

Air Navigation Systems ATFM Laboratory

1 ATFM & Arrival Manager

This laboratory consists in the resolution of an Air Traffic Flow Management (ATFM) problem minimizing the total delay of the network. A simplified ATFM model will be used on a toy example. Also, an Arrival Manager problem is to be stated and solved.

1.1 The ATFM example

The airspace can be represented as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of directed arcs interconnecting the nodes. Notice that every node represent the entrance into a sector and every arc cross a sector.

Let us consider the flight map shown in Figure 1. The example includes different types of elements: airports $\{A, B, C\}$, sectors $\{I, II, III, IV\}$ flight waypoints $\{a, b, e, f\}$ in-between two sectors, and $\{q(A), q(B)\}$ waypoints within the boundary of the respective departure airports. So, graph $\mathcal{G}$ is defined as follows:

$$\mathcal{N} = \{A, q(A), B, q(B), C, a, b, e, f\}$$

$$\mathcal{A} = \{(A, q(A)), (q(A), a), (B, q(B)), (a, b), (q(A), e), (q(B), f), (b, C), (e, B), (f, C)\}$$

The route of every flight is a path defined as a the ordered sequence of border points (connected by airways) and airports that it follows:

- Flight 1, f_1 : Route $\{A \rightarrow q(A) \rightarrow a \rightarrow b \rightarrow C\}$
- Flight 2, f_2 : Route $\{A \rightarrow q(A) \rightarrow e \rightarrow B\}$

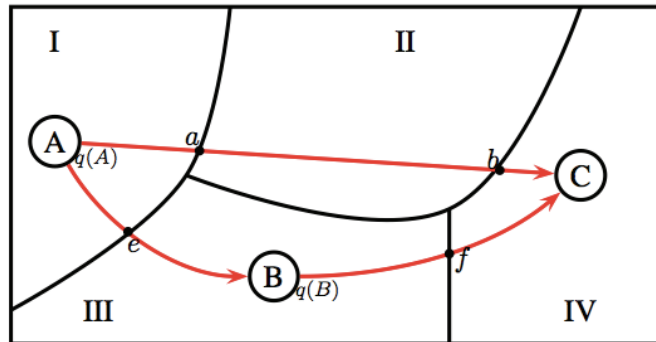


Figure 1: Example

- Flight 3, f_3 : Route $\{B \rightarrow q(B) \rightarrow f \rightarrow C\}$

For instance, for flight f_1 the subgraph $\mathcal{G}_{f_1} = (\mathcal{N}_{f_1}, \mathcal{A}_{f_1})$ is defined as:

$$\begin{aligned}\mathcal{N}_{f_1} &= \{A, q(A), a, b, C\} \\ \mathcal{A}_{f_1} &= \{(A, q(A)), (q(A), a), (a, b), (b, C)\}\end{aligned}$$

1.2 Notation

Let us describe the mathematical model definition and notation into four distinguished groups: input sets, preprocessing sets, input parameters, and decision variables. Input sets and input parameters are given. Preprocessing sets are computed based on the previous ones. Decision variables provide the solution to the problem.

1.2.1 Input sets

$\mathcal{F}$	$\equiv$	set of flights,
$\mathcal{K}$	$\equiv$	set of airports,
$\mathcal{T}$	$\equiv$	$\{0, \dots, T\}$ set of time periods, being T the maximum allowed time
$\mathcal{C}$	$\equiv$	$\{(f', f) f' \text{ is continued by } f\}$ set of paris of flights that are continued,
$\mathcal{S}$	$\equiv$	set of sectors,
$\mathcal{N}_f$	$\equiv$	ordered set of nodes that define the (only) available route for flight $f \in \mathcal{F}$,
$\mathcal{A}_f$	$\equiv$	$\{(n, m) n, m \in \mathcal{N}_f, \}$ set of arcs that define the (only) available route for flight $f \in \mathcal{F}$
$\mathcal{N}_{jf}$	$\equiv$	ordered set of nodes that belongs to sector $j \in \mathcal{S}$, $f \in \mathcal{F}$
$\mathcal{A}_{jf}$	$\equiv$	set of arcs $(n, m) \in \mathcal{A}$ that belongs to sector $j \in \mathcal{S}$, $f \in \mathcal{F}$
$orig_f$	$\equiv$	airport of departure of flight f ,
$dest_f$	$\equiv$	airport of arrival of flight f ,

1.2.2 Calculated sets in preprocessing

$T_f^n = [\underline{T}_n^f, \overline{T}_n^f]$	$\equiv$	set of feasible time periods for flight f to arrive in node n ,
$\underline{T}_n^f$	$\equiv$	first time period in the set T_n^f ,
$\overline{T}_n^f$	$\equiv$	last time period in the set T_n^f ,

1.2.3 Input parameters

d_f	$\equiv$	scheduled departure time of flight f ,
a_f	$\equiv$	scheduled arrival time of flight f ,
s_f	$\equiv$	turnaround time of an airplane after flight f ,
$D_k(t)$	$\equiv$	departure capacity of airport k at time t ,
$A_k(t)$	$\equiv$	arrival capacity of airport k at time t ,
$S_j(t)$	$\equiv$	capacity of sector j at time t ,
r_n^f	$\equiv$	scheduled arrival time to node n for flight f ,
$l_{fn,m}$	$\equiv$	$(n, m) \in \mathcal{A}_f$ scheduled travel time for flight f to arrive at node n through link (n, m)
$\mathbb{G}_f$	$\equiv$	maximum ground holding time delay units for flight f
$\mathbb{A}_f$	$\equiv$	maximum air time delay units for flight f
c_f^a	$\equiv$	air delay unit cost for flight f
c_f^g	$\equiv$	ground delay unit cost for flight f

Notice that the maximum duration allowed for flight f has the expression $r_f - d_f + \mathbb{A}_f$

1.2.4 Variables

$$w_{n,t}^f = \begin{cases} 1, & \text{if flight } f \text{ arrives at node } n \text{ by time } t \\ 0, & \text{otherwise.} \end{cases}$$

This definition of the decision variables $w_{n,t}^f$ means that if flight f arrives at time t at node n then variables of time period t and subsequent ones will be set to 1 (i.e., $w_{n,\tau}^f = 1 \ \forall \tau \geq t$). Notice that $w_{n,t}^f$ is defined in $f \in \mathcal{F}, n \in \mathcal{N}_f, t \in T_n^f$.

1.3 Preprocessing

The preprocessing scheme is very important for any optimization model. In this phase the time sets are considered in order to fix variables and eliminate redundant constraints. The following variables are to be preprocessed (use either one equation or the other):

$$\begin{aligned} r_{n_i}^f &= r_{n_{i-1}}^f + l_{f,n_{i-1},n_i} \\ r_{n_i}^f &= d_f + \sum_{n_1 \dots n_i} l_{f,n_{i-1},n_i} \end{aligned}$$

$r_{n_i}^f$ is the scheduled time period for flight f to arrive at node $n \in \mathcal{N}_f$ through the arc of the (only) route and without adding possible delays. Notice that

$$r_{orig}^f = d_f.$$

$$T_n^f = [r_n^f, \min\{r_n^f + \mathbb{G}_f + \mathbb{A}_f, T\}].$$

Thus, $T_n^f = [\underline{T}_n^f, \overline{T}_n^f]$ represent a time vector of possible node overflights times. $\overline{T}_n^f$ accounts for maximum ground and air delay (or the maximum admissible time). Notice that if the node is an airport (or the transition from the airport to the upper airspace) the air delay $\mathbb{A}_f$ does not apply.

1.4 Mathematical model

1.4.1 Objective function

$$\begin{aligned} \min \sum_{f \in \mathcal{F}} \left(\sum_{t \in T_{dest_f}^f} (c_a^f) \cdot (w_{dest_f,t}^f - w_{dest_f,t-1}^f) \cdot (t - r_n^f) + \dots \right. \\ \left. \dots \sum_{t \in T_{orig_f}^f} (c_g^f - c_a^f) \cdot (w_{orig_f,t}^f - w_{orig_f,t-1}^f) \cdot (t - d_f) \right) \end{aligned}$$

1.4.2 Capacity constraints

$$\sum_{f \in \mathcal{F}: orig_f = k} (w_{k,t}^f - w_{k,t-1}^f) \leq D_k(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}; \quad (1)$$

$$\sum_{f \in \mathcal{F}: dest_f = k} (w_{k,t}^f - w_{k,t-1}^f) \leq A_k(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}; \quad (2)$$

$$\sum_{f \in \mathcal{F}: n \in \mathcal{N}_{jf}} (w_{n,t}^f - w_{n+1,t}^f) \leq S_j(t) \quad \forall j \in \mathcal{S}, t \in \mathcal{T}. \quad (3)$$

1.4.3 Connectivity/Network constraints

$$w_{m,t+l_{f,n,m}}^f - w_{n,t}^f \leq 0 \quad \forall f \in \mathcal{F}, n \in \mathcal{N}_f, t \in T_n^f : n \neq dest_f \quad (4)$$

$$w_{orig_f,t}^f - w_{dest_f,t+(r_f-d_f)+\mathbb{A}_f}^f \leq 0 \quad \forall f \in \mathcal{F}, t \in T_{orig_f}^f \quad (5)$$

$$w_{orig_f,t}^f - w_{dest_{f'},t-s_{f'}}^{f'} \leq 0 \quad \forall (f', f) \in \mathcal{C}, t \in T_{orig_f}^f \quad (6)$$

$$w_{n,t-1}^f - w_{n,t}^f \leq 0 \quad \forall f \in \mathcal{F}, n \in \mathcal{N}_f, t \in T_n^f \quad (7)$$

$$w_{n,\overline{T}_n^f}^f = 1 \quad \forall f \in \mathcal{F}, n \in \mathcal{N}_f; \quad (8)$$

1.4.4 Binary character of the variables

$$w_{n,t}^f \in \{0, 1\} \quad \forall f \in \mathcal{F}, n \in \mathcal{N}_f, t \in \mathcal{T}; \quad (9)$$

Constraints (1) and (2) ensure that the number of flights that take off and land respectively on each airport in time period t does not exceed their correspondent capacities. Constraints (3) ensure that the flow in each sector does not exceed its allowed capacity in every time period t . Notice that each node n can be either an airport or a waypoint, and the routes are defined as ordered sequences. Constraints (4) represent the sectors connections; they impose that if a flight has arrived within time period $t + l_{f,n,m}$ to sector border point m , then it must had arrived to node n by time period t , where waypoint m is the one that follows node n as the arc (n, m) in flight f . Constraints (5) ensure that flight f does not exceed the maximum acceptable duration, i.e., does not exceed the maximum air delay time periods $\mathbb{A}_f$. Constraints (6) propagate the delay of flight f to its continuing flights: Flight f can not take off until flight f' has landed and spends at least $s_{f'}$ time periods on the ground to prepare for next departure (turnaround operation). Constraints (7) represent the time connection: If a flight has arrived at border point n by time period t , then $w_{n,t'}^f$ must be 1 for every time period $t' \geq t$. Constraints (8) apply to all flights f and nodes n , setting the binary variable value at the maximum permitted time to 1 (the last possible overflight time must be satisfied). Constraints (9) impose the binary character of the variables.

1.5 Input parameters' data

Notice that the input data are given in unit times (in other words, we are assuming constant speed and we have already calculated the routes' distances so that everything can be translated into time). Notice also that some of the above defined input parameters have no associated data. This is either because the value can be built upon the given data and/or can be extracted from the structure of the airspace (notice that the maximum allowed time is $T = 40$).

$$\begin{aligned}d_{f_1} &= 1; d_{f_2} = 1; d_{f_3} = 17 \\a_{f_1} &= 15; a_{f_2} = 13; a_{f_3} = 23 \\s_{f_1} &= 5; s_{f_2} = 5; s_{f_3} = 5 \\D_A(t) &= 1; D_B(t) = 1; D_C(t) = 1 \\A_A(t) &= 1; A_B(t) = 1; A_C(t) = 1 \\S_I(t) &= 4; S_{II}(t) = 4; S_{III}(t) = 4; S_{IV}(t) = 4 \\l_{f_1,(A,q(A))} &= 0; l_{f_1,(q(A),a)} = 5; l_{f_1,(a,b)} = 8; l_{f_1,(b,C)} = 1; \\l_{f_2,(A,q(A))} &= 0; l_{f_2,(q(A),e)} = 5; l_{f_2,(e,B)} = 7; \\l_{f_3,(B,q(B))} &= 0; l_{f_3,(q(B),f)} = 3; l_{f_3,(f,C)} = 3; \\\mathbb{G}_{f_1} &= 2; \mathbb{G}_{f_2} = 3; \mathbb{G}_{f_3} = 3 \\\mathbb{A}_{f_1} &= 5; \mathbb{A}_{f_2} = 4; \mathbb{A}_{f_3} = 4 \\c_{f_1}^a &= 3; c_{f_2}^a = 2.5; c_{f_3}^a = 2.5 \\c_{f_1}^g &= 1.5; c_{f_2}^g = 1.5; c_{f_3}^g = 1.5\end{aligned}$$

2 Assignment

1. **ATFM Problem:** Considering the example described in Section 1, the assignment consists in finding the 4D routes, i.e., waypoints (given as nodes) + altitudes (assumed to be constant) + time, for the three flights considered so that the total delay is minimized while fulfilling capacity constraints and network constraints. Consider the model and data provided in Section 1. Either AMPL or PYTHON languages can be used to code and interact with solver Gurobi/CPLEX/glpk. For the sake of simplicity, the data file is given.

The steps to be followed are as follows:

- Declare and initialize all input sets according to the given flight structure. See Coding Tip 1 and Coding Tip 2. Note that the initialisation is given in the data file.
- Declare and initialize all input parameters according to the given data. See Coding Tip 3. Initialisation is given in the data file.
- Do the preprocessing and build the variables $r_{n_i}^f$ and T_n^f . See Coding Tip 4. This preprocessing is given in the data file.
- Define the binary variables w . See Coding Tip 5
- State the objective function. See Coding Tip 6
- State the capacity and network constraints. See Coding Tip 7
- Solve using Gurobi/CPLEX (or other existing Integer Programming solvers).

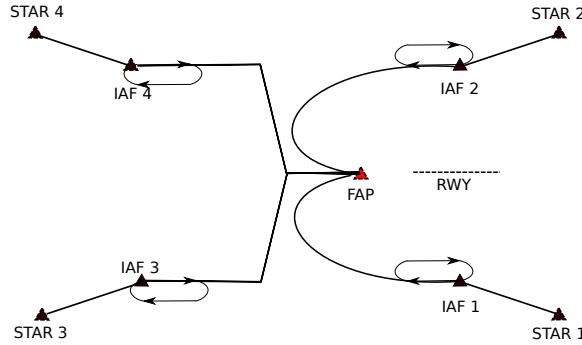


Figure 2: AMAN structure

2. **Arrival Manager:** State and solve an Arrival Manager Problem using network optimization. Notice that this is a tool approximation controllers have to sequence the traffic before the FAP/FAF. Thus is rather a tactical operational problem (to be run once aircraft are in the air), instead of the flow management one (which is always run before departure). The goal is to illustrate the modelling potentiality and versatility of network optimization. Steps to be follow are as follows:

- Consider the network structure given in Figure 2.
- Demand is composed by 12 flights, namely: f1 to f3 on STAR1, f4 to f6 on STAR2, f7 to f9 on STAR3 and f10 to f12 on STAR4
- Scheduled departure times are $t=1$ for f1, f4, f7, f10; $t=2$ for f2, f5, f8, f11, and $t=3$ for f3, f6, f9, and f12.
- Legs' time-distances are 4 time units from STAR1 to IAF1 and 5 time units from IAF1 to FAP (same for STAR2); 5 times units from STAR3 to IAF3 and 5 time units from IAF3 to FAP (same for STAR4)
- With respect of the "capacity side", notice that the bottleneck of the problem is the FAF/FAP, where aircraft must have been sequenced, metered, and merged towards RWY. Thus, you have to define a Capacity restriction at that particular point, e.g., capacity=1 per minute.
- Delays can only be assigned either at the STAR (assuming the aircraft can decelerate before reaching it) and at the established holding patterns (at the IAFs).
- The maximum permitted delay at the STARs is 3 time units; and the maximum number of aerial holdings (at the IAFs) permitted is 5 (each equivalente to 1 time unit)
- Notice that, besides the selection of the parameters, the equations of the ATFM model hold except for small changes.
- Minimize the delay while sequencing all aircraft towards RWY.



2.1 Question to be assessed

The questions the alumni should assess are as follows:

1. Solve the ATFM Problem and indicate the minimum delay for the above given configuration. **8.5 pt.**
2. State and solve an Arrival Manager Problem. **1.5 pt.**

2.2 Delivery and evaluation

You should prepare a report (pdf format) with maximum 4 pages (notice that I will not read beyond this point), which should include:

- The solution to the ATFM problem (using tables/Figures).
- With regard of the AMAN problem: include the statement, capacity-demand conditions (parameters in your problem at your discretion), and the solution to the problem.
- should NOT include index, introduction, software, conclusions, bibliography or the ATFM mathematical model (just refer to the statement)

The delivery should also include the code (with the values given in this statement) ready to be launched (besides path specifications). The evaluation will be fundamentally based on the code the ATFM (if the problem runs and converges to the optimal solution, you will get it OK) and both the code and statement of the AMAN.

Some other specifications include:

- Submission due to November 9 (Tues.), 15.00 at Aula global.
- The allowance of groups of maximum four people each.

3 Coding tips

The sequel includes some tips you might want to follow when coding the exercise:

3.1 Coding tip 1

```
# For the definition of a set, make it always "ordered"
```

```
set F ordered; # set of flights  
let F:={"f1","f2","f3"}
```

```
# Recall that the declaration should go in the model file  
# Recall that the assignment should go in the data file
```

```
# To assign values that depend on the parameter "f1", one should proceed:
```

```
let d_f["f1"]:=1;
```

3.2 Coding tip 2

```
# Declare sets that depend on existing sets, e.g., nodes for flight f:
```

```
set N_f {k in F} ordered;
```

```
# The initialization would be:
```

```
let N_f["f1"]:={"A","q(A)","a","b","C"};
```

3.3 Coding tip 3

```
# Import data file and display all input sets and parameters.  
# Make sure everything is consistent.
```

3.4 Coding tip 4

```
# Do the preprocessing.  
# T_fn should be a set.
```

```
set T_fn{k in F,N_f[k]} ordered;
```

```
# Display the value of T_fn and check they are correct:
```

```

set T_fn[f1,A] := 1 2 3;
set T_fn[f1,'q(A)'] := 1 2 3;
set T_fn[f2,A] := 1 2 3 4;
set T_fn[f2,'q(A)'] := 1 2 3 4;
set T_fn[f3,B] := 17 18 19 20;
set T_fn[f3,'q(B)'] := 17 18 19 20;
set T_fn[f1,a] := 6 7 8 9 10 11 12 13;
set T_fn[f1,b] := 14 15 16 17 18 19 20 21;
set T_fn[f1,C] := 15 16 17 18 19 20 21 22;
set T_fn[f2,e] := 6 7 8 9 10 11 12 13;
set T_fn[f2,B] := 13 14 15 16 17 18 19 20;
set T_fn[f3,f] := 20 21 22 23 24 25 26 27;
set T_fn[f3,C] := 23 24 25 26 27 28 29 30;

# Do not proceed on until this has been reached!!

```

3.5 Coding tip 5

Variables w only apply within the corresponding time intervals T_n^f . However, for the sake of simplicity in the model they are defined to be valid within the whole time interval $\mathcal{T}$. This is because capacity constraints do apply to the whole time interval. Therefore, one can define variables w to be valid within the whole time interval $\mathcal{T}$ and thereafter specify a restriction as follows:

```

subject to enforcement {k in F, j in N_f[k], i in T: i > last(T_fn[k,j])}: w[k,j,i]=1;
subject to enforcement_2 {k in F, j in N_f[k], i in T: i < first(T_fn[k,j])}: w[k,j,i]=0;
display {k in F, j in N_f[k], i in T: i > last(T_fn[k,j])}: k,j,i;

# the display command is a good practice to visualize what the assignment loop is doing.
# notice that the loop goes through F (flights), N_f (nodes per flight) and T (whole time) such

```

3.6 Coding tip 6

The objective function goes with summatories:

```

minimize delay:
sum{k in F}
(
...
);

```

```

# Display the delay function and the indexes to be sure it is properly coded

```

3.7 Coding tip 7

For the definition of the constraints, you need to define loops going into your sets with some restrictions. Also, since the sets are not numerical, you will need to use this kind of instructions:

```
# Instructions of interest.
# ord(i) # goes through a set with index i in order (element 1, element 2, etc.)
#ord(e, ORD_SET)    # The position of e in ORD_SET
#first(ORD_SET)      # The first element in ORD_SET
#last(ORD_SET)       # The last element in ORD_SET
#prev(e, ORD_SET)    # The element before e in ORD_SET
#next(e, ORD_SET)    # The element after e in ORD_SET
#member(i, ORDSET)   # The element at position i in ORD_SET
```

As an example:

```
subject to capacity_constraint_3{j in S, i in T}:
(sum{k in F, jaux in N_fS[k,j]: jaux <> last(N_S[k,j])}
(w[k,jaux,i]-w[k,next(jaux,N_S[k,j]),i])) <= S_j[j,i];
```