Lecture XI – Scheduling Planning II

Airline Planning & Optimisation (AE4423)

Bruno F. Santos

TUDelft

Lecture Content

1st Part - Aircraft Routing - MILP

- Problem definition
- Maintenance routing
- Example
- Approach
- Results

2nd Part - Aircraft Routing + Timetable

- Background
- Dynamic Programming
- Example
- Implementation
- Results

TU Delft

Program

Management

Planning structure

Performance indicators

Demand and supply

Market share

Tactical planning

Passenger Mix Flow

Fleet assignment

Aircraft rotation

Crew scheduling

Maintenance

Strategic planning

Network structure

Demand forecast

Network planning

Fleet planning

Operations research

MILP models

Multi-commodity Flow problems

Shortest-path algorithms

Dynamic programming

TU Delft

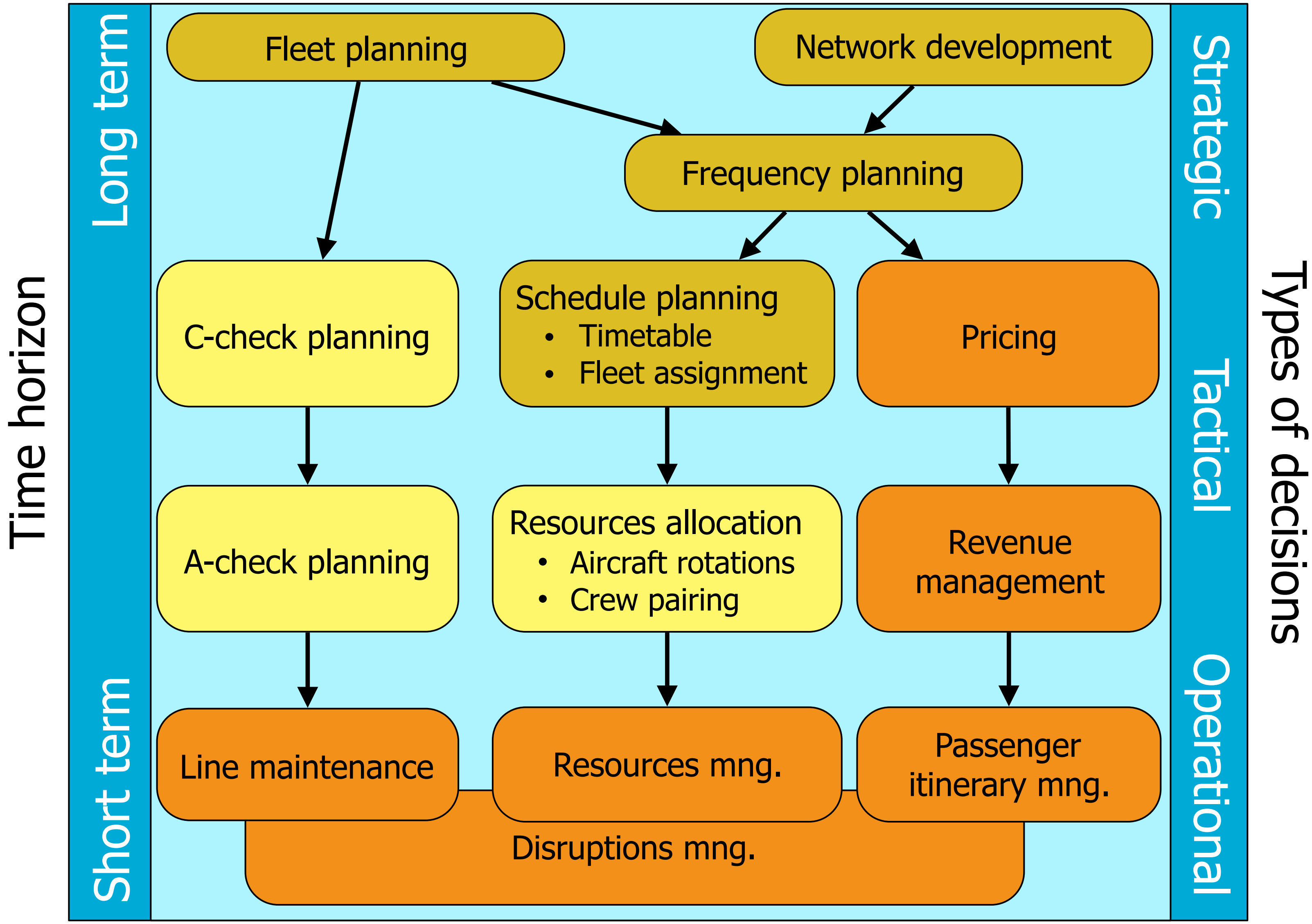
Aircraft Routing - MILP

- Problem definition
- Maintenance routing
- Example
- Approach
- Results

TUDelft

Planning Framework

Not covered in AE4423



TU Delft

Schedule development

Given a set of routes to be operated in an airline network, and a fleet of aircraft, schedule development involves four different but interrelated tasks:

Last lecture

- **Schedule design:** how often should the airline operate flights on the selected route(s) and at what times should flights departure?
- **Fleet assignment**: what type of aircraft should be used for each departure time?

Today

- **Aircraft rotation**: how should each specific aircraft (tail number) be flown over the airline's network in order to ensure a consistent operation?

Next lecture

- **Crew scheduling**: how to assign crew (cabin & pilots) to each flight guaranteeing the operation of the flights and satisfying all work rules.

TUDelft

Problem definition

Fleet assignment determines **the aircraft type** to operate on a specific route.

Aircraft routing determines which **specific aircraft in the fleet** operates each flight leg. The main objectives of aircraft routing are:

- To cover each flight leg by only one aircraft
- To balance the aircraft utilisation
- To comply with maintenance requirements

TU Delft

Problem definition

Tail number

Problem definition

Different possible objective functions:

- Minimize the total operating cost for a specific aircraft type
- Maximize aircraft utilisation
- Maximize profit
- Maximize maintenance opportunities

Subject to the following considerations:

- Flight coverage (operate each flight once and only once)
- Number of available aircraft
- Maintenance requirements
- Aircraft utilisation balance

Maintenance routing

Also called ‘Aircraft maintenance routing problem’

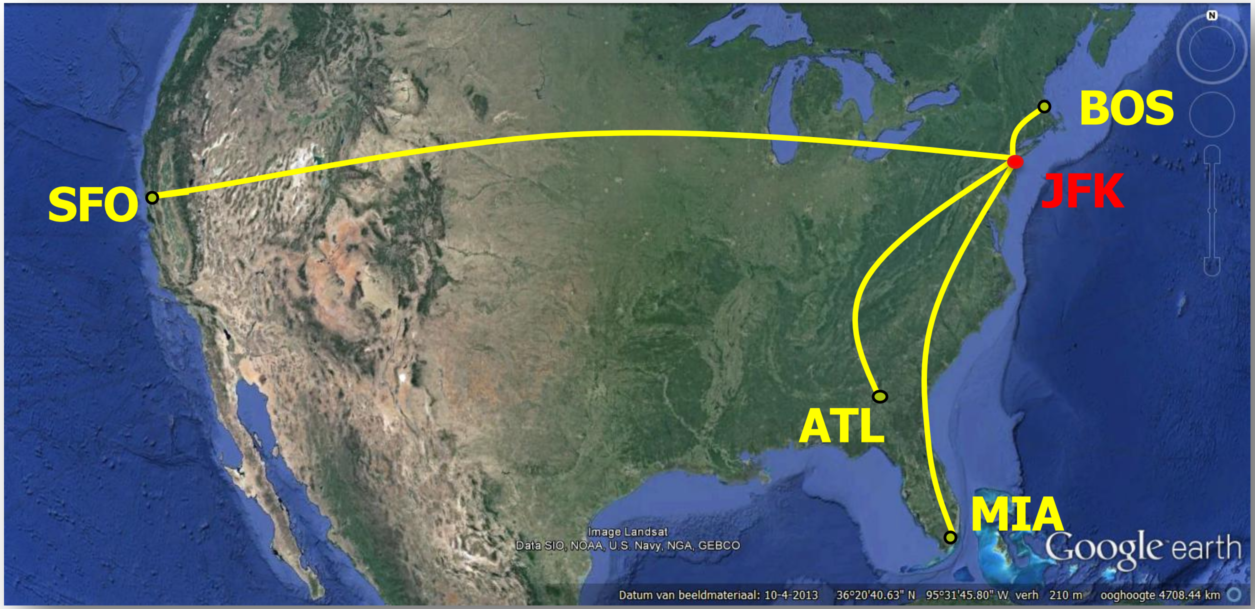
- aircraft undergo periodic maintenance checks at regular intervals
- these checks are performed at specific maintenance stations
- in general, regulations require inspections of aircraft approximately every **3–5 days**
-
- Airlines typically built aircraft routing that provide “frequent” maintenance opportunities at the maintenance station
  - Provide slack times at the station
  - Guarantee enough visits to the station every 3-5 days

TU Delft

Example

Problem 1

Define the route of 6 aircraft operating flights for an airline with the maintenance base at JFK. The flight table is given in the next slide.



Based on Bazargan's book example

Example

Flights (local times):

Flight	Origin	Dep.	Destination	Arr.	Hrs
125	JFK	7:25	SFO	9:55	5.5
110	ATL	8:10	JFK	10:40	2.5
113	MIA	9:10	JFK	12:10	3
131	JFK	9:30	ATL	12:00	2.5
105	SFO	10:50	JFK	18:20	5.5
138	JFK	12:30	BOS	14:00	2.5
111	ATL	13:10	JFK	17:30	3
114	MIA	14:30	JFK	17:30	3
118	BOS	15:00	JFK	16:30	1.5
135	JFK	15:10	MIA	18:10	3
133	JFK	18:05	ATL	20:35	2.5
136	JFK	18:10	MIA	21:10	3

TU Delft

Example

Note:

- Each aircraft is routed such that it stays overnight at JFK at least once every three days
- Valid routes include turn-around-times of 45 minutes

Flight no.	Origin	Dep. Time	Destination	Arr. Time	Hrs
113	**MIA**	**9:10**	**JFK**	**12:10**	**3**
~~138~~	~~JFK~~	~~**12:30**~~	~~BOS~~	~~14:00~~	~~2.5~~
135	JFK	15:10	MIA	18:10	3
133	JFK	18:05	ATL	20:35	2.5
136	JFK	18:10	MIA	21:10	3

Approach

Determination of valide routes:

- Three day closed cycle
- Valid route starts and ends at the same airport, and includes at least one overnight at JFK for maintenance

Flight no.	Origin	Dep. Time	Destination	Arr. Time	Hrs
Day 1					
125	JFK	7:25	SFO	9:55	5.5
Day 2					
105	SFO	10:50	JFK	18:20	5.5
Day 3					
131	JFK	9:30	ATL	12:00	2.5
111	ATL	13:10	JFK	17:30	3

TU Delft

Approach

Flight coverage consideration

- Each flight should be operated at least once and only once

Flight no.	Origin	Dep. Time	Destination	Arr. Time	Hrs
125	JFK	7:25	SFO	9:55	5.5

- Only six routing candidates (of the 455 possible) cover flight 125 (and visit JFK, at least once):

Candidate	Day 1	Day 2	Day 3
x_1	**JFK-SFO**	SFO-JFK	JFK-ATL-JFK
x_2	**JFK-SFO**	SFO-JFK	JFK-BOS-JFK
x_3	JFK-ATL-JFK	**JFK-SFO**	SFO-JFK
x_4	JFK-BOS-JFK	**JFK-SFO**	SFO-JFK
x_5	SFO-JFK	JFK-ATL-JFK	**JFK-SFO**
x_6	SFO-JFK	JFK-BOS-JFK	**JFK-SFO**

TU Delft

Approach

Flight coverage consideration

$$x_1 + x_2 = 1$$

$$x_3 + x_4 = 1$$

$$x_5 + x_6 = 1$$

Candidate	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**

Introduce $a_{i,j} = 1$ if flight i is covered by route j, and 0 otherwise:

$$\sum_{j=1}^{R} a_{i,j} \times x_j = 1, \qquad \forall i \in F$$

Approach - Model

Sets:

- $\boldsymbol{R}$ Number of possible routings
- $\boldsymbol{F}$ Set of flights

Decision variables:

- x_j 1 if route j is selected, 0 otherwise

Parameters:

- m_j Number of maintenance opportunities for route j
- $a_{i,j}$ 1 if flight i is covered by route j, 0 otherwise
- N Total number of aircraft in fleet

TU Delft

Approach - Model

Aircraft fleet constraint

- Available number of aircraft should not be exceeded
- The number of selected routes should not exceed the number of available aircraft

$$\sum_{j=1}^{R} x_j \leq N$$

TU Delft

Approach - Model

The resulting model follows **a set-partitioning problem** structure (i.e., determination of how the items in one set S can be partitioned into smaller subsets).

$$maximize \sum_{j=1}^{R} m_j \times x_j$$

Maximise maintenance opportunities, captured by the m_j

s.t.:

$$\sum_{j=1}^{R} a_{i,j} \times x_j = 1, \qquad \forall i \in F$$

Flight coverage

$$\sum_{j=1}^{R} x_j \leq N$$

Aircraft fleet

Results

One of the possible solutions

Route	Day 1	Day 2	Day 3
1	125-105 JFK-SFO-JFK	135 JFK-MIA	114 MIA-JFK
2	110-138-118-136 ATL-JFK-BOS-JFK-MIA	113 MIA-JFK	131-111-133 JFK-ATL-JFK-ATL
3	113 MIA-JFK	131-111-133 JFK-ATL-JFK-ATL	110-138-118-136 ATL-JFK-BOS-JFK-MIA
4	131-111-133 JFK-ATL-JFK-ATL	110-138-118-136 ATL-JFK-BOS-JFK-MIA	113 MIA-JFK
5	114 MIA-JFK	125-105 JFK-SFO-JFK	135 JFK-MIA
6	135 JFK-MIA	114 MIA-JFK	125-105 JFK-SFO-JFK

- Objective function value = 9

Aircraft tailor numbers are associated with each route.

Results

Concluding remarks:

- Balanced aircraft utilisation in this example is not taken into account
  - The result is (by chance) balanced - 17-19 hrs per aircraft per 3 days
  - Note: this is a low utilisation
- Only three-day cycles are considered
- Aircraft tailor numbers still have to be associated with each route
  - A rotation approach should be follow to force aircraft to have similar rotations and ageing

TU Delft

Aircraft Routing + Timetable

- Background
- Dynamic Programming
- Example
- Implementation
- Results

Background

The complexity of the set-partitioning approach is the generation of routes.

The routing problem is a dynamic network problem

- Dynamic – route over time
- Network – route over space

An alternative solution framework is Dynamic Programming

- Developed by Richard Bellman in 1950s, it breaks down a problem into smaller sub-problems

TU Delft

Dynamic Programming

Dynamic Programming (DP) is a useful mathematical technique for making a sequence of interrelated decisions.

Divide-and-conquer

- The idea: to break a large problem down into incremental steps so that, at any given stage, optimal solutions are known to sub-problems.
- The sub-problems have a 'trivial' solution (i.e., simple to compute)

Notes:

- In contrast to linear programming, there does not exist a standard mathematical formulation of "the" DP problem
- Floyd's all-pairs shortest-path algorithm was an example of DP.

TU Delft

Dynamic Programming

Bellman's Principle of Optimality

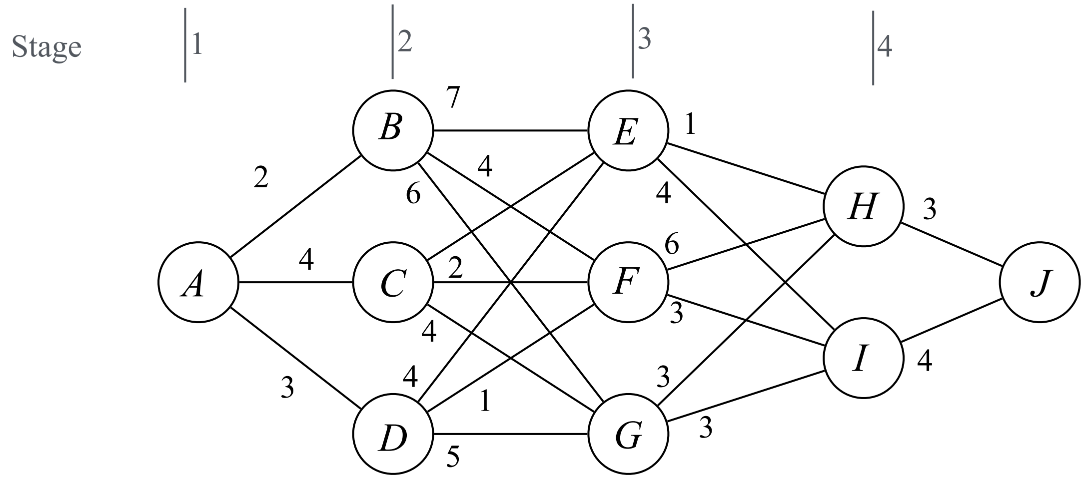
*An **optimal policy** has the property that whatever the initial **state** and initial **decision** are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision* [1].

[1] R. E. Bellman, Chap. III.3, Dynamic Programming. Princeton University Press, Princeton, NJ, 1957

- **Policy** – decision rule defined according to the current state
- **State** – current situation of the system, including all the information needed to characterise it
- **Decision** (or action) – action from the decision maker to influence the evolution of the system

TU Delft

Dynamic Programming



Example: Introduction to Operations Research - Hillier/Lieberman (Ch.11)

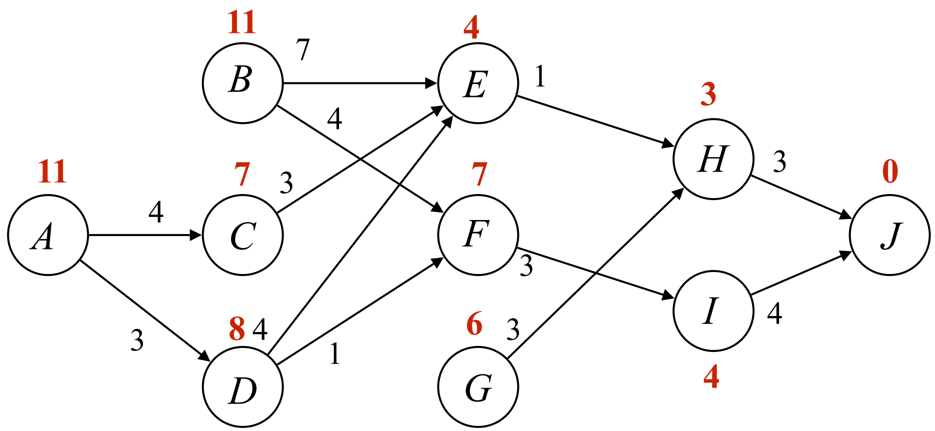
Let $f_n(s, x_n)$ be the total cost of the best overall policy for the remaining stages:

$f_n(s, x_n)$ = *immediate cost (stage n) + min future cost (stage N+1 onward)*

$$f_n(s, x_n) = c(s, x_n) + min_{x_{n+1}} f^*_{n+1}(s, x_n)$$

- s is the state
- n is the stage to start (stage 1: A; stage 2: B, C, D; …)
- x_n is the action taken (i.e., next node)
- $c(s,x_n)$ is the immediate contribution (cost) of the action taken

Dynamic Programming



Example: Introduction to Operations Research - Hillier/Lieberman (Ch.11)

For the starting node A:

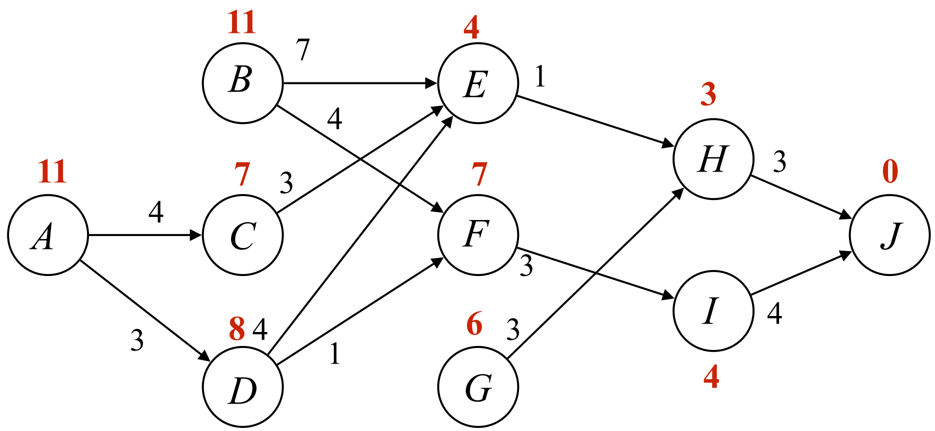
$$f_1(A, x_n) = c(A, x_n) + min_{x_{n+1}} f^*_{n+1}(A, x_n)$$

$$f_1(A, > B) = 2 + min_{x_{n+1}} f^*_{n+1}(A, > B) = 2 + 11$$

$$f_1(A, > C) = 4 + min_{x_{n+1}} f^*_{n+1}(A, > C) = 4 + 7$$

$$f_1(A, > D) = 3 + min_{x_{n+1}} f^*_{n+1}(A, > D) = 3 + 8$$

Dynamic Programming



Example: Introduction to Operations Research - Hillier/Lieberman (Ch.11)

For the starting node A:

$$f_1(A, x_n) = c(A, x_n) + min_{x_{n+1}} f^*_{n+1}(A, x_n)$$

$$f_1(A, > B) = 2 + min_{x_{n+1}} f^*_{n+1}(A, > B) = 2 + 11$$
$$f_1(A, > C) = 4 + min_{x_{n+1}} f^*_{n+1}(A, > C) = 4 + 7$$
$$f_1(A, > D) = 3 + min_{x_{n+1}} f^*_{n+1}(A, > D) = 3 + 8$$

Optimal

These are the optimal decisions, following a greedy policy and considering that the future costs are also optimal (followed the same policy)

Example

An airline is operating between six airports in Europe:

Hub airport - Amsterdam

Example

Demand per day

	EHAM	EDDF	LEMF	LIRF	EIDW	ESSA	LPPT
EHAM	0	582	168	204	288	204	90
EDDF	582	0	180	303	210	132	84
LEMF	168	180	0	357	165	105	348
LIRF	204	303	357	0	195	132	66
EIDW	288	210	165	195	0	171	72
ESSA	204	132	105	132	171	0	3
LPPT	90	84	348	66	72	3	0

Variation per hour ($Dem(t)_{i,j} = Dem_{i,j} * Coef(t)_i$)

Airport	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
EHAM	0	0	0	0	0.034	0.099	0.154	0.214	0.076	0.091	0.202	0.119	0.218	0.082	0.119	0.112	0.312	0.225	0.157	0.130	0.089	0.052	0.119	0.038
EDDF	0	0	0	0	0.039	0.039	0.097	0.246	0.410	0.200	0.084	0.084	0.112	0.097	0.133	0.131	0.191	0.177	0.223	0.068	0.248	0.108	0.070	0.022
LEMF	0	0	0	0	0.014	0.023	0.034	0.059	0.153	0.220	0.404	0.166	0.175	0.082	0.055	0.078	0.216	0.114	0.278	0.137	0.050	0.025	0	0
LIRF	0	0	0	0	0.070	0.083	0.121	0.094	0.112	0.093	0.146	0.100	0.094	0.103	0.136	0.151	0.458	0.195	0.184	0.156	0.064	0.086	0	0
EIDW	0	0	0	0	0.084	0.059	0.102	0.213	0.178	0.087	0.054	0.102	0.078	0.033	0.057	0.066	0.456	0.193	0.290	0.127	0.080	0.077	0	0
ESSA	0	0	0	0	0.116	0.132	0.143	0.320	0.294	0.094	0.053	0.089	0.082	0.056	0.103	0.110	0.301	0.187	0.296	0.160	0.045	0.097	0.075	0
LPPT	0	0	0	0	0.035	0.075	0.160	0.089	0.117	0.150	0.227	0.075	0.163	0.054	0.136	0.308	0.264	0.097	0.280	0.047	0.068	0.084	0.022	0

TUDelft

Example

Fleet

Aircraft Type	Type 2	Type 3
Speed [km/h]	810	870
Seats	50	120
Average TAT [min]	30	35
Maximum Range [km]	3 800	4 500
Runway Required [m]	1 500	1 600
Lease Cost [€]	4 540	7 340
Fixed Operating Cost (Per Fligth Leg) [€]	620	920
Cost per Hour	710	1000
Fuel Cost Parameter	2.4	2.9
Fleet	3	2

Question:

Determine the best timetable and aircraft route, assuming that:

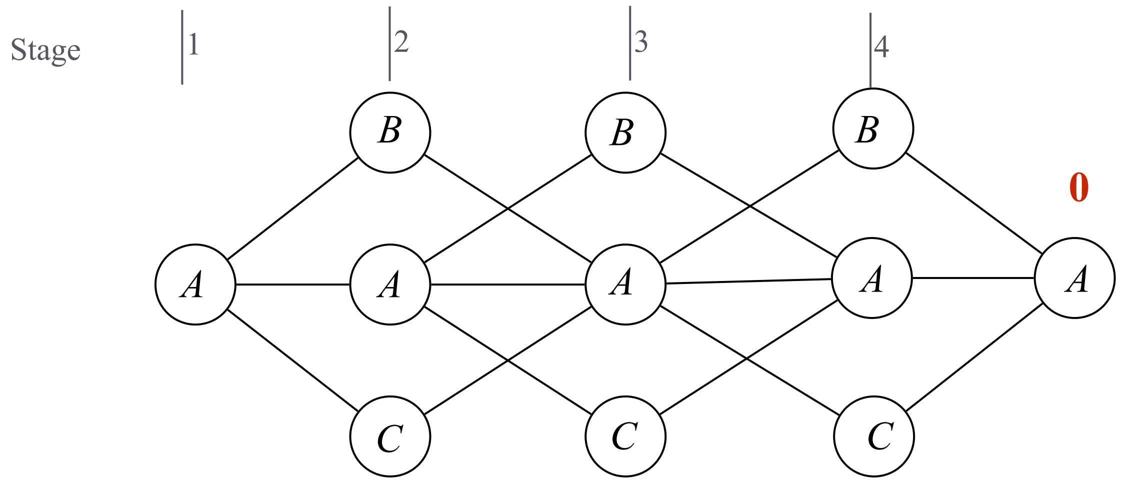
- aircraft start and end their daily routes at the hub
- only hub-spoke routes are considered
- no connections - demand is given per flight route
- passengers can adapt their flight time by max 2 hours (capture time band)

TU Delft

Implementation

Approach:

- The schedule of each aircraft is a sequential time-space problem (example for 1 hub, 2 spokes & 4 time stages)

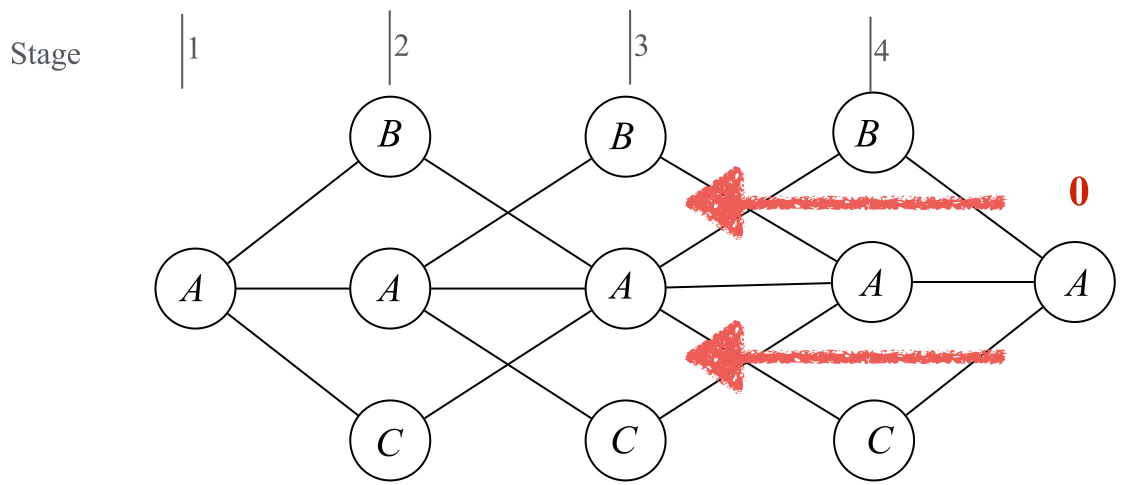


- Assume cost 0 at the end of the day
-
-

Implementation

Approach:

- The schedule of each aircraft is a sequential time-space problem (state space example for 1 hub, 2 spokes & 4 time stages)

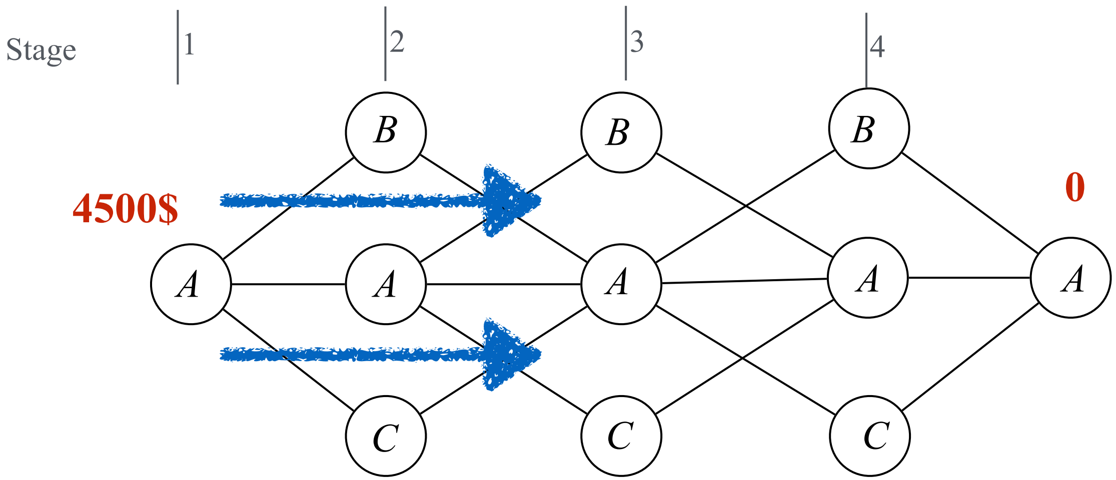


- Assume cost 0 at the end of the day
- Back propagation of the profits, until stage 1
  - Keep track of the max profit and '*node to fly to*'
-
-

Implementation

Approach:

- The schedule of each aircraft is a sequential time-space problem (state space example for 1 hub, 2 spokes & 4 time stages)

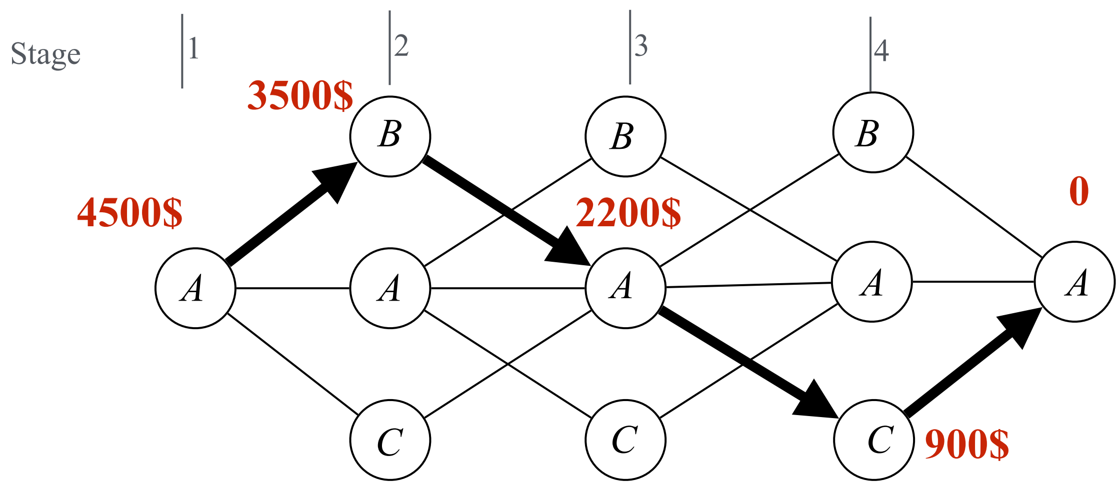


- Assume cost 0 at the end of the day
- Back propagation of the profits, until stage 1
  - Keep track of the max profit and '*node to fly to*'
- Forward propagation to create the schedule, following the '*node to fly to*'

Implementation

Approach:

- The schedule of each aircraft is a sequential time-space problem (state space example for 1 hub, 2 spokes & 4 time stages)



- Assume cost 0 at the end of the day
- Back propagation of the profits, until stage 1
  - Keep track of the max profit and '*node to fly to*'
- Forward propagation to create the schedule, following the '*node to fly to*'

Implementation

Approach:

- The schedule of each aircraft is a sequential time-space problem

- Per iteration
  - Solve the scheduling problem per aircraft type (if there are still aircraft of that type) - *DP Algorithm*
  - Compare the profit obtained per each type
    - If, at least, one aircraft type has profit > 0
      - Select the schedule of the aircraft type with the highest profit and add it to the final solution
      - Remove the demand transported in the flights in that schedule
      - Restart all over again
    - If no aircraft type has profit > 0
      - Stop the algorithm

TU Delft

Implementation

Some notes

- Pre-compute costs, demand per hour, revenue (assuming demand in the time band) and operational constraints (e.g., range or runway)

- DP Algorithm
  - Discretise the time in time steps (in this case, of 6 min)
  - Built matrixes which represent the state space - one matrix for the accumulated profit and one for the '*node to fly to*'
  - The last flight as to arrive at the airport before 24h
    - Initial profit matrix
      - = 0 for hub airport at time = 24h
      - = very large negative value for all other airports at time = 24h
      - = 0 for all other cells
    - Initial '*node to fly to*'
      - = node itself
  - The value of profit at time = 0h at the hub airport is the total profit of using that aircraft type.

TU Delft

Results

Total Profit - **29683**

All aircraft in the fleet are used

Routes for the first 3 aircraft:

Aircraft	**Type**	**Total profit**	**Utilization [h/day]**		
Aircraft 1:	Type 2	14268	9.21		
Departure Time	**Route**	**Arrival Tim**	**Passengers**	**Load-factoı**	**Profit [$]**
6 h 18	EHAM - EDDF	7 h 12	120	1	2064.47
7 h 42	EDDF - EHAM	8 h 36	120	1	2064.47
9 h 06	EHAM - EDDF	10 h 00	118	0.98	1984.27
10 h 30	EDDF - EHAM	11 h 24	49	0.41	-782.56
11 h 54	EHAM - EDDF	12 h 48	120	1	2064.47
13 h 54	EDDF - EHAM	14 h 48	78	0.65	380.31
15 h 42	EHAM - EDDF	16 h 36	120	1	2064.47
17 h 06	EDDF - EHAM	18 h 00	120	1	2064.47
18 h 30	EHAM - EDDF	19 h 24	76	0.63	300.11
19 h 54	EDDF - EHAM	20 h 48	120	1	2064.47

Aircraft	**Type**	**Total profit**	**Utilization [h/day]**		
Aircraft 2:	Type 2	11863	6.41		
Departure Time	**Route**	**Arrival Tim**	**Passengers**	**Load-factoı**	**Profit [$]**
6 h 30	EHAM - EDDF	7 h 24	120	1	2064.47
7 h 54	EDDF - EHAM	8 h 48	120	1	2064.47
11 h 54	EHAM - EIDW	13 h 12	63	0.53	59.66
15 h 06	EIDW - EHAM	16 h 24	120	1	3545.58
16 h 54	EHAM - EDDF	17 h 48	120	1	2064.47
19 h 54	EDDF - EHAM	20 h 48	120	1	2064.47

Aircraft	**Type**	**Total profit**	**Utilization [h/day]**		
Aircraft 3:	Type 1	1412	8.6		
Departure Time	**Route**	**Arrival Tim**	**Passengers**	**Load-factoı**	**Profit [$]**
6 h 00	EHAM - EIDW	7 h 24	50	1	259.48
7 h 54	EIDW - EHAM	9 h 18	50	1	259.48
11 h 54	EHAM - EDDF	12 h 48	50	1	52.02
14 h 06	EDDF - EHAM	15 h 00	50	1	52.02
15 h 30	EHAM - ESSA	17 h 24	50	1	394.52
17 h 54	ESSA - EHAM	19 h 48	50	1	394.52

TU Delft

Dynamic Programming

What is the challenge of using Dynamic programming?

The three curses of dimensionality

?

Curious?
Check Approximate Dynamic Programming or Reinforcement Learning

TU Delft

Summary of this lecture

What should I know by now?

- explain the scheduled development process
- understand the implications of frequency in demand
- describe the fleet planning problem
- explain the concept of time-space networks
- develop modelling techniques to tackle the fleet planning problem
- formulate the IFAM and explain the (most common) solution technique
- describe the dynamic programming technique and the concept of optimal sequential decision process
- explain the Bellman's principle of optimality
- apply the dynamic programming technique to solve the aircraft routing problem
- explaining the 'curse of dimensionality' for dynamic programming

TU Delft