An optimization-based procedure for the operative schedule design of an airline

Daniel Ramón-Lumbierres, Victoria Rebillas, Mohammad Kouhi

October, 2023

vueling

WE LOVE PLACES

Contents

Introduction

vueling

WE
LOVE
PLACES

Vueling Network planning process:

Illustrative

Long-term planning | Mid-term planning | Tactical planning

... D-90 D-55 D-15 D-0

Network planning

Profitability:
- Initial Flight schedule

Scheduling

Structural capacity:
- Rescheduling
- A/C Routing
- Operability
- Flight addition
- Flight cancelation

Crew planning

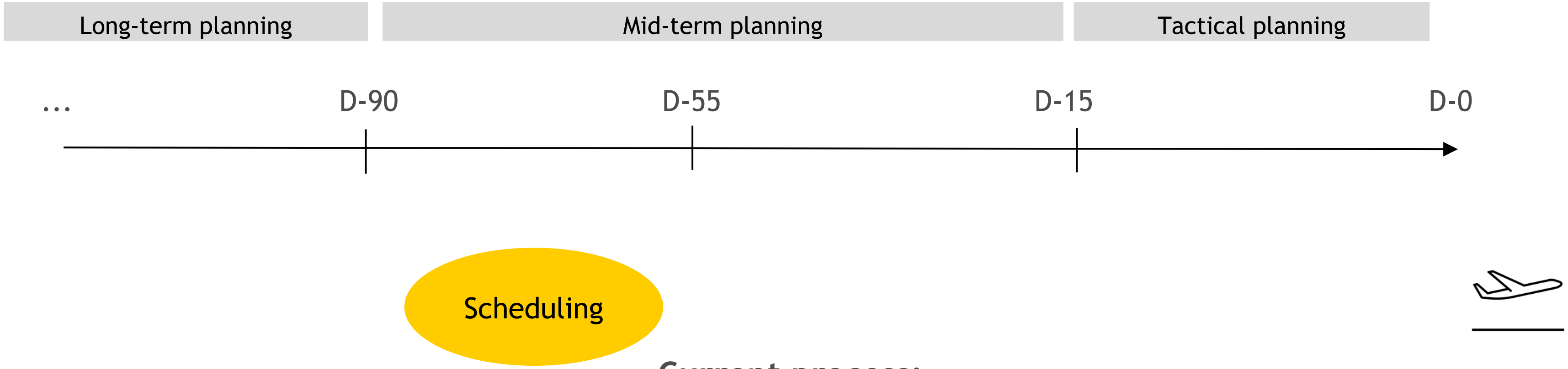
Pairing & Roster

Short term planning

- Tail assignment
- Close-in ops adjustments

Currently, Scheduling process is manual:

Illustrative



Current process:

- Optimality is not guaranteed
- Time consuming process
- Scheduling errors
- Not all requirements are met
- Only small modifications are performed

Goals and benefits:

Goals

- Modify the original network so that aircraft utilization is maximized
  - At this point, expected revenue and operational cost are not considered
- Minimize flights reschedule with respect to the original network
  - within the maximal aircraft utilization
- Facilitate the network operability at further steps

Expected Benefits

- Scheduling department
  - From manual to automatic
  - More flexibility and time-efficiency
- Operations department
  - Friendly flight patterns to operate
- Vueling
  - Profit increase due to aircraft use

Problem definition

Problem definition

Decision objects in Network Adjustment are *subrotations*:

- Sequences of 2, 3 or 4 flights with matching airports
- The beginning and ending airport is their *base*

The main problem input is an original network, where

- regular flights have initial departure and arrival times,
- they are originally associated to certain base airport,
- as well as to a given *subfleet*: A319, A320 or A321

Original network:

Colors here represent different Vueling subfleet

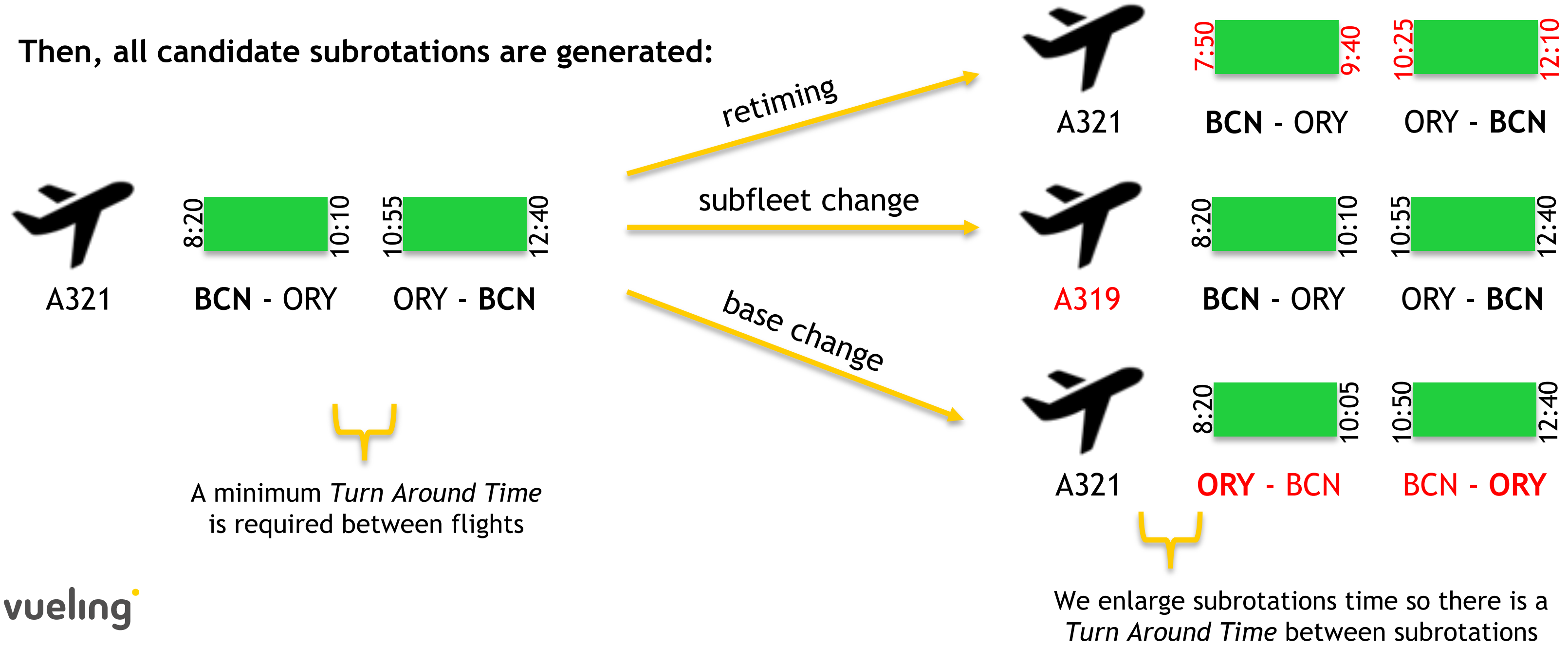
Most of the subrotations are *Round Trips*, composed by two flights

A *rotation* is the whole sequence an aircraft performs each day

Firstly, a list of optional flights are added:

- The objective is to fulfill the flying network
- They are useful in case of network modifications or cancellations
- Unlike the original ones, these flights can be scheduled or not

Then, all candidate subrotations are generated:



Model development

The objective of the optimization process is to select subrotations to operate:

- out of the combinations of original subrotations we have generated

$$\begin{array}{ll} original\ subrotations & \boldsymbol{S^{original}} \\ optional\ flights & \boldsymbol{F^{optional}} \end{array} \Longrightarrow \begin{array}{ll} mandatory\ subrotations & \boldsymbol{S^{M}} \\ optional\ subrotations & \boldsymbol{S^{O}} \end{array}$$

These candidate subrotations are copies of the original ones

- with modified times, subfleet and/or base airport
- the same flight f appears in several candidate subrotations $s \in S_f$
- unlike in the original network, where each flight appears exactly once

Then, we define the following decision variables:

$$\boldsymbol{x_s} \in \{\boldsymbol{0}, \boldsymbol{1}\}, \qquad \forall s \in S = S^M \cup S^O$$

- There is a binary variable for each candidate subrotation: we must generate them carefully!

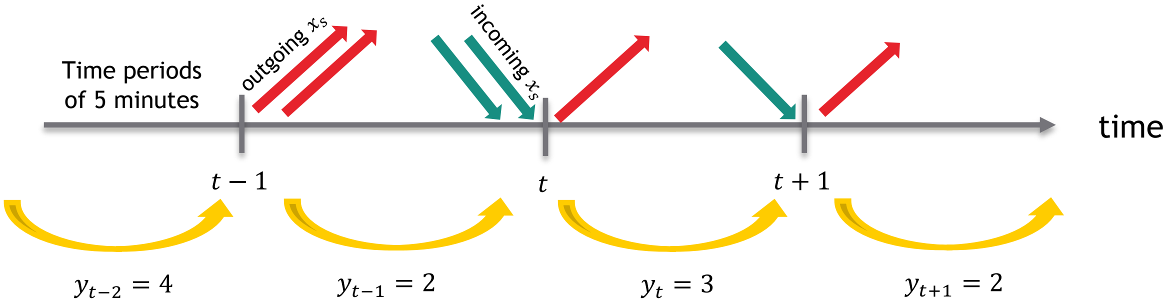
Likewise, we define variables to include conditions on the number of aircrafts

- Subfleet of each base, indexed by b, operate along a set of time periods T_b
- Besides, there is a number of available aircraft denoted as n_b^{AC}

We want to compute how many aircraft are at their base:

$$\boldsymbol{y_{bt}} \in \{\mathbf{0}, \ldots \boldsymbol{n_b^{AC}}\}, \qquad b \in Base \times Subfleet, \qquad t \in T_b$$

- y_{bt} computes the number of aircraft at their base b at the beginning of time period t



In practice,
$0 \leq y_{bt} \leq n_b^{AC}$

The daily optimization model is formulated as follows:

max	$\sum_{s \in S} duration_s \cdot x_s$		Maximize flying time
$s.t.$	$y_{bt} + \sum_{\substack{s \in S_b \\ s\ ends\ at\ t}} x_s = y_{b\ t+1} + \sum_{\substack{s \in S_b \\ s\ begins\ at\ t}} x_s$	$b \in B \times Sf$ $t \in T_b$	Balance constraints between incoming and outgoing aircraft
	$\sum_{s \in S_f} x_s = 1$	$f \in F^M$	Mandatory flights must be scheduled once
	$\sum_{s \in S_f} x_s \leq 1$	$f \in F^O$	Optional flights may be scheduled once
	$x_s \in \{0, 1\}, \quad s \in S$ $0 \leq y_{bt} \leq n_b^{AC}, \quad b \in B \times Sf, t \in T_b$		Variables domain Maximum # aircraft

Extensions to the daily problem

Extensions to the daily problem

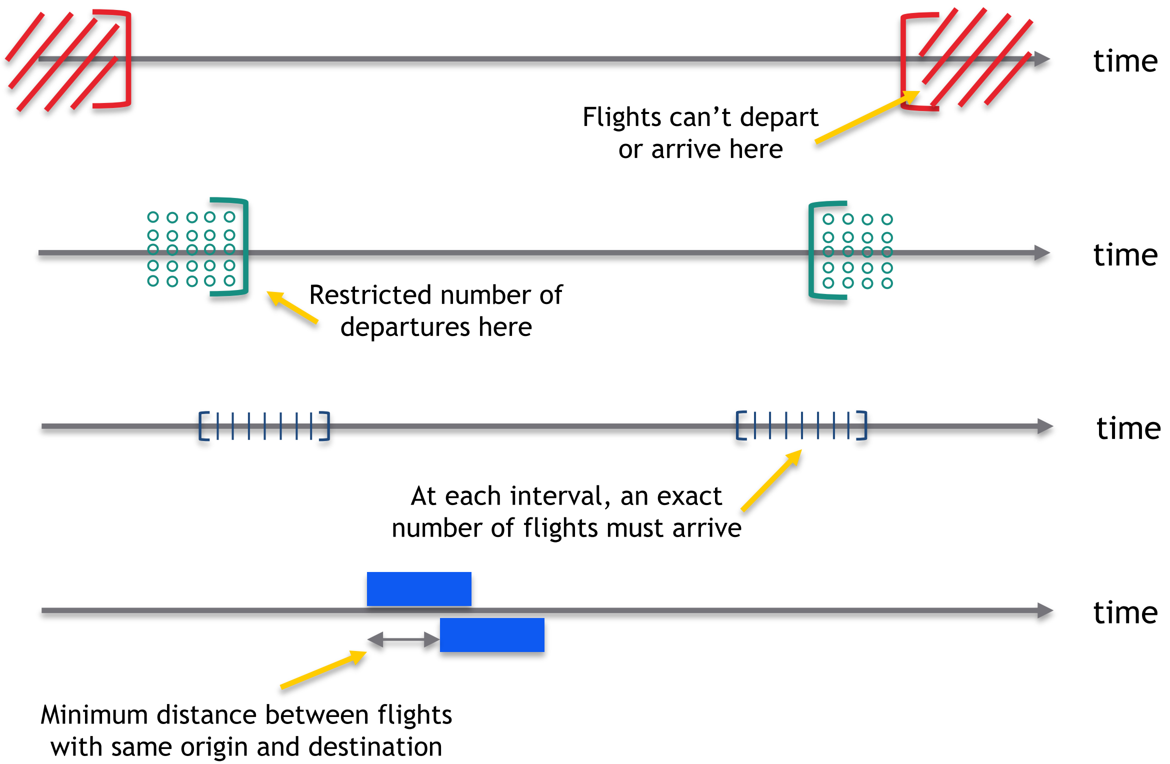
Network Adjustment problem includes several operational limitations:

Airport curfew:
for Departures & Arrivals

Early Departures
& Late Arrivals:

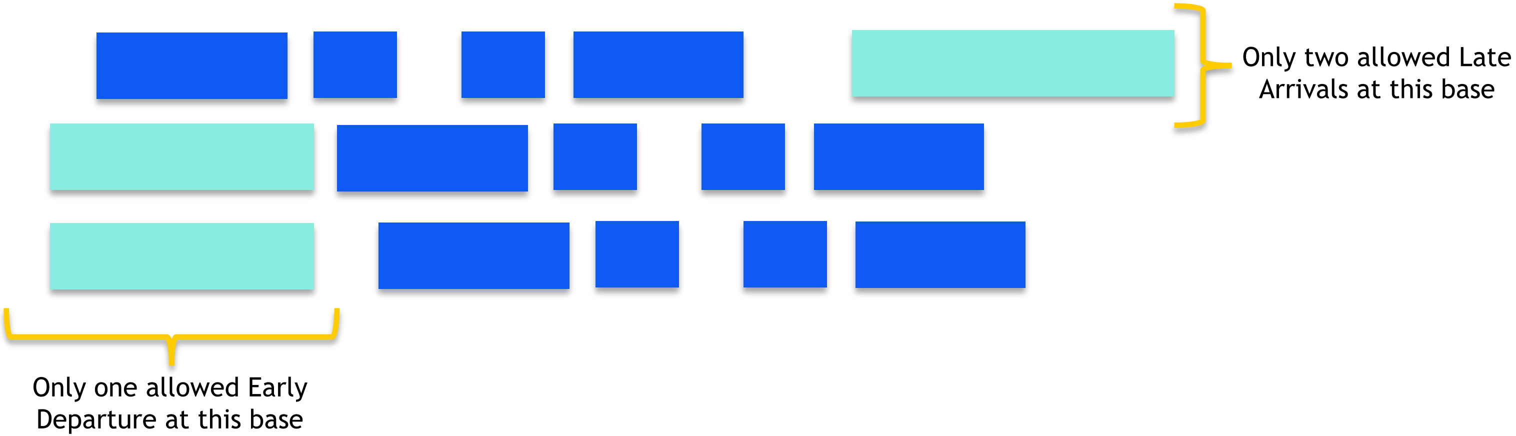
Slots:
for Departures & Arrivals

Non-overlapping:
between flights



Extensions to the daily problem

Early Departure and Late Arrival restrictions are met incorporating certain virtual tasks:

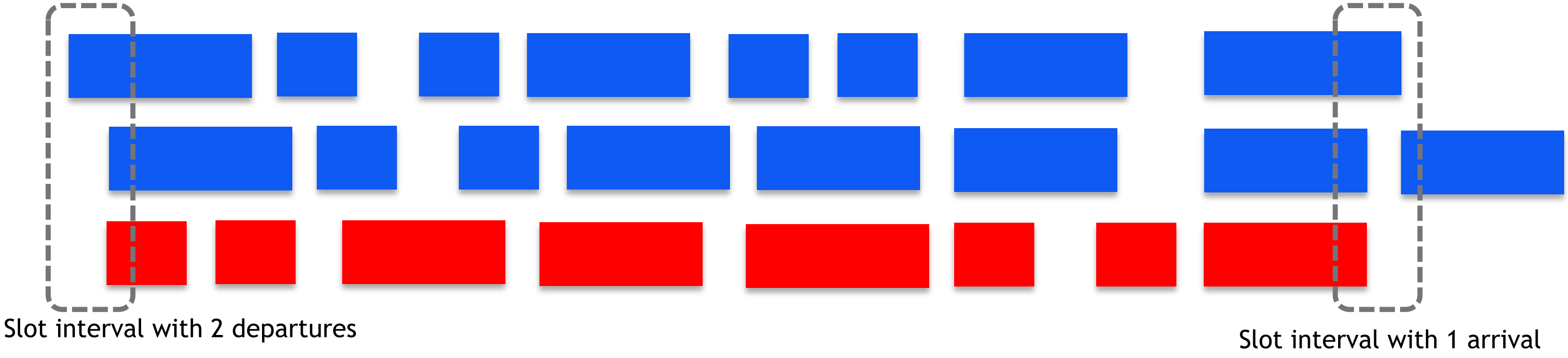


Following the stated formulation, we add Early Departure and Late Arrival tasks as subrotations and fix their associated variables to 1:

$$x_s = 1 \qquad s \in S^{ED} \cup S^{LA}$$

Inclusion of Early Departue & Late Arrival blocks to restrict flights

Slot conditions require an exact number of departures and arrivals at certain airports and time windows:



Slot conditions are implemented as follows:

$$\sum_{\substack{f \in F \\ f \text{ departs at } t}} \sum_{s \in S_f} x_s = slot_t^{DEP} \qquad t \in T^{DEP\ SLOT}$$

A certain number of flights must depart at each departure slot

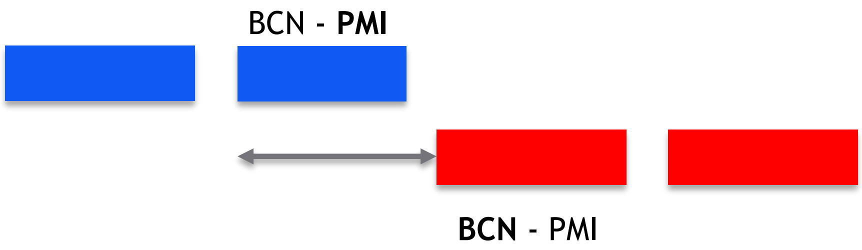
$$\sum_{\substack{f \in F \\ f \text{ arrives at } t}} \sum_{s \in S_f} x_s = slot_t^{ARR} \qquad t \in T^{ARR\ SLOT}$$

A certain number of flights must arrive at each arrival slot

Extensions to the daily problem

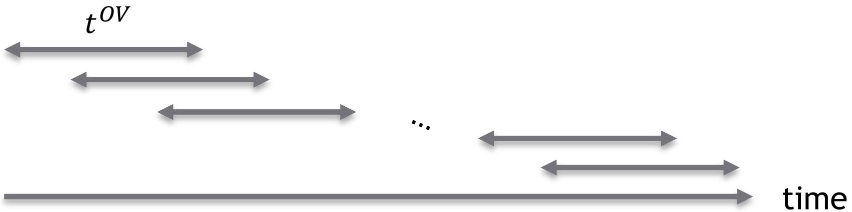
In addition, the problem includes scheduling conditions to avoid overlapping:

- Two flights with same origin and destination are overlapped whenever their depart time difference is lesser than certain quantity, defined by the user



For each origin-destination pair with high frequency, time windows are generated to tackle this:

- Given an overlapping length t^{OV}, we consider all time windows of this length



Each time window define a disjunctive set:

- For each origin-destination pairs with high frequency flights

$$s, r \in D^{OV} \Leftrightarrow s \text{ and } r \text{ contain same O-D flight within a time window}$$

- Finally, all these disjunctive sets are collected into $\Omega := \{D_1^{OV}, \dots, D_k^{OV}\}$

Therefore, non-overlapping conditions are implemented as follows

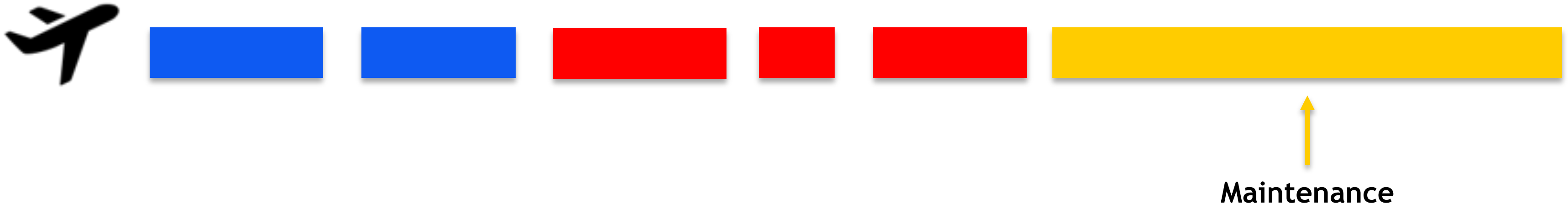
$$\sum_{s \in D_i^{OV}} x_s \leq 1 \qquad D_i^{OV} \in \Omega$$

At most one subrotation of a disjuntive set can be scheduled

Extensions to the daily problem

Maintenance tasks are also included in the problem:

- Daily tasks for Vueling aircrafts
- Usually performed at night, between operative days

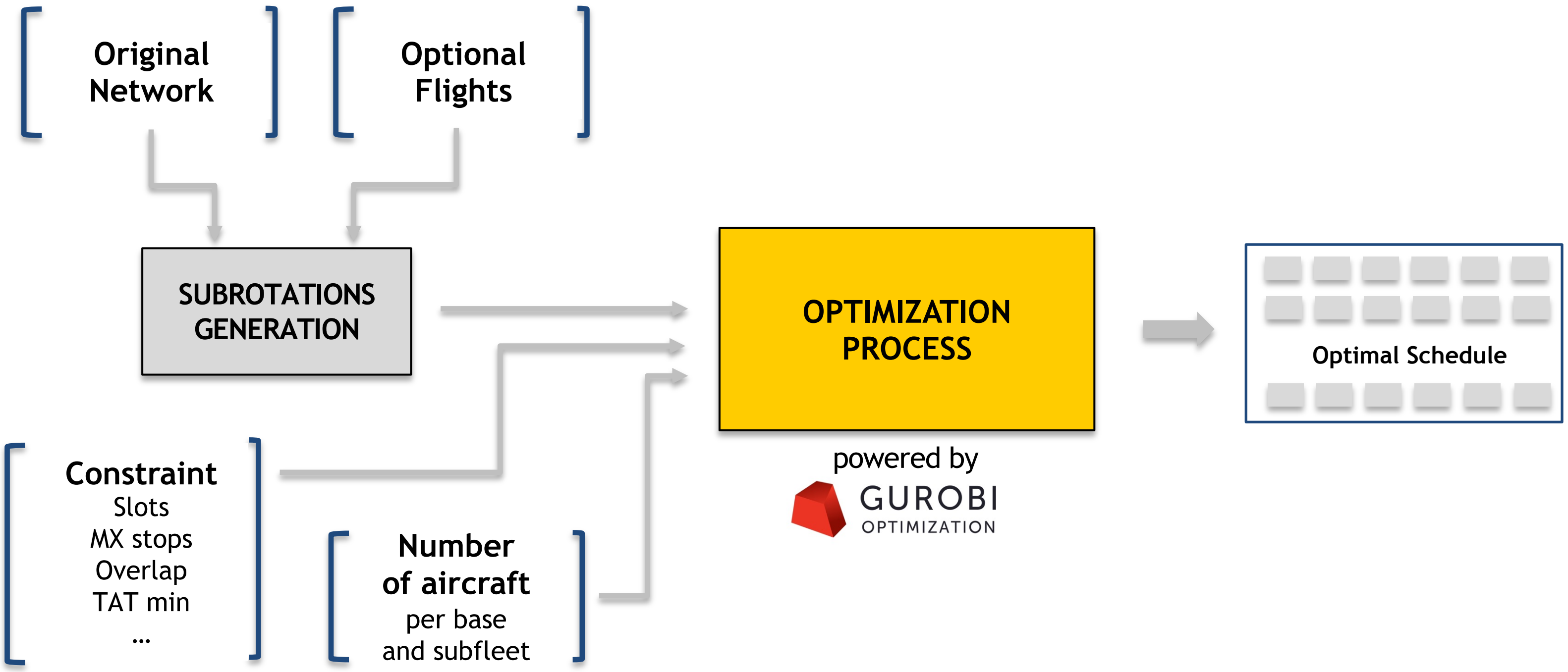


Maintenance tasks are properly assigned in later steps of the decision making process:

- Their conditions depend on factors not included in the scheduling problem

However, their incorporation here facilitates the finding of operative solutions

The optimization process is launched once subrotations are generated



Extensions to the daily problem

Daily optimization process:

- The optimization process for separated days is composed by two optimization runs:

Loop on days

Step 1: max flight time

$$\max \sum_{s \in S} duration_s \cdot x_s$$

$$s.t. \quad problem\ conditions$$

$$optimal\ \mathbf{z}^*$$

Step 2: min retiming

$$\min \sum_{s \in S} retiming_s \cdot x_s$$

$$s.t. \quad problem\ conditions$$

$$s.t. \sum_{s \in S} duration_s \cdot x_s \geq \mathbf{z}^*$$

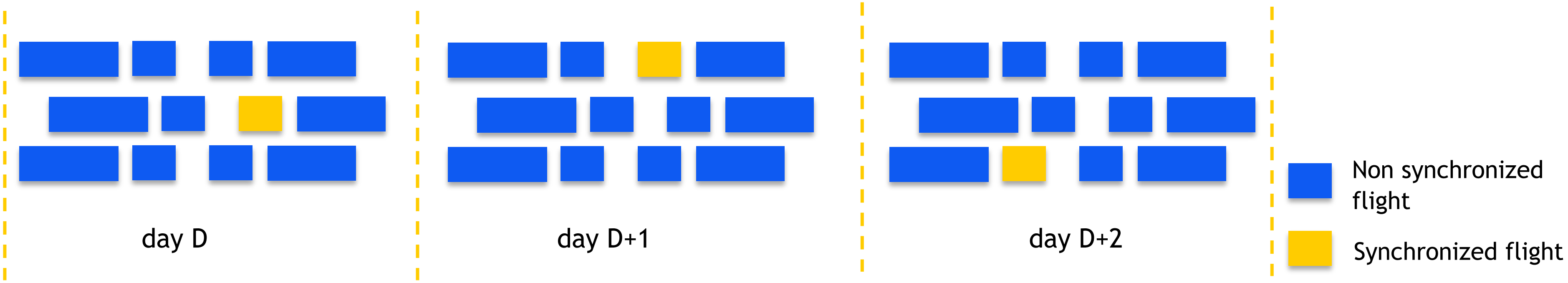
This lexicographic multi-objective approach is performed because:

- There is a priority in fulfilling the flight network as much as posible
- The first optimization problem has lots of alternative optimal solutions
- We want to minimize the impact of rescheduling already published flights

Weekly problem development

Weekly problem developemnt

We want to synchronize certain flight departure times along the week:



Extension to weekly problem with synchronization constraints:

- Synchonization conditions need a larger time horizon as they relate variables of sepatared days

Already declared variables get an extra index $\boldsymbol{d} \in \boldsymbol{D}$ for each operating day

$$(x_s, y_{bt}) \Longrightarrow (x_s^d, y_{bt}^d), \qquad d \in D$$

New variables emerge to compute the time where each synchronized flight is scheduled:

$$z_{ft} \in \{0,1\}, \qquad f \in F^{Sync}, \qquad t \in T$$

such that:

$\sum_{t \in T} z_{ft} \leq 1$	$f \in F^{Sync}$	Synchonized flights must depart at the same time
$\sum_{\substack{s \in S_f \\ f\ departs\ at\ t}} x_s^d \leq z_{ft}$	$f \in F^{Sync}$, $t \in T, d \in D$	Linking constraints for each day and departure time

Weekly optimization process:

- The problem becomes too big, cannot be solved in reasonable time
- If many flights are required to synchronize, the problem may become infeasible

Weekly problem developemnt

Approach through feasibility variables and injection of initial solutions:

- Separate problems are solved for each day
- to provide an initial solution of a synchronization feasibility problem
- which, in turn, provides an initial solution (excluding infeasible flights to synchronize)

Step 0: daily optimization

$$\max \sum_{s \in S} duration_s^{d^1} \cdot x_s^{d^1}$$
$$s.t. \quad problem\ conditions$$

$$\max \sum_{s \in S} duration_s^{d^n} \cdot x_s^{d^n}$$
$$s.t. \quad problem\ conditions$$

Step 0': sync feasibility

$$\min \sum_{d \in D} sync.infeasibilities$$
$$s.t. \quad problem\ conditions$$
$$s.t. \quad relaxed\ sync.conditions$$

$feasible$ $\boldsymbol{F^{Sync}}$

Step 1: max flight time

$$\max \sum_{d \in D} \sum_{s \in S} duration_s^{d} \cdot x_s^{d}$$
$$s.t. \quad problem\ conditions$$
$$s.t. \quad sync.conditions$$

$optimal\ \boldsymbol{z}^*$

Step 2: min retiming

$$\min \sum_{d \in D} \sum_{s \in S} retiming_s^{d} \cdot x_s^{d}$$
$$s.t. \quad problem\ conditions$$
$$s.t. \quad sync.conditions$$
$$s.t. \quad \sum_{d \in D} \sum_{s \in S} duration_s^{d} \cdot x_s^{d} \geq \boldsymbol{z}^*$$

Optimal solutions are injected as initial solutions for the next step

Results

The tested scenario containing:

- Seven flight days, 3 July 2023 to 9 July 2023,
- 98 to 103 aircraft, distributed along 16 subfleet
- 4463 mandatory flights + 448 optional flights, with a retiming of ± 60 minutes

The original optimization model has:

- 123,485 variables (almost all are binary) and 49,657 constraints
- 1% relative optimality gap

The execution times of the four above-mentioned steps are:

Step 0: daily optimization 84 seconds → **Step 0’: sync feasibility** 114 seconds → **Step 1: max flight time** 462 seconds → **Step 2: min retiming** 25 seconds

KPIs of optimized network:

Day	original flights	new flights	original flight time	optimized flight time	retiming after step 1	retiming after step 2
07/03	647	28 +4.33%	1262h	1301h +3.10%	191h	14h -92.95%
07/04	635	36 +5.67%	1240h	1290h +4.03%	184h	13h -93.09%
07/05	636	38 +5.97%	1242h	1293h +4.11%	190h	11h -94.13%
07/06	641	38 +5.93%	1240h	1292 +4.19%	197h	13h -93.26%
07/07	638	38 +5.96%	1227h	1278h +4.16%	205h	14h -93.29%
07/08	623	38 +6.10%	1260h	1310h +3.25%	196h	11h -94.42%
07/09	643	42 +6.53%	1244h	1298h +4.34%	186h	11h -94.22%
total	4463	258 +5.78%	8715h	9062h +3.98%	1350h	86h -93.62%

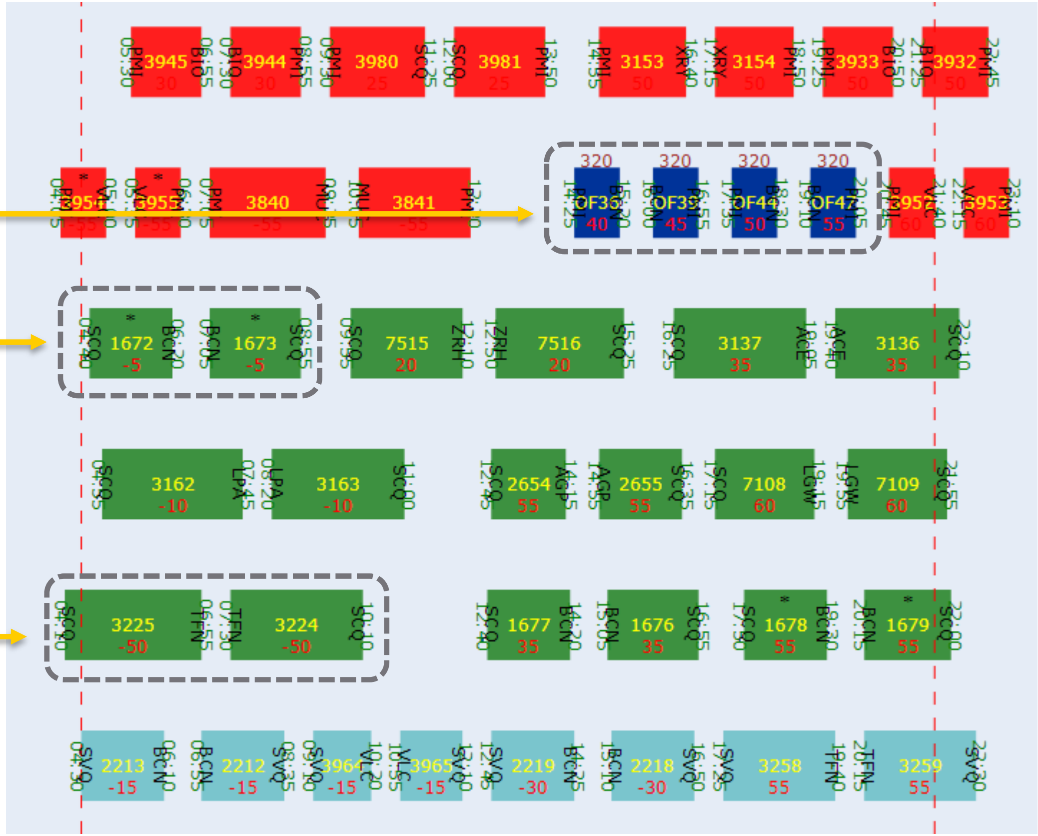
Optimized network schedule:

- Ongoing development

New optional flights, originally at base BCN320

Synchronized flights

Retimed -50 minutes



Conclusions

Current status:

- We have developed a mathematical formulation to fulfil the network with new flights so the most of operational conditions are met
- We provide a tool to automatize this process with Vueling specifications

This project is under development, so it is constantly evolving:

- Vueling is extending the scope of this project to correct original networks regarding unmet conditions besides optimizing them
- Other synchronization approach is in construction as we want to synchronize along months
- A formulation that includes maintenance tasks and buffers conditions is being developed without increasing so much the size of optimization models

THANKS FOR YOUR KIND ATTENTION

Daniel Ramón-Lumbierres: daniel.ramon@col.vueling.com
Victoria Rebillas Loredo: victoria.rebillas@col.vueling.com
Mohammad Kouhi: mohammad.kouhi@col.vueling.com

vueling

WE
LOVE
PLACES