University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Post-Operational Analysis of Network Manager Rerouting Opportunities: Methodological Approaches and Impact

Supervisor
Prof. Luigi De Giovanni
University of Padova

Master Candidate
Filip Sotiroski

Student ID
2070135

Academic Year

2022-2023

Abstract

This master thesis explores the application of a simplified 2D approach for flight proposal matching to support strategic analysis of Network Manager (NM) Flight efficiency rerouting proposals uptake, implemented by the Group Rerouting Tool (GRRT) in EUROCONTROL's Enhanced Tactical Flow Management System (ETFMS). The study also proposes a methodology for post-operational analysis with the aim to assess the tactical effectiveness of the 2D approach and a strategic utilisation of the proposals in order to provide insight analysis on the uptake of routes and segments proposed by GRRT to Aircraft Operators (AOs).

The research adopts an approach of leveraging literature review to identify relevant concepts, and utilizing tools such as SQL, Python, and Power BI for data processing, analysis, and visualization. Data is collected from EUROCONTROL's data warehouse, and an Extract, Transform, Load (ETL) pipeline is established to create datasets for the matching process. Indicators and statistical methods are applied to evaluate the performance of the 2D approach in terms of route efficiency, flight duration, and acceptance rates.

The research aims to answer the research question regarding how to develop a new approach for flight proposal matching in post-operational analysis that achieves similar or better results than the current 4D algorithm applied for the uptake of the GRRT, while enabling strategic and tactical analysis to gain insights in the uptake of the reroute proposals.

The research aims to answer the research questions regarding the new approach of the flight proposal matching process while maintaining or improving its accuracy. Additionally, it seeks to identify the limitations and constraints of the current 4D algorithm used by the GRRT and proposes methods to address these issues in the development of a new approach. The study also explores how an analysis of flight proposal acceptance can be integrated into the matching process to gain valuable insights into the uptake of route proposals. Ultimately, the research aims to demonstrate how the new approach for flight proposal matching can enhance the understanding of flight proposal acceptance.

Moreover, a tool is provided for the aviation experts who can specify the dates, city-pairs, AOs and obtain related metrics and visualizations. The developed tool has been applied to compare against the matched flights from two AOs: one that confirmed to have used the tool for operations, and another whose tool usage remains uncertain. The analysis is done during a period of three Aeronautical information regulation and control (AIRAC) cycles spanning around three months.

Overall, this master thesis offers a comprehensive analysis of a 2D approach for flight proposal matching, combining theoretical exploration, practical implementation, and insightful findings to enhance understanding and decision-making in the field of air traffic management.

Index Terms — **Flight efficiency, air traffic management, business intelligence, aviation, performance evaluation, post-operational analysis, pattern matching, data engineering**

Contents

List of figures

List of tables

List of acronyms

5LNC	Five-Letter-Name-Codes
ACC	Area Control Center
ADEP	Departure Airport
ADES	Arrival Airport
AFTN	Aeronautical Fixed Telecommunication Network
AID	Aircraft Identification
AIRAC	Aeronautical Information Regulation and Control
ANSP	Air Navigation Service Providers
AO	Aircraft Operator
API	Application Programming Interface
ARO	Air Traffic Services Reporting Office
ATC	Air Traffic Controller
ATCU	Air Traffic Control Unit
ATFM	Air Traffic Flow Management
ATS	Advanced Traffic Services
BI	Business Intelligence
BPMN	Business Process Model and Notation
CANSO	Civil Air Navigation Services Organisation
CDR	Conditional Route
CHMI	Collaboration Human Machine Interface
CLI	Command Line Interface

COBT	Calculated Off-Block Time
CSV	Comma Separated Values
CTE	Common Table Expression
DCT	Direct Routing
DD	Decimal Degree
DMS	Degrees, Minutes, Seconds
DWH	Data Warehouse
EDA	Exploratory Data Analysis
ENV	Environment
EOBT	Estimated Off-Block Time
ETFMS	Enhanced Tactical Flow Management System
ETL	Extract, Transform, Load
F15	Field 15
FAA	Federal Aviation Administration USA
FEP	Fuel Efficiency Program
FL	Flight Level
FMP	Flow Management Position
FOS	Flight Operations System
FPL	Flight Plan
FPS	Flight Planning System
GRRT	Group Rerouting Tool
ICAO	International Civil Aviation Organization
ICARD	International Codes and Route Designators
IFPS	Integrated Initial Flight Plan Processing System
IFPZ	IFPS Zone

IFR	Instrument Flight Rules
JSON	JavaScript Object Notation
LCS	Longest Common Subsequence
LOBT	Last received Off-Block Time
NAT	North Atlantic Tracks
NM	Network Manager
NM	Nautical Mile
NMIR	Network Manager Interactive Reporting
NMOC	Network Manager Operations Centre
NMP	Network Manager Portal
NO	Network Operations
NOP	Network Operations Portal
Oplog	Operations Log
PCA	Principal Component Analysis
Post-ops	Post-operations
PRISME	Pan-European Repository of Information the Management of EATM
Regex	Regular Expression
TMA	Terminal Manoeuvring Area
VFR	Visual Flight Rules

1 Introduction

This master's thesis is the outcome of a graduation project undertaken as part of the Erasmus Mundus Joint Master Degree Program in Big Data Management and Analytics (BDMA). The project was conducted within the University of Padua from Italy. Specifically, the research was carried out in collaboration with the Operations Planning unit in EURCONTROL headquarters in Brussels, Belgium.

1.1 Thesis Context

In recent years, the volume of airplane traffic in Europe has been steadily increasing (a part from the COVID-19 pandemic), resulting in significant challenges for air traffic control and airport operations [1]. The surge in air travel is pushing several Area Control Centres (ACCs) to their maximum capacities, leading to congestion and causing numerous delays. The current situation demands a proactive approach from Aircraft Operators (AOs), who should prioritize flexibility in their routes to alleviate the strain on the existing infrastructure. By embracing more adaptable flight routes, operators can contribute to the better distribution of traffic and the reduction of congestion in the airways and sectors. This, in turn, will help reduce their costs, ultimately improving the efficiency and reliability of air travel across Europe.

Furthermore, with the gradual recovery from the COVID-19 pandemic (see Fig. 1.1) the aviation industry expects a significant rebound in air traffic. As travel restrictions

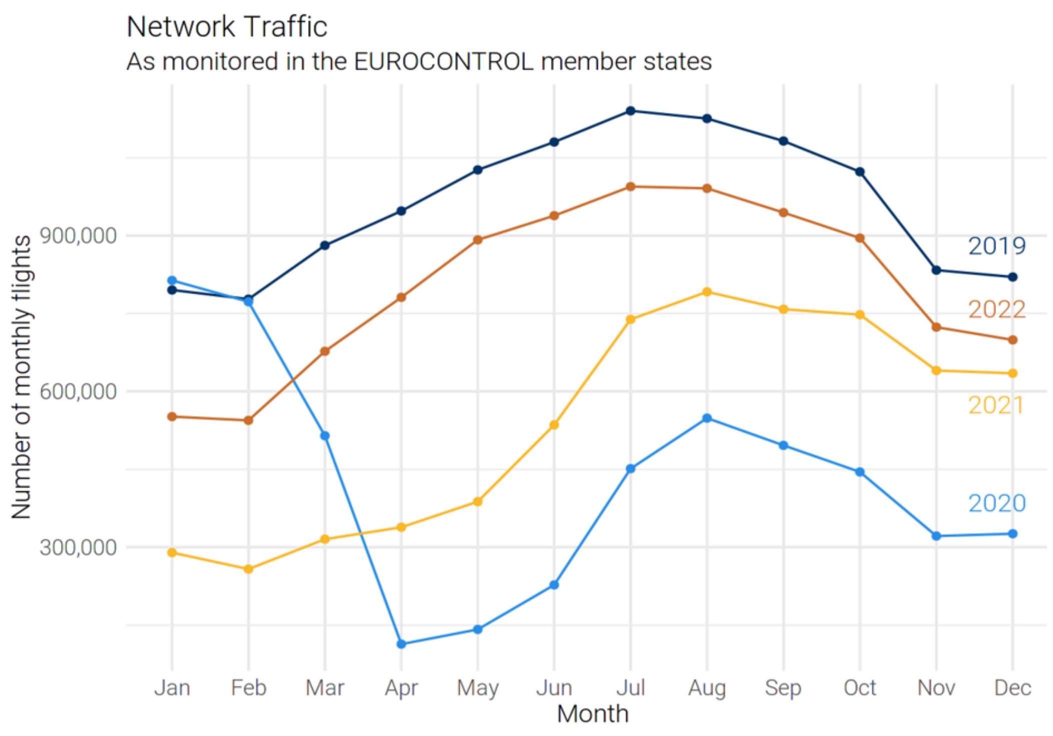


Fig. 1.1: Monthly flights in Europe post-COVID [1]

ease and passenger numbers grow, the demand for flights is anticipated to surge. While this rebound is a positive sign for the industry, it also poses additional challenges for managing air traffic.

> "On Friday 30 June 2023, the network recorded its highest number of flights (34,042) since the beginning of COVID - although still 8.5% below the record for the busiest day ever (28 June 2019), see Fig. 1.2" [2]

Considering the escalating air traffic and the rebound from the COVID-19 pandemic, it is imperative for AOs to prioritize flight efficiency by adopting a more flexible approach. By embracing flexibility in their operations, such as adjusting routes and schedules based on real-time traffic patterns and demands, AOs can enhance the overall efficiency of their flights.

EUROCONTROL, as the European Network Manager (NM) is responsible for the overall coordination and management of air traffic operations within the European airspace. One of its duties is to be responsible for the Air Traffic Flow Management

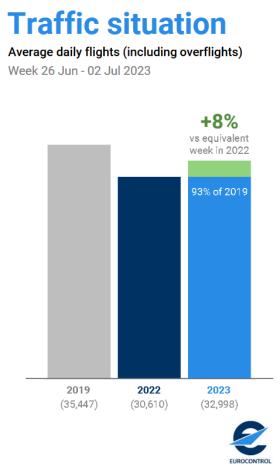


Fig. 1.2: Average daily flights pre and post-pandemic [2]

(ATFM). This involves monitoring air traffic demand, assessing capacity, and implementing measures to balance demand and capacity, such as implementing flow control measures, rerouting flights, and managing slots at congested airports. The Network Manager engages in strategic network operations planning, considering factors such as airspace design, route optimization, and capacity management. This includes developing long-term plans to improve efficiency, resilience, and performance of the air traffic management system. It also collects and analyzes data related to air traffic operations and performance. This includes monitoring key performance indicators, analyzing trends, and identifying areas for improvement in terms of safety, efficiency, and environmental impact.

Post-operational (post-ops) analysis plays a crucial role in aviation, particularly when it comes to flight reroute proposals [10]. It involves evaluating the performance and outcomes of proposed flight routes and their subsequent implementation. By conducting a thorough analysis of flight reroute proposals after they have been used, valuable insights can be gained to optimize future operations and enhance overall efficiency in the aviation industry.

AOs typically use specialized software known as flight planning systems (FPS) or flight operations systems (FOS) to create flight plans (FPLs). These software applications assist in the creation, optimization, and management of flight plans based on various factors such as aircraft performance, airspace constraints, weather conditions, and regulatory requirements.

Inside the scope of ATFM, operations planning and performance monitoring, a Group Rerouting Tool (GRRT) was developed by EUROCONTROL to provides all users, including AOs, with the ability to view alternative routing options within the network. This tool assists the AOs in selecting shorter and more efficient routes while also helping them identifying potential inefficiencies in their flight planning process. The GRRT operates using a dynamic route generator that also incorporates a historical database of previously flown routes. This enables operators to consider their selected business criteria while making route decisions.

The study proposed in the thesis explores the effectiveness of a 2D flight route matching algorithm for post-ops analysis. It is essential to ensure that the outcomes they produce are comparable with the 4D algorithm from GRRT. Moreover, this straightforward approach also enables the examination of strategic and tactical flight proposal analysis of the GRRT opportunities uptake by the AOs. Tactical analysis refers to the examination and evaluation of short-term, immediate actions or decisions taken to achieve specific goals within a limited time frame. In our context, it means analysing the proposals on a given flight date. On the other hand, strategic analysis involves a broader and long-term perspective. The aim behind this is to analyse the uptake of certain route proposals and their segments. Meaning, analysing where certain proposals of routes and specific segments from those routes were used in future flights.

The analysis that we propose has the potential to contribute to the ongoing development and refinement of the GRRT which is a part of the Enhanced Tactical Flow Management System (ETFMS) [11] in EUROCONTROL.

1.2 Research Approach

The approach consists of finding a similar logic that the ETFMS is using for matching the flight reroute proposals which could later be used for tactical and strategic analysis.

The algorithm that the GRRT system utilizes for proposal matching is using the 4D trajectory of the flight (three spatial dimensions plus time as a fourth dimension). The 4D trajectory is more precise than a 2D trajectory, which could be seen as a birds eye view of the 4D. Even, the 4D algorithm is different since it compares the whole flight trajectory, while the one based on 2D does not take into account the flight level (FL) and the flight speed. For the case of flight efficiency and post-ops analysis, we thus need to find a solution to use this flight route information later on for the analysis.

From the perspective of flight efficiency strategic analysis, the current 4D matching algorithm is more conservative when it comes to matching since it is constrained by a certain tolerance in flight level and geographic position. It is limited for performing post-ops analysis since it is not designed for that.

On the other hand, a 2D matching is point based, where a point is a pair (x, y), which is important since the flight reroute proposals are a series of flight plans which contain waypoints. Having an ordered list of points would mean that it would be easier to identify those proposals and use them for the analysis.

The aim of this thesis is to develop a new approach that can achieve similar or possibly better results than the current algorithm used by GRRT for matching flight routes. Additionally, this approach will enable historical analysis, allowing us to gain valuable insights into the actions and decisions of stakeholders. By achieving these objectives, this research will contribute to providing a better understanding of flight reroute proposal acceptance.

To achieve this end, we use the following steps:

Literature Review: We use the literature review to get a comprehensive understanding of the current state of knowledge and research on trajectory matching and exploratory data analysis. Different papers are consulted about the problematic to see which method can be useful. This will help us create a framework that we can then use for our approach.

Tools: As the task of comparing historical flight routes requires data cleaning, data processing and a full Extract, Transform, Load (ETL) pipeline, Oracle SQL [12] and Python [13] are used. Both regular expression (regex) patterns are used in both SQL and Python as well as some more advanced queries. From the processing side in Python, Pandas library [14] was used to apply different transformations to the dataset. In conclusion, all the findings and results of this research have been synthesized and presented in a concise and user-friendly format using Power BI [15]. The interactive tool developed for this thesis serves as a valuable resource for aviation experts, stakeholders, and AOs, providing them with the necessary insights and information to make informed decisions based on the research outcomes. By utilizing the output of this tool, these individuals can effectively analyze and interpret the data, facilitating improved decision-making processes in the field of Air Traffic Management (ATM).

Methodology: An ETL pipeline extracting data from the data warehouse was implemented in order to create a dataset. In the same pipeline, transformations are being done on the dataset and the algorithms are run on it. The results are then updated in multiple files depending on the number of Aeronautical Information Regulation and Control (AIRAC) cycles, AOs and city-pairs. All the results are then streamed towards Power BI, where users can interact with the data, analyze, and compare the results. The pipelines are automated, so for each new release of data, they will provide analyses that will be used in the future by aviation experts.

Evaluation of results: There are two types of analysis done for this task. Tactical analyses are conducted to examine the number of matches per flight, and strategic analyses are performed on the usage of routes and segments. For the first one, the analysis is done by using multiple AIRAC cycles to gather the data. They can be executed on data with higher granularity, for example: the number of matches for all the flights in those AIRAC cycles. They can go as low as analyzing matches of specific city-pairs on specific dates. Users can observe the matches done by our approach, the mismatches, the unique matches that occur etc. Value can be found in the reason why a certain company was picking a certain proposal (e.g., delay or route charges). For the strategic analysis we can find the difference in time for the adaptation of the new route and segments. In this case, it is important to limit the analysis to the city-pairs for different AIRAC cycles, in order to have a consistent size of the dataset. Aviation experts will later utilize the values extracted from the analysis as crucial input for making informed decisions.

1.3 Problem Statement

The research goal is to develop a new approach for flight route proposal matching in post-ops analysis that achieves similar results as the current 4D algorithm used by the GRRT in EUROCONTROL. The aim is to simplify the matching process while enabling tactical and strategic analysis, allowing insights into AO's uptake of the route proposals.

Data will be collected from EUROCONTROL's data warehouses (DWH), gathering the flight messages used in the GRRT, as well as aviation data (routes, points, segments, etc.). ETL pipelines will be set up to process this data in order to create datasets. These datasets will be used to apply transformations for performing the matching of flight reroute proposals.

To gain insight into the decision process, appropriate indexes will be defined and implemented as well as data analytics using statistical methods to evaluate the performance of the 2D approach in terms of route efficiency, flight duration, and acceptance rates. We could answer questions such as: How many route matches happened per day by an AO? What is the typical duration for an AO to use a proposed route generated by the tool? etc.

We can therefore propose this following research questions:

- How can the flight reroute proposal matching process be simplified while maintaining or improving the accuracy and efficiency of results?
- What are the limitations and constraints of the current 4D algorithm used by the GRRT, and how can these be addressed in the development of the new approach?
- How can analysis of flight reroute proposal acceptance be incorporated into the matching process to gain insights into AO's uptake of the route proposals?
- Will the new approach for flight reroute proposal matching enhance our ability to gather additional knowledge for flight reroute proposals?

One of the limitations that we need to be aware of is the quality of the data related to the flight network study. For example, dealing with flights from around the world, we could encounter faulty, duplicated data, as detailed in Chapter 5: points not belonging to certain routes, parent point problem, etc. We could also experience difficulties when working with flights that fall outside the geographical regions covered by the proposed routes, as this should be the flights from the Atlantic, flights to Africa, Asia, or anything else that is outside the NM area.

1.4 Thesis Structure

The thesis is organized in the following manner:

Chapter 1: Introduction

In the introduction given above, we have outlined the thesis context and introduced the basic concepts that will be developed in the thesis. We show the research approach that will be carried through the work. We introduce the problem statement where we define the research goal and research questions that we want to answer. We explain the thesis structure that serves as roadmap to navigate the reader through the matter.

Chapter 2: Literature Review

The literature review serves as the cornerstone for defining our approach for this topic. We analyze different papers and previous works. We define the key concepts that we will use for the future work as well as an understanding of the research landscape.

Chapter 3: Flight Plans

The main data elements needed for this thesis are explained, and an introduction is provided to non-aviation experts about flight plans and the types of flight plans that exist. Subsequently, the chapter elaborates on the process of how a flight plan is processed.

Chapter 4: Group Rerouting Tool (GRRT)

An introduction of the GRRT is made together with the different platforms through which AOs can get and evaluate the flight reroute proposals.

Chapter 5: Data - Network and Messages

The chapter provides an explanation of the databases that were utilized to obtain network data and flight messages from the ETFMS. Additionally, it provides an overview of the pipeline process and the parser logic employed for field 15 of the flight plan. This specific field holds significance for the analysis developed in the thesis.

Chapter 6: Matching Methodology

The chapter contains an explanation of the current 4D approach used by the GRRT, as well as the proposal of the 2D approach which is used for strategic and tactical analysis. This section will provide detailed explanations and step-by-step breakdowns of both algorithms.

Chapter 7: Results and Findings

The analysis of the algorithm and the comparison with the results of the existing tool are explained in this chapter. This considers as well the strategic and tactical analysis of segments and routes proposed by the GRRT.

Chapter 8: Conclusion

In this last chapter the performed work is summarized, and it is evaluated how the research goals and are fulfilled. Possible improvements are discussed and proposed for future work.

Appendix A: Additional Data Tables

The appendix contains the datasets presented as tables, each comprising a name, description, and an example row of data within a column.

Appendix B: Code Listings

A portion of the source code is displayed in the appendix. It includes the SQL query responsible for extracting data from the data warehouse. The regex patterns are also provided, along with the Python code for identifying subsets of waypoints within a route. Furthermore, the appendix presents the pseudocode for the connecting point match.

Appendix C: Technical Explanation

To have better understanding of the structure of the field 15 and the rules that are followed when being filed, a brief explanation is shown in the appendix. Moreover, the flight plan management system is explained with a diagram.

2
Literature Review

In this chapter, we provide a comprehensive literature review to explore and synthesize the existing body of knowledge related to trajectory matching in aviation, exploratory data analysis of geospatial trajectory data and string pattern matching. This review serves as a foundation to our research since it provides insight to the current state of the research and it guides us to the formulation of our research questions and objectives.

We are trying to observe the current possibilities and techniques to create trajectory matching of flights, as well as developing a system that will use the data taken from those matches for further analyses. The main challenge with the trajectory matching is to find an approach to label the trajectories and use them. There are many approaches, but the most common one is to use 2D or 3D/4D trajectories of flights. We are also exploring a way to create the strategic and tactical analysis of the proposals, to be used by aviation experts.

2.1 2D vs. 3D Perspective in Aviation

In [16], the author explored the use of 3D perspective views on flat screens for situational awareness tasks. Even though the paper is based on the human-machine interaction of 2D and 3D perspectives, it provides an initial understanding of the key differences between 2D and 3D views in aviation. The paper describes experiments conducted with simple block stimuli and complex terrain stimuli, showing that 3D views are superior for tasks involving shape understanding, while 2D views are better for tasks requiring precise relative position judgments. The utilization of 2D perspective views is favored in this context as they effectively display the points outlined in the flight plan.

2.2 Spatial Trajectory Matching

In [17], the authors present two trajectory clustering methods and their application to airspace monitoring. Trajectory clustering is a method of grouping similar flight trajectories based on their spatial and temporal characteristics. In the cited work, they re-sample the trajectories and obtain time series of equal length, to be treated as points in a geometric space.

The paper presents two algorithms: waypoint-based clustering (2D) and clustering via component analysis (3D), which are used to identify clusters of flight trajectories in the airspace. The goal is to know whether the use of clustering in trajectories would be sufficient to cluster the routes and use it for strategic analysis.

The paper introduces a waypoint-based trajectory clustering algorithm, where the waypoints are the points that guide the aircraft along its final route. It is of particular interest for observation since the flight plans are based on the sequence of waypoints. The algorithm uses the two-dimensional x and y position of the aircrafts. As stated in the paper, the algorithm is "an efficient way to determine the compliance of flown trajectories with published procedures".

From this algorithm, we can conclude that even though a clustering method was proposed for route monitoring, it gives us good direction where the work should be heading, but it has crucial drawbacks. The main drawback of this method is its limited consideration of trajectories that deviate from the waypoints, causing them to be classified as outliers even if they may be similar for specific airspace conditions, leading to the exclusion of many potentially relevant flight trajectories. Therefore, it is difficult to have a precise accuracy on flights that pass by closely to a certain waypoint. In fact, the paper is mainly focusing on finding the main flows instead of each individual flight route.

The second algorithm: "Trajectory-based clustering via component analysis" is using the 3D trajectory of the flight. The algorithm uses dimensional augmentation, principal component analysis (PCA) and clustering diagram. Similarly, to the first algorithm, we can draw the same conclusions. The algorithm is not waypoint based and is detecting a higher number of outliers. As the other algorithm, it is more useful to cluster the main flows of traffic.

Both algorithms are applied on a small scale, only analyzing arrivals approaches to the San Francisco airport (KSFO) and they do not show the accuracy of the clustering. It is unknown how this will work on a full flight observation with departure, cruising, and arrival on the whole European airspace. However, the algorithm uses both 2D in one and 3D coordinates in another algorithm which gives enough insight to understand which is more convenient for our task. Since clusters are labeled, we could use this logic of labeling routes for the post-ops analysis.

So far, reference [17] focuses on distance based trajectory clustering while considering the trajectories as continuous function of time. Other papers [18, 19, 20] propose trajectory distance measurements for road network based distances. The underlying assumption is that the paths align with the layout of the road network, which is the case in aviation. However, the accuracy depends on the precision of the devices that track the objects, GPS or radars in aviation. Therefore, this might be an issue when working with very dense network as we have in Europe.

In [19], clustering trajectories on road data are proposed as well as segment-based clustering and create a similarity graph. Trajectory-based clustering aims to group trajectories that share common parts of the road network, while segment-based clustering groups road segments traveled together.

In [18], the challenge of assessing similarity between trajectories of moving objects in the context of road networks is addressed. It introduces two concepts of similarity: network distance and time characteristics. Based on these concepts, the authors propose distance measures *Dnet* (network similarity) and *Dtime* (time-based similarity). The trajectories are decomposed into sub-trajectories and indexed using M-trees. On another aspect, its applicability may be limited in large spatial environments with many nodes such as aviation network, because of the distance preprocessing step that reduces computational costs during query processing. In aviation networks, implementing and maintaining two separate M-trees may introduce additional complexity in the system.

Yet, it is questionable how these approaches will fit with 3D trajectories as [18, 19, 20] use city network car traffic data. On the other hand, if we use the 2D trajectories of flights we could manage to find these papers useful.

In [20], authors argue that traditional density and Euclidean distance measures are inadequate for road-network aware applications. They introduce the NEAT framework, which addresses these limitations and aims to efficiently cluster trajectories for location-based services. The NEAT framework consists of a three-phase process: trajectory partitioning, merging, and clustering. The approach uses road intersections as initial partitioning points and clusters trajectory fragments based on their continuity within traffic flows. NEAT utilizes shortest path distance instead of Euclidean distance for proximity measurements. In general, the framework is a comprehensive solution for road-network trajectory clustering.

2.3 Pattern Matching

When we engage in the task of comparing multiple routes, the process of pattern matching becomes essential. This involves identifying and aligning the corresponding elements or features within each route.

String pattern matching is a fundamental technique extensively employed in computer science for diverse applications. It serves as the cornerstone for tasks such as text search in search engines, DNA sequence analysis in bioinformatics, lexical analysis in compiler design etc.

In order to evaluate the impact of this technique in our work, we first need to understand the content of the field 15 from the flight plan. It is a description of a flight route made of a string of different elements shown as words. Each word has its own meaning and represents a different element. The filed version of the field 15 in the flight plan contains the minimal number of elements (waypoints, routes) to explain how the aircraft will fly. We can either start from this minimal field 15 or expand it in order to have all the flight plans where the aircraft will fly. From here, we can see how far we need to decompose the field 15 in order to match multiple routes, as we will further explain in Section 3.2.

In view of the content of field 15, our main task can be associated to the concept of looking for similarities between strings, that has been used extensively in biological data for DNA alignment. One of the first applications of dynamic programming for biological sequences is explained in [21]. We can derive knowledge from the biology domain since we are dealing with a similar problem. Instead of DNA codons we are dealing with waypoints. The ability to see the number of matches, mismatches and gaps can be used when comparing two flight plans. We can determine the number of waypoints that need to be aligned between two given points in order to assess whether the field 15 strings match. With the introduction of a scoring matrix as in [22], we could penalize or award routes that have missing points when comparing routes. On the other hand, inspired by [23], we could calculate the distance of the flight plans. The drawback which does not enable to use this last method is that we need to have equidistance strings for the algorithm to work. On the other hand, it will require some sort of transformation of each waypoint into an encoding that is understandable by the algorithm.

Inspired by [24], the Basic Local Alignment Searching Tool (BLAST), extensively used in bio-informatics for sequence searching, can serve as a fundamental concept on which we can build the tool. This would mean a creation of a comprehensive flight plan database, with a system that will capture the relevance of waypoint sequences and a development of an algorithm that is capable of identifying matching and similarities with other flight plans. With the proper statistical measures of significance and visualization techniques, the concept can be used for better understanding the flight plans used by the AOs. While BLAST serves as inspiration for flight plan matching, there are many challenges. Flight plans involve a different type of data with unique characteristics, such as waypoints, and routes. Directly applying BLAST to flight plans would require significant modifications to accommodate the differences in data structure and semantics. Regardless of the approach, using a minimized or expanded version of field 15 will yield distinct results. Consequently, opting for an expanded version would be more advisable to ensure accurate outcomes. Flight plan databases are different in structure and size, compared to efficiency to search large sequence databases quickly. As noted from [21, 22], the alignment model does not suit the specifics of aviation data. Finally, the purpose of our work is to find exact match, while we are less interested in sequence similarities, but could be useful for future work to detect common routing patterns, deviations, anomalies, or optimizing route planning.

Even though both alignment methods from [21, 22] and Longest Common Subsequence (LCS) involve comparing sequences, they have different goals and ways of working. The alignment methods are more comprehensive. Aside from the matching parts, they look at the places where they were added, removed, changed. On the other hand, LCS focuses on finding the longest shared part without paying too much attention on the things that are missing or different.

While there are many different LCS available like [25, 26], they all have different behaviours and various applications. Since flight routes on the same city-pair are expected to have relatively few differences, faster algorithms are available [27]. The paper uses the same logic of edit graph as in [21, 22], but for finding the LCS which results in $\mathcal{O}(ND)$ time where N is the sum of the lengths of two strings and D is the minimum edit script for the two routes, and $\mathcal{O}(N)$ space, which is common for edit graph algorithms. In our case, D is small and makes this LCS algorithm more suitable for flight matching problem. With LCS, we can find the identity percentage from which we can derive if the match is equal (100% identity). However, in our case, where we specifically require a strict and complete exact match between sequences, the LCS approach may not fully meet that criterion.

The LCS is used in differential file comparison algorithms [28]. It can be adapted and used in flight plan matching to identify changes, differences, and similarities between different versions of flight plans. The referenced paper inspires us to use this logic to create the analysis of the segments between flight plans. LCS can serve to compare segments proposed in the flight reroute proposal with those outlined in the initial flight plan.

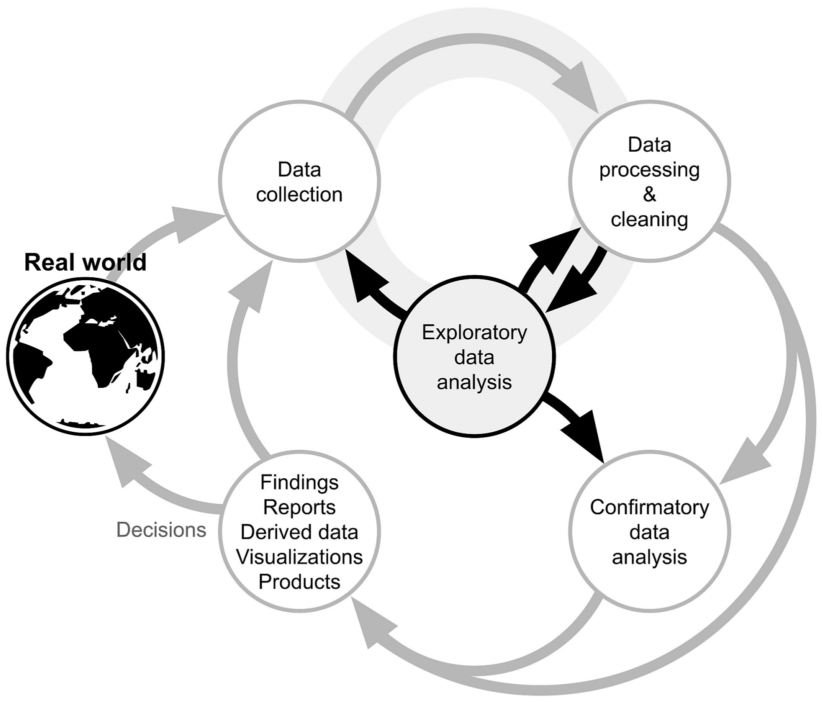
2.4 Exploratory Data Analysis

In [29], a comprehensive exploration of techniques and challenges in analyzing large-scale movement datasets is presented. It is of outmost importance to create a foundation for the post-ops analysis of the flight datasets.

Understanding the context of the data is essential, such as identifying gaps in data collection, potential data quality issues, and variations in spatial and temporal coverage. This will require a broader understanding of the data that will be used since aviation data are prone to change.

The cited paper explains the importance of exploring patterns within the set of extracted trajectories and events. This will be done through the data analysis for both strategic and tactical analysis of the proposals. The final exploration step focuses on identifying anomalous movements or spatio-temporal events that challenge preconceived assumptions about the dataset. Identifying unusual patterns is crucial for gaining insights into unique or potentially erroneous behaviour.

The paper offers a detailed examination of exploratory data analysis (EDA) as well as technical precision displayed throughout large-scale movement dataset. It covers essential topics such as data preprocessing, trajectory extraction, event identification, and outlier detection. The principles and workflow will be applied in the approach proposed in the thesis after completing data collection, processing, and cleaning, as well as during the strategic and tactical data analysis. The high-level diagram of EDA workflow for movement data can be seen in Fig. 2.1.



Data collection
Data processing & cleaning
Real world
Exploratory data analysis
Findings Reports Derived data Visualizations Products
Decisions
Confirmatory data analysis

Fig. 2.1: EDA workflow for large-scale movement data analysis [3]

2.5 Concluding remarks

In distance based trajectories [17, 18, 19, 20], most of the papers that have been published are focused on 2D trajectory clustering since they are limited to datasets that contain 3D or 4D trajectories of flights. In particular, [17] applies the method to aviation routes. It aims to cluster the trajectories to gain knowledge on which waypoint they fly above. As we have direct access to all the flight plans (which contain the waypoints), and we need to identify matches between them, we can avoid trajectory clustering for performing initial post-ops analysis.

The clustering methods discussed in [18, 19, 20] have the potential to be adjusted and used in aviation to match flight paths. They are important for aviation because road networks and airway systems are similar and we also need to study and improve flight paths.

To find the trajectories that match the flown trajectories with published procedures, it is better to use a 2D algorithm. The flight plans are based on waypoints that are predefined geographic locations that the aircraft must pass through to reach their destinations. The most fundamental point when reviewing papers about trajectory clustering is that we are trying to compare planned routes (flight plans) with actual trajectories of the flight which might not follow the flight plan. It is unfeasible to compare actual flight trajectories that deviate from their originally planned flight paths if we want to make a post-ops analysis on the flight plan proposals.

If we ignore the vertical dimension of two flight trajectories that share the same flight route, we will obtain the same flight trajectory. This enables the use of methods with better accuracy when comparing two flight plans, given that 2D information related to waypoints is taken into account to the purposes of the analysis considered in this thesis. Doing this, we will be able to identify those routes since they will clearly pass through an ordered set of points.

Drawing inspiration from DNA alignment techniques, we have explored the applicability of string similarity in flight plan analysis. LCS-based flight plan matching can provide insights into shared routing segments, enabling aviation professionals to identify common waypoints and potentially detect discrepancies or deviations between flight plans. An application of LCS is possible, but it is still beyond the exact matching that we are looking for. However, we will follow the approach [27] for the data catalogue logic of the strategic analysis for the segment usage. Currently, the utilization of pattern matching techniques proves advantageous for our operations. However, it is important to note that these techniques may not be imperative for the primary objective of exact matching of flight plans. While our immediate focus is on exact match determination through straightforward string-to-string comparison, the groundwork laid also holds the potential for uncovering routing patterns, detecting deviations, and optimizing route planning. These aspects are important considerations for future tasks.

Lastly, a foundation for data collection, processing, and cleaning will be used for analysing a large-scale spatial dataset.

3 Flight Plans

In the European airspace, it is mandatory for pilots planning to depart from, arrive at, or fly over the International Civil Aviation Organization (ICAO) EUR Region known as the Integrated Initial Flight Plan Processing System (IFPS) Zone (IFPZ) shown in Fig. 3.1, to submit a flight plan to EUROCONTROL's Network Manager Operations Centre (NMOC). The NMOC receives flight plans from AOs, which it validates and corrects (if necessary), and distributes to the relevant air navigation service providers (ANSPs) and operational partners [4].

To ensure efficient management of flight plans at a European level, the NMOC is a centralized function responsible for processing and distributing flight plans. This function is supported by the IFPS, a central system that collects initial flight plans, processes associated messages, and distributes them to ANSPs.

The regional centralization of flight plan management significantly contributes to the improvement of both the consistency and predictability of flight demand information. It simplifies operations, enhances flight efficiency, and reduces overall transaction costs for ANSPs and airspace users alike.

IFR (instrument flight rules) pilots must file a flight plan for every flight they undertake. IFR flight plans are an alternative to filing a flight under visual flight rules (VFR), where the pilot is ultimately responsible for navigation, obstacle clearance, and traffic separation. On IFR flight plans, pilots are normally assisted by air traffic controllers (ATCs). In general, most commercial air traffic and all scheduled air carriers operate

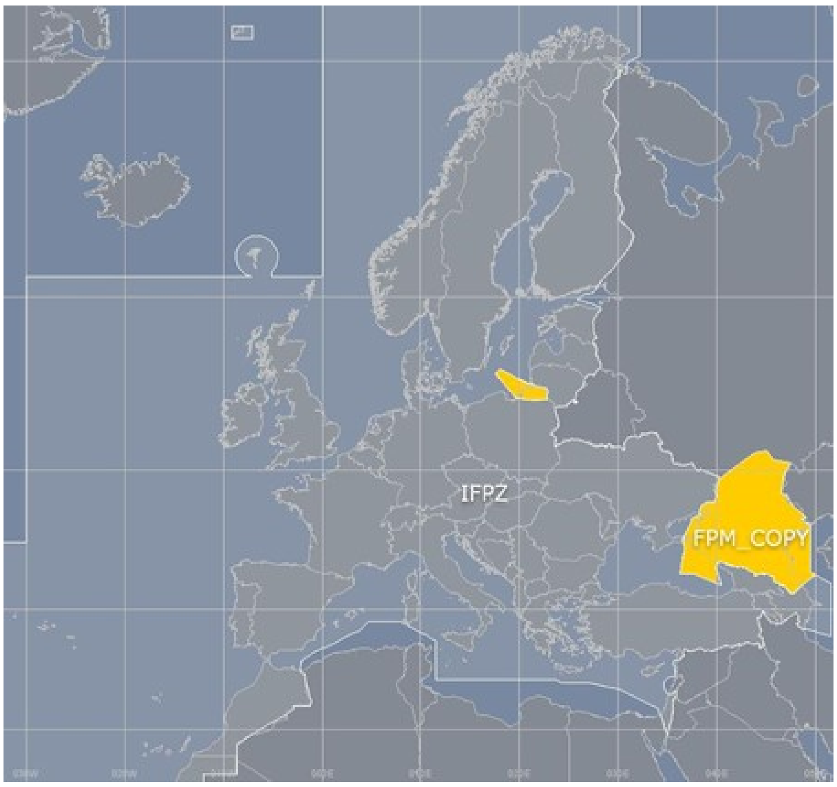


Fig. 3.1: Map of IFPS Zone (IFPZ) [4]

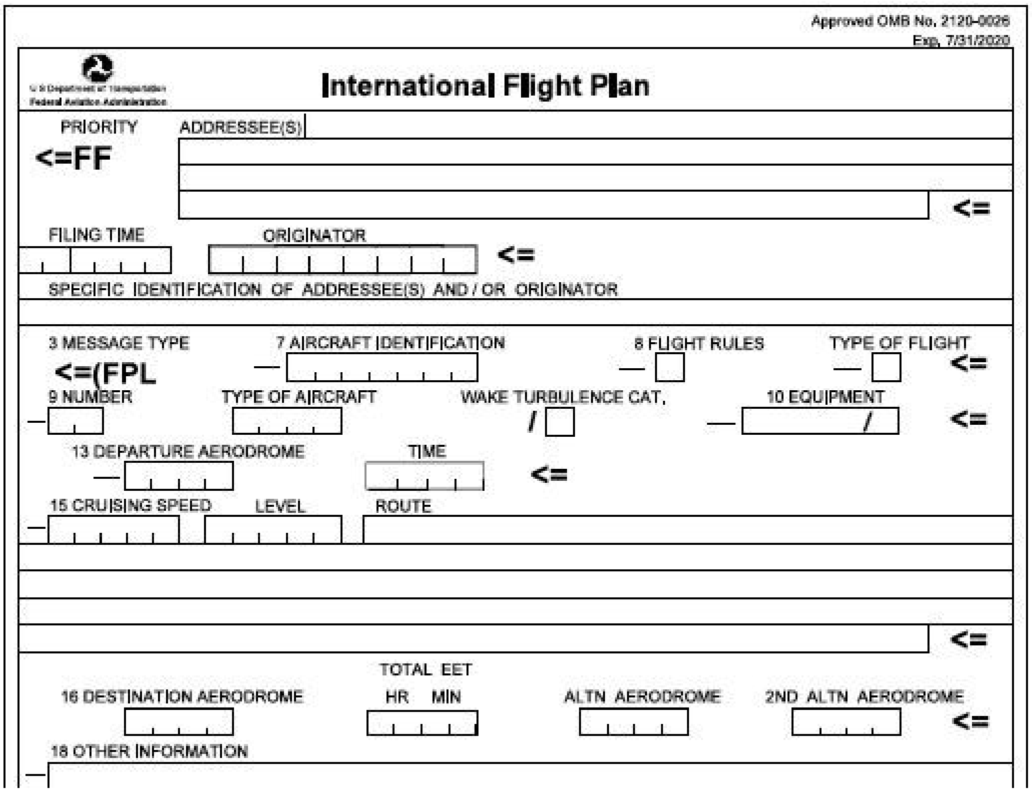
exclusively on IFR flight plans, while aircraft providing sightseeing flights, aerial photography, or lift services for parachute jumping usually operate on VFR or a mix of IFR/VFR flight plans.

Multiple interfaces are available for accessing the flight plan filing and management service, as well as for interacting with the IFPS. They can be accessed through, e.g., the Aeronautical Fixed Telecommunication Network (AFTN), the Network Operations Portal (NOP) or using the EUROCONTROL's Network Manager business-to-business (NM B2B) Application Programming Interface (API).

3.1 Flight Plan Structure

A flight plan is a an official and detailed document imposed by ICAO outlines the intended route of an aircraft for a particular flight. It is created by the AO before the aircraft takes off and serves as a comprehensive guide for the flight crew and ATC. The flight plan is the most important data element for our work since the reroute proposals are written as a field 15 (F15) or item 15 of the flight plan which we need to use.

Most flight plans are sent to the IFPS in ICAO FPL2012 [5] format (but also in ADEXP, via XML NM B2B [30] or FIXM FF-ICE [31]). In IFPS, the flight plans are converted to an ADEXP format. Here is a part of the ICAO FPL2012 format, Fig. 3.2, that every AO needs to fill and send to the NM, which in case of flights in Europe is EUROCONTROL. There are different fields that need to be filled such as: flying time, message type, aircraft, equipment, airport departure, alternative aerodrome (in case they are unable to land on the primary airport), route etc.


Approved OMB No. 2120-0026
Exp. 7/31/2020

U.S. Department of Transportation
Federal Aviation Administration

International Flight Plan

PRIORITY ADDRESSEE(S)
<=FF
<=

FILING TIME ORIGINATOR
<=

SPECIFIC IDENTIFICATION OF ADDRESSEE(S) AND / OR ORIGINATOR

3 MESSAGE TYPE 7 AIRCRAFT IDENTIFICATION 8 FLIGHT RULES TYPE OF FLIGHT
<=(FPL — — — <=

9 NUMBER TYPE OF AIRCRAFT WAKE TURBULENCE CAT. 10 EQUIPMENT
— / — / <=

13 DEPARTURE AERODROME TIME
— <=

15 CRUISING SPEED LEVEL ROUTE
—
<=

TOTAL EET
16 DESTINATION AERODROME HR MIN ALTN AERODROME 2ND ALTN AERODROME
<=

18 OTHER INFORMATION
—

Fig. 3.2: Snippet of flight plan using the ICAO format [5]

3.2 Field 15

The field 15 is an item from the flight plan where the AO files the route on which it has intention to fly. It is composed of three elements: cruising speed, cruising level and route description. Details on how to file the field 15 are given in the Appendix C.1.

Here is an example of the field 15 for the flight plan from Bucharest (LROP) to Amsterdam (EHAM), shown in Fig. 3.3:

```
N0450F380 SOKRU1K SOKRU DCT IBINU DCT NORAH DCT ODNEM DCT ELMEK
DCT VEXIL/N0450F400 DCT KATCE DCT NORKU NORKU2A
```

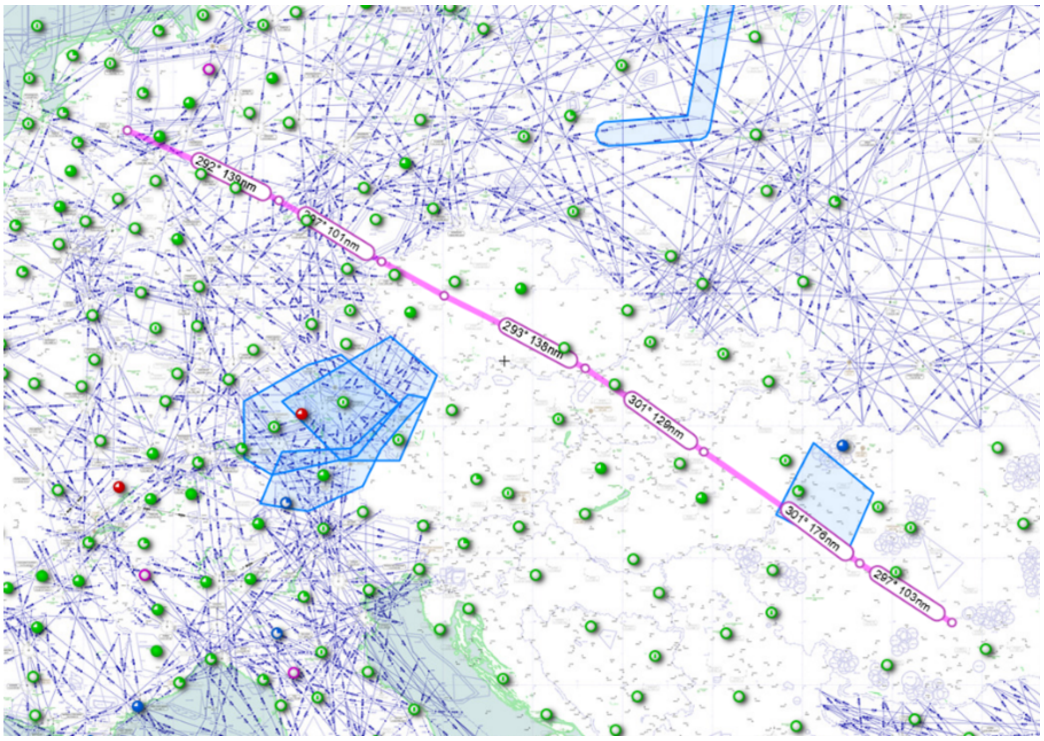


Fig. 3.3: Flight from Bucharest (LROP) to Amsterdam (EHAM) in SkyVector [6]

The field 15 is an ordered sequence of waypoints and other elements, see Section 5.3, so when we read it from left to right, it means that it will use that series of points in order to reach the destination. The field 15 can undergo changes before the departure depending on the delay, sector occupancy, weather etc. The filed flight plan serves as a reference for air traffic controllers to manage air traffic flow, provide routing guidance, and assist in search and rescue operations if necessary. In the case of EUROCONTROL as an NM, different assessments of the traffic are done on the airspace in order to see the sector load and use of regulations on those sectors, as well as how the ACCs should prepare for future traffic or manage the configurations of their sectors.

The field 15 of the flight plan is the desired route on which the AO wants to fly. During the flight, ATCs might provide instructions to adjust the aircraft's route for various reasons, including avoiding other air traffic, adverse weather conditions, or airspace restrictions (military areas). We can showcase this by using EUROCONTROL's System for Traffic Assignment and Analysis at Macroscopic Level (SAAM) software [32]. On Fig. 3.4 and Fig. 3.5, you can find the initial flight trajectory (in green) on which the flight was supposed to fly and the actual trajectory (in red). Fig. 3.4 shows the 2D view of the flight, Fig. 3.5 shows the profile of the flight (3D). Our only focus for this work is on the intended flight trajectory, which is filed in the flight plan.

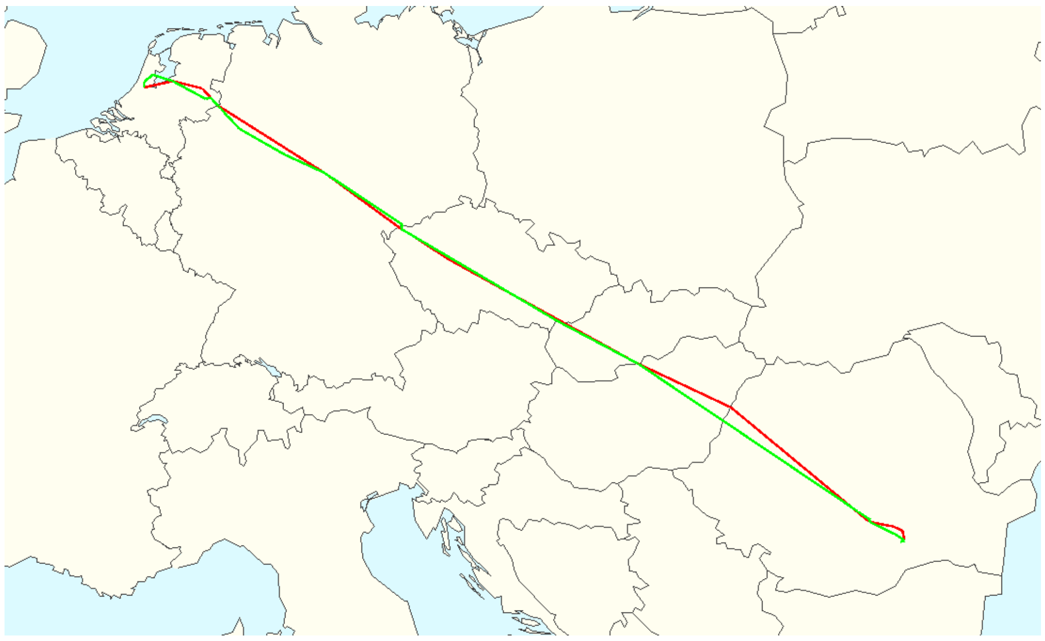


Fig. 3.4: Initial (green) vs. actual (red) 2D trajectory, flight from Bucharest (LROP) to Amsterdam (EHAM)

3.3 Flight Plan Management

As mentioned in Chapter 1, there is a process that the flight plan undergoes once it is sent to the Network Manager (NM). In this section, you can find a brief high level explanation of the flight plan Management. A more detailed overview of the process with the communication between all of the involved, a Business Process Model and

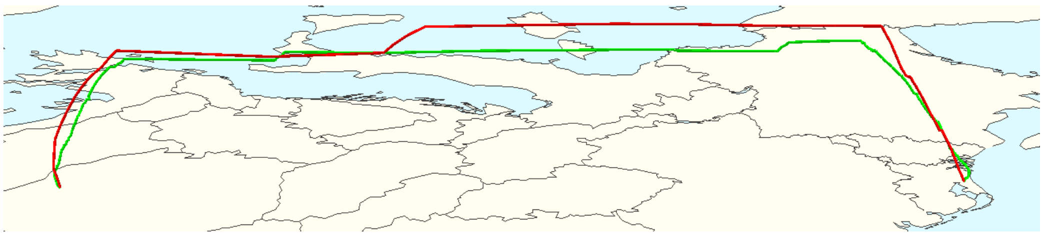


Fig. 3.5: Initial vs. actual in 3D trajectory, Flight from Bucharest (LROP) to Amsterdam (EHAM)

Notation (BPMN) can be found in the Appendix C.2, where the process is thoroughly explained.

The flight plan may be filed at any time but must be filed between 3 and 120 hours prior the estimated off-block time (EOBT) [4]. The flight plan passes through many phases once it is filed by the AO. A communication chain involves the NMOC, Air Traffic Control Units (ATCUs) and the AO for the flight plan to be correctly submitted, processed, validated and distributed. This is done to ensure the safety, effective coordination, monitoring, and execution of the flight plan throughout its journey.

The manual [4] mentions as well that the non-modifiable key fields are: aircraft identification (AID), aerodrome of departure (ADEP), aerodrome of destination (ADES), EOBT. In case there is a change in any one of them, the flight is cancelled and the AO will need to re-file a new flight plan. In case of a modification of a flight plan (new route), the key fields must be identical as the previous message as well as the EOBT time. The rules regarding the modification of the flight plan are crucial, as various changes can occur to a particular flight before its departure. These changes may include altering the flight level, adjusting the route, or even changing the aerodrome. In the case of a crucial change e.g., modification to the aerodrome, it necessitates the filing of an entirely new flight plan. Once a message is submitted and acknowledged to the IFPS for processing, a copy is sent to the ETFMS. The ETFMS analyzes the flight for any applicable flow regulations. The IFPS and ETFMS are separate systems, and messages must be acknowledged by the IFPS before being transmitted to the ETFMS for flow regulation application. In Fig. 3.6 from the IFPS user manual, we can clearly see the distribution of the flight plan through different systems.

Here is a brief explanation of Fig. 3.6: The flight plan must originate exclusively by the AO and it could be sent directly to the IFPS or through an Air Traffic Services Re-

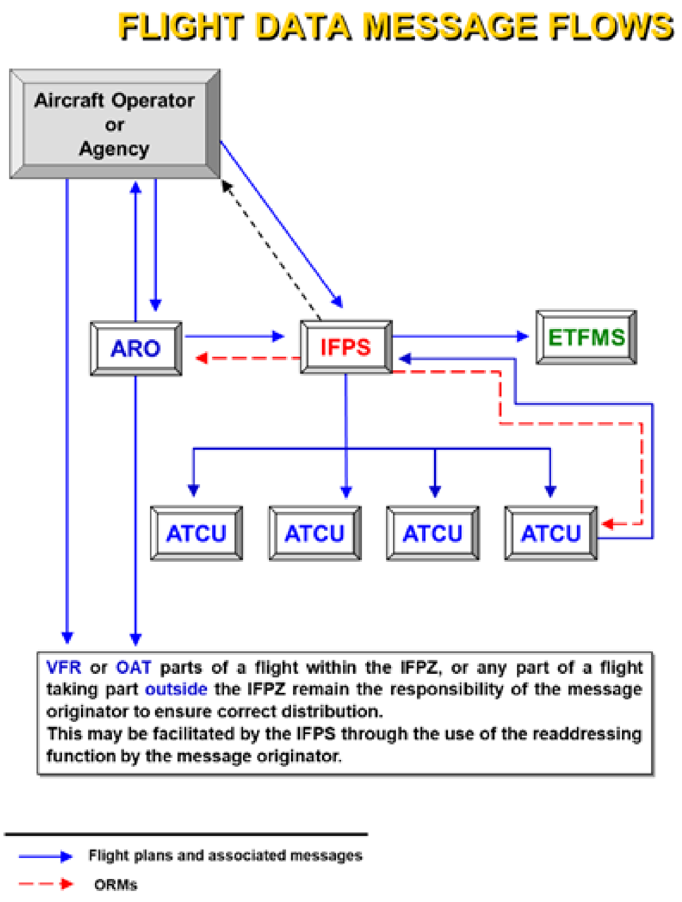

FLIGHT DATA MESSAGE FLOWS
Aircraft Operator or Agency
ARO
IFPS
ETFMS
ATCU
ATCU
ATCU
ATCU
VFR or OAT parts of a flight within the IFPZ, or any part of a flight taking part outside the IFPZ remain the responsibility of the message originator to ensure correct distribution.
This may be facilitated by the IFPS through the use of the readdressing function by the message originator.
Flight plans and associated messages
ORMs

Fig. 3.6: Flight data message flows between systems [4]

porting Office (ARO) which then forwards the flight plan to the IFPS. Once it reaches the IFPS, using all the parameters, including the ones indicated in the field 15, such as: route, aircraft performance, EOBT, ADEP, ADES etc., the system creates a 4D profile of every flight that plans to operate within the IFPZ. The purpose of this 4D profile is to find the intersection of the airspaces where the flight is going to fly through, and to obtain the addresses of the relevant ATCU to which the flight plan should be sent. 48 hours before the flight, the IFPS sends a copy of the flight plan to the ETFMS in order for them to apply the appropriate pre-tactical and tactical operations for that flight.

4
Group Reroute Tool System

The NM utilizes a tool called the Group Rerouting Tool (GRRT), sometimes referred to as the Route Opportunity Tool. The GRRT is ran multiple times throughout the day by the system to identify flight plans that could potentially benefit from an improved route. Once these routes are identified, the NM presents them to the AOs through different channels.

In 2008, the Flight Efficiency Plan [33] was adopted. The creation of this plan meant that new systems and tools should be integrated into the already existing tools that EUROCONTROL was using. The plan was originally signed by EUROCONTROL, ICAO and Civil Air Navigation Services Organisation (CANSO) as a reaction of the fuel crisis that year. The Fuel efficiency program (FEP) advocated for a collaborative approach involving airlines, air navigation service providers, airports, civil and military authorities of various states, and EUROCONTROL. The aim was to collectively implement operational measures that could result in immediate fuel savings. The five action points of the FEP [33] are:

1. Enhancing European en-route airspace design through annual improvements of European Advanced Traffic Services (ATS) route network;
2. Improving airspace utilisation and route network availability;
3. Efficient Terminal Manoeuvring Area (TMA) design and utilisation;
4. Optimising airport operations;

5. Improving awareness of performance.

As a consequence of the FEP, GRRT was created as a tool to face this problem. The GRRT is a component of the overall efforts to optimize Route Network Utilization. It analyzes flight plans and considers factors such as airspace capacity, traffic flow, and operational considerations. The GRRT identifies routes that can benefit from improved airspace utilization.

The objective of route network utilization is to reduce delays, improve flight paths, and make efficient use of available airspace. This enhances the efficiency of air transportation, benefiting both airlines and aviation authorities. It ensures smoother operations and cost-effectiveness for the industry as a whole, ultimately benefiting passengers as well [33].

4.1 Route Opportunities

The GRRT tool is implemented in the ETFMS, capable of recalculating routes based on predetermined criteria. The GRRT system is further explained in the Network Operations (NO) Flight Efficiency Manual [10].

The aim is to convert various cost criteria into a standardized unit of measure. By a weighted sum of the converted values for each criterion, a unique value called the Total Cost of the flight is obtained. Different criteria are considered when calculating the cost of a route. The system provides default weights for each criterion, allowing for adjustments to the balance between them. These criteria are pre-programmed in the ETFMS.

As stated in [10], the criteria are:

- Delay: Departure delay of the alternative. The user specifies the cost of each minute of delay incurred by the alternative. The user can also specify a limit on the deterioration of the delay;
- Flying Time: (total elapsed time). The user specifies the cost of each minute of flying time via the "cost/minute" field. The user can also specify a limit on the deterioration of the flying time;
- Route Length: The user specifies the cost of each nautical mile (NM) of route length via the "cost/NM" field. The user can also specify a limit on the deterioration the route length;

- Suspension: The user specifies a fixed cost to be added to the total cost of the alternative due to flight suspension, via the "cost if suspended" field;
- Overload: The user specifies a cost per minute of overloads caused by the alternative on all other flights;
- Fuel Cost: The user specifies the estimated price of fuel per ton via the "cost/ton" field. The user can also specify an upper limit to the fuel consumption, in kg;
- Take-Off Weight Factor (TOWF): a way to estimate route charges. User specifies the TOW (take-off weight) of the aircraft as a percentage of the maximum take-off weight of the aircraft type.

Route opportunities are generated by the tool considering the criteria. As stated in [10], a route opportunity is seen as:

- Shorter route close to the initial trajectory using missed short cuts;
- Shortest route not considering the initial route, can be far away from the filed route;
- Better utilisation of free route airspace or night network;
- Better utilisation of Conditional Route (CDR) network.

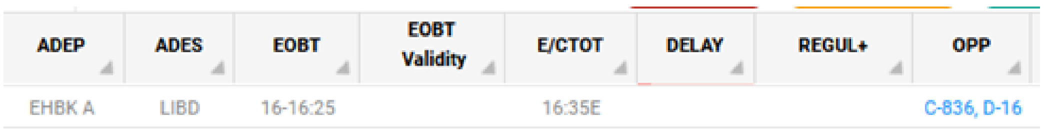
The structure and content of the reroute opportunity (REROUTE) message is explained in detail and can be found in Section 5.1.

4.2 Interaction with Group Reroute Tool Proposals

Both EUROCONTROL operators and AOs can interact with the GRRT, while the aviation experts can interact with the GRRT tool through the ETFMS by creating templates, changing parameters, adding constraints. The ETFMS can also visualize the routes as well as simulate the proposals. On the other hand, AOs have a different view. Using the EUROCONTROL's Network Manager Portal (NMP), AOs can have a visual representation of their flights and can see if there are any reroute opportunities for their flights. For example, let us see the flight in Fig. 4.1 from Maastricht (EHBK) to

Bari (LIBD). The portal shows that there is an opportunity with total cost decreased by 835 and a delay decreased by 16 min.

We see in Fig. 4.1 that there is a heading which shows us if a flight has a certain delay and if the flight has a penalising regulation. There is also the rerouting opportunity (OPP) column: The OPP indicates that the flight has been identified as possibly benefiting from a flight efficiency rerouting. If an opportunity exists for the flight, a hyperlink to the flight's operations log (oplog) REROUTE messages is provided, that indicates the costs saving (C) and delay (D) saving if any.



ADEP	ADES	EOBT	EOBT Validity	E/CTOT	DELAY	REGUL+	OPP
EHBK A	LIBD	16-16:25		16:35E			C-836, D-16

Fig. 4.1: Reroute opportunity in NMP

It is also possible for the AOs to see the visual representation of their current route and the proposed route, as shown in Fig. 4.2. Finally, the AO has an option to copy the flight plan which has been proposed to them in order to change it and file a new flight plan, see Fig. 4.3. Another way through which we can view the post-ops REROUTE messages is by using EUROCONTROL's Network Manager Interactive Reporting (NMIR) [34] archive dashboard. It is a web interface that provides access to reports containing archived data from the Network Manager's operational systems. NMIR also offers derived performance and quality indicators, which are generated from the data stored in the NM data warehouse and archive system.

Expert's View

From the expert's perspective of EUROCONTROL staff, it has been noticed that, although AO's system can provide optimised routes for each of the flights, it is a very time consuming operation to look up for better optimized routes. Therefore, the AOs are often relying on internal, historic, city-pair based routes in their catalogues. For each flight, a limited number of flight route computations are performed at a given moment in time by the AOs. This could lead to missing or not discovering better routes. These routes later become available in dynamic network only after filing the flight plan. The AOs do not always update the catalogue with the latest route network changes. For these reasons, the GRRT can provide heads-up on this info. This behaviour is consid-

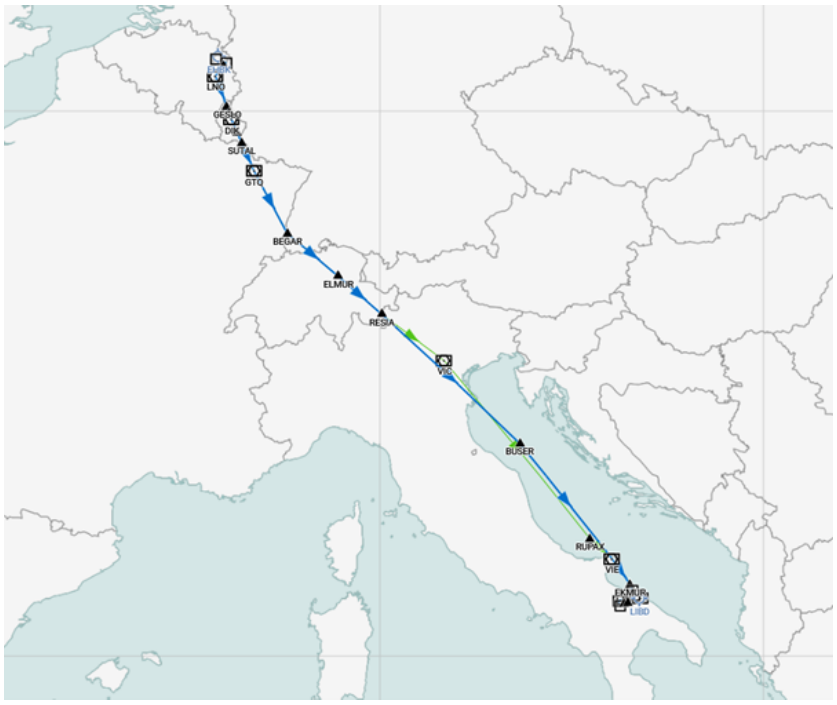
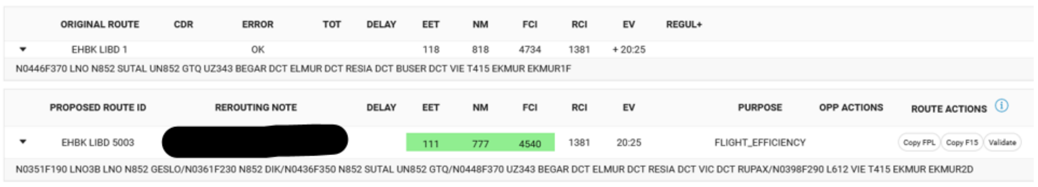


Fig. 4.2: Map of current flight plan (blue) and proposed route (green) in NMP



	ORIGINAL ROUTE	CDR	ERROR	TOT	DELAY	EET	NM	FCI	RCI	EV	REGUL+
▾	EHBK LIBD 1		OK			118	818	4734	1381	+ 20:25	

N0446F370 LNO N852 SUTAL UN852 GTQ UZ343 BEGAR DCT ELMUR DCT RESIA DCT BUSER DCT VIE T415 EKMUR EKMUR1F

	PROPOSED ROUTE ID	REROUTING NOTE	DELAY	EET	NM	FCI	RCI	EV	PURPOSE	OPP ACTIONS	ROUTE ACTIONS
▾	EHBK LIBD 5003	[redacted]		111	777	4540	1381	20:25	FLIGHT_EFFICIENCY		Copy FPL Copy F15 Validate

N0351F190 LNO3B LNO N852 GESLO/N0361F230 N852 DIK/N0436F350 N852 SUTAL UN852 GTQ/N0448F370 UZ343 BEGAR DCT ELMUR DCT RESIA DCT VIC DCT RUPAX/N0398F290 L612 VIE T415 EKMUR EKMUR2D

Fig. 4.3: Detailed view of the proposed route in NMP

ered usual as it simplifies the selection of flight plans. Through the NMP app, the AOs are looking if there is an interesting opportunity (OPP) flag on their flight. The dispatchers from the airlines can store the route, test in their planning system, and, if they like it, they modify the flight plan with this new route. This is why we later conduct both tactical and strategic analyses. The reason is that the proposed routes can be addressed either concurrently or in the future. Thus, it is intriguing for us to observe the extent to which AOs are using the newly suggested route for future flights.

However, the experts emphasize that airlines should prioritize flexibility in route selection due to changing traffic patterns and the availability of more efficient routes. This is where the GRRT becomes valuable as it enables airlines to choose more suitable and optimal routes for their operations.

5 Data - Network and Messages

Accurate network data play critical roles in addressing any aviation-related issues. Regarding the tasks considered in this thesis, we need specific types of aviation data. These include data from the oplog messages exchanged with the AOs for the filling of flight plans, as well as network data. Our main data inputs are the fields 15 of the flight plans, each containing the waypoints on which the aircraft will fly. We concluded, after the literature review in Chapter 2, that we do not need further data on flight trajectories, since we already know the waypoints from the field 15. Since many researches do not have access to oplog messages where the field 15 is stored, they are finding ways to extract the waypoints from trajectories. Also, the waypoints that are filed in the field 15 of the flight plan and the one through which the aircraft flies may not be the same and therefore it is irrelevant to use trajectories. Therefore we use the field 15, as mentioned in Section 3.2.

As the aviation industry evolves, aviation data undergoes frequent updates. This includes modifications to airspace structures, routes, navigation aids, standard instrument departure (SID), standard instrument arrival (STAR), and runway. To ensure operational efficiency and safety, it is crucial for all stakeholders such as pilots, dispatchers, air traffic controllers, air traffic flow managers, flight management systems, and aeronautical charts to access and work with consistent and up-to-date information from a shared database.

An AIRAC cycle is a standardized interval of 28 days used in the aviation industry for the regular publication and updating of aeronautical information. It is part of the global framework established by the ICAO to ensure the timely dissemination of accurate and consistent aeronautical data.

This means that every 28 days, the data in aviation can change, indicating that whenever we work with aviation data, we need to consider these changes every 28 days. This can also lead to numerous mistakes, duplicated data, and data quality issues. Hence, it is of utmost importance to use correct data from reliable sources.

On the other hand, small modifications and updates to the data found in the data warehouse can happen after every NM release. This is because they are internally managed by EUROCONTROL and each year a new version with improvements is presented in the system. It is therefore not linked to the AIRAC calendar.

For this work, different databases were used to extract data and test different queries to see which databases have the smallest amounts of missing data. Access was given to the data warehouse of EUROCONTROL. The queries were tested on the test version of the data warehouse, but the data were extracted from the operational data warehouse.

The only difference between these two, test and operational, are the different numbering of the flight identifiers (IDs). The test version only holds records of the past three months. The different IDs mean that we need to query the flights through different attributes such as last received off-block time (LOBT) or callsign. So, there are no core differences in the data on which the queries were tested and exported.

5.1 Network Data

For the network data we consulted three databases: DWH [7], Carto [8], PRISME [35].

5.1.1 Data Warehouse

The data warehouse (DWH) is the first one that we consulted since it is the main data warehouse of EUROCONTROL. It is a data warehouse system to gather operational information and to permit its analysis for reporting, data mining and statistics or incident investigation. A high-level schema of the data warehouse can be seen in Fig. 5.1.

We use the Fig. 5.1 for the sake of understanding the (archive to Unix) ARU databases in the data warehouse. From the data warehouse we are mostly interested in the databases

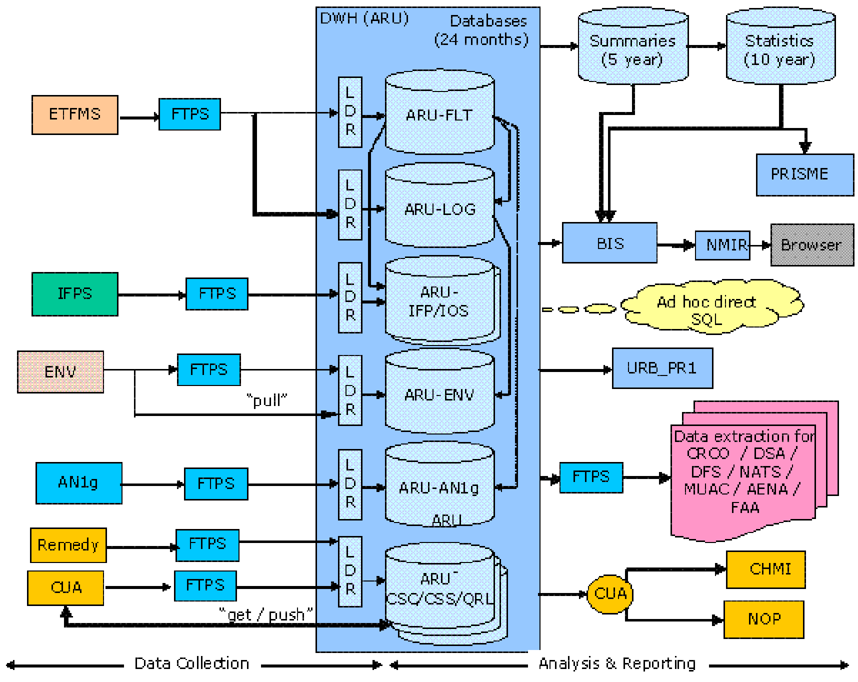


Fig. 5.1: High-level schema of the data warehouse [7]

listed in Tab. 5.1.

Table 5.1: Element of interest in the data warehouse

Database	Usage
ARU_FLT	Flight Data
ARU_LOG	Operational log
ERS_DATA	ENV Reporting Related Data

The ARU databases contain the archived data from the ETFMS systems. Since we need to look for the messages in the oplog and link them with the flights, the ARU_FLT and ARU_LOG databases contain tables that contain this data. The ETFMS is archiving all the flight details into ARU_FLT and puts all the log of all events into ARU_LOG.

The ERS_DATA is a structured relational database containing instances of data from the environment (ENV) system. The database contains data about the network environment: routes, points, sectors etc. The issue with this database is that there is a lot of data cleaning that has to be done as well as many transformation of the data, joining tables etc.

5.1.2 Carto

Carto is the database created and maintained by the cartography office in EUROCONTROL [8]. The difficulty with this database is that for each AIRAC cycle the database is dropped and new data are inserted for the new AIRAC cycle. The business justification of this move is because they simply need the data for creating the maps. Fig. 5.2 is an example of the EUROCONTROL route network chart. The database provides a comprehensive view of all the routes and points needed for our work, along with additional aviation data such as sectors, military zones, and other relevant information. Whenever a historical analysis of some flights is needed, it would be impossible to know what network data are used, unless we keep the exports of the Carto database.

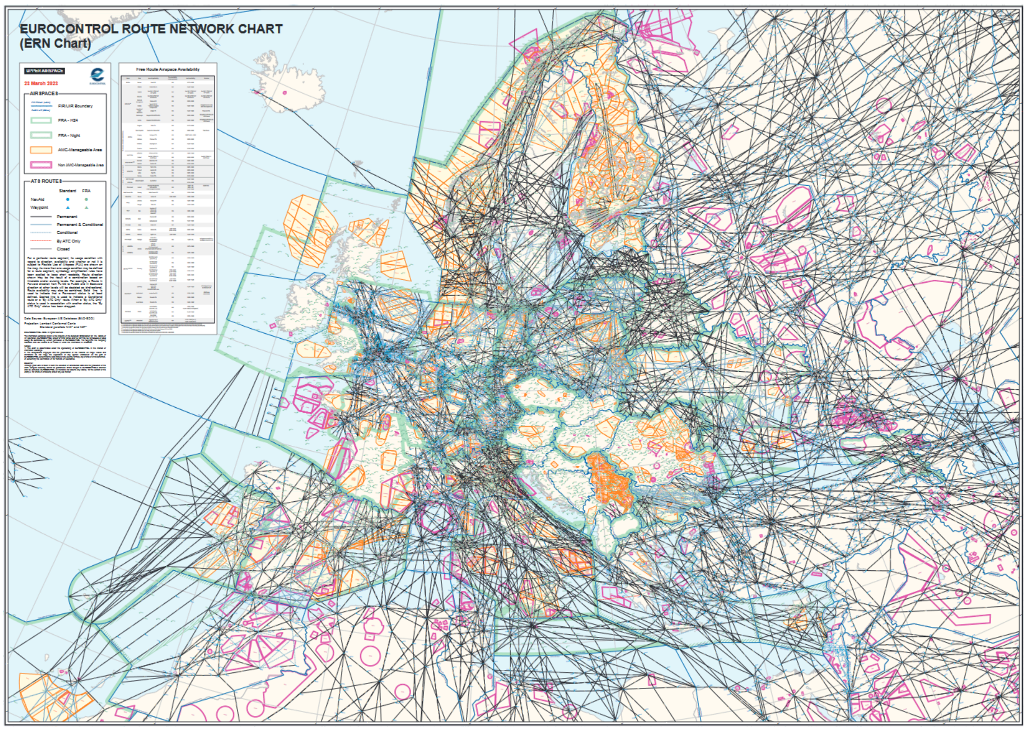


Fig. 5.2: European Route Network (ERN) map [8]

On the other hand, this database contains the most detailed data as they contain the route direction for each segment on all the different flight levels. Since the elements in field 15 are already ordered, there is no need to use a network with such level of detail. The database includes various points with different types, such as navigation aid (NavAid) points, waypoints, and free route airspace (FRA) points. These points have specific properties and play important roles in the aviation system. This is because Carto loads the data from Group EAD, which maintain the European AIS Database (EAD) [36] that serves as the central database and dissemination platform for aeronautical data as well as from ANSPs from countries outside of EUROCONTROL area - NM AREA.

This database remains unused due to several factors, including the AIRAC synchronization mentioned above, incomplete documentation, and the requirement for additional data transformations. Moreover, its primary purpose is to plot data on maps, which does not align with our current objectives.

5.1.3 Gasel - PRISME

The Gasel files are data files which are used as aviation network input to the SAAM software developed by EUROCONTROL. These files have their own suffixes like: `.navaid`, `.ncap`, `.runway` etc. Each of them is generated from the Pan-European Repository of Information Supporting the Management of EATM - PRISME database. This database has been implemented by an external company and is available for internal use. The PRISME database is designed to extract data from the data warehouse and performs data cleaning and transformation processes. This is needed due to the complexity of the data warehouse and a solution to fixing the problem was to create a new database.

The reason why the databases were chosen for environment data is because they contain cleaned data of all the network data that are in the data warehouses. They take the data from the ERS and ENV databases, clean and prepare them for operational use.

The way the data are accessed and extracted is through an internal Command Line Interface (CLI) called Prisme2Gasel. This is the software that is run every AIRAC by the SAAM software. The ease of use is a big selling point for choosing this database since the software uses an internal connection to PRISME in order to get the data and does not need obtaining credentials to access the database. Furthermore, it is using the data from the data warehouse and makes them ready to use.

5.2 Flight Messages

In order to retrieve the flight messages generated by the ETFMS, which tracks changes to the flight plans, we rely on the data warehouse. The data warehouse serves as a repository for all the data captured by the ETFMS. Since the data warehouse is an SQL database, we need to create a query that can be executed for each AIRAC cycle while also being able to adjust different attributes like LOBT, ADEP, ADES, callsign etc. After consulting the internal documentation of the data warehouse [7], the following attributes and tables were used to create our dataset of flight messages:

- `ARU_FLT.FLT` - Information about flights that has to be found in the data warehouse. The flight UniqueId, which is the unique identifier of a flight, is set, in a first time, to `lobd + tactFlightId`;
- `ARU_LOG.FLT_EVT` - Contains the oplog information about eachflight;
- `ARU_LOG.OPL_EVT_MSG` - Contains the oplog ID and the oplog messages.

The selected attributes of the three tables are explained in see Tab. 5.2. In the description of the attributes, Tactical System (TACT) is referenced as ETFMS: in 2002, the TACT system was replaced by the ETFMS [37] and the attribute name was not updated from the documentation.

Table 5.2: Explanation of attributes from the data warehouse

Attribute Name	Table	Description
`FLT_LOBT`	`ARU_FLT.FLT`, `ARU_LOG.FLT_EVT`	Last estimated off-block date and time of the flight as stored by ETFMS.
`FLT_IFPS_PLN_ID`	`ARU_FLT.FLT`	Identifier of a flight plan.
`FLT_TACT_ID`	`ARU_FLT.FLT`	Internal number of a flight within the TACT (ETFMS) system. This number is unique per day.
		Continued on next page

Table 5.2 – continued from previous page

Attribute Name	Table	Description
FLT_UID	ARU_FLT.FLT	Unique identifier of a flight in the ARU system. This identification is FLT_TACT_ID + FLT_LOBT.
FLT_REG_MARKING	ARU_FLT.FLT	Markings of registration of the aircraft, 18th ICAO field (reg).
FLT_ACFT_ID	ARU_FLT.FLT	Aircraft identification; it may be the registration marking of the aircraft.
AO_ICAO_ID	ARU_FLT.FLT	Identifies an aircraft operating agency, which is a person or an organisation or a company engaged or bidding engage an aircraft operation.
FLT_DEP_AD	ARU_FLT.FLT	ICAO identification of the aerodrome of departure (ADEP).
FLT_FTFM_ADES	ARU_FLT.FLT	ICAO identification of the aerodrome of destination (ADES) in the FTFM model.
ICAO_ACFT_TY_ID	ARU_FLT.FLT	ICAO identification of the type of aircraft.
		Continued on next page

Table 5.2 – continued from previous page

Attribute Name	Table	Description
AIRAC_CYCL	ARU_FLT.FLT	Identification of the AIRAC cycle to which must be related to the present regulation data. NOTE: Stores ETFMS identification number which is different from the ICAO number.
FLT_F_RTE	ARU_FLT.FLT	Complete ICAO field 15 information comprising of initial requested speed and flight level and route. Contains corrected route information sent from IFPS to addresses outside of NMOC.
OPL_EVT_ID	ARU_LOG.FLT_EVT, ARU_LOG.OPL_EVT_MSG	Identifier of the oplog event.
FLT_EVT_EVENT_KIND	ARU_LOG.FLT_EVT	Identifier which defines the kind of Operational Log event.
OPL_EVT_SEQ_NO	ARU_LOG.FLT_EVT, ARU_LOG.OPL_EVT_MSG	Tact Generated sequence number of the message.
OPL_EVT_TIME_STAMP	ARU_LOG.FLT_EVT	Date (year, month, and day) and Time (hours, minutes, and seconds) logged by TACT for the event.
OPL_EVT_MSG	ARU_LOG.OPL_EVT_MSG	This entity stores the full oplog message.

5.2.1 Oplog Event Messages

We are interested in extracting 4 types of event messages from the oplog: filed flight plan (FPL), change or modification (CHG), REROUTE and REROUTE MATCHING.

FPL and CHG messages

The flight plans are submitted as a FPL message, but the flight plans may be modified by sending a modification message (CHG). Both messages contain small structural differences, but from each of them we extract the field 15.

REROUTE

The REROUTE messages contains all the routes that were identified as possible candidates. In Fig. 5.3 you can see an example of a REROUTE message.

The first paragraph is the current (original) flight plan that is filed from the AO. The next paragraphs are the reroute proposals. There is not a fixed limit on how many there can be. At the end of the message, we have the reroute purpose and some other fields. Each REROUTE message can be either INTERESTING or UNINTERESTING. These two labels are put on each REROUTE message and they point out if the given route opportunities are more beneficial than the last route submitted by the AO. A reroute proposal is interesting if it proposes a better route than the existing one according to the parameters, then that message is considered INTERESTING. Otherwise, it is labelled as UNINTERESTING. There are other details such as the id of the message `AA4619649400098262920230401060236005420837300000000632` composed by a concatenation of the IFPS ID, TACT ID and some other identifiers. We see the cost values for each route. The messages are ordered by total cost. The message contains the field `Route_Id` which is the opportunity route identification. It is an internal ID of the system which holds information about how that route has been generated. There are three opportunity route identifications by the GRRT system:

- CityPair (mask: `ADEPADES5XXX`). The system keeps a database of usual routes that the AO is flying on that city pair. Example: `EHAMLWSK5001`;
- Pathfinder (mask: `ADEPADESGXXX`). A route that has been generated by the system and does not correspond to a CityPair route that the aircraft is usually flying. Example: `EHAMLWSKG1`;
- Mixer (mask: `ADEPADES9XXXX`), These routes are a mix of the Pathfinder and CityPair routes made by the system based on city-pair statistics. Example: `EHAMLWSK99999`.

```
HIREROUTE                                    CALLSIGN
AA461964940009826292023040106023600542083730000000632Flight
affected
by rerouting 01077017, outcome: EGSHEHAM5002 INTERESTING
Current route:
Field_15: N0329F190 DCT BANEM DCT SONDO M183 SONOG/N0345F210
M183 REDFA REDFA1A
Original Estimates: EET/EHAA0013
 DEPARTURE_DELAY 0035m37s
 FLYING_TIME 0033m19s
 ROUTE_LENGTH  158
 SUSPENSION FALSE
 FUEL 858 kg
 ROUTE_CHARGES  176

Route_Id: EGSHEHAM5002
Field_15: N0329F190 DCT BODSO L17 RAMID/N0345F210
    L17 MOLIX MOLIX2A
 DEPARTURE_DELAY 0033m59s (-0001m38s)
 FLYING_TIME 0031m37s (-0001m42s)
 ROUTE_LENGTH  144 (-14)
 SUSPENSION FALSE
 FUEL 803 kg (-55 kg)
 ROUTE_CHARGES  167 (-9)

Rerouting Note: [[SCENARIO]]
Publish: FALSE
```

Fig. 5.3: REROUTE message

REROUTE MATCHING

The REROUTE MATCHING message is sent every time the GRRT system finds that the flight plan from the AO was proposed by the system for that flight. The message contains the statistical information of the latest and proposed flight plan as well as which `Route_Id` had the proposed route.

5.2.2 Data Warehouse Query

Our main goal is to retrieve the `OPL_EVT_MSG` located in the `ARU_LOG.OPL_EVT_MSG` table. The key ID for this table is the `OPL_EVT_ID`. From here, the only table that contains the `OPL_EVT_ID` as a foreign key is the `ARU_LOG.FLT_EVT`. But since the database only stores information about the oplog for a flight, we are not able to see details such as ADEP and ADES. Therefore, we need to use the primary key `FLT_UID` to connect to `ARU_FLT.FLT` table where all the data about a flight are stored.

This query consists of doing 2 joins with 3 very big tables. It is worth noticing that using this database scheme it is difficult to easily extract the oplog text since we need the knowledge of the oplog ID to get the message. But to enrich the dataset with more information we would need to join already joined tables with new ones. This can cost the DBMS a lot of execution time. On the other hand, even if we need to get only the CHG messages or REROUTE messages, we would still have to apply selectivity on `FLT_EVT` for the `FLT_EVT_EVENT_KIND` attribute in order to join with `OPL_EVT_MSG`. This becomes a very big problem when creating the query. It is also worth mentioning that there are almost no indexes for some attributes that are important for the query, such as `FLT_EVT_EVENT_KIND`. Given in the Appendix B.1, you can find the SQL code to which we are referring to. Looking at the query, we can observe that there are two inner joins of three tables that create one CTE which then is used in two sub-queries that are unioned.

The utilized Common Table Expression (CTE), line 1-44, is referred to as `EVERYTHING` in order to apply various transformations to the `OPL_EVT_MSG`. The purpose of this CTE is to handle different formats of `FLT_EVT_EVENT_KIND` and extract desired fields such as field 15 or statistical information. This approach consolidates all necessary data into a single CTE due to limitations imposed by the DBMS, which restricts the use of views or materialized views.

The `ARU_FLT.FLT` table, line 23-25, provides the necessary information to specify the parameters of AIRAC_CYCL and AO_ICAO_ID. These parameters serve as the primary inputs for the SQL query, as the query requires data for each specific AIRAC cycle and different AO. The values of these parameters can be modified to query specific ADEP and ADES based on the desired criteria. From the ARU_LOG.FLT_EVT table, line 29, we are only selecting the oplog messages of the kind: FPL, CHG, REROUTE, REROUTE MATCHING.

From the analysis of the ARU_LOG.OPL_EVT_MSG, line 28-31, an issue was encountered regarding the size of the message. The OPL_EVT_MSG is encoded in varchar(4000) and almost all reroute messages and a minority of the FPL and CHG messages are passing this limit. Since they can not fit into one, they are divided into other rows having the same OPL_EVT_ID, but ordered by OPL_EVT_MSG_SEQ_NO.

The aim of this sub-query, line 37-42, is to group the rows based on the OPL_EVT_ID column. For each unique OPL_EVT_ID, the query concatenates the values of the OPL_EVT_MSG_TXT column, separated by commas, into a single string. This concatenation is achieved using the XMLAGG and XMLELEMENT functions. The extracted text content is then ordered by OPL_EVT_MSG_SEQ_NO before being trimmed and stored as the OPL_EVT_MSG_TXT. Since the message is 4000 bytes, we can not use LISTAGG function, but instead we need to use XMLAGG and XMLELEMENT functions to create and store the message as a CLOB (Character Large Object). Once we have the CTE ready, the CTE is queried in both queries which are concatenated with a union.

In the first sub-query, line 45-89, all oplog messages, excluding those labeled as REROUTE, were processed. Due to the constraints of not being able to perform any write queries on the database, all oplog message events were combined into a single output for treatment. This leads to having different columns that have null values, see line 66-82, since we do not have that data for the specific oplog event. In this case that would be the statistical values from the REROUTE message which are not found in FPL or CHG, which leads to null rows for those columns. It is important to mention that from the REREOUTE MATCHING event, we get only the matches that are done by the GRRT system.

The second sub-query, line 95-159, is more complex since we must pay attention on the structure of the REROUTE message. An explanation of the structure of the message has been given in Section 5.2.1.

The temp table in the FROM keyword is doing the opposite of what we did in the CTE, line 142-158. Since we have the complete message, undivided into multiple rows, we are putting each paragraph into a separate row while maintaining their original order. This is only done on the `REROUTE INTERESTING` messages. The order is very important so we keep it by applying the `ROW_NUMBER` window function partitioned by the `OPL_EVT_ID` and ordered by the `OPL_EVT_TIME_STAMP`. We use the time stamp since it is possible to have one `OPL_EVT_ID` in two different `FLT_UID` as the number overflows the limits of integer and starts back from 0. Once transformed, the REROUTE messages with `MSG_ORDER` 0 are the latest flight plan from the AO, to which the reroute proposals are made. The reroute proposals are the one with `MSG_ORDER` > 0. All of them continue to share the same oplog `OPL_EVT_ID`. The rest of the columns are simple regex patterns used to get different fields from the oplog message.

5.3 Field 15 Parser

One of the most challenging processes in this work is to accurately parse the field 15 from the flight plan. It is of outmost importance since our algorithm is based on the route that is filed in field 15. A Python script was developed to generate a list of waypoints based on a given input string for field 15. This script processes the input and extracts the relevant information to determine the sequence of points that the aircraft is intended to traverse during its flight.

In order to correctly explain the field 15, specific rules have to be considered [5, 38]. As given in the Appendix C.1, the string contains different words/elements. These elements can either be: point, route, SID/STAR,speed/altitude, cruise climb element. There are also subtypes, such as NavAid points, waypoints and FRA points, as well as other fixed types such as direct routing (DCT), North Atlantic Tracks (NAT), etc. In the field 15, we also have points which are geographical coordinates which are found on the flights over the Atlatinc Ocean. The route part from the field 15 in the flight plan and how to fill it has been thoroughly explained in the Appendix C.1 Let us use the following example for the flight from Geneva (LSGG) to Amsterdam (EHAM):

```
N0445F400 DIPIR6A DIPIR V25 ARBOS UL47 EPL UM624 ROUSY DCT
IDOSA DCT BUB/N0413F240 Y28 HELEN HELEN2A
```

First, the field 15 is split into a list and the predefined regex rules are matched with each word. From there, we get a enumeration of each type. The parser functions in a way that reads the field 15 string and places each word inside a list. The flight level changes are then split from the point, for example: `BUB/N0413F240`. We then run the patterns, given in the Appendix B.2, to label the positions by their type. Since the creation of pattern is very complex and the most error prone, we used pre-existing regex patterns made by aviation experts from a GitHub repository [39]. They were thoroughly tested and reviewed as correct. Given the example, the F15 is enumerated as follows:

Types:

```
POINT, ROUTE, SID_STAR, DCT, SPEED_ALTITUDE,
CRUISE_CLIMB
```

Elements:

```
['N0445F400', 'DIPIR6A', 'DIPIR', 'V25', 'ARBOS', 'UL47',
'EPL','UM624', 'ROUSY', 'DCT', 'IDOSA', 'DCT', 'BUB',
'N0413F240', 'Y28', 'HELEN', 'HELEN2A']
```

After labelling the elements, we remove all elements that are not characteristic of a 2D trajectory, namely initial speed, cruise altitude, change of altitude and speed and SID/STAR.

Because of some flights coming from North America, we need to put some manual constraints on the parser. Due to differences in naming points, routes, SID/STAR, some of the SID/STAR in USA and Canada are matched by the regex pattern as routes instead of SID/STAR. This makes critical errors to the transformations that we do later when the process looks them up as a route. In order to avoid this problem, we check that, after the removal of the types that are not point or route, that the first value must be a point, otherwise the parsed element is removed from the sequence. In the example, we obtain:

```
['DIPIR', 'V25', 'ARBOS', 'UL47', 'EPL', 'UM624', 'ROUSY',
    'IDOSA', 'BUB', 'Y28', 'HELEN']
```

A common edge case in the flight plans that we might think is a mistake, but it is not, is to have the DCT as final element of the Field 15. DCT indicates a direct routing between two waypoints without any specific airway or predefined route. The following example instructs from OTBED to go directly to the airport:

```
N0366F240 BERGI L602 SUPUR L60 SOPEK L989 BODSO Y70 OTBED DCT
```

Once we have only routes and points left, we need to expand the route elements with the points that are inside them. The basic rule is that whenever we have a route, the adjacent elements next to the route should be points. When reading `PointA Route1 PointB` in a field 15, it means that, from `PointA`, the aircraft will fly on Route1 until it reaches `PointB`. While the aircraft flies on Route1, it passes through other points. Since those points are "hidden" under the Route1 element, and because the field 15 is a minimal "compressed" version of the route, we need to extract the points from the route. Of big importance for this operation is the order of the segments. It could happen that `PointA` is only in one direction, to `PointB` for example. But since it is explicitly written that the segments on the Route1 are in direction from `PointA` to `PointB`, then we assume that the inner segments are in that direction as well. This is also one of the reason why the direction attribute is not needed from the Carto database, mentioned in Section 5.1.2. The code for this operation is given in the Appendix B.3. The routes are taken from the `.routes` gasel file and are processed with a Python script that creates a `.json` file where each key is the name of the route and the points on that route are ordered in a list.

Following the example, we see that the route `UM624` is in the list. The adjacent points are `EPL` and `ROUSY`. By the rules, these two points are on the route. In the route file that we generated we have the following for `UM624`:

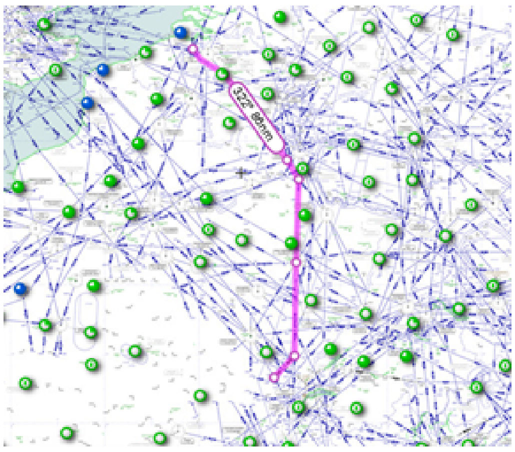
```
"UM624": ["ROUSY", "JARNY", "NANCY", "EPL", "LUL", "TORPA"]
```

Both `ROUSY` and `EPL` are there, but we also notice that they are in a different order from the one in the field 15. The algorithm will extract `JARNY` and `NANCY` from the route. Since they are in reverse order, we reverse the points to `NANCY`, `JARNY`. Then, we replace `UM624` with `NANCY`, `JARNY` and get the following result: `EPL`, `NANCY`, `JARNY`, `ROUSY`. We do this for every Route in the filtered list. The final result is an ordered list of points showing the planned route from the field 15 on the lowest granularity:

```
DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY
IDOSA BUB JAZFI HELEN
```

We can compare the result with the initial flight plan if we plot the input flight plan that was used as input and the final output of the parser. In Fig. 5.6, on the left (Fig. 5.6a) we have the initial field 15, and on the right (Fig. 5.6b we have the output of the parser. We can observe how the parsed route has the lowest point granularity. The points are represented as white dots on the magenta line:

Given one flight, we can have minor differences in the Filed 15 for different kinds of messages. In the FPL and CHG messages it can happen that the field 15 may not


(a) Flight route before parsing

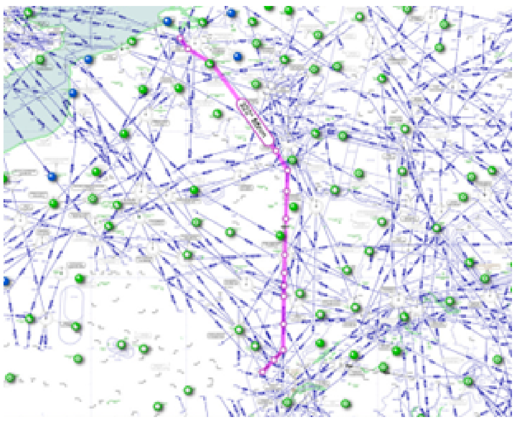

(b) Flight route after parsing

Fig. 5.4: Comparison of waypoint granularity before and after parsing the field 15

contain the connecting routes. But in the REROUTE message, there will always be the SID and STAR since it is a description of the whole trajectory. Still, both messages are showing the same route and, by using the parser, we will get the same parsed route. For example, for the same flight from Geneva (LGSS) to Amsterdam (EHAM), We have the CHG message:

```
N0444F400 DIPIR V25 ARBOS UL47 IXILU UN853 IBERA DCT RITAXM624
GILOM/N0449F380 M624 BUB/N0424F280 Y28 HELEN
```

And the REROUTE message:

```
N0444F400 DIPIR6A DIPIR V25 ARBOS UL47 IXILU UN853 IBERA DCT
RITAX M624 GILOM/N0449F380 M624 BUB/N0424F280 Y28 HELEN HELEN2A
```

Which give exactly the same output after applying the parser where only `DIPIR6A` and `HELEN2A` were missing in the raw field 15 for the CHG message:

```
DIPIR LERDU ARBOS PENDU IXILU GIVOR SORAL IBERA RITAX REMBA
GILOM BUB JAZFI HELEN
```

The Parent Point Problem

A common problem in aviation data are the naming of points. Since there is a limited number of characters to name points, it can happen that we have duplication of those points. ICAO has created ICARD (International Codes and Route Designators) which is the database of more than 280,000 five-letter-name-codes (5LNC) and more than 16,000 route designators (RD). It allows the member states of ICAO to reserve the naming of waypoints and designators for ATS route. The rules summarized by UK's civil aviation authority [9] are graphically represented in Fig. 5.5.

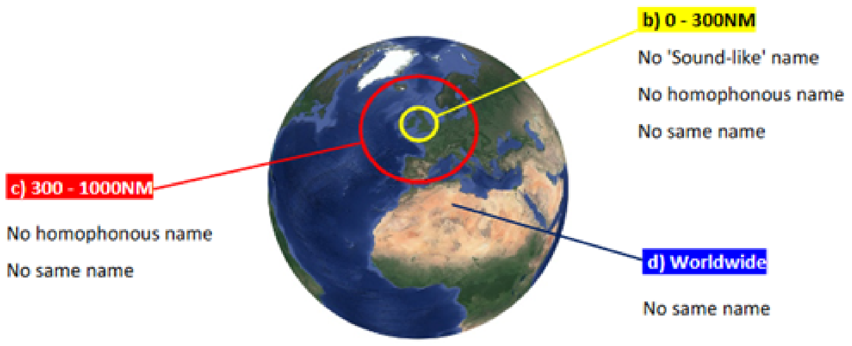


Fig. 5.5: Parent point problem [9]

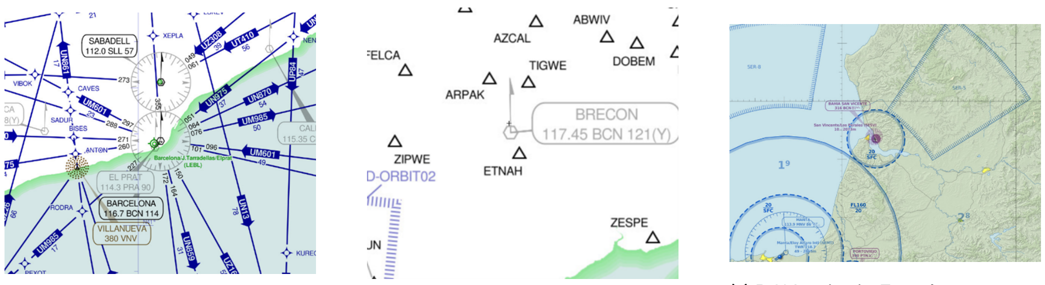
- There are no "sound-like" 5LNCs to a distance of 300nm;
- There are no homophonous 5LNCs to a distance of 1000nm;
- There is no exact match of the 5LNCs worldwide (using external tools).

Each system needs to find its own way of dealing with this issue. In the Gasel files that we use for our data, there is a file with the extension `.navpointparent` which contains the encoding of the points that have the same name. As an example, we observe the BCN point in Tab. 5.3.

Source	Destination
BCN	*BCN
BCN	*BCN6
BCN	BCN

Table 5.3: Parent point data for BCN waypoint

From Tab. 5.3, BCN is a point in Barcelona, Spain, the other one in Brecon UK, last one is a VFR point in Bahia de Caraquez, Ecuador. We can see the three points



(a) BCN point in Spain

(b) BCN point in UK

(c) BCN point in Ecuador

Fig. 5.6: Comparison of the parent point problem with the BCN waypoint

visually in Fig 5.6. Since in the flight plan the system is assigning them with their alternative names, our matching algorithm is not performing well and gives critical errors since the algorithm can not find a certain point in a route. In order to avoid this problem, each point name that contains the asterisk $*$ as the first character is searched in the `.navpointparent` file and is given the original name of the point. This is done simply because when we need to generate the hash for the route ID, we need to have the original name. Therefore, there is a separate parsed field 15 column in the dataset containing the alternative point name and another column with the original name called `F15_FOR_MATCHING`.

5.4 Transformations and Pipeline

Due to certain limitations in SQL, specific transformations were implemented using Python. However, we made efforts to maximize the utilization of SQL for extraction and transformation processes, as it generally offers better performance compared to Python for most operations. An ETL pipeline has been created to automate the process of obtaining environment and flight data. The function that runs the ETL takes the AIRAC number and the name of the AO as inputs. The goal is to run this pipeline for each AIRAC to update the data catalogues and generate the necessary data for post-ops analysis. The description of columns and example of a row from the dataset can be found in Appendix A.1.

The ETL pipeline is structured as follows:

- Extract
- Transform

- Stage

5.4.1 Extract

At this stage, data are extracted for the flights and network data. In particular:

- A connection is established with the data warehouse to execute an SQL query that retrieves all messages for a specific AIRAC and AO;
- The Prisme2Gasel executable is launched with the appropriate AIRAC and file types (e.g., points, routes, point-parent);
- The data are then temporarily stored in the same repository, ready to be used by the transformation function.

Each of these functions is encapsulated within a function that handles the extraction part of the ETL.

5.4.2 Transform

The transform part of the ETL immediately follows the extraction part. It comprises functions that use the extracted data as input, and it places the transformation outcomes in another folder. In particular:

- For points, we use the `.nnpt` file from Gasel in CSV format. The relevant columns are: `PT_ID`, `PT_LAT`, and `PT_LONG`. As the coordinates are in degrees, minutes, seconds (DMS) format, we need to convert them to decimal degrees (DD) format. For example, `432653N` in DMS is converted to `43.448056` in DD;
- For routes, we use the `.routes` file from Gasel in CSV format. Each row in the file contains the name of the route, the successive points, and the order of the segment that these two points create, along with a few other columns that we do not need, for example:

```
L;A145;AR;999999999999;000000000000;LORNO;SP;1
L;A145;AR;999999999999;000000000000;RESPA;SP;2
L;A145;AR;999999999999;000000000000;BERAP;SP;3
L;A145;AR;999999999999;000000000000;ANEPI;SP;4
L;A145;AR;999999999999;000000000000;TRL;SP;5
```

Our transformation creates a JSON file where the route name serves as the key, and the ordered list of points on that route is the value. The output for the example above is:

```
"A145":["LORNO", "RESPA", "BERAP", "ANEPI", "TRL"]
```

For flights, we use the CSV output from the SQL script with 37 columns. The SQL script can be found in the Appendix B.1. We also utilize the `.navpointparent` file to handle the parent point problem. A series of transformations are applied:

- First, some columns are renamed for simplicity;
- The dataset is sorted in ascending order by columns `FLT_LOBT`, `FLT_UID`, `OPL_EVT_TIME_STAMP`, and `MSG_ORDER`. This order is crucial for subsequent transformations and group by operations. It first orders the flights by `FLT_LOBT`, then by `FLT_UID`, and finally by the order of the oplog messages and the order of the message since we want the first message to be the one with order 0;
- The field 15 parser is applied to `FIELD_15` (detailed in Chapter 5.3);
- To avoid the parent point problem, duplicated points are substituted with their parent points (explained in Chapter 5.3);
- The routes are categorized using the `SYS_ROUTE_ID` provided by the system (explained in Chapter 5.2.1). Different regular expressions are used to identify the route identifier; The `ROUTE_ID` is then generated using a hash function applied to `F15_FOR_MATCHING`. From the statistics fields in reroute messages, we find the delta between the reroute proposal and the flight plan for which the proposals are referred. The new columns have the old column name with the prefix `DELTA_`.

5.4.3 The Stage Phase

As the data are not loaded into any system, this step may not be directly applicable. However, it is essential to store the transformed data somewhere. Therefore, we designate the final part as the "stage" phase, where the data are prepared for further usage. When the data reach this stage, they have been extracted from all sources, transformed, cleaned, and are now ready for further operations.

5.4.4 File Hierarchy

The file system for the ETL process is structured to handle the various stages of data processing efficiently. In Fig. 5.7 is an explanation of how the file system is typically organized for this ETL.

```
ETL
├───extract
│   │   flights.py
│   │   gasel_network.py
│   ├───data
│   │   ├───flight_messages
│   │   │       AO_ARIAC.csv
│   │   └───network
│   │           AIRAC.navpointparent
│   │           AIRAC.nnpt
│   │           AIRAC.routes
├───pipeline
│       main.py
├───staged
│   ├───flights
│   │       AO_ARIAC.csv
│   ├───points
│   │       AIRAC.csv
│   └───routes
│           AIRAC.json
└───transform
    │   flights.py
    │   points.py
    │   routes.py
```

Fig. 5.7: ETL file system

6 Matching Methodology

In this chapter, after an explanation of the GRRT algorithm, we describe the matching methodology that we propose, and the preparation for the strategic and tactical analysis. Our aim is to analyze the oplog messages and create an algorithm that will compare the field 15 from the oplog messages (FPL, CHG, REROUTE) using a 2D approach to detect flight route matches. For this matching problem, a method is considered that focuses on the actual 2D trajectory composed by waypoints that are filed in the flight plan. Our methodology has the possibility to convert each route from the field 15 into a unique ID that can then be used for post-ops analysis. We will be able to analyze which route proposals were used after a certain period from the first time after the GRRT introduced them to the AOs. We recall that, as explained in Chapter 5.3, having the correct network data is crucial for the problem of matching the points.

6.1 Group Reroute Tool Approach

The current ETFMS has a functionality to analyze each update of the flight plan message (FPL, CHG) that was sent after a route opportunity message (REROUTE) was proposed to the AO. An explanation of the messages can be found in Section 5.2.1. Once the route proposals were generated, the REROUTE message is sent. The AO can either store them in its local system for further use or change the flight plan by filing the field 15 with the route that GRRT proposed. The GRRT tool triggers a REROUTE

MATCHING message when its algorithm has detected that the new flight plan modification message (FPL/CHG) has a route that is an exact match to one of the reroute proposal (REROUTE) messages that was sent by the GRRT itself.

6.1.1 Creation of Reroute Proposals

We explain the process of the creation of the flight reroute proposals. We take the example of a flight from Frankfurt (EDDF) to Madrid (LEMD). The points on which the aircraft will fly for the given route in field 15 are shown in Fig. 6.1.

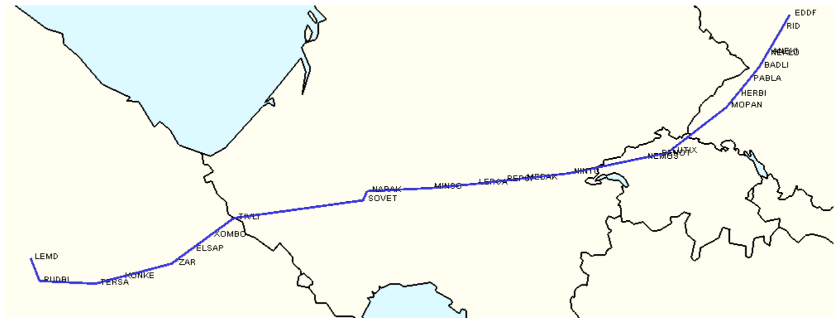


Fig. 6.1: 2D route of flight from Frankfurt (EDDF) to Madrid (LEMD)

The system then calculates the profile of the flight, shown in Fig. 6.2. The reroute proposals are now generated. They are generated as a consequence of filing the route shown in Fig 6.2. In Fig. 6.3, we see the blue lines that are the reroute proposals. We can see that some of them are going through different points and have different flight levels. With the red line, we see the initial flight route that was filed in the flight plan. The green line is the line on which the aircraft actually flew. We see that the aircraft flew higher then indicated in the flight plan. This probably happened because the ATC instructed to change the flight level. We can also see that green deviates on some points from the red line but still crosses through most waypoints.

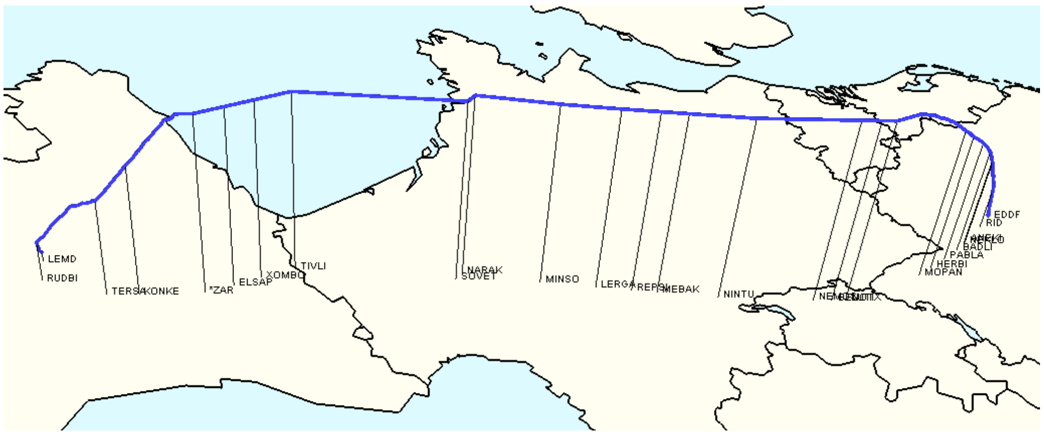


Fig. 6.2: 3D route (profile) of flight from Frankfurt (EDDF) to Madrid (LEMD)

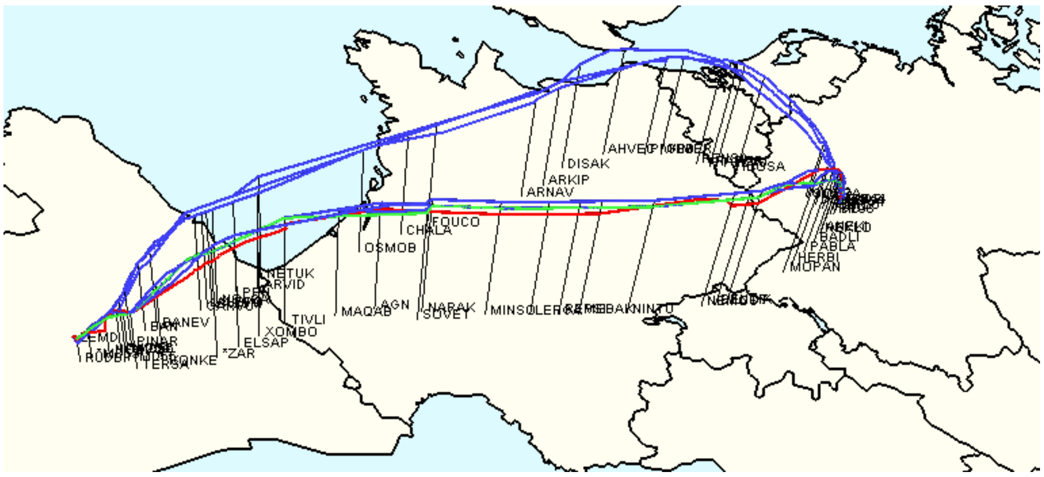


Fig. 6.3: Proposed (blue), initial (red), flown (green) route from Frankfurt (EDDF) to Madrid (LEMD)

6.1.2 Group Rerouting Tool's Matching Algorithm

The matching algorithm uses the 4D trajectory of the flight. It compares the flights in FPL and CHG with the REROUTE messages. An approximate match means that two 4D trajectories of flights have the same trajectory within a certain deviation. An approximate match is using the following rules written in the documentation of the ETFMS [11]:

- Only the aerodromes and "en-route" points of the point profile are used in the matching. Points on the departure or arrival procedure are ignored;

- There can be more points in the incoming flight than in the proposal flight. The additional points found in the incoming flight are ignored;
- The sequence of the en-route points (when ignoring the incoming flight additional points) must be identical;
- The timing on the points must be similar (absolute difference must be less than `POINT_PROFILE`: `TIME_TOLERANCE`);
- The levels on the points must be similar (absolute difference must be less than `POINT_PROFILE`: `FL_TOLERANCE`);
- If the points to be compared are geographical position, the distance between the points must be less than `POINT_PROFILE`: `GEO_TOLERANCE`.

Current value of parameters are:

- The value of `FL_TOLERANCE` is 10 FL;
- The value of `TIME_TOLERANCE` is 5 minutes;
- The value of `GEO_TOLERANCE` is 1.0 nautical miles.

Example of REROUTE MATCHING by Group Rerouting Tool

On the flight between Amsterdam (EHAM) and Nice (LFMN) there has been a REROUTE MATCHING done by the GRRT system. The initial flight plan submitted by the AO was:

```
N0454F370 WOODY N872 MEDIL UN872 KOVIN UM728 KETEX DCT KOTIS DCT
KUKOR DCT UTUVA DCT LERGA/N0436F370 DCT LIQID UY30
LATAM UY22 NISAR NISAR7R
```

Then, a CHG message proposal was sent by the system with the flight plan:

```
N0454F370 EDUPO3X EDUPO Z739 MISGO DCT ASBON DCT TUSUK DCT KRH
DCT NATOR DCT TRA DCT RIPUS DCT SOSON DCT ODINA/N0436F370 DCT
ENOBA Z185 BORDI BORDI7R
```

A REROUTE change message is being received that suggests an alternative to the current route:

```
N0456F370 EDUPO Z739 MISGO DCT ASBON DCT TUSUK DCT KRH DCT
NATOR DCT TRA DCT RIPUS/N0448F390 DCT SOSON DCT ODINA DCT
ENOBA/N0422F290 Z185 BORDI BORDI7R
```

The system creates the 4D trajectories from the flight plans and finds this to be a match of two CHG and REROUTE messages. And therefore it labels the flight as a match.

We can see that both of them will start on `FL370`, pass through the same en-route points, but the difference is the flight level change in the CHG message on point `RIPUS` to `FL390` and `ENOBA` to `FL290`, while in the REROUTE message, there is a flight level change on `ODINA`. By the rules of this algorithm, there is a match since the parameters are in the range of the tolerance for flight level, time and geographic position.

Drawback

The 4D algorithm is generic as it is created to support other mechanisms of the ETFMS, such as slot reservation after reroute proposal creation. However, for purposes of the flight efficiency, this algorithm is conservative. The 4D algorithm is not suitable for GRRT and to support post-ops analysis because, after receiving a proposal from GRRT, the AO might refile the flight plan with different time, speed, levels, while keeping newly communicated routing option. It can prevent two trajectories from matching since the algorithm uses geographic and flight level tolerances. While this allows for some flexibility, the algorithm may not capture important variation. We could have a case when a REROUTE message will propose a new route that has to fly on `FL320`. But, because of pre-tactical changes, there is a restriction on flying from `FL320` to `FL380`. And therefore, the AO can refile the flight plan with the same route but with a `FL380` and the algorithm will not see that this as a match, even though the aircraft is flying on the same waypoints.The AOs also adjust the flight levels in the flight plan depending on the daily operations, but they do not always fly on the same flight level during the flight. If difference from proposed route is significant, 4D will not match. Also, 4D is not used to continuously monitor visualization of the routes. It is applied only at the tactical stage.

6.2 2D Matching Algorithm

From the literature review in Chapter 2, we concluded that utilizing a two-dimensional algorithm for flight plans is preferable because these plans rely on predefined geographic waypoints that the aircraft must traverse to reach their intended destinations. Moreover, once we have the list of waypoints, we can easily identify those routes and use them for the strategic and tactical analysis of the proposals.

In order to prepare the dataset, given in the Appendix. A.1, for the strategic and tactical analysis, we need to generate the dataset with all the flight messages and other details needed for the analysis. The data that we use are thoroughly explained in Section 5.1, where we show where we get the data from and what kind of transformations we use in order to create the final dataset. Once the dataset is created, we parse the field 15 messages and create the parsed version of the field 15 where we only hold the points through which the flight is planned to pass during the flight. The parsing is explained in detail in Section 5.3. Upon acquiring the parsed route, for the ease of working and storing the whole message, we create the hash of that message. By creating the hash, we are sure that each unique message will have its own unique hash value. This is a very simple and effective way to handle and store unique strings, in this case unique routes. For creating a hash, we used the standard library hashlib [40] which calculates the MD5 hash value of the route string by encoding the hash using UTF-8, and then converts the hash value into a hexadecimal string. The MD5 hash will be consistent and unchanged. Through the process of generating a hash and designating it as the route ID, we gain the capability to utilize this unique identifier for subsequent analysis.

6.2.1 Tactical Analysis

We describe an example of tactical analysis. We consider the flight from Geneva (LSGG) to Amsterdam (EHAM) which is labelled as a match by our algorithm. We recall that the proposals in REROUTE messages are ordered from the one with the biggest cost benefit to the least cost effective. For the purpose of simplicity, we only select the parsed field 15, route ID and oplog message type. Only a smaller sample of messages were included from that flight for this example, as shown in Tab. 6.1.

Table 6.1: Oplog messages of flight from Geneva (LSGG) to Amsterdam (EHAM)

Message	Parsed Field 15	Route ID	Message Type
1	DIPIR LERDU ARBOS PENDU IXILU GIVOR SORAL IBERA RITAX REMBA GILOM BUB JAZFI HELEN	5333a492856294f0dd67dbc60fa	FPL
2	DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY IDOSA BUB JAZFI HELEN	5cc6ca694c47c90e20c82a54554	REROUTE
3	DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY BUB JAZFI HELEN	00770bc03a75e7c8d3564a05603	REROUTE
4	MOLUS SOSAL TELNO KORED BERSU DITON LOKTA TEDGO HAREM LOHRE MAPOX BIGGE HMM AMSAN NORKU	5a6502ade06657d515e714e1adc	CHG

Continued on next page

Table 6.1 – Continued from previous page

Message	Parsed Field 15	Route ID	Message Type
5	DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY KOMOB REMBA GILOM BUB JAZFI HELEN	857c2085bf045c767cd7c69b89b	REROUTE
6	DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY REMBA GILOM BUB JAZFI HELEN	8b0c34990ba2d96421e8b41b2c3	REROUTE
7	DIPIR LERDU ARBOS PENDU IXILU DIBEX DANAR EPL NANCY JARNY ROUSY IDOSA BUB JAZFI HELEN	5cc6ca694c47c90e20c82a54554	CHG

The AO can have a clear view of all the messages and proposals by the system for that flight. They can view all these proposed routes for that flight. If we imagine for a second that we are the AO, we would like to be able to see which of all these messages would be the most efficient for the flight.

We can establish a rule that considers a flight as "matched" if any of the REROUTE messages sent prior to the CHG message contains the same route ID as the flight.

```
FPL         checks
REROUTE     ||
RERUTE      ||
CHG_________|
REROUTE      |
REROUTE      |
CHG__________|
```

Fig. 6.4: Matching algorithm

As shown in Fig. 6.4, the algorithm will check each REROUTE message before the CHG message and do an equality on the route ID. Messages 2 and 3 of type REROUTE are compared with Message 4 of type CHG and Messages 2,3,5 and 6 of type REROUTE are compared with Message 7 of type CHG. We see that the AO decided to fly with the route under the ID `5cc6ca694c47c90e20c82a545542dd6`, which was introduced by the second REROUTE message. Suppose we reverse the order of the CHG messages and Message 4 becomes 7 and vice versa. We will still get a match since Message 4 is compared with 2 and 3 and it equals Message 2. This is still a match even though it is not the last CHG message and they will not fly by it. It could happen that a military zone opened 2 hours before the flight in the area where they AO was supposed to fly, but the AO has still considered the proposal and filled a change in their flight plan with GRRT's proposed route. These steps are applied on each flight in the dataset. Once running the algorithm through the whole dataset, we are able to see how many matches our 2D algorithm detected. On the other hand, we can compare the number of REROUTE MATCHING messages which indicate how many matches were made by the GRRT.

6.2.2 Strategic Analysis

In order to perform the strategic utilization of the flight reroute proposals from the GRRT, we need to use the 2D matching algorithm. This section provides an explanation of the methodology to construct catalogues for routes and segments. These catalogues are subsequently utilized in the analysis of the uptake of these proposed segments and routes by the AOs. The purpose of the catalogue is to have the first occurrence of the routes and segments from the system side (GRRT) and the AO side. This way, we can easily compare the usage.

Route Catalogue

To perform the analysis of the uptake of the reroute proposals, we need to use the hash values of the parsed routes, the route ID. Since the main goal is to find which routes from the GRRT were filed by the AO, we use the following rule for the uptake of routes: The route is considered as used only when the route is encountered in a REROUTE message prior to its appearance in a FPL or CHG message. In order to implement this logic, a structure is set for fast analysis of all the IDs. We do a sequential scan of the dataset which is of complexity $\mathcal{O}(N)$ and we create a dictionary of the following structure:

```
{
    'ADEP_ADES': {
        'REROUTE': {
            'ROUTE_ID': ['TIMESTAMP', 'FLT_UID']
        },
        'FPL_CHG': {
            'ROUTE_ID': ['TIMESTAMP', 'FLT_UID']
        }
    }
}
```

We remark that the use `ADEP_ADES` to refer to a city-pair, in this case arrival and departure airport. We do not use the callsign for the route since one city-pair can have multiple callsigns.

The dataset is examined row by row to determine if the `ROUTE_ID` in each row represents the most recent time occurrence for that specific message type and its corresponding `ADEP_ADES` values. This process is very fast since we are using a dictionary and it takes $\mathcal{O}(1)$ to check if the `ROUTE_ID` is still there for a specific `ADEP_ADES`. In case there is one already in the dictionary, we compare the timestamp, which is the `OPL_EVT_EVENT_TIME`, when the message was issued by the system. For a full AIRAC and one AO the algorithm takes less than 1 second to create this structure. Once we have this catalogue of first occurrences of routes per `ADEP_ADES` and message type, we can get the flights that were adopted by a simple pass through the data catalogue and comparing if the REROUTE message occurred before the FPL/CHG message.

Segment Catalogue

To perform the analysis of the uptake of the proposed segments by the AOs, we need to find the new segments that were introduced per each REROUTE message. Similar to what we discussed in Section 4.2, the AOs might not use the whole route but only segments are added to their data catalogue. They then test the routes. The motivation for doing this is because even a small segment that is introduced by GRRT can give huge improvements to the efficiency of the routes. Since we need to find the segments per REROUTE message, an algorithm needs to be created for finding the new segments. In each REROUTE message, as explained in Chapter 5.2.1, the first route is the last filled route by the AO for that flight, and the rest of the routes are the proposed ones. Using, as an example, the flight from Rovaniemi (EFRO) to Amsterdam (EHAM), we handpicked one CHG message and the proposals of one message referring to the CHG, as from Tab. 6.2.

Table 6.2: Snippet of oplog messages from Rovaniemi (EFRO) to Amsterdam (EHAM)

Message	**Parsed Field 15**	**Route ID**	**Message Type**
1	`NEPIX RUBSI AXUTI BODRI GURLI TUSKA LEGPI CORIS NIRDU JUIST TEMLU EEL`		CHG
2	`NEPIX RUBSI EKMIK ROGED NIROD KUGAL EEL`	`('RUBSI', 'EKMIK'), ('EKMIK', 'ROGED'), ('ROGED', 'NIROD'), ('NIROD', 'KUGAL'), ('KUGAL', 'EEL')`	REROUTE
3	`NEPIX RUBSI SUTEV LUTIR EEL`	`('RUBSI', 'SUTEV'), ('SUTEV', 'LUTIR'), ('LUTIR', 'EEL')`	REROUTE

Continued on next page

Table 6.2 – Continued from previous page

Message	Parsed Field 15	Introduced Segments	Message Type
4	NEPIX LENSO ZIPCO OGIRO LUTIR EEL	('NEPIX', 'LENSO'), ('LENSO', 'ZIPCO'), ('ZIPCO', 'OGIRO'), ('OGIRO', 'LUTIR'), ('LUTIR', 'EEL')	REROUTE
5	NEPIX RUBSI SUTEV VALAK ATTUS DHE EEL	('RUBSI', 'SUTEV'), ('SUTEV', 'VALAK'), ('VALAK', 'ATTUS'), ('ATTUS', 'DHE'), ('DHE', 'EEL')	REROUTE

We take Message 1 and 2, where 1 is the target string and 2 is the source string. So, we compare strings 1 and 2. We use the algorithm proposed in [28] that looks for the differences between the two strings. From it, we get the following:

```
['  NEPIX-RUBSI', '+ RUBSI-EKMIK', '+EKMIK-ROGED',
'+ ROGED-NIROD', '+ NIROD-KUGAL', '+ KUGAL-EEL', '- RUBSI-AXUTI',
'- AXUTI-BODRI', '- BODRI-GURLI', '- GURLI-TUSKA', '- TUSKA-LEGPI',
'- LEGPI-CORIS', '- CORIS-NIRDU', '- NIRDU-JUIST', '- JUIST-TEMLU',
'- TEMLU-EEL']
```

With + we mark the unique additions from the target string, with – segments that are removed from the source string and blank for those who appear in both strings. Out of this string, we filter the segments that start with +. We therefore get the segments:

```
[('RUBSI', 'EKMIK'), ('EKMIK', 'ROGED'), ('ROGED', 'NIROD'),
('NIROD', 'KUGAL'), ('KUGAL', 'EEL')]
```

From here on, we use a logic similar to the one of the dictionary used by the reroute matching. We structure the dictionary as follows:

```
{
        ADEP_ADES: {
                'REROUTE' {
                        'SEGMENT': ['TIMESTAMP', 'FLT_UID', 'ROUTE_ID']
                },
                'FPL_CHG': {
                        'SEGMENT': ['TIMESTAMP', 'FLT_UID', 'ROUTE_ID']
                }
        }
}
```

Once we have generated the segments, we go through each row and, segment by segment, create the catalogue of first occurrence of those segments per `ADEP_ADES` and message type. From there, the same logic is applied as with the routes: we pass through the data catalogue and compare if the segments from REROUTE message were introduced before the FPL/CHG message.

7 Results and Findings

For this work, we tested the outcomes using AIRAC 501, 502, and 503 cycles from two distinct AOs, which will be anonymized. We chose these three AIRACs since they were actual during the time this thesis was written. AO1 was chosen randomly, while AO2 has been handpicked because the aviation experts from AO2 are aware of GRRT and occasionally use it for their operations. Therefore, we can anticipate more matches from AO2.

7.1 Matching Algorithm

In Tab. 7.1, we see the comparison of the results of our implementation of the route matching algorithm and the algorithm by GRRT.

We describe each column (transposed) from Tab. 7.1:

- AO: The aircraft operator;
- AIRAC: The AIRAC cycle that is used;
- Total flights: The total number of flights for the corresponding AO and AIRAC cycle;
- Total oplog messages: The total number of operational log messages, explained in Section 5.2.1, recorded during the given AIRAC cycle and for the respective AO;

- Total reroute proposals (flight plans): The total count of reroute proposals from REROUTE messages within the specified AO and AIRAC cycle;
- Unique flights with reroute proposals: The count of unique flights that have received reroute proposals for the given AO and AIRAC cycle;
- Flights matched by GRRT: The number of flights matched by the GRRT for the specific AO and AIRAC cycle;
- Flights matched by our algorithm: The number of flights matched by a our 2D algorithm for the particular AO and AIRAC cycle;
- Common matches: The count of flights that are matched by both the GRRT and our 2D algorithm for the given AO and AIRAC cycle;
- Flights matched exclusively by GRRT: The number of flights exclusively matched by the GRRT;
- Flights matched exclusively by our algorithm: The number of flights exclusively matched by our 2D algorithm.

Table 7.1: Comparison of flight matching results for different AIRACs and AOs

AO	AO1	AO1	AO1	AO2	AO2	AO2
AIRAC	501	502	503	501	502	503
Total flights	16215	20110	14865	16013	17862	18586
Total oplog messages	79566	87420	95310	53370	62501	70167
Total reroute proposals (flight plans)	48452	46260	65275	21377	26705	34154
Unique flights with reroute proposals	1636	2156	1360	1061	1550	1347
Flights matched by GRRT	28	52	25	238	324	239
Flights matched by our algorithm	30	51	30	239	325	281
Common matches	28	49	23	232	321	236
Flights matched exclusively by GRRT	0	3	2	6	3	3
Flights matched exclusively by our algorithm	2	2	7	7	4	45

From Tab. 7.1, we can derive additional metrics shown in Tab. 7.2:

- AO: The aircraft operator;
- AIRAC: The AIRAC cycle that is used;
- Match coverage: The column shows the effectiveness of our 2D matching algorithm in covering the proposals generated by the GRRT. It is obtained by dividing the number of “common matches” by the total count of “flight plans by GRRT”.

- Match difference: The percentage difference in the number of matched flights between the our algorithm and the matches from the GRRT. It is obtained by dividing "flights matched by our algorithm" and "flights matched by GRRT";
- Match-to-message ratio (GRRT): The ratio of the number of flights matched by the GRRT to the total number of oplog messages recorded for the specific AO and AIRAC cycle. It is obtained by dividing "flights matched by GRRT" and "unique flights with reroute proposals";
- Match-to-message ratio (our algorithm): Similar to the previous column, it represents the ratio of the number of flights matched by the custom algorithm, "our algorithm", to the total number of oplog messages recorded for the specific AO and AIRAC cycle. It is obtained by dividing "flights matched by our algorithm" and "unique flights with reroute proposals".

Table 7.2: Comparison of match coverage, match difference, and match-to-message ratio for different AOs and AIRACs

AO	AIRAC	Match coverage	Match difference	Match-to-message ratio (GRRT)	Match-to-message ratio (our algorithm)
AO1	501	100%	7%	2%	2%
	502	94%	-2%	2%	2%
	503	92%	20%	2%	2%
AO2	501	97%	0%	22%	23%
	502	99%	0%	21%	21%
	503	99%	18%	18%	21%

From Tab. 7.1, both AOs have a similar number of flights, but AO1 is receiving almost double the amount fo REROUTE messages and has a slightly higher number of "unigue flights with reroute proposal". As mentioned in the introduction, the AO1 has a "match-to-message ratio" of around 2% while the AO2 has around 20% which confirms the hypothesis that AO2 is using the GRRT tool. We can also notice that, for AO2 in AIRAC 503, we have a huge amount of exclusive matches by our algorithm. It is unknown what the direct cause is. One of the reasons could be that the NM system passed from version 26 to 27 during AIRAC 503. A larger number of routes were proposed than in the previous version. Thus it could happen that a route with a lower `MSG_ORDER` was matched by our algorithm. But it is very difficult to draw a conclusion.

We can see from Tab. 7.2 that our algorithm manages to cover almost 100% of the flights. Even though we adapted the algorithm for tactical and strategic analysis, we still achieved almost identical results as the 4D matching by GRRT. What we are more interested in, are the so called "outliers" that each of the algorithms exclusively matched. For that, we take a small sample and analyze AO2 and AIRAC 502 in the following subsection.

7.1.1 Flights Matched Exclusively by Our Algorithm

A total of 4 flights were matched by our algorithm, as shown in Fig. 7.1. The proposals accepted by the AO are indicated with a blue line, while the red line represents the route for which the algorithm generated the proposal in blue.

Route 1: In Fig. 7.1a, we can observe the route of the flight clearly has a big deviation and the AO chose a shorter route.

```
CHG:  N0434F380 RASVI M609 BAVTA N873 TUSKA DCT EEL
REROUTE: N0434F380 RASVI2A RASVI DCT DOROR DCT EEL EEL1A
```

Route 2: In Fig. 7.1b, we can observe that the route is using DCT instead of another route which resulted in shorter route.

```
CHG: N0448F390 LOPIK N852 LNO DCT SUTAL UN852 GTQ UZ343 BEGAR
   DCT ODINA DCT ALAXI DCT AMANO DCT VAKOR DCT PELEN Q789 UXUTA
   UXUTA1Q
REROUTE: N0448F390 ROVEN3X LOPIK N852 LNO DCT SUTAL UN852 GTQ
   UZ343 BEGAR DCT ODINA DCT SRN DCT TINKU DCT VAKOR Q789 UXUTA
   UXUTA1Q
```

Route 3: In Fig. 7.1c, it is hard to notice the difference, but when we look at the flight plan, we see that they do not pass through 3 points `OSN, EHZOF, AMSAN`.

```
CHG: N0450F380 BALTU DCT DONAD DCT VARIK L602 SODRO DCT KATCE
   DCT WRB DCT OSN/N0422F360 L980 EHZOF/N0377F260 L980 AMSAN
   T281 NORKU
REROUTE: N0450F380 BALTU5A BALTU DCT DONAD DCT VARIK L602 SODRO
   DCT KATCE DCT NORKU/N0377F260 NORKU2A
```

Route 4: In Fig. 7.1d, we can observe the route of the reroute flight is a slightly longer and passes through more points.

```
CHG: N0457F380 DIPIR V25 ARBOS UL47 IXILU UN853 IBERA DCT RITAX
   M624 BUB Y28 HELEN
REROUTE: N0457F380 DIPIR6A DIPIR V25 ARBOS UL47 EPL UM624 ROUSY
   DCT IDOSA DCT BUB Y28 HELEN HELEN2A
```

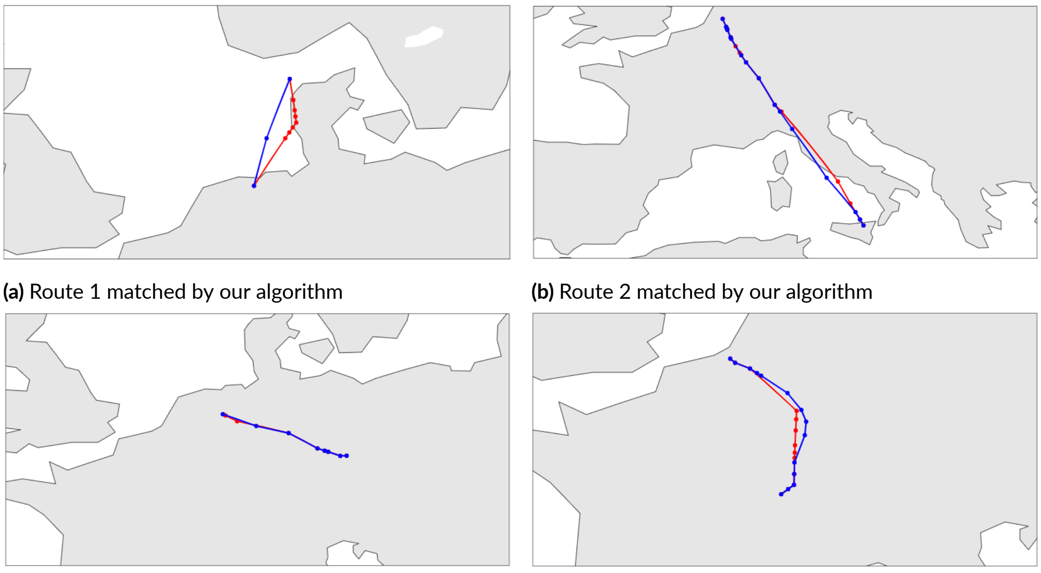

(a) Route 1 matched by our algorithm
(b) Route 2 matched by our algorithm
(c) Route 3 matched by our algorithm
(d) Route 4 matched by our algorithm

Fig. 7.1: Exclusive matches by our 2D algorithm

7.1.2 Flights Matched Exclusively by GRRT

Route 1 (see Fig. 7.3): In the first case of a flight that was not matched by our algorithm, we notice that it is a flight from North America that passes through the NAT. These flights are special because they pass through geographical coordinates when they fly on the NAT over the Atlantic Ocean, see Fig. 7.3. The NAT are updated daily by Shanwick Center (EGGX) and Gander Center (CZQX) [41]. The reason why our route matching algorithm is not matching this route is because the flight plan contains the word `NATA` followed by the point `57N020W`. Our algorithm detects `NATA` as a point, which actually exists over in the Brazilian airspace. In the context of NAT, it is seen as a route identifier. And thus we are unable to match it.

```
CHG: BERGI AMGOD SUPUR MIMVA EMLON LEGRO NALAX LIBSO ODNEK MITSO
    RIMTO HALIF GETNO MASOP APSOV SUNOT NATA 57N020W NATA 54
```

```
   N050W NATA NEEKO YBC POLTY
```

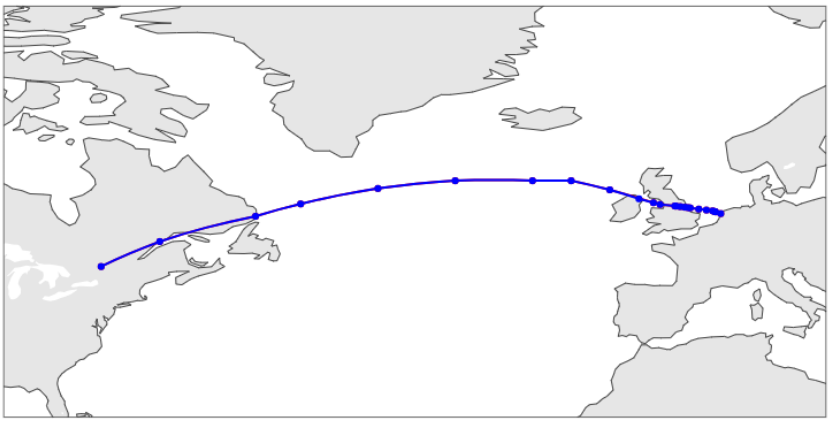


Fig. 7.2: Flight plan route over the Atlantic

Route 2: Here is another flight from over the Atlantic Ocean where instead of `NATA` we have `NATB`. We can also observe difference with the encoding of the geographical coordinates that needs further work. In the following example we see that the coordinates do not follow the same format.

```
CHG: BERGI AMGOD SUPUR MIMVA EMLON LEGRO NALAX LIBSO ODNEK MITSO
     RIMTO HALIF GETNO MASOP APSOV SUNOT NATA 57N020W NATA 54
   N050W NATA NEEKO YBC POLTY
REROUTE:
BERGI AMGOD SUPUR MIMVA EMLON LEGRO NALAX LIBSO ODNEK MITSO
   RIMTO HALIF GETNO KUGUR BILTO NATB 5430N05000W NATB MELDI
   BAREE AGLUK TAMKO MEBOK KANUR TORNI
```

Route 3 (see Fig. 7.3): The following proposal could not be matched since these are actually different routes. The reason why it is matched by the GRRT is because `HAWFA` and `YORQI` are both connecting points to the airport. If we look closely at the field 15, we see that once they connect to `HAWFA` or `YORQI` they fly straight on route `L607`. Given in Appendix B.4, you can find a solution that can be added as a second step of the 2D matching algorithm where these kind of cases will be matched.

```
CHG: N0373F230 HAWFA1X HAWFA L607 NUCHU/N0393F290 L607 ALHAD P72
     IPRIL M197 REDFA
```

```
REROUTE: N0373F230 YORQI1Z YORQI L607 ELSOF/N0393F290 L607 ALHAD
    P72 IPRIL M197 REDFA REDFA1A

CHG PARSED: HAWFA YORQI BUCFA UNZIB NUCHU ELSOF ALHAD IPRIL
   KOBBI BRAIN GASBA RATLO REDFA
REROUTE PARSED:   YORQI BUCFA UNZIB NUCHU ELSOF ALHAD IPRIL
   KOBBI BRAIN GASBA RATLO REDFA
```

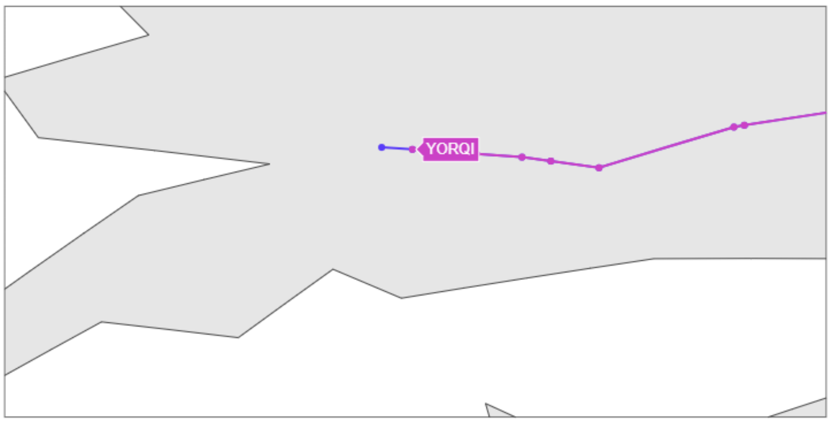


Fig. 7.3: `HAWFA` and `YORQI` points on `L607` route

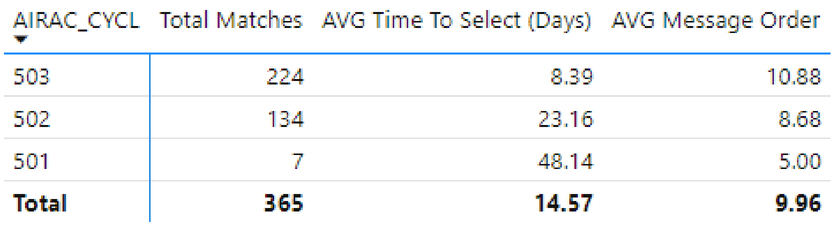
Overall, Tab. 7.1 and Tab. 7.2 suggest that our 2D algorithm performs comparably to, or even slightly better than the Generic Reroute Tool (GRRT) in terms of matching flights, and it has the potential to match flights that the GRRT might miss. The match-to-message ratios also indicate that our algorithm performs efficiently in terms of matching flights based on the recorded oplog messages. Further evaluation and fine-tuning may be needed to optimize the performance of the new proposed algorithm. After comparing the matches done by both algorithms, we see that the GRRT algorithm is not identifying some obvious matches. On the other hand, our 2D algorithm has issues with NAT flights and a very edge case reroute proposal. A potential solution for NAT flights could involve focusing exclusively on the portion of the route within the NM area, given that the proposals are specifically generated for that region.

7.2 Data Catalogue

After creating the data catalogue for the first occurrence of new segments and routes, we need to find a convenient way to analyze the results.

7.2.1 Routes

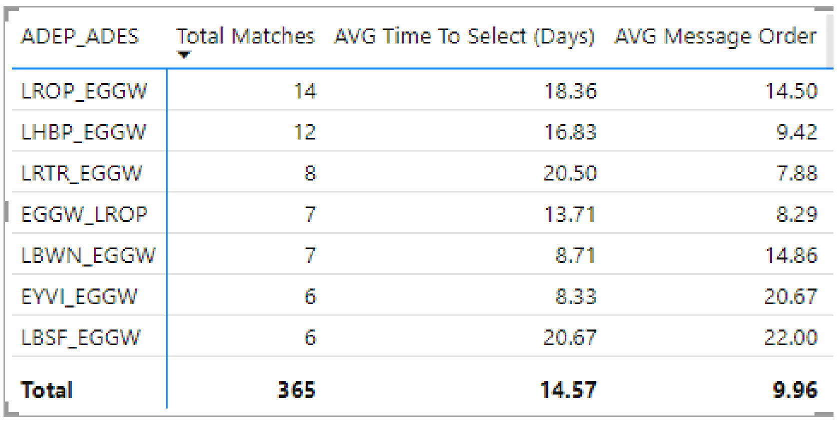
For the route uptake analysis, we are interested to see how much time was needed for the AO to use the proposed route for the next flight. As shown in Section 7.1, we can see the uptake at tactical level, on the date of the flight. But from EUROCONTROL's expert view, we know that AOs can store the routes in their database, analyze the route, and use it for another flight after a certain period of time. Since we are performing strategic analysis, it is up to the aviation experts to dive deep into the analysis in order to gain insight. However, the purpose of this analysis is to create specific metrics from which we can evaluate the uptake of the routes. For the following results of the route uptake, we will be showing the results of AO1. The figures are snapshots from Power BI tables. We first analyze the route intake on the level of AIRAC, see Tab.7.3.



AIRAC_CYCL	Total Matches	AVG Time To Select (Days)	AVG Message Order
503	224	8.39	10.88
502	134	23.16	8.68
501	7	48.14	5.00
Total	**365**	**14.57**	**9.96**

Table 7.3: Route uptake per AIRAC

We are counting the total matches, the average time for the AO to use the route once they see it and the average position of the route in the REROUTE message. The last attribute is interesting to analyze, because it reveals if the AO was usually taking the first routes or they "dug" deeper into the REROUTE message to find other interesting routes. From Tab. 7.3, we can see that there are more routes used for the first time in the later AIRACs than in 501. But we can also see that the routes which were proposed in 501 have an "average time to select" of around 48 days. On the other hand, the last AIRAC 501 has an "average time to select" of 8.39 days. This could be improved if we use longer time periods such as one year of AIRAC cycles. In Tab. 7.4, we see the top uptake of routes by `ADEP_ADES`. We immediately see that Luton airport (EGGW) has used the largest number of routes.



ADEP_ADES	Total Matches	AVG Time To Select (Days)	AVG Message Order
LROP_EGGW	14	18.36	14.50
LHBP_EGGW	12	16.83	9.42
LRTR_EGGW	8	20.50	7.88
EGGW_LROP	7	13.71	8.29
LBWN_EGGW	7	8.71	14.86
EYVI_EGGW	6	8.33	20.67
LBSF_EGGW	6	20.67	22.00
Total	**365**	**14.57**	**9.96**

Table 7.4: Route uptake per ADEP-ADES

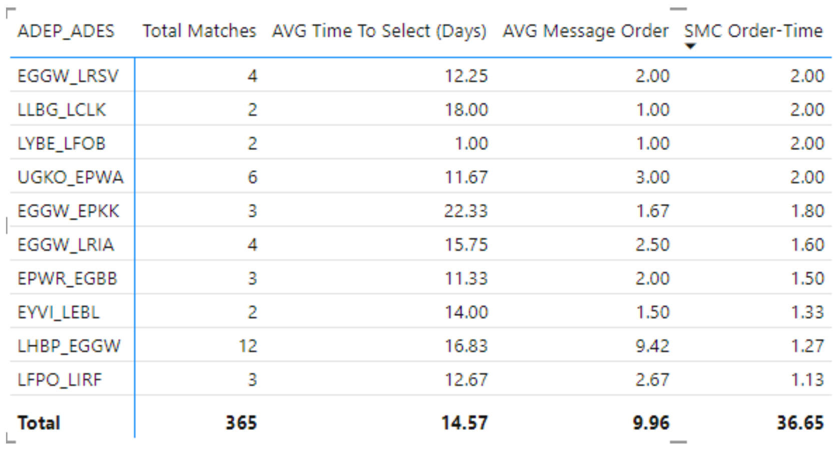
We are looking for a metric to show the efficiency of the route uptake by AO. Our goal is to achieve a higher count of route matches with a low message order score specifically for the initial messages. To do that, we use sparse mean count (SMC), where n is the number of route matches and $\bar{x}$ is the average message order of the matched routes:

$$SMC = \frac{n}{\bar{x}}$$

SMC as a metric can give use insight on how often certain events or values occur relative to their average magnitude.

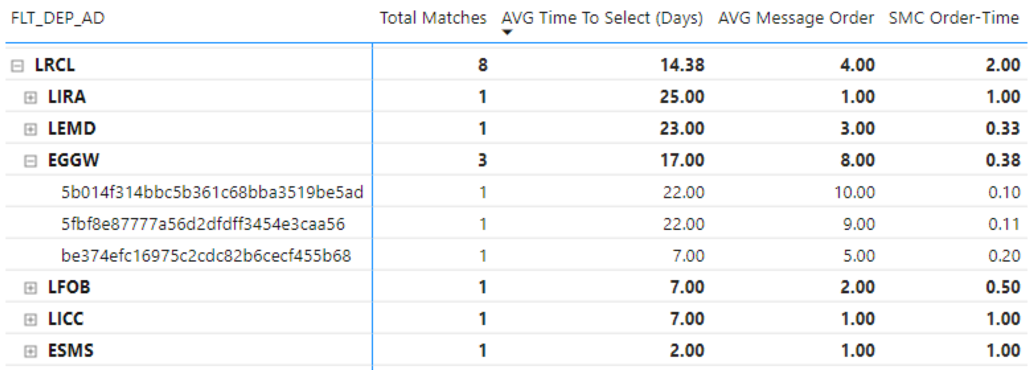
From Tab. 7.5, we have the top 10 routes that were used the most and where the AOs responded the fastest.

What holds greater importance is examining the uptake rate for individual routes. As illustrated in Fig. 7.6, we observe three approved routes from Cluj (LRCL) to Luton (EGGW) along with their respective order dates and the time it took, measured in days, to choose each route.



ADEP_ADES	Total Matches	AVG Time To Select (Days)	AVG Message Order	SMC Order-Time
EGGW_LRSV	4	12.25	2.00	2.00
LLBG_LCLK	2	18.00	1.00	2.00
LYBE_LFOB	2	1.00	1.00	2.00
UGKO_EPWA	6	11.67	3.00	2.00
EGGW_EPKK	3	22.33	1.67	1.80
EGGW_LRIA	4	15.75	2.50	1.60
EPWR_EGBB	3	11.33	2.00	1.50
EYVI_LEBL	2	14.00	1.50	1.33
LHBP_EGGW	12	16.83	9.42	1.27
LFPO_LIRF	3	12.67	2.67	1.13
Total	**365**	**14.57**	**9.96**	**36.65**

Table 7.5: Route uptake per ADEP-ADES with measures



FLT_DEP_AD	Total Matches	AVG Time To Select (Days)	AVG Message Order	SMC Order-Time
⊟ **LRCL**	**8**	**14.38**	**4.00**	**2.00**
⊞ **LIRA**	**1**	**25.00**	**1.00**	**1.00**
⊞ **LEMD**	**1**	**23.00**	**3.00**	**0.33**
⊟ **EGGW**	**3**	**17.00**	**8.00**	**0.38**
5b014f314bbc5b361c68bba3519be5ad	1	22.00	10.00	0.10
5fbf8e87777a56d2dfdff3454e3caa56	1	22.00	9.00	0.11
be374efc16975c2cdc82b6cecf455b68	1	7.00	5.00	0.20
⊞ **LFOB**	**1**	**7.00**	**2.00**	**0.50**
⊞ **LICC**	**1**	**7.00**	**1.00**	**1.00**
⊞ **ESMS**	**1**	**2.00**	**1.00**	**1.00**

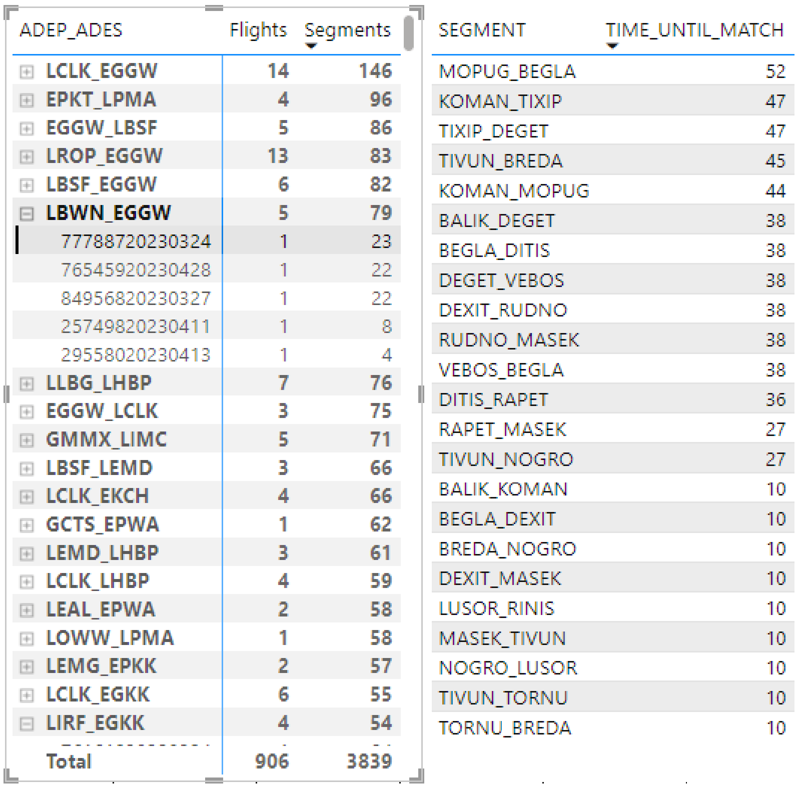
Table 7.6: Detailed view of route uptake

Since EUROCONTROL deals with thousands of flights, it could happen that the system would give a route which has already been used some time before. We would need historical data of many years to have a better picture on when each of the proposals appeared for the first time and when was the first time the AO used to route. It would be beneficial for future work to see how many times the route has been used after it was filed for the first time by the AO.

7.2.2 Segments

Similar to the routes, we want a strategic analysis for the segments. We observe the results from the methodology introduced in Section 6.2.2. The data becomes more hectic compared to the route catalogue since a route can be composed out of many segments.

From Tab. 7.7, we can analyze the first occurrence of the segments for each flight on a city-pair. We can as well see the time (in days) that was needed for that proposed segment to be used in another flight by the AO.



ADEP_ADES	Flights	Segments
⊞ LCLK_EGGW	14	146
⊞ EPKT_LPMA	4	96
⊞ EGGW_LBSF	5	86
⊞ LROP_EGGW	13	83
⊞ LBSF_EGGW	6	82
⊟ LBWN_EGGW	5	79
77788720230324	1	23
76545920230428	1	22
84956820230327	1	22
25749820230411	1	8
29558020230413	1	4
⊞ LLBG_LHBP	7	76
⊞ EGGW_LCLK	3	75
⊞ GMMX_LIMC	5	71
⊞ LBSF_LEMD	3	66
⊞ LCLK_EKCH	4	66
⊞ GCTS_EPWA	1	62
⊞ LEMD_LHBP	3	61
⊞ LCLK_LHBP	4	59
⊞ LEAL_EPWA	2	58
⊞ LOWW_LPMA	1	58
⊞ LEMG_EPKK	2	57
⊞ LCLK_EGKK	6	55
⊟ LIRF_EGKK	4	54
Total	906	3839

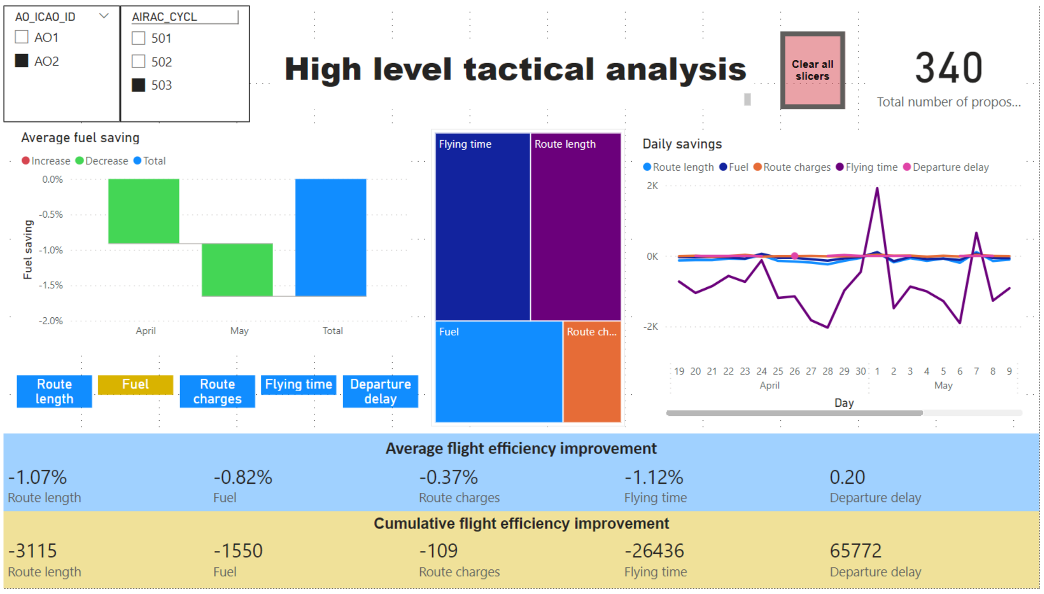
SEGMENT	TIME_UNTIL_MATCH
MOPUG_BEGLA	52
KOMAN_TIXIP	47
TIXIP_DEGET	47
TIVUN_BREDA	45
KOMAN_MOPUG	44
BALIK_DEGET	38
BEGLA_DITIS	38
DEGET_VEBOS	38
DEXIT_RUDNO	38
RUDNO_MASEK	38
VEBOS_BEGLA	38
DITIS_RAPET	36
RAPET_MASEK	27
TIVUN_NOGRO	27
BALIK_KOMAN	10
BEGLA_DEXIT	10
BREDA_NOGRO	10
DEXIT_MASEK	10
LUSOR_RINIS	10
MASEK_TIVUN	10
NOGRO_LUSOR	10
TIVUN_TORNU	10
TORNU_BREDA	10

Table 7.7: Segment uptake analysis pare ADEP-ADES

7.3 Dynamic Visualizations

Aviation data can be complex and voluminous. Having visualizations, users can have an intuitive way to interpret large datasets. This will enable to quickly understand the meaning of the data and get value from them. For our work, we want to ensure that the users, aviation experts, have a better understanding of the reroute proposals and make better decision making in the future. In order to do this, we used the tool Power BI to create dashboards regarding the strategic and tactical analysis. The statistical parameters “route length”, “fuel”, “route charges”, “flying time”, “departure delay” are used through all three dashboards.

In Fig. 7.8, we are showcasing the first dashboard for our analysis. The idea is to give the user a general picture of the flight efficiency analysis. We showcase the data on the highest granularity, by AO and AIRAC cycle. On the bottom of the dashboard, we can see the "average flight efficiency improvement" and "cumulatively flight efficiency improvement". The reason why we use average improvement is because we compare the matched reroute proposal with the flight plan message for which the opportunity was generated. It is a much better indicator than cumulative, but on the other side we can apply other formulas to the cumulative statistics to measure, for example, the amount of money the company saved by multiplying the kilograms of fuel. On the left, we see a waterfall chart. We use this to showcase the improvement made through the months. We are also able to pick one of the statistical parameters and the dashboard will show a different view. In this example, the fuel is selected. In the middle, we have a TreeMap where we can immediately view where does that AO save the most. The bigger the size of the square, the more they save. On the right, we have the savings through time. We can observe that for some days of the month there is more variety. This is more noticeable with the flying time. Like this, we can see the trends of the AO that influence their decisions.



Average flight efficiency improvement

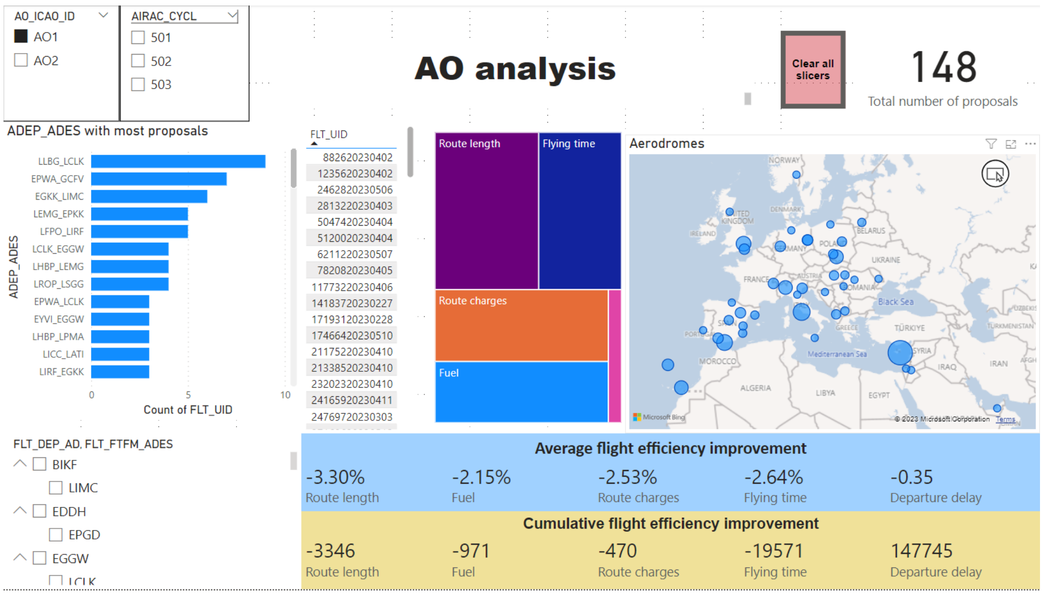
-1.07%	-0.82%	-0.37%	-1.12%	0.20
Route length	Fuel	Route charges	Flying time	Departure delay

Cumulative flight efficiency improvement

-3115	-1550	-109	-26436	65772
Route length	Fuel	Route charges	Flying time	Departure delay

Table 7.8: High level tactical analysis of reroute proposals

In Fig 7.9, we want to analyze a further level of detail and make a much broader analysis of the AO. At the bottom, as in the previous dashboard, we have the average and cumulative statistics. On the left, we can see the city-pair with the most matched routes. We can also search by the departure and arrival aerodromes. In the middle we can pick a certain flight from the selected attributes from the left. We observe the TreeMap as well as in the dashboard of Fig. 7.8. On the right, we see the bubbles around each airport that uses the matches. The bigger the bubbles, the more matches there are. We can see that the biggest bubbles are in Cyprus, UK, Italy.



AO_ICAO_ID
■ AO1
☐ AO2

AIRAC_CYCL
☐ 501
☐ 502
☐ 503

148
Total number of proposals

FLT_UID
882620230402
1235620230402
2462820230506
2813220230403
5047420230404
5120020230404
6211220230507
7820820230405
11773220230406
14183720230227
17193120230228
17466420230510
21175220230410
21338520230410
23202320230410
24165920230411
24769720230303

FLT_DEP_AD, FLT_FTFM_ADES
☐ BIKF
☐ LIMC
☐ EDDH
☐ EPGD
☐ EGGW
☐ LCLK

Average flight efficiency improvement

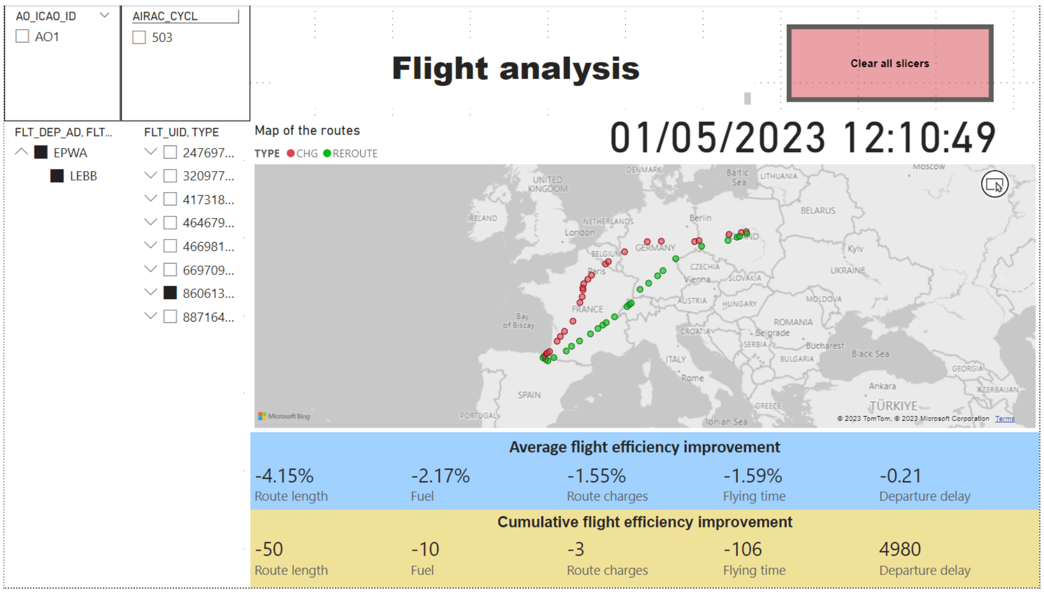
-3.30%	-2.15%	-2.53%	-2.64%	-0.35
Route length	Fuel	Route charges	Flying time	Departure delay

Cumulative flight efficiency improvement

-3346	-971	-470	-19571	147745
Route length	Fuel	Route charges	Flying time	Departure delay

Table 7.9: Analysis by AO

In Fig 7.10, we want to analyze each flight. The two bottom ribbons are showcasing the improvement, as the previous dashboards in Fig. 7.8 and Fig 7.9. On the left, we can pick the specific flight from the city-pair. Interesting here is to observe the other flights that were matched. In our case we picked one out of eight between EPWA and LEBB. On the right, we see the REROUTE in green and the CHG flight plan for which the opportunity was created. Once we hover the points, we can see the name of each waypoint. Unfortunately, due to limits of Power BI, it is not possible to create segments and custom tool should be created for better visualization.



Flight analysis

01/05/2023 12:10:49

Average flight efficiency improvement

Route length	Fuel	Route charges	Flying time	Departure delay
-4.15%	-2.17%	-1.55%	-1.59%	-0.21

Cumulative flight efficiency improvement

Route length	Fuel	Route charges	Flying time	Departure delay
-50	-10	-3	-106	4980

Table 7.10: Analysis by flight

8 Conclusion

The thesis proposes a new method that matches the flight plan proposals by the GRRT with the flight plans from the AOs that use the GRRT for their operations. With our algorithm, we successfully accomplished the objective of obtaining same results as the current GRRT matching algorithm and even find new flight plans that are exclusively matched by our approach. After simplifying the matching process, we achieved the ability to identify the routes and use them for the tactical and strategic post-ops analysis of the uptake of routes and segments. This enables a better and broader understanding of the flight reroute proposal uptake, and provides a better insight on the analysis of the GRRT tool.

After evaluating the state of the art in the field of flight plan matching, trajectory matching, methods for string pattern matching, as well as EDA of spatial data, we set up the backbone of our research. The current trajectory matching algorithms rely on clustering the flight routes based on 4D trajectories. In our study, the need for a more accurate detection of identical flight routes, as well as the need of fast processing for following analysis, leads us to focus on the string pattern matching algorithms.

The matching algorithm and the data structures that enable the following analysis are inspired by biological data processing. We create a catalogue to compare the routes that we parse and obtain an ID that uniquely identifies each route. From there, we compare the parsed flight routes by their ID. Then, we create data catalogues for routes and segments to find the first time they were proposed by GRRT and used by the AOs for their future flight. Finally, different metrics and visualization dashboards are used so that the users, aviation experts, can deep-dive into the analysis of the statistics and gain important knowledge for the uptake of routes and segments.

We remark that the current 4D algorithm used by the GRRT for matching is more conservative since the algorithm includes certain tolerance in flight level and geographic position. It mainly does not match flights that have bigger difference in flight level. This issue can be addressed if we remove the profile of the flight and we come back to a 2D representation of the flight. From there, we get inspired to use a 2D algorithm for the matching of the routes. On the other hand, it is very difficult to identify 4D trajectories. When we drop down to two dimensions, we manage to use the sequence of points for the flight and identify the route based on the unique sequence of points. Concerning the accuracy of the algorithm, we are achieving the same results as the 4D matching algorithm and we even discover other matches not found with the 4D algorithm. This proves that our 2D approach is effective for covering the routes that the 4D algorithm matched while enabling the opportunity to use the matches for further analysis. Moreover, the simplified approach enables us to identify the routes, so that we can perform further analysis that gain insight on the tactical and strategic decisions of the AOs.

Summarizing, the new flight route proposal matching algorithm enhances the ability to gain additional knowledge from the proposals since we are matching more flights than the current approach by GRRT and we can gain additional knowledge for uptake by creating a tactical and strategic post-ops analysis.

8.1 Future Work

As of the new release of the NM system, NM 27, the reroute proposal system received an upgrade where AOs can leave a like/dislike and a comment on a given reroute proposal that was suggested to them. This means that there is going to be a larger interaction between the GRRT and the AOs. In short, EUROCONTROL will have more knowledge whether the AOs who are interacting through the NMP find any of the proposals interesting and can leave comments why they would use the proposed route. By classifying the proposals, we could implement machine learning algorithms in order to predict which route has a higher chance to be liked and used by each AO.

The most amount of time is spent on extracting the data from the data warehouse. This is because the data warehouse is simply not designed for what we want to query. It could be more efficient if some design changes could be implemented like following the star or snowflake schema where we could have easy access with joins to certain fields that we need. The tables also requires some indexation on some regularly used columns.

The Carto database could be used since it has the highest level of detail. In the new version of the database, it is planned to keep all the previous AIRAC cycles and thus we would only have to implement the transformations to the database in order to use it.

There could be more improvement done in the strategic analysis and visualizations. More complex measurements could be added that the users (aviation experts) would use. This also requires more input from their side, since we need to know what are the more detailed requirements that they want to look into.

The tool can be utilized to analyze routes within the specific airspace managed by EUROCONTROL. This is because the GRRT algorithm focuses on rerouting only in that region. For instance, if a flight travels from South America to Europe, the tool will generate proposals only for the portion of the route when the flight enters European airspace. By doing so, we can effectively address the issue of unmatched North Atlantic flights in our algorithm.

There is a need for a broader analysis over a longer time period in order to confirm that the GRRT has not suggested a route that the AO used in the past. This is an important aspect that impacts the quality of the analysis and should be taken into consideration. On the other side, the ETL pipeline can select a custom period, city-pair and AO from which it can create an analysis for that period. Additionally, the implementation of a thorough analysis over an extended time frame will facilitate a more nuanced adjustment of parameters for route generation. This will effectively enhance the level of proposal acceptance while simultaneously identifying and eliminating routes that lack appeal to airlines.

A more advanced 4D matching algorithm could be developed with additional flexibility parameters since not all routes are available in all flight levels. This can be achieved through better integration of airspace data and flight information.

A
Additional Data Tables

In this appendix, we provide additional data tables related to the work.

A.1 Dataset

The following table contains information about the final dataset that we use for running the algorithm and obtaining results. We can see the name, description of each attribute and an example row.

Name	**Description**	**Example**
`FLT_LOBT`	The estimated time at which the aircraft will commence movement associated with departure	`01/01/2023 16:05:00`
`FLT_IFPS_PLN_ID`	The flight IFPS ID	`AA46008449`
`FLT_UID`	The unique flight identifier	`80454620230325`
`FLT_TACT_ID`	The flight TACT ID	`804546`
`FLT_ACFT_ID`	Aircraft registration for the flight	`AOXXXX`

Continued on next page

Table A.1 – continued from previous page

Name	Description	Example
AO_ICAO_ID	Aircraft callsign	AO
FLT_DEP_AD	Flight departure aerodrome	EHAM
FLT_FTFM_ADES	Flight arrival aerodrome	LFLL
ICAO_ACFT_TY_ID	Aircraft type	B737
AIRAC_CYCL	Airac cycle	502
OPL_EVT_ID	Oplog id of the message	2023010100990740
OPL_EVT_TIME_STAMP	Time when oplog message was posted	01/01/2023 09:00:00
MSG_ORDER	Order of the message	4
FLT_EVT_EVENT_KIND	Event type of message	REROUTE
FIELD_15	Field 15 string from flight plan	N0424F350 LOPIK3S LOPIK N852 LNO Z283 IDOSA DCT SUTAL UN852 GTQ/N0422F350 UN852 MOROK Z24 AKITO/N0437F350 Z24 BOLGI T14 MILPA MILPA7S
NB_OF_MSG	Maximal number of sub-messages for one message.	1
SYS_ROUTE_ID	Route Id given by the system. Explained in Chapter 5.2.1	EHAMLFLLG2
DEPARTURE_DELAY_S	The departure delay in seconds of reroute proposal.	4197

Continued on next page

Table A.1 – continued from previous page

Name	Description	Example
FLYING_TIME_S	The expected flying time in seconds of reroute proposal.	4121
ROUTE_LENGTH	The route length of reroute proposal.	465
SUSPENSION	Suspension of reroute proposal.	False
OPPORTUNITY	Opportunity of reroute proposal.	False
OVERLOAD	Overload of reroute proposal.	None in this example, but is an integer value.
FUEL	Estimated fuel (kg) of reroute proposal.	233
ROUTE_CHARGES	Route charges of reroute proposal.	66
ROUTE_DEVIATION	Route deviation of reroute proposal.	None in this example, but is an integer value.
MSG_ID	Id of the oplog message.	AA46008449000804546202301011300040379198806000003599
FLT_F_RTE	Field 15 of last filed flight plan.	N0424F350 LOPIK N852 SUTAL UN852 GTQ/N0422F350 UN852 MOROK Z24 AKITO/N0437F350 Z24 BOLGI T14 MILPA MILPA7S
ISGRRT	Is the rerouting matching done by GRRT algorithm.	False

Continued on next page

Table A.1 – continued from previous page

Name	Description	Example
RR_MATCH_ID	In REROUTE MATCHING messages, which SYS_ROUTE_ID was matched	Unavailable in this example, but value corresponds to SYS_ROUTE_ID
PARSED	The Parsed Field 15 only containing the points.	WOODY AMMOF NIK BUB GILOM REMBA SOPOK RITAX DEMUL DIK LIMGO SUTAL GTQ POGOL LASAT MIRGU SHARA TIRSO ARPUS TORPA MOROK AKITO DOMIL BOLGI MILPA
ROUTE_TYPE	Algorithm type used for creating the proposed route by GRRT.	pathfinder
F15_FOR_MATCHING	Parsed Field 15 without the parent point problem. This field is used for the matching.	WOODY AMMOF NIK BUB GILOM REMBA SOPOK RITAX DEMUL DIK LIMGO SUTAL GTQ POGOL LASAT MIRGU SHARA TIRSO ARPUS TORPA MOROK AKITO DOMIL BOLGI MILPA
ROUTE_ID	Unique ID of the route	c5e46f4da3f04dc64-3924458eb8a9b4c
DELTA_ROUTE_LENGTH	Difference between the route length of proposal and referenced route.	24

Continued on next page

Table A.1 – continued from previous page

Name	Description	Example
DELTA_FUEL	Difference between the fuel length of proposal and referenced route.	6
DELTA_ROUTE_CHARGES	Difference between the route charges of proposal and referenced route.	4
DELTA_FLYING_TIME_S	Difference between the flying time of proposal and referenced route.	150
DELTA_DEPARTURE_DELAY_S	Difference between the departure delay of proposal and referenced route.	-493
DELTA_ROUTE_DEVIATION	Difference between the route deviation of proposal and referenced route.	4

B
Code Listings

In this appendix, we provide the complete source code of some functionalities used in this thesis.

B.1 SQL code

The following SQL code contains the query that is used for extracting the data from the data warehouse.

```
WITH EVERYTHING AS (
    SELECT
        to_char(FLT.FLT_LOBT, 'YYYY-MM-DD HH24:MI:SS') AS FLT_LOBT,
        FLT.FLT_IFPS_PLN_ID AS FLT_IFPS_PLN_ID,
        FLT.FLT_REG_MARKING AS FLT_REG_MARKING,
        FLT.FLT_UID AS FLT_UID,
        FLT.FLT_TACT_ID AS FLT_TACT_ID,
        FLT.FLT_ACFT_ID AS FLT_ACFT_ID,
        FLT.AO_ICAO_ID AS AO_ICAO_ID,
        FLT.FLT_DEP_AD AS FLT_DEP_AD,
        FLT.FLT_FTFM_ADES AS FLT_FTFM_ADES,
        FLT.ICAO_ACFT_TY_ID AS ICAO_ACFT_TY_ID,
        FLT.AIRAC_CYCL AS AIRAC_CYCL,
        FLT.FLT_F_RTE AS FLT_F_RTE,
        MSG.OPL_EVT_ID as OPL_EVT_ID,
```

```sql
        OPL_EVT_MSG_TXT AS OPL_EVT_MSG_TXT,
        NB_OF_MSG,
        to_char(EVT.OPL_EVT_TIME_STAMP, 'YYYY-MM-DD HH24:MI:SS') AS OPL_EVT_TIME_STAMP,
        EVT.OPL_EVT_SEQ_NO AS OPL_EVT_SEQ_NO,
        EVT.FLT_EVT_EVENT_KIND AS FLT_EVT_EVENT_KIND
    FROM
        (
                SELECT FLT_LOBT, FLT_IFPS_PLN_ID, FLT_REG_MARKING, FLT_UID, FLT_TACT_ID, FLT_ACFT_ID, AO_ICAO_ID, FLT_DEP_AD, FLT_FTFM_ADES, ICAO_ACFT_TY_ID, AIRAC_CYCL, FLT_F_RTE
                FROM ARU_FLT.FLT
                WHERE AO_ICAO_ID = 'AO' AND AIRAC_CYCL=501
            ) FLT
        INNER JOIN
            (
                    SELECT OPL_EVT_ID, FLT_UID, FLT_LOBT, FLT_EVT_EVENT_KIND, OPL_EVT_SEQ_NO, OPL_EVT_TIME_STAMP
                    FROM ARU_LOG.FLT_EVT
                    WHERE FLT_EVT_EVENT_KIND IN ('FPL', 'REROUTE', 'CHG', 'REROUTE MATCHING')
            ) EVT
        ON FLT.FLT_UID = EVT.FLT_UID AND FLT.FLT_LOBT = EVT.FLT_LOBT

        INNER JOIN
            (
                SELECT OPL_EVT_ID,
                RTRIM(XMLAGG(XMLELEMENT(E,OPL_EVT_MSG_TXT,',').EXTRACT('//text()') ORDER BY OPL_EVT_MSG_SEQ_NO).GetClobVal(),',') AS OPL_EVT_MSG_TXT,
                MAX(OPL_EVT_MSG_SEQ_NO) as NB_OF_MSG
                FROM ARU_LOG.OPL_EVT_MSG
                GROUP BY OPL_EVT_ID
            ) MSG
        ON MSG.OPL_EVT_ID = EVT.OPL_EVT_ID
)
SELECT /*csv*/
       e.FLT_LOBT AS FLT_LOBT,
       e.FLT_IFPS_PLN_ID AS FLT_IFPS_PLN_ID,
       e.FLT_REG_MARKING AS FLT_REG_MARKING,
       e.FLT_UID AS FLT_UID,
        e.FLT_TACT_ID AS FLT_TACT_ID,
        e.FLT_ACFT_ID AS FLT_ACFT_ID,
        e.AO_ICAO_ID AS AO_ICAO_ID,
        e.FLT_DEP_AD AS FLT_DEP_AD,
        e.FLT_FTFM_ADES AS FLT_FTFM_ADES,
        e.ICAO_ACFT_TY_ID AS ICAO_ACFT_TY_ID,
        e.AIRAC_CYCL AS AIRAC_CYCL,
        e.OPL_EVT_ID as OPL_EVT_ID,
        e.OPL_EVT_TIME_STAMP as OPL_EVT_TIME_STAMP,

        0 as MSG_ORDER,
        e.FLT_EVT_EVENT_KIND AS FLT_EVT_EVENT_KIND,
        CAST(REGEXP_REPLACE(regexp_substr(opl_evt_msg_txt, '(ROUTE )
    ([^-]*)',1,1,null,2 ), CHR(10), ' ') AS VARCHAR(4000)) as
    FIELD_15,
        '0' as ROUTE_ID,
        NB_OF_MSG,
        --stats
         null AS DEPARTURE_DELAY_S,
         null AS DEPARTURE_DELAY_STRING,
         null AS FLYING_TIME_S,
         null as ROUTE_LENGTH, -- problem, matche
    en_route_route_length
         null as EN_ROUTE_ROUTE_LENGTH,
         null as SUSPENSION,
         null as OPPORTUNITY,
         null AS OVERLOAD,
         null as FUEL,
         null as ROUTE_CHARGES,
         null as ROUTE_DEVIATION,

         CAST(regexp_substr(e.OPL_EVT_MSG_TXT, '(\w+)(Received from:)'
    ,1,1,null,1 ) AS VARCHAR(4000)) as MSG_ID,
         null as PUBLISH,
         null as NOTE,
         null as PURPOSE,
         null as RR_EXE_ID,

        e.FLT_F_RTE AS FLT_F_RTE,
         CASE WHEN FLT_EVT_EVENT_KIND IN ('REROUTE MATCHING') THEN
             CASE WHEN OPL_EVT_MSG_TXT LIKE '%GRRT%' THEN 'TRUE' ELSE
    'FALSE' END
        ELSE null END AS isGRRT,

        CASE WHEN FLT_EVT_EVENT_KIND IN ('REROUTE MATCHING') THEN
    regexp_substr(to_char(OPL_EVT_MSG_TXT), 'Route_Id\s+(\w+)\s+'
    ,1,1,null,1 ) ELSE null END AS RR_MATCH_ID


FROM everything e
WHERE FLT_EVT_EVENT_KIND IN ('CHG', 'FPL', 'REROUTE MATCHING')
UNION
SELECT /*csv*/
        temp.FLT_LOBT AS FLT_LOBT,
        temp.FLT_IFPS_PLN_ID AS FLT_IFPS_PLN_ID,
        temp.FLT_REG_MARKING AS FLT_REG_MARKING,
        temp.FLT_UID AS FLT_UID,
        temp.FLT_TACT_ID AS FLT_TACT_ID,
        temp.FLT_ACFT_ID AS FLT_ACFT_ID,
        temp.AO_ICAO_ID AS AO_ICAO_ID,
        temp.FLT_DEP_AD AS FLT_DEP_AD,
        temp.FLT_FTFM_ADES AS FLT_FTFM_ADES,
        temp.ICAO_ACFT_TY_ID AS ICAO_ACFT_TY_ID,
        temp.AIRAC_CYCL AS AIRAC_CYCL,
        temp.OPL_EVT_ID as OPL_EVT_ID,
        temp.OPL_EVT_TIME_STAMP as OPL_EVT_TIME_STAMP,

        ROW_NUMBER() OVER (partition by OPL_EVT_ID order by
    OPL_EVT_TIME_STAMP) - 1 AS MSG_ORDER,
        temp.FLT_EVT_EVENT_KIND AS FLT_EVT_EVENT_KIND,
        regexp_substr(text_p, '(Field_15: )(.*)',1,1,null,2 ) as
    FIELD_15,
        regexp_substr(text_p, '(Route_Id: )(\w+)',1,1,null,2 ) as
    ROUTE_ID,
        NB_OF_MSG,
        -- stats
        (TO_NUMBER(REGEXP_SUBSTR(text_p, 'DEPARTURE_DELAY (\d+)m(\d+)
    s', 1, 1, NULL, 1)) * 60) +
        TO_NUMBER(REGEXP_SUBSTR(text_p, 'DEPARTURE_DELAY (\d+)m(\d+)s
    ', 1, 1, NULL, 2)) AS DEPARTURE_DELAY_S,
        regexp_substr(text_p, 'DEPARTURE_DELAY\s+(.*)',1,1,null,2 )
    AS DEPARTURE_DELAY_STRING,
        (TO_NUMBER(REGEXP_SUBSTR(text_p, 'FLYING_TIME (\d+)m(\d+)s', 1, 1, NULL, 1)) * 60) +
        TO_NUMBER(REGEXP_SUBSTR(text_p, 'FLYING_TIME (\d+)m(\d+)s', 1, 1, NULL, 2)) AS FLYING_TIME_S,
        regexp_substr(text_p, 'ROUTE_LENGTH\s+(\d+)',1,1,null,1 ) as ROUTE_LENGTH,
        regexp_substr(text_p, 'EN_ROUTE_ROUTE_LENGTH\s+(\d+)',1,1,null,1 ) as EN_ROUTE_ROUTE_LENGTH,
        regexp_substr(text_p, 'SUSPENSION\s+(\S*)',1,1,null,1 ) as SUSPENSION,
        regexp_substr(text_p, 'OPPORTUNITY\s+(\S*)',1,1,null,1 ) as OPPORTUNITY,
        (TO_NUMBER(REGEXP_SUBSTR(text_p, 'OVERLOAD\s+(\d+)m(\d+)s', 1, 1, NULL, 1)) * 60) +
        TO_NUMBER(REGEXP_SUBSTR(text_p, 'OVERLOAD\s+(\d+)m(\d+)s', 1, 1, NULL, 2)) AS OVERLOAD,
        regexp_substr(text_p, 'FUEL\s+(\d*)\S',1,1,null,1 ) as FUEL,
        regexp_substr(text_p, 'ROUTE_CHARGES\s+(\d+)\S',1,1,null,1 ) as ROUTE_CHARGES,
        regexp_substr(text_p, 'ROUTE_DEVIATION\s+(.*)\S',1,1,null,1 ) as ROUTE_DEVIATION,


        CAST(regexp_substr(OPL_EVT_MSG_TXT, '(\w+)(Flight affected)' ,1,1,null,1 ) AS VARCHAR2(4000)) as MSG_ID,
        CAST(regexp_substr(OPL_EVT_MSG_TXT, 'Publish: (.*)',1,1,null ,1 )AS VARCHAR2(4000)) as PUBLISH,
        CAST(regexp_substr(OPL_EVT_MSG_TXT, 'Rerouting Note: (.*)' ,1,1,null,1 )AS VARCHAR2(4000)) as NOTE,
        CAST(regexp_substr(OPL_EVT_MSG_TXT, 'Rerouting purpose: (.*)' ,1,1,null,1 )AS VARCHAR2(4000)) as PURPOSE,
        CAST(regexp_substr(OPL_EVT_MSG_TXT, 'Flight affected by rerouting (\d*)',1,1,null,1 )AS VARCHAR2(4000)) as RR_EXE_ID,

        temp.FLT_F_RTE AS FLT_F_RTE,

        null as isGRRT,
        null as RR_MATCH_ID
FROM (SELECT
        m.*,
        CAST(
            regexp_substr(
                OPL_EVT_MSG_TXT,
                '(.*?)(' || CHR(10)||CHR(10)||'|$)',
                1,
                LEVEL,
                'n',    -- Allow . to match newlines
                1       -- Return the first capturing group
            ) AS VARCHAR2(4000)) as text_p
            FROM EVERYTHING m
            WHERE FLT_EVT_EVENT_KIND = 'REROUTE' AND  OPL_EVT_MSG_TXT
        LIKE '% INTERESTING%'
            CONNECT BY OPL_EVT_ID = PRIOR OPL_EVT_ID
                AND PRIOR sys_guid() is not null
                AND LEVEL < REGEXP_COUNT(OPL_EVT_MSG_TXT, '(.*?)(' || CHR
        (10)||CHR(10)||'|$)') -1
        ) temp
 ;
```

B.2 Field 15 Regex

The following code is the class `F15Types` which identifies the elements of the field 15.

```
class F15Types(Enum):
    """
    We use Enum to identify the types of F15 elements.
    Many more can be added depending on the need
    """
    DCT = 0
    POINT = auto()
    ROUTE = auto()
    SID_STAR = auto()
    SEGMENT = auto()
    SPEED_ALTITUDE = auto()
    CRUISE_CLIMB = auto()
    UNKNOWN = auto()

# The regex rules which are used to identify the types
# IMPORTANT: we use literals first. Order mathers
```

```python
# Points: PRPs, Latitude / Longitude, Point / Bearing / Distance or
    Aerodrome
# Connectors: ATS routes, SID, STAR, DCT
# Modifiers: Speed and Level

regex_rules = {
    F15Types.DCT: ['DCT'],
    F15Types.POINT: [
        '[A-Z]{1,3}',
        '[A-Z]{4}',
        '[A-Z]{5}',
        '[A-Z]{1,5}[0-9]{6}',
        '[0-9]{2}[NS][0-9]{3}[EW]',
        '[0-9]{4}[NS][0-9]{5}[EW]',
        '[0-9]{2}[NS][0-9]{3}[EW][0-9]{6}',
        '[0-9]{4}[NS][0-9]{5}[EW][0-9]{6}',
    ],
    F15Types.ROUTE: [
        '[A-Z]{1,2}[0-9]',
        '[A-Z][0-9]{1,3}[A-Z]',
        '[A-Z][0-9]{2,3}',
        '[A-Z]{3}[0-9]{3}',
        '[A-Z]{3}[0-9]{1,2}',
        '[A-Z]{2}[0-9][A-Z]',
        '[A-Z]{2}[0-9]{2,3}',
        '[A-Z]{4}[0-9]{1,2}',
        '[A-Z]{2}[0-9]{2,3}[A-Z]',
        '[A-Z][0-9]{3}[A-Z]',
    ],
    F15Types.SID_STAR: [
        '[A-Z]{3}[0-9]{1,2}[A-Z]',
        '[A-Z]{5}[0-9]{1,2}',
        '[A-Z]{4,6}[0-9][A-Z]',
        '[A-Z]{5}[0-9]{2}[A-Z]',
    ],
    F15Types.SPEED_ALTITUDE: [
        "M[0-9]{3}F[0-9]{3}",
        "M[0-9]{3}S[0-9]{4}",
        "M[0-9]{3}A[0-9]{3}",
        "M[0-9]{3}M[0-9]{4}",
        "K[0-9]{4}F[0-9]{3}",
        "K[0-9]{4}S[0-9]{4}",
        "K[0-9]{4}A[0-9]{3}",
        "K[0-9]{4}M[0-9]{4}",
        "N[0-9]{4}F[0-9]{3}",
        "N[0-9]{4}S[0-9]{4}",
        "N[0-9]{4}A[0-9]{3}",
        "N[0-9]{4}M[0-9]{4}",
    ],
    F15Types.CRUISE_CLIMB: [
        "M[0-9]{3}F[0-9]{3}F[0-9]{3}",
        "M[0-9]{3}F[0-9]{3}S[0-9]{4}",
        "M[0-9]{3}F[0-9]{3}A[0-9]{3}",
        "M[0-9]{3}F[0-9]{3}M[0-9]{4}",
        "M[0-9]{3}S[0-9]{4}F[0-9]{3}",
        "M[0-9]{3}S[0-9]{4}S[0-9]{4}",
        "M[0-9]{3}S[0-9]{4}A[0-9]{3}",
        "M[0-9]{3}S[0-9]{4}M[0-9]{4}",
        "M[0-9]{3}A[0-9]{3}F[0-9]{3}",
        "M[0-9]{3}A[0-9]{3}S[0-9]{4}",
        "M[0-9]{3}A[0-9]{3}A[0-9]{3}",
        "M[0-9]{3}A[0-9]{3}M[0-9]{4}",
        "M[0-9]{3}M[0-9]{4}F[0-9]{3}",
        "M[0-9]{3}M[0-9]{4}S[0-9]{4}",
        "M[0-9]{3}M[0-9]{4}A[0-9]{3}",
        "M[0-9]{3}M[0-9]{4}M[0-9]{4}",
        "N[0-9]{4}F[0-9]{3}F[0-9]{3}",
        "N[0-9]{4}F[0-9]{3}S[0-9]{4}",
        "N[0-9]{4}F[0-9]{3}A[0-9]{3}",
        "N[0-9]{4}F[0-9]{3}M[0-9]{4}",
        "N[0-9]{4}S[0-9]{4}F[0-9]{3}",
        "N[0-9]{4}S[0-9]{4}S[0-9]{4}",
        "N[0-9]{4}S[0-9]{4}A[0-9]{3}",
        "N[0-9]{4}S[0-9]{4}M[0-9]{4}",
        "N[0-9]{4}A[0-9]{3}F[0-9]{3}",
        "N[0-9]{4}A[0-9]{4}S[0-9]{4}",
        "N[0-9]{4}A[0-9]{3}A[0-9]{3}",
        "N[0-9]{4}A[0-9]{4}M[0-9]{4}",
        "N[0-9]{4}M[0-9]{4}F[0-9]{3}",
        "N[0-9]{4}M[0-9]{4}S[0-9]{4}",
        "N[0-9]{4}M[0-9]{4}A[0-9]{3}",
        "N[0-9]{4}M[0-9]{4}M[0-9]{4}",
        "K[0-9]{4}F[0-9]{3}F[0-9]{3}",
        "K[0-9]{4}F[0-9]{3}S[0-9]{4}",
        "K[0-9]{4}F[0-9]{3}A[0-9]{3}",
        "K[0-9]{4}F[0-9]{3}M[0-9]{4}",
        "K[0-9]{4}S[0-9]{4}F[0-9]{3}",
        "K[0-9]{4}S[0-9]{4}S[0-9]{4}",
        "K[0-9]{4}S[0-9]{4}A[0-9]{3}",
        "K[0-9]{4}S[0-9]{4}M[0-9]{4}",
        "K[0-9]{4}A[0-9]{3}F[0-9]{3}",
        "K[0-9]{4}A[0-9]{3}S[0-9]{4}",
        "K[0-9]{4}A[0-9]{3}A[0-9]{3}",
        "K[0-9]{4}A[0-9]{3}M[0-9]{4}",
        "K[0-9]{4}M[0-9]{4}F[0-9]{3}",
        "K[0-9]{4}M[0-9]{4}S[0-9]{4}",
        "K[0-9]{4}M[0-9]{4}A[0-9]{3}",
        "K[0-9]{4}M[0-9]{4}M[0-9]{4}",
        "M[0-9]{3}F[0-9]{3}PLUS",
        "M[0-9]{3}S[0-9]{4}PLUS",
        "M[0-9]{3}A[0-9]{3}PLUS",
        "M[0-9]{3}M[0-9]{4}PLUS",
        "N[0-9]{4}F[0-9]{3}PLUS",
        "N[0-9]{4}S[0-9]{4}PLUS",
        "N[0-9]{4}A[0-9]{3}PLUS",
        "N[0-9]{4}M[0-9]{4}PLUS",
        "K[0-9]{4}F[0-9]{3}PLUS",
        "K[0-9]{4}S[0-9]{4}PLUS",
        "K[0-9]{4}A[0-9]{3}PLUS",
        "K[0-9]{4}M[0-9]{4}PLUS",
    ],
}
```

B.3 Lookup Points in Route

The following `lookup_points_in_route` function that finds a subset of points in between two adjacent points within a route. It is used in the field 15 parser.

```
def lookup_points_in_route(route_name: str, adjacent_point_left: str,
    adjacent_point_right: str, routes) -> list[str]:
    """Function that returns us the subset of points in between two
    adjacent points from a field 15 string
    Takes into consideration the direction of the route.
    - Finds the adjacent points and finds the sub points in the route.

    Example:
        ex. We have the following sequence of routes/points:
        input: ZMR UL155 ADORO
        UL155 is consisted of: AAA ZMR BBB CCC DDD ADORO ZZZ
        we take all the points in between ZMR ADORO
        output: ZMR BBB CCC DDD ADORO

    Args:
        route_name (str): Element that corresponds to the route name
    type
        adjacent_point_left (str): Element that corresponds to a point
    and is on LHS of route name
        adjacent_point_right (str): Element that corresponds to a point
     and is on RHS of route name

    Returns:
        list[str]: subset of points in between the adjacent points in
    route
        return can be null as we can have no lines in between
```

```python
    """

    route = get_points_from_route(routes, route_name)


    # CASE 1: the route is not in the files. Therefore, we just remove the route_name from field 15.
    # we return an empty array since we will just have to remove the route.
    # the same logic will work as when there are no points in between
    if route is None:
        return []

    # print(f"route I am searching {route_name}: {route}")

    index_left = get_index_of_adjacent_point(route, adjacent_point_left, route_name)
    index_right = get_index_of_adjacent_point(route, adjacent_point_right, route_name)

    # CASE 2: The point is not in the database of routes
    # IMPORTANT we will just leave it in the database as it is
    if index_left == None or index_right == None:
        return []
    # print(index_left, index_right)

    # dealing with no poiunts in between
    if(index_left - index_right == 1):
        # logging.debug(f"{route_name}: {adjacent_point_left}, {adjacent_point_right} - No points between")
        return []

    # dealing with direction
    # return the splice in reverse
    if(index_left > index_right):
        # logging.debug(f"{route_name}: {adjacent_point_left}, {
```

```
adjacent_point_right} - Reverse direction")
        rt =  route[index_right+1:index_left]
        return rt[::-1]

    # splicing
    return route[index_left+1:index_right]
```

B.4 Connecting Point Match

Below you can find the pseudo-code of the connecting point match.

Algorithm B.1 logic(seq1, seq2)

```
 1: function LOGIC(seq1, seq2)
 2:     Split seq1 by ' ' and assign to seq1_list
 3:     Split seq2 by ' ' and assign to seq2_list
 4:     Assign first element of seq1_list to seq1_sid_conn
 5:     Assign last element of seq1_list to seq1_star_conn
 6:     if seq1_sid_conn not in seq2_list or seq1_star_conn not in seq2_list
 7:         return False
 8:     else
 9:         Find index of seq1_sid_conn in seq2_list and assign to start_index
10:         Find index of seq1_star_conn in seq2_list and assign to end_index
11:         Slice seq2_list from start_index to end_index (inclusive) and assign to seq1_
12:     end if
13:     Join elements of seq1_list into a string with ' ' separator and assign to seq1_
14:     Join elements of seq2_list into a string with ' ' separator and assign to seq2_
15:     return whether seq1_ is a substring of seq2_
16: end function
17: function SID_STAR_MATCH(seq1, seq2)
18:     return LOGIC(seq1, seq2) or LOGIC(seq2, seq1)
19: end function
```

C
Technical Explanation

C.1 Filing a Field 15

The following sections in the appendix give additional information about the filing of the field 15 and flight plan.

The rules for filing the route part from the field 15, as from [42] are:

1. For flights along designated ATS Routes:

   - If the departure aerodrome is on or connected to the ATS route, insert the designator of the first ATS route;
   - If the departure aerodrome is not on or not connected to the ATS route, insert "DCT" followed by the joining point of the first ATS route and then the designator of the ATS route;
   - Include each point where there is a change of speed, level, ATS route, or flight rules planned;
   - After each point, add the designator of the next ATS route segment, even if it is the same as the previous one, or use "DCT" if the flight to the next point is outside a designated route, unless both points are defined by geographical coordinates.

2. For flights outside designated ATS Routes:

- Insert points that are not more than 30 minutes flying time or 370 km (200 NM) apart;
- Include each point where there is a change of speed, level, track, or flight rules planned or when required by the appropriate ATS authority(ies);
- Use ”DCT” between successive points unless both points are defined by geographical coordinates or by bearing and distance.

3. ATS route (2 to 7 characters):

   - Insert the coded designator assigned to the route or route segment;
   - Include the coded designator assigned to the standard departure or arrival route if applicable.

4. Significant point (2 to 11 characters):

   - Insert significant points on the route using their designated codes (2 to 5 characters);
   - If no code has been assigned, use degrees only (7 characters) or degrees and minutes (11 characters) to describe latitude and longitude, or use bearing and distance from a navaid or navigation point.

5. Change of speed and level (maximum 21 characters):

   - Insert the point where a change of speed or a change of level is planned;
   - Express it similarly to point (1), followed by an oblique stroke (/) and both the cruising speed and cruising level without a space between them.

6. Change of flight rules (maximum 3 characters):

   - Insert the point where the change of flight rules is planned;
   - Follow it with a space and use “VFR‘‘ if changing from IFR to VFR or “IFR” if changing from VFR to IFR.

Additional details about the flight plan can be found in the Federal Aviation Administration (FAAs) Form 7233-4 [38].

C.2 BPMN of Flight Plan Management System

In order to have a clearer picture of the flight plan management process, we can use the BPMN to graphically represent the process, see Figure C.1.

We first create two pools, first for the AO, and the second for the system that is processing the flight plans. In the second one we have two lines, one for NMOC and another for the ATCU. In the NMOC, there is the IFPS system and the ETFMS. In the second lane there are two sublines, one for the flow management position (FMP) and second for technical system of ATCU. FMP's task is to monitor sector's load and if it is required to react with regulation for sector overload. FMP can insert regulation directly into ETFMS or via communication with communication people in ETFMS.

Once a flight plan is submitted by the AO, it is received by the IFPS system in the NMOC. Here it undergoes checks on the format of the flight plan and other data conventions. This happens because it is very common for the operator to make small mistakes, for example syntax error. In case of an error, the AO is notified by IFPS that there is an issue and they need to correct it.

In case there is no error in the flight plan, the IFPS attempts to associate incoming FPL data with existing flight data.

Then, the 4D profile is made and re-validated every 30 minutes. In case of invalidated flight plan, it will be sent to AO for action which can be to change or cancel and in IFPS system it will be temporary suspended.

Once a FPL is labelled as valid, the 4D trajectory is used to locate the ATCUs that need to be communicated if the trajectory is intersecting with the respective ACC/sectors. Then, the flight plan is distributed to the ETFMS and to related ATCUs.

The ETFMS stores the flight plan and creates a additional ID (which we can see in the DWH). A monitoring of sector load is performed both by ETFMS operator and the FMP operator. In order to see if the flight overload sector capacity. It is important to say that the FMP is using the Collaboration Human Machine Interface (CHMI) to monitor sector load data from ETFMS. The coordination happens between ETFMS operator and FMP operator by series of calls or messages. If regulation are required for all overloaded sectors, they will be applied. As a consequence flights impacted by regulation will have slot with the new time of departure, meaning that a new calculated off-block time (COBT) is given for the flight. A consequence of the slot allocation is the delay that is given to the flight. This information will be sent to the AO.

In case the flight is impacted by regulation, the AO can accept it or start the procedure to cancel it and to make a new flight plan.

In case no regulation happens, they will proceed to fly with their original flight plan.

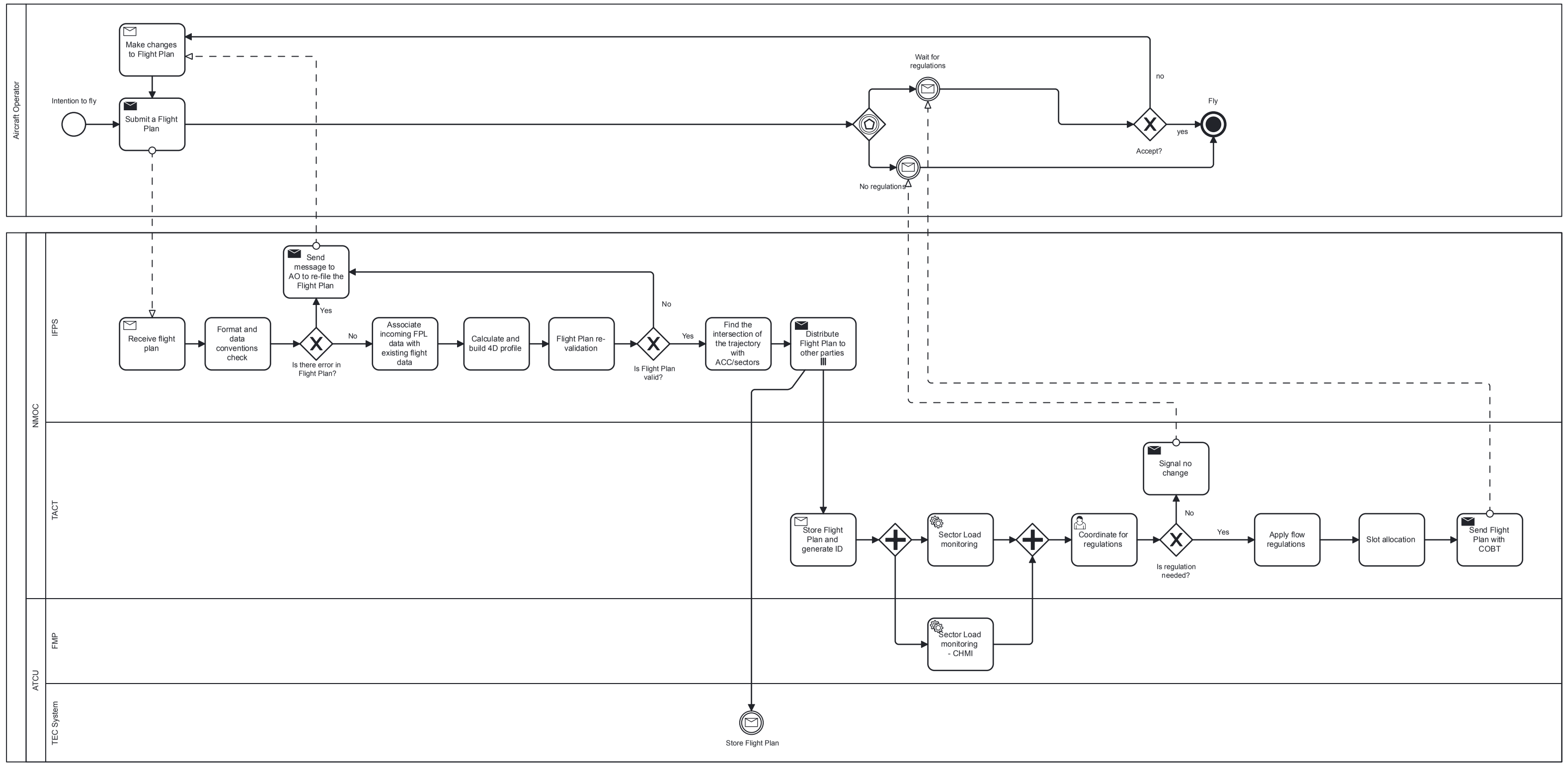

Aircraft Operator
Intention to fly
Make changes to Flight Plan
Submit a Flight Plan
Wait for regulations
No regulations
no
yes
Accept?
Fly
NMOC
IFPS
Receive flight plan
Format and data conventions check
Is there error in Flight Plan?
Yes
No
Send message to AO to re-file the Flight Plan
Associate incoming FPL data with existing flight data
Calculate and build 4D profile
Flight Plan re-validation
Is Flight Plan valid?
No
Yes
Find the intersection of the trajectory with ACC/sectors
Distribute Flight Plan to other parties
TACT
Store Flight Plan and generate ID
Sector Load monitoring
Coordinate for regulations
Is regulation needed?
No
Yes
Signal no change
Apply flow regulations
Slot allocation
Send Flight Plan with COBT
ATCU
FMP
Sector Load monitoring - CHMI
TEC System
Store Flight Plan

Fig. C.1: BPMN diagram of the FPL Management System

References

[1] EUROCONTROL. EUROCONTROL data snapshot 38. [Online]. Available: www.eurocontrol.int/sites/default/files/2023-01/eurocontrol-data-snapshot-38-revised.pdf

[2] EUROCONTROL. European aviation overview. [Online]. Available: www.eurocontrol.int/sites/default/files/2023-07/eurocontrol-european-aviation-overview-20230704.pdf

[3] A. Graser, “Data Science Workflow Framework,” 1 2020. [Online]. Available: www.figshare.com/articles/figure/Data_Science_Workflow_Framework/11638368

[4] EUROCONTROL. IFPS user manual 27.0. [Online]. Available: www.eurocontrol.int/sites/default/files/2023-03/eurocontrol-ifps-user-manual-27-0.pdf

[5] ICAO - International Civil Aviation Organization. Air traffic management. [Online]. Available: www.icao.int/EURNAT/Other Meetings Seminars and Workshops/FPL 2012 ICAO EUR Region Plan/Documentation related to FPL 2012 Amendment/Amendment 1 Doc4444.EN.pdf

[6] SkyVector. Flight planning / aeronautical charts. [Online]. Available: skyvector.com/

[7] EUROCONTROL, “DWH NM 27.0 - internal report,” 2023.

[8] EUROCONTROL. Cartography. [Online]. Available: www.eurocontrol.int/service/cartography

[9] UK Civil Aviation Authority. SIGNIFICANT POINT NAME CODES 5LNC ATS Route Name Policy. [Online]. Available: publicapps.caa.co.uk/docs/33/5LNC ATS Route Name Policy 20211221.pdf

[10] EUROCONTROL. Network operations - flight efficiency user manual. [Online]. Available: www.eurocontrol.int/sites/default/files/2019-12/no-flight-efficiency-user-manual-5.0.pdf

[11] EUROCONTROL. Enhanced tactical flow management system. [Online]. Available: www.eurocontrol.int/system/enhanced-tactical-flow-management-system

[12] ORACLE. SQL. [Online]. Available: www.oracle.com/database/technologies/appdev/sql.html

[13] Python Software Foundation. Python. [Online]. Available: www.python.org/

[14] Pandas. [Online]. Available: pandas.pydata.org/

[15] Microsoft. Power BI. [Online]. Available: powerbi.microsoft.com

[16] M. B. Cowen, "Perspective view displays and user performance," *SSC San Diego Biennial Review*, pp. 186–191, 2001.

[17] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1511–1524, 2011.

[18] E. Tiakas, A. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan, *Searching for similar trajectories in spatial networks*. Elsevier, 2009, vol. 82, no. 5.

[19] B. Han, L. Liu, and E. Omiecinski, "Road-network aware trajectory clustering: Integrating locality, flow, and density," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 416–429, 2013.

[20] M. K. El Mahrsi and F. Rossi, "Graph-based approaches to clustering network-constrained trajectory data," in *New Frontiers in Mining Complex Patterns: First International Workshop*. Springer, 2013, pp. 124–137.

[21] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[22] T. F. Smith, M. S. Waterman *et al.*, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[23] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[24] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, *Basic local alignment search tool*. Elsevier, 1990, vol. 215, no. 3.

[25] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*. La Coruna, Spain: IEEE, 2000, pp. 39–48.

[26] P. Jokinen, J. Tarhio, and E. Ukkonen, "A comparison of approximate string matching algorithms," *Software: Practice and Experience*, vol. 26, no. 12, pp. 1439–1458, 1996.

[27] E. W. Myers, "An O(ND) difference algorithm and its variations," *Algorithmica*, vol. 1, no. 1-4, pp. 251–266, 1986.

[28] J. W. Hunt and M. D. McIlroy, "An algorithm for differential file comparison," 1976. [Online]. Available: api.semanticscholar.org/CorpusID:7143186

[29] A. Graser, M. Dragaschnig, and H. Koller, "Exploratory analysis of massive movement data," in *Handbook of big geospatial data*. Springer, 2020, pp. 285–319.

[30] EUROCONTROL. Specification for ats data exchange presentation (ADEXP). [Online]. Available: www.eurocontrol.int/publication/eurocontrol-specification-ats-data-exchange-presentation-adexp/

[31] EUROCONTROL. Flight information exchange model. [Online]. Available: www.eurocontrol.int/model/flight-information-exchange-model

[32] EUROCONTROL. System for traffic assignment and analysis at a macroscopic level. [Online]. Available: www.eurocontrol.int/database/system-traffic-assignment-and-analysis-macroscopic-level

[33] EUROCONTROL. Flight efficiency plan - fuel and emisson savings. [Online]. Available: www.eurocontrol.int/sites/default/files/2019-04/airspace-flight-efficiency-plan-aug2008.pdf

[34] EUROCONTROL. Network manager interactive reporting dashboard. [Online]. Available: www.eurocontrol.int/dashboard/network-manager-interactive-reporting-dashboard

[35] EUROCONTROL, "Pan european repository of information supporting the management of EATM - internal report," 2023.

[36] EUROCONTROL. European AIS database. [Online]. Available: www.eurocontrol.int/service/european-ais-database

[37] A. Cook, *European air traffic management: principles, practice, and research*, 1st ed. Ashgate Publishing, Ltd., 2007.

[38] Federal Aviation Administration (FAA). Appendix a. FAA form 7233-4 - international flight plan. [Online]. Available: www.faa.gov/air_traffic/publications/atpubs/fs_html/appendix_a.html

[39] P. Venton. ICAO field 15 parser. [Online]. Available: github.com/pventon/ICAO-F15-Parser

[40] Python Software Foundation. Hashlib. [Online]. Available: docs.python.org/3/library/hashlib.html

[41] Federal Aviation Administration (FAA). North Atlantic Tracks. [Online]. Available: www.notams.faa.gov/common/nat.html

[42] SKYbrary. Flight plan completion. [Online]. Available: www.skybrary.aero/articles/flight-plan-completion

Acknowledgments

I want to express my gratitude to everyone who supported me on this journey. Especially to my family, relatives, friends and loved ones who were with me during these challenging months. Thankful to all the friendships that I made during my semesters in Brussels, Barcelona and Padova.

I am very grateful to Denis Odić who gave me the opportunity to write this thesis in EUROCONTROL. Together with Boris Radovanović, they spent countless of hours supervising, helping, and sharing their knowledge. Thank you to all other EUROCONTROL staff that was here to help me. Thankful to Prof. Luigi De Giovanni for being my supervisor.

I sincerely appreciate the decision of the consortium of the Big Data Management and Analytics (BDMA) Erasmus Mundus Joint Master Degree for offering me an Erasmus Mundus Scholarship. I deeply wish that they continue this amazing story.

Filip Sotiroski