimized, efficiency and utilization are maximized, subject to meeting the requirements and regulations, on crew scheduling, presented the process of assigning flight crews to flight legs.

Mathematical Modeling Case Study:

- The weekly manpower requirements for ground operations (check in counters and baggage handlers) at JFK for our Ultimate Air airline example.
- The weekly manpower requirements are normally different at different times of the day and different days of the week.
- The daily operations are divided into four time blocks with duration of four hours on each Mondays from 6 a.m. – 10 a.m., we need eight employees, and so on.
- The following contractual issues and airline policies apply: Each employee works for eight hours consecutively in a day.
- There are currently three working shifts: shift 1 (6 a.m. – 2 p.m.), shift 2 (10

a.m. – 6p.m.) and shift 3 (2 p.m. – 10 p.m.).

- Each employee works for five days consecutively followed by two days off.

Index for shifts (j)

8-hour shift	Index (j) for shift
6 a.m. – 2 p.m.	1
10 a.m. – 6 p.m.	2
2 p.m. – 10 p.m.	3

Index for days of the week (i)

Starting day of the working week	Index (i) for day
Mon	1
Tues	2
Wed	3
Thu	4
Fri	5
Sat	6
Sun	7

Check-in counter agents requirement at JFK for Ultimate Air

Shift/day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
6 a.m. – 10 a.m.	8	8	8	8	10	10	6
10 a.m. – 2 p.m.	12	10	12	10	16	16	8
2 p.m. – 6 p.m.	16	12	16	12	20	20	8
6 p.m. – 10 p.m.	9	8	9	8	12	12	4

- The objective is to determine the minimum size for the workforce and their working schedules so that the above manpower requirements and regulations are met.
- The mathematical approach discussed in this section is a modified version of

the Personnel Scheduling model by Brusco et al. (1995).

- This method has been used in the development of the automated manpower planning system at United Airlines.
- For other mathematical approaches to manpower planning see Brusco and Jacobs
- We adopt the following decision variable: x_i, j = number of employees who begin their weekly work in day i adopting shift j In this decision variable, index i, represents the day that an employee starts his/her five-day work week. Index j, represents the shift that the employee is assigned to. Tables 7.2 and 7.3 show the indices used to represent shifts and days of the week respectively
- The number of employes should start their work week on Monday from 6 a.m. – 2 p.m. shift, and soon. The objective function is to minimize the total workforce(headcount)as follows:
- Note that the employees in decision variables are disjoint, meaning no employee appears in two decision variables.
- As an example, those employees who start their working week on Monday from 6 a.m. – 2 p.m., represented by decision variable $x1,1$, are different from those who start on Monday from 10 a.m.– 6p.m. represented by $x1,2$. So adding all the decision variables represents the total workforce for this case study, which we wish to minimize.
- For the constraints, we should satisfy the manpower requirements for each time block of the day.
- We have seven days with four time blocks covering the three shifts in each day, resulting in a total of 28 constraints. We classify these constraints in their four respective time blocks.

Constraints:

- The constraints must cover the manpower requirements for every shift of everyday. The following presents the constraints for each time block.

Time Block 6 a.m. – 10 a.m.

- The employees working in this time block include only those who start their shift at 6 a.m. (first shift).
- Those who start their shifts at 10 a.m. or 2 p.m. (second or third shifts) will not be present during this time block.
- To express this constraint for Monday, 6 a.m. – 10 p.m., we have the following constraint.
- The above constraint specifies that the total number of employees available for work on Monday from 6 a.m. to 10 a.m. includes those who start their working week on Monday ($x1$, 1), plus those who start on Thursday ($x4$, 1), Friday ($x5$, 1), Saturday ($x6$, 1), and Sunday ($x7$, 1).
- We require eight employees on Monday in the first time block. This number appears as the right hand side for the constraint.

- Each employee works five days consequently followed by two days off, those who start their working week on Tuesday or Wednesday will not be present for work on Monday.

Constraints for Time Block 10 a.m. – 2 pm:

- Since each employee works for eight hours, then the employees working in this time block include those who start their shifts at 6 a.m. (first shift) and 10 a.m. (second shift). The constraint for this time block for Monday is as follows:
- The first five terms are the same as the constraint for Monday 6 a.m. – 10 a.m. time block. The second five terms represent those employees who start their shift sat 10 a.m. on different days.

Constraints for Time Block 2 p.m. – 6 p.m:

- The employees working in this time block include those who start their shifts at 10a.m. (second shift) and 2 p.m. (third shift). The constraint for this time block for Monday is as follows:
- The first five terms are the same as the constraint for Monday 10 a.m. – 2 p.m. time block. The second five terms represent those employees who started their shifts at 2 p.m. on different days.
- The right hand side represents the number of required employees for Monday's third time block. Similarly six more constraints are added for other days of the week

Constraints for Time Block 6 p.m. – 10 p.m:

- The employees working in this time block include only those who start their shifts at 2 p.m. (third shift).
- Those who have started at 6 a.m. (first shift) or 10a.m. (second shift) have already finished their 8-hour working day and are not present during this time block. The constraint for Monday's time block is as follows
- We see that only those employees with the third shift appear in this constraint. Similarly six more constraints are added for other days of the week.

Solution:

- The above linear integer programming model has 21 integer-decision variables and 28 constraints.
- A total of 36 employees are required to meet the manpower requirement

Solution to manpower planning

Day/shift	Shift 1 (6 a.m. – 2 p.m.)	Shift 2 (10 a.m. – 6 p.m.)	Shift 3 (2 p.m. – 10 p.m.)
Mon	2	1	3
Tue	4	0	7
Wed	0	1	0
Thu	2	4	4
Fri	2	0	0
Sat	2	2	2
Sun	0	0	0

Mathematical Model:

- The mathematical model proposed by addresses both part- time and full-time employees, their limits, numerous combinations of shifts, working days, and weekly rotations.
- This method has been used in the development of the automated manpower planning system at the United Airlines called Pegasus.
- This automated system aids the airline in determining the optimal manpower planning system in their 119 domestic airports as well as many international locations.
- Pegasus uses flight schedules, passenger forecasts, baggage and cargo loads to compute labor requirements.
- The mathematical model for this automated system utilizes personnel tour scheduling which involves the determination of work and non-work days during the week as well as the associated daily shift starting and finishing times for each employee. The mathematical model is as follows:

Sets:

D = Set of days in the weekly planning
S = Set of allowable shifts
T = Set of all time-blocks in the weekly planning *Index*
i = Index for day in the weekly planning J = Index for shift
k = Index for time block

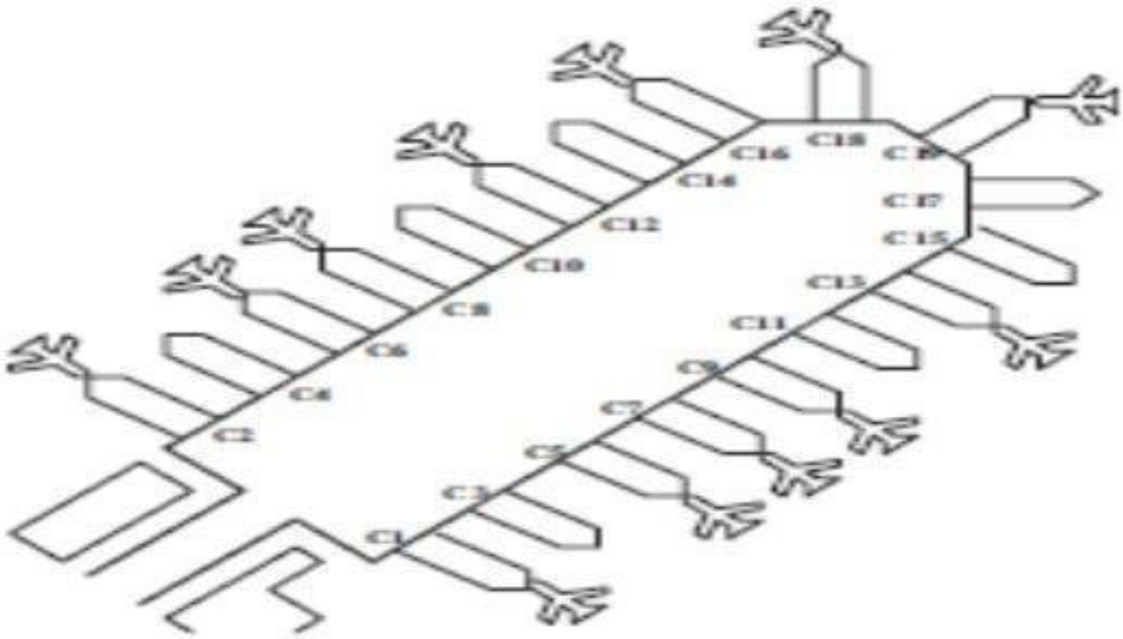
UNIT V
GATE ASSIGNMENT AND AIRCRAFT BOARDING STRATEGY

Introduction:

- The hub-and-spoke system has resulted in a large volume of baggage and passengers transferring between flights.
- Assigning arriving flights to airport gates is therefore an important issue in daily operations of an airline.
- Although the costs of these activities are generally small portions of the overall airline operation costs, they have a major impact on maintaining the efficiency of flight schedules and passenger satisfaction.
- Some of the factors that impact the assignment of gates to arriving flights include aircraft size, passenger walking distances, baggage transfer, ramp congestion, aircraft rotation, and aircraft service requirements.
- The problem of finding a suitable gate assignment is usually handled in three levels. In the first level, the ground controllers use the flight schedule to examine the capacity of gates to accommodate these flights.
- The second level involves the development of daily plans before the actual day of operation.
- In the third level, because of irregular conditions such as delays, bad weather, mechanical failure and maintenance requirements, these daily plans are updated and revised on the same hour/day of the operation.
- The problem of gate assignment is well studied in operations research. A common approach in formulating this problem is from the passenger's perspective in a way that the total passenger-walking distance is minimized.
- The gate assignment problem (GAP) is defined as follows: Given a set of available gates and flights, the distance matrix between the gates, the passenger transfers the gates so that the total passenger-walking distances are minimized.

Mathematical Model for a Case Study:

- The following case study (not related to Ultimate Air!) involves the assignment of flights to gates.
- The C Concourse at San Francisco (SFO) Airport, which has 19 gates (C1-C19). There are already 12 aircraft at the gates (as shown) getting ready for their departures.
- Within the next 15 minutes seven flights will be arriving in this concourse that should be assigned to the remaining gates.
- These flights are referred to as F1, F2, F3, F4, F5, F6, and F7.
- In these seven flights, there are passengers who will connect to other departing flights.



- The number of passengers in these flights who will connect to other departing gates.
- As an example, five passengers from flight F1 should walk to gate 1, and so on.

Passenger flow

Flight	Departing gates																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
F1	5	5	10	8	15	8	2	10	8	20	5	4	0	9	3	4	1	2	1
F2	5	2	1	4	19	9	4	2	3	2	27	3	8	4	0	2	1	7	2
F3	10	0	4	9	13	4	4	4	3	5	5	8	4	9	11	7	9	4	4
F4	4	8	5	4	10	4	1	0	0	2	4	19	1	2	4	5	5	8	2
F5	4	11	9	9	6	3	1	4	4	2	1	0	3	5	1	2	2	3	4
F6	1	2	42	5	2	7	6	2	4	7	2	3	6	4	10	2	1	0	0
F7	3	3	2	5	9	13	11	2	2	3	7	22	4	0	1	1	2	2	9

- Note that in this matrix, only the distances between the candidate arrival gates and other gates are shown.

Distance matrix (yards)

Gates	Departing Gates																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
3	10	40	-	30	10	40	20	50	30	60	40	70	50	80	60	90	70	90	80
4	40	30	30	-	40	10	50	20	60	30	70	40	80	50	90	60	90	70	80
10	70	40	60	30	50	20	40	10	30	-	40	10	50	40	60	30	70	40	50
11	50	80	40	70	30	60	20	50	10	40	-	30	10	40	20	50	30	50	40
14	90	60	80	50	70	40	60	30	50	20	40	10	30	-	40	10	50	20	30
15	70	100	60	90	50	80	40	70	30	60	20	50	10	40	-	30	10	30	20
17	80	100	70	90	60	80	50	70	40	60	30	50	20	40	10	30	-	20	10

- Through this model we seek to assign the arriving flights to candidate gates so that the total passengers' walking distance is minimized.
- We can find the total walking distances of passengers on flight *i* if this flight is assigned to arrival gate *j*. The walking distance is calculated as follows:

Traveling distances (yards)

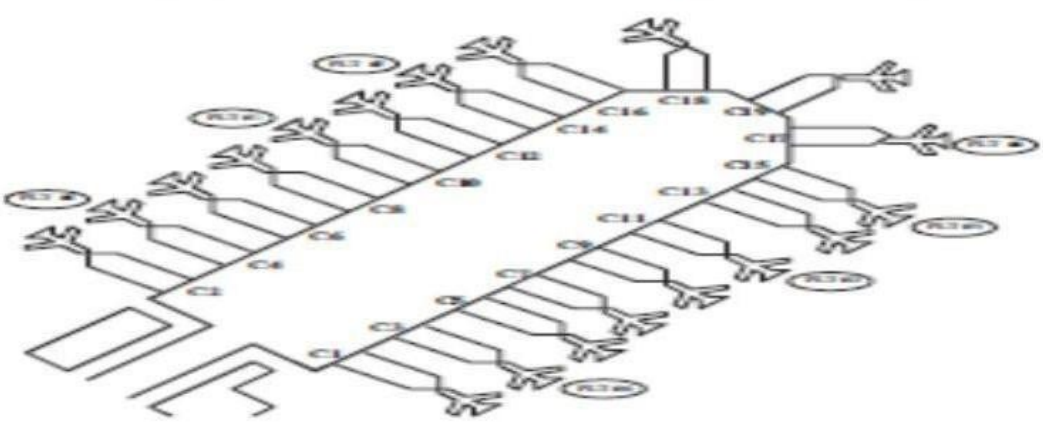
Flight/gate	3	4	10	11	14	15	17
F1	5,010	4,390	3,820	4,870	5,060	6,650	7,090
F2	4,240	5,290	4,190	3,020	4,650	4,400	4,970
F3	5,610	5,950	4,930	4,270	4,910	4,950	5,320
F4	4,500	3,990	3,280	3,580	3,460	4,320	4,460
F5	2,950	2,720	3,060	,3490	3,620	4,330	4,530
F6	3,060	4,310	4,740	3,900	5,760	5,300	6,020
F7	4,680	4,380	3,290	3,620	3,970	4,960	5,220

- Walking distance = number of passengers _ × distance For example, if flight F1 is assigned to the candidate arrival gate 3, then the total walking distance for all passengers on this flight assigned to this gate is calculated as follows:
- In other words, by assigning flight F1 to gate 3, five passengers on this flight must walk a distance of 10 yards each to gate 1, five passengers must walk 40 yards each to gate 2; and 10 passengers will depart from the same gate as they arrived and therefore not having to walk.

- We repeat the above calculations for every flight assigned to every candidate gate.
- We define the following binary decision variable:
- The objective function is therefore:
- For constraints, we should ensure that every flight is assigned to a gate. We have seven gates (3, 4, 10,11,14,15, and 17) available. The constraint for flight F1 is the above constraints ensure that flight F1 is assigned to one and only one gate among the seven available gates.
- Similarly we write the other six constraints for other flights. If we run this integer linear program with the above constraints we see that one gate is assigned to two or more flights at the same time; rendering it not feasible.
- So we must ensure that each gate is also assigned to one flight (aircraft) only. The following additional set of constraints imposes this restriction for gate 3.Similarly we write the constraints for other the six gates.
- The above integer linear programming model has 49 binary decision variables and 14 constraints.
- Solving this problem using an optimization software generates the following matching flights to gates solution.
- The total walking distance for this optimal solution among all passengers is 26,000 yards shows the allocation of these gates to flights.
- Now, we can relax the assumption that any gate can accommodate any aircraft. Let us assume that gates 10 and 14 cannot be used for the aircraft in flight F1.

Solution to gate assignment

Flight	Gate assigned to
F1	10
F2	11
F3	15
F4	17
F5	4
F6	3
F7	14



Assignment of gates to flights

- To address this, simply add the following constraints to restrict the assignment of gates 10 and 14 to flight F1.
- Running the model with these new constraints generates the following solution with a total walking distance of 26,700 yards.

Revised assignments of gates to flights

Flight	Gate assigned to
F1	4
F2	11
F3	15
F4	14
F5	17
F6	3
F7	10

Baggage Handling:

- The above model considers only the flow and movement of passengers. The introduction of hub and spoke concept has represented the airlines with challenging and demanding task of baggage handling for transit passengers.
- Unlike the passengers who can typically walk from one gate to another, the bags actually need to be transported from one gate to another for these transit passengers.
- The transportation of baggage poses many challenges to airlines, including scheduling the number of baggage handlers, baggage trailers, delays, lost baggage, and missed connections. In fact for major airlines baggage handling for transit passengers seems to be the dominant factor in gate assignment in their major hubs.
- The airlines normally assign their baggage handlers and trailers according to the ascending order of departure time.
- The concept of baggage handling has been studied under different scopes. Some of these studies focus on baggage handling for security purposes and detection of explosives.
- These studies include Mc Lay et al. 2006 and Jacobson et al. 2005. Others study the baggage handling system under gate assignment.

Mathematical Model for Baggage Handling:

- The mathematical approach presented earlier in this chapter for gate assignment is revised to incorporate baggage handling distances too.
- In this revised model the objective is to assign gates to arriving flights so that the total traveling distance for transit passengers and bags is minimized.
- Referring to the above case study for passenger flow presents the amount of transit bags, mail, and cargo from each of the arriving aircraft to departing gates.

Baggage flow from arriving flights to departing gates (units of baggage)

Flight	Departing gates																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
F1	19	28	11	8	30	25	33	5	40	14	38	38	14	23	17	4	20	44	8
F2	43	40	22	29	4	49	8	6	20	21	17	5	27	29	29	40	42	34	25
F3	22	17	36	45	22	28	17	23	18	44	12	8	41	48	25	11	27	47	28
F4	47	11	4	26	16	21	24	8	45	22	45	20	14	22	32	32	9	39	7
F5	3	24	46	38	48	7	24	33	29	43	7	21	45	47	28	11	17	3	23
F6	9	47	18	3	44	14	4	27	34	38	17	26	2	3	28	40	11	8	46
F7	46	34	48	42	26	12	40	49	18	36	24	6	18	9	2	30	14	47	9

- Again in this model it is assumed that the departing flights are initially, or tentatively have been, designated to gates.
- The baggage is normally transported by baggage trailers from gates to gates on the ramp.
- We assume that the capacity of the trailer is 5 bags per trailer. Therefore we can convert the number of baggage flow in to the number of trips that the trailers need to make in order to distribute the baggage among the gates.
- It presents this baggage flow in numbers of trips for trailers. These figures have been all rounded up. As an example, in we have 19 baggage units to be transported from flight F1 to gate 1.
- The capacity of the trailer is 5 bags. Therefore, the number of trips that the trailer needs to make to move these bags is given by where [= the rounded-up integer for •.The distances in yards between the gates on the airport ramp are presented in.
- In this matrix, only the distances between the candidate arrival gates and other gates are shown.
- Similar to the process described for calculating the passenger walking distance, we can calculate the distance traveled by trailers to transport baggage from arriving flights to departing gates, as follows:
- Baggage transport distance = Σ number of trips × distance.
- The following table presents the total baggage transport distances in yards for each of the arriving flights to each of the candidate gates.
- We can now revise our objective function to accommodate for both passenger

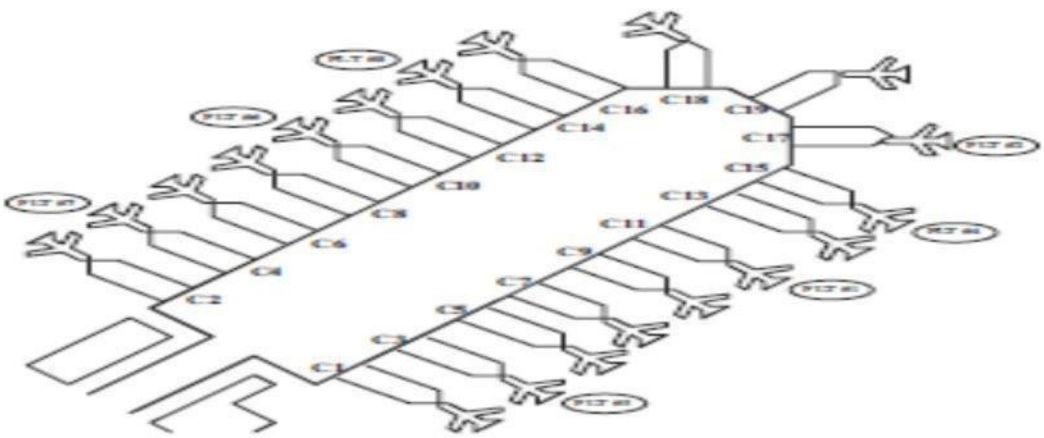
and baggage traveling and transport distances. The total distance can be represented as: Total distance = w1(passenger traveling distances) + w2(baggage transport distances)

- In this revised total distance, we assign weights $w1$ for passenger traveling and$w2$ to baggage transportation distances respectively. As indicated earlier, the airlines probably assign a higher weight to transport of baggage than to passenger traveling distances. For this case study, we assume the weights to be $w1 = 1$ and $w2 = 3$ respectively. Therefore the objective function is:

Solution to gate assignment for both passenger and baggage Transport:

Flight	Gate assigned to
F1	11
F2	17
F3	14
F4	15
F5	3
F6	10
F7	4

Mathematical Model:



Assignment of gates to flights:

Sets:

F = set of arriving flights

G = set of available gates for arriving flights K = set of departing gates *Parameters*

$p_{i,k}$= number of passengers arriving on flight i and departing from gate k

$dp_{k,j}$= distance units (in yards, meters, feet, etc) for passengers from gate k to gate j

$TP_{i,j}$= Total walking distance for all passengers on flight i assigned to arrival gate j

$t_{i,k}$= number of trips to transport baggage from flight i to departing gate k $db_{k,j}$= distance units (in yards, meters, feet, etc.) to transport baggage on ramp from departing gate k to arriving gate j

$TB_{i,j}$ = Total transport distance for all baggage on flight i assigned to arrival gate j

$w1$, $w2$ = Weights assigned to total passenger walking and baggage transport distances respectively

$TP_{i,j}$ and $TB_{i,j}$ are calculated as follows:

Special Cases:

- If there are more gates than arriving flights, then constraint. The above inequality denotes that an arriving gate can be assigned to a flight by taking a value of 1 or will not be assigned to any flight at all by taking a value of zero.
- If, on the other hand, there are more flights than arriving gates then mathematically we can write an inequality for constraint similar to the previous special case. However, it will not be realistic.
- Each flight must land and be accommodated at a gate. If there are no gates available for an arriving flight, as sometimes is experienced in busy airports, then the aircraft has to wait on the ramp or taxiway until a gate becomes available.

Aircraft boarding strategy:

Introduction:

- The airlines are currently undergoing difficult financial times. The increase in fuel prices, competition from low cost carriers, and operational inefficiencies have resulted in bankruptcies and major losses for airlines around the world.
- It is therefore extremely important for the airlines to be efficient in areas that they have control over. Airlines generate revenue by utilizing and flying their aircraft; they do not generate any revenue while their aircraft are on the ground.
- The time from the arrival of the aircraft until its next departure constitutes turnaround time. To have high utilization of their aircraft, airlines attempt to minimize the turnaround time.
- The typical aircraft turn- around time for short-haul flights is approximately 30–60 minutes. A major component of turnaround time is the passenger boarding time.
- Horst Meier and Haan (2001) provide detailed analyses of all the components in an aircraft turn-around time including passenger deplaning, refueling, cleaning, catering, maintenance, and boarding.
- Because of safety and operational constraints, passenger boarding is the last task in this timeline. Any time saved through efficient boarding directly reduces the turnaround time.

Common Strategies for Aircraft Boarding Process:

- Airlines seem to adopt different aircraft boarding strategies based on airline culture and service level.

- Some airlines do not impose any strategy and let the passenger's board randomly.
- Others arrange passengers into groups, zones or call-offs based on specific boarding strategy adopted by the airline.

Back-to-Front:

- Back-to-front (BF) boarding strategy is widely adopted by many airlines for both narrow and wide-body aircraft. In this strategy, first class, business class, and special-need passengers are boarded first.
- Then, as the name implies, passengers start filling up the aircraft from back to front. Passengers are called to board the aircraft based either on their seat row numbers or by groups or zones.
- Each group is then called in sequence to board the aircraft. The back-to front boarding strategy where all the passengers on this aircraft are divided into six groups. Then passengers in groups 2 to 6 are called to board the aircraft.

Window-Middle-Aisle:

- Window-middle-aisle boarding strategy (or sometimes called out-in), as the names implies, boards the passengers in window seats, middle seats and finally in the aisle seats.
- Passengers are usually divided into four groups to follow this boarding strategy.
- First, business class and special need passengers are assigned to group 1 and board first.
- Then all the economy class passengers in window, middle, and aisle seats are assigned to groups 2, 3, and 4 respectively and board the aircraft according to their assigned groups.
- A major disadvantage of this boarding strategy is that the passengers in parties of two or more seated next to each other board the aircraft separately and at different times. This boarding process may not appeal to either passengers and/or airlines.

Random:

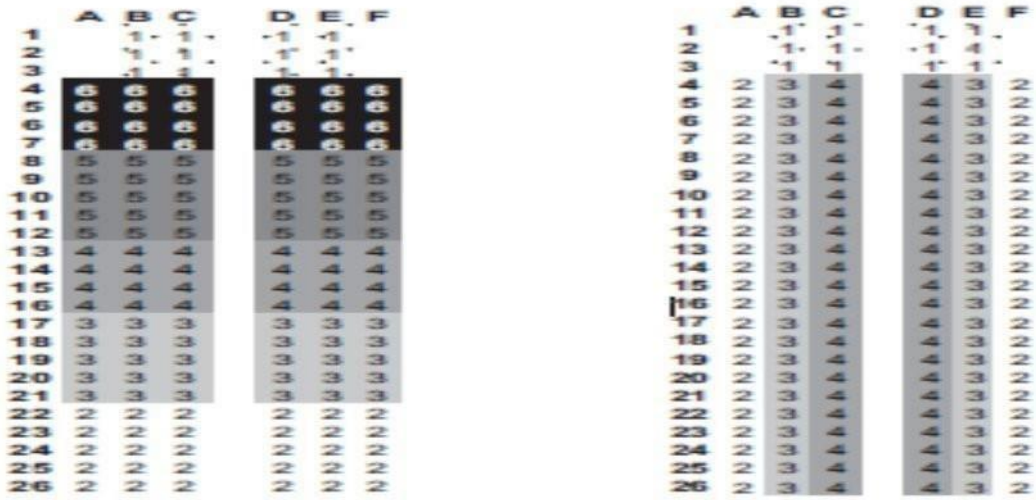
- In random boarding strategy, no specific strategy is used and all passengers aboard the aircraft in one zone randomly.

Rotating Zone:

- In rotating zone, passengers are grouped into zones and board the aircraft first in the front, then in the back, then front again, then back in a rotating manner.
- In this boarding strategy, passengers sitting in the middle of the aircraft are seated last.

Mathematical Model:

- Most of the studies on aircraft boarding strategies focus on modeling the problem using computer-based simulations (see, for example, Ferrari et al. 2004,Ferrari



Sample of back-to-front and window-middle-aisle boarding process

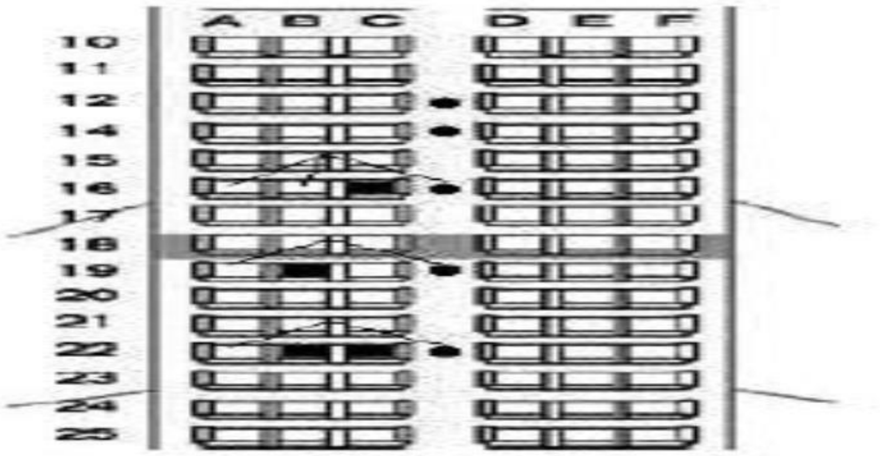
- While these methods provide a good understanding of existing boarding strategies and enable us to evaluate various known strategies and conduct what-if scenarios, they do not help us find the best and other possible unknown alternatives.
- Analytical approaches can help achieve these alternatives. Some of the existing analytical models include: Van den Briel et al. (2005) proposed a non-linear assignment model with quadratic and cubic terms.
- The model attempts to minimize the total seat and aisle interferences among passengers (discuss edlater); Bachmat et al. (2006) derived a family of back-to-front boarding policies using stochastic geometry under the assumption of passengers being infinitely thin.
- The application of this model to a specific aircraft or airline has not been reported. Bazargan (2007) adopted a binary/integer linear program to minimize the total seat and aisle interferences.
- The analytical model on a binary/integer linear program introduced by Bazargan (2007).

Interferences:

- Boarding interferences occur when a passenger blocks another passenger from proceeding to his or her seat. Two types of interferences, seat interferences and aisle interferences may occur.
- Seat interferences occur when a passenger blocks another passenger assigned to the same row
- In all these cases, the blocking passenger(s) need to exit, for the passengers

assigned to the middle or window seats to be seated.

- Aisle interferences occur when a lower row passenger is in front of the high error passengers while boarding the aircraft.



Seat and aisle interferences

Model Description:

- Our focus in this section is to develop a mathematical model which captures the behavior of passengers boarding the aircraft.
- The objective of this model is to minimize the total number of interferences subject to operational and side constraints.
- Note that the model assumes a single aisle or narrow-body aircraft such as an Airbus A-320 or Boeing 737 and all passengers' board through a single aircraft door.
- We assume that each seat in this aircraft is represented by (i,j) where i (i=1,.,N) is the row and j (j=A,B,.,F) is the location of the seat within row i

Location of seats within rows i:

- In this model we attempt to assign each seat to a group. Each passenger in seat (i,j) is assigned to a group k (k=1,.,G).
- Each group is then called in sequence to board the aircraft.
- The following binary decision variable is adopted for our integer linear.

Programming model:

- Our objective is to assign seats (i,j) to groups ($k \in G$) so that the total number of interferences with penalties attached to them (as described later) is minimized.

Seat Interferences:

- There are two types of seat interferences: between-groups and within- groups described as follows:

Between-Groups Seat Interferences:

- This type of seat interference occurs when a passenger from an earlier group blocks another passenger in a later group.
- If the passenger in seat 16C (aisle seat) boarded in group 2 and passenger in seat 16B (middle seat) boarded in a later group, then the passenger in seat 16C is blocking the passenger in seat 16B.
- In this case, the passenger in seat 16C needs to exit, to allow the passenger in seat 16B to be seated, thus blocking the flow of passengers in the aisle.
- More seat interferences occur if the passenger in seat 16A (window seat) boards after passengers are seated in seats 16B and16C.
- Considering all the possible combinations, four types of seat interferences between different groups can occur as follows.
- Aisle-seat passenger blocking the middle-seat passenger; aisle-seat passenger blocking the window-seat passenger; middle-seat passenger blocking the window-seat passenger; aisle-and middle-seat passengers blocking the window-seat passenger; We examine the mathematical model for each case separately.

Aisle-Seat Passenger Blocking Middle-Seat Passenger:

- First we develop the seat interferences model for seats B and C on the left hand side of the aisle for seats B and C.
- We define SB_i, BC, k as a binary variable representing number of seat interference between the seat C (aisle seat)and seat B (middle seat) in row i, who boarded in group k. SBi, BC, k, takes a value of1 if an interference occurs or 0 otherwise.
- Mathematically, SBi, BC, k can be expressed as the following constraint in our mathematical model:
- On the right hand side of the constraint, we have xi, B, k, which represents if the passenger in the middle seat (seat B) boards in group k.
- Indicates if the passenger sitting in the aisle seat (seat C) has boarded in any of the earlier groups (before k).
- The term (-1) in the above constraint is added to set the value of SBi, BC, k equal to 1 if there is an interference or 0 otherwise.
- We use greater or equal sign (≥). This ensures that SBi, BC, k takes a value of 0 for the last case in the above table if both terms on the right hand side are 0.
- Similarly we can write the constraints for aisle and middle seat interferences (seats E and D) on the right hand side of the aisle.

Examining aisle- and middle-seat interference

$x_{i,B,k}$	$\sum_{l=1}^{k-1} x_{i,C,l}$	$SB_{i,BC,k}$	Comments
1 (The passenger in middle seat B is in group k)	1 (The passenger in aisle seat C has already boarded in an earlier group)	1	Interference occurs
0 (The passenger in middle seat B is not in group k)	1 (The passenger in aisle seat C has already boarded in an earlier group)	0	No interference
1 (The passenger in middle seat B is in group k)	0 (The passenger in seat C has not boarded in any of the earlier groups)	0	No interference
0 (The passenger in middle seat B is not in group k)	0 (The passenger in the aisle seat C has not boarded in any of the earlier groups)	-1	No interference

Aisle-Seat Passenger Blocking Window-Seat Passenger:

- We adopt a similar approach as aisle- and middle-seat interference to expresses the number of seat interferences and its corresponding constraints between aisle- and window-seats among different groups for both sides of the aisle as follows

Middle-Seat Passenger Blocking Window-Seat Passenger:

- The number of seat interference and constraint between window- and middle-seats among different groups for both sides of the aircraft are as follows:

Aisle- and Middle-Seat Passengers Blocking Window-Seat Passenger:

- Having established the above seat interferences, we do not need to express a specific set of constraints for a window-seat passenger when both middle- and aisle-seat passengers have already been seated.
- This type of interference has already been addressed in the form of two separate constraints (interferences).
- These two interferences are window with middle and window with aisle seats.
- For example, consider a case when a passenger in seat
- A board just after both passengers in seats B and C. In this case, according to constraints above SBi, *AC*, *k* and *SBi*, *AB*, *k* will take a value of 1 implying that the passenger sitting in seat A (window) will have a total of 2 seat interferences with aisle and middle seats.

Total Seat Interferences Among Different Groups:

- We can now express the total number of seat interferences between different groups by adding all the seat interferences.

- Let *TSB* represent the total seat interferences between groups, then:

Within-Groups Seat interferences:

- This type of interference occurs among passengers boarding in the same group.
- We assume the sequence in which the passengers within a group board the aircraft is random.
- For example, passengers in seats 16A and 16B are boarding in the same group. When their group is called, passenger 16A may board first and be in front of 16B in the respective group or vice versa. In the former case when the passenger in seat 16A boards before 16B, no interference occurs.
- However, in the latter case when the passenger in 16B boards before 16A, there will be a seat interference.
- Adopting the same argument as between groups seat interference, we denote the binary variable *SWi, BC, k* to represent the seat interference between the aisle(seat C) and middle seat (seat B), who board in the same group.
- Similar to the section on aisle and middle seat interferences, *SWi, BC ,k*can take a value of 1 or 0 depending on values of *xi, B, k* and *xi, C, k*.
- If passengers in seats B and C in a row *i* are boarding in the same group *k*, then the constraint returns a value of 1 for *SWi, BC, k*, otherwise it will be 0.
- However, as indicated before the order of these two passengers is random.
- Therefore the expected number of seat interferences between passengers in seats B and C within the same group.
- The expected number of seat interferences within each group is ½ each of the above *SW*.
- Again we do not need to add a new constraint for the case when all three
- Neighboring passengers are in the same group.
- The total of seat interferences within the same groups (*TSW*) is therefore obtained by:

Aisle Interferences:

- Similar to seat interferences, there are two types of aisle interferences, within groups and between groups.

Within-Groups Aisle Interferences:

- This common type of aisle interference relates to cases where passengers assigned to the same group block each other.
- This occurs when a passenger in a lower row blocks other passengers behind him or her in order to be seated.
- The problem becomes compounded when the passenger has multiple bags to store in the overhead bin.
- We further break down these within group aisle interferences into interferences with lower rows and interferences with same rows.

Within-Groups Aisle Interferences with Lower Rows:

- Let the integer variable $AW1i,j,k$ represent the (maximum) number of aisle interferences for the passenger in seat *(i,j)* assigned to group *k* with lower row passengers in the same group.
- Similar to the previous section, we can write the constraint for $AW1i,j,k$ as follows:
- On the right-hand side of this constraint, the first term takes a value of *6(i-1)* if the passenger in seat *(i,j)* is assigned to group *k, or* zero otherwise.
- The second term adds up all the passengers in the same group *k*, who have a lower row seat assignment than *i*.
- The third term on the right-hand side is adopted to provide the correct count on aisle interferences for $AW1i,j,k$.
- This implies the worst case where the passenger in seat *(i,j)* boards after all passengers in the lower rows in the same group.
- Therefore, $AW1i,j,k$ represents the maximum number of aisle interferences for the passenger in seat (*i,j)* assigned to group *k.*
- The lowest number of aisle interferences for the passenger in seat *(i,j)* assigned to group *k* occurs if the passenger boards before all the passengers in lower rows of that group.
- In this case since the passenger moves all the way down the aisle to his or her designated row without being blocked by anyone within this group, then there are no within-group aisle interferences.
- Therefore the expected number of aisle interferences for passenger in seat *(i,j)* assigned to group *k* is:
- The total expected number of aisle interferences for all passengers with their lower row passengers presented by *AWL* is therefore determined by:

Within-Groups Aisle Interferences in the Same Row:

- In the section within-groups aisle interferences with lower rows above, we did not consider possible aisle interferences among passengers in the same row and same group.
- This section addresses the expected number of aisle interferences for those passengers. We define integer variable $AW2i,j,k$ to represent the (maximum) number of aisle interferences between the passenger in seat *(i,j)* and all other passengers in the same row *i*, boarding in group *k*.
- The first term on the right hand side takes a value of 5 or zero depending on whether the passenger in seat (i,j) is in group *k* or not.
- The second term adds up the number of passengers in row *i* and number of passengers in group *k*, except for the passenger sitting in *(i,j)*.
- The third term is used to provide the right number of counts for aisle interferences. Similar to the previous section, the minimum number of interferences for passenger in seat (*i,j*) boarding in group *k* is 0 and the maximum is $AW2i,j,k$.
- Therefore the expected number of aisle interferences in the same row within

the same group for this passenger is 1/2 $AW2_{i,j,k}$.

- We define *AWS* to represent the total expected number of same-row aisle interferences for all passengers given by:

Between-Groups Aisle Interferences:

- This type of aisle interference, as many of us have experienced, occurs when a group of passengers are called to board the aircraft while some or all of the passengers in the previous group are still in the jet-way (staircase) or aircraft door waiting to be seated.
- These interferences occur and get worse as the time between boarding the passengers and groups decreases.
- We define the integer variable $AB_{i,j,k}$to represent the maximum number of aisle interferences for passenger in seat (i,j) who boards in group k (k>1) with all the passengers in the previous group (k-1) We write the following constraint for $AB_{i,j,k}$: The first term on the right hand side of this constraint takes a value of 6i if the passenger in seat (i,j) is assigned to group k, or 0 otherwise.
- The second term adds up all the passengers who boarded in group (k-1) and the third term is used to provide the correct count.
- The constraint for $AB_{i,j,k}$ assumes that none of the passengers from the earlier group is seated when the passenger in seat *(i,j)* assigned to group k boards the aircraft. Of course, the expected number of aisle interferences for this passenger depends on how quickly each group of passengers is called to board the aircraft.
- We will assume that for any passenger in group k (k>1) boarding the aircraft, there are a fraction of passengers from the previous group (k-1) still in the jet-way trying to reach to their seats.
- We call this fraction α (0≤α≤1). Therefore the expected number of aisle interferences between the passenger in seat (i,j) assigned to group k with passengers in group (k-1) is $\alpha \, AB_{i,j,k}$.
- When α is 0, no aisle interferences occur between groups. This occurs when a new group of passengers is called to board the aircraft when all the passengers from the earlier group are fully seated. On the other extreme, when α is equal to 1, the time between calling groups to board is so short that the passengers in each group line up behind the previous group in the aisle or jet-way. In our later analyses, we examine various values for α and its impact on boarding pattern and strategy.
- Let *ABG* represent the total aisle interferences between groups for all passengers boarding in group k (k>1). *ABG* is therefore determined by:
- To keep the model simple and without loss of generality, we only include aisle interferences between passengers in group k (k>1) with passengers in group(k- 1).
- It is, of course, possible to mathematically include the aisle interferences

between passengers in group *k* and groups (*k-2,* for *k>2*) or (*k-3* for *k>3*), and soon.

- Our simulation study, discussed later, also confirmed that for realistic times between passengers to board the aircraft, these second or third level interferences are relatively very low compared to the first level that is considered in the model.

Mathematical Model:

- In this mathematical model, we attempt to minimize all the seat and aisle interferences that were examined in section 4 as follows:

- The objective function   attempts to minimize the total expected number of seat and aisle interferences. *p*1, *p*2, *p5* represent the penalties assigned to different types of interferences.
- The set of constraints in ensures that each seat in the aircraft is assigned to one and only one group.
- Typically the airlines favor a balanced number of passengers among different groups.
- The two sets of constraints ensure that the number of passengers assigned to each group is not less than min_pax and not more than max_pax.

Model Parameters:

- As indicated before, *p*1, *p*2,..., *p5* are adopted to assign weights to different seat and aisle interferences.
- The literature adopting simulation models for boarding strategies mainly uses triangular distributions to model the times for seat and aisle interferences.
- Similar time parameters are used in the simulation study by Ferrari and Nagel (2004).
- We adopted the mean of these distributions to represent the penalties in this model. These distributions have a mean of 3.6 for seat interference and 2.4 for aisle interferences.
- Without loss of generality, we assign the same weight to seat (*TSB* and *TSW*) and same weight to aisle (*AWL, AWS, ABG*) interferences as follows: In the between-groups aisle interference section, we assumed that for passengers in group *k (k*>1) boarding the aircraft, there is a fraction of passengers from the previous group (*k*-1) still in the jet-way trying to reach their seats.
- We called this fraction α (0≤α≤1). The two extreme cases, when α is 0 or 1 represent situations when there is 0% or 100% between group interferences.
- To identify the impact of α on boarding strategy, we considered various values for this parameter.
- We solved the above mathematical model for the following values of α: Airlines typically assign 4, 5, or 6 groups to board their passengers on a single- aisle aircraft

- To provide a better understanding of boarding strategy as the number of groups change, we solved our integer programming model with 4, 5, and 6 groups, that is $G \in \{4,5,6\}$In our model, we set *min_pax*and *max_pax*, to allow a maximum of 20% fluctuations around the mean as follows: Where [denotes the integer value of •. *N* represents the number of rows.
- To evaluate the performance of the model, we applied it to an Airbus A-320aircraft with26rows. The first three rows (with 4 seats in each row) are assigned to first and business class passengers (group 1), who always board first. In our model we study all other passengers who are assigned to the other 23 rows (*N*=23), with six seats in each row, who must be allocated to different groups.

Simulation Model:

- In order to study and determine the values of α for different boarding times, a simulation model in Arena Simulation Modeling Software.
- Similar aircraft load-factor and time distributions for aisle and seat interferences were adopted.
- We ran the simulation models with inter-arrival times of passengers for boarding changing from 3 seconds to 10 seconds or 20 to 6 passengers per minute.

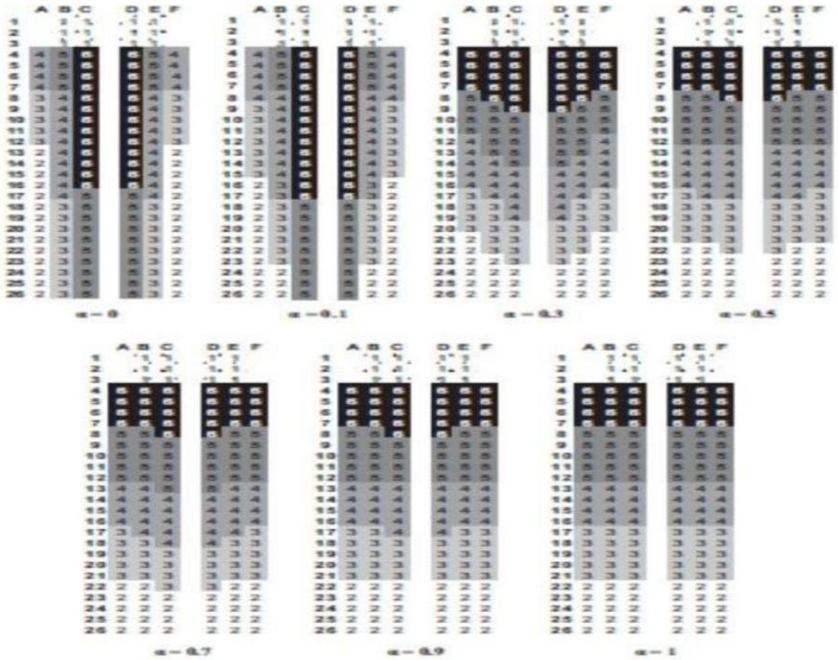


Figure 12.4 Solution for boarding patterns based on 6 groups and different values of α

Table 12.2 Seat, aisle and total interferences for solution to 6-groups boarding process

α	Interference	Solution	Sum	Solution
0	TSB	0	Total seat interferences	0
	TSW	0		
	AWL	884	Total aisle interferences	953
	AWS	69		
	ABG	0		
	Obj. Function	2287.2	Total interferences	953
0.1	TSB	0	Total seat interferences	4
	TSW	4		
	AWL	878	Total aisle interferences	1059.4
	AWS	85		
	ABG	96.4		
	Obj. Function	2556.96	Total interferences	1063.4
0.3	TSB	0	Total seat interferences	47
	TSW	47		
	AWL	792	Total aisle interferences	1091
	AWS	257		
	ABG	42		
	Obj. Function	2787.6	Total interferences	1138

- Our main task of measuring performance in this study was to identify the number of passengers at the door, when new passengers will line up behind them for different times between passengers boarding.
- The number of passengers at the aircraft door (#pax) for passenger inter-arrival times ranging from 3 to 10 seconds.
- It also presents values of α based on the number of passengers at the door (#pax). On average, for an Airbus A-320 with 6 groups, there are 28

passengers per each economy group. Therefore α is determined by dividing #pax by28.

Expected number of passengers and values of α for boarding based on varying inter-arrival times

Time between arrivals (sec)		3	4	5	6	7	8	9	10
6 groups	# Pax	25.9	15.8	9	5.7	3.2	0	0	0
	α	0.9	0.6	0.3	0.2	0.1	0.0	0.0	0.0

- For average passenger arrival times, Van den Briel et al (2005) considered 7seconds with 1 gate agent and 5 seconds with 2 gate agents (rounded to the nearest second) and Van Landeghem and Beuslinck (2002) considered 6–7 seconds in their simulation models.
- These times are based on actual observations of passenger boarding times at different airlines and at different airports.
- Using these inter arrival times and based on Table 12.3, the realistic values for α range from 0.3for 5 seconds to 0.1 for 7 seconds inter-arrival times.
- For α taking value 0.1 and 0.3 seem to be appropriate for boarding strategies depending on 1 or 2 gate agents.
- These solutions indicate that back-to-front boarding strategy, as adopted by many airlines, is not necessarily an efficient process.

Airline Information Technology (IT) Solutions:

- Airlines that do not develop their in-house customized software outsource their IT needs to Airline IT solutions providers such as Sabre, SITA, Lufthansa Systems, Jeppesen, and EDS.
- According to Lufthansa, the global market for airline IT was estimated to have a volume of $11.47 billion during 2008, and around 40% of this volume is outsourced to airline IT-solution providers.
- Some of the main software solutions offered by the airline IT companies are in the areas of crew scheduling, fleet operations, revenue management, ticket distribution, and aircraft maintenance as follows.

Crew Scheduling:

- Companies like Sabre, Jeppesen, Lufthansa Systems, and SITA offer crew scheduling softwares that are suitable for large airlines.
- Most solutions are provided on a modular basis and some are customized to integrate with the airlines in-house software.
- Some of the main features provided in crew scheduling solutions are planning, monitoring, pairing optimizer, and crew access.
- Planning modules help managers generate long-term crew requirements for up to two years; taking into account factors such as reserve needs, training, and vacation.
- Monitoring modules help managers with the daily operations and allow tracking of crew and flight operations on a real-time operational basis.
- Input to generate the crew roster is obtained from the crew through the crew access module.
- The crew access module also gives crew the ability to directly access their roster through the internet and swap or trade duty shifts.
- The input from the crew is then optimized using the crew pairing optimizer taking into consideration the flight schedule, cost, and crew needs.

Flight Operations:

- Flight-operation solution contains a range of modules that assist from flight planning to daily tracking and managing of flight operations.
- Flight operation solutions typically contain modules that assist with flight planning, flight dispatch, operations monitoring, fleet optimization, load planning, and critical decision support. Lufthansa’s Net Line/Plan Route Optimizer, for instance, allows managers to create the optimal flight routing based on simulation of new connections,

List of airline IT-solution providers offering crew scheduling solutions

Company	Product	Website
AOS	Integrated Crew Planning System	www.aos.us
Jeppesen	Carmen Crew Management System	www.jeppesen.com
Lufthansa Systems	NetLine/Crew	www.lhsystems.com
Navitaire	Geneva Operations Control & Management Suite	www.navitaire.com
Ortec	Integrated aircraft and Crew planning	www.ortec.com
Sabre	AirCentre Crew	www. sabreairlinesolutions. com
SITA	CrewWatch	www.sita.aero

- Fore cast passenger flows, costs, revenues, and identified strengths and weaknesses of the network.
- The optimized flight routing is then used to assign the available aircraft to achieve maximum overall profitability.
- Day-to-day operations and flight monitoring are achieved through the operations monitoring module. SITA's Fleet Watch, for instance, provides managers with pertinent information about current operations, maintenance events, and helps to evaluate problems and determine the most cost-effective solution.
- Critical-decision support tools are also provided and integrated with flight-operations solution to help plan for disruption events and create optimized recovery solutions.

Revenue Management:

- Revenue-management tools help airlines increase profitability though better yield and better price structure.
- The revenue-management solutions contain modules that help managers forecast demand, allocate seats, calculate optimal ticket price, distribute fares to Global Distribution System (GDS), monitor competitor price, manage over-bookings, and cancel multiple bookings.
- Revenue management tools utilize historical and current data to forecast booking activities and make informed revenue-management decisions.
- Revenue-management solutions have been continually evolving and one of

the recent changes in revenue-management solution is the migration from leg-based revenue-management system to Passenger Name Record (PNR) based Origin and Destination (O&D) revenue-management system. PNR based O&D revenue management system allows managers to tap.

List of major flight-operation solution-providers

Company	Product	Website
Jeppesen	Carmen Integrated Operations Control	www.jeppesen.com
Lufthansa Systems	Integated Operations Control Center	www.lhsystems.com
Navitaire	Geneva Operations Control & Management Suite	www.navitaire.com
Ortec	Integrated aircraft and Crew planning	www.ortec.com
Sabre	AirCentre Flight	www.sabreairlinesolutions.com
SITA	FleetWatch, FleetPlan, Flight Planning	www.sita.aero

List of major revenue-management solution-providers

Company	Product	Website
Amadeus	Altéa Revenue Management	www.amadeus.com
Lufthansa Systems	ProfitLine	www.lhsystems.com
Navitaire	RMS Host Revenue Management System	www.navitaire.com
PROS	PROS O&B, PROS NPRS	www.prospricing.com
Sabre	AirMax Suite	www.sabreairlinesolutions.com
SITA	Airfare Price	www.sita.aero

Ticket Distribution:

- Distribution and ticket sales are another area where airline solution providers play a major role.
- Owing to the rapid growth of the internet, airlines have begun phasing out travel agents and have led to the growth of computer reservation systems (CRS) and global distribution systems (GDS).
- Airlines employ ticket- distribution modules to seamlessly transfer fares from their system to the GDS.
- Ticket- distribution solutions providers sell solutions that include booking engines, business process management, channel distribution, customer relationship management, reservations, customer data, and analysis and ticketing.
- These systems work together to maximize revenue through maximum distribution of tickets through all available channels.

List of major ticket-distribution solution-providers

Company	Product	Website
Abacus	FareX	www.abacus.com.sg
Amadeus	Amadeus Airline Retailing Platform	www.amadeus.com
Galileo	Galileo Agency Private Fares	www.travelport.com
KIU	KIU System Solutions	www.kiusys.com
Lufthansa Systems	Passenger Core Systems	www.lhsystems.com
Navitaire	New Skies	www.navitaire.com
Patheo	PAL, FareMate	www.patheo.com
Sabre	SabreSonic	www.sabreairlinesolutions.com
SITA	Horizon	www.sita.aero
Worldspan	Worldspan FareSource	www.worldspan.com

Supplementary Airline IT Solutions:

- In addition to the above solutions, airline IT providers develop other IT solutions to improve efficiency and effectiveness in areas such as maintenance, repair and overhaul, air-cargo handling, administration, finance, flight navigation, flight planning, gate assignment, aircraft load management, ground handling integration, check-in systems.