

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Analysis of the Route-Based Aggregate Model for Strategic Air Traffic Control

Permalink

<https://escholarship.org/uc/item/5rj4p7w7>

Author

De Los Santos Bernad, Victor

Publication Date

2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Analysis of the Route-Based Aggregate Model for Strategic Air Traffic Control

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

In Mechanical and Aerospace Engineering

by

Victor De Los Santos Bernad

Thesis Committee:
Professor Kenneth D. Mease, Chair
Professor Faryar Jabbari
Professor Wenlong Jin

2014

DEDICATION

To my family and everyone that has supported me during this Masters.

You know who you are.

TABLE OF CONTENTS

	Page
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
ACKNOWLEDGEMENTS	viii
ABSTRACT OF THE THESIS	ix
1 Introduction	1
2 Large-Capacity Cell Transmission and Link Transmission Models vs Route-Based Aggregate Model	4
2.1 Previous Models	4
2.1.1 Continuous-time models.....	5
2.1.2 Discrete-time models.....	5
2.2 Motivations for the RBAM before the CTM and LTM	6
2.3 The Large-Capacity Cell Transmission Model.....	7
2.3.1 Modelling the National Airspace System	8
2.3.2 The Dynamics of the CTM(L)	10
2.3.3 CTM(L) optimization problem.....	11
2.4 The Link Transmission Model	12
2.4.1 Dynamics of the LTM	13
2.4.2 The LTM optimization problem	15

2.5	The Route-Based Aggregate Model	15
2.5.1	Modeling the National Airspace System with the RBAM	16
2.5.2	The dynamics of the RBAM.....	17
2.5.3	The optimization problem for the RBAM	18
2.6	CTM(L) vs LTM vs RBAM.....	21
2.6.1	Differences in the NAS modelling	22
2.6.2	Differences in the dynamics.....	23
2.6.3	Differences in the optimization problem.....	24
3	Implementation of the model in Matlab: A Ground delay and rerouting example	25
3.1	Creating the model in Matlab	25
3.2	Definition of the example	26
3.3	Inputs.....	28
3.3.1	Flight information	28
3.3.2	Sectors' information	29
3.3.3	Routes information	30
3.3.4	Airports information	31
3.3.5	Other inputs.....	31
3.4	System matrices	33
3.4.1	Obtaining A and B	33
3.4.2	Classification of the flights.....	34

3.4.3	Forced input $s(t)$	35
3.5	Cost function	36
3.6	Constraints	37
3.6.1	Maximum Sector Capacity constraint.....	38
3.6.2	No early departures constraint.....	39
3.6.3	No more delays than flights constraint:	40
3.6.4	Airport maximum departures constraint.....	40
3.6.5	Maximum cancelled flights constraint.....	41
3.6.6	Non-controllable origins constraint	42
3.7	Solving the optimization problem.....	42
3.7.1	Solving the problem with Matlab	42
3.7.2	Example 1.....	44
3.7.3	Example 2.....	45
3.7.4	Example 3.....	47
4	Calibration of the Cost Function	49
4.1	Overlook of the rerouting weighting value	49
4.2	Cost of the flights delays and rerouting.....	49
4.3	Example with real values for the rerouting weighting value	51
5	Conclusions	54
6	Bibliography	56

LIST OF FIGURES

	Page
Figure 1: Overview of Lagrangian vs Eulerian applied for Air Traffic Control	2
Figure 2: Examples of vertices and links in the CTM(L) (14)	9
Figure 3: Aggregation of cells into Links	12
Figure 4: Minimum cancelled flights algorithm	20
Figure 5: Overview of the Matlab sequential process	25
Figure 6: ABC example	27
Figure 7: Flight classification algorithm	36
Figure 8: $u(t)$ for the different routes in example 3	48
Figure 9: Two route example to see the effects of KR	52

LIST OF TABLES

	Page
Table 1: Summary of CTM(L), LTM and RBAM.....	21
Table 2: Flight information table	29
Table 3: Sector Information	29
Table 4: Routes information	30
Table 5: Airports information	31
Table 6: Boundary Cells Table	34
Table 7: Ground delays and rerouting for $K_{RR} = 1$	45
Table 8: Ground delays and rerouting for $K_{RR} = 10$	46
Table 9: Results for $KRR = 1.79$	53

ACKNOWLEDGEMENTS

I would like to start thanking Professor Kenneth D. Mease for all the help, support and guidance during the development of this project.

Thanks to NASA Ames Research Center for funding this project. My kindest gratitude to Lluís Soler and Alessandro Bombelli, as their help was invaluable to understand the project and help me move forward.

Last but not least, my eternal gratitude to Pete J. Balsells, Professor Roger H. Rangel and the Balsells Fellowship for the life changing opportunity of pursuing this Master of Science in the University of California, Irvine.

ABSTRACT OF THE THESIS

Analysis of the Route-Based Aggregate Model for Strategic Air Flow Control

By

Victor De Los Santos Bernad

Master of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2014

Professor Kenneth D. Mease, Chair

Because of the vital importance of the National Airspace System (NAS) and its diagnosed growth over the next years, the planning and prediction at a strategic phase of the Traffic Flow Management (TFM) proves to be a difficult task but a useful tool to reduce the airspace congestion. Research has led to the creation of several models in order to address this challenge. Because of the complexity of the problem, the Eulerian (aggregate) approach may be the best to reduce the dimension and complexity of the problem, whilst maintaining accuracy.

This study analyzes one of the latest aggregate models created, the Route-Based Aggregate Model (RBAM), and compares it to the Large-Capacity Cell Transmission Model (CTM(L)) and the Link Transmission Model (LTM). These three models share some similarities such as the non-existence of diverging nodes or, in the case of the CTM(L), the condition of submitting all the airplanes in a cell to the next cell after one time-step. But there are also big differences which make them different enough to coexist. For example, the RBAM can be used without the need of historical data in order to model the NAS, only the information of the upcoming flight plans.

Also, the RBAM is designed to base its controls from a ground perspective, allowing ground rerouting and ground delay.

An explanation on how to implement the RBAM in Matlab can be found in this project, explaining the peculiarities of the translation of the cost function constraints into a Linear Programming (LP) problem, with several examples that show how the solution to the LP problem distributes the delays between ground delays and ground reroutings. Because the cost of a ground rerouting is different from the cost of a ground delay because of the extra fuel expense that the rerouting may cause (assuming always that the original route is shorter), a proper weighting of both controls is found, considering different variables such as the cost of the fuel or the cost of overtime parking at the airports for the delayed aircraft. Future research will study how to define the alternative routes for the ground rerouting and also how to implement airborne rerouting.

1 Introduction

Civil aviation is unarguably of vital importance for the economic development and growth of the US. Despite history changing events such as the impact of the global economic recession, the skyrocketing prices for fuel and the terrorist attacks of September 11, civil aviation has found its way to not only survive but also grow in both profitability and size. There is an expected growth of an averaged 2.2% for the next 20 years (1). For an industry which represents more than a 5% of the national GDP (2), the impact in the US economy and development cannot be overlooked.

This increase comes along with obvious benefits, but also with concerns as the already congested airspace will be even more congested. This congestion implies delays which at the end represents a huge cost of money. Only in 2013, the total cost of the delays for U.S. Airlines was of \$8,064 million, including the costs of fuel, crew, maintenance, aircraft ownership and other handling fees (3). Therefore it is completely necessary to provide the Air Traffic controllers of the National Airspace System (NAS) with conflict detection and resolution tools.

The strategic phase of Air Traffic Management is critical for the proper prediction of the state of the Airspace and how to control it. In Air Traffic Flow Management, this phase is about 30 minutes to 8 hours before the scheduled departures. At that point it is possible to simulate how the NAS will be affected by not only the flights but also external conditions which have a

major impact in the airspace such as weather. It is therefore of major interest to find ways of not only simulating the NAS but also to be able to optimize the system.

Comparing the NAS modelling problem to a fluid mechanics problem, there are two possible ways of specifying the flow field: Lagrangian and Eulerian. In the Lagrangian model, the system is observed following the fluid throughout the whole flow field. In a Eulerian model, the observer will only pay attention to an specific point of the whole field where the fluid flows as the time passes. Translating this into the NAS modelling problem, the Lagrangian model implies following the behaviour of each aircraft inside the system, whilst the Eulerian model will observe the flow of aircraft at a specific point of the air space as time goes by.

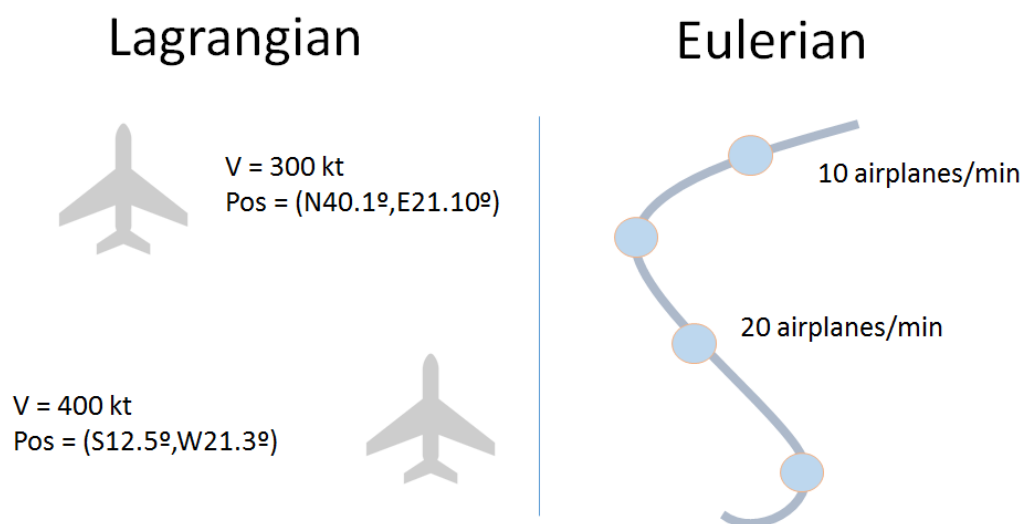


Figure 1: Overview of Lagrangian vs Eulerian applied for Air Traffic Control

On an average day, more than 28.000 commercial flights fill the US airspace (4). This means that, in a Lagrangian model, more than 28.000 variables would be observed. With the current computational power it is possible to not only model but also predict possible delays from a Lagrangian point of view (5). Adding control though would be an impossible task because of

the amount of possibilities when it comes to finding the optimal trajectories of each of the aircraft of the system extended in time (at least 3 ordinary differential equations would need to be integrated per each aircraft). This is impractical as frequent and periodic optimal control solutions would be computed. Instead, the common trend is to use an Eulerian approach, where the NAS is discretized in several ways and then static points of it are observed. This not only reduces considerably the system but also the optimal control problem, as a linear-system problem is faced. Depending on the model the size of the system and the amount of variables to control may change. These models are usually called Aggregate models, as the information that each discretized part of the model provides is the aggregation of all the aircraft within it. Several models have been created following this Eulerian approach which will be discussed shortly in the following section.

2 Large-Capacity Cell Transmission and Link Transmission Models vs Route-Based Aggregate Model

2.1 Previous Models

An analysis of the different models has already been conducted (6), but it is necessary to mention them again as two important models were omitted. These two models contain several similarities to the model being discussed in this work, so a deep understanding of them is compulsory, and that includes studying the previous work done that led to those models. First of all, it is necessary to state that the models for Air Traffic Flow management are inspired by the work done for road transportation (7). Therefore, the dynamics and assumptions of the models are similar. All these models discretize the airspace into cells where the aircraft are aggregated, losing their identity and the ability to track each flight individually. The different models define how the cells are interconnected and the flow of aircraft between them. Thus, the state of the system $x(t)$ will be the amount of aircraft in each cell and it will not be possible for a cell to transmit more aircraft than its state indicates it has to another cell. A first way of categorizing the models is by differentiating between continuous-time and discrete-time models.

2.1.1 Continuous-time models

The two major models that consider a continuous-time problem are the *Partial Derivative Equation model* (PDE model) and the *Time Continuous model* (8).

- **PDE model** (9): derived from the *Lighthill-Whitham-Richards model* for ground transportation, discretizes the airspace in paths (air routes) and models each as a partial derivative equation, where the speed of the aggregated aircraft depends only on the cell where the aircraft are. The control is applied in the velocity profile.
- **Time-Continuous model** (8): This is a simplification of the PDE model, by assuming a constant speed inside each cell, transforming a system of PDEs into an ODEs system. The way of controlling the air traffic flow in this model is by rerouting.

2.1.2 Discrete-time models

Most of the models developed follow a discrete-time problem approach, including the Route-Based Aggregate Model. These models are the *Modified Menon Model* (MMM), the *Eulerian Flow Model* (EFW), the *Linear Dynamic Systems Model* (LDSM), the *Large-Capacity Cell Transmission Model* and the *Link Transmission Model*. The last two models were not considered during the creation of the RBAM, and therefore they will be studied with more detail in the next section. A brief description of the other models follows:

- **Modified Menon Model (10):** The airspace is discretised in 1D cells, where each cell can transmit to more than one cell (creating diverging nodes). To optimize the ATF this model provides ground and air delay control.

- **Eulerian Flow Model (11):** A modification of the MMM, by transforming the cells into 2D elements which can transmit to up to 8 neighbour cells. This implies up to 64 divergence parameters. The control allows flow metering and ground delay.

- **Linear Dynamic Systems Model (12):** This model discretizes the Air Traffic System in 2D cells, which can either receive or send aircraft from the two neighbor cells. This model, which was the principal point of departure for the RBAM, features ground delay and rerouting as controls.

2.2 Motivations for the RBAM before the CTM and LTM

At the time of creation of the Route-Based Aggregate Model, the idea was to address two main problems: eliminating the presence of fractional terms caused by the transmission of only some of the aircraft from a cell to the next cell in the next time step and the presence of diverging nodes and multiple possibilities of transmission per each cell. To solve these issues, two design decisions were taken:

- The cells will be 1D and will only transmit to at most one cell. This prevents the appearance of diverging nodes and creates a time-invariant state transition matrix.
- All the cells in the model would have the same transition time and the speed of all the aircraft would be fixed and equal, so the cells would be placed in a way that at every time step all the cells would transmit the whole aggregation of aircraft in it to the next cell. The idea behind this is making the Courant-Friederichs-Lewy (CFL) number $CFL = 1$ (13).

The two discrete-time models remaining, the *Large-Capacity Cell Transmission Model* and the *Link Transmission Model* which will be presented now were created with the idea of solving most of the same issues the RBAM tried to. Because they were not considered at the time of creation of the RBAM it is necessary to study and understand them, so a thorough comparison can be made between the models, especially as they show certain similarities on a first glimpse (although it will be shown that there are several concept ideas which make them different enough to coexist).

2.3 The Large-Capacity Cell Transmission Model

Developed in 2008 by Sun and Bayen, the *Large-Capacity Cell Transmission Model* (CTM(L)) (14) constructs a traffic flow model by using *Enhanced Traffic Management System* (ETMS) and *Aircraft Situation Display to Industry* (ASDI) air traffic data. The name comes from the *Cell*

Transmission Model in highway traffic (15). This model discretises the airspace first in links connecting sectors and then these links into equally spaced cells. To obtain the links, the historical data from the ETMS/ASDI is processed, which includes the information of the position and altitude of all airborne aircraft in the US every minute.

2.3.1 Modelling the National Airspace System

First, the vertices of the links are found. With the airspace being distributed in sectors, every sector will have links connecting each pair of neighbours. Following Figure 2 as an example to understand how the links are created, the link of sector ZOA15 that connects the origin sector ZOA33 with the destination sector ZLA27 will be processed. It is important to point out that the links are not bidirectional, which means that the entry vertex is not the same as the departure vertex from the opposite direction link. The vertices are physically placed by averaging the boundary crossing points obtained from the historical flights data, for both the entrance and the exit of the sector.

Having the position of the vertex located, they are united by a straight line which is the link. To determine the time length of the link (how long do the aircraft stay in that link) the flights of a full year are computed. Then, the link is discretised by dividing the time length of the link with the time step desired, obtaining the amount of equally spaced cells of the link.

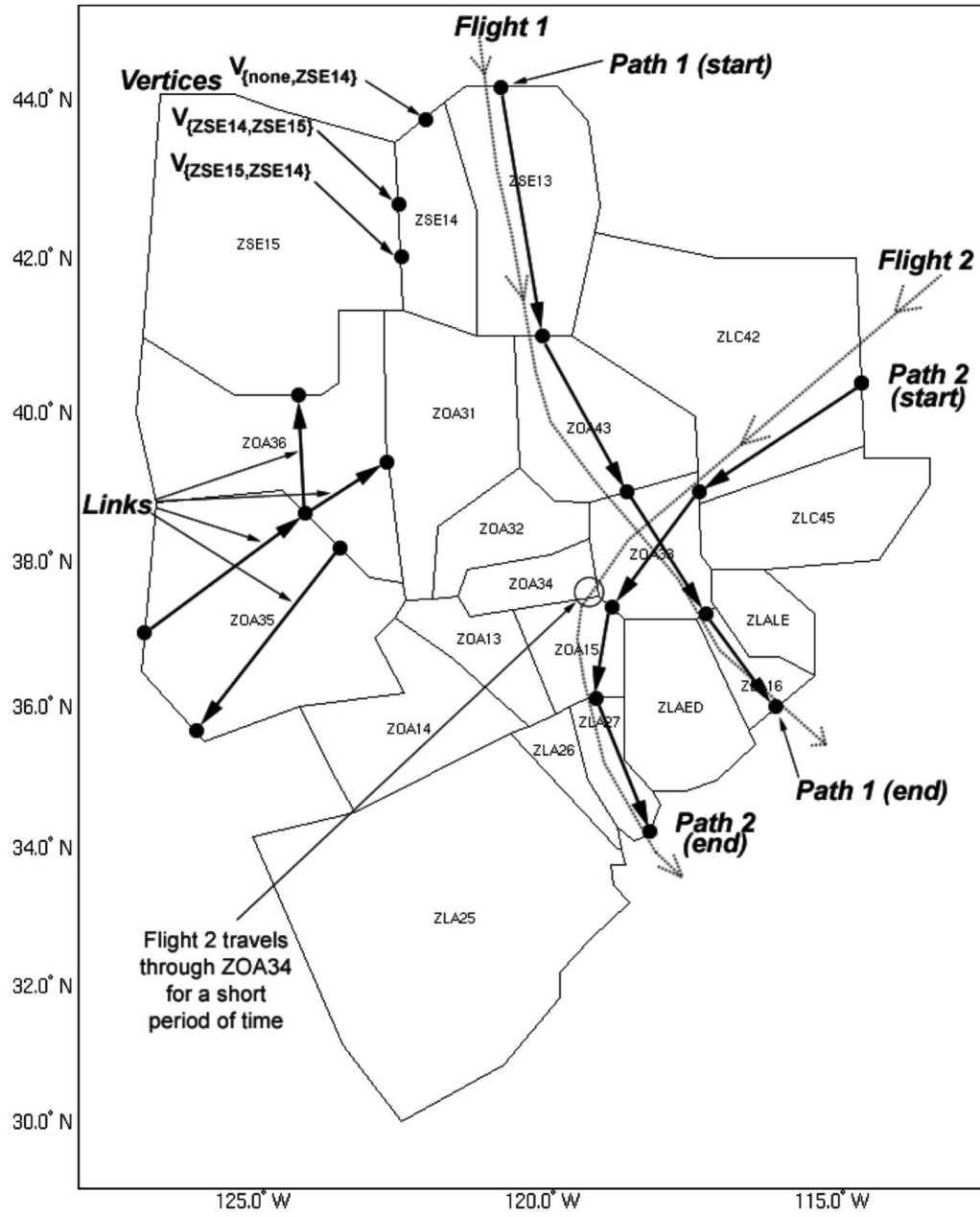


Figure 2: Examples of vertices and links in the CTM(L) (14)

2.3.2 The Dynamics of the CTM(L)

The dynamics of the system are defined as:

$$x_i(t+1) = A_i x_i(t) + B_i^u u_i(t) + B_i^f f_i(t) \quad (2.1)$$

i represents the link, x_i is the state of the system, represented as the number of airplanes in every cell at the time step (t), A_i is the system matrix which is a $m_i \times m_i$ (where m_i is the number of cells in the system) nilpotent matrix with ones on its superdiagonal. B_i^f is a $m_i \times 3$ matrix of the form $B_i^f = [B_i^{in}, B_i^{climb}, B_i^{desc}]$. They are the forcing input matrices, the climb input matrices and the descent input matrices respectively, and indicates the coordinates where the inputs will be in the system, represented as $f_i(t) = [f_i^{in}(t), f_i^{climb}(t), f_i^{desc}(t)]^T$. Finally B_i^u is the controlled input matrix, an $m_i \times m_i$ with ones on its diagonal and negative ones on its superdiagonal; and $u_i(t)$ is the control vector itself. This is a Linear Time Invariant system.

In this model, the way airplanes are added in the system is through the forced input vectors, and they can be added either at the beginning of a link in a boundary sector from the region under control, or at any cell inside the links from airplanes reaching the high-level altitude inside the system. The system also considers airplanes leaving the control zone by descending lower than 24.000 ft at a particular cell. The model considers primarily airborne delays. These delays always take place at the beginning of the time step and the control is applied in one time increment units. This delay in real life implies having an aircraft doing patterns over a region until it can continue with its route. In the Appendix A in (16), the model is extended to include airports by adding a vertex to model it, which would allow to have ground delay

also, but no examples have been given of the model with that kind of delay as for now. It is also commented in (16) that no rerouting is considered, but that it would be studied in future work.

2.3.3 CTM(L) optimization problem

To solve the optimization problem for a system the cost function of this model is stated as

$$\min_{x,u} \sum_{t=0}^T c^T x_t \quad (2.2)$$

This function encodes the minimization of the total travel time for all the flights in the NAS for the time horizon of interest. The constraints for this minimization problem ensure that accumulated departures cannot exceed the amount of schedule departures, that all scheduled flights will depart and that they will land at their destinations by end of the planning time horizon. It is also constrained the capacity for each sector and that for every cell the number of delay controlled aircraft cannot exceed the total number of aircraft in the cells.

The initial approximation to solve this problem was dual decomposition (16), but then in (17) a different approach was applied by using the total unimodularity property of the A matrix, which, considering that the problem is feasible, means there exists at least one integral optimum that can be found by using the *Dantzig-Wolfe Decomposition* method.

2.4 The Link Transmission Model

The *Link Transmission Model* is an evolution of the CTM(L) created by Cao and Sun (18). It is a higher-level approach with a reduced system size as it does not make the discretization of the links into cells. Thus, the way to obtain the model is the same as with the CTM(L), as the links are the same. In other words, as can be seen in Figure 3: Aggregation of cells into Links, the cells which have an aggregated count of aircraft are then aggregated into links, so the links will have the sum of all the cells and therefore all the aircraft in each sector following that route.

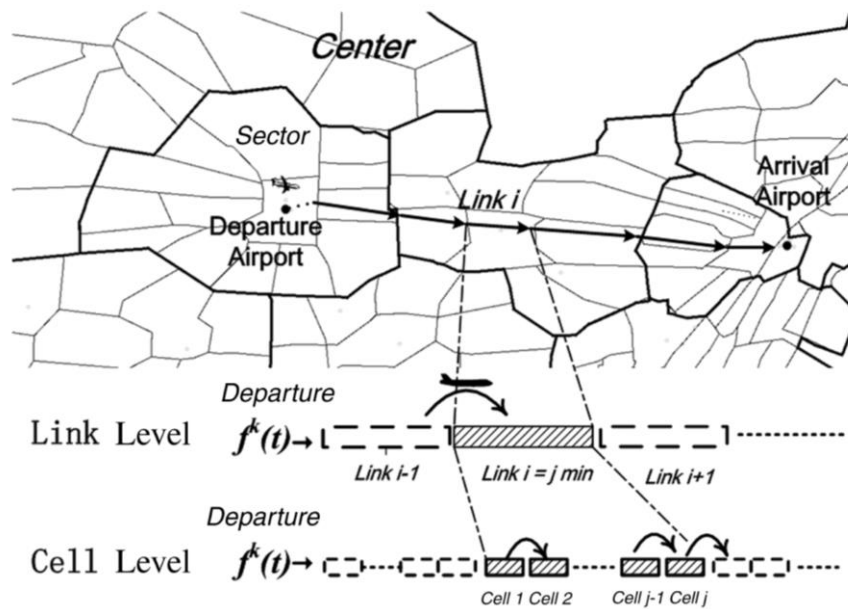


Figure 3: Aggregation of cells into Links

2.4.1 Dynamics of the LTM

The dynamics of this model is what makes the big difference with respect to the CTM(L). The dynamics are defined as:

$$X^k(t + 1) = X_1^k(t + 1) + X_2^k(t + 1) \quad (2.3)$$

Where

$$\begin{cases} X_1^k(t + 1) = A_1^k(t)X_1^k + B^k f^k(t) \\ X_2^k(t + 1) = A_2^k(t)X_2^k(t) \end{cases}$$

Being

$$A_1^k = \begin{bmatrix} [1 - \beta_0^k(t)] & & & \\ \beta_0^k(t) & [1 - \beta_1^k(t)] & & \\ & \ddots & \beta_{n^k-1}^k(t) & [1 - \beta_{n^k}^k(t)] \end{bmatrix}$$

$$B^k = [1 \ 0 \ \dots \ 0]$$

$$A_2^k = \begin{bmatrix} [1 - p_0^k(t)] & & & \\ p_0^k(t) & [1 - p_1^k(t)] & & \\ & \ddots & p_{n^k-1}^k(t) & [1 - p_{n^k}^k(t)] \end{bmatrix}$$

And

$$p_i^k(t) = \frac{\bar{\lambda}_i^k(t)}{\bar{x}_i^k(t)}$$

The initial dynamics equations is divided in two components: the deterministic component $X_1^k(t)$ and the probabilistic component $X_2^k(t)$. Therefore, this model adds a new dimension by inserting a probabilistic part. This probabilistic part is needed as, whenever there is an aircraft entering a link from outside the region at a random point of the link, the physical

location is unknown. The probabilistic terms are defined with p_i^k which describes the transmission in history of link i at time t , taken from historical traffic data, being $\bar{\lambda}_i^k(t)$ the average number of aircraft moving from link i to link $i+1$ and $\bar{x}_i^k(t)$ the average count in link i . It is not taken into account the influence of weather, as only good weather days are considered for these estimations.

A peculiarity of this model is that there is no control term. Instead, the transmission coefficients $\beta_i^k(t)$ are used to add control to the system. This coefficient indicates the amount of aircraft that stay in a link after a time step and how many of them move to the next link. Thus, altering this coefficient can control the flow rate between links, by keeping the aircraft which the system needs to delay in the link until the right time. This makes the system a non-linear time-variant problem, increasing the complexity of the system.

The type of control this method provides is both ground and airborne delay. Link 0 represents the departure queue in airports, and so by affecting the transmission coefficient for link 0 what it is being altered is the departures at the airports, so this would be ground delay. Variations in the other links transmission coefficients will represent airborne delay. The model already considers that the cost of airborne delay is higher than ground delay, being the latest preferred (a ratio of 1:2 between the cost of ground delay and airborne delay is exposed in (18)).

2.4.2 The LTM optimization problem

The link transmission optimization problem tries to minimize the objective function:

$$\min \sum_{t=0}^T \sum_{k=1}^K \sum_{i=0}^{n^k} c_i^k x_i^k(t) \quad (2.4)$$

c_i^k is a weight imposed in link i , and for $c_i^k = 1$ the route reflects the minimum total flight time in the planning time horizon. Basically this minimum cost function tries to minimize the weighted sum of all the state variables, which implies minimizing the total delays of the flights. Again, the constraints include sector constraints (no more flights in a sector than the maximum capacity of the sector), arrival constraints (all the flights land in their destination airports by the end of the planning time horizon) integer constraint and introduces a minimum dwell time constraint, which enforces a flight stay in a link for at least the link length, so the traffic flow is guaranteed to not move too fast. The method to solve this optimization problem is by using a Dual Decomposition Algorithm, as the scale of the problem is too large to be solved as a whole.

2.5 The Route-Based Aggregate Model

The model presented by Lluís Soler in (6), called Route-Based Aggregate model is an Eulerian (Aggregate system) that tries to solve some of the problems previous models had. Before stating the differences between CTM(L) and LTM, the model is presented.

2.5.1 Modeling the National Airspace System with the RBAM

The model discretizes the systems in routes. These routes are taken from the flight plans provided by the ASDI for the time and region that is going to be simulated and optimized. The point-to-point routes connect the origin-destination airport pairs. The routes are then divided into same transition time cell, being the transition time the same as the time step of the system; and the cells can only transmit to one neighbor cell (thus preventing from diverging nodes). Because it is sought to have a CFL = 1, the amount of cells in a route can be found by dividing the total distance of the route by the desired system time step times the velocity of the aircraft in the system (averaged). The cells are then assigned to a sector depending on their geographical location.

The flights are classified in:

- Internal flights: origin and destination airport in the region inside the boundaries of the system.
- Entering flights: only the destination airport is inside the boundaries of the system
- Exiting flights: only the origin airport is inside the boundaries of the system
- Transiting flights¹: neither the origin nor the destination airports are within the boundaries but the trajectory intersects the boundaries at some point.

The internal and exiting flights are assigned to the initial cell which is linked to the origin airport and the correspondent route. For the entering flights and transiting flights it is

¹ Alessandro Bombelli, Unpublished work on RBAM, University of California Irvine

necessary to calculate at what time they will enter the boundaries and then assign to the corresponding time step and cell in the system.

2.5.2 The dynamics of the RBAM

The dynamics of the RBAM are stated by:

$$x(t + 1) = Ax(k) + B(s(t) + u(t)) \quad (2.5)$$

$x(t)$ represents the states of the system, a vector of size n , where n is the number of cells of the system. A is the system transition matrix, of size $n \times n$. A is formed by different submatrices A_i representing each and every route of the system. The submatrix A_i is a nilpotent matrix with 1s on its subdiagonal:

$$A_i = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix}$$

where p is the number of routes of the system. Although originally A_i also had a 1 in the last element of its matrix (last row and last column) that would work as a counter of airplanes that arrived to its destination, it has been eliminated as it is not necessary for the system and increases the complexity of the system (specially at the time of calculating the amount of flights in a sector).

B is a constant matrix mapping the origin cells to the associated cells, of size $n \times p$. It multiplies both the forced input vector $s(k)$ and the control vector $u(k)$, both column vectors of size p . The system is prepared for ground delay and ground rerouting. Only in the origin cells aircraft can be added to the system, which are also the only places where there can be some control (explaining the reason B multiplies both $s(k)$ and $u(k)$). As it is intended to be an optimization problem at a strategic phase, ground delay is always going to be preferred over airborne delay. If the origin-destination pair is also the same for a flight and there are at least two routes with that pair, rerouting is possible. It is being studied how to design “extra” routes outside the flight plans in order to allow the rerouting option in a consistent way, starting with the application of the National Severe Weather Playbook (19).

2.5.3 The optimization problem for the RBAM

The RBAM defines the cost function as:

$$\min J = \min(-w_{GD} + k_{RR}w_{RR})\bar{u}^T \quad (2.6)$$

Where

$$w_{GD} = [k_f^1 \dots k_f^p | k_f^1 - 1 \dots k_f^p - 1 | 1 \dots 1]^T$$

$$w_{RR} = [n_1 \dots n_p | \dots | n_1 \dots n_p]$$

The vector $\bar{u}$ is of size $k_f p \times 1$, so it represents all the controls not until for each route but also in each time step. The term w_{GD} is the cost weight of the Ground Delay, and with the control vector $\bar{u}$ forms a linear function which penalizes the total number of delayed time

steps during the whole simulation. The term w_{RR} forms another linear function with $\bar{u}$ that penalizes the longest route for the rerouting delay. The term k_{RR} establishes how expensive is to make a rerouting over a ground delay, as a rerouting usually implies a longer route and therefore a bigger expense in fuel. The value of this term is studied with more detail in Chapter 4. To solve this optimization problem, a simplex algorithm is applied.

The system constrains are:

- Do not allow early departures: $A_{dep}\bar{u} \leq 0$. It prevents flights to be set on a time step earlier than they should be. This constraint is also responsible for connecting the different routes with same origin-destination pair.
- Do not delay more flights than scheduled: $-\bar{u} \leq \bar{s}$.
- No more flights in a sector than its maximum capacity: $A_{cap}\bar{u} \leq b_{cap}$
- No more departures than maximum airports departure capacity: $A_{md}\bar{u} \leq b_{md}$

A new inequality constraint and a new equality constraint are added to the model at this point:

- No more cancelled flights than an established percentage: $A_{can}\bar{u} \leq b_{can}$. This inequality constraint is a variation of the equality constraint presented in (6). Instead of fixing that all the flights must depart, which may make the problem infeasible, a percentage of minimum flights is established. It can happen that the problem also becomes infeasible for the given percentage. To solve this, the problem can be solved iteratively increasing the percentage of flights cancelled at each iteration until a feasible solution is reached (Figure 4).

- Not controlling entering or transiting flights constraint: $A_{ncu}\bar{u} = b_{ncu}$. Because of the way the model works, it is necessary to establish which origin cells can be controlled and which cells cannot. Due to the formulation of the model, some origin cells may be from routes which have inputs coming from $s(k)$ but because of the boundaries of the regions those cells represent entering cells for already airborne flights. Therefore it is necessary to establish which cells can be controlled and which cannot, and to avoid touching the original dynamics of the model, this constraint is essential.

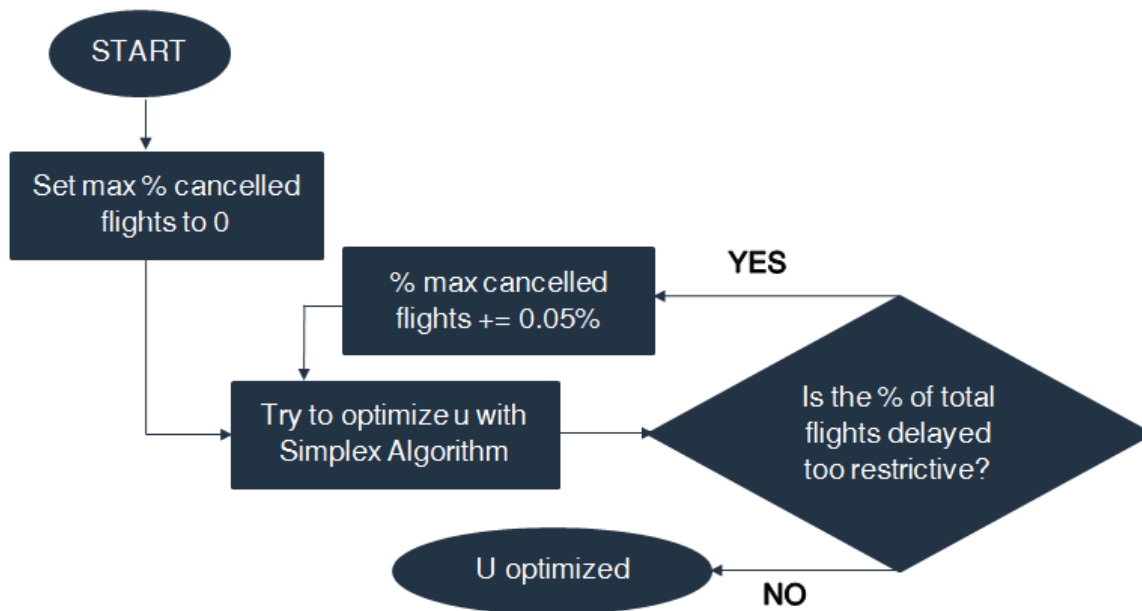


Figure 4: Minimum cancelled flights algorithm

2.6 CTM(L) vs LTM vs RBAM

After presenting the three models, the comparison will be focused in the three main aspects of each: how the NAS is modeled in each method, how the dynamics differ and how they approach the optimization problem. Table 1 summarizes the three models.

Table 1: Summary of CTM(L), LTM and RBAM

	Modelling the NAS	Dynamics	Optimization problem
CTM	- Link based connecting sector neighbors. - Requires historical data - Discretized links into equally spaced cells (CFL = 1) - No diverging nodes	- Airborne delay - Control in each cell - Linear-time invariant	- Simplex method - Minimize the total travel time of all the flights (dependent of $x(t)$)
LTM	- Link based connecting sector neighbors - Requires historical data - No diverging nodes - No airplanes physical location available apart from sector location.	- Airborne and ground delay - Control in each link through the transmission coefficient (not through the common control term)	- Dual decomposition - Minimize the total travel time of all the flights (dependent of $x(t)$)
RBAM	- Route based connecting origin-destination points (airports or region boundaries) - Based on the flight plans (no historical data needed) - Discretized routes into equally spaced cells (CFL = 1)	- Ground delay and ground rerouting - Control at the beginning of each origin-destination where the origin is an airport (internal and exiting flights only)	- Simplex method - Minimize the delay and rerouting procedures (dependent on $u(t)$)

2.6.1 Differences in the NAS modelling

The way the CTM(L) and the LTM model the NAS is by using historical data to determine where will the links vertices be in each sector. Because these vertices are averaged, it may lead to error in many cases where the airplanes actually follow a route through that sector between neighboring sectors which is way longer or shorter than where the link is placed. The RBAM instead uses the flight plans for the upcoming flights, which is the intended path the airplanes will follow therefore diminishing the error. A consequence of the RBAM modeling is that it can happen that those cells close to the sector boundaries will be assigned to a sector even if the airplanes in those cells spend some time in the next sector geographically talking. Because of the way the CTM(L) and the LTM work this issue would not happen, although as this is at an strategic level the error is acceptable (being the cell assigned to that sector where the aircraft stay the longest at the time step). The three models are designed to avoid diverging nodes, but in the RBAM there are as many routes in the model as different routes in the flight plans, whilst in the CTM(L) and LTM the links will be duplicated as necessary to avoid the diverging nodes.

The RBAM routes have an established origin (airports or boundary region cells) and an established destination cell which will be constant for all the aircraft in that route. The CTM(L) and the LTM on the other hand do not have a fixed starting cell where all the aircraft are incorporated. Although there is a modification of the model to allow having airports (which would be the initial cell for the aircraft there) the model is majorly based in assigning flights that are climbing to a high-altitude to the cell which is closest to the point where the flights will enter the model boundaries.

2.6.2 Differences in the dynamics

The dynamics is the place where the models differ the most. In this case it is necessary to compare the RBAM with the CTM(L) and the LTM separately.

The CTM(L) and the RBAM present both a Linear Time-Invariant problem, with the common $x(t) = Ax(t - 1) + Bu(t - 1)$ structure. The sizes of the problems though differ considerably, substantially for the control vectors and consequently for the optimization problem. In the CTM(L), each cell can be controlled. In the RBAM only those origin cells that represent airports can be controlled. Because of the high-altitude characteristic of the cells in the CTM(L) whichever control that is applied to any of them means airborne delays, and with this comes air patterns. These patterns imply an expense of fuel which does not happen in the delays provided by the RBAM, because they are only ground focused. This is a more logical approach for a strategic air traffic flow management, as it would not make sense to have airborne delays scheduled 8 hours in advanced when a ground delay can be applied. It also reduces considerably the optimization problem (more on this in the following point).

The LTM has a completely different approach when it comes to the dynamics. It is a Non-linear time-variant problem. Because it attempts to reduce the size of the system with respect to the CTM(L) by using links as the smallest discretized part, it produces a matrix A which is time-variant. This happens because A is created with transmission coefficients that set the amount of aircraft that move from a link to each other and this coefficients change in every time-step. Also, the system is divided in two components, a deterministic component and a probabilistic component. This last one models those flights which are airborne at the time of starting the simulation and that will enter the system at a certain link. It needs historical data

to process those probability terms (which also change in every time step). Furthermore, these terms do not take into account the weather impact. The RBAM needs no historical data as it is based only on the routes the scheduled flights will use and because it is at a strategic phase, the weather can be predicted and act consequently to prevent delays. Another particularity which makes the LTM more complex than the RBAM is the lack of a control term. Instead, it uses the transmission coefficients inside A to apply control (airborne delay on its majority).

2.6.3 Differences in the optimization problem

As seen in the dynamics differences, the sizes of the problems to solve are different. The RBAM only needs to apply control on as many cells as number of routes are, whilst the LTM and the CTM(L) have an increased number of variables as all the links and cells can be controlled respectively. Both the RBAM and the CTM(L) can be solved with a simplex method, whilst the LTM uses a Dual-Decomposition method.

In this optimization point, the major difference is the definition of the cost function to be minimized. The three models try to minimize the total delay, but whilst the CTM(L) and the LTM use the states $x(t)$ as variables, the RBAM uses the control term $u(t)$. Also, because the RBAM allows ground rerouting, the cost function is designed to determine when it is better to apply a delay or a reroute by weighting it. As rerouting is not considered in CTM(L) and RBAM no weighting factors appear. The CTM(L) has barely explored the idea of ground-rerouting, therefore no study is done on weighting between ground-delay and airborne-delay, but the LTM does consider it, and applies a ratio of 1:2 respectively.

3 Implementation of the model in Matlab: A Ground delay and rerouting example

3.1 Creating the model in Matlab

Matlab is a powerful computing environment and programming language, specially designed to deal with mathematical problems with a heavy matrix component, which makes it perfect for the type of discrete-time problems being faced. It has linear programming solvers with different types of algorithms, including a Simplex-algorithm, which as explained is the optimization algorithm used in the RBAM.

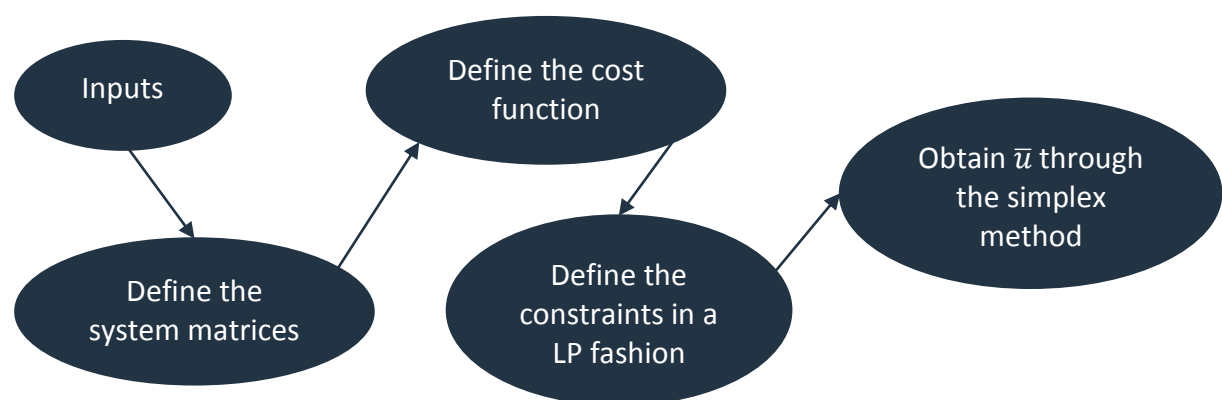


Figure 5: Overview of the Matlab sequential process

Figure 5 shows the sequence the code in Matlab will follow in order to make the RBAM simulation. First, the inputs need to be inserted in Matlab, which include all the information about flights, sectors and routes, among other variables needed to compute the information. After the inputs are inserted, the system matrices from the dynamics are defined and then the cost function. Finally, the constraints are obtained in a Linear Programming fashion (further details on what this means in following paragraphs). Finally, the optimization problem is solved through the linear programming solver with the simplex algorithm implemented in Matlab. To explain how this is done, a practical example is defined.

3.2 Definition of the example

Because there was not any easy to see example documented on how the RBAM deals with a Ground-Rerouting and Ground-Delay over a region with different sectors and varying sector capacities, along with flights coming and leaving the control region, an example was modeled. Figure 6 shows a good overview of the example which will be worked on. The idea is to have 3 airports and a total of 8 routes connecting all them. For each route there is another route in the opposite direction, so two routes connect A and B, for a distance of 2000 nm, another two routes connect A and C for a distance of 2000 nm, and 4 routes connect B and C, the first two are 2000 nm long and the other two 2400 nm.

In the example, the whole area has 4 sectors: ZA, ZB, ZC and ZD, but it is decided that the control is only applied in zones ZB, ZC and ZD. This is made on purpose in order to deal with airborne entering flights as they will increase the state of some cells but no control will be

able to be applied on them, so all the delay or rerouting controls will affect flights either exiting the region (exiting flights) or internal flights.

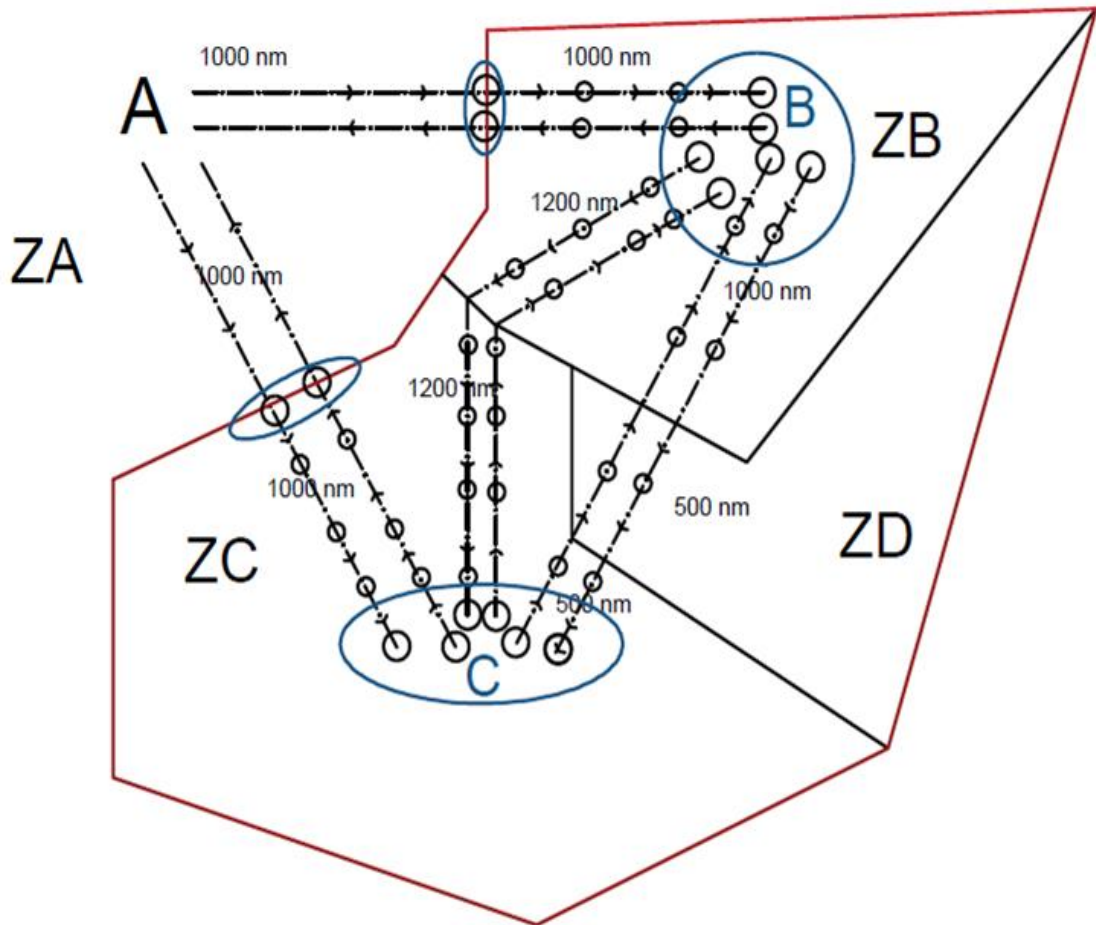


Figure 6: ABC example

The routes connecting B and C will be responsible for showing the capabilities of ground-rerouting. The first pair of B-C routes (this means, route from B to C and route from C to B) are 2000 nm which connect airports B and C. These routes fly over regions ZB, ZC and ZD. The main idea behind the sector ZD is having a sector whose capacity can be set to 0, so no flights can fly over a period of time. The other pair of routes only fly over ZB and ZC, but they are

400 nm longer. So the system (through the cost function) will have to decide whether it is better to delay flights at B and C or if it'll be better to reroute them to the longer route.

3.3 Inputs

To define the model it is necessary to know the inputs to be introduced. The inputs needed include all the information related to the airports, routes, sectors and flights, and also other variables which are necessary, such as the duration of the simulation, the desired time-step and if there are some sector or airport constraints to simulate external conditions such as airport closures or weather impact. Some of the information comes from large tables, and to make it easier to deal with that information, they are saved in *Microsoft Excel* “.xls” files, which are then imported in *Matlab* with the *xlsread* function, which only needs the filename to process the data.

3.3.1 Flight information

The flight information comes in a table of the style of Table 2. The Origin and Destination fields include the name of the origin and the destination airports. The ETD indicates the estimated time of departure of the flight, whilst the CS represents the Cruise Speed for that flight. The *Osector*, *Dsector* and *Tcenter* have the information of the origin sector, the destination sector and the transition sectors (the sectors the flights cross). Finally, there is the *RouteID*, which indicates which route is following (each Route has a unique ID).

Table 2: Flight information table

Origin	Destination	ETD	CS	Osector	Dsector	Tsectors	RouteID
B	C	1:00	500	ZB	ZC	ZD	3

For the example, it is considered that every 5 minutes 2 flights depart, one for each of the other 2 airports, in a time-lapse of two hours. There are no flights scheduled following the longest Route B-C neither the Route C-B, to use it as an auxiliary route in case of need.

3.3.2 Sectors' information

The sectors' information is loaded from a source with the aspect of Table 3. This table includes all the information from all the sectors known, not only those that will be in the bounded region which will be simulated. The field *SectorID* indicates the unique numerical ID from the sector. Because of the way Matlab works, it is better to use numerical IDs instead of Strings as the processing power to process the latest is significantly bigger and harder to implement. The field *Sector* indicates the name of the Sector assigned to the previous ID. Finally, the field *Capacity* indicates the standard capacity of the sector. For this example, the capacity is set to 400 in order to be sure it will be able to handle all the flights scheduled, as the capacity can be individually constrained as will be discussed in paragraph 3.3.5.

Table 3: Sector Information

SectorID	Sector	Capacity
1	ZA	400
2	ZB	400
3	ZC	400
4	ZD	400

3.3.3 Routes information

The routes are inserted in Matlab through Table 4. As in the Sectors, each route has a unique ID, set in *RouteID*. The *Origin* and *Destination* fields include the origin and destination airports. The field *Total Distance* indicates the length of the route in nautical miles. *Route number* is used to identify routes with same origin-destination pair. So for routes B-C and C-B there is a route with route number 1 and another with route number 2, differentiating all the routes that connect those pair of airports. *Sectors Order* indicates the order of sectors the flights fly over in order from the origin sector to the destination sector. They use the Sectors ID instead of the names to make it easier to process with Matlab. Finally, the last columns are linked to the sectors order, as they represent from left to right the distances each route has set in each sector. So for the route with ID 3, the route starts with 1000 nm in sector 2 (Sector ZB), then 500 nm in sector 4 (Sector ZD) and finally 500 nm in sector 3 (Sector ZC). This makes the system have a lot of freedom at the time of defining the routes, as it allows the routes to be in unlimited sectors and also to fly over the same sector more than once (non-convex sectors).

Table 4: Routes information

RouteID	Origin	Destination	Total Distance	Route number	Sectors Order			
1	A	B	2000	1	1 2	1000	1000	
2	A	C	2000	1	1 3	1000	1000	
3	B	C	2000	1	2 4 3	1000	500	500
4	B	C	2400	2	2 3	1200	1200	
5	B	A	2000	1	2 1	1000	1000	
6	C	A	2000	1	3 1	1000	1000	
7	C	B	2000	1	3 4 2	500	500	1000
8	C	B	2400	2	3 2	1200	1200	

3.3.4 Airports information

The last of the tables imported contains the information for the airports as in Table 5. The column *Airport* contains the name of the airport. *Sector ID* indicates the ID of the sector where the airport is placed. *MaxCap* indicates the maximum capacity of the airport. Finally, *Airport ID* provides a unique numerical ID for each airport. For the example, all the airports are set to have a standard maximum capacity of 1 per minute, which can be modified later to simulate different scenarios.

Table 5: Airports information

Airport	Sector ID	MaxCap	Airport ID
A	1	1	1
B	2	1	2
C	3	1	3

3.3.5 Other inputs

In Matlab there are other variables which need to be inserted manually so the problem can be optimized. They are presented to show how they look like with the information of the example:

- ***Region* = {'ZB','ZC','ZD'}**. This represents the region that will be optimized. It forms an array with the names of the Sectors inside the bounded region.
- ***k_{rr}* = 2**. This is the cost ratio of the rerouting with respect to the ground delay.
- ***dt* = 5**. This is the time-step of the system. It is fixed in 5 minutes for this example.
- ***t₀* = 0**. This is the initial time in minutes.

- $t_f = 420$. This is the final time of the simulation in minutes.
- $kmc = 0$. This is the maximum percentage of cancelled flights allowed, 0% in the example.
- $sectorMaxCapKf = [2 \ 0 \ 42 \ 47; 3 \ 10 \ 50 \ 80]$. This variable indicates specific sector capacity constraints at certain time-steps. It is a matrix where every row indicates a constraint. The first column indicates the Sector ID where the constraint is applied. The second indicates the capacity for that sector. The third the initial time-step at which the constraint starts acting and the last the final time step where the constraint will be active. The values used now are not the values that will be used when the system is executed, they are used here as an example. In this case, it can be read that for sector ID 2 ('ZB'), the constraint has 0 capacity from time step 42 to time-step 47 (simulating for example a weather storm which closes that sector's airspace). Also, for sector 3 ('ZC') there will be a constraint of only 10 flights in that sector from time-step 50 to time-step 80.
- $airportMaxDepKf = [1 \ 2 \ 32 \ 31]$. This sets the constraints for the airports departures, in the same fashion as the $sectorMaxCapKf$ variable, but being the first column the airport ID.

3.4 System matrices

3.4.1 Obtaining A and B

Equation 2.5 shows the dynamics of the system. With the inputs already inserted in Matlab, A , B and the forced input $s(t)$ can be set. A and B can be obtained by knowing the routes, the time-step and the speed of the aircraft. The speed is calculated by averaging the Cruise Speed of all the flights inserted in the model. Then the cell size can be calculated as:

$$cellSize = (dt * v)/60 \quad (3.1)$$

Where v is the average speed of the system and it is divided per 60 as it is in knots and it is desired to work with nautical miles per minute. With the size of the cell, the size of the system (the number of cells, m) and the boundary cells can be obtained (the origin and destination cells for each route). Knowing the boundary cells, A , an $m \times m$ matrix, is created by adding 1s in a zero-matrix on its subdiagonal for each route. B is even easier to create as it is an $m \times p$ matrix, with p being the number of routes, where all are zeros except for the origin cells of each route, which will be ones.

For this example, the boundary cells can be seen in Table 6, with the same format as in Matlab. A is a matrix of size 316×316 and B a matrix of size 316×8 . B has 1s in the coordinates [Origin Cell, Route ID] using the information from Table 6. Although the size of these matrices are relatively small, for bigger problems they can become too big even for a program like Matlab. Sparse matrices are used instead of normal matrices in order to prevent the system

from collapsing. Because most of the terms in the system matrices are zeros, instead of assigning values (including zeros) to all the terms of the matrix, sparse matrices only save the coordinates of those terms which have a value different from zero, reducing the size and the computing time whilst providing the same functionalities as normal matrices.

Table 6: Boundary Cells Table

Route ID	Origin Cell	Destination Cell
1	1	25
2	26	50
3	51	99
4	100	158
5	159	183
6	184	208
7	209	257
8	258	316

3.4.2 Classification of the flights

The flights are classified in entering flights, exiting flights, transiting flights and internal flights, following the algorithm in Figure 7. The algorithm compares the ID of the sectors where the airports are located with the sectors inside the bounded region, and sort the flights depending on the result. For the entering and transiting flights, a function is created to determine the time at which they will enter the bounded airspace. This is possible thanks to knowing the whole route the flights are following and the average cruise speed.

3.4.3 Forced input $s(t)$

The matrix $s(t)$ defines the flights that will enter the system at an specific point through one of the origin boundary cells. It is a $p \times k_f$ matrix, where p is the number of routes and k_f the number of time-steps of the system (which is calculated by $k_f = (t_f - t_0)/dt$), for each column, which represents every time-step, the number in each row is the number of flights entering the system. This is done by scanning all the flights departures times and assigning them to their respective time-step. Because of the linear programming style, this matrix is then rearranged to become a vector, of size $pk_f \times 1$, of the type:

$$\bar{s}(t) = \left[s_1^1 \dots s_p^1 \mid \dots \mid s_1^{k_f} \dots s_p^{k_f} \right]^T \in \mathbb{R}^{k_f * p \times 1} \quad (3.2)$$

In the example $k_f = 85$. Therefore, the size of $\bar{s}(t)$ is $85 * 8 \times 1 = 680 \times 1$.

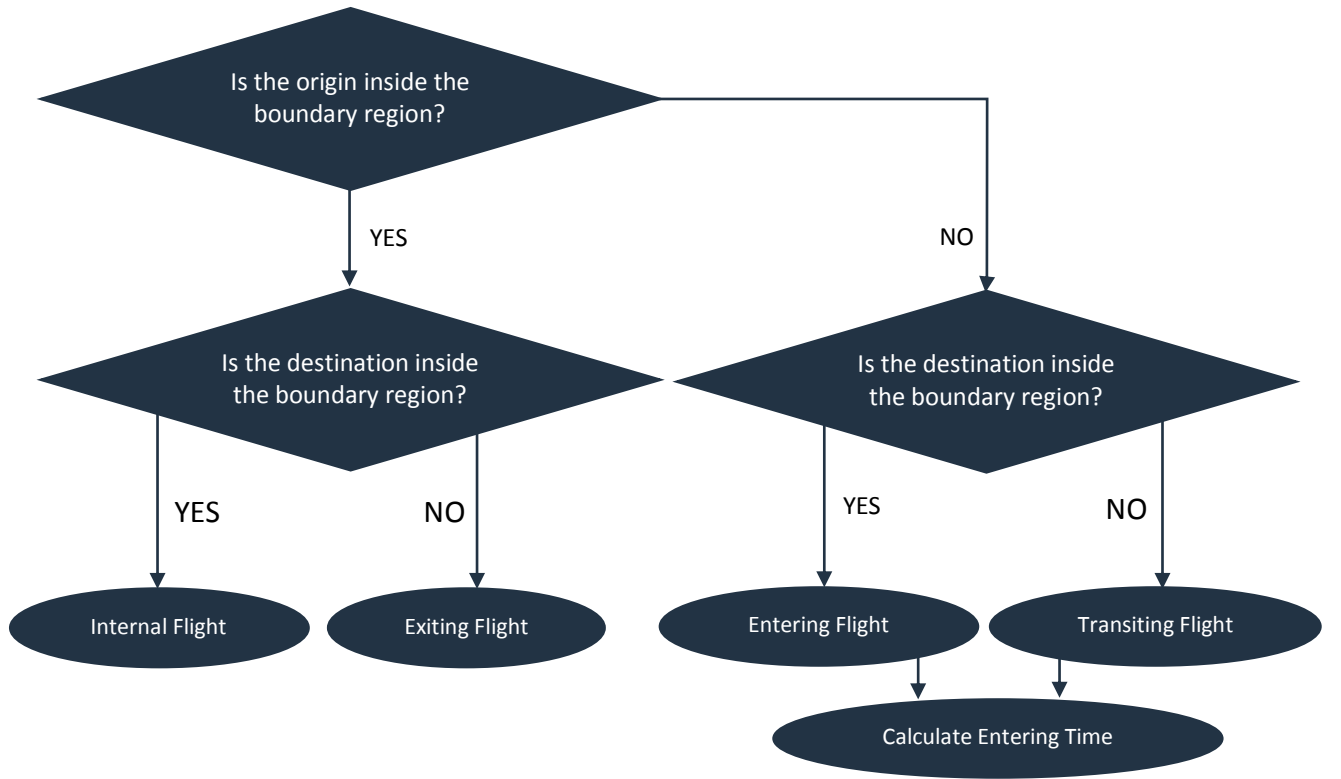


Figure 7: Flight classification algorithm

3.5 Cost function

The cost function of the system is defined as Equation 2.6. Again, this cost function needs to be expressed in a Linear Programming fashion, so the control vector will be of the type:

$$\bar{u}(t) = \left[u_1^1 \dots u_p^1 \mid \dots \mid u_1^{k_f} \dots u_p^{k_f} \right]^T \in \mathbb{R}^{pk_f} \quad (3.3)$$

It has the same size as $\bar{s}(t)$. The terms w_{GD} and w_{RR} are also obtained easily as the length of the routes and the number steps is already known. In our example, the vectors are made of $k_f = 85$ blocks of 8 (as many routes in the system), forming:

$$w_{GD} = [85 \ 85 \ 85 \ 85 \ 85 \ 85 \ 85 \ 85 \ | \ ... \ | \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$w_{RR} = [24 \ 24 \ 48 \ 58 \ 24 \ 24 \ 48 \ 58 \ | \ ... \ | \ 24 \ 24 \ 48 \ 58 \ 24 \ 24 \ 48 \ 58]$$

The vector w_{GD} has blocks with descending numbers from 85 to 1 representing the cost of the ground delay, penalizing more the delays at early time steps over the delays at the end. The vector w_{RR} has blocks which are repeated as many times as time steps are, and each value represent the number of cells each route has. The value k_{RR} was already inserted in the inputs, but Chapter 4 will address how to properly calculate this value.

3.6 Constraints

Constraints are the most important part and the most difficult to model in the system because they have to be modeled in a Linear Programming fashion. Also, the size of the matrices increase considerably, so the use of sparse matrices is completely necessary.

3.6.1 Maximum Sector Capacity constraint

The maximum sector capacity constraint is the most difficult to model from all the constraints, as it implies transforming a constraint depending on the states $x(t)$ (Equation 3.4) into a constraint depending on the control vectors $u(t)$.

$$\sum_{s: x_s \in X_j} x_s(t) \leq c_j(t) \quad (3.4)$$

X_j is the set of cells which belongs to sector j at time t . To transform this equation, Equation 3.5 taken from (6) is used:

$$x(t) = A^k x(0) + \sum_{i=1}^k A^{i-1} B(s(i) + u(i)) \leq c(k) \quad (3.5)$$

As all the terms are known ($x(0)$ are the initial condition of the states, which for our example will be 0 considering there are no airborne flights at the start of the simulation), $u(t)$ can be isolated and express the constraint in the form of $A_{cap} \bar{u} \leq b_{cap}$. Equation 3.5 does not use a Linear Programming style, which means the system matrices needs to be modified in order to be able to put in the solver. The result of this is:

$$A_{cap} = M * A_2 * B_1 \quad (3.6)$$

$$b_{cap} = \bar{c}(t) - M * A_1 * \bar{x}(0) - M * A_2 * B_1 * \bar{s}(t) \quad (3.7)$$

Where

$$\bar{c}(t) = [c_1(1) \dots c_j(1) | \dots | c_1(k_f) \dots c_j(k_f)] \in \mathbb{R}^{jk_f} \quad (3.8)$$

$$M = \begin{bmatrix} l & & \\ & \ddots & \\ & & l \end{bmatrix} \in \mathbb{R}^{jk_f \times mk_f} \quad (3.9)$$

being I a matrix of size $j \times k_f$ which maps all the cells which belong to the same sector,

$$A_1 = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \in \mathbb{R}^{mk_f \times mk_f} \quad (3.10)$$

$$A_2 = \begin{bmatrix} I & & & \\ A & & & \\ \vdots & & \ddots & \\ A^{k_f-1} & \dots & A & I \end{bmatrix} \in \mathbb{R}^{mk_f \times mk_f} \quad (3.11)$$

The results are a matrix $A_{cap} \in \mathbb{R}^{255 \times 680}$ and a vector $b_{cap} \in \mathbb{R}^{255 \times 1}$. The importance of using Sparse matrices appears explicitly when calculating A_1 and A_2 , as even for a small problem like this their size is of 26860×26860 (which is the size of A times the number of time-steps).

3.6.2 No early departures constraint

This constraint prevents flights from departing before they scheduled to. It also connects the same family routes so flights can be rerouted.

$$A_{dep} \bar{u} \leq 0 \quad (3.12)$$

Where

$$A_{dep} = \begin{bmatrix} A_d & & \\ \vdots & \ddots & \\ A_d & \dots & A_d \end{bmatrix} \in \mathbb{R}^{rk_f \times pk_f}$$

Being r the number of family routes and $A_d \in \mathbb{R}^{r \times p}$ matrix where every row defines a route of families and each column a route. A 1 is assigned in each row to those routes that have the same origin and destination. The rest are zeros. In the example $A_{dep} \in \mathbb{R}^{510 \times 680}$ and

$$A_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

3.6.3 No more delays than flights constraint:

This constraint is easy to implement as Equation 3.13 can be achieved by multiplying $\bar{u}(t)$ by the negative of an Identity matrix of size $k_f p \times k_f p$ (680 x 680 in the example).

$$-\bar{u}(t) \leq \bar{s}(t) \quad (3.13)$$

3.6.4 Airport maximum departures constraint

This constraint, expressed in Equation 3.14, prevents to insert more flights through the airport cells than the ones the airports can handle, where a is the number of airports inside the bounded region.

$$A_{md} \bar{u} \leq b_{md} \quad (3.14)$$

Where

$$A_{md} = \begin{bmatrix} A_{airp} & & \\ & \ddots & \\ & & A_{airp} \end{bmatrix} \in \mathbb{R}^{ak_f \times pk_f} \quad (3.15)$$

$$b_{md} = airpMaxConstrain - A_{md} * S_1 \quad (3.16)$$

Being A_{airp} a matrix that maps the airports (rows) to the routes (columns) with ones and the rest being zeros; and $airpMaxConstrain$ a vector with the maximum departures capacity for every airport at every time step. In the example:

$$A_{airp} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

3.6.5 Maximum cancelled flights constraint

This is a new inequality constraint added to the model which is defined in Equation 3.17, which forces the model to find a solution with less than the maximum percentage of cancelled flights allowed set at the beginning with the input k_{mc} . It generally starts with 0 and then at the time of executing the solver increases iteratively until a feasible solution is found. This forces the system to have the minimum amount of cancelled flights possible. Only the internal flights and exiting flights can be cancelled, as they are the only type of flights which the system has control over.

$$A_{can}\bar{u} \leq b_{can} \quad (3.17)$$

$$A_{can} = [-1 \dots -1] \in \mathbb{R}^{p_{kf}} \quad (3.18)$$

$$b_{can} = k_{mc} * \sum(\#internalFlights + \#exitingFlights) \quad (3.19)$$

3.6.6 Non-controllable origins constraint

The last constraint is an equality needed to avoid the system from controlling the origin cells of the transiting and the entering flights, as the flights entering at those cells are airborne and therefore no ground control can be applied. Equation 3.20 describes this constraint:

$$A_{ncu}\bar{u} = b_{ncu} \quad (3.20)$$

Where

$$A_{ncu} = \begin{bmatrix} A_{ancu} & & \\ & \ddots & \\ & & A_{ancu} \end{bmatrix} \in \mathbb{R}^{a_n k_f \times p k_f} \quad (3.21)$$

Where A_{ancu} map the origin cells (rows) with their respective routes (columns) and a_n is the number of origin cells for routes representing entering or transiting flights. In the example:

$$A_{ancu} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.7 Solving the optimization problem

3.7.1 Solving the problem with Matlab

Having all the constraints and the cost function defined, it is almost trivial to set Matlab to solve it. The inequality constraints are assembled together, forming $A_{ineq}\bar{u} \leq b_{ineq}$. In the

example, this results in a matrix $A_{ineq} \in \mathbb{R}^{1616 \times 680}$ and a vector $b_{ineq} \in \mathbb{R}^{1616}$. The equality constraint is Equation 3.20. The function to solve the optimization problem is:

$$[u, fvar, exitflag] = linprog(J, A_{ineq}, B_{ineq}, A_{ncu}, b_{ncu}, [], [], [], options)$$

The inputs are the constraints, the empty spaces that represent the lower bounds, the upper bounds and the initial conditions of $\bar{u}(t_0)$ which are not used for this problem, and the options of the system, which forces the solver to use a Simplex algorithm.

The outputs are u , $fvar$ and $exitflag$, which are $\bar{u}(t)$, the cost function J calculated and an exit flag which indicates the state of the optimization. This exit flag is used in case the maximum cancelled flights constraint is too restrictive. An exit flag indicating that the problem has no feasible solutions will appear, and with that information the system will increase the maximum cancelled flights percentage iteratively until the exit flag indicates that the problem is feasible and solved.

Three cases have been used with different parameters in order to see the behaviour of the model. The three cases are as defined at the beginning of Chapter 3: 150 flights departing in a 2 hour window, a total simulation time of 7 hours with a maximum airport capacity of 1 flight/min. The first two examples compare how the ground rerouting factor k_{RR} affects the delays of the system. Because this factor has not defined values set in the literature, a $k_{RR} = 1$ will be used for the first example whilst a $k_{RR} = 10$ will be used for the second example, being the rest of conditions the same. The third example will present a situation with different constraint capacities in the different sectors of the system.

The simulations are run in a computer with an Intel i7 CPU Q720 @ 1.60 GHz, 12.0 GB of DDR3 RAM and Windows 8 64 bit OS.

3.7.2 Example 1

The first example will simulate a 30 minute period where sector 'ZD' will have null capacity, starting at 3 hours and 30 minutes after the start of the simulation. The desired maximum cancelled flights is 0%. As mentioned above, $k_{RR} = 1$. After running the simulation, the results can be seen in Table 7. The red cells indicate that there is a ground rerouting from that route. The green cells indicate that the route is receiving a flight from another route with same origin and destination. The orange cells indicate a ground-delay in that route and a blue cell indicate that a flight that was delayed in ground is authorized to depart. There are:

- **17 Reroutes:** 10 from the 2000 nm B-C route to the 2400 nm B-C route and 7 reroutes from the 2000 nm C-B route to the 2400 nm C-B route.
- **7 Ground delays:** In route B-C. Because of the airport departure maximum capacity constraint, the flights are released in two blocks of three flights and a final block of one flight in consecutive time-steps. This is easy to understand as the capacity of one flight in consecutive time-steps. This is easy to understand as the capacity of the airports is 5 flights/time-step (as the time step is 5 minutes and the capacity 1flight/minute), and there are two departing flights scheduled in every time-step in every airport, making the total of 5 flights. Without that constraint, the 7 flights would have left at the same time.
- **0 % cancelled flights.** All the flights have departed inside the simulating time.
- 35 minutes maximum ground delay for a flight and 48 minutes of delay caused by ground re-routing.

Table 7: Ground delays and rerouting for $K_{RR} = 1$

t	35	40	45	50	55	60	65	70	75	80
(B,C) ₂₀₀₀	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
(B,C) ₂₄₀₀	1	1	1	1	1	1	1	1	1	1
(C,B) ₂₀₀₀	0	0	0	0	0	0	0	0	0	0
(C,B) ₂₄₀₀	0	0	0	0	0	0	0	0	0	0
t	85	90	95	100	105	110	115	120	125	130
(B,C) ₂₀₀₀	-1	-1	-1	-1	-1	-1	-1	3	3	1
(B,C) ₂₄₀₀	0	0	0	0	0	0	0	0	0	0
(C,B) ₂₀₀₀	0	0	-1	-1	-1	-1	-1	-1	-1	0
(C,B) ₂₄₀₀	0	0	1	1	1	1	1	1	1	0

3.7.3 Example 2

This is the same example as Example 1, but in this case $k_{RR} = 10$. This means that the cost of ground-rerouting is 10 times more expensive than the cost of ground-delay. The results can be seen in Table 8.

- **24 ground delays.** 17 flights delayed in Route B-C and 7 flights delayed in Route C-B.
- **0 Ground reroutes.** It seems logic to see that the value of $k_{RR} = 10$ makes the ground delay much cheaper than the ground re-routing and in this case the function is minimum only with ground delays.

- **85 minutes maximum delay.** This happens for the first flight delayed in Route B-C, which is delayed at $t=35$ min and is not authorized to depart until $t=120$. It is important to see that this does not mean the flight will depart at that time, and it may be delayed longer, because this is an aggregate level and there is no information on which aircraft are set to depart after a delay, but it is logic to think that the flights with a longer delay will be the first to depart. This would be an interesting field of study.
- **0 % cancelled flights.** As in the first example, no flights have been cancelled.

Table 8: Ground delays and rerouting for $K_{RR} = 10$

t	35	40	45	50	55	60	65	70	75	80	85	90
(B,C) ₂₀₀₀	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
(C,B) ₂₀₀₀	0	0	0	0	0	0	0	0	0	0	0	0
t	95	100	105	110	115	120	125	130	135	140	180	185
(B,C) ₂₀₀₀	-1	-1	-1	-1	-1	3	3	5	5	1	0	0
(C,B) ₂₀₀₀	-1	-1	-1	-1	-1	-1	-1	0	0	0	5	2

3.7.4 Example 3

This last example tries to consider a scenario where all the sectors have their capacity affected at some point of the simulation and also the capacity of airport B is reduced for a small period of time. $k_{RR} = 1$ in this example.

The sector and airport capacity constraints in this example are:

- Sector 'ZD' has null capacity from $t = 5$ h. during 30 minutes.
- Sector 'ZB' has a capacity of 20 airplanes from $t = 50$ min during 50 minutes.
- Sector 'ZC' has a capacity of 20 airplanes from $t = 2$ h. 55 min during 30 minutes
- Airport B has a reduced departure capacity of 3 flights/time-step at $t = 1$ h. 40 min for 15 minutes.

The results can be seen in Figure 8. Analyzing the data, it can be seen that there is a combination of ground delay and ground re-routing. The first that can be seen is that the Route CB (2000 nm), which could have rerouted flights, have only delayed them instead, so Route CB (2400 nm) is not used. Route BC (2400 nm) receives flights which have been rerouted from Route BC (2000 nm). The constraints of the maximum departures show how no more than 5 flights are set to depart at any given time. Because there are no scheduled departures after $t = 2$ h, blocks of 5 delayed flights are set to depart at the following time-steps (if necessary), whilst before $t = 2$ h. because there are flights already departing the control does not reach the maximum capacity of the airports. It is also interesting to see that some flights are not only delayed, but rerouted. This means that the total delay will be the ground delay plus the extra time because of the longer route (in this case 48 minutes). In numbers:

- **49 flights are ground delayed.** 12 from Route BC (2000 nm), 24 in Route BA, 11 in Route CA and 2 in Route CB (2000 nm).
- **8 flights are rerouted.** The 8 flights that are rerouted fly from B to C and the rerouting implies taking the route that is 400 nm longer.
- **2 hours maximum ground delay.** This is the maximum ground delay for a flight, in the Route BA, scheduled to depart at $t=0$ and departing the earliest at $t = 2$ h.
-

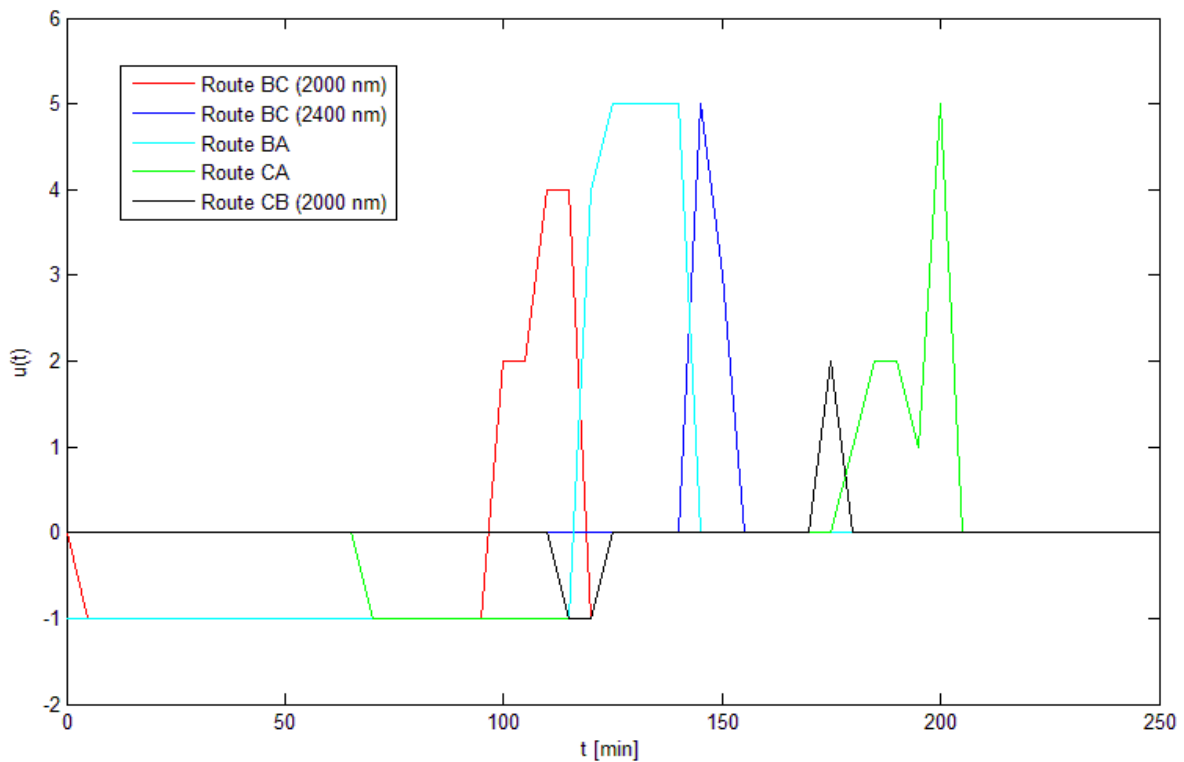


Figure 8: $u(t)$ for the different routes in example 3

4 Calibration of the Cost Function

4.1 Overlook of the rerouting weighting value

The cost function is designed with the idea of reducing the total delay by either applying a ground rerouting or a ground delay in the upcoming flights. The term k_{RR} works as a way of weighting the two types of controls. Because no study was made on the difference in cost between a ground delay and a ground rerouting, there was no formal criteria to establish the value of this term. The ground rerouting will always imply an extra expense of fuel, which is the major concern for airlines right now. But also time is important, as a too long delay may imply a bigger cost because of the cost towards the passengers and the crew. Therefore an analysis of the cost of flights needs to be addressed to properly weight the ground rerouting and the ground delay.

4.2 Cost of the flights delays and rerouting

A delay produces an increasing cost because it implies adding extra time to the cost of a flight. Whether there is a rerouting or a ground delay, both increase the time of the flight. The total cost of a flight can be defined as Equation 4.1 (20).

$$C = C_F * \Delta F + C_T \Delta T + C_c \quad (4.1)$$

With

- C_F = cost of fuel per kg
- C_T = Time-related cost per minute of flight
- C_c = fixed costs independent of time
- ΔF = Trip fuel
- ΔT = trip time

For flight, it is being considered from the time the flight is preparing to depart until it has landed and it is ready for a new flight. Any type of delay increases the time-related costs, which includes the crew salaries, maintenance, and others. Here it can also be included the cost of the delay for a passenger per minute as in (21). There are two main factors which differentiate the cost of the ground rerouting and the ground delay. The ground rerouting produces a delay because of the use of a longer route which implies more time flying and consequently a bigger expense in fuel as reflected in Equation 4.2. The ground delay produces an increase of the flight time and also an added cost produced by the cost of the overtime parking at the airport, as in Equation 4.3.

$$C_{RR} = C_T * \Delta T_{reroute} + C_F * \Delta F + C_{passenger} * \Delta T_{reroute} \quad (4.2)$$

$$C_{GD} = C_T * \Delta T_{GD} + C_{overtime} * \Delta T_{GD} + C_{Passenger} * \Delta T_{GD} \quad (4.3)$$

With

$$\Delta T_{reroute} = \frac{\text{Length new route} - \text{Length original route}}{\text{Cruise Speed}} \quad (4.3)$$

The fuel consumption can also be expressed as a function of $\Delta T_{reroute}$, by knowing the fuel consumption rate per minute: $C_{F/min} * \Delta T_{reroute}$.

The value of K_{RR} is the ratio between the cost of the rerouting and the ground delay for the same time delay: $\Delta T_{reroute} = \Delta T_{GD} = \Delta T$. Therefore, K_{RR} can be defined as:

$$K_{RR} = C_{RR}/C_{GD} = \frac{C_T + C_{passenger} + C_{F/min}}{C_T + C_{passenger} + C_{overtime}} \quad (4.4)$$

4.3 Example with real values for the rerouting weighting value

To prove how the system works with a calibrated K_{RR} , a simple example is done where it is easy enough to calculate by hand whether it is better to have a ground delay or a ground rerouting. There are two routes connecting airports A in Sector 1 and B in Sector 2, one of length 300 nm and the other of length 600 nm, as in Figure 9. The first route crosses Sector 3. By setting the capacity of Sector 3 to 0 and varying the time this happens it is possible to determine by hand whether it will be better to have a ground delay or a rerouting. To know this, it is necessary to know how much the reroute costs and then equal it to the cost of a ground delay and isolate the time. For this example, an Airbus A320 flying at an average speed of 500 knots is used.

The costs for this example are as follow:

- $C_T = \$36.67/min$ (21)
- $C_{passenger} = \$66/min$ (21)
- $C_{F/min} = \$39.44/min$ (with a cost of \$4/gallon (22) and a fuel consumption of 2500 kg/h (23))
- $C_{overtime} = \$3.36/min$ (24)

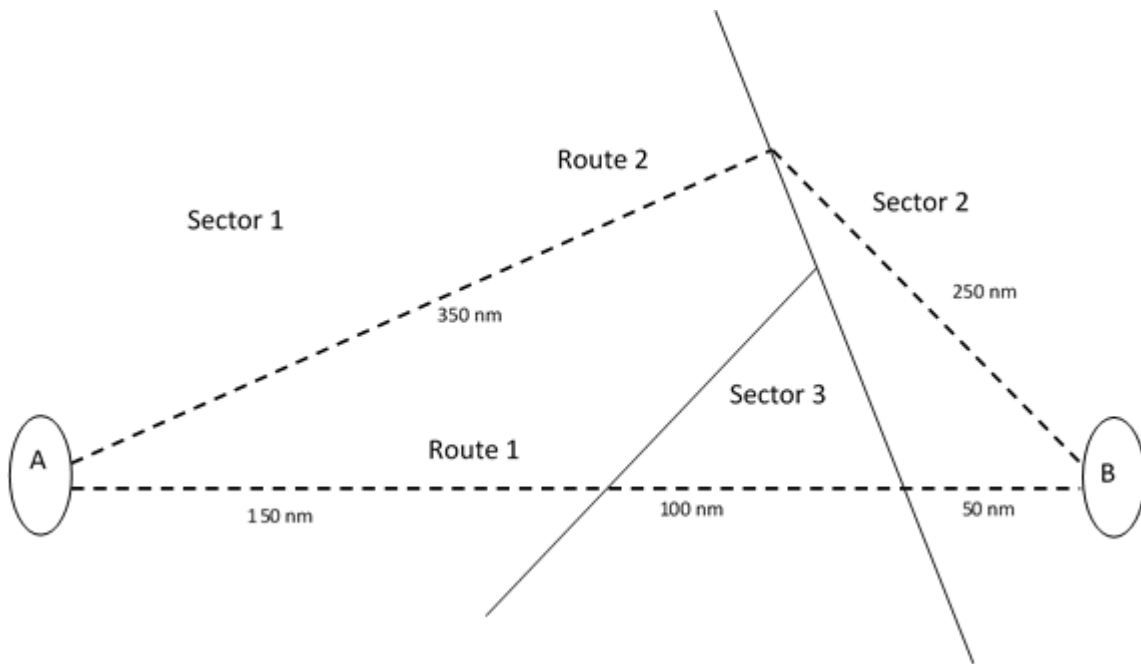


Figure 9: Two route example to see the effects of K_{RR}

For this case, using Route 2 instead of Route 1 implies flying 300 nm extra. At a speed of 500 nm, this means an extra flying time of 36 min. Thus, the total cost of the rerouting is:

$$C_{RR} = (\$36.67/min + \$39.44/min + \$66/min) * 36 min = \$5115.96$$

Now, isolating ΔT_{GD} from Equation 4.3:

$$\Delta T_{GD} = \frac{\$5115.96}{\$36.67/min + \$39.44/min + \$3.36/min} = 64.38 \text{ min}$$

This result means that, for delays smaller than 64.38 minutes, a ground delay is preferable over a ground rerouting, but for a delay bigger than that value, a rerouting would be better.

Now this can be tested in the Matlab model, with

$$K_{RR} = \frac{\$36.67/min + \$39.44/min + \$66/min}{\$36.67/min + \$39.44/min + \$3.36/min} = 1.79$$

The model is tested in two ways, to see the border point between the ground delay and the ground rerouting. First the Sector 3 will have null capacity from $t = 20$ min to $t = 80$ min (60 min restriction), and then it will have null capacity until $t = 85$ min (65 min restriction). The initial time is set at $t = 20$ min as is the time the flight should arrive to Sector 3 if flying in Route 1. The time-step size is 1 min.

Table 9 shows the results for both simulations. For a restriction of 60 min (which implies a delay of 60 min) a ground delay cost less than a ground rerouting (even when the time of the rerouting implies a delay of 36 min). But if the restriction is 65 min, then a rerouting is more cost effective than a ground rerouting. These results agree with the predicted results made by hand.

Table 9: Results for $K_{RR} = 1.79$

	60 min restriction		65 min restriction	
t	1	60	1	65
u(t) Route 1	-1	1	-1	0
u(t) Route 1	0	0	1	0

5 Conclusions

This research has led to the following conclusions:

- Most of the models used in Traffic Flow Management follow an aggregate approach. These models simplify and reduce the order of the problem, allowing a way of implementing not only an accurate simulation of the TFM problem but also a solution for the optimization problem.
- The Route-Based Aggregate Model is one of the latest models created to address the TFM problem. At the time of creation it tried to address the problem of previous models by eliminating the diverging nodes and the transmission coefficients, by using a Route-Based modeling with a $CFL = 1$. Two constraints are added to the model. The first one is an inequality constraint to allow more control on the number of cancelled flights by minimizing it iteratively. The second simply defines which origin cells cannot be controlled because they represent entering and transiting flights routes.
- After this model was presented, two models, the Large-Capacity Transmission Model CTM(L) and the Link Transmission Model LTM were discovered, addressing some of the same issues the RBAM did, but with a different approach. These models need the usage of historical data to model the NAS, and only consider delay (airborne on its majority). The CTM(L) has similar dynamics to the RBAM, but allowing control in all its cells in opposition to the RBAM which only allows control on the origin cells because of the ground control approach, therefore reducing the size of the optimization

problem. The LTM moves away from the Linear Time-Invariant problem and present a Non-Linear Time Variant problem where the control is applied over the transmission coefficients between links, inserted in the system matrix.

- Implementing the RBAM in Matlab is not trivial, because of the need to translate the model into a linear programming problem. An explanation on how to focus the coding of the model, specially the constraints has been done. Even for small optimization problems, some matrices can become extremely large, requiring the use of sparse matrices. Some examples are provided including situations with ground delays, ground reroutings and combined ground delays and reroutings.
- Although the cost function of the RBAM contemplates the difference of cost between a ground delay and a ground rerouting, there was no study done on how different the weights are. This has been solved by studying the cost of both controls and creating an equation to find the ratio between the cost of the ground delay and the ground rerouting, which depends on the cost of fuel, the cost of the overtime parking at the airport for delayed flights, the cost for the passengers and the time-based costs (crew salaries, maintenance, etc.). An example with real data is also provided and tested in the Matlab model.
- Future work will address the creation of the alternative routes in the NAS so the flights can use them instead of being ground delayed. Also, airborne rerouting and airborne delay can be implemented for the entering and transiting flights, in order to have control over airborne flights entering the bounded region.

6 Bibliography

1. **Federal Aviation Administration.** *FAA Aerospace Forecast Fiscal Years 2014-2034.* s.l. : FAA, 2014.
2. —. *The Economic Impact of Civil Aviation on The US Economy.* s.l. : FAA, 2011.
3. **Airlines for America.** Per-Minute Cost of Delays to U.S. Airlines. [Online] 2013. <http://airlines.org/data/per-minute-cost-of-delays-to-u-s-airlines/>.
4. **National Oceanic and Atmospheric Administration.** Air Traffic with Day/Night Terminator. [Online] <http://sos.noaa.gov/Datasets/dataset.php?id=44>.
5. *Lagrangian Delay Predictive Tool for Sector-Based Air Traffic Flow.* **Grieder, Pascal, et al.** 5, s.l. : Journal of Guidance, Control, And Dynamics, 2005, Vol. 28.
6. *Optimal Re-Routes and Ground Delays Using a Route-Based Aggregate Model.* **Soler, Lluís.** s.l. : University Of California Irvine, 2013.
7. *A simplified theory of kinematic waves in highway traffic, part I: General theory.* **Newell, G. F.** 4, s.l. : Transportation Research Part B: Methodological, 1993, Vol. 27.
8. *Distributed sliding mode control design for a class of positive compartmental systems.* **Arneson, H. M. and Langbort, C.** s.l. : American Control Conference, 2008, pp. 2455-2460.
9. *Adjoint-based constrained control of Eulerian transportation networks: Application to air traffic control.* **Menon, P. K., et al.** s.l. : American Control Conference, 2004, Vol. 6, pp. 5539 - 5545.
10. *New approach for modeling analysis and control of air traffic flow.* **Menon, P. K., Sweriduk, G. D. and Bilimoria, K. D.** 5, s.l. : Journal of Guidance, Control and Dynamics, 2004, Vol. 27, pp. 737-744.
11. *Air Traffic Flow modeling, analysis and control.* **Menon, P. K., et al.** s.l. : AIAA Guidance, Navigation, and Control Conference, 2003.
12. *Aggregate flow model for air-traffic management.* **Sridhar, B., et al.** 4, s.l. : Journal of Guidance, Control, and Dynamics, 2006, Vol. 29, pp. 992-997.
13. *On the partial difference equations of mathematical physics.* **Courant, R., Friedrichs, K. and Lewy, H.** 2, s.l. : IBM journal of Research of Development, 1967, Vol. 11, pp. 215-234.
14. *Multicommodity Eulerian-Lagrangian Large-Capacity Cell Transmission Model for En Route Traffic.* **Sun, Dengfeng and Bayen, Alexandre M.** 3, Berkeley : Journal of Guidance, Control, And Dynamics, 2008, Vol. 31.

15. *The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory.* **Daganzo, C.** 4, s.l. : Transportation Research, Part B, 1994, Vol. 28, pp. 269-287.
16. *A Dual decomposition method for sector capacity constrained traffic flow optimization.* **Sun, D., Clinet, A. and Bayen, A. M.** s.l. : Transportation Research: Part B, 2011, Vol. 45, pp. 880-902.
17. *Total unimodularity and decomposition method for large-scale air traffic dell transmission model.* **Wei, P., Cao, Y. and Sun, D.** s.l. : Transportation Research: Part B, 2013, Vol. 53, pp. 1-16.
18. *Link Transmission Model for Air Traffic Flow Management.* **Cao, Yi and Sun, Dengfeng.** 5, s.l. : Journal of Guidance, Control, and Dynamics, 2011, Vol. 34.
19. **Federal Aviation Authorities.** National Severe Weather Playbook . [Online] <http://www.fly.faa.gov/PLAYBOOK/pbindex.html>.
20. **Airbus Flight Operations Support & Line Assistance.** Getting the grips with Cost Index. 1998. 2.
21. **m2p Consulting.** *The cost of delays & cancellations: Analysis & Means for Cost Reductions.* Dubai : AGIFORS, 2006.
22. **NYSERDA.** New York Home Kerosene Price Monitoring Program. [Online] 2014. <https://www.nyserda.ny.gov/Energy-Data-and-Prices-Planning-and-Policy/Energy-Prices-Data-and-Reports/Energy-Prices/Kerosene/Monthly-Average-Kerosene-Prices.aspx>.
23. **Air Berlin.** Airbus A320-200. [Online] 2014. http://www.airberlin.com/en-US/site/seatplan.php?seatTyp=A320_200&LANG=eng&.
24. **Miami-Dade Aviation Department.** Rates, Fees & Charges. 2013.