University of Padova
Department of Statistical Sciences

Master Degree in
Statistical Sciences

Stochastic optimization of airport delays controlled by a dynamic programming algorithm: an application to Marco Polo airport in Venice

Supervisor: Professor Mauro Bernardi
University of Padova - Department of Statistical Sciences

Co Supervisor: Professor José Niño Mora
University Carlos III of Madrid - Department of Statistical Sciences

Student: Valentina Zambon
Serial number: 1155252

Academic Year 2017/2018

To my family, for its priceless support, moral help and suggestions.

To a big friend, Andrea, always reachable and with the right words wished to be heard.

To all those beautiful people met while in Madrid, my best flatmates Bianca, Irene and Conrado, my best friends Manuel, Gülsum, Beatrice, Giovanna and Vanessa.

Acknowledgments

First of all I would like to thank my supervisor Professor Mauro Bernardi for his instructions, support and dedication to the work. Thanks to his creativity and passion for his profession I've expanded my knowledge and learned how to be critical in a constructive way.

I would also like to thank Professor José Niño Mora with whom I had the pleasure of collaborating during my period of study and thesis elaboration in Madrid. In quality of co supervisor he taught me new study and research methods for which I found the passion. Thanks to his professionalism and devotion I've learned how to approach in new ways to real problems.

Many thanks to ENAV S.p.A. and SAVE S.p.A. for their collaboration in forwarding the data we used for our purpose. In particular I would like to thank Antonio Ulderico Prete, the current vicar of ENAV Group, for his time and kindness during the collaboration. I take this opportunity to inform that the information relating to the data used in this document forwarded by ENAV S.p.A. has been provided "as is", without any express or implied warranty with regards to the completeness and/or accuracy of the data in question and their suitability for any direct or indirect damage, however deriving from the users or third parties in relation to the processing of the information in this document.

Thanks to Paolo Scopelliti and Vincenzo Agosto for their permission to access university servers and for their IT support.

And finally thanks to my family for the patience, the help and for believing in my dreams. Especially thanks to my dad for sending me his big passion for aircraft and for flying.

Contents

List of Figures

List of Tables

Introduction

Airport delays are at anytime in the history a current problem from both the travelers' and the airport system's sight. For passengers delays are annoying and can cause troubles to their plans, for the airport organization including the airline companies and the air traffic controllers instead imbalances between scheduled flight plans and actual operations are behind the delays and very often translate into higher managerial costs. Many authors and researchers focused on this challenging issue and proposed different valid models, deterministic or stochastic, and feasible solutions.

What we propose is a stochastic optimization algorithm with the attempt to model the airport congestion system and work out a solution to minimize delays and reduce associated costs. Taking inspiration from a past research developed by Jacquillat and Odoni (2015 and 2017) we built a Finite–Horizon Dynamic Programming algorithm able to return the optimal combination of rates between arrivals and departures at which serve aircraft asking for landing and take–off respectively. The airport configuration is indeed seen as a queuing system where customers are the aircraft and the server is the runway. Queues are modeled as k–order Erlang distributions and probabilities of state transitions are defined. The elaborate details how the algorithm works and which parameters it requires to be run. The estimates of them and the validation of the model have been made possible through the application of the research to a real dataset forwarded by ENAV S.p.A, the National Flight Assistance Institution. The work has indeed acquired concreteness through its application to the case of Marco Polo airport in Venice.

One of the strong points of the model is taking into consideration also the weather influence on the choice of the best service rate to adopt. Up to our knowledge the previous literature on air traffic management does not directly consider the weather variability exerted on service rates by building a stochastic model accounting for the dynamic evolution of the variables influencing the possible fu-

ture changes of the weather state. Actually meteorological factors with no doubts condition traffic operations by constraining the airport capacity, the aircraft efficiency and the airlines and systems' on time performance. On the one hand we thought about the underlying trade–off between the number of arrivals and departures served in a same period of time and tried to model this type of relationship by means of a non parametric shape–constrained B–spline. On the other hand we proposed an advanced Hidden Markov model that extracts a latent factor representing the weather variable from the observed data. Thinking about that variable as a combination of visibility conditions and wind strength and direction, supposing for the former a Gaussian model and for the latter a Binomial distribution, we considered a Gaussian–Binary model for the weather variable. With the aim to succeed in extrapolating information about which state the airport system is in a given time period we completed the model by means of a Markov chain that drives the hidden states at each time point. For the parameters' estimation we made use of the Expectation–Maximization (EM) algorithm (McLachlan and Krishnan 2007) introducing two missing data structures, one for the unobservable Markovian states and the other relying to the Pólya–Gamma distribution (Polson, Scott, and Windle 2013) introduced for the representation of the logistic model about the parameter of the aforementioned Binomial distribution. More precisely we will talk about Conditional EM algorithm as, splitting a day of operation into periods of equal time–length, we formalized one transition probability matrix of the Markov chain per each time period, conditioning the probability to be in one of the possible weather states by the time.

To add credit to our study research, after applying the models to data we compared our results with a hypothetical solution and found out that our optimal proposal produces a gain in costs up to the 80% in some cases. We finally advanced a comparison with what is supposed to be happened in reality by means of the resulting queue lengths, directly linked to congestion costs.

Chapter 1

Operational Research and Statistics

1.1 The airport system and its management

In the air transportation industry flights are usually planned several months in advance, with airlines companies requiring and conquering the slots in the origin and destination airports. A *slot* is the permission that an aircraft is given to operate its arrival or departure using the runway and the necessary infrastructure at the coordinated airport. The Network Manager (NM) is the central figure of the European Air Traffic Management (ATM) and he looks after the alignment of air traffic demand with available slots and airport capacity (Ivanov et al. 2017).

Air traffic system plans and regulations on which airports base their operations are different between European and non–European contexts. Indeed thinking about Europe, EUROCONTROL is the intergovernmental Supporting European Aviation organization in charge of building a Single European Sky (SES) that will deliver the ATM performance. Its role's aim is to help the Member States which form part of EUROCONTROL's operational area run safe, cost–efficient and effective air traffic operations throughout the European region (https://www.eurocontrol.int/).

EUROCONTROL has been nominated in July 2011 by the European Commission as the Network Manager, with a mandate that will last until the end of the

year 2019. The NM's mission is to contribute to address the ATM's performance in the European network in the safety, environment efficacy, cost–efficiency and capacity areas. The Network Manager covers the whole of Europe, from Ireland to Armenia and from Morocco to Finland. It handles millions of flights a year accounting summer peaks resulting inevitably in the creation of bottlenecks, in the airspace or at some airports during a day of operations. Any disruption, namely a runway out of action, bad weather conditions or technical failures for example, can translate into management difficulties. EUROCONTROL indeed receives flight plans for all the commercial flights in the area it covers and the declared capacity limits for air traffic control centers and airports across the continent. Then if a system disruption occurs, the organization may reduce the rate at which aircraft can land. This operation is called a regulation.

Contextualizing an airport into the specific Operational Research field, it is compared to a *queuing system*. The service is provided by the runway(s) used to serve arrivals and take–offs (referring to movements in general), while the entities that require for it are the aircraft. These last join the queue when they demand for the usage of the runway system to land or to depart. Airports contemplate a service based on a First–Come–First–Served (FCFS) policy, thus equally considering every single aircraft and airline. On the other side the demand is managed by giving the aircraft a slot stating when it can take off, and the corresponding time is indicated as `CTOT` (Calculated Take–Off Time). Normally, the aircraft should take off within 15 minutes of the time stated in its flight plan[1] otherwise it loses its chance to use that slot with the consequent need to reapply for another one.

From the on time performance sight, the gap between scheduled and actual times represents the delay. Most of times delays occur in response to noticeable imbalances between demand scheduling and airport capacity and these imbalances pour on the total costs. For example in the United States the total cost was es-

[1] Normally this is the rule but if a slot is necessary then the 15–time minutes window is shrunk to within 5 minutes before the `CTOT` or within 10 minutes after the `CTOT`.

timated at over $30 billion in 2007 (Jacquillat and Odoni 2015). The question of alleviating airport congestion has become in the years a real challenge in air traffic management. It can be faced for example through interventions like infrastructure expansions thinking about the construction of new extra runways. This intervention however other than requesting a high money investment needs also the material space and specific conditions to be realized. Because of that it's a hardly feasible solution to the problem. Other ways could be improving air traffic controls and technologies or even adopting different demand management measures like congestion pricing or slot control policies.

1.2 Literature review

Existing approaches even if concerning all about the disruptions in flight schedules and aiming all to the related delays' minimization differentiate for the objective of the models and for where they want it to assist the problem. As Baspinar et al. (2016) point out, cause of the rapid growth of the air transportation industry and the continuous increase in the number of flights, the ATM system must see an operational transformation to face the challenge. Airports are the most fragile components of the air traffic network system as the most influential events to the traffic flow happen there. Hence it's quite common and proper to focus on the airports on model construction (Baspinar, Koyuncu, and Inalhan 2017). Recalling some previous works and researches on airport congestion systems, there are many and following different approaches as anticipated. For example Bard, Yu, and Argüello (2001) in their work build a time–band optimization model with the aim of minimizing costs associated with aircraft schedule disruptions by reassigning aircraft to flights. They take into account the system constraints like the available resources or the curfews and represent the airport system as a two dimensional time–space network made of station–time and station–sink nodes and flight and termination arcs. Pyrgiotis, Malone, and Odoni as well as computing

local delays at airports also study how they propagate over the network and elaborate a dynamic ad stochastic queuing model, named the Approximate Network Delays (AND) model (for a detailed description see Pyrgiotis, Malone, and Odoni 2011). Many authors attempted in different ways to mitigate the airport congestion for example improving slot adherence (see Ivanov et al. 2017), or changing slot allocation with deterministic models (see Baspinar, Koyuncu, and Inalhan 2017), or again intervening in the original flight schedules, allowing flights to be delayed, swapped or canceled if necessary (see Arias et al. 2013). Other works and models have been developed to solve this problem regarding the Air Traffic Flow Management (ATFM) and the ATM. Some examples are the Air Traffic Network Flow Optimization (ATNFO) and the Multi–objective Air Traffic Network Flow Optimization (MATNFO) models whose aims are alleviating airspace congestion and reducing flight delays globally and simultaneously respectively. There exist various ATFM actions to rebuild aircraft flight routes and reaching then the optimization aim, and these are ground–holding, airborne–holding, rerouting and speed control. An alternative model considering two effective ATFM actions, the airborne–holding and the speed control, comes from Cai et al. 2017. Their model is an improvement of the MATNFO model but additionally including these two possible actions. The approach takes inspiration from a large–scale, non–linear, discrete and multi–objective model called Multi–objective Evolutionary Algorithm (MOEA) and results in the Route and Time–slot Assignment (RTA) algorithm.

A different and innovating approach among all has been brought by Jacquillat and Odoni that in 2014 proposed and published a new approach involving the representation and estimation of the airport capacity itself, being directly determining the discussed imbalances and delays. Their model develops an original approach towards the airport congestion representation. It integrates a subtle characterization and tactical modeling of capacity utilization into a strategic model of airport congestion, combining a control of arrival and departure service rates into a stochastic and dynamic queuing process. In the real situation of airports air traffic managers

exercise a control over the runway configuration in use and the balance of arrivals and departures to make the best use of available capacity over the course of the day. Hence the attempt of the model is to reflect the task of air traffic managers, making gaining high value to the work. Additionally airport capacity depends on many operational factors including weather and wind conditions, the service rate at which landings and take–offs are performed and the runway configuration in use. Given that these factors are varying with the time and cannot be exactly known before they are really observed, including the stochasticity element in airport congestion models proves to be relevant. Deterministic methods even if they might be faster and more efficient from the computational point of view are nevertheless limited, then the advantage of introducing stochasticity to the models is about the ability to reflect the inherent stochasticity in air traffic networks (Baspinar, Koyuncu, and Inalhan 2017).

1.3 Model categories and definitions

Models of airport congestion can be classified into three main categories: microscopic, mesoscopic and macroscopic models (Jacquillat and Odoni 2015).

Microscopic models consider individual behaviors of every single aircraft and focus precisely on the sequencing of movements that determine the operational day at an airport. Due to their precision and peculiar analysis, microscopic models provide a high level of detail but require a large amount of data to be run.

Mesoscopic models predict taxi times and runway delays at airports using real available data, such as the runway configuration in use, the arriving and departing flight schedules, etc. Their contribute might be useful but their requirement of real operational data represents a limit in their applicability for strategic planning aims when data are not available.

Macroscopic models instead work on an aggregate level. These models develop efficient estimation procedures for the relationship between scheduled and actual

Rank	Airport	Location
1.	Hartsfield-Jackson Atlanta International Airport	Atlanta, Georgia, United States
2.	O'Hare International Airport	Chicago, Illinois, United States
3.	Los Angeles International Airport	Los Angeles, California, United States
4.	Dallas/Fort Worth International Airport	Coppell, Euless, Grapevine, and Irving, Texas, United States
5.	Beijing Capital International Airport	Chaoyang-Shunyi, Beijing, China
6.	Denver International Airport	Denver, Colorado, United States
7.	Charlotte Douglas International Airport	Charlotte, North Carolina, United States
8.	Amsterdam Airport Schiphol	Haarlemmermeer, North Holland, Netherlands
9.	Shanghai Pudong International Airport	Pudong, Shanghai, China
10.	Charles de Gaulle Airport	Paris, France
11.	Toronto Pearson International Airport	Mississauga, Ontario, Canada

Table 1.1: *Classification of the busiest airports by aircraft movements with reference to the year 2017 – Airports Council International.*

flight times and airport capacity, thus offering support to the strategic planning.

Among many different approaches applied to airport systems and real situations, we chose to follow the one whose procedure and appliance would have lead to a chance in finding an optimal solution in the strategical airport operational planning. The model we propose can thus be classified into the category of the macroscopic ones. The idea of studying a model which might be as much as possible adaptable to the real situation is not for its own sake. Its validation through the appliance to real circumstances at Marco Polo airport has made concrete the optimization theory and may contribute to develop a better strategic plan for the airport management. Already in several past researches, models from less complex to more elaborated have been applied to real data but the majority of these applications is with regards to the US airports. The reason is linked to the fact that they have been the first busiest in the world by aircraft movements for at least the last 12 years, as shown in Table 1.1 referring to the year 2017. The factors that may cause disruptions in the flight schedules are many: weather phenomena, cascade propagation delays, hindrances to on time performance from passengers and also technical problems, low efficiency of pilots and controllers in their functions, ground operations and the organization of the service system.

According to the circumstances and the reasons causing a delayed flight, the

Airport name	Country	Sub Region	Star Rating
Milan Malpensa Apt	Italy	Western Europe	3 stars
Catania	Italy	Western Europe	1 star
Pisa	Italy	Western Europe	2 stars
Rome Ciampino Apt	Italy	Western Europe	2 stars
Rome Fiumicino Apt	Italy	Western Europe	2 stars
Milan Linate Apt	Italy	Western Europe	4 stars
Florence Peretola Apt	Italy	Western Europe	3 stars
Palermo	Italy	Western Europe	4 stars
Venice Marco Polo Apt	Italy	Western Europe	3 stars
Bologna Guglielmo Marconi	Italy	Western Europe	2 stars
Milan Bergamo/orio al Serio Apt	Italy	Western Europe	4 stars
Naples Capodichino Apt	Italy	Western Europe	3 stars

Table 1.2: *Review on the on time performance for Italian airports with reference to the year 2017. One star is the minimum score, five starts the maximum. Results are provided from the Star Ratings Programme – OAG.*

delay can be of two types (Pyrgiotis, Malone, and Odoni 2011): upstream or local. The upstream delay is the one propagated and dependent on what has happened before the delayed flight, while the local one refers to the delay incurred at each individual airport during a visit by an aircraft using that runway system. As controlling upstream delays is hard, not to say impossible, once they depend on other factors than those directly linked to the delayed flight, the best way of control focuses on the local delays. Furthermore the decision to center particularly on the local tardiness is even more reasonable if we want to improve the on time performance of flights with regards to a specific airfield. Thus as our analysis is especially applied to Marco Polo airport in Venice the intention here finds justification. According to the On–Time Performance Stars Ratings Programme issued by

the OAG (Official Airline Guide) organization the major Italian airports occupy in 2017 a middle position. Table 1.2 shows the assigned ratings to them and Marco Polo received 3 stars, in a range from 1 to 5. The Programme works on an ongoing accreditation of airlines and airports awarding a star rating twice a year based on 12 months' rolling performance. Airports with the best profile receive five stars (https://www.oag.com/) while the poorest performers just one.

Chapter 2

Stochastic optimization model of airport congestion

2.1 Theory formulation

Taking inspiration from the study developed by Jacquillat and Odoni (2015 and 2017), whose original approach was applied to airport congestion modeling, the purpose of the thesis is the optimization of airport efficiency in terms of costs and on time performance. The idea was to imitate the air traffic managers' work to continuously decide the best service rate during each day of operations at airports. With *service rate*, distinguished between *arrival* and *departure*, we mean the number of aircraft given the allowance to land or to take off respectively. As the airport has its own capacity first of all resulting from the disposable and available to use runway(s), there is a maximum limit of aircraft which can take part in the service and there exists a trade–off between those requiring to land and those asking for departure. And it is exactly from here that the need arises to wisely allocate departing and arriving flights in time in order to avoid the formation of long queues and the increase of congestion costs. Indeed there are some costs associated to the

queue lengths and usually the ones stemming from the arrivals are higher than those stemming from the departures. Closely related to the queues, flight delays are quantified as the difference between the actual and the scheduled time for the operations, being it either the gate–in/gate–out or the take–off/landing time. Flight delays affect the on time performance of the airport.

We already introduced the airport system into the queuing theory prospective. Provided that, the attention must be paid now to the two main processes that take place in the airfield: the *landing* and the *take–off*. Arrival and departure queues can be considered as two distinct but not independent $M(t)/E_k(t)/1$ queuing systems, where the demand and the service are respectively modeled as Poisson and Erlang of order k processes. The value of k can be chosen according to a reasonable criteria, with the consciousness that the smaller the value is set the more variable the service process is assumed to be.

For each of the considered systems, both their demand and service processes are random time–varying and even if they are not independent their stochastic evolution instead is. On the one hand the arrival and departure queuing systems are dependent considering that they share the same weather conditions to which they are subjected to in a same period. Moreover they should result in being negatively correlated: in general an increasing departure throughput is associated with a decreasing arrival throughput, and conversely. On the other hand the evolutions of the two queues, stochastic with respect to the dependence on out–of–control and no deterministic factors, are to be considered independent between each others. Indeed for example on a given day and for given values of service rates, the arrival queue length might be shorter than expected while the departure one longer (Jacquillat and Odoni 2015).

The Erlang distribution assumed for the dependent service processes is a wise choice. Typically in queuing systems formulations the favorite combination is Poisson plus Exponential (for further details refer to Gross and Harris 1999), but the choice of an Erlang distribution brings its advantages. Providing more flexibility

in modeling than the Exponential family that only has one parameter is the determining one.

If the aim was to quantify and study the evolution of queue lengths in a day of operations, this model would have been a good proposal. But being the goal related to the choice of the best service policy that minimizes the congestion at the airport, it was allowed to use real data information at least for the demand process. As a matter of fact at the beginning of the day air traffic managers are given the plan of the arriving and departing aircraft. Hence instead of complicating the model introducing a stochastic demand, like the summoned Poisson process, we decided to use the available information about the scheduled flight plans for a selected day. The flight plan however can reasonably change during the day, just think about late passengers or bad weather conditions that force departures to be postponed or delays occurring from late arriving aircraft that change the initial schedules. Accordingly omitting the stochastic element would lead to miss some information and most of likely lose accuracy in the model. So if on the one hand we used the deterministic input for the flight schedules, on the other hand we accounted for the stochasticity in an fully–developed way.

The innovative framework that our model brings is the integration of an advanced Markov model for the weather state forecasting to existing approaches. Up to our knowledge the previous literature on air traffic management does not directly consider the weather variability exerted on service rates by building a stochastic model accounting for the dynamic evolution of the variables influencing the possible future changes of the weather state. Actually meteorological factors with no doubts condition traffic operations by constraining the airport capacity, the aircraft efficiency and the on time performance of airlines and systems. Starting with the application of an *ad hoc* model for our case study, we further develop the aforementioned forecasting model for the weather variable.

2.1.1 Finite–Horizon Dynamic Programming algorithm

The only way to drain the queues is through the given allowance to land for arriving aircraft and to take off for departing ones. To meet our goal, it was necessary to study then a way to control the service rate dynamically in a day. Following Jacquillat and Odoni's idea we adapted a Finite–Horizon Dynamic Programming algorithm, where the objective of the control is to minimize congestion costs. In their first approach, Jacquillat, Odoni, and Webster formulated this control in its full outline as a function of a five–dimensional state variable including weather and wind conditions, the runway configuration in use and the arrival and departure queue lengths. The complexity and high computational effort required to solve the problem, with the related inefficiency in terms of supporting strategic planning, however made the authors decide upon a simplified version for the control, reducing the number of influencing variables from five to three. In this way, if initially the idea was to optimize the organization of air traffic operations choosing the best combination between runway configuration to use and arrival and departing service rates to adopt, finally the control was restricted to just selecting the latter under capacity constraints depending on the weather conditions.

What we propose is an algorithm that has the potential to dynamically select the best combination of arrival and departure service rates for one day of operations. The selection is optimal as it tests all the possible alternatives and chooses the one that from that moment in time onwards is the best among all. The choice is just limited to the service rates and not to the runway configuration as the Venice airport only has one.

In traditional approaches the selection of the runway configuration to use and the service trade–off to select between landings and departures is related to the proportion of the corresponding movements that are scheduled in advance. For example, if more arrivals than departures are scheduled, then traditional approaches might suggest the use of a runway configuration which gives priority to arrival over

departures during that period. Dynamic approach plays the role to "adjust" this priority according to the observed arrival and departing queues. In other words, a dynamic approach might yield operational benefits in the face of queue uncertainty.

To let the decision policy be functional and bearing in mind the dynamic approach, we considered one day as a sequence of time periods each one of R–minute of length. For each of these intervals then the algorithm chooses the ideal service rate which is supposed to be constant over the R minutes of duration of the period. The input elements, summarized with regards to each interval, leading to the decision of the rate are the number of flights scheduled to land, the corresponding scheduled to depart, the weather and wind conditions and any delayed aircraft left to be served from the previous time period. Of course, depending on the criteria we based the split of the day, different step–to–step results were going to be obtained, but in an overall vision the final result would have been the same, the optimal one. The first element to be defined is the **state variable** at the starting period t. It clarifies the situation where the airport is supposed to be at the corresponding index time. Denoted with s_t it is a three–dimensional variable identified by the arrival a and departure d queue lengths at the end of the previous time interval and the weather state w at the beginning of the current period t. The mathematical expression is

$$s_t = (a_{t-1}, d_{t-1}, w_t). \tag{2.1}$$

The **time periods** t are the intervals into which one single day is divided. They can assume values from $1, \ldots, T$ and each one is of length R minutes.

The objective functions are the **one–period costs** associated with every single time period and are to be minimized. They are modeled as depending quadratically on the queue lengths as shown in the expression

$$\mathbf{c}_t^{\mu^a} = \rho\, a_{t-1}^2 + d_{t-1}^2. \tag{2.2}$$

The parameter ρ plays the role of weighing the influence of the arrival queue on the cost function. When ρ greater than 1, the associated cost is more affected by the arrival queue length than the correspondent departure.

Once defined the state variable and the cost function to be minimized the algorithm needs to choose the **control variables** which are the arrival and departure service rates, respectively denoted as μ^a and μ^d. Actually the variable that needs to be chosen is just one, the arrival service rate. For what concerns the correspondent departure service rate, it is deducted from the first one. Indeed, as previously mentioned, there exists a trade–off between these two variables that is first of all determined by the runway capacity. To better catch the relationship between the two variables we define a constrained function that links the two and call it *Operational Throughput Envelope* (OTE), just to distinguish it from the airport capacity limit. This function was first defined by Simaiakis (Simaiakis et al. 2014). The definition and estimation process of such function is detailed in subsection 2.3.

Imagining a temporal line scanning a day of operations starting from $t = 1$ and ending with $t = T$, we can define the so called **cost–to–go function** that finalizes the cost of being in state s_t at time t. We define one cost–to–go function per each interval of time and more precisely at the beginning of each time period we settle the minimum one as

$$\mathbf{v}_t^*(s_t) = \min_{0 \le \mu_t^a \le \mathbf{A}_{w_t}} \left\{ \mathbf{c}_t^{\mu^a}(s_t) \; + \; \mathbb{E}[\mathbf{v}_{t+1}^*(s_{t+1}) \mid s_t, \mu_t^a] \right\}. \tag{2.3}$$

The first term $\mathbf{c}_t^{\mu^a}(s_t)$ is the actual cost, i.e. the one arising from being in state s_t at the beginning of the present interval time t. From the definition of the state it can be derived how this actual cost depends on the queue lengths at the end of the previous period and on the current weather conditions. The apex μ^a of the term is to remind that its value depends on the previous choice of such control variable.

The second term in the formula $\mathbb{E}[\mathbf{v}_{t+1}^*(s_{t+1}) \mid s_t, \mu_t^a]$ is instead the future cost, i.e. that deriving from the usage of the chosen control variables μ_t^a and μ_t^d for the period t as service rates, given that the airport was in state s_t at the beginning

of the same. Because it is not known, it is accounted as an expectation. With a general reference it is thus defined as follows

$$\mathbb{E}[\mathbf{v}^*_{t+1}(s_{t+1}) \mid s_t, \mu^a_t] = \sum_{s_{t+1}} \Bigg\{ \Big[\mathbb{P}\left(s_{t+1}(a_t, d_t, w_{t+1}) \mid s_t \;=\; (a_{t-1}, d_{t-1}, w_t)\right), \mu^a_t \Big] \times \mathbf{v}^*_{t+1}(s_{t+1}(a_t, d_t, w_{t+1})) \Bigg\}. \tag{2.4}$$

The minimization is with regards to the choice of the service rate, whose values can range in the set named $\mathbf{A}_{w_t}$, theoretically depending on the weather.

For all apart the last cost–to–go function, it is immediate to realize that they can't be minimized because of their dependence from the unknown second additive element in formula 2.3 which is the cost related to the next time period. Therefore a way to solve the equations is proceeding with a backward approach, thus starting solving them from the last one in sequence to the first one. The last equation indeed does not contain any unknown values, as it is assumed not to be any existing period after the last T. In other words, at the end of interval T the day of operations is supposed to be concluded or at least no more congested. The functions written in sequence define the *system of Bellman equations*. We are now going to describe in detail the way it has been solved. Starting from the last equation

$$\mathbf{v}^*_T(s_T) = \min_{0 \le \mu^a_T \le \mathbf{A}_{w_T}} \left\{ \mathbf{c}^{\mu^a}_T(s_T) \;+\; \mathbb{E}[\mathbf{v}^*_{T+1}(s_{T+1}) \mid s_T, \mu^a_T] \right\}, \tag{2.5}$$

we need to solve a problem of optimization expressed in function of μ^a. Because the period $T+1$ is not defined (it's considered as not existing), the solution can be easily obtained by substituting the second additive term $\mathbb{E}[\mathbf{v}^*_{T+1}(s_{T+1}) \mid s_T, \mu^a_T]$ with the corresponding cost faced at the end of the period T after having taken all the decisions before, thus depending on all the previous states. In other words the problem of minimization becomes

$$\mathbf{v}^*_T(s_T) = \min_{0 \le \mu^a_T \le \mathbf{A}_{w_T}} \left\{ \mathbf{c}^{\mu^a}_T(s_T) \;+\; \mathbb{E}[\rho\, a^2_T + d^2_T] \right\}. \tag{2.6}$$

We point out that depending on the cost function we chose, the arrival and departure queue lengths related to the last time period are used just once and specifically

in the last equation, i.e. the first one in the solution process. If the cost was expressed in another form or through variables differently depending on the time index, the solution to the last cost–to–go function could have been differently obtained. Indicating with $\mathbf{v}_T^*(s_T)$ the optimal solution to the last cost–to–go function (the first to be solved), we are able to find the optimal solution for all the other functions in cascade, substituting from time to time the correspondent last optimal value in the second element of the expression 2.3. At the end of the estimation we expected to dispose of the optimal selection of the arrival and departure service rates, μ^a and μ^d respectively, for each of the time periods $t = 1, \ldots, T$ of a chosen day.

2.2 Inputs for the DP algorithm

The inputs required from the system of Bellman equations to be solved are the demand rates and the transition probabilities from a state to the successive one.

The demand rates as for the service rates are assumed to be constant over each R–minute period of time t. They are equal to the expected number of aircraft asking for landing if arrivals and taking off if departures for each period. We denote these rates respectively as λ_t^a and λ_t^d.

The transition probabilities are connected to the possibility to change state during time and they are conditioned to the current state. The state variable s_t is three–dimensional and defined as combination of the discrete arrival a_{t-1} and departure queue lengths d_{t-1} and the factorized weather state w_t. When thinking about the change of state we mean it as an evolution of the queues and/or the change of weather conditions. Specifically we assumed the evolution of one of these three phenomena to be independent of the others, thus we separately modeled and studied the systems, making the estimation process less tricky. As a matter of fact although the two movements are not independent, their evolution can be considered as such. Besides both are related to the weather conditions but defining an OTE

function per each possible weather state permitted to overcome this conditioning issue (subsection 2.3).

To better understand the dynamic evolution of the queues we refer to the schematic representation in Figure 2.1 showing the state transition diagram, which can be thought equivalently as a discrete–parameter Markov chain continuously varying in time. The explanation is the same whether we refer to the arrival or the departure queues. The circles indicate the all possible existing states and the

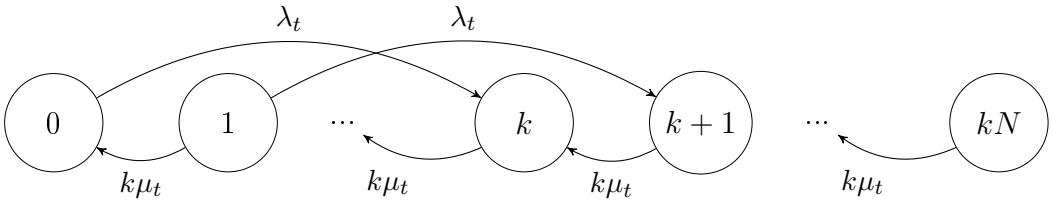


Figure 2.1: *State transition diagram of a generic $M(t)/E_k(t)/1$ queuing system.*

arrows connect them in line with the underlying Erlang distribution assumed. On the one hand the probability to move from a state to its next is regulated by the rate λ_t constant over time t. Note that the rate λ_t connects a state i to its next correspondent $i+k$. Indeed once an aircraft demands for the service, the request translates into the completion of k phases of work. On the other hand the probability to complete one single stage is given by the rate $k\mu_t$, where again μ_t is assumed to be constant over time t.

Stated that, we needed to estimate the state transition probabilities from a time period to the immediate successive one, taking into consideration the evolution of them over that interval of time. We use the notation $\mathbf{P}_i^t(r)$ to indicate the probabilities of being in state i at time r, where r varies from 0 to R, at the end of time period t given all the possible initial states where the system could be at the beginning of the same period. Hence we fixed a time period t, starting at time t and ending at t+1, and studied the evolution of the state probabilities between $(t-1)R$ and tR, where R is equal to the chosen width for the intervals. In other words even though we discretized the time splitting a day in periods, we realistically

studied the evolution of the system under continuous time. To do so we made use of the *Chapman–Kolmogorov system of first–order differential equations*. The system needed to be solved for each fixed value of μ_t (as this is the control variable in the Dynamic Programming algorithm) and for each possible initial condition of the system. The latter refers to the observed queue length at the beginning of the corresponding time period t. More precisely we assumed that no aircraft is in service when a period begins so the system is in state l where l denotes the number of queuing aircraft $\tilde{q}$ multiplied per the number of required stages for full service k. To this aim we needed to set a maximum queue length capacity, denoted with N. The selection had to be for a not too big neither a too small value. In the first case indeed the related high computational cost and time would have been an useless effort. In the second case instead a small value for N could have lead to underestimate delays. The optimal N value thus might be such as to approximate an infinite queue capacity.

2.2.1 Stages of work transition probabilities

The system of differential equations along with the initial conditions is described in detail below ((2.7)).

$$
\begin{aligned}
\frac{d\mathbf{P}_0(r)}{dr} &= -\lambda_t \mathbf{P}_0(r) + k\mu_t \mathbf{P}_1(r) \\
\frac{d\mathbf{P}_i(r)}{dr} &= -(\lambda_t + k\mu_t)\mathbf{P}_i(r) + k\mu_t \mathbf{P}_{i+1}(r) \qquad \forall i \in \{1, \dots, k\}, \\
\frac{d\mathbf{P}_i(r)}{dr} &= \lambda_t \mathbf{P}_{i-k}(r) - (\lambda_t + k\mu_t)\mathbf{P}_i(r) + k\mu_t \mathbf{P}_{i+1}(r) \\
&\qquad\qquad \forall i \in \{k+1, \dots, k(N-1)\}, \\
\frac{d\mathbf{P}_i(r)}{dr} &= \lambda_t \mathbf{P}_{i-k}(r) - k\mu_t \mathbf{P}_i(r) + k\mu_t \mathbf{P}_{i+1}(r), \\
&\qquad\qquad \forall i \in \{k(N-1)+1, \dots, kN-1\}, \\
\frac{d\mathbf{P}_{kN}(r)}{dr} &= \lambda_t \mathbf{P}_{k(N-1)}(r) - k\mu_t \mathbf{P}_{kN}(r),
\end{aligned} \tag{2.7}
$$

giving as initial conditions:

$$\mathbf{P}_i((t-1)R) = \begin{cases} 1 & \text{if } i = k\tilde{q} \quad \tilde{q} \in \{0, \dots, N\} \\ 0 & \text{otherwise,} \end{cases} \tag{2.8}$$

where $\tilde{q}$ is the number of queuing aircraft at the end of time period $t-1$, i.e. at the beginning of time period t. System (2.8) gives the required starting conditions to solve system (2.7). It assumes that at the beginning of each time period t there is no aircraft being served.

2.2.2 Queue length transition probabilities

The state transition probabilities matrices $\mathbf{P}_i^t$, for $i = 1, \dots, kN$ and $t = 1, \dots, T$ are a necessary information to be able to run the dynamic programming algorithm, but actually they consider the *stages of work* for the service process. Having identified a correspondence with the hypothesized phases and the real steps an aircraft performs when requiring for the service, as better shown in the next chapter, we should be able to observe the number of remaining stages of work but, to avoid complexity in the successive algorithm, we decided to estimate the queue length transition probabilities deriving them from the state transition ones. Actually, while observing the number of remaining stages of work for the departure process can be possible, for the arrival service it's not so easy. Calling the new probabilities $\mathbf{Q}_{l,n}^t$ they are the probability of having n aircraft in queue at the end of time period t, i.e. in tR, given that at the beginning of the same there were l. Namely,

$$\mathbf{Q}_{l,n}^t = \mathbb{P}\{q_{t-1} = n \mid q_t = l\} \quad l, n = 1, \dots N. \tag{2.9}$$

The advantage of dealing with the matrices $\mathbf{Q}^t = \left(\mathbf{Q}_{l,n}^t\right)_{l,n}, l, n = 1, \dots, N$ is also related to the reduced final number of states which now is $N+1$ while before it was $kN+1$.

The relationship between the matrices $\mathbf{P}^t$ and $\mathbf{Q}^t$ is detailed in formula 2.10.

$$\mathbf{Q}^t_{l,n} = \begin{cases} \mathbf{Q}^t_{l,0} = \mathbf{P}_0(tR) & \text{if } n = 0 \\ \sum_{l=1}^{k} \mathbf{P}_{(n-1)k+l}(tR) & \text{if } n = 1, \dots, N \end{cases} \tag{2.10}$$

2.2.3 Weather transition probabilities

For the last transition probabilities, those with regards to the weather and wind state, we created a new model able to estimate the probability of being in a weather state among the possible, by means of a latent structure extracted from the observed data. Section 4.2 is entirely dedicated to the explanation of the aforementioned Hidden Markov model.

Using the proper terminology adopted by the air traffic controllers we call *Visual Meteorological Conditions* (VMC) and *Instrumental Meteorological Conditions* (IMC) respectively good and bad weather to which correspond maximum airport capacity for the former and reduced airport capacity for the latter. Just to introduce the problem that is going to be further detailed later, we define

$$\mathbf{W}_t = \begin{pmatrix} p_t & 1 - p_t \\ 1 - q_t & q_t \end{pmatrix} \tag{2.11}$$

as the matrix for the weather state variable w_t, where the values in that mean the transition probabilities from a situation to the other according to

	VMC_{t+1}	IMC_{t+1}
VMC_t	p_t	$1 - p_t$
IMC_t	$1 - q_t$	q_t

As implied from the subscripts t, there is one matrix defined per each time period in a day and, as it will be further detailed in section 4.2, we will provide a weather forecasting model accounting also for an expected component of seasonality. Traditional approaches generally estimate matrix $\mathbf{W}$ as not depending on time or

seasonality.

Another way to define the probability parameters p_t and q_t is

$$p_t : \mathrm{VMC}_t \rightarrow \mathrm{VMC}_{t+1}$$

and

$$q_t : \mathrm{IMC}_t \rightarrow \mathrm{IMC}_{t+1}.$$

One of the strong points of the approach is taking into consideration the weather variability and developing a strategic management policy under different possible situations affected by stochasticity. As it will be cleared later in this thesis, with the weather state variable we refer to both visibility conditions and wind strength and direction. As a matter of fact the meteorologic factors determining the capacity at airports are multiple and although predictable always random elements. From that the necessity to adopt an advanced stochastic model.

2.3 Operational Throughput Envelope

Operational Throughput Envelope (OTE) is a key word in the analysis we follow. Its representation characterizes the non–increasing relationship between the average number of served arrivals (arrival throughput) and the respective average of served departures (departure throughput) under continuous demand for a given day and time period. The first variable is taken as the independent one, while the second is meant to be the dependent as determined by the arrival throughput.

The choice to consider the arrival throughput as the independent variable is justified by the fact that what concerns the arrivals is hardly controllable and depends on some exogenous out–of–control factors. Moreover the air traffic managers have more control power on the departing system, while for the arrivals it is limited as it depends on decisions taken previously in time and from other entities, namely the connected airports and their congestion, the weather conditions and other uncontrollable elements. Actually the departure throughput, the dependent

variable, might be explained and predicted as a function of many independent factors other than the arrival throughput, such as the fleet mix, the departure demand, operators efficiency and passengers delays. Existing approaches exploit that type of relationship, but without considering the constraints determining the physics of the system. Indeed the higher the number of landing flights the lower the corresponding number of departing flights should be, cause of the trade–off between the two variables. This means that the function should be monotonic and non–increasing. Additionally, higher values of landing aircraft constrain more the number of aircraft allowed to depart, thus meaning the function to be concave.

From here the necessity to proper define the relationship between the two central elements of our analysis arises. We indeed thought about a combination of B–spline basis to build a proper function suitable to express this linkage. We will explain in detail the estimation process in chapter 4.

Chapter 3

Venice Marco Polo airport dataset

This chapter is dedicated to the reality we chose to apply our model and test its power. After a description of the airport setting with a specific reference to the Marco Polo system in Venice, we will present the dataset, describe the data, detail what we used for the estimation of the parameters and the criteria driving such choices. In chapter 2 we gave space to the formulation of the problem and all the theory basis, introducing parameters that of course need to be estimated. From the collected data we couldn't dispose of all the required variables for how defined. Hence what we did was building new ones, introduced through this chapter.

3.1 Airport setting

Marco Polo airport in Venice classifies itself as the third Italian intercontinental hub with direct line connections to New York, Philadelphia, Atlanta, Dubai, Doha, Montreal, Toronto and, since April 2014, Tokyo. Its system including both Venice and Treviso airports is after Roma Fiumicino and Milano Malpensa the third biggest in Italy since 2012, counting in that year a total of more than 10,5 million of passengers. This position has then been maintained up to now, recording a continuous increase in the number of served passengers and flights (https:

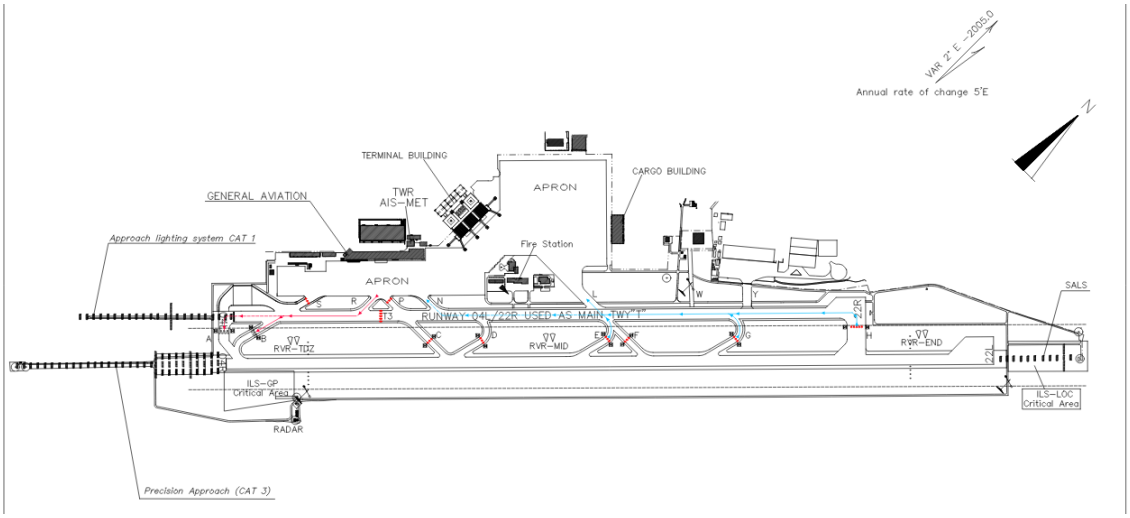


Figure 3.1: *Representative map of the Marco Polo airport configuration – ENAV*

//www.grupposave.it/).

Generally an airport is structured into internal buildings and an open space where aircraft are involved. Among the first apart from the platforms where actions like acceptance of passengers and bureaucratic operations are handled, there are the maintenance workshops, the fire station and the luggage storage properly called cargo building. As we are interested in the air traffic flow we focus more on detailing the open space area. The *terminal* is the structure where aircraft are engaged to let passengers reach the plane from the acceptance internal area. Each terminal can have more doors and an airport can have more than one terminal. The space where the aircraft is parked and waiting for the landing/boarding of passengers is called *gate*. From gates aircraft leave and running across the so called *taxiways* (TWY) they reach the assigned holding point, marked in Figure 3.1 (https://www.platinumairways.org/) with capital letters from A to H. *Holding points* are the stations where aircraft are ready to depart. From that indeed the next step is using the *runway(s)* (RWY) where take–offs and landings take place. In an airport there could be one or more runways. Venice has two but only one is used for its main purpose; the other shorter is kept as junction, scilicet just used

as main taxiway.

There exists two types of airports, major and minor. The former hosts scheduled national and international flights while the latter just carries out service information via radio. An example of minor is the civil airport in Padua. Marco Polo instead is a major airport and as such it has the *Control Tower* (TWR) whose task is to authorize the air traffic maneuvers. More specifically, all the air movements are controlled and managed from entities at the airport but while the ground ones from the gate to the holding point are handled by *Ground* (GND), the ones from the holding point to the take–off are handled by the Control Tower. These entities are part of the national entity for flight assistance, ENAV (National Flight Assistance Organization) (https://www.enav.it/sites/public/it/Home.html).

Runway system is generally the main bottleneck of operations at congested airports (de Neufville and Odoni 2013). In Marco Polo airport landings and take–offs can be managed relying on just one runway, namely the 04R/22L. The name used internationally for the runways depends on the distance in degrees from the North point on the compass to which the runway is oriented. As then the calculation can range from 0° to 360°, conventionally the orientation can be between 00° to 36°. To the runway name a capital letter is then added, R standing for right or L for left, depending on the side from where the counting of the degrees is started, always with reference to the North point. For instance, 04R means a runway orientated towards 40 degrees to North on the compass, looking at its right–hand side (toward East then). Depending on the winds, each runway can be used on both sides. For this reason one single runway is named with a double acronym. For the same previous instance, to 04R corresponds 22L, where 22 (220° from the North point) comes out from 04 (40°) + 18 (180°), that is the opposite orientation of the same runway. With regards to Marco Polo airport setting, the used runway is the 04R/22L, meaning the one in line between South–West and North–East (and correspondingly between North–East and South–West).

Previous approaches and researches have been applied for the most part to

American settings, as the most congested in the world. Unlike them (for example JFK, EWR and LGA studied by Jacquillat and Odoni (2015) and (2017)) for which the optimizing management rules could have counted also on different runway configurations usage, for the Venice case this choice was not feasible. Therefore the simplification made by the authors on the choice of the control's influencing variables (see 2.1.1) ends up to be as the reality. Indeed, reminding Jacquillat and Odoni's application, even if their model bases on playing with different usage of the runways and changing the service rates for arrivals and departures during the day of operations, at the end only the latter played the significant role, while the runway to be used was exogenously chosen in advance.

3.2 Data availability

To give reason and validity to the model the requirement was disposing of real data where to extract information. Most of the time data are very sensitive or likely in this specific context those required for the analysis could be not disposable or even not existing as no recorded. To our aim the most significant information was with regards to the times of the maneuvers and movements, both the scheduled and the actual ones. As a matter of fact the measure of the delay can be obtained through a comparison of the two. Thus a delay occurs when the difference between actual and expected time for an action is positive.

In order to comprehend the needed data for the real application, we recall the inputs required by the model, which are the schedules of landings and take–offs and the estimates of the airport capacity. As already mentioned the dynamic control advanced through the model is related to the determination of the best combination of arrival and departure service rates to be applied in each period aiming to maximize the airport efficiency and minimize the congestion costs. The specific sequence of decisions to be applied ends to be a function of the schedule of flights and the evolution of arrival and departure queue lengths.

To get the dataset we got in touch with the two Groups ENAV S.p.A. and SAVE S.p.A., the latter being the company managing the Venice airport (https://www.enav.it/sites/public/it/Home.html and https://www.grupposave.it/). From the first entity we got itemized data about scheduled and actual times of the movements performed at Marco Polo airport, with particular details regarding the departures. The second company instead could forward to us just summary data about scheduled and actual movements executed at the airport. The last set of data was however less informative to our purpose cause less detailing variables. For example, with reference to this one, while the actual movements are exactly in line with what was actually performed at the airport in the corresponding time period and day, the scheduled ones are those planned months in advance and subjected in the meantime (until the day of operations) to amendments, switches or even cancellations. As a consequence the reliability of the last piece of data could have been a little bit low for our aim. For this reason we preferred keeping them to create some essential and summarizing on time performance graphics or just in case we didn't have any better information.

What we need to make the model work is related to the times, scheduled and effective, of the following check points:

***i*) Off–Block Time**

`TOBT` *for the scheduled*, `AOBT` *for the actual.*

It's the time when the aircraft is put in motion, technically when its heel (the block piece) is removed or again when the aircraft leaves the gate to reach the holding point where ready to take off. We can refer to it also using the term Gate–Out time.

The Off–Block time is relative to departures.

***ii*) In–Block Time**

`TIBT` *for the scheduled*, `AIBT` *for the actual.*

It's the time when the aircraft is stopped and its engine is switched off,

technically when its heel is put on, or again when it reaches the gate after the landing. Another equivalent term for it is Gate–In Time.
The In–Block time is related to arrivals.

***iii*) Taxi–Out Time**

It's the needed time for the aircraft to reach the holding point from the gate, once its heels are removed. The holding point is the area where the aircraft is ready to depart, i.e. to take the runway to take off.
The acronym used for the Taxi–Out Time is `TOT`.

***iv*) Take–Off Time**

`TTOT` *for the scheduled*, `ATOT` *for the actual.*
It's the time when the aircraft detaches the wheels from the ground, i.e. takes off, or again it's the runway release time.
It can also be called *Wheels–Off Time.*

***v*) Landing Time**

`TLDT` *for the scheduled*, `ALDT` *for the actual.*
It's the time when the aircraft places the wheels on the ground, i.e. lands, or again it's the runway arrival time.
The Landing Time is also called Wheels–On Time.

To distinguish between scheduled, meaning the last estimated, and the effective time, the air traffic managers and controllers use respectively the words *target* (T) and *actual* (A). In reality, there are not only these two types of labels. Indeed, especially for the departures, more than the target and the actual times are recorded for some check points. We are referring indeed to the further attributes *scheduled* (S), which differently from the target they refer to the planned times set months in advance, and the *expected* (E) which instead are the ones set when building the flight plans, updated up to three hours before the time they indicate. Incidentally the Expected Off–Block time (`EOBT`) is particularly meaningful for

the A–CDM (Airport – Collaborative Decision Making) procedure incorporated in the European Air Traffic Flow and Capacity Management (ATFCM). It coordinates the turn–round and pre–departure sequencing processes and reveals the check points, so called milestones, starting three hours before the indicated `EOBT`. A–CDM is a way useful and efficient system, quite recently adopted by the Venice airport. Indeed after the signature of the MoU (Memorandum of Understanding) on September 17^{th}, 2012, the Marco Polo's operations management started benefiting from a full A–CDM procedure from January 20^{th}, 2015.

The split of the day in time periods was though to make the situation manageable. The split base is arbitrary, just it must be borne in mind that the width has not to be too big otherwise the model loses precision and efficiency. With regards to the American airports, the operations' rhythm and the magnitude of the performed movements usually drive researches to consider 15 minutes as time period's length. Instead for the Venice airport, that compared to the American airports registers a lower affluence, it has been decided to lengthen the time period from 15 to 30 minutes. In this way we avoided the rate ending to be too much low and were able to manage and efficiently control the service rate during a day of operations. As a result each day has been divided into 48 intervals of time, each of 30 minutes–length. Cause the operations at the airport are gathered between 5.30 a.m. and 12 p.m., as it will be proved in chapter 5, we decided to refer to $T = 37$ intervals, skipping the earliest rarely operative hours. The encoding used for time periods is wholly decrypted in Appendix B.

One of the strength points of the proposed model is its capacity to focus on the stochastic element regarding the weather state. Additional information required as inputs concern thus the winds and the visibility conditions per period. Winds typically constrain the runway/s usable at any time and with reference to the operations in Venice they rule the usage of the 04R/22L or the 22L/04R runway (the same one but towards the two opposite poles). The runway used as far as possible is the 04R/22L oriented, as it gives the maximum achievable service rate, which is

RVR	Service rate per hour	Landing Rate per hour
[1500, 800) mt	32 movements	19 arrivals
[800, 550) mt	18 movements	7–10 arrivals
[550, 400) mt	14 movements	6–8 arrivals
[400, 125) mt	12 movements	6–8 arrivals
$\leq$ 125 mt	6 movements	3–6 arrivals

Table 3.1: *Service rates between arrivals and departures allowed under different visibility conditions, i.e. RVR values*

32 movements, allowing a maximum of 19 arrivals, per hour. If the wind exceeds 10 knots of strength, then the runway to use has to be the 22L/04R. The latter allows a maximum of 22 movements per hour. From here it can be understood that the change of runway usage is only determined by the wind directions and strength and it doesn't depend on any strategic intervention. Hence again our attempt to optimize the airport efficiency and minimize congestion costs will be only through the best choice of the service rates for arrivals and departures per time period. Visibility conditions instead limit the efficiency of airport operations. Table 3.1 shows the constrained movements allowed. The *RVR* acronym stays for *Runway Visual Range* and is expressed as distance in meters. It is the range over which the pilot of an aircraft on the center line of a runway can recognize the central line, the delimiting lights or the surface markings of the runway.

By the ENAV air traffic management, the priority is given to the departures while with regards to the arrivals once aircraft land and free the runway their detailed movements and check times on ground are no more recorded from the Group operators' side. Hence for the estimation of arrival and departure queue lengths at the beginning of each time period we were able to collect only a part of the necessary data, which were those for the estimation of the departure queue lengths. In fact estimating the arrival queue length from the available records of

operations is really hard and almost impossible (Jacquillat and Odoni 2015) as there is no record of when arriving aircraft demand for the usage of the runway.

3.3 Variables: selection and creation

As anticipated, while the actual times are registered in correspondence to what has actually been performed at that instant during the day of operations, those times we refer to as scheduled could be named differently depending on when and from whom they were communicated. Hence the selection of the latter among more than one possibility is followed by a brief justification.
If not otherwise specified, the data used for the analysis are those provided by ENAV S.p.A.

***a*) Number of scheduled departures** per time period[1]

The estimation of this variable is made by means of the *Expected Off–Block Time* (`EOBT`), which is technically the time assigned and registered in the flight plan to put the aircraft in motion.

Even if it's not the time when the aircraft takes off, it is used to indicate the named variable because when the aircraft starts moving it represents an entity to be monitored on its movements until it takes off. We chose `EOBT` to obtain the estimates because it's the most precise, as the last modified in planning, scheduled time.

***b*) Number of scheduled arrivals** per time period

The estimation of this quantity would have properly been the corresponding *Expected In–Block Time* (`EIBT`) registered in the flight plan. Nevertheless from the ENAV data we weren't able to extract this type of information as from their side, as already announced, for the arrivals only the *actual* Landing and In–Block times are recorded.

[1] When referring to time periods, we mean the decided time period length of 30 minutes

Actually also the `EOBT` is registered but it is with regards to the estimated time the aircraft would commence the movement associated with its departure from the origin airport. One idea would have been to calculate the `EIBT` adding to the corresponding `EOBT` the taxi–out–times and the airborne time, which is that spent on air by the aircraft. While the latter can be obtained quite precisely knowing the distance from the origin to the destination airports and the aircraft type, the taxi times in both the airports are more difficult to calculate. Indeed they depend on the airport setting and on the gate where the aircraft is going to be stopped.

Avoiding introducing manual errors building the `EIBT`, we decided to use the best scheduled times we could have which are those provided by the SAVE Group. We felt free to use them for our estimates, but reminding that they refer to the *Scheduled In–Block Time* (`SIBT`), i.e. the time that the aircraft is scheduled to arrive at its first parking position, and they could have been subjected to amendments from when they were planned till the day of operations.

***c*) Departure throughput**

With this locution adopted from Simaiakis' terminology we refer to the number of aircraft that take off during a defined period of time.

It is indicated as DT and is expressed per each interval as

$$DT(t) = \sum_i | \, i \in F \text{ st} \quad tR \leq ATOT_i < (t+1)R \, | \, .$$

***d*) Arrival throughput**

This works as the same of the previous indicator but for the arrivals, hence it quantifies the number of aircraft that land per each considered time period.

It is indicated with AT and is expressed per each interval as

$$AT(t) = \sum_i | \, i \in F \text{ st} \quad tR \leq ALDT_i < (t+1)R \, | \, ,$$

where ALDT stands for *Actual Landing Time*.

Sometimes it can happen that records for some of the arrivals ALDT are missing and among them some of the *Actual In–Block Times* (AIBT) are missing as well. The reasons might be cancellation of the corresponding flights or not recorded information from the air traffic management operators. This behavior, if so, will be coherent with the low importance given to arrivals in comparison with landings. For these missing records, we just decided to ignore them.

***e*) Departure queue length** d_t at the end of each time period

This variable, like the corresponding arrival queue length, is particularly relevant as it is a proxy variable quantifying the congestion at the airport. They properly indicate the length of the relative queue at the end of the period t. The estimation process involves the already defined *Actual Off–Block Time* (AOBT) and *Actual Take–Off Time* (ATOT) plus the *Taxi–Out–Time* (TOT), as the formula below shows:

$$\widehat{d_t} = \sum_i \Big| \, i \in F \text{ st } \quad AOBT_i + TOT_i^P \leq (t+1)R \quad \& \quad ATOT_i > (t+1)R \, \Big| \, .$$

The apex P added to the TOT *stands for* ***predicted*** *and indeed refers to the expected calculated Taxi–Out Time.*

***f*) Arrival queue length** a_t at the end of each time period

The estimation of this other relevant variable is again really hard to obtain. Indeed there are no records about the exact time when an aircraft demands for the usage of the runway to land. As a consequence, we needed to find an alternative way to estimate it. If we had the EIBT that would have been a reasonable record where to extract information about the arrival queue length. As again we had no access to it, we used the ALDT as source where trying to extrapolate a for sure erratic but at least tolerable estimate. Hence we used accordingly the actual number of landings whose related times are

included in the range whose upper bound is tR plus 7 minutes (arbitrarily chosen) and the lower is equal to tR. The interval's upper bound choice comes from a rough approximation about the time when aircraft ask for the landing allowance. On the one hand, the inclusion of the actual Landing Time is to count the aircraft that ask for landing in a range of time close to the "boundaries" (00 and 30 minutes of each hour of the day). On the other hand it would have been more correct to include also the scheduled In–Block Time in the estimation of the variable. It would have been meaningful because we are supposed to take into account also those aircraft that are demanding for landing but as no service is available they are constrained to wait and queue in the airspace. Indeed in a situation of congestion, where the demand at the airport is higher than the available capacity, aircraft can be made queuing already before taking off, so queuing at the ground (if a Ground Delay Program (GDP) is applied) or in the airspace, inducting to higher related congestion costs though.

Due to the impossibility to extrapolate such information, we preferred being less accurate in the estimate but proposing a still valid one. A further attempt might have been adding a random value to the arrival queue estimate as defined here, in order to catch the waiting aircraft queued in the airspace. The next formula specifies mathematically the rough adopted estimate:

$$\widehat{a}_t = \sum_i | \, i \in F \text{ st } \quad tR \leq ALDT_i \leq (tR + 7) \, | \, .$$

g) Airport Capacity

Along with the scheduled flights, the other input that the strategic model requires is the estimate of the airport capacity. It is worth to dwell on one term largely used as relevant in this research field, namely the *Operational Throughput Envelope* (OTE).

Following the Simaiakis definition (see details in Simaiakis 2012) we use the

OTE term to indicate the airport capacity characterizing the combination of the arrival and the departure throughputs.

The OTE upholds the non–increasing relationship between the average number of landings and the average number of take–offs that can be served per each 30–minute period under continuous demand. It depends on many factors, mainly on the weather conditions, the proportions of landings and take–offs performed and the runway configuration in use (in our case the side from which the only runway is caught). To build the OTE we plotted the scheduled number of arrivals and the corresponding of departures per each time period and traced the broken line showing the maximum reachable trade–off between serviceable landings and take–offs under different weather conditions.

***h*) Weather state variable**

We explained the importance of considering the weather state **w** in our analysis. This dichotomous variable **w** is defined as a mixture of more variables, like wind force, wind direction and visibility conditions of the runway and of the distances related to the movements in the airport space. A summary index considering all the named elements does not exist and in the reality air traffic managers dispose of real time information about winds and weather and their decisions about service control are based on choices made at the moment.

What we managed to get was a detailed dataset registering every five minutes the visibility measures (RVR values). That was the only recorded variable determining the weather state. About the winds, we had no such historical data but we extrapolated information from the runway used. Indeed the wind strength and direction constrain the usage of the more efficient runway and in case of strong winds they even prevent the use of it forcing to serve aircraft with the 22L/04R one. Knowing then which runway has been used from aircraft to land or take off we could deduct additional significant infor-

mation to better define the weather variable. In this the dichotomous $\mathbf{w}_t$ is to be considered as a combination of the RVR value and the runway used, proxy for the wind strength.

Chapter 4

Statistical modeling

This chapter is devoted to the statistical approach towards the estimation of two elements of relevancy in the thesis. In the first section we focus on the process estimation of the already introduced OTE function (2.3) by means of a non parametric shape–constrained B–spline. The second section instead we propose an advanced forecasting Hidden Markov model that extracts a latent factor representing the weather variable from the observed data. In this chapter we will formalize the problem and provide the estimates of the models, while for their application to real data and the presentation of the results we refer to the next chapter, 5.

4.1 Model for Operational Throughput Envelope estimation

In chapter 2 we defined the OTE as the function describing the relationship between arrival and departure throughputs and we know that there is a trade–off between the two. As we did not have any further information about the behavior of the underlying function, we thought about splines.

A *spline* is a mathematical tool defined as a combination of piecewise linear

functions of a given order associated with a sequence $\boldsymbol{\tau}$ of knots. It's particularly useful in statistics when we need to approximate a function expressing the relationship between two variables for which we only know a set of observations (for further details see Hastie, Tibshirani, and Friedman 2001, Azzalini and Scarpa 2012 and Wood 2006). In our specific case, we used this class of functions to approximate the OTE. Precisely we sought to estimate such function under shape constraints of non–increasing monotonicity and concavity. In other words denoting the function by $\bar{f} : \mathbb{R} \to \mathbb{R}$ the estimate was for the mean $\overline{DT}(t) = \bar{f}(AT(t))$. Thus we made use of a non parametric shape–constrained spline, built as combination of a set of B–splines. According to de Boor (2001), given $x \in \mathbb{R}$ a B–spline basis function of order r defined on the set of knots $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{r+1})$ is given by

$$B_r(x|\boldsymbol{\tau}) = \frac{x - \tau_1}{\tau_r - \tau_1} B_{r-1}(x|\boldsymbol{\tau}_{-(r+1)}) + \frac{\tau_{r+1} - x}{\tau_{r+1} - \tau_2} B_{r-1}(x|\boldsymbol{\tau}_{-1}), \tag{4.1}$$

where $B_1(x|(a,b)) = I(a \leq x \leq b)$ and $\boldsymbol{\tau}_{-j}$ is the vector $\boldsymbol{\tau}$ without the j–th element. For how it is built, function (4.1) is always positive between τ_1 and τ_{r+1} and null outside, and it is unimodal for $r > 1$. For these reasons, it can be seen as an unnormalized density function with vector of parameters $\boldsymbol{\tau}$. Depending on the distances between the values of $\boldsymbol{\tau}$, (4.1) can be symmetric or not. These characteristics make the B–splines a suitable tool for modeling irregular or low–information content data, such as in our case. In this way the information from data has been used rather in an efficient way not to understand the shape of the function, but to estimate the trade–off between arrival throughput, the independent variable, and the respective departure, the dependent one. The shape is instead imposed by means of constraints. The $\bar{f}$ function has been approximated then as follows

$$u_i = \bar{f}(x_i) \approx \sum_{k=1}^{g+r} \vartheta_k B_k(x_i), \tag{4.2}$$

where B_k is the (non–negative) k^{th} B–spline of order r, $\vartheta_k \in \mathbb{R}$ its coefficient and g the number of breaks.

To give the desired shape to the function, as already stated, we needed to constrain

the coefficients ϑ_k through linear combinations of the same, as follows

$$\vartheta_k - \vartheta^{k-1} \leq 0, \quad k = 2, \ldots, K, \tag{4.3}$$

$$\vartheta_k - 2\vartheta^{k-1} + \vartheta^{k-2} \leq 0, \quad k = 3, \ldots, K. \tag{4.4}$$

The two conditions are equivalent to constraints on the derivatives of the $\bar{f}$ function. The first one for the non–increasing monotonicity indeed requires a non–positive first derivative, $\left(\bar{f}'(\mathbf{x}; \boldsymbol{\vartheta}) \leq 0\right)$, while the second one for the concavity a non–positive second derivative $\left(\bar{f}''(\mathbf{x}; \boldsymbol{\vartheta}) \leq 0\right)$. If we express them in matrix terms that means

$$D_1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}$$

as the first difference matrix, and

$$D_2 = \begin{bmatrix} 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \end{bmatrix}$$

as the second difference matrix. Then the constraints become

$$\bar{f}'(\mathbf{x}; \boldsymbol{\vartheta}) \leq 0 \text{ and } \bar{f}''(\mathbf{x}; \boldsymbol{\vartheta}) \leq 0 \quad \Rightarrow \quad \begin{bmatrix} -D_1^T & D_2^T \end{bmatrix}^T \boldsymbol{\vartheta} \geq 0, \tag{4.5}$$

for the sought monotonically non–increasing and concave function.

Given n pairs of observations of the type (x_i, u_i), $i = 1, \ldots, n$, we can formalize the problem as

$$\arg\min_{\vartheta} \frac{1}{n} \sum_{i=1}^{n} \left[u_i - \sum_{k=1}^{g+r} \vartheta_k B_k(x_i) \right]^2, \tag{4.6}$$

subjected to constraints (4.3) and (4.4), where $\mathbf{u} = (u_1, \ldots, u_n)$, $\mathbf{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_{g+r})$. As the reachable service rate depends on the weather

conditions, we divided the dataset into two, to distinguish the VMC from IMC observed cases. The estimation process was then done twice, one for each partial dataset.

The solution is obtainable with any quadratic programming solver, like the software R.

4.2 Hidden Markov model for weather forecasting

4.2.1 Introduction

In this section we first introduce the basic MSM framework and we extend it to account for the dynamic evolution of the weather variable state w_t defined in chapter 2. Then, we detail the estimation methodology of the models parameters. For an up to date review of MSMs see, e.g., Cappé, Moulines, and Rydén 2005, Zucchini and MacDonald 2009 and Dymarski 2011.

Let $\{\mathbf{Y}_t, t = 1, 2, \ldots, T\}$ denote a sequence of multivariate observations, where $\mathbf{Y}_t = \{Y_{1,t}, Y_{2,t}, \ldots, Y_{p,t}\} \in \mathbb{R}^p$, while $\left\{\bar{S}_t, t = 1, 2, \ldots, T\right\}$ is a Markov chain defined on the state space $\{1, 2, \ldots, L\}$. A MSM is a stochastic process consisting of two parts: the underlying unobserved process $\left\{\bar{S}_t\right\}$, fulfilling the Markov property, i.e.

$$\mathbb{P}\left(\bar{S}_t = \bar{s}_t \mid \bar{S}_{1:t-1} = \bar{s}_{1:t-1}\right) = \mathbb{P}\left(\bar{S}_t = \bar{s}_t \mid \bar{S}_{t-1} = \bar{s}_{t-1}\right),$$

where $\bar{S}_{1:t-1} = \left(\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_{t-1}\right)$ and $\bar{s}_{1:t-1} = (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_{t-1})$ and the state–dependent observation process $\{\mathbf{Y}_t\}$ for which the conditional independence property, i.e.

$$f\left(\mathbf{Y}_t = \mathbf{y}_t \mid \mathbf{Y}_{1:t-1} = \mathbf{y}_{1:t-1}, \bar{S}_{1:t} = \bar{s}_{1:t}\right) = f\left(\mathbf{Y}_t = \mathbf{y}_t \mid \bar{S}_t = \bar{s}_t, \mathbf{y}_{1:t-1}\right),$$

holds, where $f(\cdot)$ denotes a generic probability density function.

4.2.2 Model specification

Let $\mathbf{y}_1 = (y_{1,1}, y_{1,2}, \ldots, y_{1,T})$ be a random sample of T Gaussian observations, $\mathbf{y}_2 = (y_{2,1}, y_{2,2}, \ldots, y_{2,T})$ be a random sample of T binary observations, $\mathbf{h} = (h_1, \ldots, h_T)$ is a vector of exogenous regressors accounting for the expected seasonal pattern in the meteorological conditions, with

$$\mathbf{h}_t = \begin{cases} 1 & \text{if } t = 1 \\ 2 & \text{if } t = 2 \\ \quad \vdots & \\ 37 & \text{if } t = 37. \end{cases} \tag{4.7}$$

We consider the following Gaussian–binary model:

$$y_{1,t} \mid \bar{S}_t = \bar{s}_t \sim \mathsf{N}\left(\bar{\mu}_{\bar{s}_t} + \gamma_{\bar{s}_t} h_t, \sigma^2_{\bar{s}_t}\right) \tag{4.8}$$

$$y_{2,t} \mid \bar{S}_t = \bar{s}_t \sim \mathsf{Bin}\left(1, \psi_{\bar{s}_t}\right) \tag{4.9}$$

$$\psi_{\bar{s}_t} = \mathrm{F}_{\mathrm{Lo}}\left(\eta_{\bar{s}_t} + \alpha_{\bar{s}_t} h_t\right) = \frac{\mathrm{e}^{\eta_{\bar{s}_t} + \alpha_{\bar{s}_t} h_t}}{1 + \mathrm{e}^{\eta_{\bar{s}_t} + \alpha_{\bar{s}_t} h_t}}, \tag{4.10}$$

for $t = 1, 2, \ldots, T$, where $\bar{\mu}_{\bar{s}_t}$ is the constant and $\gamma_{\bar{s}_t} h_t$ the regression part of the mean for the Gaussian distribution, $\sigma^2_{\bar{s}_t}$ its variance, $\psi_{\bar{s}_t}$ is the probability parameter for the Binomial distribution, $\eta_{\bar{s}_t} + \alpha_{\bar{s}_t} h_t = \log \frac{\psi_{\bar{s}_t}}{1+\psi_{\bar{s}_t}}$ is the log–odds ratio with $\alpha_{\bar{s}_t}$ the parameter that controls the seasonal behavior of the transition matrix $\mathbf{W}_{h_t}, h_t = 1, \ldots, T$ and $\mathrm{F}_{\mathrm{Lo}}(\cdot)$ denotes the logistic–link function.

For the purpose of developing the inferential procedures in the next Section, we remind that the logistic model can be expressed as a scale mixture of Pólya–Gamma distribution, see Polson, Scott, and Windle 2013. Specifically, the fundamental integral identity at the heart of the Pólya–Gamma representation of the logistic model is that, for $b > 0$,

$$\frac{\left(\mathrm{e}^{\psi}\right)^a}{\left(1 + \mathrm{e}^{\psi}\right)^b} = 2^{-b} \mathrm{e}^{\psi\kappa} \int_0^{\infty} \mathrm{e}^{-\omega\psi^2} p\left(\omega\right) d\omega, \tag{4.11}$$

where $\kappa = a - \frac{b}{2}$ and $\omega \sim \mathsf{PG}(b, 0)$.

The model is completed by the specification of the Markov chain that drives the hidden states at each time point t. To this purpose let $q_{l,k} = \mathbb{P}\left(\bar{S}_t = k \mid \bar{S}_{t-1} = l\right)$, $\forall l, k \in \{1, 2, \ldots, L\}$ denote the probability that state k is visited at time t given that at time t–1 the chain was visiting state l. We indicate with $\delta_l = \mathbb{P}\left(\bar{S}_1 = l\right)$ the initial probability of being in state $l = \{1, 2, \ldots, L\}$ at time 1, the corresponding vector with $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_L)$ and we refer to $\boldsymbol{W} = \{q_{l,k}\}_{l,k=1,2,\ldots,L}$ as the transition probability matrix of the Markov chain.

4.2.3 Estimation and inference

The MSM parameters are generally estimated using the maximum–likelihood method, see, for example, McLachlan and Peel 2000 and Cappé, Moulines, and Rydén 2005. Let $\boldsymbol{\vartheta} = \left(\{\bar{\mu}_l, \sigma_l^2, \eta_l\}_{l=1}^{L}, \boldsymbol{W}, \boldsymbol{\delta}, \right)$ be the set of all model parameters and let $\boldsymbol{f}(\mathbf{y}_t)$ be a diagonal matrix with conditional probabilities $\boldsymbol{f}\left(\mathbf{Y}_t = \mathbf{y}_t \mid \bar{S}_t = \bar{s}_t, \mathbf{y}_{1:t-1}\right)$ on the main diagonal, then, the likelihood of a MSM can be written as

$$\mathcal{L}(\boldsymbol{\vartheta}) = \boldsymbol{\delta} \boldsymbol{f}(\mathbf{y}_1) \boldsymbol{W} \boldsymbol{f}(\mathbf{y}_2) \boldsymbol{W} \times \cdots \times \boldsymbol{f}(\mathbf{y}_{T-1}) \boldsymbol{W} \boldsymbol{f}(\mathbf{y}_T) \mathbf{1}'. \tag{4.12}$$

Finding the values of the parameters $\boldsymbol{\vartheta}$ that maximize the log–transformation of equation (4.12) under the constraints $\sum_{l=1}^{L} \delta_l = 1$ and $\sum_{k=1}^{L} q_{l,k} = 1$, is not an easy problem. Instead, it is straightforward to find solutions of equation (4.12) using the Expectation–Maximization (EM) algorithm of Dempster, Laird, and Rubin 1977. Hereafter, we focus on the EM algorithm which has been previously applied to the case of finite mixtures of univariate Student–t distributions by Peel and McLachlan 2000.

For the purpose of application of the EM algorithm the vector of observations $\mathbf{y}_t, t = 1, 2, \ldots, T$ is regarded as being incomplete. Following the implementation described in Peel and McLachlan 2000 in a finite mixture context, two missing data structures are consequently introduced. The first one is related to the unobservable Markovian states, i.e., $\mathbf{z}_t = (z_{t,1}, z_{t,2}, \ldots, z_{t,L})$ and $\mathbf{zz}_t =$

$(zz_{t,1,1}, zz_{t,1,2}, \ldots, zz_{t,l,k}, \ldots, zz_{t,L,L})$ defined as

$$z_{t,l} = \begin{cases} 1 & \text{if} \quad \bar{S}_t = l \\ 0 & \text{otherwise} \end{cases}$$

$$zz_{t,l,k} = \begin{cases} 1 & \text{if} \quad \bar{S}_{t-1} = l, \bar{S}_t = k \\ 0 & \text{otherwise.} \end{cases}$$

The second type of missing data structure is $\omega_t \sim PG(1,0)$, $\forall t = 1, 2, \ldots, T$ relies to the Pólya–Gamma representation of the logistic model in equation (4.11) which are assumed to be conditionally independent given the component labels $z_{t,l}, z_{t,l,k}, l, k = 1, 2, \ldots, L, \forall t = 1, 2, \ldots, T$.

Augmenting the observations $\{\mathbf{y}_t, t = 1, 2, \ldots, T\}$ with the latent variables

$$\{\omega_t, z_{t,l}, zz_{t,l,k}, t = 1, 2, \ldots, T; l, k = 1, 2, \ldots, L\}$$

gives the following complete–data log–likelihood:

$$\begin{aligned} \log \mathcal{L}_c(\boldsymbol{\vartheta}) \propto & \sum_{l=1}^{L} z_{1,l} \log(\delta_l) + \sum_{l=1}^{L}\sum_{k=1}^{L}\sum_{t=1}^{T} zz_{t,l,k} \log(q_{l,k}) \\ & - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} z_{t,l}\left(\log(2\pi) + \log\left(\sigma_l^2\right)\right) \\ & - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} \frac{z_{t,l}\left(y_{1,t} - \bar{\mu}_l - \gamma_l h_t\right)^2}{\sigma_l^2} \\ & + \sum_{l=1}^{L}\sum_{t=1}^{T} z_{t,l}\tilde{y}_{2,t}\left(\eta_l + \alpha_l h_t\right) - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} z_{t,l}\omega_t\left(\eta_l + \alpha_l h_t\right)^2, \end{aligned} \tag{4.13}$$

where $\tilde{y}_{2,t} = y_{2,t} - \frac{1}{2}$ and $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_t)$.

The EM algorithm consists of two major steps, one for expectation (E–step) and one for maximization (M–step), see McLachlan and Krishnan 2007. At the $(m+1)$–th iteration the EM algorithm proceeds as follows:

(i) **E–step:** computes the conditional expectation of the complete–data log–likelihood (4.13) given the observed data $\{\mathbf{y}_t\}_t^T$ and the m–th iteration pa-

rameters updates $\boldsymbol{\vartheta}^{(m)}$

$$\begin{aligned}\mathcal{Q}\left(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{(m)}\right) &= \mathbb{E}_{z,\omega}\left[\log \mathcal{L}_c\left(\boldsymbol{\vartheta}\right) \mid \{\mathbf{y}_t\}_{t=1}^T\right] \\ &= \mathbb{E}_{\omega|z}\mathbb{E}_z\left[\log \mathcal{L}_c\left(\boldsymbol{\vartheta}\right) \mid \{\mathbf{y}_t\}_{t=1}^T\right], \end{aligned} \tag{4.14}$$

where the expected value of the latent factor ω taken with respect to the conditional distribution of $\omega|z, y$, i.e., $\mathbb{E}_{\omega|z}\left(\omega \mid y, z\right)$, where $\omega \mid y, z \sim \mathsf{PG}\left(1, \eta_z\right)$ and the expectation of the Pólya–Gamma distribution can be calculated as

$$\mathsf{E}\left(\omega\right) = \frac{b}{2c}\tanh\left(\frac{c}{2}\right) = \frac{b}{2c}\left(\frac{\mathrm{e}^c - 1}{1 + \mathrm{e}^c}\right), \tag{4.15}$$

with $\omega \sim \mathsf{PG}\left(b, c\right)$.

(ii) **M–step:** choose $\boldsymbol{\vartheta}^{(m+1)}$ by maximizing (4.14) with respect to $\boldsymbol{\vartheta}$

$$\boldsymbol{\vartheta}^{(m+1)} = \arg\max_{\boldsymbol{\vartheta}} \mathcal{Q}\left(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{(m)}\right). \tag{4.16}$$

One nice feature of the EM algorithm is that the solution of the M–step exists analytically for example for Gaussian and Student–t HMMs, for all the parameters with the only exception of the degrees–of–freedom $\nu_l, l = 1, 2, \ldots, L$. For the proposed model we provide in the next subsection a Conditional Expectation–Maximization algorithm (see, e.g., McLachlan and Krishnan 2007 and Frühwirth-Schnatter 2006).

4.2.4 Conditional Expectation–Maximization algorithm

E–step: at iteration $(m + 1)$, the E–step requires the computation of the so–called $\mathcal{Q}$–function, which calculates the conditional expectation of the complete–data log–likelihood given the observations and the current parameter esti-

mates $\boldsymbol{\vartheta}^{(m)}$

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\vartheta}) \propto & \sum_{l=1}^{L} \widehat{z}_{1,l}^{(m)} \log(\delta_l) + \sum_{l=1}^{L}\sum_{k=1}^{L}\sum_{t=1}^{T} \widehat{zz}_{t,l,k}^{(m)} \log(q_{l,k}) \\ & - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} \left(\log(2\pi) + \log\left(\sigma_l^2\right)\right) \\ & - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} \frac{\widehat{z}_{t,l}^{(m)} \left(y_{1,t} - \bar{\mu}_l - \gamma_l h_t\right)^2}{\sigma_l^2} \\ & + \sum_{l=1}^{L}\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} \tilde{y}_{2,t} \left(\eta_l + \alpha_l h_t\right) - \frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} \widehat{\omega}_{t,l}^{(m)} \left(\eta_l + \alpha_l h_t\right)^2,\end{aligned} \tag{4.17}$$

where the conditional expectations $\widehat{z}_{t,l}^{(m)} = \mathbb{E}(z_{t,l} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$ and $\widehat{zz}_{t,l,k}^{(m)} = \mathbb{E}(zz_{t,l} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$, $\forall t = 1, 2, \ldots, T$ and $\forall l, k = 1, 2, \ldots, L$ are computed via the well–known Forward–Filtering Backward–Smoothing (FFBS) recursive algorithm (see Baum et al. 1970). For an introduction to the FFBS algorithm we refer the reader to the book of Frühwirth-Schnatter 2006.

M–step: at iteration $(m+1)$, the M–step maximizes the function $\mathcal{Q}\left(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{(m)}\right)$ with respect to $\boldsymbol{\vartheta}$ to determine the next set of parameters $\boldsymbol{\vartheta}^{(m+1)}$. The updated estimates of the hidden parameters, the mean vector $\bar{\boldsymbol{\mu}}_l$, and the scale

matrix $\boldsymbol{\Sigma}_l$ are given by the following expressions:

$$
\begin{aligned}
\delta_l^{(m+1)} &= \widehat{z}_{1,l}^{(m)} \\
\widehat{q}_{l,k}^{(m+1)} &= \frac{\sum_{t=2}^{T} \widehat{zz}_{t,l,k}^{(m)}}{\sum_{k=1}^{L}\sum_{t=2}^{T} \widehat{zz}_{t,l,k}^{(m)}} \\
\widehat{\bar{\mu}}_l^{(m+1)} &= \frac{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} y_{1,t}}{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)}} \\
\widehat{\gamma}_l^{(m+1)} &= \frac{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} h_t y_{1,t}}{\sum_{t=1}^{T} \widehat{\omega}_{t,l}^{(m)} h_t \widehat{z}_{t,l}^{(m)}} \\
\widehat{\alpha}_l^{(m+1)} &= \frac{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} h_t \tilde{y}_{2,t}}{\sum_{t=1}^{T} \widehat{\omega}_{t,l}^{(m)} h_t \widehat{z}_{t,l}^{(m)}} \\
\widehat{\sigma}_l^{2\,(m+1)} &= \frac{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} \left(y_{1,t} - \widehat{\bar{\mu}}_l^{(m+1)} - \widehat{\gamma}_l^{(m+1)} h_t \right)^2}{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)}} \\
\widehat{\eta}_l^{(m+1)} &= \frac{\sum_{t=1}^{T} \widehat{z}_{t,l}^{(m)} \tilde{y}_{2,t}}{\sum_{t=1}^{T} \widehat{\omega}_{t,l}^{(m)} \widehat{z}_{t,l}^{(m)}},
\end{aligned}
$$

$\forall l = 1, 2, \ldots, L$, where $\widehat{\omega}_{t,l}^{(m)}$ denotes the current estimate of the conditional expectation of ω_t given the observation at time t, $\mathbf{y}_t$, and $z_{t,l} = 1$

$$
\widehat{\omega}_{t,l}^{(m)} = \mathbb{E}_{\omega|z}\left(\omega \mid y, z\right) = \frac{1}{2\eta_l} \left(\frac{\mathrm{e}^{\widehat{\eta}_l^{(m)}} - 1}{1 + \mathrm{e}^{\widehat{\eta}_l^{(m)}}} \right), \tag{4.18}
$$

$\forall t = 1, 2, \ldots, T$ and $\forall l = 1, 2, \ldots, L$.

Proof. The M–step of the EM algorithm requires the maximization of the $\mathcal{Q}$ function with respect to the parameters η_l. Observe that, conditional to the latent factors $(\boldsymbol{\omega}, \mathbf{z}, \mathbf{zz})$, the objective function is proportional to

$$
\mathcal{Q}\left(\eta_l\right) \propto -\frac{1}{2} \sum_{l=1}^{L} \sum_{t=1}^{T} \widehat{\omega}_{t,l}^{(m)} \widehat{z}_{t,l}^{(m)} \left(\frac{\tilde{y}_{2,t}}{\widehat{\omega}_{l,t}^{(m)}} - \eta_l - \alpha_l h_t \right)^2 \tag{4.19}
$$

which completes the proof. □

Chapter 5

Application to real data

After presenting the airport organization, detecting the congestion problem and developing a possible model to solve it, we now show the results of the Finite–Horizon Dynamic Programming algorithm along with a summary of the probability matrices estimates. The last section of this chapter was thought to highlight the power of the algorithm and our success in optimizing a real congestion problem. Indeed we will show the results of a comparison between what we could have obtained using our solution and what instead happened in reality. Finding a utility and being able to gain some profits by using the model is for sure a good review for it.

Before going into that, we give a summarizing descriptive overview about the performances of the airport for the year 2017.

5.1 Measures of Marco Polo airport on time performance

In a day of operations generally delays start creating in the morning until reaching peak congestion hours in the half part of the day when typically more demands are scheduled.

Figure 5.1 plots the number of planned arrivals and departures giving a time se-

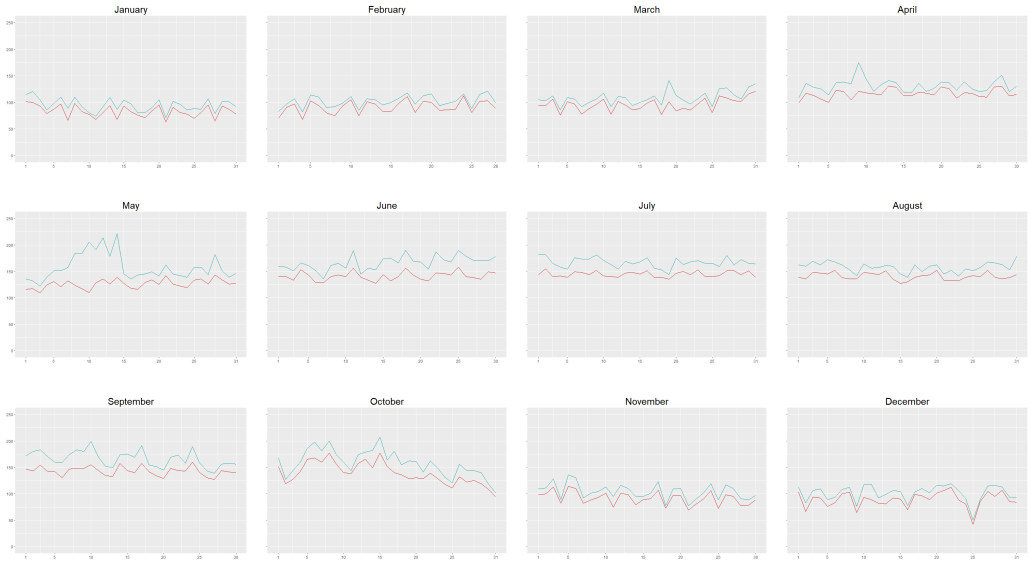


Figure 5.1: *Distribution of scheduled flights at Marco Polo airport in Venice for the year 2017. Scheduled arrivals are marked with the red line, scheduled departures with the blue.*

ries for the year 2017. Particularly we segment it to show the results divided per month. Indeed reasonably air traffic changes from month to month and even more from to season to season. As shown in graphic 5.1, we notice how total demand starts increasing in spring, reaches the highest peaks in summer and drastically falls from the end of October. Cause we are observing just one year of data we can't properly define it as a trend, but most of likely this behavior is typical in years. We also notice that in general departures and arrivals follow the same flow, but always the former are higher than the latter, especially in those periods of large total demand.

To test the model, our analysis will focus particularly on two specific months, April and December. We chose them as representative special months. In fact as we are focusing on queues and associated costs and we are taking into consideration also the variability from weather conditions, our application needed to find particularly congested days and differently affected by the weather. So we chose spring and winter as seasons and April and December as representing congested

months thinking that in April 2017 bank holidays like Easter (16^{th} day for the year 2017) and respectively in December Christmas (25^{th} day) and special new year holidays occurred.

To resume the statement made at the beginning of this section, about the scheduled plans, we report more in detail the ones for April (5.2) and December (5.3). Differently from the overview in Figure 5.1 where for each day the scheduled

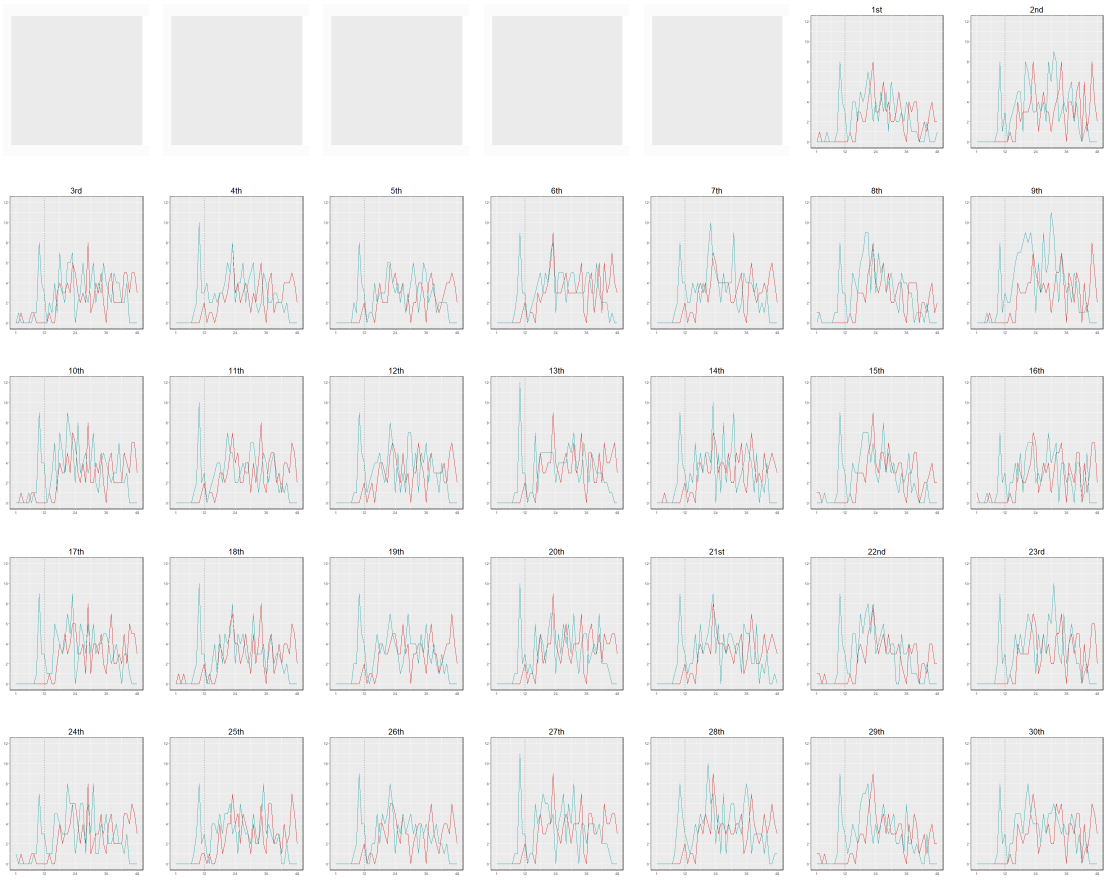


Figure 5.2: *Distribution of scheduled flights for April 2017 distinguished for type of movement. Arrivals are marked with the red line, departures with the blue. Representation is detailed for each day of the month, with reference to each 30–minute length time interval. The dotted vertical line is in correspondence to the time period $t = 1$, i.e. between 5.30 a.m. and 6.00 a.m.*

Figure 5.3: *Distribution of the scheduled flights for December 2017 distinguished for type of movement. Arrivals are marked with the red line, departures with the blue. Representation is detailed for each day of the month, with reference to each 30–minute length time interval. The dotted vertical line is in correspondence to the time period* $t = 1$*, i.e. between 5.30 a.m. and 6.00 a.m.*

movements are the sum of those planned for that day, in Figures 5.2 and 5.3 the details are for each time period of the day. Interpreting the x–axis as the time interval, namely the first ranging from 00.00 to 00.30 till the last one from 23.30 to 24.00, we can confirm that generally the majority of flights is planned starting from mid morning, especially if we focus on the arrivals. These indeed reach their peaks in the half part of the day and in the last hours they are always higher than the planned departures. The days are disposed as in a calendar, from the first column showing all the Mondays till the last one showing all the Sundays. In this way it's even better understandable how the planning flows during the weekdays.

Figures 5.4 and 5.5 show instead the formation and propagation of the observed queues distinctly from arrivals and departures. The vertical dotted line traces the first significant time period $t = 1$, between 5.30 and 6.00, from the series $t = 1, \ldots, T = 37$ chosen for the model application. The images confirm the formation of considerable queues is not before the early morning, thus giving reason to our split–of–the–day choice (for details see chapter 3). All considered we need to bear in mind that while for the departures the estimation of queue lengths has been made with a reasonable criteria thanks to large available information, for the arrivals we used a more approximate formula (see details in section 3.3).

5.2 Estimates and results

All the estimates were obtained using the software R, a free open–source environment for statistical computing and graphics. For codes and packages used for our application we refer to Appendix C.

Assuming an order $k = 2$ for the Erlang distribution of the service process, a practical queue capacity limit $N = 10$ and a demand rate λ_t equal to the scheduled movements request, we programmed the estimates as detailed in chapter 2, testing all the possible combinations between arrival and departure service rates. For the parameter ρ we set it equal to 1.2, that implying for arrival queues costs 20%

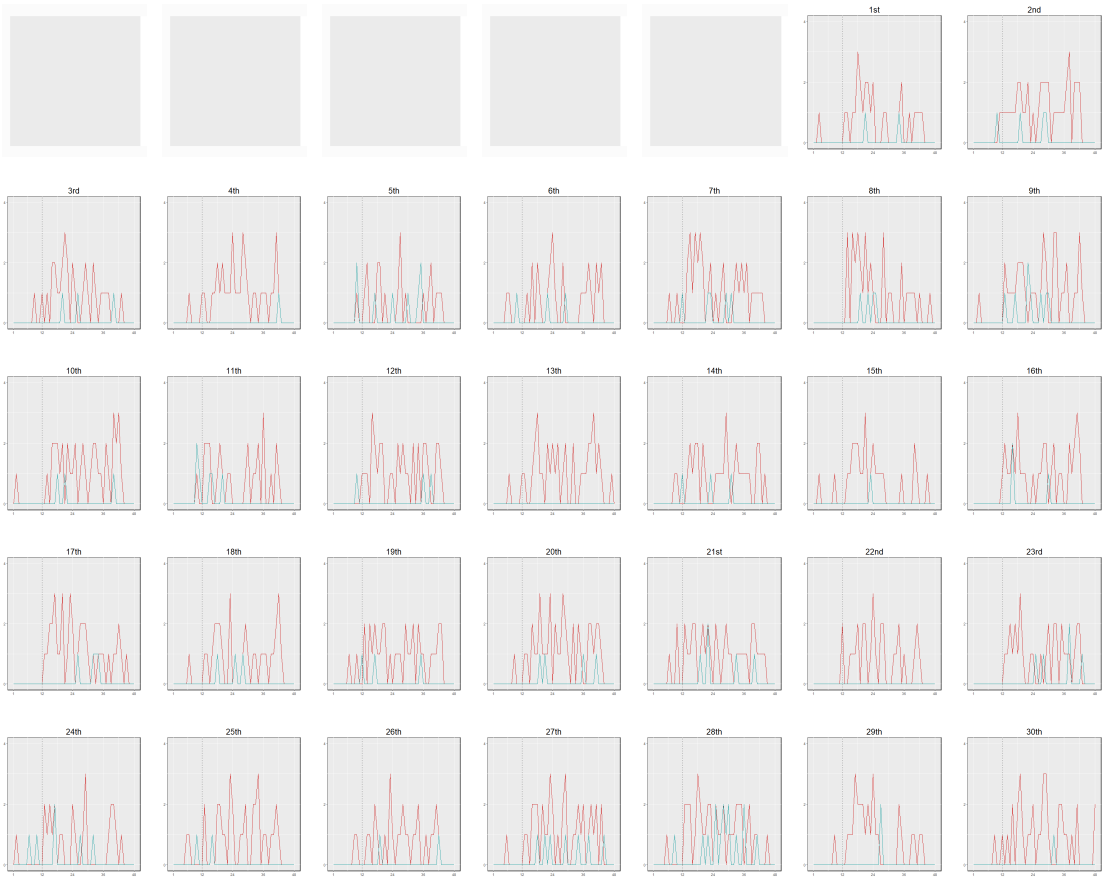


Figure 5.4: *Distribution of the observed arrival and departure queues lengths for April 2017. Arrivals are marked with the red line, departures with the blue. Representation is detailed for each day of the month, with reference to each 30–minute length time interval. The dotted vertical line is in correspondence to the time period* $t = 1$*, i.e. between 5.30 a.m. and 6.00 a.m.*

Figure 5.5: *Distribution of the observed arrival and departure queues lengths for December 2017. Arrivals are marked with the red line, departures with the blue. Representation is detailed for each day of the month, with reference to each 30–minute length time interval. The dotted vertical line is in correspondence to the time period* $t = 1$*, i.e. between 5.30 a.m. and 6.00 a.m.*

higher than for the departure ones. Actually that sounds reasonable as keeping an aircraft waiting on air usually means more fuel wasting if compared to wait on the ground.

With regards to the set of possible service rates, $\mathbf{A}_{w_t}$, it was chosen to be different whether referring to good or poor weather conditions:

$$\mathbf{A}_{w_t} = \begin{cases} \{0;1;2;3;4;5;6;7;8;9;10\}, & \text{if } w_t = VMC \\ \{0;1;2;3;4;5\}, & \text{if } w_t = IMC. \end{cases}$$

The choice to set as maximum service rate 10 and 5 respectively under VMC and IMC has been suggested by the limits as reported in Table 3.1. The set was only necessary for the arrival service rate because the corresponding for departures was determined by the OTE and what has been reached in reality.

The Erlang distribution's order is to be interpreted as the number of phases that a customer, an aircraft in our case, requires to be completed for a full service. From the departure's side, we decided to set $k = 2$ as once the aircraft leaves the gate, so when its service starts, it might join another queue when reaching the holding point. In fact here the plane is considered ready to depart but this time can differ from the moment when it actually takes off. We decided to set the same order's value also for the arrivals.

The manual choice of N was dictated from the maximum possible service rate of 32 movements per hour, with a maximum of 19 arrivals, which was roughly converted into 16 and 10 movements per each 30–minute time period respectively. For a seek of simplicity and coherency we set the same practical queue capacity value of $N = 10$ both for arrivals and for departures. We could have set the practical queue capacity bigger than 10 but it would have meant more complexity into the model and a bigger computational cost.

The imputation to λ_t of a deterministic value sounded reasonable as seeking for an optimal service policy considering one specific day for which we know in advance its complete flight plan. Furthermore we remind that the flight plan we took as

input is not the one planned months in advance, but the last updated up to three hours before the actual performance, at least for departures. All considered however in the real world such scheduled flights may be affected by unpredictable changes for example cause of unexpected events from the connecting airports or unforeseen delays from the passengers or again low efficiency by the flight operators.

Finally with regards to the weather variable we used the approach detailed in section 4.2.1, where for our application the number of possible states (L) was 2, VMC or IMC, good or bad weather conditions respectively.

5.2.1 Probabilities matrices

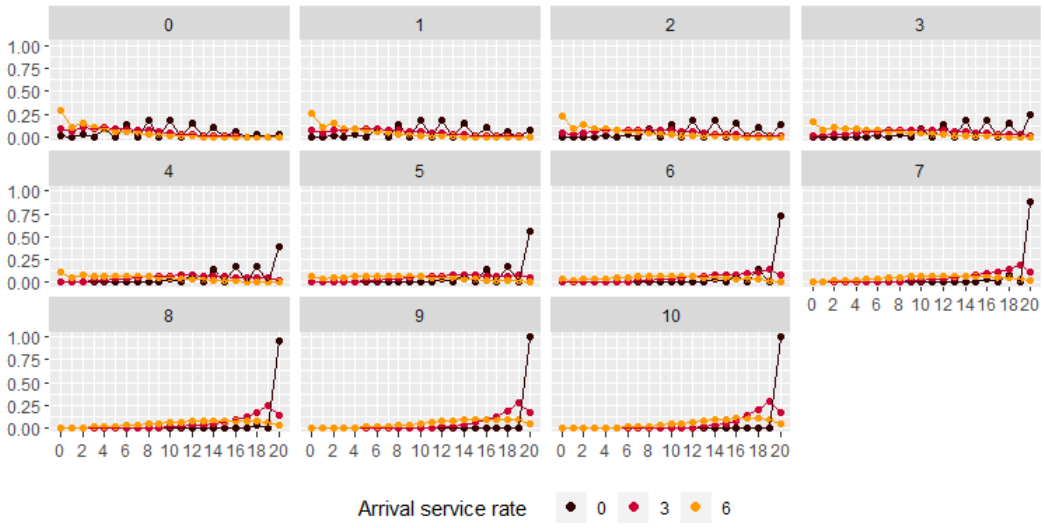


Figure 5.6: *Example of* $\mathbf{P}$ *matrix estimate for arrival queues. Separately for the number of aircraft waiting to be served at the beginning of the period, each plot shows the state probability of the airport system with regards to the number of stages of work to be completed at the end of the time period, which is* $t = 16$ *(13.00–13.30).*

Figures 5.6 and 5.7 are an example of the estimates for the Chapman–Kolmogorov system of first–order differential equations, detailed in 2.2.1.

The results from each running of the algorithm were graphs showing the evo-

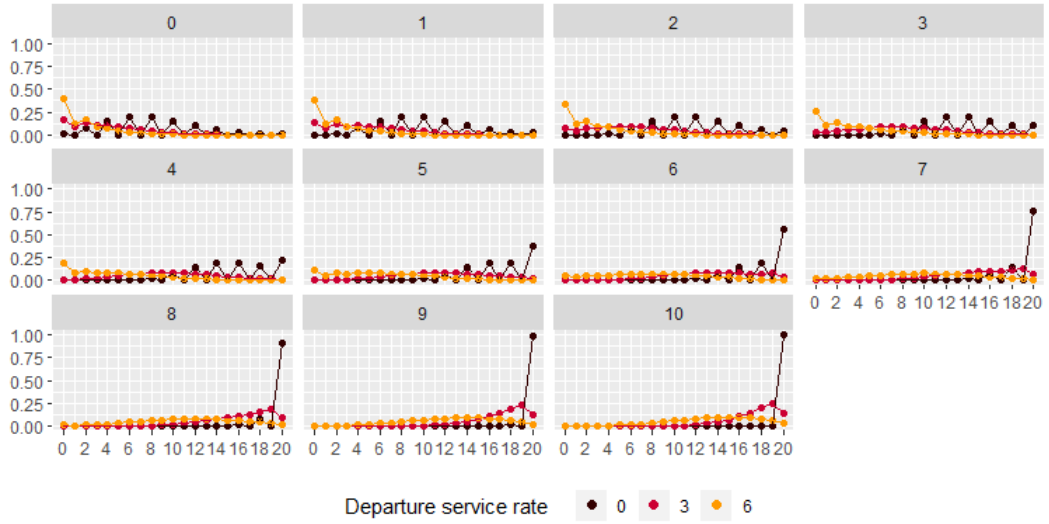


Figure 5.7: *Example of* $\mathbf{P}$ *matrix estimate for departure queues. Separately for the number of aircraft waiting to be served at the beginning of the period, each plot shows the state probability of the airport system with regards to the number of stages of work to be completed at the end of the time period, which is* $t = 16$ *(13.00–13.30).*

lution of the state probabilities over time period t. For our purpose we just needed to store in a look–up table the values referring to the beginning (the initial conditions) and the end of the time period (the one we were looking for). The estimation process took about 5 minutes to run in a laptop computer, as much for the arrival as for the departure rates. It is worth to note that the computational effort was higher depending on the values of k and N. In our case we obtained a total of $(N+1) \times |\mathbf{A}^{\mu}| \times T = 4477$ stages of work transition probabilities matrices $\mathbf{P}$ of dimension $(kN+1) \times (kN+1)$ which is for us (21×21) for arrivals and as many for departures[1]. The examples are with reference to time period $t = 16$, i.e. between 13.00 and 13.30, for the chosen day December 10^{th}, 2017. Figures 5.6 and 5.7, the former for arrival and the latter for departure, depict the possible state at which

[1] We computed the matrices for all the possible queues lengths and service rates, even if some of them, for example the highest departure service rates, were never reached or proposed to be adopted as solution to the congestion.

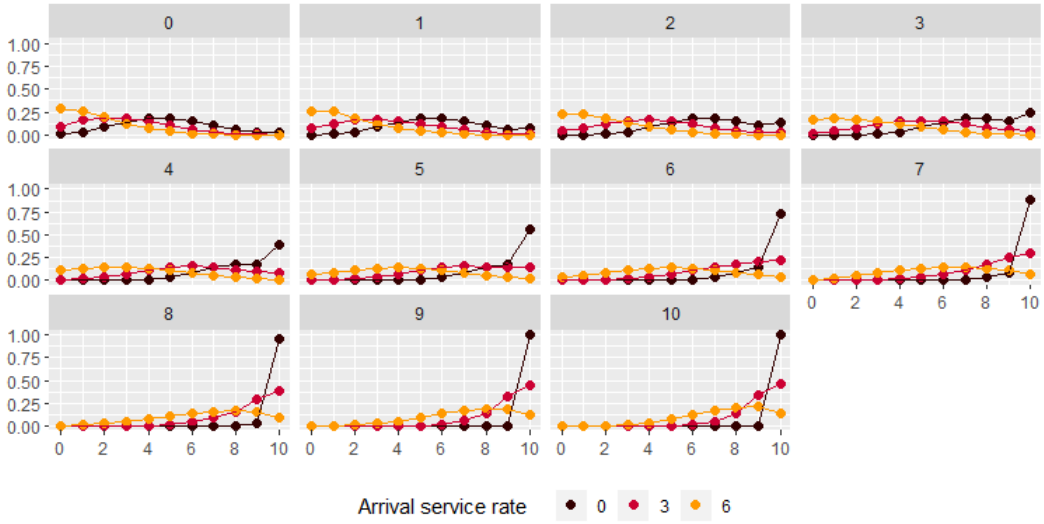


Figure 5.8: *Example of* $\mathbf{Q}$ *matrix estimate for arrival queues. Separately for the number of aircraft waiting to be served at the beginning of the period, each plot shows the state probability of the airport system with regards to the number of aircraft remaining to be fully served at the end of the time period, which is* $t = 16$ *(13.00–13.30).*

the airport system could be at the end of the time period, i.e. at 13.30, given that at the beginning of the same there were a known number of aircraft in queue asking to be served. The initial condition which translates into this number is indicated at the top of the each box, the x–axis is the total stages of work remaining to be completed at the end of the period, with probability reported in the y–axis. The dynamic realistically changes depending on the corresponding service rate adopted in the considered interval of time. We chose 0, 3 and 6 as representative for the example. Figures 5.8 and 5.9 show in the same way as the stages of work transition probabilities matrices $\mathbf{P}$ an example of the queue length transition probabilities $\mathbf{Q}$, detailed in 2.2.2. The total number of states for them is reduced to 11, while for the matrices $\mathbf{P}$ it was 21.

About the estimation of the weather state transition probabilities (matrices $\mathbf{W}_t, t = 1, \ldots, T = 37$), an example of the results is shown in Table 5.1, where the instance is for $h_t = 16$ (for details about definitions see subsection 4.2.2). Figure

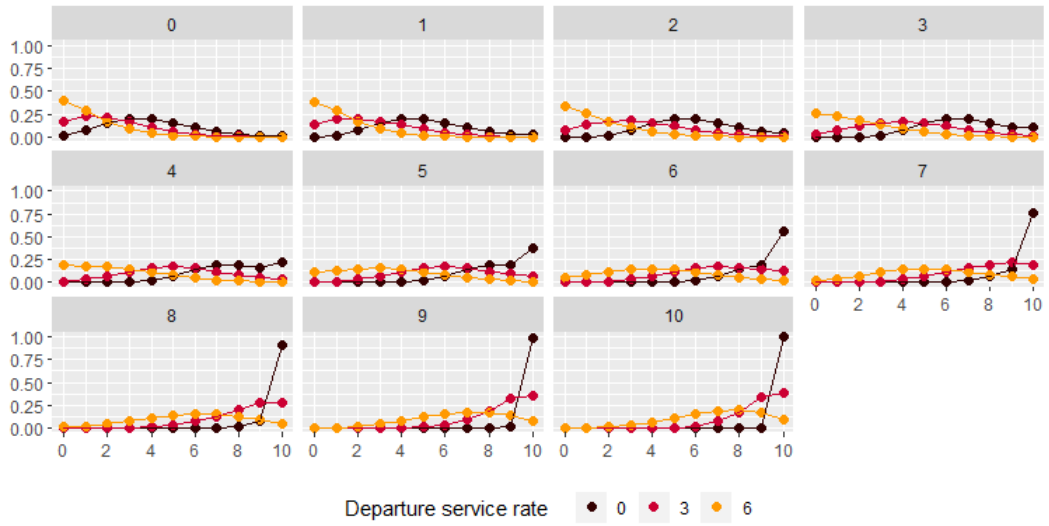


Figure 5.9: *Example of* **Q** *matrix estimate for departure queues. Separately for the number of aircraft waiting to be served at the beginning of the period, each plot shows the state probability of the airport system with regards to the number of aircraft remaining to be fully served at the end of the time period, which is* $t = 16$ *(13.00–13.30).*

5.10 instead shows the filtered probabilities for the estimated HMM, introduced in section 4.2, for the good weather state $\bar{s}_t = VMC$, i.e. $\mathbb{P}(\bar{s}_t = 1|\mathbf{y}_{1:t}, \mathbf{h}_{1:t})$. The estimates are obtained using the FFBS algorithm described in Appendix A.1.

5.2.2 OTE functions estimates

Reminding that the OTE function estimates the average number of allowed take–offs in correspondence to the allowed landings for a time period, we show below the estimated functions, whose formulation is detailed in chapter 4. We present the estimates for the two selected months but we add that they can be valid in general for spring and winter seasons respectively. Indeed the related probabilities have been obtained using aggregated data from months similar in weather to April in the first case and to December in the second. Figures 5.11 and 5.12 report such estimated functions. With reference to them, lines are the estimated OTE

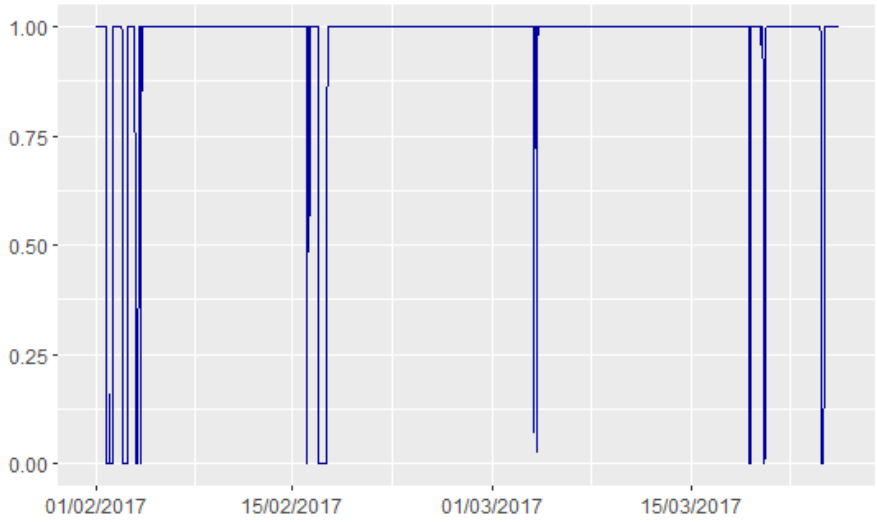


Figure 5.10: *Filtered probabilities for the estimated Hidden Markov model for the good weather state* $\bar{s}_t = VMC$.

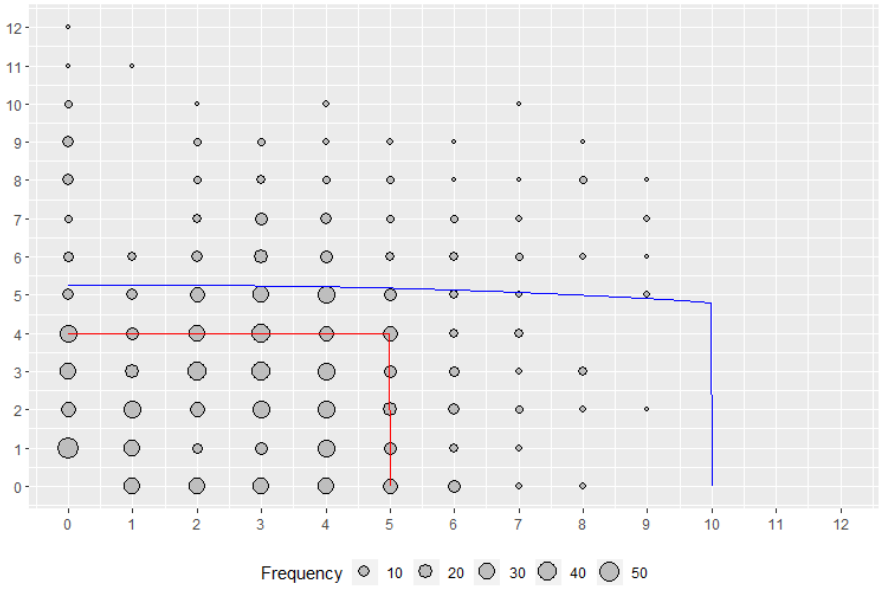


Figure 5.11: *Operational throughput envelope estimate for April. The blue function is to be considered when under VMC state, the red one under IMC. The dots whose size is proportional to the frequency are the scheduled flights planned for one time period in a day. They summarize the schedules for an hypothetical spring month, taking as example April 2017.*

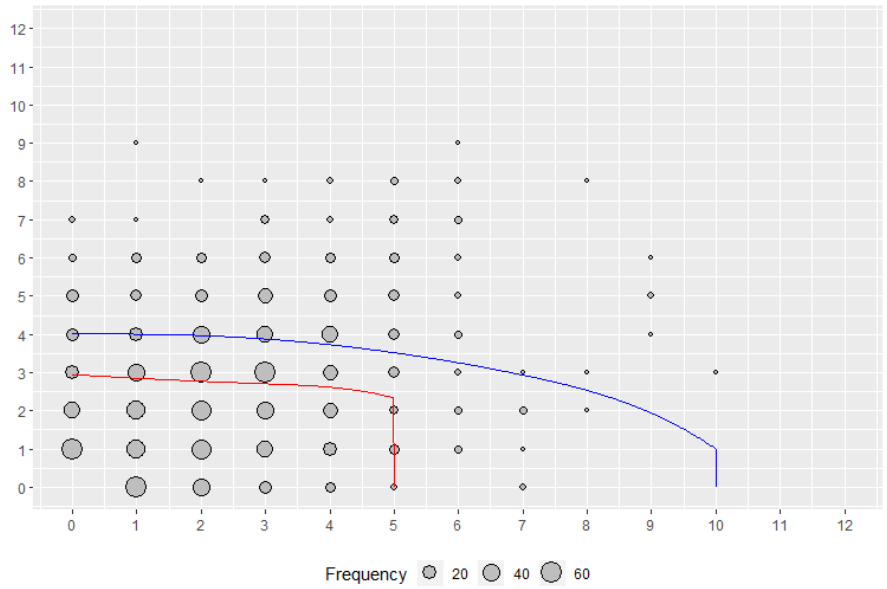


Figure 5.12: *Operational throughput envelope estimate for December. The blue function is to be considered when under VMC state, the red one under IMC. The dots whose size is proportional to the frequency are the scheduled flights planned for one time period in a day. They summarize the schedules for an hypothetical winter month, taking as example December 2017.*

	$\widehat{p}_{h_t}$	$\widehat{q}_{h_t}$
$h = 16$	0.995	0.890
	(0.0472)	*(0.0826)*

Table 5.1: *Estimation of the probabilities of transition from a weather state to another, conditioned on seasonality. $\widehat{p}_t$ is the probability to be in VMC at the beginning of time period $t+1$ given that the state was in VMC at the beginning of t. $\widehat{q}_t$ is the probability to be in IMC at the beginning of time period $t+1$ given that the state was in IMC at the beginning of t.*

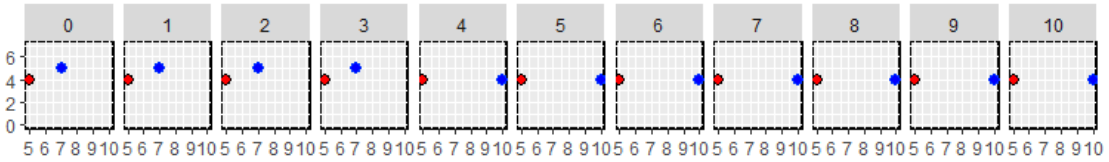
functions, the blue ones show the trade–off between the two variables under VMC, i.e. good weather conditions, and the red ones the same under IMC, i.e. bad weather conditions. It's clearly noticeable the restriction inferred by the meteorologic state on the achievable service rates. Under VMC as well as allowing more departures on average for the same chosen rate for arrivals, it can be reached the maximum possible arrival service rate. With regards to the function in spring, we see that the departure service rates are almost constant and equal to 5 and 4 movements, respectively under VMC and IMC conditions, up to the correspondent maximum arrival service rates of 10 and 5, where there they fall to 0, according to the airport capacity. The reason why the estimates result in a constant line can be due to lack of information from data. Indeed in spring and equivalently in summer there are only a few occasions of bad weather or windy conditions, especially if compared to those of good ones. We also decided to plot in the same graphic some points which represent the pairs of scheduled arrivals and departures in a 30–minute time period for a hypothetical day. The size of such dots is proportional to the frequency of the observed variables' pairs, indeed we summarized data for the same daily interval from the whole month of April (5.11) and December (5.12) of the year 2017. We can notice in this way that the flight schedules per day laying above the OTE lines

are index of a congestion problem. As a matter of fact their position above the service limit line means there were more aircraft planned to be served than those the airport system was really able to serve. This is an example of how queues arise.

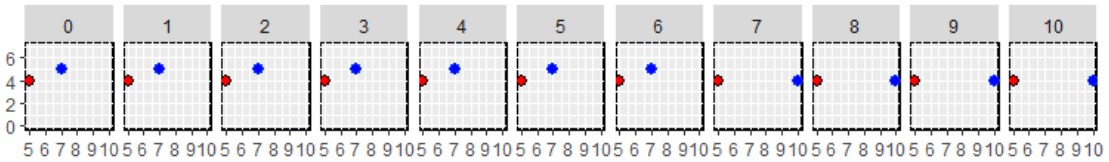
5.2.3 Solution of the DP algorithm

Taking as inputs all the necessary estimated parameters, we succeeded in running the Dynamic Programming algorithm. In a desktop PC the effective time to run the dynamic part was roughly 15 minutes per day. To gain a feedback on which draw some consistent conclusions we decided to solve the algorithm per each day of the two selected months. At the end of the running for one day we had one matrix for every time period with $(N+1) \times (N+1) \times 2 = 242$ rows, one for each possible three–dimensional state. Every matrix, taking generally the one for period t, suggests the optimal service rate to adopt in the next 30 minutes of time starting from tR, given that we are in the state we are observing (for the state variable definition, see chapter 2, expression (2.1)). The forwarded solution from the algorithm is to be read as the best service to be applied to reach the minimum cost at the end of the day, so from clock time tR on.

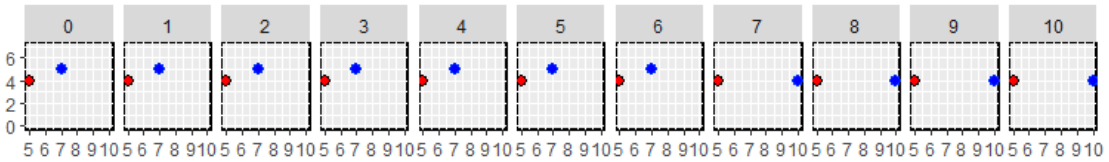
Figure 5.13 reports a piece of solution, taking as example one day of April and specifically for the time period $t = 16$. Each figure is conditioned to the number of aircraft queuing for departing (we only report some of the most relevant examples), while each of their plots is conditioned on the arrival queue length. Depending on both the arrival and departure queue lengths and the expected number of aircraft demanding for service in that time period indeed the rate chosen as optimal from the DP algorithm can be different. Furthermore for the way it has been built the model choses that service rate that is expected to be the best one to adopt since then in view of what is going to happen in the future, i.e. until the end of the day. Also, considering that under bad weather conditions the service could be limited, the optimal choice depends on that as well. For this reason each box in Figure 5.13 contains two points, the blue one associated to VMC and the red one to IMC



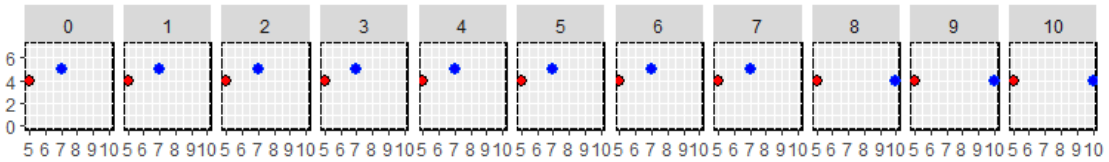
(a) 0 aircraft queuing for departing



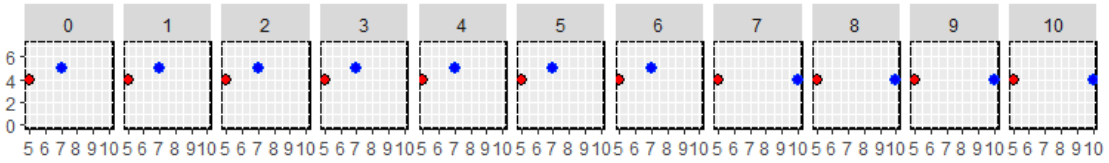
(b) 2 aircraft queueing for departing



(c) 5 aircraft queueing for departing



(d) 8 aircraft queueing for departing



(e) 10 aircraft queueing for departing

Figure 5.13: *Example of the optimal selection for arrival (x–axis) and departure (y–axis) service rates from the DP algorithm. The sample is taken from the solution for the time period* $t = 16$*, i.e. between 13.00 and 13.30, of one day of April 2017. Each figure is conditioned on the departure queue length at the beginning of the period (13.00) and each box shows the best combination to choose as service rate depending on the corresponding arrival queue length (indicated with numbers from 0 to 10). Blue dots are the optimal choice under VMC weather conditions, red dots under IMC.*

conditions, as observed at the beginning of the time period, that is in the example at 13.00.

5.3 Comparison

Last step in this analysis is about the comparison between optimal situation and reality. When saying optimal solution we refer to the selection of the best control variables, arrival and departure service rates, for a determined time period. Such selection is again depending on the current state in which the airport system is at the beginning of the interval. The solution is disposable at any time and before the day of operations and its power is just that. Being indeed known, once the situation is observed, it suggests the optimal management of the traffic, in anticipation of what is expected to happen until the end of the day of operation and considering also the weather influence on the airport capacity and operations' efficiency.

To build a proper comparison between model proposal and real decisions, we should know what the service policy criteria adopted by air traffic managers is. From the dataset we disposed of information about arrival and departure service rates, but they are the ones chosen in real time, thus not provided before the relative operation. For this lack of information we were not able to define a comparable service policy from real data, but instead we built a hypothetical one. We fixed thus for each day its selected service rates equal to the scheduled demand rates. After that we run again the DP algorithm with the same backward approach, knowing this time the service rates to be adopted. In this way we had for each day of April and December two possible strategies to follow: the optimal one coming from our model and the naïve one, chosen as a hypothetical adopted policy. The comparison was made by means of the resulting costs quantified as shown in expression 2.2, choosing for our application $\rho = 1.2$. We just compared the total costs related to the first time period t estimated in both cases by the algorithm. As a matter of fact, the first time period is the resume of the whole day once it

takes into consideration through the probability estimates all the possible dynamic evolutions of the system during each day and all the combinations of choices that can be taken.

Barplots in Figures 5.14 and 5.15 draw the gain in cost in percentage scale, where

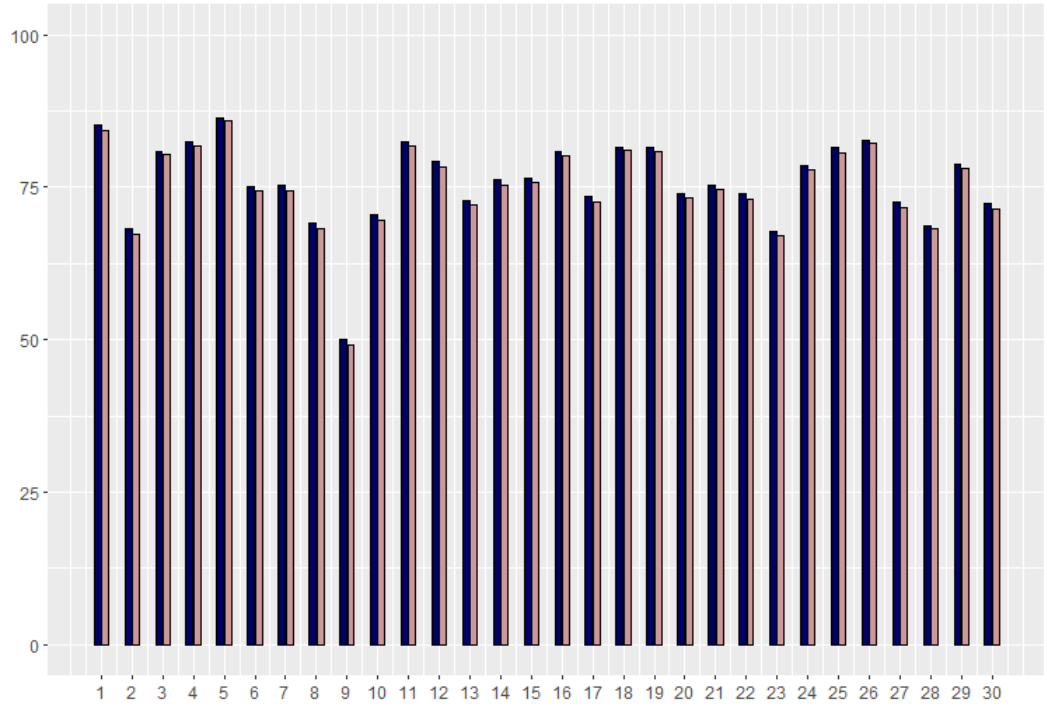


Figure 5.14: *Barplots for the resulting cost comparison between model suggestion and naïve control service policy. Bars quantify the gain from the model in percentage scale. X–axis indicates the days of April 2017 and for each the two bars stay for the gain in cost if the relative day began under VMC weather conditions (dark bar), the gain in cost if the relative day began under IMC (light bar).*

a positive value means that the model performed better than the naïve solution. Two bars are assigned to each day, which is the x–axis. The dark bar is the percentage gain in cost if the day begins under VMC conditions, while the light bar if under IMC condition. Always, even if sometimes only slightly, in the example the gain when the day begins under good weather is higher than when in bad weather state. This can find explanation in the fact that under bad weather condition the

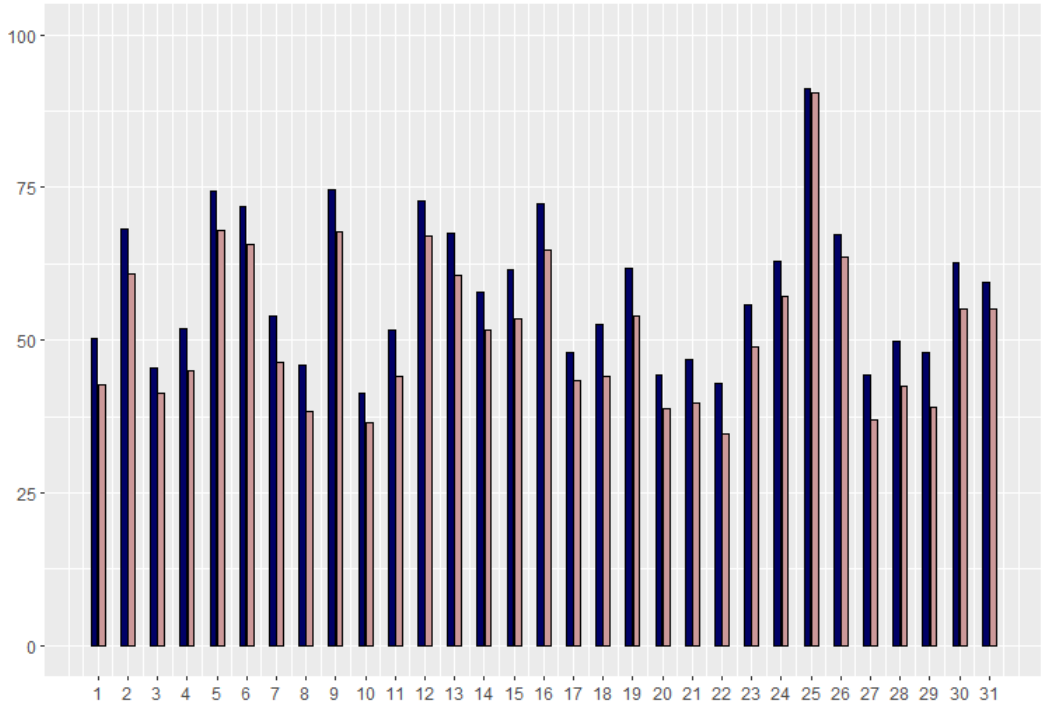


Figure 5.15: *Barplots for the resulting cost comparison between model suggestion and naïve control service policy. Bars quantify the gain from the model in percentage scale. X–axis indicates the days of December 2017 and for each the two bars stay for the gain in cost if the relative day began under VMC weather conditions (dark bar), the gain in cost if the relative day began under IMC (light bar).*

services are limited, thus also the attempt to do better is constrained. Anyway as barplots show, it seems that our solution performs better than the naïve one up to gain in costs of the 80%.

For a comparable service rate from reality instead it was not possible to extract one as we said, but at least we could study a way to compare model solution to reality by means of queues lengths. Indeed although conscious about the approximation and the ignorance of any unpredictable events that might have occurred, we took as real service rates the actual arrival and departure throughputs (for definitions see section 3.3) and used them for a comparison with the optimal service policy from the model by measuring the final queues lengths. Assuming there were no queues in $t = 1$ which means for our analysis the beginning of the day, the comparison took shape by calculating at the end of each time period the arrival and departure queues lengths as the sum of the respective queue length at the beginning of the period and the number of movements planned to serve, minus the chosen service rate. Given that the queues lengths at the beginning of a time period correspond to those at the end of the previous one, the only element to define was the number of flights planned to serve. Here we had two chances: first one was using the flight plan disposable at the beginning of the day, which is the same used to define the rates λ_t in the probability matrices' definition, see chapter 2. The second alternative was considering as flight plan the actual demand arisen in that specific day, which might be different from the previous proposal. Again from data we were not able to reach sufficient information about the second alternative, even if it would have been more reliable and motivating the service rates' choice adopted by air traffic managers. Anyway, flight plan is still a good choice given that it is the last updated up to three hours before the corresponding movement, at least for departures.

To show the evolution of arrival and departure queues when adopting the two different service policies we refer to Figures 5.16 and 5.18 and Figures 5.17 and 5.19 respectively. Each plot is for one day comparison. Red dots are with reference to

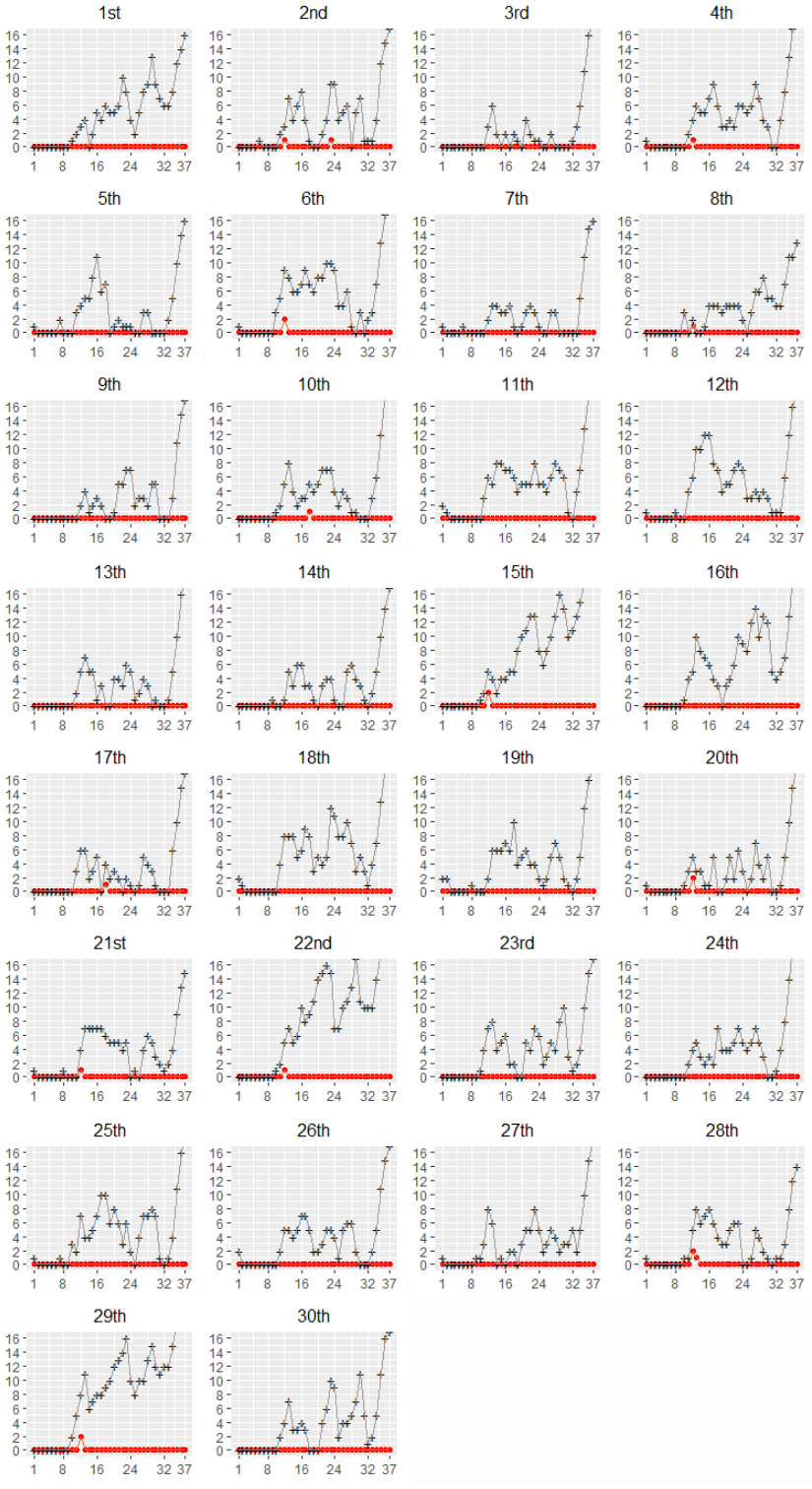


Figure 5.16: *Arrival queue lengths comparison between model solution and actually performed service. Red dots refer to the model, black crosses to reality. The comparison takes as schedules the last updated flight plan. Each plot is with reference to the days of April 2017.*

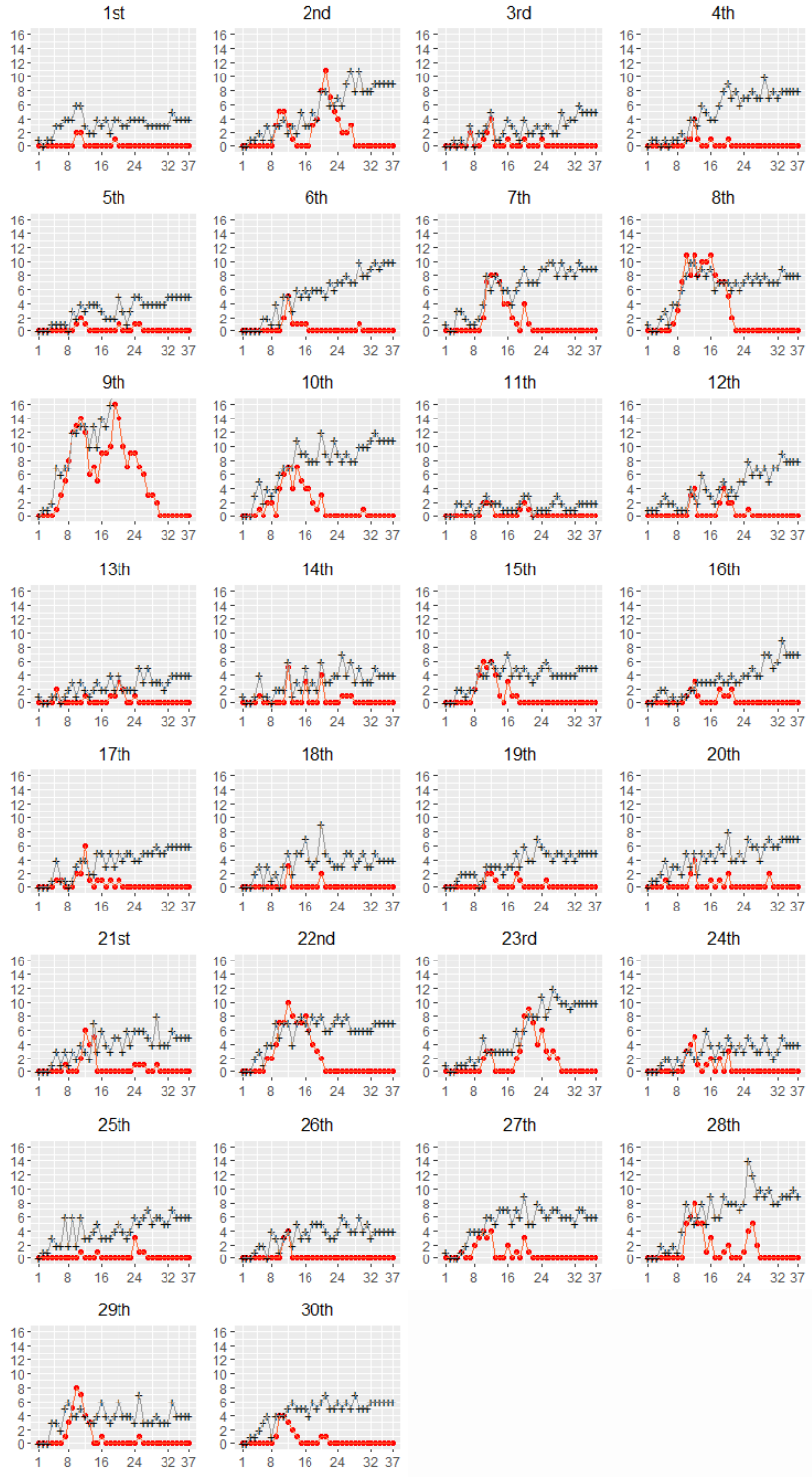


Figure 5.17: *Departure queue lengths comparison between model solution and actually performed service. Red dots refer to the model, black crosses to reality. The comparison takes as schedules the last updated flight plan. Each plot is with reference to the days of April 2017.*

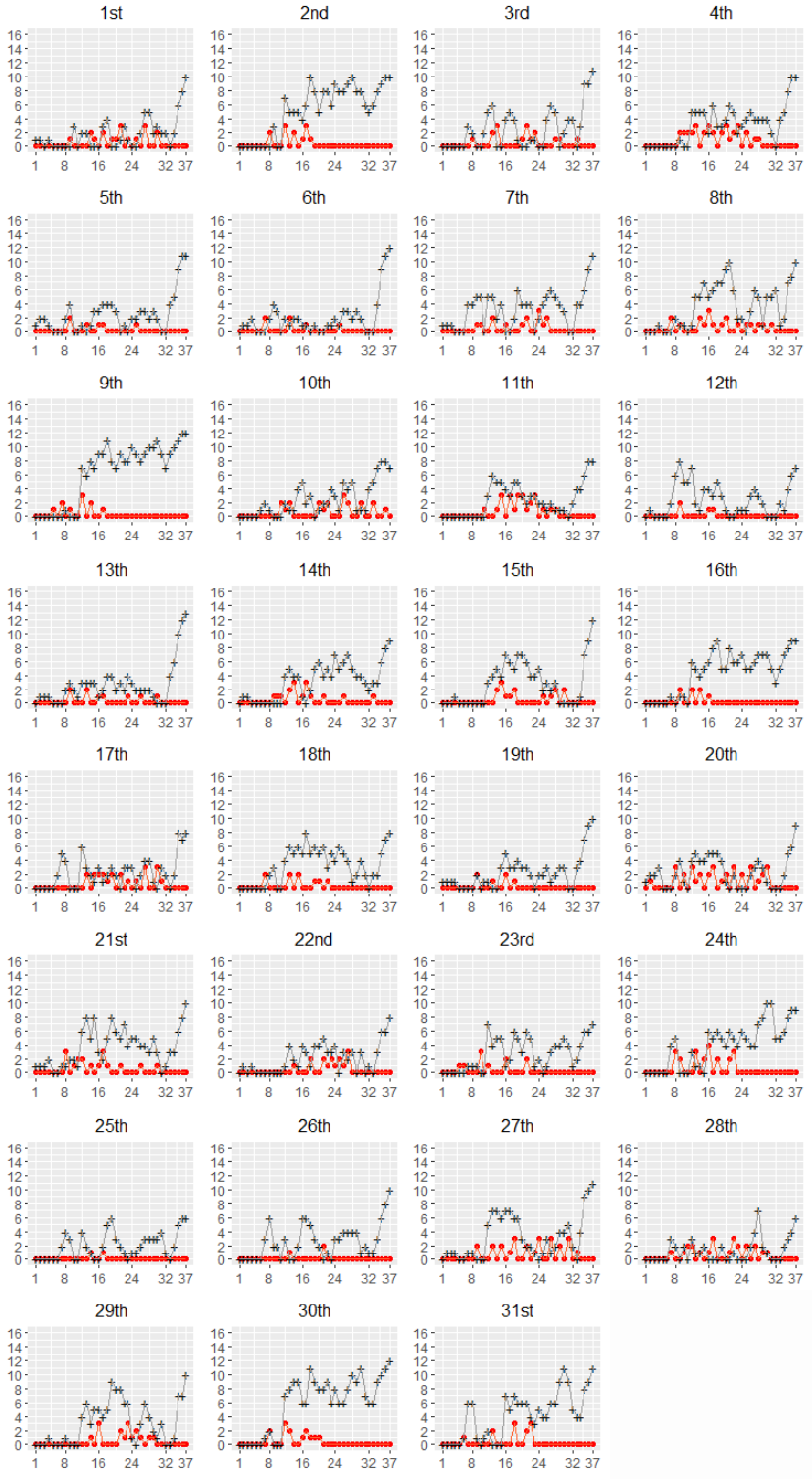


Figure 5.18: *Arrival queue lengths comparison between model solution and actually performed service. Red dots refer to the model, black crosses to reality. The comparison takes as schedules the last updated flight plan. Each plot is with reference to the days of December 2017.*

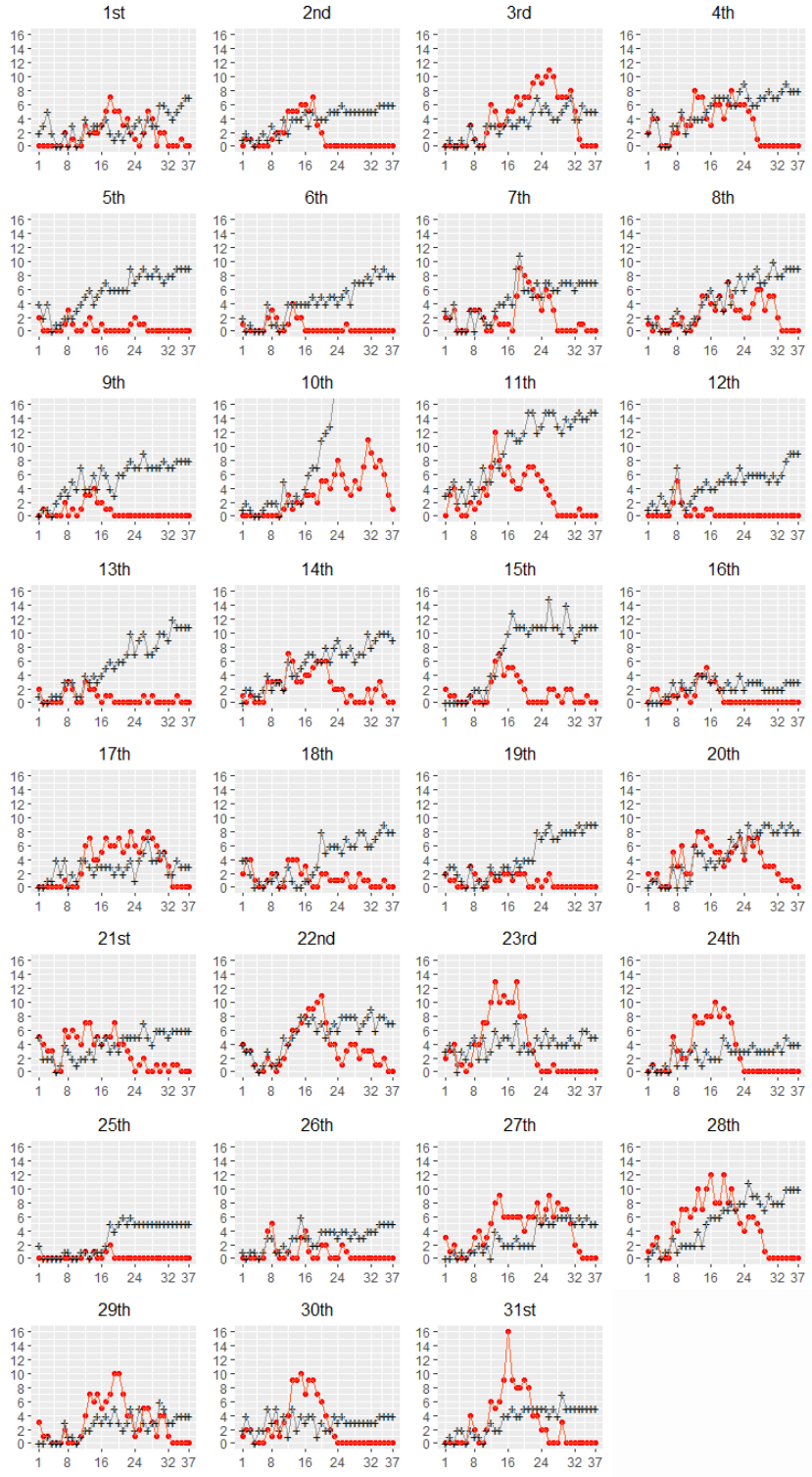


Figure 5.19: *Departure queue lengths comparison between model solution and actually performed service. Red dots refer to the model, black crosses to reality. The comparison takes as schedules the last updated flight plan. Each plot is with reference to the days of December 2017.*

the service rates we would have adopted if using the model optimal solution, while black crosses to the actual arrival and departure aircraft served in the indicated days. The difference between arrival and departure queues is evident. The former, apart from few occasions, result in low values from the model solution appliance (red dots) and values lower than the reached in reality (black crosses). For the latter instead, lengths are taller and sometimes model queues are even longer than the real ones. Reminding that this remains an approximation given that we can't know what really drove the air traffic managers' choice for service, we can justify the difference as follows. Considering that the model gives priority to serve arrivals (the meaning of $\rho > 1$), arrival queues are likely to be disposed of as first, consequently increasing departure queues length. However a solution to this congestion can be proposed. An idea could be reorganizing the departure exits from the gates in order to avoid congestion at holding points or runway.

Appendix A

Estimates for the EM algorithm

A.1 The FFBS algorithm

In the E–step of the EM algorithm, we need to compute the following conditional expectations

$$\widehat{z}_{tl} = \mathbb{E}(z_{tj} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

and

$$\widehat{zz}_{tlk} \;\; = \;\; \mathbb{E}(zz_{tj} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

The above quantities can be computed via the well–known forward–backward recursive algorithm (Baum et al. 1970). Let us define the forward variable

$$\alpha_{tl} = f(\mathbf{y}_1, \ldots, \mathbf{y}_t, \bar{S}_t = l)$$

which represents the probability of seeing the partial sequence ending up in state l at time t, and the backward variable

$$\beta_{tl} = f(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid \bar{S}_t = l).$$

It is worth noting that the computation of the forward and backward probabilities is susceptible to under– or over–flow error. In applying EM as described here, a

scaling procedure is adopted in order to prevent, or at least reduce the risk of, such error. It will be convenient to work on the log–scale. In order to do so, we exploit a simple computational device which is based on the following general equality

$$\log(a+b) = \log(a) + \log(1 + \exp(\log(b) - \log(a))).$$

Hence, if one has the log of two quantities $\log(a)$ and $\log(b)$, only their difference must be exponentiated in order to obtain $\log(a+b)$, drastically reducing the risk of underflow. By iterating this reasoning, one can sum a vector of quantities on the log–scale. We call this operation $\oplus$. The forward recursion, on the log–scale, is then given by

$$\log(\alpha_{1l}) = \log(f(\mathbf{y}_1 \mid \bar{S}_1 = l)) + \log(\delta_l).$$

Then, for $t = 2, \ldots, T$ we compute

$$\log(\alpha_{tk}) = \log(f(\mathbf{y}_t \mid \bar{S}_t = k)) + \bigoplus_{l=1}^{L} \log(\alpha_{t-1,l}) + \log(q_{lk}).$$

Similarly, it is possible to implement the following backward recursion.

$$\log(\beta_{Tl}) = 0.$$

Then, for $t = T-1, \ldots, 1$, we have

$$\log(\beta_{tl}) = \bigoplus_{k=1}^{L} \log(f(\mathbf{y}_{t+1} \mid \bar{S}_{t+1} = k)) + \log(\beta_{t+1,k}) + \log(q_{lk}).$$

The expected values of the quantities involved in the E–step can be computed as follows

$$\widehat{z}_{tl} = \frac{\alpha_{tl}\beta_{tl}}{\sum_{l=1}^{L} \alpha_{tl}\beta_{tl}}$$

and

$$\widehat{z}\widehat{z}_{tlk} = \frac{q_{lk}\alpha_{tl} f(\mathbf{y}_{t+1} \mid \bar{S}_{t+1} = k)\beta_{t+1,k}}{\sum_{l=1}^{L} \alpha_{Tl}}.$$

A.2 The observed information matrix

Under standard regularity conditions, the score vector $\nabla_t(\boldsymbol{\phi}, \mathbf{y}_t)$ evaluate at the true parameter vector $\boldsymbol{\phi}_0$ has the martingale difference property, therefore the maximum likelihood estimator will be consistent and asymptotically normally distributed with asymptotic variance–covariance matrix which is the inverse of the information matrix

$$\begin{aligned}\mathcal{I}(\boldsymbol{\phi}_0) &= p\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \nabla_t(\boldsymbol{\phi}_0, \mathbf{y}_t)\,\nabla_t(\boldsymbol{\phi}_0, \mathbf{y}_t)' \\ &= \mathbb{E}\left(\nabla_t(\boldsymbol{\phi}_0, \mathbf{y}_t)\,\nabla_t(\boldsymbol{\phi}_0, \mathbf{y}_t)'\right), \end{aligned} \tag{A.1}$$

which is not available in closed form (see Fiorentini, Sentana, and Calzolari 2003) and can be consistently estimated as

$$\widehat{\mathcal{I}}\left(\hat{\boldsymbol{\phi}}\right) = \frac{1}{T}\sum_{t=1}^{T} \nabla_t\left(\hat{\boldsymbol{\phi}}, \mathbf{y}_t\right)\nabla_t\left(\hat{\boldsymbol{\phi}}, \mathbf{y}_t\right)',$$

where $\nabla_t\left(\hat{\boldsymbol{\phi}}, \mathbf{y}_t\right)$ is the observed score at time t evaluated at $\hat{\boldsymbol{\phi}}$, the maximum likelihood estimate of $\boldsymbol{\phi}$.

Appendix B

Encoding

Each interval is codified with a number, that corresponds to the t label in formulas, $pos = 1, \ldots, T = 37$. Table B.1 encodes the symbols:

t	from	to
1	05:30	06:00
2	06:00	06:30
3	06:30	07:00
4	07:00	07:30
5	07:30	08:00
6	08:00	08:30
7	08:30	09:00
8	09:00	09:30
9	09:30	10:00
10	10:00	10:30
11	10:30	11:00
12	11:00	11:30
13	11:30	12:00
14	12:00	12:30
15	12:30	13:00

t	from	to
16	13:00	13:30
17	13:30	14:00
18	14:00	14:30
19	14:30	15:00
20	15:00	15:30
21	15:30	16:00
22	16:00	16:30
23	16:30	17:00
24	17:00	17:30
25	17:30	18:00
26	18:00	18:30
27	18:30	19:00
28	19:00	19:30
29	19:30	20:00
30	20:00	20:30

t	from	to
31	20:30	21:00
32	21:00	21:30
33	21:30	22:00
34	22:00	22:30
35	22:30	23:00
36	23:00	23:30
37	23:30	24:00

Table B.1: *Encoding of the representative $T = 37$ periods of time t in which days have been split.*

Appendix C

Codes

```
###
# Chapman-Kolmogorov equations
# Example for arrivals
###
f.equations.arr <- function(time, P, parms)
{
lam <- lambda.arr[floor(time[1])]
mu <- parms

P0 <- P[1]; P1 <- P[2]; P2 <- P[3]; P3 <- P[4]; P4 <- P[5]; P5 <- P[6];
     P6 <- P[7];
P7 <- P[8]; P8 <- P[9]; P9 <- P[10]; P10 <- P[11]; P11 <- P[12]; P12 <-
     P[13]; P13 <- P[14];
P14 <- P[15]; P15 <- P[16]; P16 <- P[17]; P17 <- P[18]; P18 <- P[19];
    P19 <- P[20]; P20 <- P[21]

dP0 <- -lam*P0 + 2*mu*P1
dP1 <- -(lam+2*mu)*P1+2*mu*P2
dP2 <- lam*P0 -(lam+2*mu)*P2+2*mu*P3
```

```
dP3 <- lam*P1 - (lam+2*mu)*P3 + 2*mu*P4
dP4 <- lam*P2 - (lam+2*mu)*P4 + 2*mu*P5
dP5 <- lam*P3 - (lam+2*mu)*P5 + 2*mu*P6
dP6 <- lam*P4 - (lam+2*mu)*P6 + 2*mu*P7
dP7 <- lam*P5 - (lam+2*mu)*P7 + 2*mu*P8
dP8 <- lam*P6 - (lam+2*mu)*P8 + 2*mu*P9
dP9 <- lam*P7- (lam+2*mu)*P9 + 2*mu*P10
dP10 <- lam*P8 - (lam+2*mu)*P10 + 2*mu*P11
dP11 <- lam*P9 - (lam+2*mu)*P11 + 2*mu*P12
dP12 <- lam*P10 - (lam+2*mu)*P12 + 2*mu*P13
dP13 <- lam*P11 - (lam+2*mu)*P13 + 2*mu*P14
dP14 <- lam*P12 - (lam+2*mu)*P14 + 2*mu*P15
dP15 <- lam*P13 - (lam+2*mu)*P15 + 2*mu*P16
dP16 <- lam*P14 - (lam+2*mu)*P16 + 2*mu*P17
dP17 <- lam*P15 - (lam+2*mu)*P17 + 2*mu*P18
dP18 <- lam*P16 - (lam+2*mu)*P18 + 2*mu*P19

dP19 <- lam*P17 - 2*mu*P19 + 2*mu*P20
dP20 <- lam*P18 - 2*mu*P20

list(c(dP0,dP1,dP2,dP3,dP4,dP5,dP6,dP7,dP8,dP9,dP10,dP11,dP12,dP13,dP14
    ,dP15,dP16,dP17,dP18,dP19,dP20))
}

t1 <- seq(from=1,to=2, by=1/60)
time.period <- matrix(NA, nrow=37,ncol=61)
for(h in 1:37)
{
time.period[h,] <- seq(from=h,to=h+1,by=1/60)
}
```

```
library(deSolve)
for(w in 0:10)  # w is the queue length
{
P.init <- rep(0,21)
boom <- (2*w)+1
P.init[boom] <- 1
for(k in seq(0,10,by=1))   # k is the service rate for arrivals
{
for(h in 1:37)  # h is the time period
{
out <- ode(times=time.period[h,], y=P.init, func=f.equations.arr, parms
    =k)
keep <- out[which(out[,1]%in%seq(1,38,by=1)),]
rm(out)
save(keep,file=paste("P",paste(h,paste(".m",paste(w,paste(".a",k,sep=""
    ),sep=""),sep=""),sep=""),sep=""))
}
}
}

###
# Queue length Transition probabilites matrices
# Example for arrivals
###
mu.arr <- c(0:10)

now <- timestamp()
for(t in 1:36)
{
for(a in mu.arr)
{
```

```
Q <- matrix(NA,11,11)
rownames(Q) <- c("m0","m1","m2","m3","m4","m5","m6","m7","m8","m9","m10
    ")
colnames(Q) <- c("n0","n1","n2","n3","n4","n5","n6","n7","n8","n9","n10
    ")
for(m in 0:10)
{
for(n in 0:10)
{
load(paste("P",paste(t,paste(".m",paste(m,paste(".a",a,sep=""),sep=""),
    sep=""),sep=""),sep=""))
if(n==0){
foo <- keep[,-1][2,1]
} else {
foo <- keep[,-1][2,2*n]+keep[,-1][2,(2*n)+1]
}
Q[m+1,n+1] <- foo
}
}
save(Q, file=paste("Q",paste(t,paste(".a",a,sep=""),sep=""),sep=""))
}
}

###
# OTE estimation B-spline construction
###
library(splines)
kn <- c(0,4,6)
bspline.imc <- bs(xi,knots=kn,degree=3)

lm1 <- lm(yi~0+bspline.imc)
```

```
betasi <- lm1$coefficients
betasi

xxi <-seq(min(xi),max(xi),1/50)

library(quadprog)
n <- length(betasi)
Dmat <- diag(n)
dvec <- betasi
D1 <- matrix(0,nrow=n-1,ncol=n)
D2 <- matrix(0,nrow=n-2,ncol=n)
D1[cbind(1:(n-1),1:(n-1))] <- -1
D1[cbind(1:(n-1),2:n)] <- 1
D1
D2[cbind(1:(n-2),1:(n-2))] <- 1
D2[cbind(1:(n-2),2:(n-1))] <- -2
D2[cbind(1:(n-2),3:n)] <- 1

Amat <- matrix(0,nrow=nrow(D1)+nrow(D2),ncol=ncol(D1))
Amat[1:nrow(D1),] <- -D1
Amat[(nrow(D1)+1):nrow(Amat),] <- -D2
Amat <- t(Amat)
bvec <- rep(0,ncol(Amat))

ri <- solve.QP(Dmat,dvec,Amat,bvec)

# verify constraints
soli <- ri$solution

for(i in 2:length(soli))
{
```

```
print(isTRUE(soli[i] <= soli[i-1]))
}

for(i in 3:(length(soli)))
{
print(isTRUE(soli[i] -2*soli[i-1] + soli[i-2] <= 0))
}



ri$solution
bspline.test.imc <- bs(xxi,knots=kn,degree=3) # nuova matrice disegno
fun.imc=bspline.test.imc%*%ri$solution



# Definition of cost-to-go functions
CTG.last <- as.data.frame(matrix(cbind(rep(seq(0,10,by=1),11),rep(seq
    (0,10,1),each=11),rep(NA,11*11)),ncol=3))
names(CTG.last) <- c("a","d","v")

rho <- 1.2

for(i in 1:nrow(CTG.last))
{
CTG.last$v[i] <- rho*(CTG.last$a[i]^2) + CTG.last$d[i]^2
}



CTG <- as.data.frame(matrix(NA,nrow=11*11*2, ncol=17))
colnames(CTG)[1:4] <- c("a.prev","d.prev","w.now","cost.now")

j <- 0
```

```
for(r in 5:15)
{
colnames(CTG)[r] <- paste("ExpCost.a",j,sep="")
j <- j+1
}

colnames(CTG)[16:17] <- c("MinCost","mustar.a")

CTG[,1] <- rep(seq(0,10,by=1),length.out=nrow(CTG))
CTG[,2] <- rep(seq(0,10,by=1),each=11,length.out=nrow(CTG))
CTG[,3] <- c(rep(0,121),rep(1,121))   # w=0 => VMC   w=1 => IMC

for(i in 1:nrow(CTG))
{
CTG$cost.now[i] <- rho*(CTG$a.prev[i]^2) + CTG$d.prev[i]^2
}

V_37 <- as.data.frame(matrix(NA,nrow=11*11*2, ncol=15))
colnames(V_37)[1:4] <- c("a.36","d.36","w.37","cost.37")

j <- 0
for(r in 5:15)
{
colnames(V_37)[r] <- paste("ExpCost.a",j,sep="")
j <- j+1
}

V_37[,1] <- rep(seq(0,10,by=1),length.out=nrow(V_37))
V_37[,2] <- rep(seq(0,10,by=1),each=11,length.out=nrow(V_37))
V_37[,3] <- c(rep(0,121),rep(1,121))
```

```
for(i in 1:nrow(V_37))
{
V_37$cost.37[i] <- rho*(V_37$a.36[i]^2) + V_37$d.36[i]^2
}

V_37 <- rbind(V_37,c(rep(NA,4),seq(0,10,1)))

rate <- 1
for(k in 5:15)
{
for(i in 1:(nrow(V_37)-1))
{
a.37 <- max(0,V_37$a.36[i] + lambda.arr[37] - mu.arr[rate])

if(V_37$w.37[i]==0) {
mu.dep <- OTE.vmc(mu.arr[rate])
} else {
mu.dep <- OTE.imc(mu.arr[rate])
}

if(is.na(mu.dep)| a.37 > 10) {  # 10=N
V_37[i,k] <- NA
} else {
d.37 <- max(0,V_37$d.36[i] + lambda.dep[37] - mu.dep)
V_37[i,k] <- CTG.last$v[which(CTG.last$a==a.37&CTG.last$d==d.37)] + V_
    37$cost.37[i]
}

}
rate <- rate+1
}
```

```
# lim0 is the maximum reachable arrival service rate under VMC
# lim1 is the maximum reachable arrival service rate under IMC

for(i in 1:121)
{
abc <- which.min(V_37[i,c(5:(lim +5))])
ex <- attr(abc,"names")
V_37$MinCost[i] <-  V_37[i,ex]
V_37$mustar.a[i] <- V_37[nrow(V_37),ex]
if(V_37$w.37[i]==0) {
V_37$mustar.d[i] <- OTE.vmc(V_37$mustar.a[i])
} else {
V_37$mustar.d[i] <- OTE.imc(V_37$mustar.a[i])
}
}

for(i in 122:242)
{
abc <- which.min(V_37[i,c(5:(lim1+5))])
ex <- attr(abc,"names")
V_37$MinCost[i] <-  V_37[i,ex]
V_37$mustar.a[i] <- V_37[nrow(V_37),ex]
if(V_37$w.37[i]==0) {
V_37$mustar.d[i] <- OTE.vmc(V_37$mustar.a[i])
} else {
V_37$mustar.d[i] <- OTE.imc(V_37$mustar.a[i])
}
}

t <- 36
```

```
rate <- 1
future <- V_37

now <- timestamp()
for(k in 5:15)
{
for(i in 1:(nrow(CTG)-1))
{
if(CTG$w.now[i]==0) {
mu.dep <- OTE.vmc(mu.arr[rate])
} else {
mu.dep <- OTE.imc(mu.arr[rate])
}

if(is.na(mu.dep)) {mu.dep <- 0}

load(paste("Q",paste(t,paste(".a",mu.arr[rate],sep=""),sep=""),sep=""))
row.a <- CTG$a.prev[i]+1
arr <- Q[row.a,]

load(paste("Q",paste(t,paste(".d",mu.dep,sep=""),sep=""),sep=""))
row.d <- CTG$d.prev[i]+1
dep <- Q[row.d,]

load(paste("Weather.trans.t",t,sep=""))
if(CTG$w.now[i]==0) {W <- Weath.trans[1,]} else {W <- Weath.trans[2,]}

vstar <- c(NA, length=242)
go <- 1
# h <- 1  # for weather VMC -> VMC  or IMC -> VMC
# h <- 2  # for weather VMC -> IMC  or IMC -> IMC
```

```
for(h in 1:2)
{
for(g in 1:11) # g for departure queue length
{
for(m in 1:11)  # i for arrival queue length
{
P <- c(NA, length=11)
P[m] <- arr[m]*dep[g]*W[h]
vstar[go] <- P[m]*future$MinCost[go]
go <- go+1
}
}
}
exit <- sum(vstar)
CTG[i,k] <- exit
}
rate <- rate+1
}

for(i in 1:121)
{
abc <- which.min(CTG[i,c(5:(lim0+5))])
ex <- attr(abc,"names")
CTG$MinCost[i] <-  CTG[i,ex] + CTG$cost.now[i]
CTG$mustar.a[i] <- CTG[nrow(CTG),ex]
if(CTG$w.now[i]==0) {
CTG$mustar.d[i] <- OTE.vmc(CTG$mustar.a[i])
} else {
CTG$mustar.d[i] <- OTE.imc(CTG$mustar.a[i])
}
}
```

```
for(i in 122:242)
{
abc <- which.min(CTG[i,c(5:(lim1+5))])
ex <- attr(abc,"names")
CTG$MinCost[i] <-  CTG[i,ex] + CTG$cost.now[i]
CTG$mustar.a[i] <- CTG[nrow(CTG),ex]
if(CTG$w.now[i]==0) {
CTG$mustar.d[i] <- OTE.vmc(CTG$mustar.a[i])
} else {
CTG$mustar.d[i] <- OTE.imc(CTG$mustar.a[i])
}
}

for(t in seq(35,1,by=-1))
{
rate <- 1
for(k in 5:15)
{
for(i in 1:(nrow(CTG)-1))
{
if(CTG$w.now[i]==0) {
mu.dep <- OTE.vmc(mu.arr[rate])
} else {
mu.dep <- OTE.imc(mu.arr[rate])
}

if(is.na(mu.dep)) {mu.dep <- 0}

load(paste("Q",paste(t,paste(".a",mu.arr[rate],sep=""),sep=""),sep=""))
row.a <- CTG$a.prev[i]+1
```

```
        arr <- Q[row.a,]

        load(paste("Q",paste(t,paste(".d",mu.dep,sep=""),sep=""),sep=""))
        row.d <- CTG$d.prev[i]+1
        dep <- Q[row.d,]

        load(paste("Weather.trans.t",t,sep=""))
        if(CTG$w.now[i]==0) {W <- Weath.trans[1,]} else {W <- Weath.trans[2,]}

        vstar <- c(NA, length=242)
        go <- 1
        for(h in 1:2)
        {
        for(g in 1:11)
        {
        for(m in 1:11)
        {
        P <- c(NA, length=11)
        P[m] <- arr[m]*dep[g]*W[h]
        vstar[go] <- P[m]*future$MinCost[go]
        go <- go+1
        }
        }
        }
        exit <- sum(vstar)
        CTG[i,k] <- exit
        }
        rate <- rate+1
        }

        for(i in 1:121)
```

```
{
abc <- which.min(CTG[i,c(5:(lim0+10))])
ex <- attr(abc,"names")
CTG$MinCost[i] <-  CTG[i,ex] + CTG$cost.now[i]
CTG$mustar.a[i] <- CTG[nrow(CTG),ex]
if(CTG$w.now[i]==0) {
CTG$mustar.d[i] <- OTE.vmc(CTG$mustar.a[i])
} else {
CTG$mustar.d[i] <- OTE.imc(CTG$mustar.a[i])
}
}

for(i in 122:242)
{
abc <- which.min(CTG[i,c(5:(lim1+10))])
ex <- attr(abc,"names")
CTG$MinCost[i] <-  CTG[i,ex] + CTG$cost.now[i]
CTG$mustar.a[i] <- CTG[nrow(CTG),ex]
if(CTG$w.now[i]==0) {
CTG$mustar.d[i] <- OTE.vmc(CTG$mustar.a[i])
} else {
CTG$mustar.d[i] <- OTE.imc(CTG$mustar.a[i])
}
}
}
```

Bibliography

Arias, P. et al. (2013). "A Methodology Combining Optimization and Simulation for Real Applications of the Stochastic Aircraft Recovery Problem". In: *2013 8th EUROSIM Congress on Modelling and Simulation*, pp. 265–270.

Azzalini, Adelchi and Bruno Scarpa (2012). *Data analysis and data mining*. Oxford University Press.

Bard, Jonathan, Gang Yu, and Michael Argüello (2001). "Optimizing Aircraft Routings in response to Groundings and Delays". In: *IIE Transactions* 33.10, pp. 931–947.

Baspinar, Baris, Emre Koyuncu, and Gokhan Inalhan (2017). "Large Scale Data-Driven Delay Distribution Models of European Air Traffic Flow Network". In: *Transportation Research Procedia* 22, pp. 499–508.

Baspinar, Baris et al. (2016). "Analysis of Delay Characteristics of European Air Traffic through a Data-Driven Airport-Centric Queuing Network Model". In: *IFAC PapersOnLine* 49.3, pp. 359–364.

Baum, LE et al. (1970). "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains". In: *Ann. Math. Statist.* 41, p. 164.

Cai, Kai-Quan et al. (2017). "Simultaneous Optimization of Airspace Congestion and Flight Delay in Air Traffic Network Flow Management". In: *Intelligent Transportation Systems, IEEE Transactions on* 18.11, pp. 3072–3082.

Cappé, Olivier, Eric Moulines, and Tobias Rydén (2005). *Inference in hidden Markov models*. Springer.

De Boor, C. (2001). *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York.

De Neufville, R and Amedeo R. Odoni (2013). *Airport Systems: Planning, Design, and Management, 2d ed.* McGraw-Hill Education.

Dempster, AP, NM Laird, and DB Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *J. R. Statist. Soc. Ser. B* 39, p. 1.

Dymarski, P. (2011). *Hidden Markov Models, Theory and Applications*. Rijeka, HR, Intech, pp. 207–222.

ENAV S.p.A. URL: https://www.enav.it/sites/public/it/Home.html.

EUROCONTROLl. URL: https://www.eurocontrol.int/.

Fiorentini, Gabriele, Enrique Sentana, and Giorgio Calzolari (2003). "Maximum Likelihood Estimation and Inference in Multivariate Conditionally Heteroscedastic Dynamic Regression Models With Student t Innovations". In: *Journal of Business & Economic Statistics* 21.4, pp. 532–546. DOI: 10.1198/073500103288619232. eprint: http://dx.doi.org/10.1198/073500103288619232. URL: http://dx.doi.org/10.1198/073500103288619232.

Frühwirth-Schnatter, Sylvia (2006). *Finite Mixture and Markov Switching Models: Modeling and Applications to Random Processes*. Springer.

Gross, Donald and Carl Harris (1999). "Fundamentals of queueing theory (3rd ed.)" eng. In: *Journal of the American Statistical Association* 94.445, pp. 348–348. ISSN: 0162-1459. URL: http://search.proquest.com/docview/38731994/.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001). *The elements of statistical learning : data mining, inference, and prediction*. eng. Springer series in statistics. New York ; Barcelona [etc.: : Springer. ISBN: 0387952845.

Ivanov, Nikola et al. (2017). "Air Traffic Flow Management slot allocation to minimize propagated delay and improve airport slot adherence". In: *Transportation Research Part A* 95.C, pp. 183–197.

Jacquillat, Alexandre and Amedeo R. Odoni (2015). "Endogenous control of service rates in stochastic and dynamic queuing models of airport congestion". In: *Transportation Research Part E* 73.C, pp. 133–151.

Jacquillat, Alexandre, Amedeo R. Odoni, and Mort D. Webster (2017). "Dynamic Control of Runway Configurations and of Arrival and Departure Service Rates at JFK Airport Under Stochastic Queue Conditions." eng. In: *Transportation Science* 51.1, pp. 155–177.

McLachlan, Geoffrey and Thriyambakam Krishnan (2007). *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

McLachlan, GJ and D Peel (2000). *Finite mixture models*. John Wiley & Sons.

OAG. URL: https://www.oag.com/.

Peel, D. and G. J. McLachlan (2000). "Robust mixture modelling using the t distribution". In: *Statistics and Computing* 10.4, pp. 339–348. ISSN: 1573-1375. DOI: 10.1023/A:1008981510081. URL: http://dx.doi.org/10.1023/A:1008981510081.

Platinum Airways. URL: https://www.platinumairways.org/.

Polson, Nicholas G., James G. Scott, and Jesse Windle (2013). "Bayesian inference for logistic models using Pólya-Gamma latent variables". In: *J. Amer. Statist. Assoc.* 108.504, pp. 1339–1349. ISSN: 0162-1459. DOI: 10.1080/01621459.2013.829001. URL: https://doi.org/10.1080/01621459.2013.829001.

Pyrgiotis, Nikolas, Kerry M. Malone, and Amedeo Odoni (2011). "Modelling delay propagation within an airport network". In: *Transportation Research Part C* 27.

SAVE S.p.A. URL: https://www.grupposave.it/.

Simaiakis, I. (2012). "Analysis, Modeling and Control of the Airport Departure Process". PhD thesis. Massachusetts Institute of Technology.

Simaiakis, Ioannis et al. (2014). "Demonstration of reduced airport congestion through pushback rate control". eng. In: *Transportation Research Part A* 66.1, pp. 251–267.

Wood, Simon N. (2006). *Generalized additive models.* Texts in Statistical Science Series. An introduction with *fR.* Chapman & Hall/CRC, Boca Raton, FL, pp. xviii+391. ISBN: 978-1-58488-474-3; 1-58488-474-6.

Zucchini, Walter and Iain L MacDonald (2009). *Hidden Markov models for time series: an introduction using R.* CRC Press.