# Clustering and Complexity Estimation for Air Traffic Flow Management

**Joana Inês Gonçalves Pires Costa**

Thesis to obtain the Master of Science Degree in

## Aerospace Engineering

Supervisors: Prof. Nuno Filipe Valentim Roma
Eng. Delfim Pedro Martins do Rego

### Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Eng. Delfim Pedro Martins do Rego
Member of the Committee: Eng. Thomas Dubot

**January 2021**

To my family.

# Acknowledgments

My gratitude to my Supervisors, Prof. Nuno Roma and Eng. Delfim Rego, for the guidance and motivation throughout the project.In the absence of their advices and extensive knowledge this project would neither be possible nor have the same results. Eng. Delfim for the experience on ATFM software, for helping me on algorithmic solutions and exploration of certain ideas, and Prof. Roma for the academic experience, for guiding me on how to organize my dissertation, explain and express the contribution of this research.

I also would like to thank all the people who I have reached out throughout the duration of this project - Christian Gallego (Air Transport and Airports Department,Universidad Politécnica de Madrid), Tanja Luchkova and Ingrid Gerdes (DLR) and Xavier Olive (Onera) - to explore certain ideas that came out in their articles. Lastly, to Tomas Dubot, an Onera's researcher for computational and artificial intelligent (AI) methods applied to air traffic flow management, for his final review and suggestions to take further.

To the people that I came across, during my time on Residência Maria Imaculada, for the coziness they try implement on all the girls that decide to be with them throughout college. To my long-time friends from high-school for the joyful moments, effort and support. To my faculty friends for the projects, studies together and quality time. Lastly, to Architect Noé, an old friend from family, for his small, full of ideas life lessons which motivate to curiosity, creativity, self-determination and ongoing learning on several topics (art, history, languages) along with my academic and working continuous studies.

To my family for the love, understanding, presence and support, as well as happiness, confidence, certainty, strength and determination.

Cover: UK Air Traffic Control Press, *flickr.com*; Credit: NATS.

# Resumo

A gestão e controlo de Tráfego aéreo é considerada uma das profissões mais exigentes no mundo, uma vez que os controladores são responsáveis pela segurança de milhares de passageiros. Vários sistemas computacionais foram desenvolvidos para identificação de rotas de voo típicas e para a medição de congestionamento e complexidade. Estas ferramentas ajudam os controladores a antever a trajetória de uma aeronave e alertam para a existência de conflitos, dando assim mais tempo aos controladores para procurar solucões viáveis.

Neste contexto, esta dissertação foca-se na identificação de rotas típicas de tráfego aéreo e na determinação de complexidade do espaço aéreo. De forma a concretizar estes objetivos, um novo método de avaliação é proposto, que compara diferentes técnicas de agrupamento de rotas aplicadas ao tráfego aéreo. Ao ordenar as várias técnicas de agrupamento, permite assim determinar os resultados da melhor técnica. Por outro lado, também é proposto um novo método de estimação do esforço dos controladores, tendo por base os dados de tráfego e a configuração do espaço aéreo. Este último método pode também sugerir a melhor configuração de setores para um intervalo de tempo de interesse.

A avaliação experimental realizada demonstrou que o algoritmo de agrupamento OPTICS, aliado a uma fase de pré-processamento PCA com duas componentes, é a melhor combinação para o processamento de dados de tráfego aéreo. Também foi possível definir uma ferramenta capaz de identificar o cluster correspondente a uma trajetória em tempo real, sem necessidade de processar novamente quantidades consideráveis de dados. No que respeita ao esforço dos controladores, os resultados obtidos mostraram-se capazes de estimar de forma precisa e prever a complexidade no espaço aéreo, permitindo antecipar a alteração da configuração de setores para um intervalo de tempo de interesse.

**Palavras-chave:** Gestão de Tráfego Aéreo, agrupamento, rotas típicas, indicadores de complexidade, parâmetros de CAPAN, carga de trabalho do controlador aéreo, configuração de setores.

# Abstract

Air traffic control is usually considered one of the most demanding and stressful jobs in the world, as they are responsible for the safety of thousands of passengers. To support their tasks, several computer-aided systems have been developed, helping them to identify typical aircraft routes they have to control, and measuring air traffic congestion and complexity. Such tools help them to know and better manage the airspace where they work and predict time periods of excess of workload to provide them more time to process the viable solutions.

In this context, this thesis focuses on identifying and clustering typical aircraft routes and on computing the air space complexity. To achieve these objectives, a new method is proposed to evaluate and compare different clustering techniques applied to traffic datasets, by sorting them according to the best clustering result. On the other hand, it is also proposed a new method to estimate the controller's workload based on traffic data and airspace volume configuration. This last method can also suggest the best configuration for a time interval of interest.

The conducted experimental evaluation showed that the OPTICS clustering algorithm, allied with a pre-processing phase based on a 2-components PCA, is the best combination at clustering the traffic dataset. In what concerns the controller's workload, the obtained results showed to accurately estimate and predict the airspace complexity, allowing to anticipate a sector configuration change for the time intervals of interest.

**Keywords:** Air Traffic Flow Management (ATFM), Clustering, Typical Flows, Complexity Indicators, CAPAN parameters, Air Traffic Controller Workload, Sector Configuration.

# Contents

# List of Tables

# List of Figures

# Glossary

**ACC** Air Control Center. xii, xiii, xv, xvi, 2, 4, 6, 10, 13, 19, 30, 36, 42–44, 53, 62–64, 72, 73

**AI** Artificial Intelligence. 79

**ANN** Artificial Neural Network. 16, 17

**ANSP** Air Navigation Service Providers. 2, 12

**ATC** Air Traffic Control. xi, xv, 1, 5, 13–16, 18, 19, 29

**ATCO** Air Traffic COntroller. xv, 1, 3–6, 9, 14–16, 18, 19, 30, 64, 68, 78

**ATFCM** Air Traffic Flow and Capacity Management. 5

**ATFM** Air Traffic Flow Management. ix, 5, 7

**ATM** air traffic management. 14

**ATWIT** Air Traffic Workload Input Technique. 9–11, 13

**CAPAN** CAPacity ANalyser. vii, ix, xi, xii, xv, xvi, 5–7, 17, 18, 30, 31, 40–45, 71

**CE** complexity estimator. 41

**CIV** fraction of cluster intra-valid. 37, 54, 55, 57, 60, 61

**CRIDA** ATM Research, Development and Innovation Reference Center (in Spanish *Centro de Referencia de Investigación, Desarrollo e Innovación ATM*. 14, 15

**DBCV** Density Based Clustering Validation. 29, 30

**DBSCAN** Density Based Spacial Clustering of Application. 19, 25, 29, 32, 54, 56, 60, 77

**DSC** Density Sparseness of a Cluster. 30

**DSPC** Density Separation of a Pair of Clusters. 30

**EC** Executive Controller. 3, 5, 18, 66–68, 70, 71, 73, 76

**EUROCONTROL** European Organization for the Safety of Air Navigation. 12, 18

**FAA** Federal Aviation Administration. 11

**FL** Flight Level. xvi, 2, 13, 42, 64, 66

**FMP** Flight Management Position. 1, 6, 9, 19, 31, 62, 78

**FRA** Free Route Airspace. 1, 4, 16, 32, 78

**FTS** Fast Time Simulator. 15

**HDBSCAN** hierarquical DBSCAN. xv, 19, 24, 27–29, 32, 54, 77

**HEC** Hourly Entry Count. xvi, 40, 41, 64–67, 70, 71

**HITL** Human-In-The-Loop. 16

**IST** Instituto Superior Técnico, Lisboa. 19

**K-NN** k nearest neighbour. 29

**KL** Kullback-Leibler. 22

**MF_VC** mean of flights per valid cluster. 37, 54, 55

**MLS_NVC** mean of lateral similarity inside cluster with are non valid. 37, 55, 57, 60, 62

**MLS_VC** mean of lateral similarity inside clusters with are valid (intra- valid). 37, 54, 55, 57, 60, 62

**MUAC** Maastricht Upper Area Control Center. 18

**MVS_NVC** mean of vertical similarity inside clusters with are non valid. 37, 55, 57, 60

**MVS_VC** mean of vertical similarity inside clusters with are valid (intra- valid). 37, 54, 55, 57, 60

**NC** number of clusters. 37, 55

**NN** Neural Networks. 16, 79

**OCC** Occupancy Count. 41

**OPTICS** Ordering Points to Identify Clustering Structure. 19, 24, 26, 29, 32, 54, 56

**PASC** Pre-selection of Airspace Sector Configurations. xii, 40, 41

**PC** Planning Controller. 3, 5, 18, 66–68, 73, 76

**PCA** Principal Component Analyses. 20, 32, 54, 56, 77

**PCM** fraction of clusters that were merged comparing with the total number of clusters. 60, 61

**PIV** clusters that are inter- cluster valid, i.e., none of the other clusters is similar to it. xvi, 37, 55, 57, 60, 61

**PRU** Performance Review Unit from EUROCONTROL. 12

**R-DBSCAN** recursive DBSCAN. 19, 24, 25, 29, 32, 54, 56, 61, 77

**RVSM** reduced vertical separation minima. 13, 44, 70

**SAAM** System for traffic Assignment and Analysis at a Macroscopic level. 18

**SDF_VC** standard deviation of flights per valid cluster. 37, 54, 55

**SEC** Sector Executive Controller. 18

**SPC** Sector Planning Controller. 18

**STAM** Short Term ATFCM Measures. 5

**SWC** Silhouette Width Criterion. 29, 30

**t-SNE** t-distributed Stochastic Neighbour Embedding. 20, 32, 54, 56, 77

**TAC** fraction of trajectories assign to clusters. 37, 55, 57, 60, 78

**TAVC** fraction of trajectories assign to valid clusters. 37, 55, 57, 60, 78

**TBO** Trajectory Based Operations. 16, 17

**WL** Workload. 15

# Chapter 1

# Introduction

*”Safety has the highest priority within air traffic management.”* [1]

Air Traffic Control (ATC) is a crucial and unquestionable pillar in aviation, to ensure and maintain airspace safety and control. The main purpose of this study is to improve controller working experience, capacity to answer and undertake air traffic demand. To do so, the main focus shall go to the identification of typical routes and a new tool to determine air traffic capacity, through controller workload.

Considering that most European countries are applying Free Route Airspace (FRA) policies, giving aviation companies more freedom to plan their own flight routes, it is now important to adopt a clustering mechanism that can identify and describe the most used routes, to the Air Traffic Controller (ATCO) as well as to facilitate the tasks of the Flight Management Position (FMP) by anticipating the most common routes happening in the following hours.

Different density based clustering methods shall be analysed, improved and compared in terms of clusters’ resulting quality. The comparison between methods is done by comparing the different techniques using a variety of quality indicators, addressing a value for each (technique, indicator) given the place they occupy in comparison to the others. Then, all the values for each technique are summed up into a unique defining value that characterizes each technique, allowing to identify which one presents the best results.

This work also focus on identifying the optimum sector configuration with three to six hours in advance. In fact, six hours before corresponds to the time when most of the flight plans have already been submitted and accepted, while three hours of advance is the amount of time that is required by the FMP to have all the possible decisions in case there is conflict in the trade-off safety, complexity/capacity and existing flows.

## 1.1 Air Traffic Management and Control

### 1.1.1 ACC, sector configuration and sector

Area Control Centers are facilities responsible for controlling aircraft flying in a particular airspace volume. Figure 1.1 shows how the ACCs are organized in Europe.

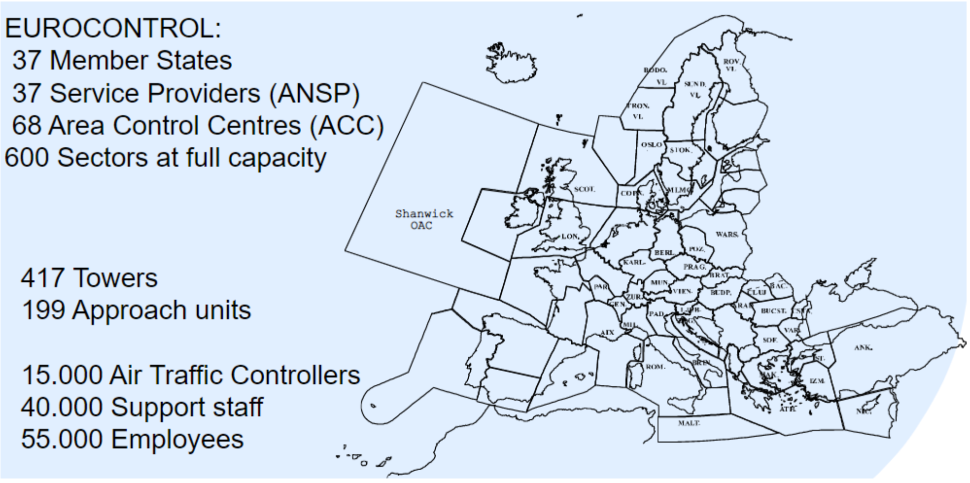


Figure 1.1: European ACCs [2]

A sector configuration divides an ACC in sectors, usually smaller airspace volumes to split the controllers responsibilities and have many controllers working at the same ACC but on designed sectors. A sector configuration is mutable, it may be composed of only one sector, in this case the sector has the same volume and coordinates as the ACC, or more than one. This way the Air Navigation Service Providers (ANSPs) manage to handle traffic Demand and Airspace Availability. When there is a lot of traffic the ACC's sector configuration has more opened sectors and when the traffic decreases the ACC's sector configuration has less opened sectors.

Very succinctly, an ACC is organized in sector configurations which change according to the air traffic demand, the sector configuration takes all the airspace the ANSP is responsible to control and the sector configurations are made by a set of sectors that serve to split controllers responsibility into smaller volumes.

### 1.1.2 Air Traffic Controller

Occupies the position of someone that deals with traffic in real-time and prepares for traffic at very close future. He has to hierarchize and decide which are the priorities and keep the calm to have a clear thinking and focused mind. There is a lot of responsibility on his/her shoulders.

Some of the tasks that he has to perform are:

- Establish communication with neighbouring ACCs and aircraft pilots;
- Monitor, report in reaching a specified flight level (FL);
- conflict search, supervision and sometimes intervention;
- last call the aircraft pilots and transmit of flight information to next ACC.

Each sector is controlled by two air traffic controllers (ATCOs), one is the Executive Controller (EC) and the other is the Planning Controller (PC), they have different tasks and responsibilities (see Table 1.1).

Table 1.1: Advantages and disadvantages of each expert-based complexity estimation method

| **EC** | **PC** |
|---|---|
| communicate with aircraft pilots (entry, exit, report on reaching specific flight level);<br>conflicts supervision and intervention;<br>supervision when conflict close to sector boundary. | communicate with neighbouring ACCs (transmit and receive information);<br>conflict search;<br>monitoring;<br>help in conflicts supervision and intervention. |

### 1.1.3 Flow Management Position

Occupies the position of planning and management, he decides the sector configurations that are going to be ahead, is aware of time intervals which may portray difficult situations and orients and supervises the air traffic controllers.

## 1.2 Improving Air Traffic Controller experience

Ensuring a good air traffic controller working experience is an important requisite to ensure his/her capacity to react and undertake agile air traffic demands. Consequently, several aspects have been considered in order to achieve this objective:

- **Time Based Separation (TBS)**, a system that defines the safety distances to land in airports given wake vortex and winds at the airport runways in real-time. This system increases the airport landing capacity and was deployed for the first time at Heathrow Airport, in 2015. Many airports, especially the ones that suffer from strong winds, are adopting the integration of this system. https://www.nats.aero/discover/intelligent-approach/.

- **Speech Recognition** for communications between controllers and pilots, a system provides on a screen the main information of what the controller has said, in a succinct way. This project started in 2014 and has perspectives to launch by 2024 [3];

- **Trajectory Prediction**, based artificial intelligence, weather, information as restricted airspaces and past data. This trajectory prediction is very helpful on estimation workload in advance and also estimate fuel costs, is one of the studies being developed at Thales.

- **Go-around detection**: mainly attributed to weather and capacity problems at airport of landing, Go-around is a safety measure, but it may also cause air traffic disruption and delays due to the time taken for the aircraft to re-position. Richard Proud [4] presents a detector of go-arounds with very good results;

- **Digital Control Towers**: instead of looking through the tower windows and turning the head to see flights coming, digital control towers are based on cameras and various other sensors. This may be a very favourable option, since these systems also correlate important information from radar, weather and sensor sources without the need for the air traffic controller to combine all this data to make the right decision.

## 1.3 Air Traffic Clusters

The Free Route Airspace (FRA) concept defines a specific volume where users can freely plan their route between predefined entry and exit points, subject to airspace availability. This concept allows the reduction of fuel consumptions and emissions, while providing flight efficiency. Currently, FRA is implemented in most European Countries. Useful tools, such as Trajectory Prediction and main-flows identification are two applications that can improve Air Traffic Controllers (ATCOs) working environment in this new way of airspace exploration.

In this dissertation, we focus on main flows identification by adopting the machine learning clustering mechanisms. Clustering is an unsupervised Machine Learning method, since it does not require pre-existing knowledge and minimises human supervision, as opposed to other machine learning methods. Clustering by itself is a pattern finding technique, which is typically used to find common behaviours or define the major characteristic of an enormous amount of data. Therefore, it is used in biology, sciences, neurology, finance and any kind of engineering solution where we want to find patterns.

At this dissertation the focus goes to air traffic clustering, through already existing methods, and defining mechanisms to evaluate and compare, adequate to air traffic datasets, alongside with some enhancements possible to be done. The study is going to be applied at a full-day ADS-B traffic data, Swiss Airspace, more specifically two upper-airspace area control centers (ACCs): the Zurich's ACC and the Geneva's ACC.

## 1.4 Air Traffic Complexity Estimation

There are many approaches on how to tackle air traffic complexity. It may be in terms of airspace (sector boundaries, sector area and shape, presence of restricted airspaces, number of active sectors), in terms of conflicts and airplanes, in terms of meteorology and/or in terms of flows.

In this study, the main focus were the air traffic controllers (ATCOs). Usually there is a set of indicators defined as factors of complexity such as the number of flights in the airspace, the number of horizontal and vertical interactions. To set how those factors correlate between each other, the actual complexity value in a given situation must be known: the *reference value*.

In the majority of other studies the reference value was obtained asking a group of controllers to evaluate certain air traffic situations, usually in a real-time simulator. At this project, the reference value is based on a set of tasks that controllers have to perform during their work and the mean time each task takes to be performed.

This reference value is very reliable and, opposing, to the most used techniques until now, it can be always applied to past studies, present and future. However, the fact that it is so detailed may cause it impracticable to give answers and update in a useful time during ATCOs operations.

At this respect, the estimation of airspace complexity allows the prediction of periods with demanding workload, which can be one approach to predict ATC capacity more accurately and consequently be a creditable tool in order to re-route flights that are causing a considerable increase of workload or, in a worst scenario, delay one.

### 1.4.1 Reference Value

*Complexity Estimation* depends on the sector configuration, weather and 4D air traffic, airport problem, restricted airspaces and air traffic controllers. At this analyses an approach based on controllers workload is proposed. The CAPacity ANalyser (CAPAN) parameters are 38 tasks, very detailed and specific, defined by EUROCONTROL who tries to table all kinds of tasks the controllers need to perform [1]. This table (included in Appendix A.2) also tells the amount of time each controller spends on each task, given that he/she is EC or PC. Throughout this dissertation, the CAPAN tasks are identified by the numbering presented in Appendix A.2.

Complexity estimation may have multiple applications such as deciding Short Term ATFCM Measures (STAM), selecting optimal sector configurations or switching to another sector configuration in the existence of hotspots (time intervals of workload demand beyond the *safety* level). This work presents a proposition of optimal sector configurations between a given $t_0$ and $t_1$ time period.

## 1.5 Motivation

As Figure 1.2 portraits, Air Traffic Flow Managment (ATFM) responsibilities include three main stages: Strategic, Pre-tactical and Tactical.

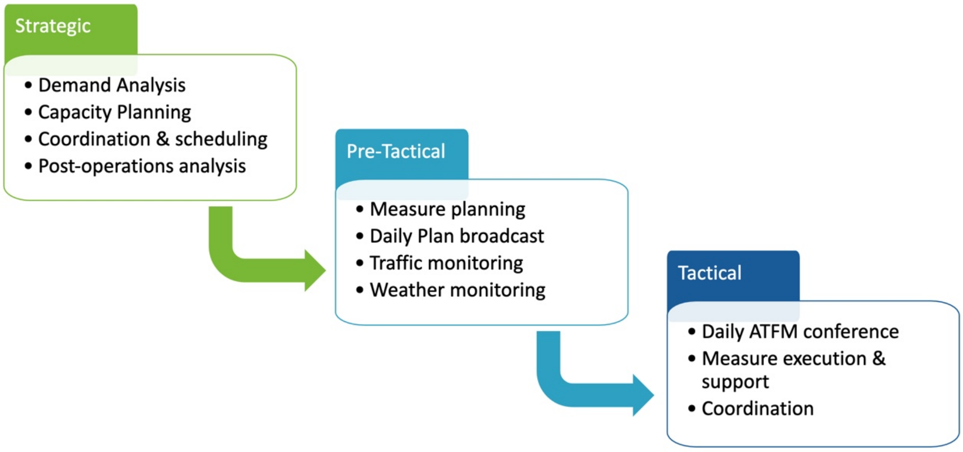


Figure 1.2: ATFM Phases

The Strategic phase is performed within days or months in advance and it can include post-operation analysis. The Pre-tactical is performed in three to six hours in advance: six hours before since most of the flight routes have already been submitted and confirmed and three hours before because most of the flights had not yet taken off. Above six hours, the accuracy of capacity-balance estimation decreases. Finally, the Tactical stage occurs almost in real-time or very close to, and refers to dealing with unexpected events and change of events, such as weather, some airport with problems, monitoring, supervision and control.

However, while the Strategic and Pre-tactical stages are mostly a responsibility of the FMP, the Tactical is a responsibility of the ATCO. In accordance, this study focuses on questioning the questions at Figure 1.3.

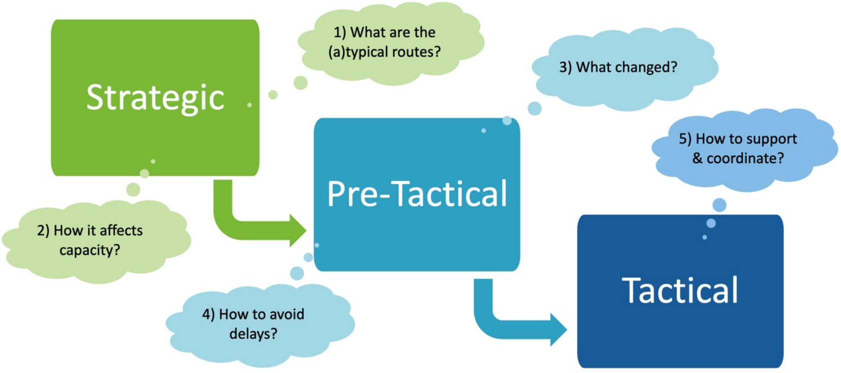

1) What are the (a)typical routes?
Strategic
3) What changed?
5) How to support & coordinate?
Pre-Tactical
2) How it affects capacity?
Tactical
4) How to avoid delays?

Figure 1.3: Questions per ATFM Phases

## 1.6 Objectives

The main aim of this thesis is to propose new tools that can be added to the already existing ones to improve controllers (ATCOs) and FMPs working experience and also to promote new possible measures which are simulated before applying.

Two main topics are studied:

- *Main Flow Determination*: the application (and comparison) of different clustering techniques to identify main flows, which also includes the definition of an evaluation method to validate the clustering results and benchmark them.

- *Complexity Estimation*: the implementation of a tool based on CAPacity ANalyser (CAPAN) parameters .With this method of complexity definition, an algorithm to select the best sector configuration will be created.

With the *Main Flow Determination* it is easier to visualize flight routes and identify patterns, at Figure 1.4 in the left are the daily traffic trajectories at the upper airspace, Switzerland Area Control Centers (ACCs), in the right are the typical routes identified from the dataset on the left. It is easy to see, that there is a much better understanding when analysing clustered data.

The *Main Flow Identification* also lets the computation of a task reported on the CAPAN which is "conflict search to establish sector planning clearance": whenever an aircraft comes in a sector, it is

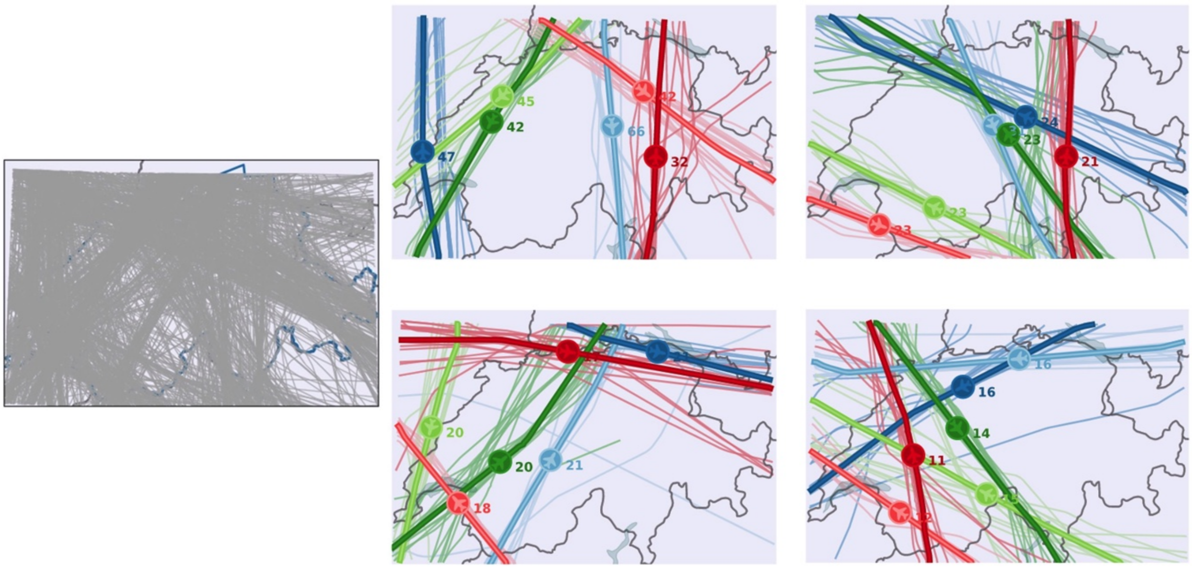


Figure 1.4: At the left the routes of a day without labelling (grey), at the right the typical routes identified.

checked if the route belongs to one of the main flows already present on the sector or if it is an atypical route or from another cluster, in these two last cases the task ”conflict search to establish sector planning clearance” is taken into account.

*Complexity Estimation*, may be used by Air Traffic Flow Managements (ATFMs) to estimate the complexity at each sector and take a more informed decision when choosing a sector configuration for each time interval. On hotspot detection, i.e., short time intervals when it is expected a lot of complexity, the ATFM can opt for a level capping or a re-route of a flight that will cause a lot of conflicts.

## 1.7 Thesis Outline

In the first Chapter, a brief exposure of where this project inserts is presented together with the enumeration of some projects are mentioned currently being developed or deployed at air traffic management.

In chapter 2, the state of the art on Clustering and Complexity is presented. The analysis of some clustering methods and the clustering techniques that are going to be used are explained in more detail.

In chapter 3, the proposed method to evaluate the quality of the clusters, obtained from each clustering method, is explained and divided into two sets: the intra-cluster (inside same cluster) and the inter-cluster (between different clusters). The main principles of a tool, called Best-Fit-Cluster, are presented. Best-Fit-Cluster is used as a faster and viable application in the day-to-day basis, to identify to which cluster does a trajectory belongs to. This tool does not substitutes the clustering methods, instead, it uses the clusters obtained by the best clustering method at the dataset. At last, the second topic, based on CAPAN Parameters, is aimed at determining the complexity based on the controller workload and also on each flight.

In Chapter 4, the clusters are evaluated by quality parameters and the process of sorting and finding the best clustering method. At the best-fit-cluster are depicted the results for specific hours and the comparison with the results provided from the clustering technique. At the Complexity Estimation, an analysis is performed over the suggested sector configuration for each time interval and the complexity inherent to the sectors.

In the Conclusion, Chapter 5, a brief summary of the best results is presented, some comments are made and suggestions for future work and improvements.

# Chapter 2

# Background

At this chapter, the focus is to introduce existing methods about *Trajectory Clustering* and *Complexity Estimation*.

*Trajectory Clustering* is useful to identify typical routes, this identification will be provided to FMPs and ATCOs to facilitate the visualization of the traffic they will be managing and help them anticipate and take secure decisions.

*Complexity Estimation* will go through methods used to define complexity at air traffic. Having a strong complexity predictor will help the FMPs and ATCOs take decisions in advance and by having an accurate indicator of complexity be able to tranfer traffic to times of less complexity and eventually accept an increase of traffic, safely and consciously.

## 2.1 Complexity Estimation

Air Traffic Complexity is a topic that has been studied since 1960. In 1978, Schmidt [5] studied the qualitative relations between workload (strain, fatigue) and the controller performance.

Hurst and Rose [6] were the first to measure the correlation of expert workload ratings with traffic density having come up with a correlation value of 53%.

Stein [7] used an Air Traffic Workload Input Technique (ATWIT) in which controllers reported workload levels during simulation. After a regression analysis, they were able to explain 67% of the variance. Four factors were considered: localised traffic density, number of handoffs outbound, total amount of traffic, number of handoffs inbound. At appendix A.1 a small compilation of some real examples of air traffic situations with high complexity reported in 2017 in Zagreb's Airspace [8].

### 2.1.1 Traffic Complexity

There is no one definition for air traffic complexity accepted by all members working at the aviation sector. It is a variety of indicators considering different layers of aviation and aspects too: aerodromes, airports, technology the ATCOs work with, weather, airspace, flow organization, conflicts existence, flight phases, traffic mix, individual differences, controller cognitive strategies, controller workload, restricted airspaces, easiness or difficulty in communicating with neighbouring ACCs, etc.

### 2.1.2 Three ways to estimate complexity

Petar Andraši et al. [8] defined three methods of air traffic complexity estimation: **expert-based**, **indicator-based** and **interaction-based**.

#### Expert-based Air traffic complexity estimation

There are, at least, three methods of expert-based complexity estimation: static image, over-the-should, real-time human-in-the-loop. Table 2.1, explains the main principle of this three techniques and the benefits and drawbacks.

Table 2.1: Advantages and disadvantages of each expert-based complexity estimation method

| Method | Principle | Benefit | Drawback |
|---|---|---|---|
| Static-<br>-image | the controller evaluates a traffic situation by its complexity by analysing a picture of it. | The controller has time to think and assess the traffic situation. | it is not very real because traffic is a dynamic situation, it is difficult to have an accurate static-image based on a real air traffic situation; the sense of urgency and time to solve the matter are not considered. |
| Over-the-<br>-shoulder | One controller is under a simulation or in real-life control controlling the air traffic he/she encounters, another air traffic controller performs the scoring, by analysing both the situation and the working controller. | Better situation awareness compared to static-image scoring. | May be lacking the sense of urgency and reactions when dealing with stressful situations. May not be able to see all traffic information in order to be able to evaluate correctly. |
| Real-time-<br>-human-in-<br>-the-loop | Usually it is performed over a simulation because the controller has to stop to score the situations he is encountering over. A real-time simulator, as close as possible from the reality, is providing 4D trajectories of different characteristics, it may include weather factors. Also, someone must be simulating the pilots and the other ACCs. | The controller has the situational awareness, the one that executes is the one that scores, and sees all information he wants to. | The controller needs to interrupt what he is doing to score the traffic situation. |

Some types of scoring the complexity of traffic situation are NASA Task Load Index (NASA-TLX), overall workload (OW) and Air Traffic Workload Input Technique (ATWIT). NASA Task Load Index splits the workload assessment into six categories (metal demand, physical demand, temporal demand, performance, effort, frustration) and scores each situation in a range [0, 100].Overall Workload (OW) is not split in categories as the previous scoring method, but is also between [0, 100]. Air Traffic Workload Input Technique (ATWIT) ranged from 1 to 7, it is simpler and enough.

**Indicator-based air traffic complexity estimation**

Kopardekar et al. [9] mentioned the term dynamic density, which is also known as complexity to be ”the collective effect of all factors, or variables, that contribute to sector level air traffic control complexity or difficulty at any given time”, an FAA definition, and made a compilation of the most known complexity indicators, some of them are mentioned bellow:

1. number of aircraft divided by occupied volume of airspace also known as traffic density;
2. number of aircraft divided by sector volume;
3. CRI(convergence recognition index) measure of the difficulty of detecting converging aircraft with shallow angles;
4. SCI(separation criticality index) proximity and conflicting aircraft with respect to their separation minima;
5. DOFI(degrees of freedom index) based on maneuver options in a conflict situation;
6. aircraft distance from the sector boundary prior to handoff;
7. different formula based on the same principle as CTI1;
8. sector volume;
9. aircraft count;
10. number o climbing aircraft, number of cruising aircraft, number of descending aircraft;
11. horizontal proximity, vertical proximity;
12. time-to-go to conflict;
13. variance of speed;
14. ratio of standard deviation of speed to average speed;
15. conflict resolution difficulty based on crossing angle.
16. number of aircraft with velocity change greater than 10 knots or 0.02Mach;
17. number of aircraft with altitude change greater than 750 feet;
18. number of aircraft with 3D-Euclidean distance between 0-5 nautical miles excluding violations, number of aircraft with 3D-Euclidean distance between 5-10 nautical miles excluding violations, number of aircraft with lateral distance between 0-25 nautical miles and vertical separation less than 2000/1000 feet above/below 29000ft, number of aircraft with lateral distance between 25-40 nautical miles and vertical separation less than 2000/1000 feet above/below 29000ft.
19. angle of converge between aircraft in a conflict;
20. number of altitude changes above a threshold value within the sector, number of bearing changes above a threshold value within the sector;
21. squared difference between the heading of each aircraft in a sector and the direction of the major axis of the sector, weighted by the sector aspect ratio.

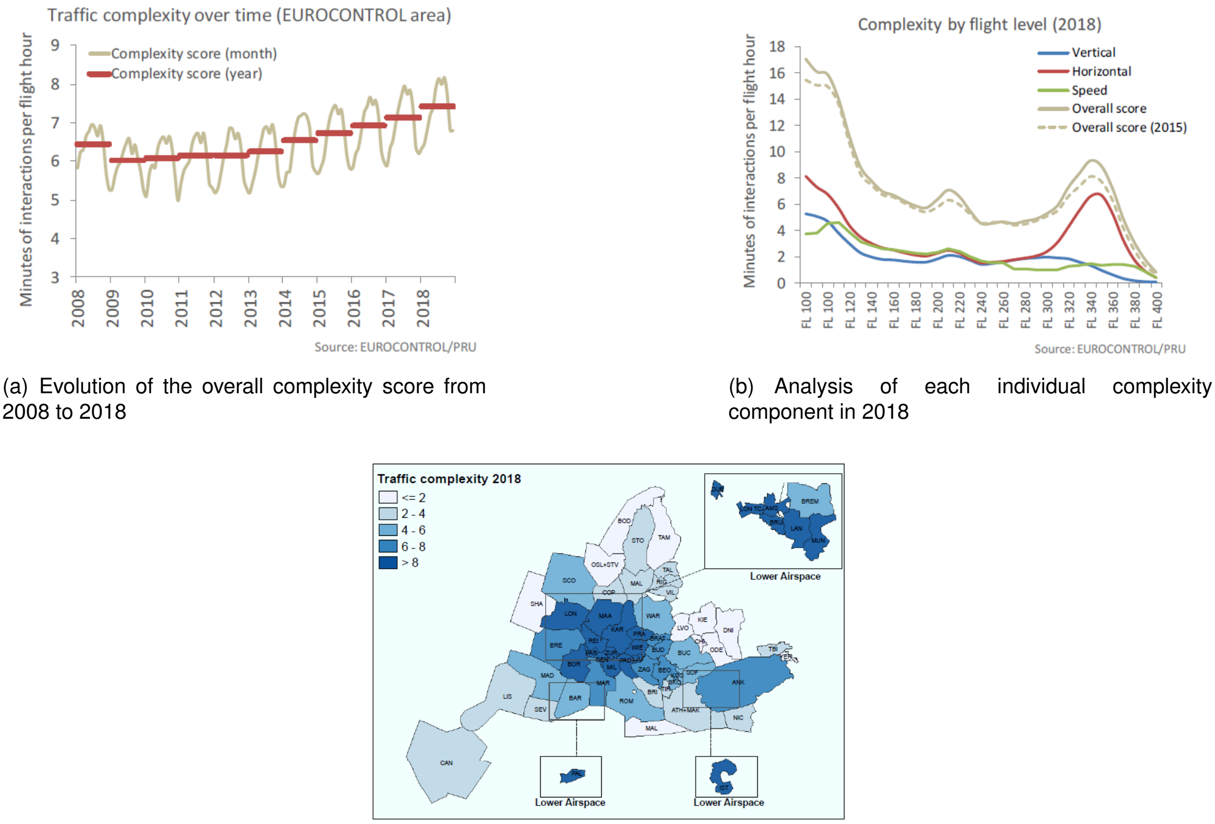
### Interaction-based air traffic complexity estimation

Petar Andraši et al. [8] focus on introducing EUROCONTROL's Report [10] on measuring complexity, it is based on aircraft-to-aircraft interactions. They divide the airspace into parallelepipeds of 20*20NM and an altitude of 3000feet and compute four indicators. Those four complexity metrics are Adjusted density, Potential vertical interactions (VDIF), Potential horizontal interactions (HDIF) and Potential speed interactions (SDIF).

After computing these four indicators, they determine the relative value of the vertical, horizontal and speed interactions indicators, which is obtained dividing VDIF, HDIF and SDIF by the adjusted density. Finally the complexity score is equal to the *adjusted density * structural index*. where the structural index is:

$$\text{Structural index} = r_{VDIF} + r_{HDIF} + r_{SDIF} \tag{2.1}$$

This method is applied at PRU Reports (Performance Review Reports), to calculate complexity on annual basis for each ANSP (air navigation service provider), it is sometimes called "PRU complexity".



(a) Evolution of the overall complexity score from 2008 to 2018

(b) Analysis of each individual complexity component in 2018

(c) average complexity score over the year in each EUROCONTROL's ACC

Figure 2.1: Complexity score graph's presented in the PRR2018 [11]

Three graphs taken from the Performance Review Report 2018 [11], at Fig. 2.1, show the complexity score method applied at the EUROCONTROL's reports. Fig. 2.1 (a) shows the complexity evolution along the years, presenting a pattern, every year in the summer months the complexity is higher and it is also possible to see that the overall complexity rose or kept from 2009 until 2018.

However to understand in which areas the complexity is greater it is necessary to analyse not only the overall score but how each indicator separately influences it, Fig. 2.1 (b) shows the fluctuation of influence of each indicator in the overall score as the flight level increases. The Flight level is the altitude where the aircraft is, FL1 = 100ft, FL300 = 30.000ft. When an aircraft is at cruising the FL must end with a zero, the aircraft may be flight at FL260, but not at FL265. Before 1982, the reduced vertical separation minima (RVSM) was 2000 feet, but since than, RVSM has implemented 1000 feet of vertical separation minima between FL290 and FL310, in order to increase capacity. Flights must be certified for RVSM [12].

The vertical indicator has larger influence at low flight levels, the speed interaction has its larger influence from FL100 to FL120, the horizontal interaction indicators is also significant at low altitudes however it has higher values from FL320 to FL370 which is the upper airspace where most flights are at the cruising stage. At Fig. 2.1 (c) they perform an analysis of the European ACCs' annual score. As it was expected, the core of Europe has complexity scores larger than the ones that are on the outskirts.

### 2.1.3 Relationship Complexity-Workload

Over the years, a variety of investigations showed a strong relationship between complexity metrics and controller workload. Hurst and Rose [6] measured the correlation of expert workload ratings with traffic density with a result of 53% of correlation. Stein [7] used the Air Traffic Workload Input Technique(ATWIT) for four metrics ( localised traffic density, number of handoffs outbound, total amount of traffic, number of handoffs inbound) having a regression correlation of 67%. Laudeman [13] introduces Dynamic Density as a combination of ”both traffic density and traffic complexity”.

Mogford et al. [14] reviewed a number of studies examining effects of ATC complexity on workload and performance. They created a model relating ”source factors” and mediating factors resulting in controller workload ( see Fig. 2.2).

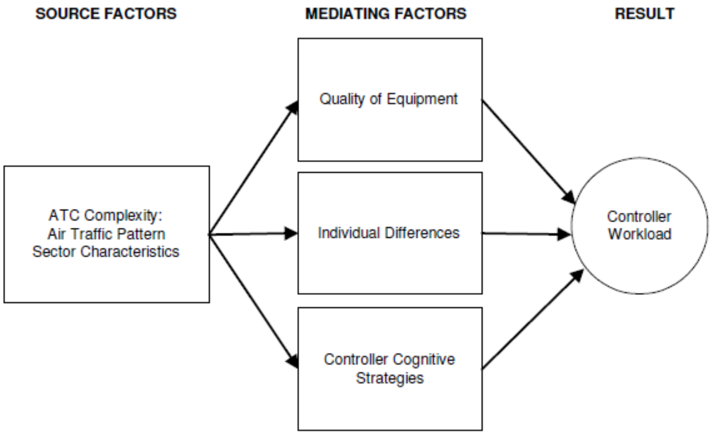


Figure 2.2: Relationship between ATC Complexity and controller workload [15]

In 1978, Schmidt [5] saw the understanding, quantification and prediction of workload factors as an important aspect in order to increase air traffic control (ATC). He studied the qualitative relations between workload measures, strain, fatigue and performance of the man in the system. In the man-system's point of view, the ATCO work has tasks such as conflict recognition and resolution, control-jurisdiction transfer (handoff) and intersector coordination. The important aspects of these tasks is the task frequency, difficulty and relative priority. As depicted in Fig. 2.3 these tasks are affected by airspace environment (traffic and weather), the hardware and software environment and other personnel.

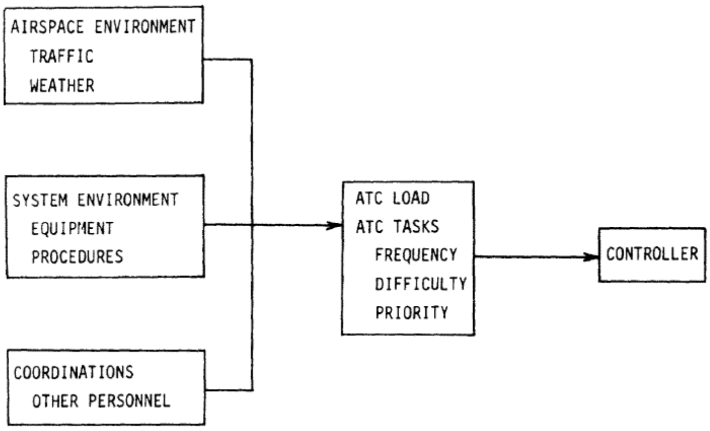


Figure 2.3: ATC load on controller [5]

The stress indicators that were analysed by Schmidt were the following:

- **Percent Busy Time** the percentage of time the controller is busy accomplishing the task $P_{busy}$. The results indicated that the maximum traffic capacities from controllers estimates, resulted in a work-load measure of 45-48 man-min/h, or a $P_{busy}$=0.75-0.80.

- **Task Delay** One stressor suggested that deals with the concept of time required versus time available to accomplish the task. The average task delay $\overline{t_d}$ rapidly increases when $P_{busy}$ is approximately 0.65-0-75.

- **Idle Period Length** ATCOs need some time to update their mental image. The update rate tends to average 1/min. For aircraft travelling at 480knots(889km/h), 1.25 min is required for aircraft to travel 10NM (the actual minimum separation). Schmidt suggests a processing rate of 40-45aircraft/h, as the controller capacity per sector for low-arrival and 50-55 aircraft/h for high-transition.

### 2.1.4 Mental Workload

Suárez et al. [16] suggest that classical capacity indicators are not efficient any more as they do not reflect the traffic complexity in short periods of time and as a consequence the system is underused. A key search problem is the ability to quantify the complexity inside sectors or in one region of airspace in short periods of time and the accuracy of the prediction. For them ATM complexity has two facets: Technical Complexity and Cognitive Complexity. In this paper they focus on cognitive complexity. Air Traffic Controller (ATCO) mental workload is typically acknowledge to be strongly related to complexity.

Suárez et al. work at CRIDA (ATM R&D+innovation Reference Centre) a reference center for the ATM in Spain.

Cognitive Processes are a function of perception (visual and audio), central processing (comprehension, strategic thinking, decision making) and response execution (manual and verbal). As shown in Fig. 2.4, cognitive processes are used to perceive, comprehend, think, make decision and execute. From the point of view of an ATC, cognitive processes are used to acquire and maintain ”situational awareness”, ”make decision” and ”respond” to a problem at hand.

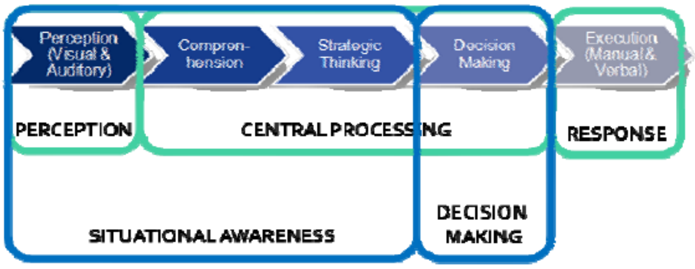


Figure 2.4: Cognitive Process ATCO [16]

The situational awareness regards the airspace environment (traffic, weather), the decision making focus on prioritising (Is there any conflict? Any handoff to be made? Any communication?) and response focus on executing what the ATCO decided to.

CRIDA developed a Mental Workload which is based in three concepts:

- *Demanded Mental Resources*(task-load): ”Physical and mental activities demanded to carry out perceptual actions, cognitive actions and motor skills. To model this concept, it is assumed that Flight Events result in Control Events that are driven by underlying Operating Concepts and that their implementation requires a specific set of cognitive resources.”

- *Available Mental Resources* ”Physical and mental abilities that an ATCO has available to provide the control service, considering only a set amount of base resources.”

- *Threshold* ” Value beyond which Demanded Mental Resources (task-load) exceeds the Available Mental Resources.”

The results from [16] investigation have proven the possibility to determine hourly sector capacities using mental workload estimations.

CRIDA developed a WAC (Workload Analysis Component) platform prototype. WAC is a Workload estimation and measurement tool which calculates the workload required per sector for a set of predefined Sector configurations. It integrates four components:

- Pre-processor ( receives Predicted Trajectories and transforms them into traffic scenario to be applied at the Fast Time Simulator (FTS))
- Fast Time Simulator ( generates the set of Control Events)
- Sector Translator (maps basic volumes to sectors, links each control event to the sector it applies depending on the implemented sector configuration)
- WL(workload) Calculator

### 2.1.5 Real-time ATC Simulator and Indicator-based Complexity System

Some researchers combined two of three approaches to complexity refred on section 2.1.2, first a simulator (*Expert-based Air traffic complexity estimation* approach) followed by the *indicator-based complexity estimator*.

One was Kopardekar et al. [17] in 2007, they presented their results in a Seminar. The simulation included six certified experienced professional controllers, 3 sectors: two high altitude and one low altitude, nine scenarios (75 minutes each) and a keypad for controllers to enter complexity ratings.

After having the simulation results they performed a regression analysis over 52 indicators getting 17 statically significant ( Aircraft count, Sector volume, Standard deviation of speed/mean of speed, Number of aircraft with predicted horizontal separation under 8 NM, angle of convergence in a conflict situation, Time-to-conflict, Horizontal proximity measure, Heading variance and Number of aircraft changing altitude, are some of them).

Another study performed by Tomislav Radišić et al. [18] aimed and analysing the implementation of trajectory-based operations (TBOs), i.e., trajectories performed over Free Route Airspace (FRA) regulations. They developed a Human-in-the-loop (HITL) simulation, to get a subjective complexity score from experienced ATCOs who would rate from 1=very low to 7=very high complexity, and later establish the correlation between defined complexity indicators.Ten experienced ATCOs working with Croatia Control Ltd. were selected, to participate in this experience.

After computing a set of complexity indicators, the researchers performed the step-wise regression of 52 complexity indicators and managed to eliminate the insignificant variables, i.e., with a p-value $>$ 0.05, the researchers also had to exclude indicators that presented an inter-correlation with other indicators, keeping the ones with greater linear correlation to the subjective scores.

Tomislav et al. reached to the conclusion that the six most important indicators, taking in consideration TBO, were:

1. Number of aircraft
2. Number of conflicts between conventional aircraft and aircraft flying according to TBO
3. Fraction of aircraft in climb or descent
4. Number of aircraft near sector boundary ($<$10NM)
5. Fraction of TBO aircraft
6. number of aircraft with 3D euclidean distance less than 5NM

Petar Andraši et al. [19] continued the studies of Tomislav [18]. They wanted to apply Neural Networks (NN) and compare with the linear model. They defined a three layered Neural Network: input layer made of 20 neurons (the number of complexity indicators), hidden layer and output layer (one neuron). They used the subjective assessment performed by the ATCOs at the Croatia Control Ltd. from 1 to 7, mentioned in [18].

To compare with the linear regression model, they calculated the correlation coefficient for the best performing ANN. Table 2.2 shows a synthesis of the results Petar et al. [19] have obtained. $R^2 - adjusted$ adjusts $R^2$ preventing spurious inflation

Table 2.2: Comparison of performance of different complexity estimation models [19]

| Model | Kopardekar et al. [9] | Linear model [18] | TBO-specific linear model [18] | ANN Model [19] |
|---|---|---|---|---|
| R | 0.83 | 0.746 | 0.833 | 0.783 |
| $R^2$ | 0.69 | 0.556 | 0.693 | 0.609 |
| $R^2$-adjusted | N/A | 0.554 | 0.691 | N/A |

The group got to the conclusion that the ANNs have an accuracy similar to the linear models. In their point of view, this means that the issues with linear model accuracy are not due to non-linear relations between the indicators and the subjective complexity score.

They also noticed *human expert inconsistency* in assessing various different scenarios could be a source of error. They verified there were a few controllers who rated the whole scenario with the same complexity score. This lead to situations with one or two aircraft well separated having the same result as a situation with 12 aircraft present simultaneously.

As future improvements, they planned to continue their investigation to find a way that requires less input from the experts and relies more in the traffic data, ”though it is impossible to explore complexity without involving humans.”

### 2.1.6 CAPAN tasks

CAPAN (Capacity Analyser) parameters try to summarize onto one single table all possible tasks air traffic controllers have to perform and the mean time they take. The time means were obtained through studies obtained in different European Countries (see Figure 2.5).

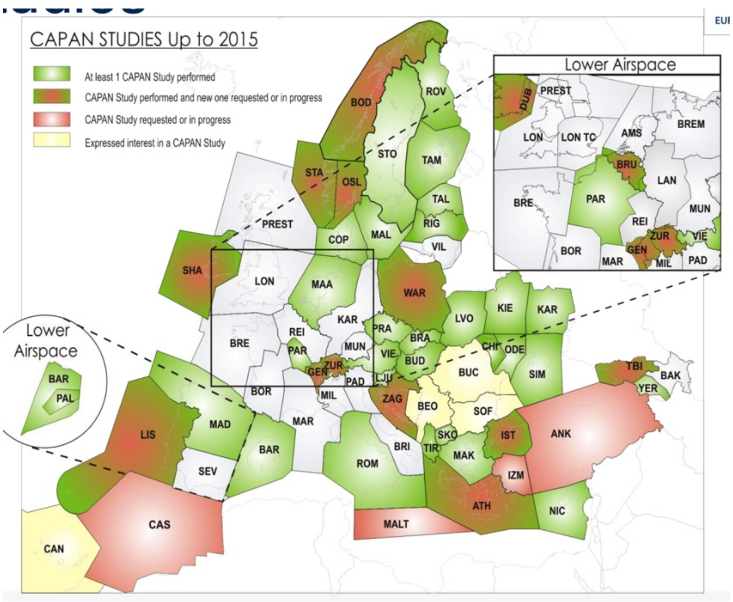


Figure 2.5: CAPAN studies performed over Europe until 2015. [20]

Appendix A.2 presents a list of the CAPAN parameters. The time means present at this table were also the ones used at the Proposed System, chapter 3.

Schickel explains there are a set of sequential tasks that controllers have to fulfil in each case and there are always two controllers working on one active sector volume, the SEC (sector executive controller) or EC (executive controller) and the SPC (sector planning coordinator) or PC .

SAAM (System for traffic Assignment and Analysis at a Macroscopic level), one of the simulation tools at Maastricht Upper Area Control Center (MUAC) [21] has CAPAN integrated in its system. Its purpose is to help improve airspace design and sector capacity, through designing and redesigning air traffic networks and sector volume shapes.

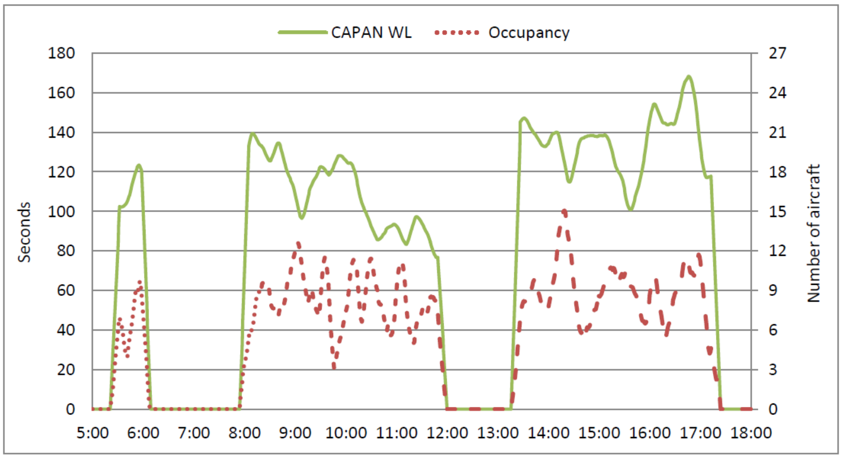


Figure 2.6: The CAPAN workload, the green line and the occupancy levels, the red dashed line for BRUSSELS EAST HIGH on Friday $10^{th}$ of June 2011. [1]

Fig. 2.6 shows that the CAPAN measured workload is not proportional to the occupancy, which leads us to conclude there are other consonants that need to be taken into account besides occupancy.

Raffaele Russo [20] presented an overview about EUROCONTROL's CAPAN Parameters. It is a methodology to determine the sector capacity correlated with the controller workload and he defines the workload threshold as 70% of the absolute working time. The 30% represents task which cannot be captured or time to rest and adapt in order to stay focused and complete the enumerable tasks successfully. Table 2.3 shows the ATCO workloads into sets.

Table 2.3: Description of different controller task-loads

| **Threshold** | **Interpretation** | **Recorded Working time during 1 hour** |
|---|---|---|
| 70% or above | Overload | 42 minutes + |
| 54% - 69% | Heavy Load | 32 -41 minutes |
| 42% - 53% | Medium Load | 25-31 minutes |
| 18% - 41% | Light Load | 11-24 minutes |
| 0% - 17% | Very Light Load | 0-10 minutes |

In 1993, EUROCONTROL Experimental Center (EEC) at Bretigny (France) and CACI Inc. developed RAMS (Reorganized ATC Mathematical Simulator) a fast-time simulation tool that also compounds ATCOs Workload.

Currently, it is called RAMS Plus it was developed and maintained by ISA Software company. According to Skybary [22] it is a simulator ”that allows the users to create a model of an air traffic control system, ATC procedures, 4D performance of over 300 aircraft, 4D conflict detection and rule -based conflict resolution, and controller actions based on the current demand”.

It was also used in two IST Thesis in 2014 and 2016 [23, 24] to estimate Lisbon Airspace Capacity and study the possibility of new arrival procedures using the Point Merge System.

## 2.2 Trajectory Clustering

Having had a glimpse of what controllers do and the kind of tasks they perform while they are working, we observe that it would increase capacity, if their working environment could be improved. Trajectory clustering leads to typical routes identification. This helps the two positions (ATCO and FMP) who are responsible to control and design ahead the sector configurations that are going to be applied. It would improve the FMP working experience because it is helpful to see the typical routes existent in a time interval of interest to decide the sector configuration to use. By the ATCOs work side, seeing if a coming flight will perform a typical trajectory, that he already happens to know and knows its frailties, will tell him the moments where he needs to be more focused at that trajectory.

Traffic data at a given ACC is organized in a matrix containing the following information:

- flight_id, callsign, icao24
- grounspeed
- altitude, latitude and longitude
- track
- vertical rate

traffic data comprehends a set of flights and has their information at a rate of 10 seconds.

In order to be able to cluster, the quantity of data matters. It was observed that clustering one hour data had much worse results than clustering for one whole day. Clustering for the most crowded three hours also did not have so good results as the whole day.

There are five main clustering methods: partitioning method, hierarchical method, fuzzy clustering, density based clustering, model based clustering.

The partitioning method splits the data into a given number of cluster, defined by the user. One of the most well-known techniques that fits in this category is k-means [25]. Studies have been taken to overcome the issue of having to give the number of apriori clusters, such as [26] in 2004, that tests the hypothesis that a subset of data follows a Gaussian distribution. However, the main focus is going to be on the hierarchical (HDBSCAN, OPTICS, R-DBSCAN) and density-based (DBSCAN) clusterings.

Hierarchical clustering can aither be from ’top-down’ (divisive clustering) or ’bottom-up’ (agglomerative) approach. Nevertheless, all the clustering methods herein revised will be based on a divisive approach.

Pre-processing methods identify main characteristics and reduce the data original dimensions to the main components, they contribute to increase the velocity of computation of machine learning algorithms and they can improve results quality, however some scientist believe they bias the data. The clustering techniques will be evaluated with and without the pre-processing methods.

### 2.2.1 Pre-processing

Gariel et al. [27] suggest applying PCA for an *extended data matrix*, with more information than the original, before clustering, claiming that it gives better results. Gallego et al. and Eckstein et al. [28, 29] have also applied a pre-processing tool before applying the clustering. Nevertheless, Enriquez [30] says there is no real gain in the results quality by applying these techniques, in fact it may even present worse results by eliminating some of the information implied by the reduction of the state-space. Meanly, t-SNE (t-distributed stochastic neighbour embedding) [31] a more recent method overcomes some of the limitations delivered by PCA, especially the non-linear state-space. Another, pre-processing method presented by Almeida et al. [32] is the removal of outliers before running the clustering method.

The following paragraphs discuss some of these pre-processing methods.

#### Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimensionality reduction method applied to a state-space, while preserving as much data variance as possible [33, 34]. These main characteristics may be the relationship of one or more of the original dimensions, creating the principal components.

Usually, it is applied in application domains that deal with high dimensional data, such as genetics, biology, artificial intelligence techniques (such as facial recognition), computer vision and image compression. It may also be used to find patterns in finance and banking, data mining, psychology, etc.

PCA has its foundations from 1900. Its goal is to preserve as much of the dataset variation as possible. The first component is the direction that maximizes the data's variation as much as possible. The second is the second direction maximizing the data variance, and so on.

PCA can be computed by:

1. Data covariance matrix eigen-decomposition
2. Singular Value Decomposition (SVD) [35]

One of the limitations of PCA is that it is a linear dimension reduction tool, it does not work well on non-linear data, because it tries to preserve the global structure of the data. PCA objective is to maximize variance, which involves minimizing the squared error between distances in the original data and distances in the reduced data. The square error in large pairwise distances is more affected than for small distances between points. Taking Figure 2.7 for instance, PCA would focus on preserving large pairwise distances and incorrectly perform data reduction.

#### t-SNE

The t-distributed stochastic neighbour embedding (t-SNE) method searches for local similarities and does not bother in keeping distances of far apart points in the reduced state-space. Its main aim is to minimize the divergence between two distributions: a distribution that measures the pairwise similarities of the original data ($\mathrm{p}_{ij}$) and a distribution that determines the pairwise similarities of the corresponding low-dimensional space ($\mathrm{q}_{ij}$) [36].

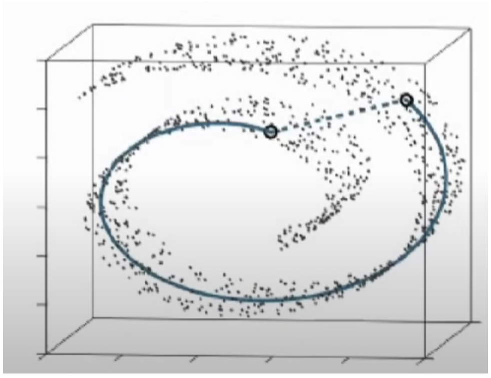


Figure 2.7: Non-linear Manifold - Swiss Roll

$$p_{ij} = \frac{\exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-\left\|\mathbf{x}_k - \mathbf{x}_l\right\|^2 / 2\sigma^2\right)} \tag{2.2}$$

Considering X={x$_1$, ..., x$_N$} are the set of input objects, the p$_{ij}$ term in Equation 2.2 represents the probability distribution from point i to j, such that j $\neq$ i, considering a Gaussian distribution around i in the numerator. The denominator represents a normalization, by averaging all the distributions around all points.

In Fig. 2.8 depicts an illustration example, where the red point represents point $i$, while the other blue points represent $j$ points. A Gaussian is defined around point i and the probability $p_{ij}$ is calculated as in Eq. 2.2. The same applies for all the other points, until all the probability distributions are computed.

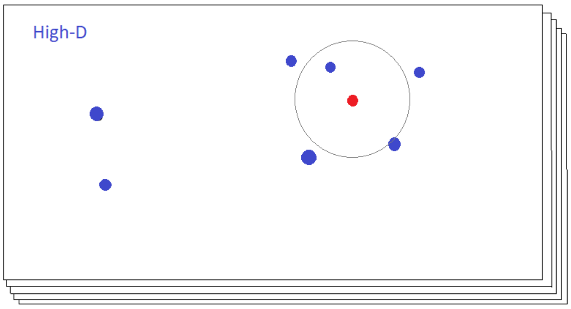


Figure 2.8: Simple, visual representation of an original dataset

If the distance between $i$ and $j$ is very small, p$_{ij}$ is high. On the other hand if the $i$ and $j$ are quite far apart p$_{ij}$ is infinitesimal.

In the resulting lower dimension, the distribution q$_{ij}$ is measured as in Eq. 2.3, where $\varepsilon$ = {y$_1$, y$_2$, ..., y$_N$} is the resulting lower dimension data.

$$q_{ij} = \frac{\left(1 + \left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2\right)^{-1}}{\sum_k \sum_{k \neq l} \left(1 + \left\|\mathbf{y}_k - \mathbf{y}_l\right\|^2\right)^{-1}} \tag{2.3}$$

q$_{ij}$ is not a Gaussian Distribution but a Student-t distribution, which is more heavy tailed. The reason of using this distribution is that t-SNE focus on modelling local similarity, allowing dissimilar points to be

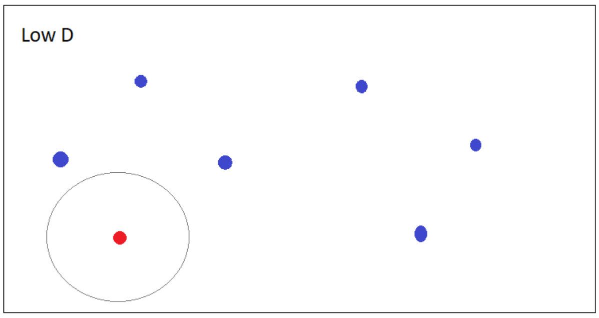


Figure 2.9: reduced dimension space

modelled further than in the original data.

Since the aim is to have a lower dimension space as similar as possible to the input objects, the algorithm searches for the minimization of Kullback-Leibler (KL) divergence between P and Q is applied, Eq. 2.4.

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.4}$$

Table 2.4 summarizes the benefits and drawbacks of the two techniques.

Table 2.4: Summary of the advantages and disadvantages of PCA an t-SNE

| **Algorithm** | **Advantage** | **Disadvantage** |
|---|---|---|
| PCA [33, 34] | Created in the beginning of twentieth century it is still the most well-known algorithm for dimensionality- -reduction, it is based on the identification principal components which may be a combination of two or more state-spaces of the original data. | Based on the squared error which is greater for larger distances than for smaller, this makes PCA unsuitable for non-linear state-spaces. |
| t-SNE [31, 36] | As t-SNE bases on the minimization of Kullback Leibler (KL) divergence between $p$ (original) and $q$ (reduced dataset) only the large $p$ interfere, those large $p$ only take into account neighbouring datasets, overcoming the limitation of PCA. | slightly more complex computation and more recent (2008). |

### Outlier Removal

Suggested by Almeida and Segarra [32, 37], outlier removal steps before calling the clustering method promises to provide great enhancement. According to [32], outliers are depicted as isolated objects or small groups of objects located in low density zones, contrasting with the denser intra-cluster structure. In other words, outliers are regarded as objects with low connectivity in opposition to higher connectivity in the intra-cluster region.

The convergence process to remove the outlier is to split the dataset into two. At the first one, the radius of the target is taken as $4\bar{d}_j$, where $\bar{d}_j$ is the average nearest-neighbor distance previous to iteration j. In the second, a smaller multiplier is used and the radius is $2\bar{d}_j$. In both processes, points with a connectivity lower than 1/3 of the average value for connectivity, $\bar{c_j}$, are discarded in each iteration. The value of $\bar{d_j}$ is recalculated and the process repeated, until the number of discarded objects is zero. The identification of outliers, based on [32] may be summarised as follows:

```
• for k in {2, 4}:
    – Set iteration counter j=1
    – Repeat until the number of discarded objects = 0
        ∗ Calculate $\bar{d_j}$
        ∗ Set R =k$\bar{d_j}$
        ∗ Calculate $\bar{c_j}(R)$
        ∗ Discard objects i if $c_i$ < P$\bar{c_j}$
        ∗ Increase j
```

According to [32], the first iteration, $4\bar{d_j}$, has the objective to remove both scattered objects or small groups and thin bridges of noise linking ‘natural clusters’. The second step is more sensitive to outliers present in regions close to the boundaries of the clusters, overcoming the limits of the first step.

Almeida et al. [32] decided that P=1/3, after performing a battery of test, identifying this value as a good coefficient to attain a degree of inter-cluster that facilitates clustering.

Segarra [37] applied this approach to air traffic data an decided P=0.03, because high P value only behaved well in the presence of dense clusters of similar density. In air traffic application, the choice of a low P value is necessary in order not to eliminate relevant information when there are clusters of very different density.

### 2.2.2 Density Based Spacial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a density-based clustering mechanism is a density base technique with two mandatory hyper-parameters ($Min\_trajectories$, $epsilon$). $epsilon$ is a real value and corresponds to the maximum distance between neighbouring trajectories to be considered in the same cluster. First, the algorithm starts by checking a random trajectory, when one or more trajectory is at distance $\leq epsilon$ they are considered in the same set. Then, it takes the new added trajectories and checks if distance is $\leq epsilon$ to any other outside trajectory and this search until no trajectories are found anymore. At this stage, the algorithm picks randomly another trajectory from the ones that have not yet been analysed and performs the same search until the algorithm has passed through all trajectories.

$Min\_trajectories$ is a integer and represents the minimum number of trajectories considered inside a group for the group to be taken as a cluster, otherwise the group is not a cluster and these trajectories are assigned as anomalies.

### 2.2.3 Hierarchical Clustering

Since the algorithms to which a greater emphasis is given throughout the whole thesis are the hierarchical clusterings, a whole section was dedicated to explain succinctly the methods which are going to be tested. Those methods are R-DBSCAN (Recursive Density Based Spacial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify Clustering Structure), HDBSCAN (hierarchical DBSCAN) and Hierarchical Completely Automated Method. Then tough hierarchical clusterings are more complex than the partitioning methods, they do not need to be told into how many clusters the data must be split and can also identify outliers.

### R-DBSCAN

R-DBSCAN was proposed in [28] by Gallego et al.. Meanwhile, Thomas Dubot developed an algorithm [38], in order to determine the occupancy peak estimations using mostly sector geometry and traffic data [39].

A schematic interpretation at Figure 2.10 shows the development of the first cluster at the main branch, it gives a very good intuition of how things work at Recursive-DBSCAN.

R-DBSCAN is based on DBSCAN but with a dynamic epsilon, demanding only one hyper-parameter the $Min\_trajectories$. It is based on two step:

- compute $epsilon$;
- run DBSCAN ($epsilon$, $Min\_trajectories$).

The dynamic epsilon is computed according to the dataset density and is computed each time iteratively, the first time is with the whole dataset, then the iteration is performed over smaller datasets, that in the previous iteration were regarded as one cluster, to try the segment is smaller clusters instead of one big cluster. To get the $epsilon$ two steps are performed:

- find the minimum radius for all trajectories in order the encompass the nearest $Min\_trajectories$ trajectories;
- find the (x, y) of maximum curvature from all the trajectories considered, where x is the index representing the trajectory and y is the distance given by the first step (see Figure 2.11). The point_of_maximum_curvature$_{epsilonKNN}$ is the $y$ given at the intersection of the blue line with the dashed black line.

$$epsilon = eps\_factor \cdot \lambda^{n\_interaction-1} \cdot point\_of\_maximum\_curvature_{epsilonKNN} \quad (2.5)$$

At the algorithm, $Min\_trajectories$= 10 which coresponds to 1% of the total dataset, $eps\_factor$ = 1 and $\lambda$ = 0.9, the $point\_of\_maximum\_curvature$ differs from iteration, is obtained with [40]. The value is never bellow or equal to zero, at this experience the greatest value was 2.82 and the lowest 0.26.

Figure 2.11 is a graphical representation of the minimum radius (y-axis), ordered by descending order, from each trajectory (x-axis) such that it groups the 10 nearest trajectories around, at the first iteration. The first iteration takes all the trajectories in account so the x-axis goes from 0 to 1244, the total number of trajectories in the original dataset. The other iterations are focused into a cluster given from previous iteration to segment the clusters in smaller and more precise ones, taking into account less trajectories in the x-axis.

After having the clusters given at first iteration, it computes the new $epsilon$ with the data given at

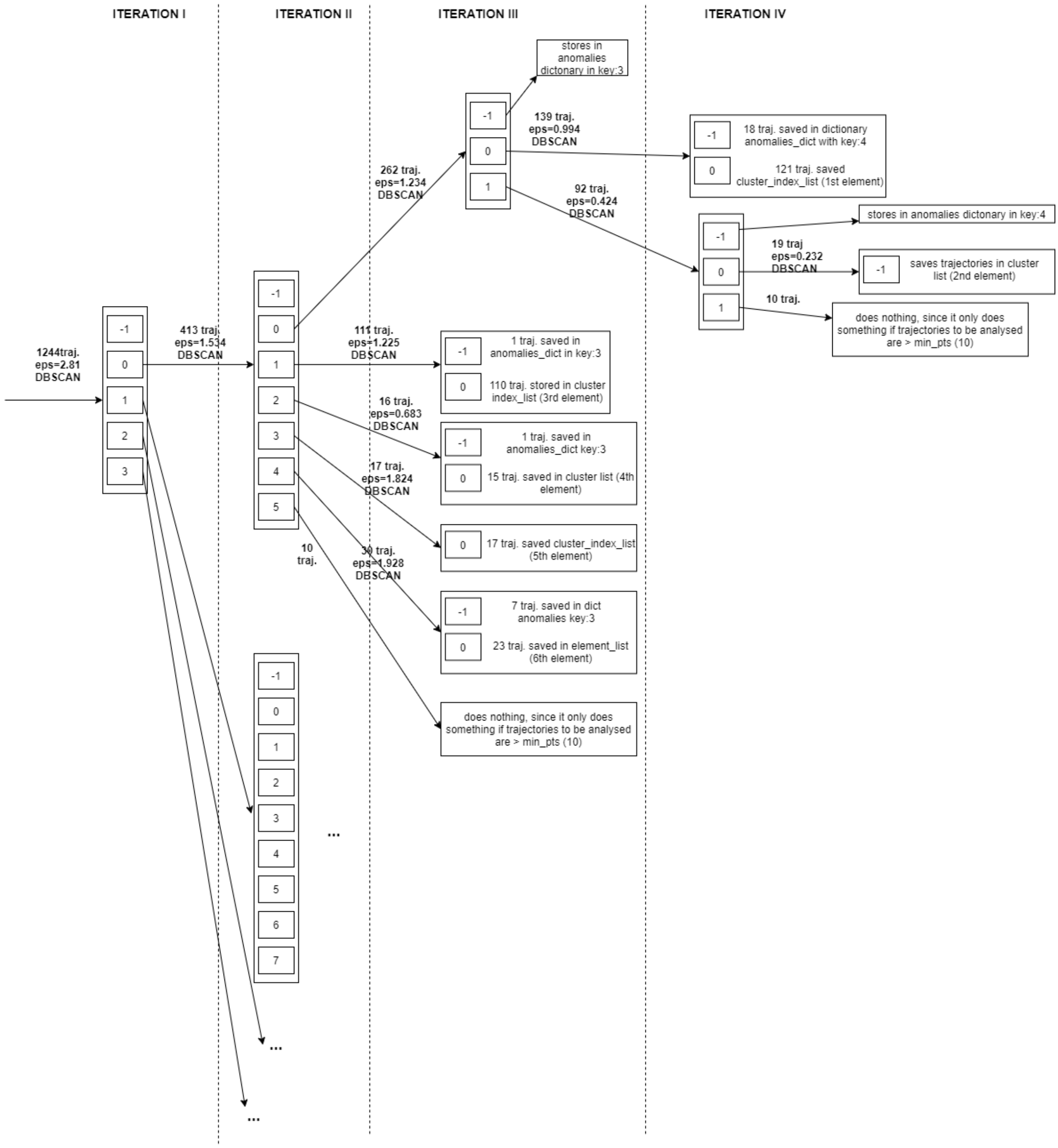

ITERATION I
ITERATION II
ITERATION III
ITERATION IV
1244traj.
eps=2.81
DBSCAN
-1
0
1
2
3
413 traj.
eps=1.534
DBSCAN
-1
0
1
2
3
4
5
262 traj.
eps=1.234
DBSCAN
-1
0
1
stores in anomalies dictonary in key:3
139 traj.
eps=0.994
DBSCAN
-1
18 traj. saved in dictionary anomalies_dict with key:4
0
121 traj. saved cluster_index_list (1st element)
92 traj.
eps=0.424
DBSCAN
-1
0
1
stores in anomalies dictonary in key:4
19 traj
eps=0.232
DBSCAN
-1
saves trajectories in cluster list (2nd element)
10 traj.
does nothing, since it only does something if trajectories to be analysed are > min_pts (10)
111 traj.
eps=1.225
DBSCAN
-1
1 traj. saved in anomalies_dict in key:3
0
110 traj. stored in cluster index_list (3rd element)
16 traj.
eps=0.683
DBSCAN
-1
1 traj. saved in anomalies_dict key:3
0
15 traj. saved in cluster list (4th element)
17 traj.
eps=1.824
DBSCAN
0
17 traj. saved cluster_index_list (5th element)
10
traj.
30 traj.
eps=1.928
DBSCAN
-1
7 traj. saved in dict anomalies key:3
0
23 traj. saved in element_list (6th element)
does nothing, since it only does something if trajectories to be analysed are > min_pts (10)
-1
0
1
2
3
4
5
6
7
...
...
...

Figure 2.10: Schematic view of R-DBSCAN iterative process only at the first cluster identified in iteration I, there are still other three cluster to extend.

first cluster and then calls DBSCAN method. In case more than one cluster resulted from DBSCAN, it performs the R-DBSCAN again for each resulting cluster, otherwise the algorithm stops.

The algorithm stops in each branch, in the following conditions: reached the maximum number of iteration; the algorithm found one cluster from previous one; there are 10 trajectories in the cluster's set.

The characterizing output of each cluster is its centroid, which is the flight chosen to be the best describing trajectory between the set of flights assign to the cluster.

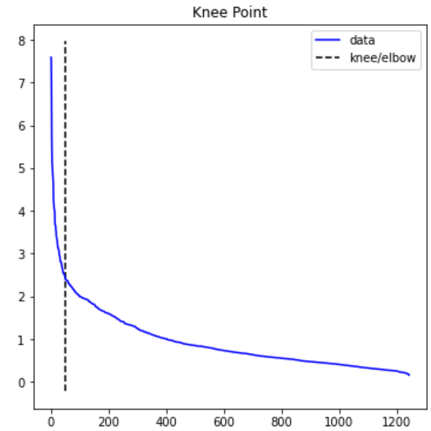


Figure 2.11: Graphical representation of the knee-point search at the $1^{st}$ iteration

### OPTICS

The OPTICS [41–43] algorithm only demands one hyper-parameter $Min\_trajectories$, is based on the idea the higher density objects should be processed first in order to find a high density cluster first. OPTICS is based on two fundamental concepts: *core distance* and *reachability distance*. The *core distance* of object $p$ is the smallest value $epsilon$ such that the neighborhood of $p$ has at least $Min\_trajectories$. The core distance can also be undefined in case the $epsilon$ needs to be very large. If an object has a core distance than it is also a core-object. The reachability distance between object $q$ and a core-object $p$ is Max(dist($p$,$q$), core_dist($p$)) or undefined if $p$ is not a core object.

OPTICS builds a reachability plot, such that the objects are organized in the x-axis such that neighbouring objects at the original dataset are still side-by-side on the reachability plot and the y-axis represents the reachability distance of each object. Objects belong to a cluster have low reachability distance to their nearest neighbours, valleys in reachability plots correspond to clusters and the deeper the valley the denser the cluster is. At Figure 2.12 is a reachability plot where three clusters can be identified, an horizontal line may be drawn to define the reachability distance from which higher reachability distance define anomalies and lower define the objects that belong to the clusters.

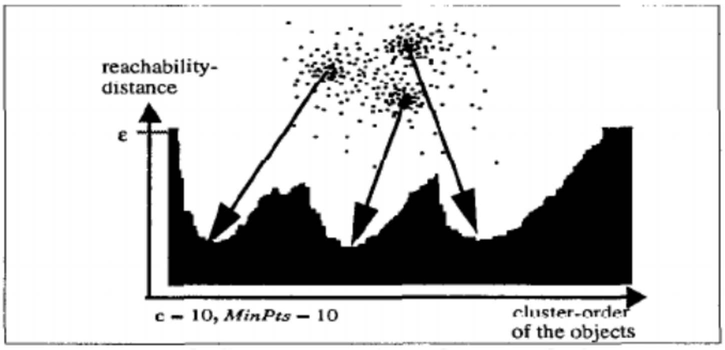


Figure 2.12: illustration of reachability plot [41]

### HDBSCAN

Hierarchical Density Based Spacial Clustering of Applications with Noise (HDBSCAN) [42–46] finds clusters over data of varying densities. HDBSCAN can be explained in four steps:

1. *Mutual Reachability Distance* to introduce robustness to noise, an attempt to increase distance between anomalies and main-data is applied:

$$d_{mreach_k}(a,b) = max\{core_k(a), core_k(b), d(a,b)\} \tag{2.6}$$

   $k$ is an integer and corresponds to the minimum number of trajectories need for a group of objects to be called a cluster. $core_k(a)$ means the radius of a cylinder depicted around trajectory $a$ in order to have $k$ trajectories inside cluster centered at $a$; $core_k(b)$ is the same intuition applied to trajectory $b$, and $d(a,b)$ is the distance between these 2 core trajectories. Eq. 2.6 means, in case the distance between two trajectories is less then the radius of one(or both) of the two trajectories, the mutual distance between the two would not be d(a, b), as the intuition says, but it would increase to the greatest core distance;

2. *Minimum Spanning Tree* formulates a path joining the data such that the path formulated has minimum sum of edge weights over all possible paths;

3. *Cluster Hierarchical Graph* splitting the most distance sets at the $1^{st}$ level deep and splitting again at the $2^{nd}$ level to create other leafs from the parent.

4. *Condensed Cluster Tree* once there is a minimum cluster size it is possible to go through the hierarchy and at each split see if one of the new clusters has fewer trajectories than the minimum cluster size. If so, these clusters will fall out of the tree and be addressed as anomalies and the larger cluster will retain the cluster identity of the parent. If on the contrary the split is into two valid clusters then the search continues over both branches that split from the parent node and persist until only one valid cluster is found. The search ends up after walking through the whole hierarchical graph;

5. *Extract the Clusters* if a cluster is selected, then it is not possible to select any of his descendant clusters. Short lived clusters are probably mere artefacts of linkage.If the sum of stabilities from the child clusters is greater than the stability of the parent cluster, than the search continues over the branch, until the opposite observes and the cluster is the parent and not the descendants.

Figure 2.13 is a graphical representation of what happens at steps 2-4 of HDBSCAN.

### Hierarchical completely automated method

The hierarchical methods mentioned before have the particularity of requiring the defenition of *MinClstSize*. Almeida et al. [32] propose a fully automated hierarchical method instead. In fact, most of the hierarchical clustering methods analyse the data in a dendogram or tree, and they are pruned such as explained in point *4.* of HDBSCAN.

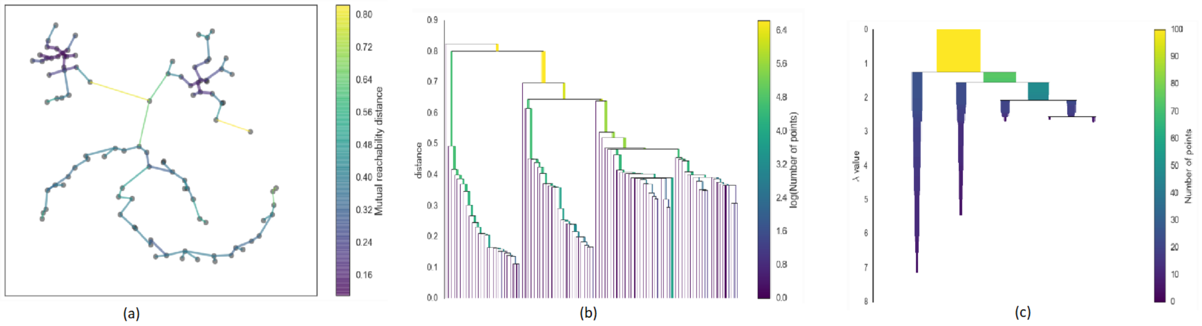


Figure 2.13: Step 2 - 4 HDBSCAN (a) *Minimum Spanning Tree*, (b) *Cluster Hierarchical Graph* and (c) *Condensed Cluster Tree* [45]

The descriptive function (DF) for a pair of consecutive objects i, i + 1 corresponds to the squared minimal distance measure of all linkage steps in which both objects participate (or cophenetic distance):

$$DF_{i,i+1} = t^2_{i,i+1} \tag{2.7}$$

Almeida et al. [32] suggest the dendogram to be pruned at:

$$DF_{sef} = Y \times (Q_3 - Q1) \tag{2.8}$$

where $Q_1$ and $Q_3$ are the upper limits of the first and third quartile of the distribution of values in the descriptive function. Y was assign to 6 in [32]. However its application is in Chemistry. Segarra [37] decided to perform a set of tests over air traffic dataset, changing coefficient Y at equation 2.8 and evaluate it with the average Silhouette value:

$$\bar{S}_N = \frac{1}{N_r}\sum_{1}^{N_r} S(i) \tag{2.9}$$

Where S(i) equals equation 2.10.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.10}$$

This method is suggested after performing the outlier removal, in order to avoid having the outliers compromising the results.

### 2.2.4 Synthesis of some clustering methods

Table 2.5 summarizes the advantages and disadvantages of the clustering methods to which greater focus was given in the pages before.

Table 2.5: Comparison of density based clustering methods.

| Clustering Method | Main Principle | Advantages | Disadvantages |
|---|---|---|---|
| DBSCAN [47, 48] | Based on two hyper-parameters (epsilon, MinClustSize) | Generic, does not demand any previous knowledge allowing it to be applied in any airspace. Detects outliers. | Does not adapt to the data since it has a fixed epsilon. |
| R-DBSCAN [28, 38] | Iterative search, computing the epsilon by K-NN before calling DBSCAN. Only demands MinClusterSize. | Generic, adapts to the dataset (low densiy, high density), detects outliers. | Computational Time. |
| HDBSCAN [28, 45] | Based on the reachability plot (neighbouring trajectories placed side by side in 2D plot vs reachability distance). Demands hyper-parameter (MinClusterSize) | Outlier detection. Generic. | Computational Time. |
| OPTICS [28, 41, 43] | Based on the condensed cluster Tree. Demands (MinClustSize) | Outlier detection. Generic. | Computational Time. |
| Automatic Hierarchical clustering [32, 37] | Dendogram build up and a descriptive function that defines a dynamic pruning | No need to provide MinClustSize | Suggests an outlier removal method before clustering |

### 2.2.5 Evaluation Parameters

Gallego et al. [28] tackle the topic - density clustering validation - to compare different clustering methods.

They believe that clustering validation at an external point of view should combine verification against planned trajectories and frequent patterns due to ATC actions, but so was not performed in their research.

Two techniques are presented to evaluate and compare methods - Silhouette Width Criterion (SWC) and Density Based Clustering Validation (DBCV) metric, based on [49].

Silhouette Width Criterion (SWC), proposed by Rousseeuw [50], compares the ratio of intra- and inter-cluster, to evaluate compactness and separation between them. It works well with globular clusters, but its performance decreases when dealing with clusters with varying shapes.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.11}$$

a(i) is the average intra-cluster distance of point i to the other points in the same cluster and b(i) is the mean inter-cluster distance from i to the other points that belong to the nearest cluster. When the

Silhouette value tends to 1, the clustering algorithm has good results, i.e., the inter-cluster distance is much larger the the intra-cluster distance. Otherwise, if Silhouette value is close to -1, the clustering has bad results.

Density Based Clustering Validation (DBCV) is more appropriate to validate clusters with arbitrary shapes and varying densities. This metric mirrors the notion of Compactness versus Separation . DBCV is defined in terms of Density Sparseness of a Cluster (DSC) and Density Separation of a Pair of Clusters (DSPC). It is given by:

$$DBCV(C) = \sum_{i=1}^{i=l} \frac{|C_i|}{|O|} V_C\left(C_i\right) \tag{2.12}$$

where $|O|$ is the total number of objects under evaluation, $|C_i|$ takes into account the size of the clusters and V$_C$(C$_i$) is given by:

$$V_C\left(C_i\right) = \frac{\min_{1 \leq j \leq l, j \neq i}\left(DSPC\left(C_i, C_j\right)\right) - DSC\left(C_i\right)}{\max\left(\min_{1 \leq j \leq l, j \neq i}\left(\mathrm{DSPC}\left(C_i, C_j\right)\right), DSC\left(C_i\right)\right)} \tag{2.13}$$

In this chapter a thorough study and descriptive analysis of complexity studies and definitions of ATCO workload as well as clustering methods, with special focus on hierarchical clustering methods, was performed. The Hierarchical clustering mechanisms demand less hyper-parameters than other clustering techniques, and may apply in any ACC, without asking for additional information. The clustering study ends describing some evaluation techniques such as Silhouette Width Criterion (SWC) and Density Based Clustering Validation (DBCV). Relatively to Air Traffic Complexity, it is very clear that air traffic complexity cannot be described without considering ATCOs. After analysing many studies and approaches, it was settled that building a simulator and asking controllers to perform their subjective evaluation was beyond the scope of this thesis, besides it had already been implemented and studied by many researchers and had its limitations [19]. At the end, it was decided to create an algorithm that could compute the CAPAN tasks autonomously given traffic data, ACC and sector information.

In next chapter, Proposed Approach, an evaluation method based on quality indicators, inherent to air traffic, is presented. Considering Complexity Estimation, the interpretation and autonomous computation of the CAPAN tasks, is explained in detail with the help of flowcharts, finalizing with the explanation of the working fundamentals inherent to the optimal sector configuration selector.

# Chapter 3

# Proposed System

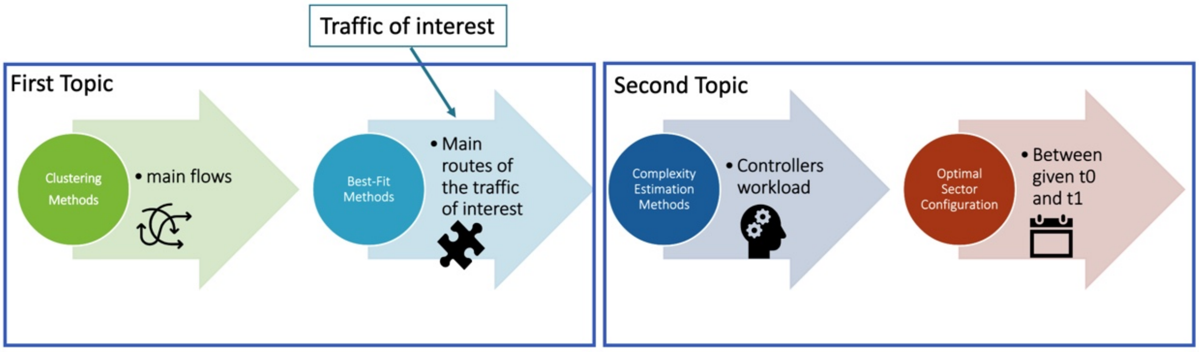

Traffic of interest
First Topic
Clustering Methods
• main flows
Best-Fit Methods
• Main routes of the traffic of interest
Second Topic
Complexity Estimation Methods
• Controllers workload
Optimal Sector Configuration
• Between given t0 and t1

Figure 3.1: Proposed System's main phases.

Figure 3.1 presents the four main phases of the proposed system, considering two major topics.

- The first topic ('Clustering method', 'Best-fit methods'), is about the identification of typical and atypical routes using machine learning algorithms. This tool will help the FMP deciding the best sector configurations by visualizing the most common routes that are going to be happen in the day ahead and by anticipating the anomalies (the atypical routes) that might cause problems;

- The second topic ('Complexity workload', 'Optimal sector configuration'), uses the CAPAN parameters to define the controller workload in a given sector configuration. This method involves computing 38 tasks that may me considered by the controller, depending on the type of entry, route inside volume and exit. This will help the FMP to identify the hotspots. An application of this topic is proposed, using the workload estimation to find the optimal sector configuration between a given time interval $[t_0, t_1]$.

The traffic already assigned to a cluster (or considered atypical) provides the input to the *Complexity Estimation* algorithm. Hence, if an entering aircraft belongs to a new cluster (or is atypical), an additional complexity workload needs to be added.

## 3.1 Main-flows determination - Clustering

Main-flows are a fundamental probabilistic indicator related to Free Route Airspace (FRA) implementation at European Sky.

The creation of a new clustering technique was not in the scope of this analysis. Instead, the objective is to provide validation and enhancement, by defining quality indicators to allow the comparison of different clustering methods. Hence, this approach can be described as a combination between machine learning and deterministic benchmarking. Figure 3.2 refers to the main steps of the clustering algorithm.

Some of the most mentioned articles about air traffic clustering techniques [27, 28, 38, 44–46] are tested, with the objective to qualify their outcomes. Also included in this study is the influence of pre-processing techniques.

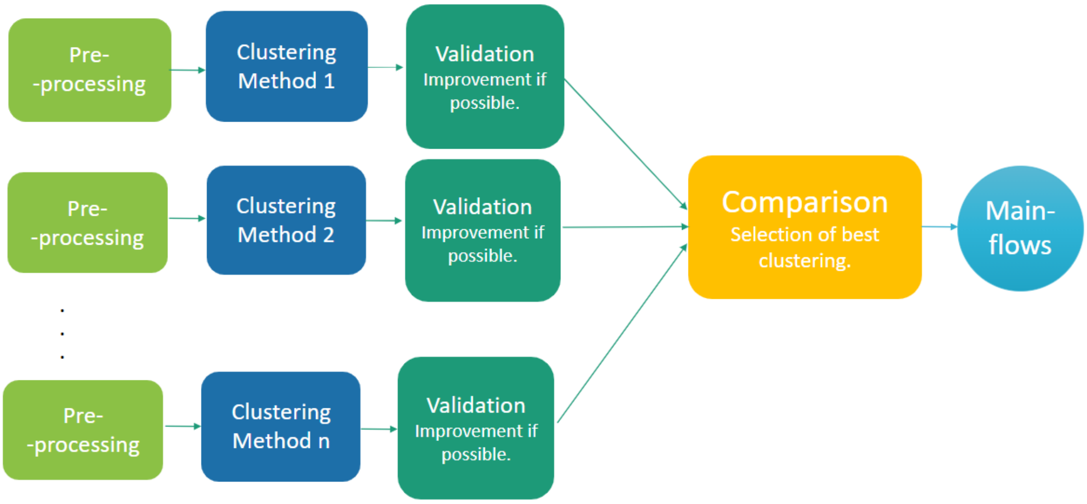


Figure 3.2: Clustering Implementation

### 3.1.1 Pre-processing

A clustering method may be pre-processed to give a faster answer or improve the final results. Here two algorithms are tested, the PCA and the t-SNE.

The PCA pre-processing accepts as input the number of dimensions provided by the operator. it will be tested in 3 different ways: PCA with number of components equal to 2, 4 and 6. The used t-SNE [51] algorithm is only based on 2 dimensions.

Figure 3.3 provides a visual summary of this section.

### 3.1.2 Clustering Methods

The methods selected to evaluate and compare are DBSCAN, R-DBSCAN, R-DBSCAN* (which is a small change of R-DBSCAN), HDBSCAN* and OPTICS. Except for DBSCAN, which was only selected to be able to compare with R-DBSCAN, the selected methods are all hierarchical and demand very little

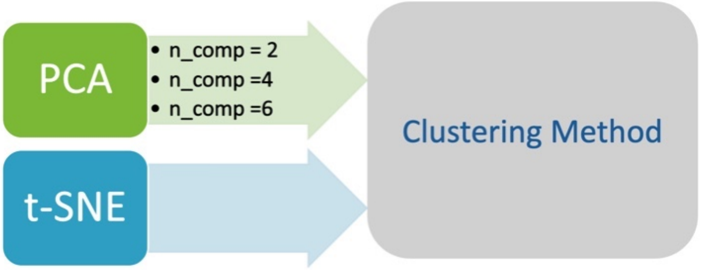


Figure 3.3: The four pre-processing tools to be integrated on testing

knowledge of the airspace and traffic volume. This may be very advantageous because it gives the characteristic of generality, i.e., being able to be applied on any airspace.

### 3.1.3 Validation Method

Two types of validation are defined:

- *intra-cluster validation* - regarding the evaluation and a quality measure indicator inside a given cluster.
- *inter-cluster validation* - regarding the validation of clusters between each other, the goal is to evaluate if there are clusters which similar to another.

Hence, the intra-cluster validation tests if the majority of the trajectories are similar to the cluster's centroid they were assigned to. The inter-cluster tests if the majority of a centroid's cluster are distinct from the other clusters' centroids.

Each validation type uses two evaluation measures:

- **Lateral evaluation** - for lateral evaluation, trajectories and centroids were first resampled by *resample_number*.
  - 2D lateral distance similarity
    - Only the 2D lateral distance is taken, because flights may be similar but distant in altitude because they are in different FL. However, they still belong to the same main-flow. At least *percentage_samples_dist_similar* of the total trajectory's samples, must verify lateral distance between the trajectory's i$^{th}$ sample and the centroid's i$^{th}$ sample $\leq$ *lateral_threshold*, for the trajectory to be considered 2D lateral distance similar with the centroid, this in case of testing intra-cluster validity. During the inter-cluster validity testing, it would be between two centroids.
  - heading similarity
    - Applies the same logic explained in 2D lateral distance evaluation, except that now it is the heading difference between sample $i^{th}$ of one trajectory and sample $i^{th}$ of the other. Samples verifying bellow or equal *heading_threshold* must be $\geq$ *percentage_samples_head_similar* for heading similarity to exist.

At Figure 3.4 are two example of lateral evaluation between a trajectory and a centroid. In the example on the left the trajectory is always bellow 20NM, at all samples (only some samples are at the Figure represented). In the heading difference part of the flight is very different from the centroid, but if the percentage of headings bellow *heading_threshold* is greater or equal to *percentage_samples_head_similar* the flight is still heading similar. At the right example, there is a period during which the trajectory is further than the *lateral_threshold* if the number of samples in this phase are more than (1 - *percentage_samples_dist_similar*) the trajectory is not 2D lateral distance similar with the centroid and that is enough for the trajectory not to be considered as contributing to the cluster intra-validity.

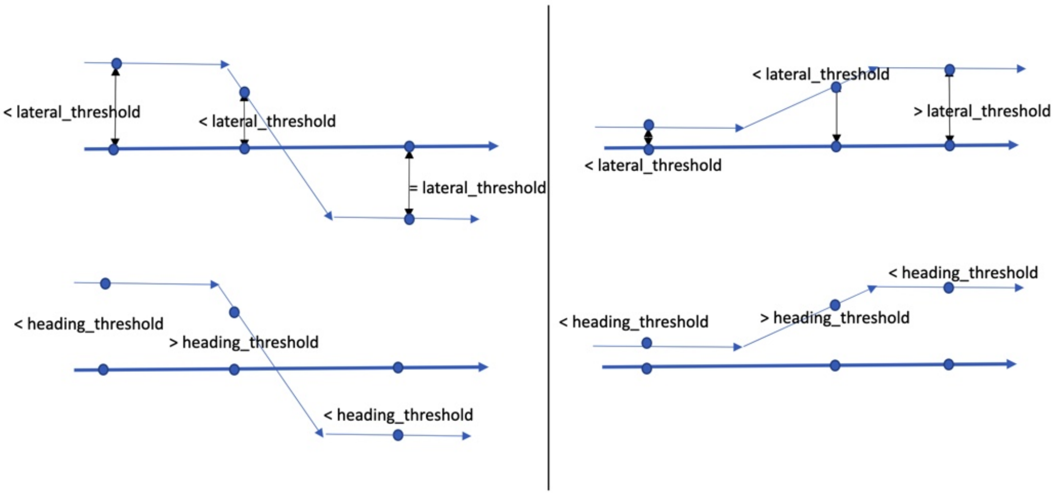


Figure 3.4: Two examples of measuring 2D lateral distance similarity and heading similarity between one trajectory (narrower line) and one centroid.

- **Vertical evaluation**
  - Vertical evolution similarity
    - A pre-processing of the data is performed according to Equation 3.1, to define the flight's phase of the trajectories being analysed. The values are then compared and if the $i^{th}$ sample of one trajectory is equal to the $i^{th}$ of the other trajectory, it is added to the percentage of samples with similar vertical evolution. If the percentage of with similar samples vertical evolution is $\geq$ *percentage_samples_vertical_evolution_similar*, they are considered vertical evolution similar.

$$\text{Vertical Evolution} = \begin{cases} 1, & \text{inside a minute interval, altitude different} \geq \textit{vertical\_evolution\_threshold} \\ 0, & \text{inside a minute interval, } (-\textit{vertical\_evolution\_threshold}) < \text{altitude different} \\ & < \textit{vertical\_evolution\_threshold} \\ (-1), & \text{inside a minute interval, altitude different} \leq (-\textit{vertical\_evolution\_threshold}) \end{cases} \tag{3.1}$$

When evaluating the intra-cluster, the comparison is made between one trajectory and the corresponding cluster centroid. On inter-cluster, the comparison is done between two centroids (of different clusters).

For a cluster to be considered intra-cluster valid, the *percentage_trajectories_to_be_considered_valid* of trajectories must verify the 2D lateral distance similarity. The same applies to the heading similarity and also to the vertical evolution similarity.

For inter-cluster, only if two cluster centroid verify distance and heading similarity, is the vertical evolution verified. Bearing in mind that at the inter-cluster the desired is that the cluster are different, it is enough that one of the three constraints does not verify.

For the sake of simplification, from this page on, a valid cluster represents a cluster that has intra-cluster validity. Table 3.1 presents all validity constants that were considered in this thesis.

With these two main constraints an *applicability value* is assigned to the trajectory, such that it characterizes the type of validity. If it is vertical and lateral valid, value ”2” is assign to it; otherwise, if it is only lateral valid value ”1” is assigned to the trajectory. Otherwise, it is enough that the lateral validity is not verified and the applicability value is ”-1”.

Table 3.1: Constants for evaluation and validation methods, value assignment

| **Constant** | **Meaning** |
|---|---|
| *resample_number* | Value to which the flight's trajectory is resampled. [27] |
| *lateral_threshold* | 2D distance above which the sample is not considered inside percentage of sample lateral distance similar. [52]. 20 NM is the surveillance range defined by TCAS. |
| *percentage_samples_dist _similar* | Percentage of samples that need to verify a 2D lateral distance equal or below 20NM, otherwise the trajectory will not be taken as lateral valid. |
| *heading_threshold* | Heading difference above which the sample will not be considered inside the percentage of samples heading similar. |
| *percentage_samples_head _similar* | Percentage of samples needed to verify heading difference below or equal to *heading_threshold*, otherwise the trajectory will not be taken as lateral valid ( a good characteristic in inter-cluster, a bad characteristic in intra-cluster). |
| *percentage_samples_vertical _evolution_similar* | Percentage of samples needed to verify vertical evolution similarity, otherwise the trajectory is not vertical valid. |
| *percentage_trajectories_to_be _considered_valid* | Percentage of trajectories that were assign to the cluster at the Machine Learning algorithm that need verify lateral and vertical evolution such that the cluster is *intra-* valid. |
| *vertical_evolution_threshold* | EUROCONTROL defined that a flight that is changing altitude by 500ft/min is a flight in climb or descent. [10] |

Figure 3.5 provides a very simple and hypothetical example, where it is possible to distinct two different clusters and the trajectories associated to them. The intra-cluster would compare the trajectories with the respective centroid and define the cluster as valid or not. The inter-cluster study would conclude that both centroids of the two clusters are very distinct: it is impossible that their samples prove a 2D lateral distance similarity.

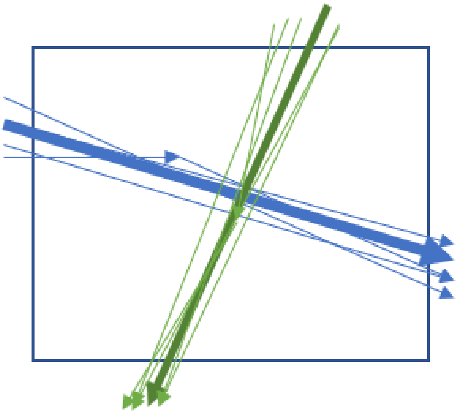


Figure 3.5: Very simple hypothetical example of two clusters in an airspace volume seen from above

### 3.1.4 Similar clusters merge - iterative process

For pairs of clusters which tested to be very similar, intra-cluster validity is tested for an hypothetical merged cluster. A new centroid must be defined and the three similarities defined in section 3.1.3 must be re-computed. Hence, when the intra-cluster validity results in a valid cluster, the merge is accepted and inserted in the original data.

However, the insertion of the merged cluster addresses some consequences. The similar clusters pair is saved in a tuple $(c_i, c_j)$. In case the merge is accepted, the cluster $c_j$ identified by a unique number in the trajectories assigned to it, is replaced by the cluster number $c_i$. The centroid corresponding to $c_j$ is also eliminated. Moreover, it must be checked if the cluster $c_j$ is paired with another cluster in the list of similar pairs. If so, the list must be re-built.

To re-build the list, the inter-cluster validation must be run again, because the new cluster that got assigned to $c_i$ may pair with other clusters.

Figure 3.6 illustrates an hypothetical example of four clusters, where two (the blue and the yellow) are very similar. The possibility of a merge is tested and accepted (the created merge is a valid cluster) which resulted in a new cluster, with a new centroid.

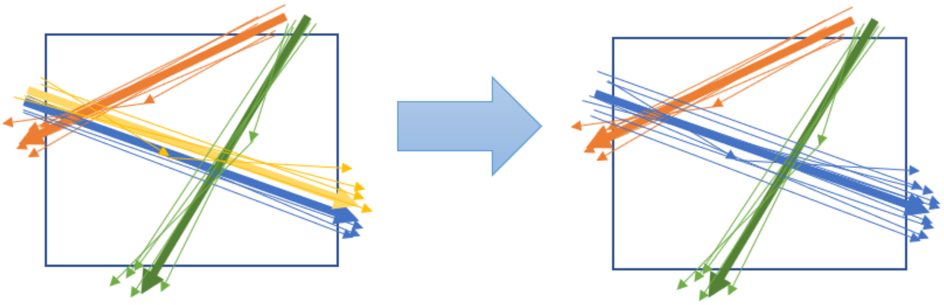


Figure 3.6: Hypothetical example a rectangular ACC with four cluster and the result of a merge.

### 3.1.5 Quality Indicators

A number of parameters were selected to qualitatively and quantitatively evaluate each clustering method:

1. *number of clusters* (NC) - each technique identifies a different number of clusters;
2. *trajectory assigned to clusters* (TAC) - ratio of flights that were labelled as non-outliers;
3. *trajectories assigned to valid cluster* (TAVC) - number of trajectories which belong to valid clusters (those who are *intra-* valid);
4. *clusters intra- valid* (CIV) - fraction of cluster which verified intra- valid condition from the total;
5. *Mean flights per valid cluster* (MF_VC) - mean of the number of flights per valid cluster;
6. *standard deviation number of flights over valid cluster* (SDF_VC) - the disparity of the mean values between valid clusters;
7. *inter-clusters valid* (PIV) - ratio of clusters which do not have other cluster similar to it.
8. *mean vertical similarity over valid clusters(%)* (MVS_VC) - mean of vertical validity over valid clusters.
9. *mean lateral similarity over valid clusters (%)* (MLS_VC) - mean of lateral similarity (both 2D lateral distance and heading difference) over valid clusters.
10. *mean of vertical validity inside non-valid clusters (%)* (MVS_NVC) - mean of vertical evolution similarity inside non-valid clusters, it is expected to be lower than MVS_VC.
11. *mean of lateral validity inside non-valid clusters (%)* (MLS_NVC) - mean of lateral similarity over non-valid clusters, it is expected to be lower than MLS_VC.

One may wonder about the last two quality indicators (mean of vertical validity among non-valid clusters and mean of lateral validity among non-valid clusters), since they are evaluating the non-valid clusters. The non-valid indicators will not be discarded because taking them out would decrease the performance of the best-fit-cluster.

### 3.1.6 Comparison Technique

To identify the best clustering method, the following steps will be followed:

1. For each quality indicator, compute a comparison measure between methods by using Equation 3.2. This equation evaluates how good or worse the indicator is from the mean (second term of Equation 3.2).

2. These values are summed up for each clustering technique, creating one single metric that correlates all eleven quality indicators, as described in Equation 3.3.

$$\text{comparing\_value}_{i,j} = \text{rank}_{i,j} \pm \frac{\text{quality\_mean}_i \; - \; \text{quality\_value}_{i,j}}{100} \tag{3.2}$$

In Equation 3.2, $i$ is the quality indicator and $j$ is the Clustering Technique. $\text{rank}_{i,j}$ is the rank the clustering algorithm $j$ occupies compared to the others at a specific quality indicator $i$. $\text{quality\_mean}_i$ is the mean of all the values at the quality indicator $i$, and the quality_value$_{i,j}$ is the corresponding value of technique $j$ at parameter $i$.

rank$_{i,j}$, in Equation 3.2, takes values between {1, ..., number_of_clustering_techniques} $\in N$ , where 1 is the best value. *rank* is assigned to each column of each metric. If there are two equal values, they get the same *rank*.

In Equation 3.2, if the indicator referring to has the logic:

- *the larger the value, the better*, ± turns to $+$;
- *the smaller the value, the better*, ± is a $-$.

The + or - chosen for each indicator, at Equation 3.2, is synthesised in Table 3.2.

Table 3.2: signal Assignment comparing Clustering techniques

| number of clusters | trajectories cluster membership assignment (%) | trajectories cluster's valid membership assignment (%) | intra-cluster valid clusters (%) | average flights per cluster | standard deviation flights per cluster |
|---|---|---|---|---|---|
| none | + | + | + | none | none |
| percentage of clusters inter-cluster valid (%) | mean of Vert. validity among valid clusters (%) | mean of Lat. validity among valid clusters (%) | mean of Vert. validity among non-valid clusters (%) | mean of Lat. validity among non-valid clusters (%) | |
| + | + | + | + | + | |

All quality indicators selected follow the logic ”the larger the value the better”. When some characteristics do not express quality, its value is assigned to $none$.

The final results for each technique ($R_j$) is the sum of all results from Equation 3.2 with the same $j$.

$$R_j = \Sigma_{i=1}^{N} comparing\_value_{i,j} \tag{3.3}$$

Where N are the quality parameters.

The technique with the lowest $R_j$ is the best quality clustering technique (Equation 3.4). It means that, in the majority of quality indicators, the technique was better-positioned (first places) compared to the other.

$$\text{best-clustering-technique = Min}(R_j) \tag{3.4}$$

### 3.1.7 Best-fit Cluster

When comparing the results of applying the *clustering and validation* algorithms for a whole day or just for one hour it was observed that the latter provided considerably worse results. This was already expected since machine learning techniques usually require considerable amounts of data to converge.

To counter-balance this limitation, the *Best-Fit-Cluster* algorithm was devised to effectively analyse small amounts of data, such as a couple of hours of flights in a given airspace. This algorithm is deterministic, and defines which cluster centroid, saved from the *Main Flow Identification*, is more similar to each trajectory.

It can also be applied to identify which cluster, if any, a small set of trajectories belongs to. It can even be used for a single trajectory.

Figure 3.7 provides a schematic representation of the devised *Best-Fit-Cluster* algorithm. It receives as input the traffic to be analysed and the cluster centroids resulted from the best Clustering Technique. Then, the algorithm tags each trajectory with the best cluster (or tags it as anomaly).

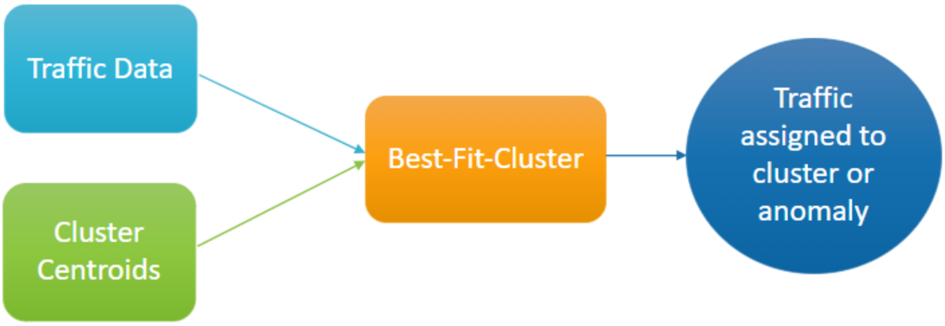


Figure 3.7: Schematic view of the best-fit-cluster algorithm.

The deterministic search for the best algorithm has the same logic as the one used at the intra-cluster validation (see section 3.1.3), but with some nuances. Instead of performing the analyses over a group of trajectories to the corresponding centroid, it is about selecting the cluster most similar to the corresponding trajectory, if any. Succinctly, the three steps of Best-Fit-Cluster:

- Perform Lateral and Vertical Evaluation (Section 3.1.3) from one trajectory to all existing clusters;
- For each trajectory, one of three possible cases is possible:
  - **Case 1.** the trajectory has *min_percentage_of_samples_valid* on all three constraints (2D lateral similarity, heading similarity, vertical evolution similarity) at one or, possibly, more clusters;
  - **Case 2.** the trajectory only has *min_percentage_of_samples_valid* at the lateral constraints, but not at the vertical evaluation, at one or more clusters; it means that the trajectory could not comply all the three constraints with not even one cluster;
  - **Case 3.** the trajectory does not even comply Lateral Evaluation with any cluster.

  These cases are tested by this order. If the trajectory observes *Case 1* at one or possibly more, the next cases will not be tested; the same applies to *Case 2* (*Case 3* will not be tested).
- *Case 3* trajectories will be assigned as anomalies. *Case 1* or *Case 2* trajectories will be assigned to a cluster.

Note: it was considered the possibility that the trajectory verified *Case 1* or *Case 2* with more than one cluster. In case this happens, the mean of similarity_percentage at the three constraints is taken, and the cluster with highest mean is the one selected.

Table 3.3 summarizes the Constants mentioned above, in this Section 3.1.7.

Table 3.3: Constants assignment

| Constant | Meaning |
|---|---|
| *min_percentage_of_samples_valid* | the minimum amount of samples that need to be valid in each of the three constraints. At this study all of the constraints have the same *min_percentage_of_samples_valid*. |

## 3.2 Complexity Estimation

Figure 3.8 represents a sketch of the proposed system for Complexity Estimation using the computation of which CAPAN tasks are used at the traffic, time interval and space that is intended to analyse. The objective is to compute the optimal sector configuration that minimizes complexity. It starts by Pre-Selecting Airspace Sector Configurations (PASC). Then, it performs the complexity estimation (CE) and proposes a new Airspace Sectorisation (AS), in order to provide the Optimal Sector Configuration.

There are a lot of necessary inputs to get one only output. Besides a given traffic data of interest, it is required to provide all existing sector configurations and their corresponding 3D geometries (latitude, longitude, altitude limits). The Hourly Entry Count (HEC) of aircrafts that enter each sector, is used during the pre-selection stage, and the CAPAN table provides the mean time spent on each task. Finally, $[t_0; t_1]$ and ”workload time resolution” parameters defines the analysis time constraints.

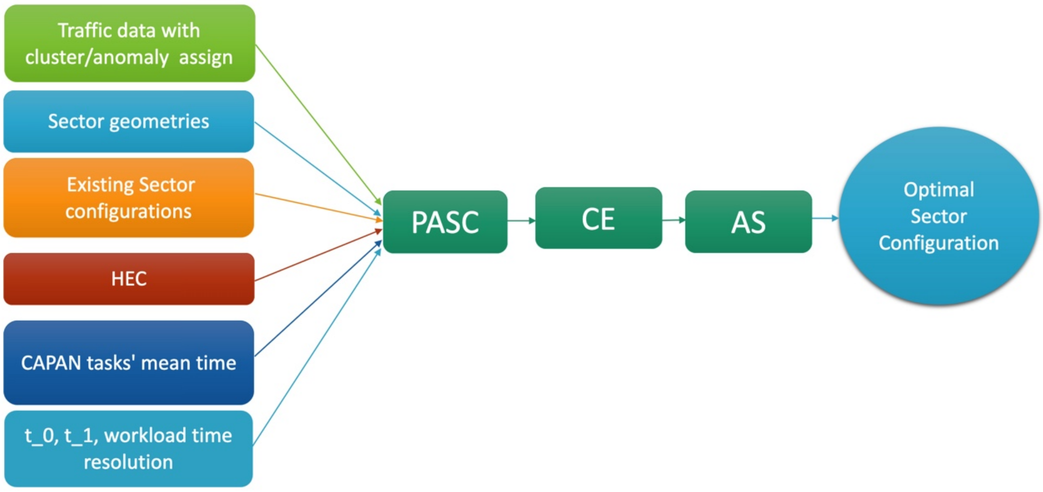

Traffic data with cluster/anomaly assign
Sector geometries
Existing Sector configurations
HEC
CAPAN tasks' mean time
t_0, t_1, workload time resolution
PASC
CE
AS
Optimal Sector Configuration

Figure 3.8: Complexity Estimation Scheme

### 3.2.1 Pre-selection of Airspace Sector Configurations (PASC)

Hourly Entry Counts (HEC) of aircrafts that enter each sector is a well-studied number that has been in use at the air traffic control sector for long. In particular, [53, 54] describe in a detailed way how these numbers are obtained. It is an entity of each sector, i.e., each sector has its own tabled peak and sustain HEC, which may also change with the day time. Another parameter that is also widely accepted, and used is Occupancy Counts (OCC). OCC could also have been used here, instead of HEC, or conjunction of both.

While it is possible to change its step/duration parameters, HEC is usually measured every 20 minutes, for a given interval which comprehends 1h+20minutes. For instance, [$t_0$; $t_0$ + 1hour + 20min.], [$t_0$ + 20min.; $t_0$ + 1hour + 40min.], [$t_0$ + 40min.; $t_0$ + 2 hour] and so on.

Usually, the tabled HEC are compared with the predicted HEC to identify hotspots. Figure 3.9 is a representation of the peak and sustain values, which in this example are constant, and the anticipated HEC values calculated every 20 minutes.

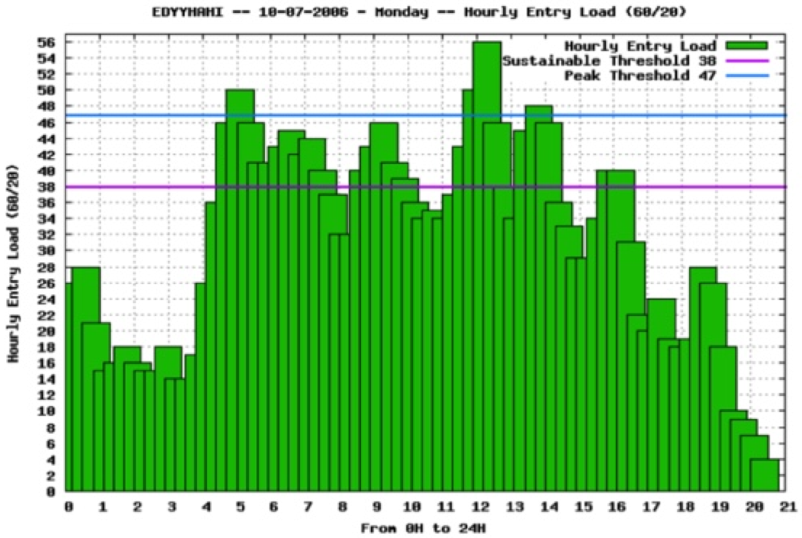


Figure 3.9: Graph demonstrating HEC between 0 to 24H. In this example, it seems that from 21 to 24H there was not any traffic

Usually the sustain level is taken as the advisable maximum value. In this example, it can be observed almost continuous hotspot values between 04:20 and 7:40, 8:40 and 10:00, and then again between 11:20 and 14:40, as well as between 15:20 and 16:40.

The peak level, is a value that must not be surpassed. Therefore, correction measures must be taken from 04:20 to 5:40, such as applying a regulation, changing sector configuration or even cherry-picking flights to be delayed, for example.

During PASC, tabled HEC values are compared with anticipated values, to select sector configurations that can be tested at phase CE, since those sector configurations will need to have all HEC below the capacity level. Thales - EDISOFT provided the HEC values for this dissertation.

### 3.2.2 Complexity Estimation through CAPAN

The CAPAN table with the mean values each task takes to be executed is provided at [1]. For the sake of complexity estimation, an algorithm was created to, given traffic data (past, present or future) and a sector configuration, determine the complexity at a given time resolution including the complexity related with each flight. Hence, the CAPAN tasks can be classified as:

- Tasks that only refer to an isolated aircraft, such as the entry, the exit, the report on reaching a specific flight level (FL), instruction to climb, etc.;

- Tasks that refer to the interaction of two flights; usually flights that get to a distance which demands levels of cautiousness or possibly intervention.

Figure 3.10 exemplifies different tasks that can happen inside a sector. Tasks related to flight x ($flx$) are not written at the Figure, in order not to fill the picture with lots of information. All the tasks related to Flight y ($fly$) are represented, except for task 16, which happens in case the flight stays in sector for more than two minute and refers to monitoring, it happens every 2 minutes (if flight stays in sector for seven minutes, task 16 happens three times).

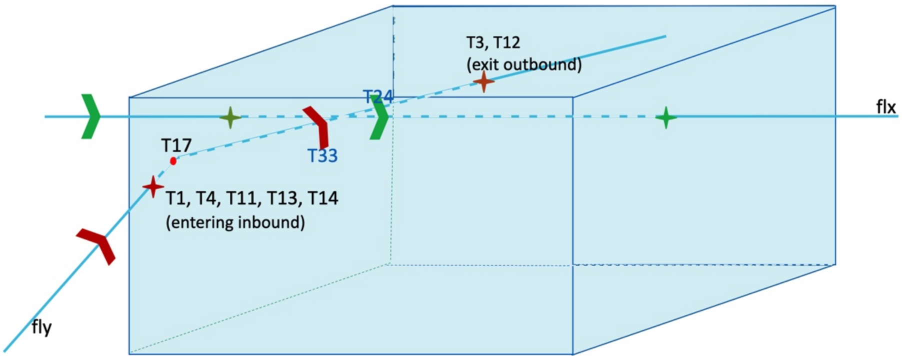


Figure 3.10: Example CAPAN tasks related to the flights according to where they come from, in which flight phase they are, if they change flight phase, the time they spend in sector, the trajectory and the existence of conflict.

When flight $y$ comes in at climbing from inbound, five tasks are associated: receiving flight information (T1), receive time and level estimation from same ACC (T4), first call with aircraft that comes from same ACC (T11), conflict search to establish initial level clearance for flight entering the sector in climb or descent (T13) and, considering that the only flight in sector is $flx$, by the the time $fly$ passes, conflict search to establish sector planning clearance (T14).

When $fly$ changes from climb to cruise phase, T17 happens, report on an aircraft on reaching a specified level has to be performed. T33 (radar intervention two aircraft on crossing tracks, both in cruise at same FL) needs to be called when the flights are that close, as it is drawn inside sector, their distance is bellow the minimum. Some time after, T24 is called to supervise the two aircraft, at crossing tracks, same FL.

T3 (transmit time and level estimate to different ACC) and T12 (last call when leaving sector), need to be performed when aircraft is leaving, considering that is goes to another ACC.

The full list of tasks is at Appendix A.2. In the pages that follow on this Chapter, the interpretation and application of the Tasks written at Appendix A.2 will be explained in detail. The explanations that follow make use of flowcharts to describe how some tasks are inter-twinned with others and the common thread between the tasks.

### Sequence of Individual Tasks for each Aircraft

The flowchart presented in Figure 3.11 a flowchart presents how the different individual tasks are handled. The algorithm tests the flights individually in a sector. To know if the aircraft comes from inbound or outbound, the time when the aircraft enters in ACC is compared with the time when it enters in the sector of analysis.

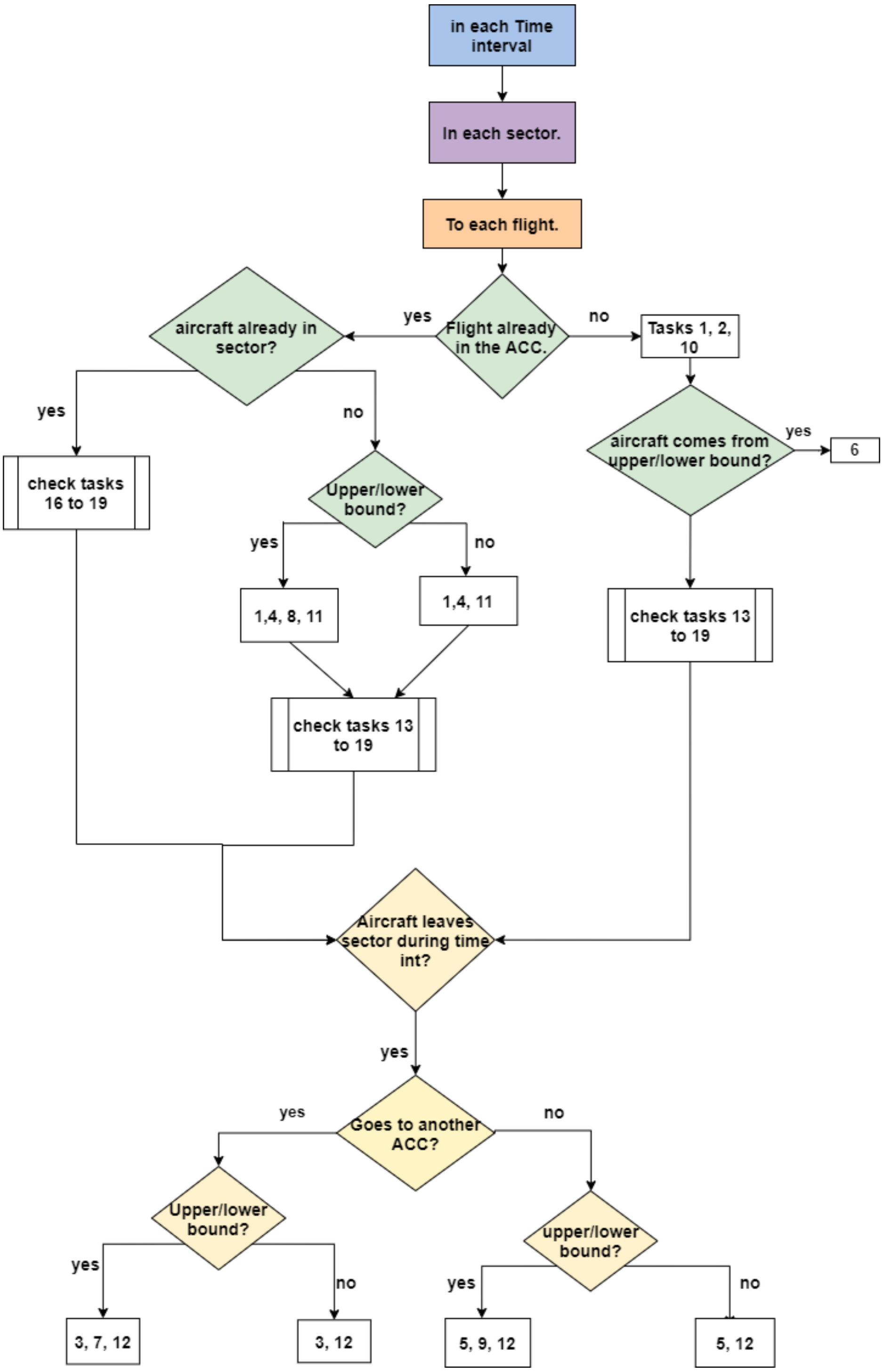


Figure 3.11: Verification of CAPAN individual parameters

As said before, the complexity is computed between a given time interval [$t_0$, $t_1$]. If the aircraft was already in ACC before $t_0$, the tasks on the right of the Flowchart (Figure 3.11) are tested. If the flight was in ACC but enters in sector during [$t_0$,$t_1$], it is an inbound entrance but if it comes from upper or lower bound must also be tested (it is only tested in case of climbing or descending aircraft).

the entrance from upper or lower bound is a verification if the altitude of the flight's first sample is between the sector's lower bound altitude and the sector's lower bound altitude plus *threshold_upper_lower_bound* or between the sector's upper bound altitude and the sector's upper bound altitude minus *threshold_upper_lower_bound*. The sample corresponds to a pandas dataframe row containing information: flight_id, icao24, callsign, latitude, lontitude, groundspeed, vertical rate, cluster, altitude, track.

In case, one of these tests is verified the flight is considered to come in from upper or lower bound. notice that flights RVSM certified, do fly at vertical separation 1000ft, between FL290 and FL410, and the sectors lower and upper boundaries are such that do not disturb this logic, for instance between FL325 and FL355.

Whether the aircraft comes from outbound (another ACC) or from inbound, tasks 13 to 15 need to be tested, they are enunciated at Figure 3.12 (the tasks and decisions (lozenges) in white boxes).

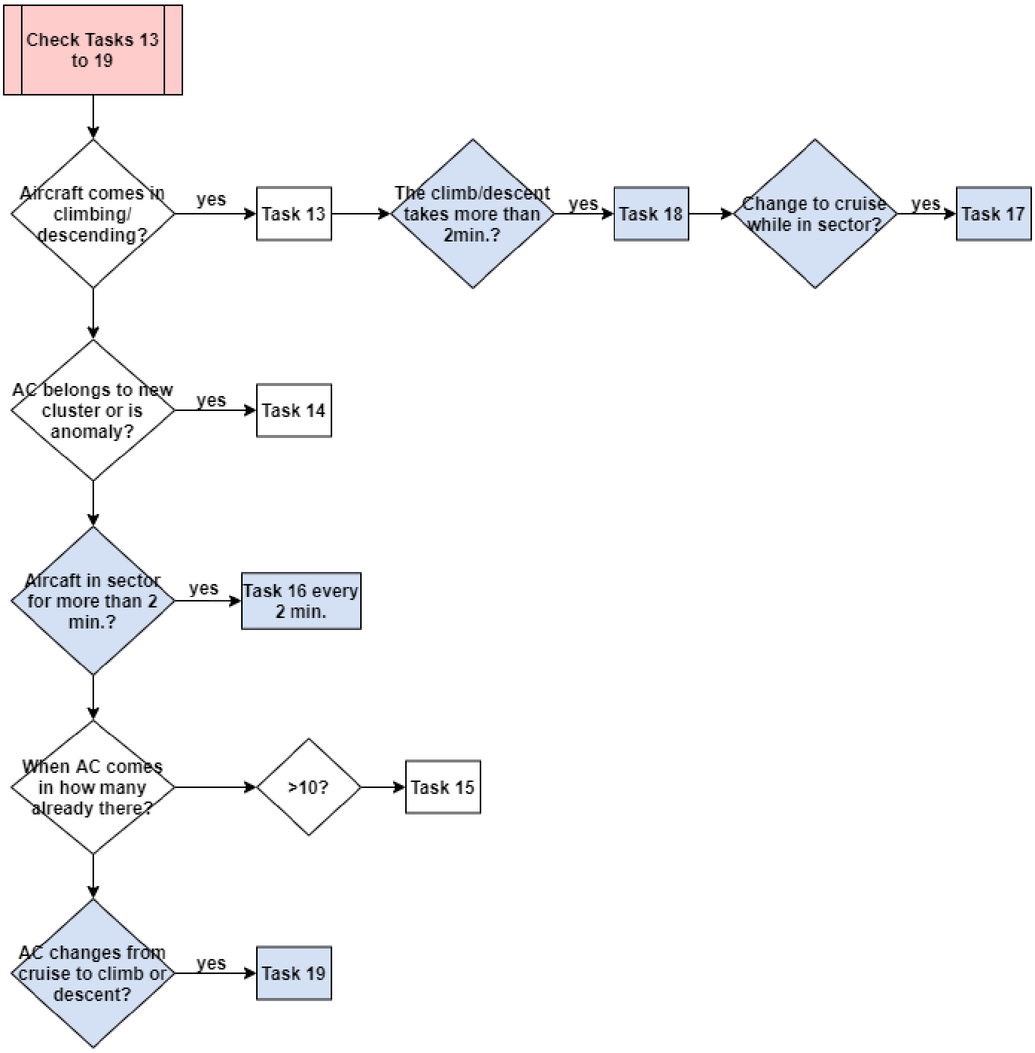

Check Tasks 13 to 19
Aircraft comes in climbing/ descending?
yes
Task 13
The climb/descent takes more than 2min.?
yes
Task 18
Change to cruise while in sector?
yes
Task 17
AC belongs to new cluster or is anomaly?
yes
Task 14
Aircaft in sector for more than 2 min.?
yes
Task 16 every 2 min.
When AC comes in how many already there?
>10?
Task 15
AC changes from cruise to climb or descent?
yes
Task 19

Figure 3.12: CAPAN individual parameters corresponding to Tasks 13 to 19

For any kind of flight, whether its was already in sector or not before $t_0$, tasks 16 to 19 need to be verified. An important issue to be aware of, is that the same aircraft can enter and exit in a sector more than once, and only in case the time between the aircraft going out and entering again is more than two minutes, a new entry has to be considered.

The flowchart at Figure 3.11 also shows how tasks for flights that exit before $t_1$ are tested, it also depends whether it is a outbound or inbound exit and if the aircraft leaves from upper or lower bound.

### Tasks corresponding to aircraft interactions

There are three fundamental types of interaction between flights, as shown in Figure 3.13. In turn, each possible interaction has another three types that differ with the conflict heading (light pink).

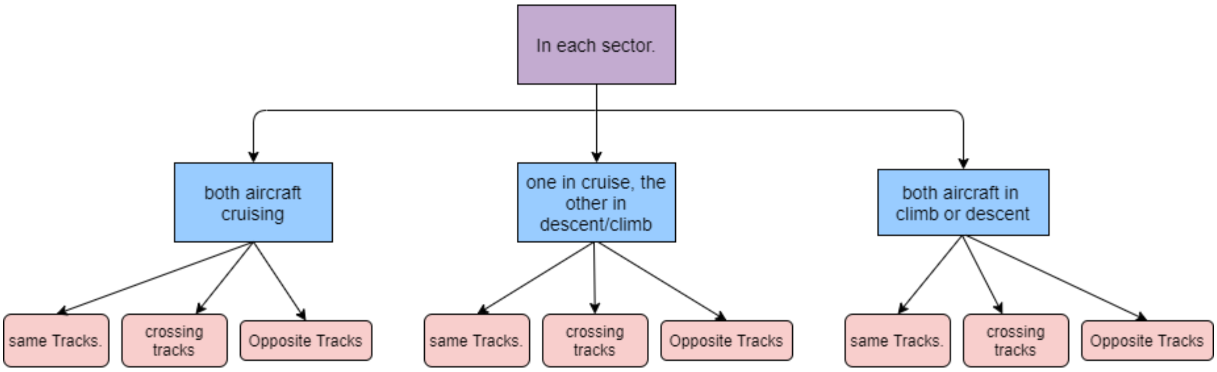


Figure 3.13: CAPAN aircraft interactions

The type of heading at conflict is defined by the conditions presented in Equations 3.5, 3.6 and 3.7, where $heading\_diff$ is nothing more nothing less than the difference between the mean of headings of *flight x* with the mean of headings of *flight y*, while they are at conflict.

$$\text{same track} : |heading\_diff| < 45^o \tag{3.5}$$

$$\text{crossing track} : 45^o < |heading\_diff| < 135^o \tag{3.6}$$

$$\text{opposite track} : |heading\_diff| > 135^o \tag{3.7}$$

It is the mean of headings, because the flight information is updated every 10 seconds, and the conflict lasts as long as the distance between both aircrafts is below a certain constant variable. Those constant variables are depicted in Figures 3.14, 3.15 and 3.16, according to the type of interaction and conflict heading.

It is possible that the standard deviation between headings of one of the pair of flights which are at conflict is greater than *constant_standard_deviation_threshold*. In this case, after having applied more filters according to the conflict heading that previously gave, the $heading\_diff$ is computed again, and the hypothesis of changing to another case is given with the dataset previous to filtering, because the new case may have larger conflict distance constants.

It is possible that both supervision and intervention happen in the same pair of flights at different epochs[1]. It is also possible, that a supervision happens, followed by an intervention, which in term, is

[1] a measure of time, unix epoch started 00:00:00 UTC 1$^{st}$ January 1970

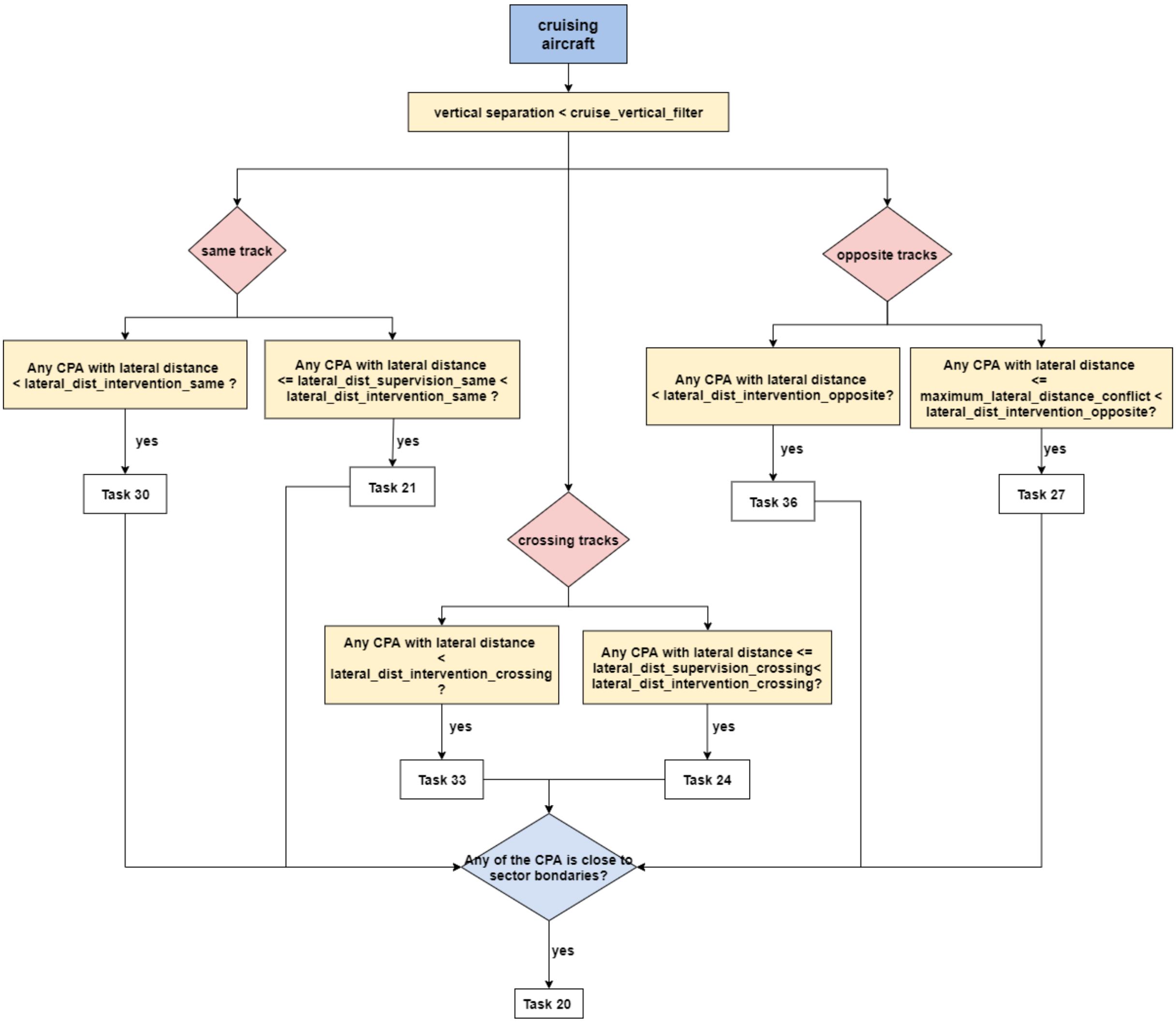
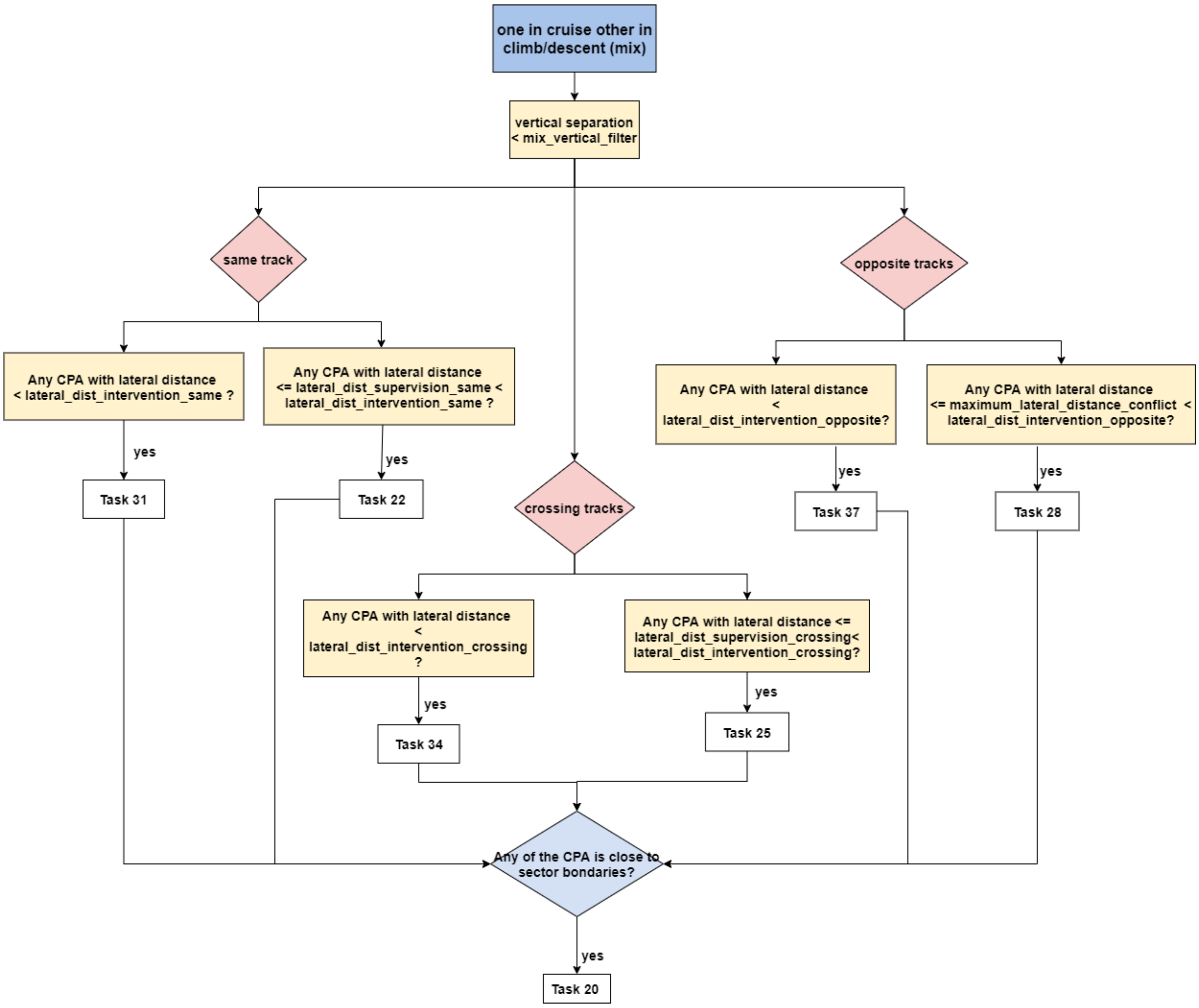

cruising aircraft
vertical separation < cruise_vertical_filter
same track
opposite tracks
Any CPA with lateral distance < lateral_dist_intervention_same ?
Any CPA with lateral distance <= lateral_dist_supervision_same < lateral_dist_intervention_same ?
Any CPA with lateral distance < lateral_dist_intervention_opposite?
Any CPA with lateral distance <= maximum_lateral_distance_conflict < lateral_dist_intervention_opposite?
yes
yes
yes
yes
Task 30
Task 21
Task 36
Task 27
crossing tracks
Any CPA with lateral distance < lateral_dist_intervention_crossing ?
Any CPA with lateral distance <= lateral_dist_supervision_crossing< lateral_dist_intervention_crossing?
yes
yes
Task 33
Task 24
Any of the CPA is close to sector bondaries?
yes
Task 20

Figure 3.14: CAPAN parameters regarding interactions between pairs of flights such that both aircraft are in cruise.

followed by a supervision, again.

To define which Tasks happen and which do not happen the algorithm starts with a pre-built function *closest_point_of_approach* [55], this function demands *maximum_lateral_distance_conflict* and *maximum_vertical_distance_conflict* and returns the pairs of flights which, at some point, verify lateral distance below *maximum_lateral_distance_conflict* or vertical distance below *maximum_vertical_distance_conflict*, as well as their information (flight_id, callsign, icao24, latitude, longitude, altitude, groundspeed). Since it is an ”or” rather then an ”and” there are many false pairs that are at vertical distance below *maximum_vertical_distance_conflict* but at lateral distance greater than *maximum_lateral_distance_conflict*, a filter is defined to eliminate these false pairs.

The next step is to identify which type of interaction does the pair of flights belong to. By checking the flight phase (cruise, climb, descent) of each. If both are cruising, the kind of tasks to be applied reduce to Figure 3.14, if one is cruising and the other is climbing or descending the kind of tasks focus on the ones mentioned at Figure 3.15, otherwise the Tasks reduce to the set at Figure 3.16. The yellow boxes at Figures 3.14, 3.15 and 3.16 represent filters that are applied as the search is straightened. In case of



one in cruise other in climb/descent (mix)
vertical separation < mix_vertical_filter
same track
opposite tracks
Any CPA with lateral distance < lateral_dist_intervention_same ?
Any CPA with lateral distance <= lateral_dist_supervision_same < lateral_dist_intervention_same ?
Any CPA with lateral distance < lateral_dist_intervention_opposite?
Any CPA with lateral distance <= maximum_lateral_distance_conflict < lateral_dist_intervention_opposite?
yes
Task 31
Task 22
crossing tracks
Task 37
Task 28
Any CPA with lateral distance < lateral_dist_intervention_crossing ?
Any CPA with lateral distance <= lateral_dist_supervision_crossing< lateral_dist_intervention_crossing?
Task 34
Task 25
Any of the CPA is close to sector bondaries?
Task 20

Figure 3.15: CAPAN parameters regarding interactions between pairs of flights such that one is in cruise and the other is in climb or descent.

a conflict between two cruising or two mixing a vertical filter is applied.

Afterwards, the type of heading at conflict is determined and, in case the standard deviation of one or both is to large, there is the possibility of changing to another heading case without loss of information.

Finally, having the type of interaction and the type of heading conflict the existence of supervision once or more and intervention once or more is tested and the right tasks are assigned. There is another detail, Task 20 is considered if any of the flights during conflict is at distance lower or equal to *distance_to_consider_task_20*.

Table 3.4 compiles the constants referred to in this section - Tasks Corresponding to aircraft interactions - and the previous - Sequence of Individual Tasks for each aircraft.

### Heading Conflict

The heading conflict is the absolute value of the difference of headings between both aircraft. However there are five special cases:

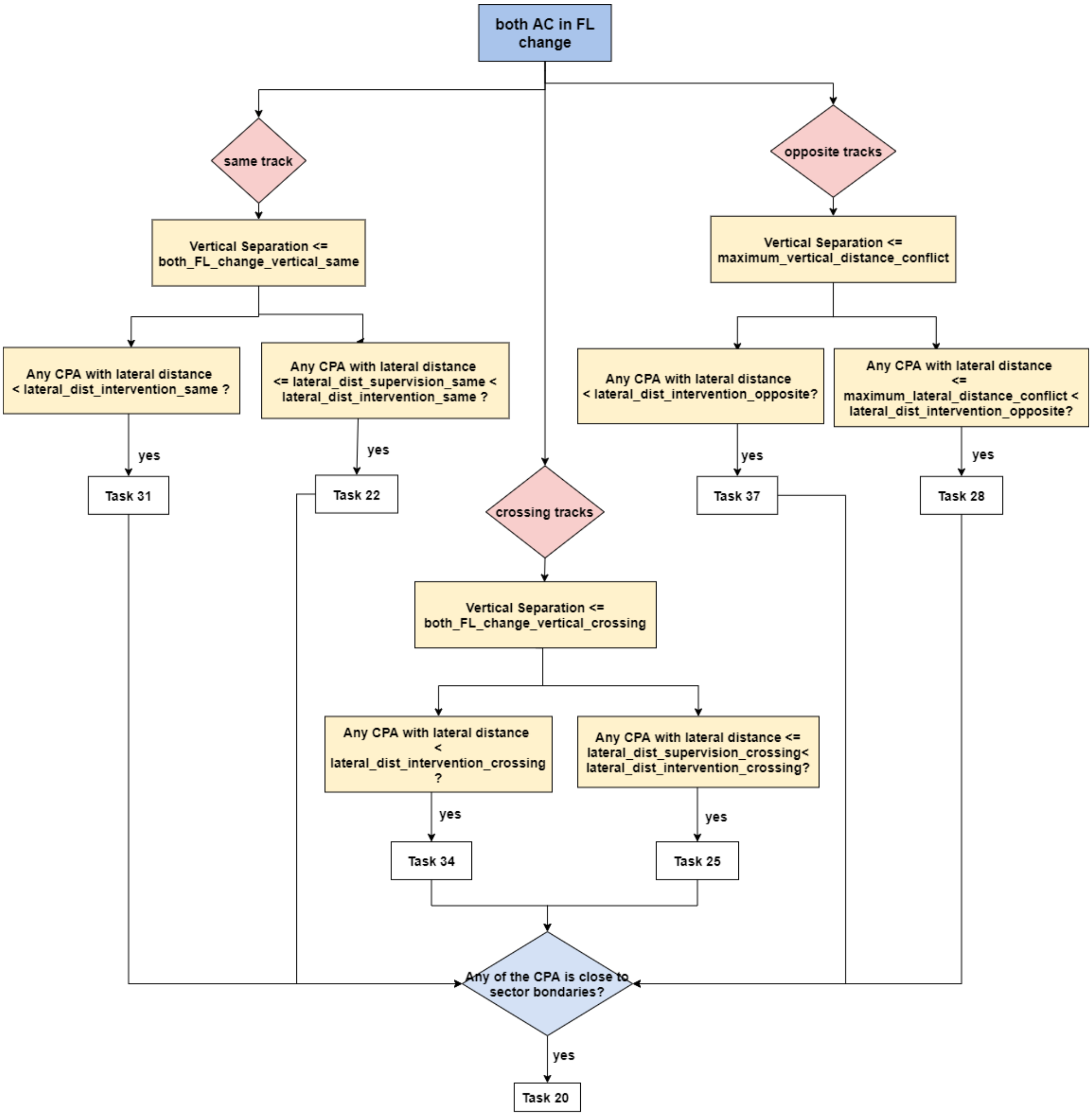

both AC in FL change
same track
opposite tracks
crossing tracks
Vertical Separation <= both_FL_change_vertical_same
Vertical Separation <= maximum_vertical_distance_conflict
Vertical Separation <= both_FL_change_vertical_crossing
Any CPA with lateral distance < lateral_dist_intervention_same ?
Any CPA with lateral distance <= lateral_dist_supervision_same < lateral_dist_intervention_same ?
Any CPA with lateral distance < lateral_dist_intervention_opposite?
Any CPA with lateral distance <= maximum_lateral_distance_conflict < lateral_dist_intervention_opposite?
Any CPA with lateral distance < lateral_dist_intervention_crossing ?
Any CPA with lateral distance <= lateral_dist_supervision_crossing< lateral_dist_intervention_crossing?
yes
Task 31
Task 22
Task 37
Task 28
Task 34
Task 25
Any of the CPA is close to sector bondaries?
Task 20

Figure 3.16: CAPAN parameters regarding interactions between flights in climb or descent.

1. If one flight is at 90º and the other flight heading is greater than 270º, the heading at conflict is:

$$360 - (h_{>270}) + 90 \tag{3.8}$$

2. If one flight is at 0º and the other has heading is greater than 180º, the heading at conflict is:

$$360^{o} - (h_{>180}) \tag{3.9}$$

3. If one flight is at heading 270º during the conflict and the other has heading lower than 90º:

$$90 + (h_{<90}) \tag{3.10}$$

Table 3.4: Constants assignment

| **Constant** | **Meaning** |
|---|---|
| *threshold_upper _lower_bound* | the threshold to sum to the sector's lower bound and the subtract to the sector's upper bound to test if an entry or an exit happens from these bounds |
| *maximum_lateral _distance_conflict* | greatest lateral distance from which supervision starts (both at FL change at opposite tracks) |
| *maximum_vertical _distance_conflict* | greatest vertical separation from which supervision starts (both at FL change at opposite tracks) |
| *cruise_vertical _filter* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant |
| *lateral_dist_intervention _same* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict of same tracks |
| *lateral_dist_supervision _same* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance supervision for the heading conflict of same tracks |
| *lateral_dist_intervention _crossing* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict case of crossing tracks |
| *lateral_dist_supervision _crossing* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance supervision for the heading conflict case of crossing tracks |
| *lateral_dist_intervention _opposite* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict case of opposite tracks |
| *mix_vertical _filter* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant |
| *distance_to_consider _task_20* | maximum distance of one of the flights during conflict for task 20 to be considered also |
| *both_Fl_change_vertical_same* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant |
| *both_Fl_change_vertical _crossing* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant |
| *time_workload _resolution* | time resolution to complexity determination (Equation 3.13) |
| *percentage_policy _stop_search* | percentage decision-maker (Equation 3.16) |

4. If one flight is in first quadrant and the other is in the third such that the heading is greater than ($h_{1q}$+180) or forth:

$$(360 - h_{3q\_or\_4q}) + h_{1q} \tag{3.11}$$

5. If one flight is in second quadrant and the other is in forth with heading greater than ($h_{2q}$+180):

$$(360 - h_{4q}) + h_{2q} \tag{3.12}$$

### 3.2.3 Airspace Sectorisation

Complexity is a function of time and tasks performed. When a task is confirmed to happen it is also assign to the time it happens. Just like a dictionary, where the keys are the time when tasks are suppose to happen, more precisely, the epoch [2] and the value, assigned to the key, is a cumulative sum of the time means associated to the tasks that happened at the same epoch.

Whenever the algorithm finds a task that is predicted to happen, according to a flight trajectory or a conflict, it also determines when it happens. Hence, the epoch to when it is predicted to happen is searched in the dictionary and, if there is a key with the same epoch, the time mean associated to the task (see Appendix A.2) is added to the value already there (a simple sum). Otherwise, a new key is defined, assigned to the time mean related to the task.

After having computed the complexity at the sector configurations, pre-selected at stage one (3.2.1), the airspace Sectorisation phase is designated to compute the $Complexity$ at a time resolution between $t_0$ and $t_1$ as shown in Equation 3.13:

$$Complexity(\%)_j = \frac{\begin{bmatrix}\text{total workload time by the sum of tabled\_tasks\_mean\_time} \\ \text{occurred inside } [t_i,\, t_i + \textit{time\_workload\_resolution}]\end{bmatrix}}{\textit{time\_workload\_resolution}} \times 100 \quad (3.13)$$

where $j$ = $\{t_0$, $t_0$+*time_workload_resolution*; $t_0$+ (2 ×*time_workload_resolution*);t$_0$ + (3 ×*time_workload _resolution*); ...;t$_1$ - *time_workload_resolution*$\}$.

#### Objective function

Finally, the delta function is the cumulative sum of the excess of complexity (above 70% [20]), associated to the sectors (at each time resolution) of the sector configuration being tested.

$$delta(sector\_conf) = \sum_{i}^{\text{(total\_\#\_sectors)}} \sum_{j=t_{init}}^{(t_{end}-\textit{time\_workload\_resolution})} \begin{cases} (complexity(\%)_j - 70\%) & \text{if Complexity(\%)}_j > 70\% \\ 0 & \text{else} \end{cases} \quad (3.14)$$

Where $i$ identifies the sector from the sectors which are part of *sector_conf* and $j$ goes through the time rate from the start of the time period of interest, to (t$_{end}$ - *time_workload_resolution*), because it goes in intervals of *time_workload_resolution*.

$$best\_delta_n = min[values\_from\_dictionary(delta(sector\_conf))] \quad (3.15)$$

The delta function is computed for configurations with the same number of active sectors (Equation 3.14. Each configuration has its own delta_function result, and is saved in a dictionary (key=sector _configuration_id, value = delta(sector_conf_id)). Then, the sector configuration with the smallest delta is saved in a short term memory, as best_delta$_n$ (Equation 3.15). After having the best_delta$_{n+1}$, which

[2] unix epoch started at 00:00:00 UTC 1$^st$ January 1970

corresponds to opening of one more sector, the deltas are compared. If, Equation 3.16 is verified, the search stops and the sector configuration chosen is the one related to best_delta$_n$. Otherwise, the search continuous until the next best_delta does not considerably improve the results to justify the opening of one more sector.

$$best\_delta_n \leq best\_delta_{n+1} + percentage\_policy\_stop\_search \tag{3.16}$$

Equation 3.16 means that, if the best_delta$_{n+1}$ sector configuration, with the penalty of having to open one more sector, has cumulative excess which is greater than (best_delta$_n$ - percentage_policy_stop _search), it is not worth it to open an additional sector, since the (possible) decrease in excess of workload is not significant enough.

# Chapter 4

# Experimental Results

This chapter presents the results of the conducted evaluation to validate the proposed *main-flows determination* and *complexity estimation* methods.

In order to provide meaningful results, an input dataset with a significant representation of real life traffic has to be used preferably, with typical challenges that air traffic controllers face everyday. Therefore, from the available datasets, the flights crossing the upper Switzerland airspace, from August $1^{st}$ 2018 was selected. This airspace is divided into two ACCs (Geneva ACC and Zurich ACC), which is relevant for computing the workload between different ACCs and analyse how it translates in terms of complexity.

## 4.1 Main Flow Determination

This section presents the testing results for the clustering techniques discussed on section 2.2. Figure 1.4 at the Introduction is one of the experimental results taken, where at the left is all the trajectories that happened in the day (grey lines) and on the left are the identified clusters at the best clustering technique (OPTICS align with the pre-processing, PCA with 2 components).

Table 4.1 presents the real-values associated to the constants referred to in section 3.1.3, these real-values were used to obtain the results at clustering evaluation that are going to be analysed in this section. Same values were taken from EUROCONTROL documents [10], articles [27] and other aircraft-integrated systems [52]. The rest of the values assignment, were defined as a good measure in order to disregard outliers but still consider attributes that are similar enough, when performing the lateral and vertical evaluation (explained in section 3.1.3).

### 4.1.1 Comparison of the clustering methods

This section presents the obtained results when comparing clustering methods using the approach proposed on sections 3.1.5 and 3.1.6. It is assumed that pre-processing the data using the two considered techniques (PCA or t-SNE) will provide better results or at least speed-up the machine learning method. At the first two tables (Table 4.2 and Table 4.3) it is presented the quality parameter results of each

Table 4.1: Constants for evaluation and validation methods, value assignment

| Constant | Meaning | Value |
|---|---|---|
| *resample_number* | Value to which the flight's trajectory is resampled. [27] | 50 |
| *lateral_threshold* | 2D distance above which the sample is not considered inside percentage of sample lateral distance similar. [52]. 20 NM is the surveillance range defined by TCAS. | 20NM |
| *percentage_samples_dist _similar* | Percentage of samples that need to verify a 2D lateral distance equal or below 20NM, otherwise the trajectory will not be taken as lateral valid. | 70% |
| *heading_threshold* | Heading difference above which the sample will not be considered inside the percentage of samples heading similar. | $10^o$ |
| *percentage_samples_head _similar* | Percentage of samples needed to verify heading difference below or equal to *heading_threshold*, otherwise the trajectory will not be taken as lateral valid ( a good characteristic in inter-cluster, a bad characteristic in intra-cluster). | 70% |
| *percentage_samples_vertical _evolution_similar* | Percentage of samples needed to verify vertical evolution similarity, otherwise the trajectory is not vertical valid. | 70% |
| *percentage_trajectories_to_be _considered_valid* | Percentage of trajectories that were assign to the cluster at the Machine Learning algorithm that need verify lateral and vertical evolution such that the cluster is *intra*- valid. | 70% |
| *vertical_evolution_threshold* | EUROCONTROL defined that a flight that is changing altitude by 500ft/min is a flight in climb or descent. [10] | 500ft |

resulting technique. The clustering names appears as <main_method_name> + X, being X an integer, X=0 means that no pre-processing technique was executed, X=[2, 4, 6] refers to PCA pre-processing, where the number at X is the number of principal components considered, and X= (-1) refers to t-SNE pre-processing. The False and the True at column clustering_name was needed to call R-DBSCAN and R-DBSCAN*, and all the other technique also needed to have this information appended to them because the function they call is the same, to be as generic as possible.

All these techniques were subjected to clustering, evaluation, possible merge and re-evaluation.

The DBSCAN algorithm ran with the default value of $\epsilon$ = 0.5 and $min\_trajectories$ =10. The R-DBSCAN and R-DBSCAN* algorithms, using PCA 6 were not able to identify more than one cluster, which is a very bad result when clustering a full and busy day. The algorithm is set not to discard and do not execute evaluation and enhancement when the clustering technique only identifies one cluster.

OPTICS and HDBSCAN have $min\_trajectory = 10$, i.e., the minimum number of flights clustered together to be considered a cluster is 10, which is around 1% of the total data, given that the total flights in that day (upper airspace) was 1224.

Apart from MF_VC and SDF_VC, the values represent fractions. Figure 4.1 represents the top four in rank clustering methods: OPTICS with PCA 2, DBSCAN with t-SNE, DBSCAN with PCA 2 and R-DBSCAN* with t-sNE. DBSCAN with t-SNE (related to as DBSCAN -1 False) presents null lateral and vertical similarity mean among non-valid clusters. However, the other quality indicators (CIV, MVS_VC and MLS_VC) are high enough in comparison to the other techniques to put DBSCAN t-SNE at the top

Table 4.2: Main Characteristics of Clustering Techniques (part1)

| clustering_name | NC | TAC | TAVC | CIV | MF_VC | SDF_VC |
|---|---|---|---|---|---|---|
| DBSCAN 0 False | 17 | 0.45 | 0.45 | 0.88 | 32.88 | 17.56 |
| DBSCAN 2 False | 12 | 0.90 | 0.08 | 1.00 | 26.00 | 19.24 |
| DBSCAN 4 False | 15 | 0.60 | 0.34 | 1.00 | 38.45 | 19.34 |
| DBSCAN 6 False | 17 | 0.54 | 0.41 | 0.76 | 34.07 | 16.55 |
| DBSCAN -1 False | 14 | 0.15 | 0.15 | 0.86 | 13.36 | 4.09 |
| RDBSCAN 0 False | 15 | 0.48 | 0.21 | 0.87 | 26.60 | 20.01 |
| RDBSCAN 2 False | 19 | 0.37 | 0.27 | 0.79 | 21.25 | 13.18 |
| RDBSCAN 4 False | 12 | 0.62 | 0.14 | 1.00 | 19.22 | 10.72 |
| RDBSCAN -1 False | 32 | 0.61 | 0.40 | 0.56 | 24.75 | 13.80 |
| RDBSCAN 0 True | 18 | 0.50 | 0.23 | 0.78 | 23.83 | 19.22 |
| RDBSCAN 2 True | 21 | 0.43 | 0.33 | 0.71 | 22.78 | 12.73 |
| RDBSCAN 4 True | 13 | 0.64 | 0.16 | 1.00 | 20.30 | 10.24 |
| RDBSCAN -1 True | 29 | 0.44 | 0.40 | 0.38 | 19.80 | 8.06 |
| OPTICS 0 False | 23 | 0.49 | 0.43 | 0.74 | 28.00 | 13.66 |
| OPTICS 2 False | 24 | 0.50 | 0.31 | 0.67 | 25.60 | 10.82 |
| OPTICS 4 False | 26 | 0.52 | 0.41 | 0.54 | 23.05 | 14.27 |
| OPTICS 6 False | 26 | 0.60 | 0.46 | 0.62 | 28.60 | 15.63 |
| OPTICS -1 False | 36 | 0.71 | 0.44 | 0.39 | 27.35 | 15.76 |
| HDBSCAN 0 False | 18 | 0.81 | 0.21 | 1.00 | 32.50 | 17.00 |
| HDBSCAN 2 False | 20 | 0.75 | 0.24 | 0.80 | 29.90 | 23.40 |
| HDBSCAN 4 False | 17 | 0.82 | 0.15 | 1.00 | 27.43 | 14.84 |
| HDBSCAN 6 False | 20 | 0.82 | 0.19 | 1.00 | 29.75 | 17.80 |
| HDBSCAN -1 False | 25 | 0.95 | 0.29 | 0.84 | 51.43 | 18.31 |

Table 4.3: Main Characteristics of Clustering Techniques (part2)

| clustering_name | PIV | MVS_VC | MLS_VC | MVS_NVC | MLS_NVC |
|---|---|---|---|---|---|
| DBSCAN 0 False | 0.88 | 0.98 | 0.91 | 0.00 | 0.00 |
| DBSCAN 2 False | 1.00 | 0.99 | 0.84 | 0.99 | 0.32 |
| DBSCAN 4 False | 1.00 | 0.98 | 0.87 | 0.78 | 0.47 |
| DBSCAN 6 False | 0.76 | 0.98 | 0.85 | 0.96 | 0.57 |
| DBSCAN -1 False | 0.86 | 0.99 | 0.96 | 0.00 | 0.00 |
| RDBSCAN 0 False | 0.87 | 0.97 | 0.94 | 0.99 | 0.46 |
| RDBSCAN 2 False | 0.79 | 0.99 | 0.89 | 0.95 | 0.41 |
| RDBSCAN 4 False | 1.00 | 0.98 | 0.94 | 0.98 | 0.27 |
| RDBSCAN -1 False | 0.56 | 0.98 | 0.88 | 0.97 | 0.35 |
| RDBSCAN 0 True | 0.78 | 0.97 | 0.95 | 0.99 | 0.49 |
| RDBSCAN 2 True | 0.71 | 0.99 | 0.88 | 0.95 | 0.41 |
| RDBSCAN 4 True | 1.00 | 0.98 | 0.95 | 0.98 | 0.27 |
| RDBSCAN -1 True | 0.38 | 0.97 | 0.89 | 0.98 | 0.37 |
| OPTICS 0 False | 0.74 | 0.98 | 0.88 | 0.97 | 0.37 |
| OPTICS 2 False | 0.67 | 0.98 | 0.86 | 0.95 | 0.44 |
| OPTICS 4 False | 0.54 | 0.98 | 0.89 | 0.97 | 0.53 |
| OPTICS 6 False | 0.62 | 0.98 | 0.87 | 0.94 | 0.44 |
| OPTICS -1 False | 0.39 | 0.98 | 0.87 | 0.96 | 0.38 |
| HDBSCAN 0 False | 1.00 | 0.99 | 0.85 | 0.98 | 0.54 |
| HDBSCAN 2 False | 0.80 | 0.98 | 0.80 | 0.98 | 0.53 |
| HDBSCAN 4 False | 1.00 | 1.00 | 0.81 | 0.97 | 0.53 |
| HDBSCAN 6 False | 1.00 | 1.00 | 0.85 | 0.98 | 0.54 |
| HDBSCAN -1 False | 0.84 | 0.99 | 0.81 | 0.92 | 0.46 |

4 ranks.

The quality indicators MLS_NVC and MVS_NVC are used in this comparison, because non-valid

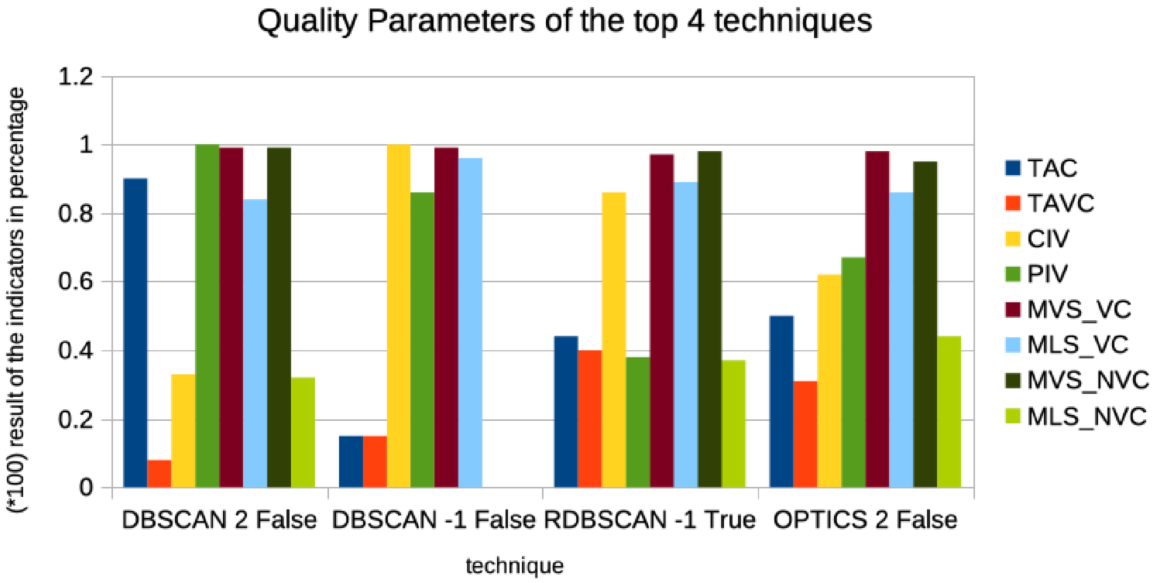


Figure 4.1: Results of the quality indicators of the top 4 clustering techniques.

clusters (that do not verify intra validity) will still be taken into account at the best-fit-cluster. Otherwise, the results at the best-fit-cluster in tagging a trajectory to a cluster would decrease significantly, and we would have more atypical routes. Since the best-fit-cluster is a deterministic algorithm, there is no problem that the cluster is non-valid.

### Rank

Table 4.4 shows the intermediate values of the comparison technique, assign to each ($quality\_indicator$, $clustering\_method$), in order to be possible to rank the clustering techniques. These values correspond to Equation 3.2, at the Proposed System chapter.

Table 4.5 ranks the comparison results between the clustering techniques, which is the sorting from the lower value of Equation 3.3 to the highest. The lower value is the best, because it means that in general the technique occupied the first places at the different quality indicators.

Looking at Table 4.5, it is concluded that the best technique is OPTICS with PCA 2. It was thought that DBSCAN would occupy the lower ranks at the Ranking Table 4.5 because of the $epsilon$ static nature. Nevertheless, DBSCAN with t-SNE gets the second place at this Table and DBSCAN with PCA 2 gets the third.

There's no significantly differences between R-DBSCAN and R-DBSCAN*, which accepts smaller clusters. They are almost always hand in hand.

Hence, it is clear that the use of a pre-processing method improves their place on the rank. Many researchers discuss that PCA and t-SNE can bias the data, but, at least here, it created better results. However, it is not possible to say which method creates the best results, t-SNE and PCA with 2 components occupy the first places.

Figure 4.2 presents the obtained cluster centroids at the best technique (OPTICS with PCA 2),

Table 4.4: Intermediate results comparison

| clustering tech | TAC | TAVC | CIV | PIV | MVS_VC | MLS_VC | MVS_NVC | MLS_NVC |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| DBSCAN 0 False | 5.15 | 17.84 | 15.91 | 17.68 | 2.00 | 8.97 | 1.88 | 1.40 |
| DBSCAN -1 False | 1.45 | 3.14 | 13.93 | 17.68 | 2.99 | 11.92 | 1.88 | 1.40 |
| RDBSCAN 4 False | 12.98 | 2.15 | 16.79 | 10.93 | 2.00 | 9.94 | 7.90 | 2.13 |
| RDBSCAN 4 True | 13.96 | 4.13 | 16.79 | 11.91 | 2.00 | 10.93 | 7.90 | 2.13 |
| DBSCAN 2 False | 18.70 | 1.21 | 16.79 | 2.35 | 2.99 | 3.04 | 8.89 | 3.08 |
| RDBSCAN -1 False | 11.99 | 13.89 | 4.23 | 8.06 | 2.00 | 7.00 | 6.91 | 4.05 |
| RDBSCAN -1 True | 4.16 | 13.89 | 1.41 | 15.82 | 1.01 | 7.99 | 7.90 | 5.03 |
| OPTICS 0 False | 7.11 | 15.86 | 8.05 | 12.85 | 2.00 | 7.00 | 6.91 | 5.03 |
| OPTICS -1 False | 14.89 | 16.85 | 2.40 | 7.12 | 2.00 | 6.01 | 5.92 | 6.02 |
| RDBSCAN 2 False | 2.23 | 9.02 | 11.00 | 13.84 | 2.99 | 7.99 | 4.93 | 6.99 |
| RDBSCAN 2 True | 3.17 | 11.96 | 7.08 | 15.82 | 2.99 | 7.00 | 4.93 | 6.99 |
| OPTICS 2 False | 8.10 | 10.98 | 6.12 | 8.06 | 2.00 | 5.02 | 4.93 | 7.96 |
| OPTICS 6 False | 11.00 | 18.83 | 5.17 | 11.91 | 2.00 | 6.01 | 3.94 | 7.96 |
| RDBSCAN 0 False | 6.12 | 6.08 | 14.92 | 9.01 | 1.01 | 9.94 | 8.89 | 8.94 |
| HDBSCAN -1 False | 19.65 | 10.00 | 12.95 | 1.40 | 2.99 | 2.07 | 2.96 | 8.94 |
| DBSCAN 4 False | 11.00 | 12.95 | 16.79 | 9.95 | 2.00 | 6.01 | 2.10 | 9.93 |
| RDBSCAN 0 True | 8.10 | 7.06 | 10.01 | 9.01 | 1.01 | 10.93 | 8.89 | 10.91 |
| OPTICS 4 False | 9.08 | 14.88 | 3.25 | 14.83 | 2.00 | 7.99 | 6.91 | 11.87 |
| HDBSCAN 2 False | 15.85 | 8.05 | 11.99 | 6.18 | 2.00 | 1.08 | 7.90 | 11.87 |
| HDBSCAN 4 False | 17.78 | 3.14 | 16.79 | 4.27 | 3.98 | 2.07 | 6.91 | 11.87 |
| HDBSCAN 0 False | 16.79 | 6.08 | 16.79 | 5.24 | 2.99 | 4.03 | 7.90 | 12.86 |
| HDBSCAN 6 False | 17.78 | 5.10 | 16.79 | 3.28 | 3.98 | 4.03 | 7.90 | 12.86 |
| DBSCAN 6 False | 10.06 | 14.88 | 9.03 | 16.80 | 2.00 | 4.03 | 5.92 | 13.83 |

Table 4.5: Results Comparing Techniques

| Rank | Method | Qualification |
|---:|---:|---:|
| 1 | OPTICS 2 False | 53.16 |
| 2 | DBSCAN -1 False | 54.38 |
| 3 | DBSCAN 2 False | 57.04 |
| 4 | RDBSCAN -1 True | 57.20 |
| 5 | RDBSCAN -1 False | 58.12 |
| 6 | RDBSCAN 2 False | 58.98 |
| 7 | RDBSCAN 2 True | 59.93 |
| 8 | HDBSCAN -1 False | 60.95 |
| 9 | OPTICS -1 False | 61.20 |
| 10 | OPTICS 0 False | 64.80 |
| 11 | RDBSCAN 4 False | 64.81 |
| 12 | RDBSCAN 0 False | 64.90 |
| 13 | HDBSCAN 2 False | 64.91 |
| 14 | RDBSCAN 0 True | 65.91 |
| 15 | HDBSCAN 4 False | 66.80 |
| 16 | OPTICS 6 False | 66.81 |
| 17 | RDBSCAN 4 True | 69.74 |
| 18 | DBSCAN 4 False | 70.72 |
| 19 | OPTICS 4 False | 70.80 |
| 20 | DBSCAN 0 False | 70.82 |
| 21 | HDBSCAN 6 False | 71.71 |
| 22 | HDBSCAN 0 False | 72.67 |
| 23 | DBSCAN 6 False | 76.54 |

and trajectories related to them. Table 4.2 confirms 24 clusters were identified, here the clusters were organized by six per graph to facilitate the understanding. The top six clusters are represented at the top left of Figure 4.2 with a number of trajectories per cluster between 65 and 32.

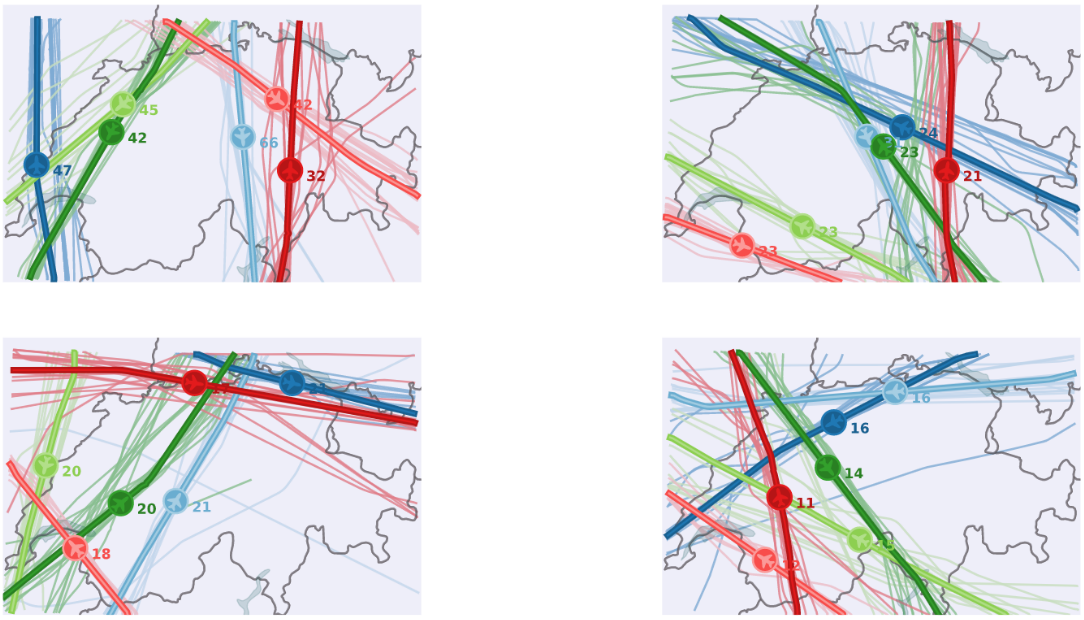


Figure 4.2: The resulting best clustering technique OPTICS PCA 2 represented with the use of sectflow library [56]. The top six clusters are at the the upper right.

The six main clusters (upper left) of Figure 4.2 show that there are at least two flows from South heading North in each corner of Switzerland upper airspace (the one more to the left may be heading France and the other may be heading Austria, Germany or Czech Republic). Another two are coming from Northeast (from German Airspace) to Southwest (heading South of France). There is also one, the most common one (with 66 flights), coming from North (maybe Germany, Austria or Poland) to South in direction to Italy or North Africa. Another identical cluster with 32 routes, is doing the opposite route heading North of Europe. At last, there is one in pink (42 flights) heading Hungary, Slovenia or Croatia. It is very interesting to find out how just one day of analysis finds such patterns.

Figures 4.3 and 4.4 isolate the valid clusters from the non-valid ones. However, the non-valid clusters are not discarded, because it ensures better performance when using the best-fit-cluster.

Some trajectories are very distant from the centroid. As an example, in cluster 20 one can see the two trajectories are very peripheral from the respective centroid. However, cluster 20 still remains a valid cluster, in opposition to cluster 6, which also has very distinct trajectories, but is considered non-valid. This happens because at least 70% of the trajectories in cluster 20 comply with the three constraints, and the two very distinct routes from the rest of trajectories are less than 30% of the total. However, in cluster 6 the story is different, because more than 30% of trajectories do not comply with Lateral (distance & heading) or Vertical Validity. The same pattern can be seen on the valid clusters 25, 8 and 14.

Looking at Figure 4.3, it is also observed that clusters 20, 25, 8 and 14 (which are inside the valid group) do have some trajectories that are very far apart from the expected center. It is because at least 70% of their trajectories verify the three constraints.

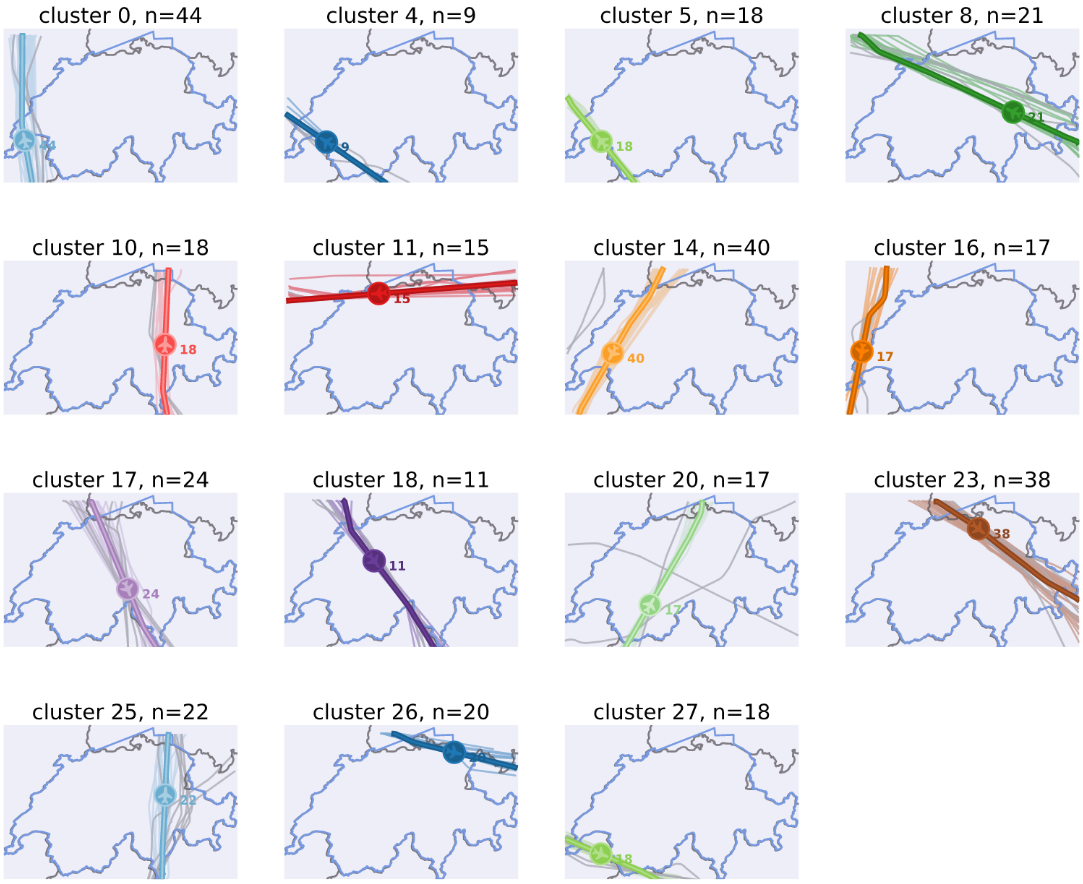


Figure 4.3: Valid clusters resulted from OPTICS with PCA 2, the technique that got the first place in rank, represented isolated, using sectflow library [56].

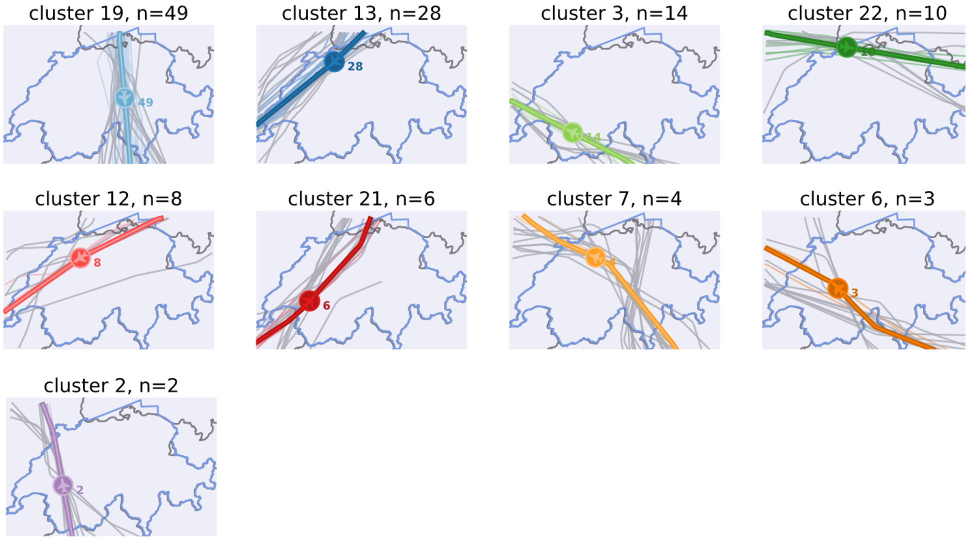


Figure 4.4: Invalid clusters resulted from OPTICS with PCA 2, represented isolated, using sectflow library [56].

One improvement that could be applied in the future would be to use the ”applicability value” of the trajectories, mentioned in 3.1.3, which tell if the trajectory verifies the 3 constraints or just the lateral ones or none, to exclude the ones that verified none. The number of flights that were still considered anomalies are 624, i.e., 50% could not be tagged to a cluster. This is still a very high number of non assignments. If DBSCAN with PCA 2 was chosen instead, only 10% of the trajectories would be tagged as anomalies as one can see at Table 4.2 at quality indicator TAC trajectories assigned to clusters, but the very low value of trajectories assigned to valid clusters TAVC and intra-clusters valid CIV reduces the level confidence of this technique and takes this technique to the second place in rank. A future enhancement would be to eliminate the quality indicators TAC and TAVC and have another indicator measuring the reason of trajectories with *applicability value* 1 or 2 (section 3.1.3).

### 4.1.2 Influence of the improvement inserted

The effectiveness of the cluster merging technique proposed in section 3.1.4 is determined by computing the mean of:

$$X_{merge} - X_{no\_merge} = diff_X \tag{4.1}$$

Where X is the considered clustering method.

Table 4.6: Results difference between merge and no merge

| clustering_name | PCM | TAC | TAVC | CIV | PIV | MVS_VC | MLS_VC | MVS_NVC | MLS_NVC |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| DBSCAN 0 False | 0.30 | 0.0 | 0.0 | 0.0 | 0.28 | 0.0 | -0.02 | 0.0 | 0.0 |
| DBSCAN 4 False | 0.12 | 0.0 | 0.0 | -0.02 | 0.12 | 0.0 | 0.01 | 0.0 | 0.0 |
| DBSCAN 6 False | 0.21 | 0.0 | 0.0 | -0.01 | 0.18 | 0.0 | 0.0 | 0.0 | 0.0 |
| DBSCAN -1 False | 0.13 | 0.0 | 0.0 | 0.0 | 0.13 | 0.0 | 0.0 | 0.0 | 0.0 |
| RDBSCAN 0 False | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| RDBSCAN 2 False | 0.19 | 0.0 | 0.0 | -0.02 | 0.17 | 0.0 | -0.01 | 0.0 | 0.0 |
| RDBSCAN 4 False | 0.15 | 0.0 | 0.0 | -0.02 | 0.15 | 0.0 | -0.02 | 0 | 0 |
| RDBSCAN -1 False | 0.12 | 0.0 | 0.0 | -0.03 | 0.09 | 0.0 | -0.01 | 0.0 | 0.0 |
| RDBSCAN 0 True | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| RDBSCAN 2 True | 0.50 | 0.0 | 0.02 | 0.04 | 0.64 | 0.0 | -0.01 | -0.02 | -0.1 |
| RDBSCAN 4 True | 0.38 | 0.0 | 0.0 | -0.04 | 0.38 | 0.0 | -0.01 | 0.0 | 0.0 |
| RDBSCAN -1 True | 0.34 | 0.0 | 0.01 | 0.0 | 0.47 | -0.01 | -0.04 | 0 | -0.05 |
| OPTICS 0 False | 0.36 | 0.0 | 0.0 | -0.03 | 0.38 | -0.01 | -0.04 | 0.0 | 0.0 |
| OPTICS 2 False | 0.29 | 0.0 | 0.01 | -0.02 | 0.24 | 0.0 | -0.02 | -0.01 | 0.0 |
| OPTICS 4 False | 0.27 | 0.0 | 0.0 | -0.02 | 0.27 | -0.01 | -0.01 | 0.0 | 0.0 |
| OPTICS 6 False | 0.21 | 0.0 | 0.0 | -0.02 | 0.24 | 0.0 | -0.01 | 0.0 | 0.0 |
| OPTICS -1 False | 0.29 | 0.0 | 0.04 | -0.01 | 0.39 | 0.0 | -0.01 | 0.01 | -0.02 |
| HDBSCAN 2 False | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| HDBSCAN -1 False | 0.08 | 0.0 | 0.0 | -0.03 | 0.07 | 0.0 | -0.01 | 0.0 | 0.0 |
| mean | 0.21 | 0.0 | 0.0 | -0.01 | 0.22 | 0.0 | -0.01 | 0.0 | -0.01 |

In Table 4.6 is presented the fraction of changes introduced in the quality indicators for all the considered clustering methods that had clusters which were not inter-cluster valid. Some cluster techniques are not represented at Table 4.6 because all their clusters were inter-valid, these techniques are DBSCAN 2, HDBSCAN, HDBSCAN 4 and HDBSCAN 6.

The meaning of the abbreviations, at first row Table 4.6, was explained in section 3.1.5, except for PCM which refers to the fraction of clusters that were merged from the total and is not a quality indicator, it is only to show how many clusters were able to be merged by the insertion of this enhancement. The indicators that changed most considerably are PCM and PIV. This enhancement is very favourable because it prevents the existence of clusters which in fact are too similar and can be misleading.

At Figure 4.5 are represented the results of PCM and PIV of two techniques which got the first and second place (OPTICS with PCA 2 and DBSCAN with t-SNE) and also the clustering techniques which most favoured from this enhancement those are R-DBSCAN with PCA 4, R-DBSCAN with t-SNE, OPTICS and OPTICS with t-SNE.

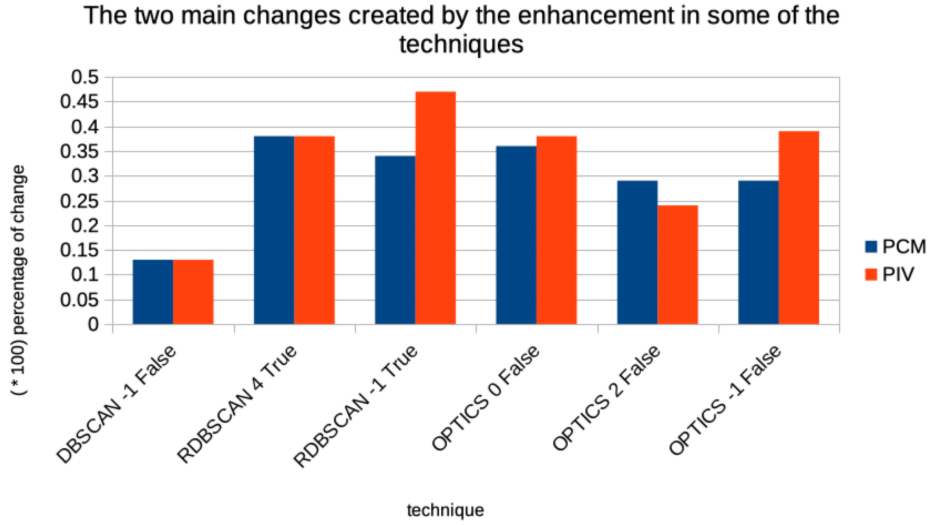


Figure 4.5: The percentage of cluster merged from total number of clusters (PCM) and the quality indicator percentage of inter-clusters valid (PIV) of selected clustering methods.

Figure 4.6 shows the mean over changes inherent to the enhancement application, the main improvements are found in the fraction of cluster merged (PCM) and on inter-cluster valid clusters (PIV) because many were merged, which shows that this enhancement is not in vain. However, looking at 4.6, there is a small decrease at the fraction of intra-clusters merged (CIV). This happens when a merged is performed between two cluster, where one (or both) were intra-cluster valid, since the number of total clusters is reduced with the merge.

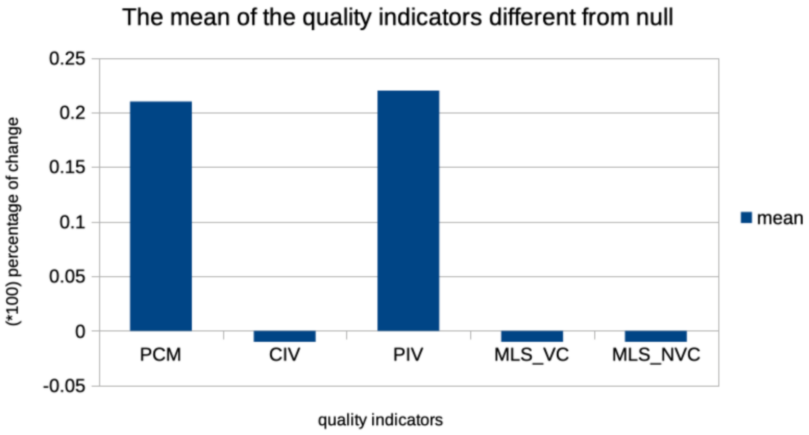


Figure 4.6: Mean of all the techniques that had clusters which were not inter-valid and tried the hypothesis of a merge

The same applies at the decrease of mean lateral similarity inside valid clusters (MLS_VC), this can be explained by the situation of two clusters, where one or both which were already lateral-valid were merged and the lateral validity, even though above 70%, decreased a bit, from the one which, before merging, had better lateral validity or from both. The mean of lateral similarity inside non-valid clusters (MLS_NVC) also decreases, this can be explained by the situation of a non-valid cluster but very close to be valid, which was improving this quality parameter, merged with another cluster creating a valid-cluster, leaving only clusters with worse results contributing to this parameter.

### 4.1.3 Best-fit Cluster

Best-fit-cluster benefits, in opposition to clustering, is that it uses clusters provided from the clustering technique that tested to be the best, to find out to which typical route a set of flights belong to. This is a valuable benefit because clustering demands considerable amounts of data to find patterns and present useful results. Best-fit-cluster is a tool that can be applied in smaller analysis for instance a morning, a specific hour or a set of flights that changed trajectory unexpectedly. It is faster and more practical. An analysis of the best-fit-cluster is performed over a specific hour (9 a.m. to 10 a.m.), then the focus goes to Geneva's ACC to show which of the common trajectories were identified at that airspace, along with the trajectories that did not fit into any of the typical routes.

At last, as this analyses, the best-fit, ran with the same data used at clustering, but shortened. A comparison is made between the results of the best-fit cluster in each hour analysed in isolation with the results that the clustering provided and four different cases were identified. This shows the different nature of the machine learning method with the deterministic technique, still both are useful, but have different applications. Best-fit-cluster is a very useful tool to be used with smaller dataset giving faster and practical results. Table 4.7 presents the real-values assigned to the constant defined at section 3.1.7, the value was obtain as a trade-off of not accepting trajectories which are very distinct, nevertheless accepting trajectories with small differences but still very similar.

Table 4.7: Constants assignment

| **Constant** | **Meaning** | **Value** |
|---|---|---|
| *min_percentage_of _samples_valid* | the minimum amount of samples that need to be valid in each of the three constraints. At this study all of the constraints have the same *min_percentage_of_samples_valid*. | 70% |

The considered methodology to analyse if a flight belongs to a cluster using the independent *Best-fit Cluster* approach, is deterministic, as explained at Section 3.1.7. As an example, flights taken between 9 a.m. and 10 a.m. are submitted to best-fit-cluster algorithm with the cluster centroids obtained from OPTICS algorithm with PCA 2 pre-processing (the best qualified method at ranking 4.1.1). Figure 4.7, illustrates the resulting clusters of such time period, showing the importance and potential of clustering the trajectories and finding the cluster that best-fits in it. Anticipating the clusters that are going to exist in a specific time interval helps visualizing and understanding better the conflict and consequences before they take place. Furthermore, it is very helpful for the FMP to choose the best sector configuration according to the main traffic.

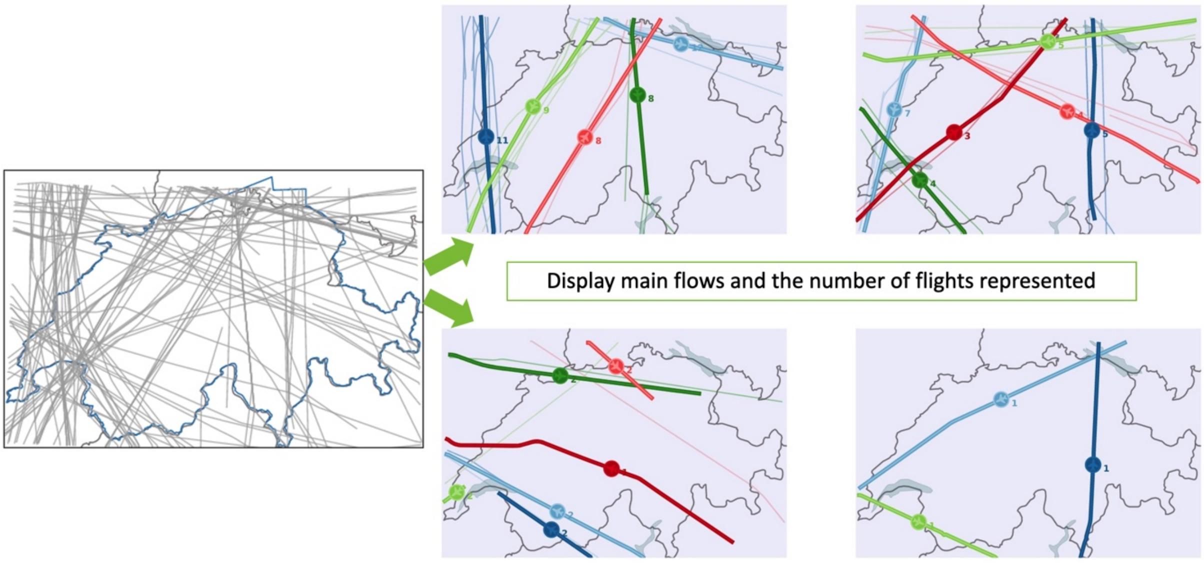


Figure 4.7: At the left the trajectories between 9 a.m. and 10 a.m.. At the right, the clusters identified by the trajectories that fitted in and the number of trajectories that fit in each cluster.

Hence, between 9 a.m. and 10 a.m., there are many flights coming from North of France heading France's South (dark blue line); flights coming from Germany heading South France; and others heading North Italy (Turin and Milan).

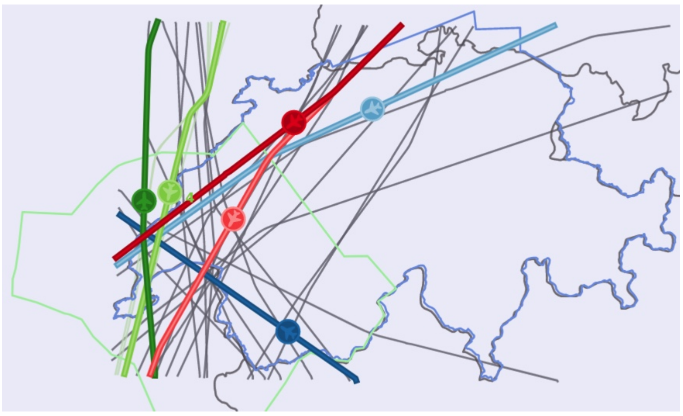


Figure 4.8: Best-fit cluster applied between 9 a.m. and 10 a.m. at Geneva's ACC. The grey lines are trajectories that did not fit in any cluster.

Figure 4.8 shows the typical routes that cross Geneva's upper airspace between 9 a.m. and 10 a.m.. It also shows other atypical routes, represented in grey. There are 6 typical routes with very distinct directions, which, depending on the time, may cause a lot of complexity. Clustering promotes a great visual simplification and decreases the complexity induced by Figure 4.7 at the left. By observing closer Figure 4.8, it can be observed that most of the routes are coming to South of France (Marseille, Lyon, Montpellier) (light green, red, pink and light blue trajectories). There is one coming from Italy heading North France (maybe Paris), the dark blue. At last, the dark green, is almost doing the opposite route that light green is performing, coming from South France heading North France close to the border with Germany.

## 4.2 Complexity Estimation

This section presents the results of the developed algorithm that searches for the best sector configurations in terms of the ATCOs workload and HEC (section 3.2.3). For such purpose, Thales - EDISOFT provided the possible sector configurations as well as the Hourly Entry Count (HEC) to develop this algorithm. Table 4.8 compiles the values assigned to the constant defined in section 3.2.2.

Apart from the value assigned to the constant *lateral_dist_intervention_same*, all the other values were considered with this value in mind and how much more or less dangerous the situations assigned to the constants were from *lateral_dist_intervention_same*. The fact that supervision must take place at higher distance than intervention, was also taken into consideration.

### 4.2.1 Optimal sector Configuration at Geneva's ACC upper airspace for time interval

Two particular time intervals are going to be studied: 5 a.m.to 6 a.m. and 9a.m. to 10 a.m.. At the first morning hour the number of flights is 71 and between 9 a.m. and 10 a.m. the number of flights increase to 128 ( + 80%). Accordingly, the optimal sector configuration suggested for each time intervals should have more sectors in the later morning hour, the study will only be based at Geneva's ACC. Results will show that between 5 a.m. and 6 a.m. the advised sector configuration is $U3K$ and between 9 a.m. and 10 a.m. the suggested sector configuration is $U4J$.

Figure 4.9 shows the simplistic representation of some possible sector configurations in Geneva's ACC, in the upper airspace. The initials, on top of the two large rectangles, represent the sector configuration identification, the initials, inside each rectangle, identify the sector. The FLs at the side, in each transition, represent the altitude limits of each sector. As the study focus on the Upper Airspace it is expected that sector configurations are vertical slices, since most flights on those FL are cruising.

Sector $U7A$ is not suggested in any of the time intervals where an extended analysis will be performed, but it represents the greatest granularity the sector configuration at Geneva's ACC can have. A sector configuration of seven sectors corresponds to the maximum number of opened sectors at Geneva's control center.

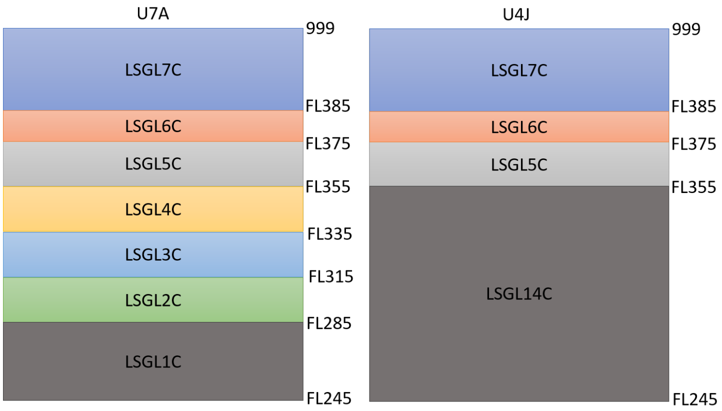


Figure 4.9: Sector configuration $U7A$ and $U4J$ focusing on Flight levels of sectors.

Table 4.8: Constants assignment

| Constant | Meaning | Value |
|---|---|---|
| *threshold_upper_lower_bound* | the threshold to sum to the sector's lower bound and the subtract to the sector's upper bound to test if an entry or an exit happens from these bounds | 400ft |
| *maximum_lateral_distance_conflict* | greatest lateral distance from which supervision starts (both at FL change at opposite tracks) | 22NM |
| *maximum_vertical_distance_conflict* | greatest vertical separation from which supervision starts (both at FL change at opposite tracks) | 6000ft |
| *cruise_vertical_filter* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant | 1000ft |
| *lateral_dist_intervention_same* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict of same tracks [57] | 10NM |
| *lateral_dist_supervision_same* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance supervision for the heading conflict of same tracks | 15NM |
| *lateral_dist_intervention_crossing* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict case of crossing tracks | 12NM |
| *lateral_dist_supervision_crossing* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance supervision for the heading conflict case of crossing tracks | 18NM |
| *lateral_dist_intervention_opposite* | filter both flights dataset in conflict such that it only considers the samples that concern lateral distance intervention for the heading conflict case of opposite tracks | 15NM |
| *mix_vertical _filter* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant | 2000ft |
| *distance_to_consider_task_20* | maximum distance of one of the flights during conflict for task 20 to be considered also | 10NM |
| *both_Fl_change_vertical_same* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant | 2000ft |
| *both_Fl_change_vertical_crossing* | filter both flights dataset in conflict such that it only considers the samples that are at vertical distance bellow or equal to this constant | 2000ft |
| *time_workload_resolution* | time resolution to complexity determination | 3 minutes |
| *percentage_policy_stop_search* | percentage decision-maker (Equation 3.16) | 20% |

**5 a.m. - 6 a.m. period**

The algorithm tested first configurations of one opened sector, if the pre-selection by the HEC accepted, computed the cumulative sum of the excesses of workload (equation 3.14) and saved at the short-term memory $best\_delta_n$. Then, it tested for the configurations of two sectors, which passed through the pre-selection of HEC, computed its current $best\_delta$ and compared with the previous $best\_delta$. Equation 3.16 did not verified, so the search continued to three opened sector, then to four, and Equation 3.16 was finally verified. It was possible to have three opened sectors, configuration $U3K$.

A simple representation of the sector configuration $U3K$, is provided on Figure 4.10. There is a big sector in the lower bound $LSGL14C$, one narrower $LSGL5C$ and a medium in the upper bound $LSGL67C$. Although, $LSGL5C$ is the narrower sector, as it is going to be seen, it is the one with greatest complexity.

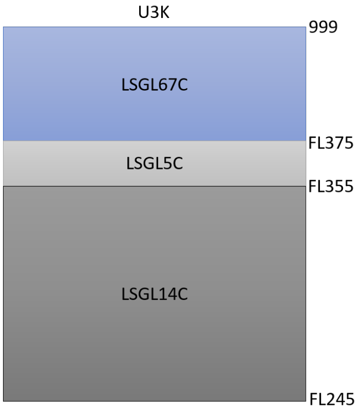


Figure 4.10: Simple Representation focused on the FL limits, $U3K$

Figures 4.11 and 4.12 present the results of HEC and complexity of the three sectors. The horizontal axis of these graphs represent the number of minutes since the start time of the analysis. For example, the value 21 translates to 5:21 a.m.. As it can be observed, sector $LSGL4C$ has a low complexity, since its workload percentage does not pass the 25% of workload, which let us assume that most of the flights are at the higher flight levels. In fact, the HEC is first 11 flights and than 12 flights, very much below the sustain value (48). Now, looking at $LSGL67C$ the complexity does not pass the 50% mark. Hence, the greatest peak for the Executive Controller (EC) is at [05:39, 05:41[ with 45% workload. The greatest peaks for the Planning Controller is (PC) 40% at [05:33, 05:36[ and [05:42, 05:45[.

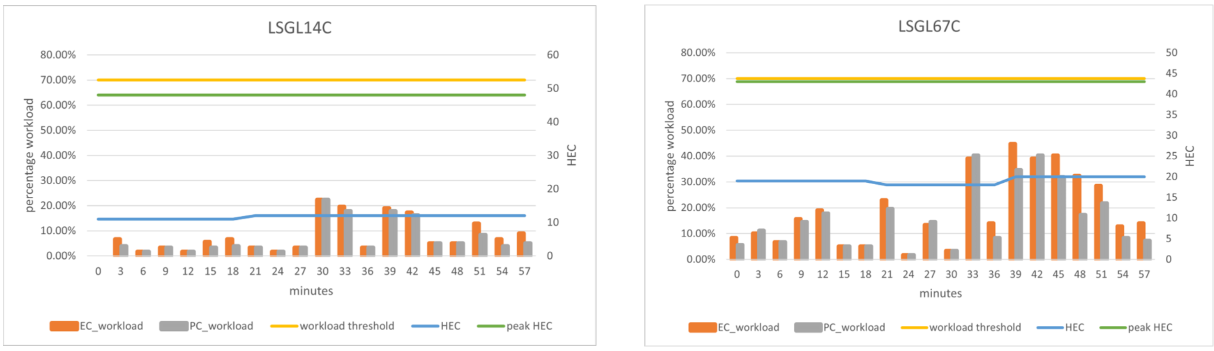


Figure 4.11: Complexity and HEC LSGL14C and LSGL67C from sector configuration $U3K$, time interval 5 a.m. to 6 a.m., the minutes are such that minute=0 is considered time interval [5:00, 05:03[ $1^{st}$ August 2018

The $LSGL5C$ analysis, represented in Figure 4.12, reveals the sector with increasing workload. It shows that usually a high complexity value comes after a high HEC. In fact, the HEC is larger (with

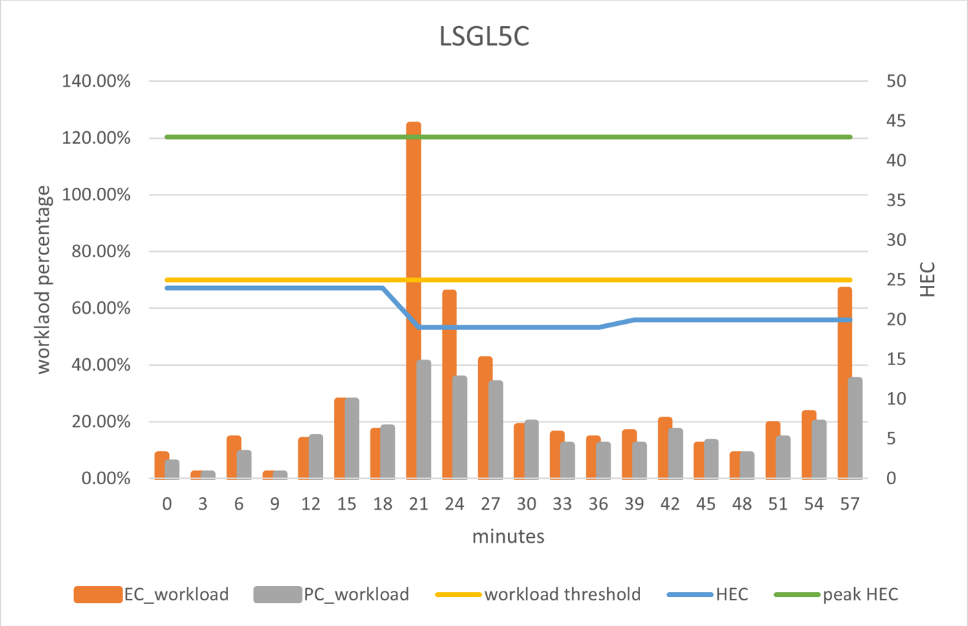


Figure 4.12: Complexity and HEC LSGL5C from sector configuration $U3K$, time interval 5 a.m. to 6 a.m., 1$^{st}$ August 2018

24 flights) between 05:00 and 05:20 than the ones in the following minutes, nevertheless this sector has capacity HEC of 43 flights. By analysing the presented data, it is clear that the capacity does not always depend on the the number of entry flights but the type of trajectories ocurring simultaneously and possible conflicts. The high peak is greater than 120% and comes at time interval [05:21, 05:24[. This also shows that, the increase of complexity is due to conflicts, the tasks that demand more time (an analysis of the high peak will be performed in section 4.2.3).

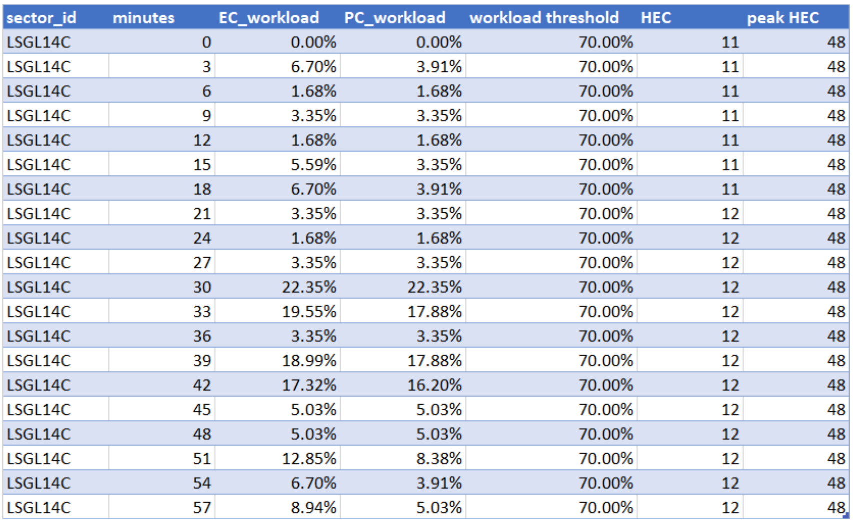
Hene, although the sectors have a HEC of half or even less than the sustain HEC value, due to the high complexity between 05:21 and 05:24, the sector configuration had to be chosen such that $LSGL5C$ was isolated from the others.

Figures 4.12 and 4.11 are based on Tables 4.9, 4.11 and 4.10, which represent the minutes between 5 a.m. and 6 a.m., the Executive Controller Workload (EC) ad the Planning Controller (PC) workload in percentage, the workload threshold with is constant and same for both controllers, the HEC and the HEC capacity.Each row, corresponds to a time interval of 3 minutes, in which the minute represented is the $t_{init}$. The minute 0 is associated with 05:00, and minute 3 is associated with 05:03.

Those Tables confirm that HEC values are well below the sustain value but due to complexity peak on sector $LSGL5C$, the best suited sector configuration is $U3K$. Table 4.9 confirms the highest peak at $LSGL14C$ happens at [05:30, 05:33[ with complexity 22.35% for both the EC and the PC. The higher peaks at sector $LSGL67C$, presented in Table 4.10, happens at [05:42, 5:48[. Table 4.11 confirms a very high peak, at $LSGL5C$, for the EC at [05:21, 05:24[ even though PC workload stays very much below safety levels (41%). This discrepancy calls the attention that there may be a Conflict. The next highest peak for the EC is at [05:24, 05:27[ but the complexity (65%) is half of the first and below the safety threshold.

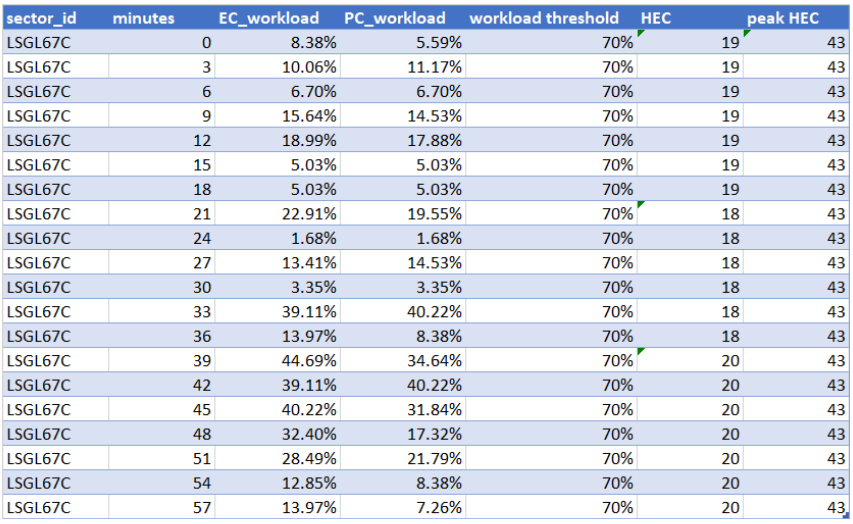
Appendix A.3 presents the level of detail this algorithm can go, which helps explaining hotspots

Table 4.9: HEC and Complexity HEC and LSGL14C for the time interval 5 a.m. to 6 a.m.



| sector_id | minutes | EC_workload | PC_workload | workload threshold | HEC | peak HEC |
|---|---|---|---|---|---|---|
| LSGL14C | 0 | 0.00% | 0.00% | 70.00% | 11 | 48 |
| LSGL14C | 3 | 6.70% | 3.91% | 70.00% | 11 | 48 |
| LSGL14C | 6 | 1.68% | 1.68% | 70.00% | 11 | 48 |
| LSGL14C | 9 | 3.35% | 3.35% | 70.00% | 11 | 48 |
| LSGL14C | 12 | 1.68% | 1.68% | 70.00% | 11 | 48 |
| LSGL14C | 15 | 5.59% | 3.35% | 70.00% | 11 | 48 |
| LSGL14C | 18 | 6.70% | 3.91% | 70.00% | 11 | 48 |
| LSGL14C | 21 | 3.35% | 3.35% | 70.00% | 12 | 48 |
| LSGL14C | 24 | 1.68% | 1.68% | 70.00% | 12 | 48 |
| LSGL14C | 27 | 3.35% | 3.35% | 70.00% | 12 | 48 |
| LSGL14C | 30 | 22.35% | 22.35% | 70.00% | 12 | 48 |
| LSGL14C | 33 | 19.55% | 17.88% | 70.00% | 12 | 48 |
| LSGL14C | 36 | 3.35% | 3.35% | 70.00% | 12 | 48 |
| LSGL14C | 39 | 18.99% | 17.88% | 70.00% | 12 | 48 |
| LSGL14C | 42 | 17.32% | 16.20% | 70.00% | 12 | 48 |
| LSGL14C | 45 | 5.03% | 5.03% | 70.00% | 12 | 48 |
| LSGL14C | 48 | 5.03% | 5.03% | 70.00% | 12 | 48 |
| LSGL14C | 51 | 12.85% | 8.38% | 70.00% | 12 | 48 |
| LSGL14C | 54 | 6.70% | 3.91% | 70.00% | 12 | 48 |
| LSGL14C | 57 | 8.94% | 5.03% | 70.00% | 12 | 48 |

Table 4.10: HEC and complexity LSGL67C for the time interval 5 a.m. to 6 a.m.



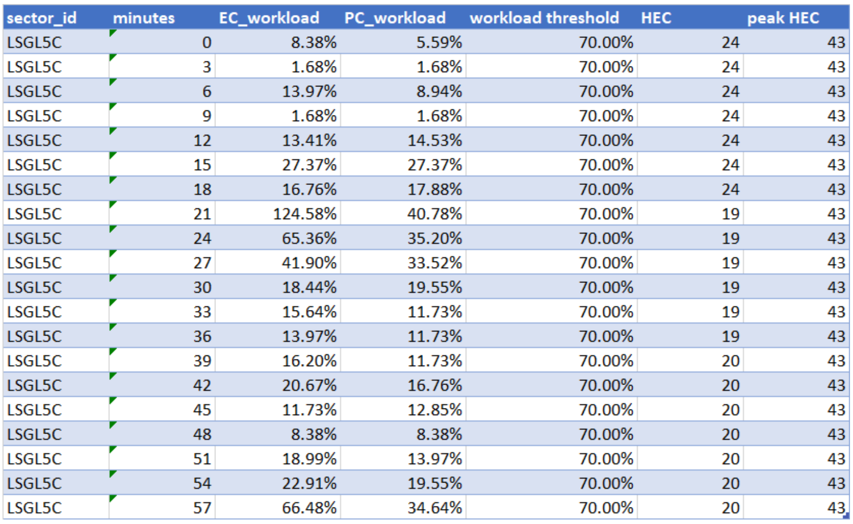
| sector_id | minutes | EC_workload | PC_workload | workload threshold | HEC | peak HEC |
|---|---|---|---|---|---|---|
| LSGL67C | 0 | 8.38% | 5.59% | 70% | 19 | 43 |
| LSGL67C | 3 | 10.06% | 11.17% | 70% | 19 | 43 |
| LSGL67C | 6 | 6.70% | 6.70% | 70% | 19 | 43 |
| LSGL67C | 9 | 15.64% | 14.53% | 70% | 19 | 43 |
| LSGL67C | 12 | 18.99% | 17.88% | 70% | 19 | 43 |
| LSGL67C | 15 | 5.03% | 5.03% | 70% | 19 | 43 |
| LSGL67C | 18 | 5.03% | 5.03% | 70% | 19 | 43 |
| LSGL67C | 21 | 22.91% | 19.55% | 70% | 18 | 43 |
| LSGL67C | 24 | 1.68% | 1.68% | 70% | 18 | 43 |
| LSGL67C | 27 | 13.41% | 14.53% | 70% | 18 | 43 |
| LSGL67C | 30 | 3.35% | 3.35% | 70% | 18 | 43 |
| LSGL67C | 33 | 39.11% | 40.22% | 70% | 18 | 43 |
| LSGL67C | 36 | 13.97% | 8.38% | 70% | 18 | 43 |
| LSGL67C | 39 | 44.69% | 34.64% | 70% | 20 | 43 |
| LSGL67C | 42 | 39.11% | 40.22% | 70% | 20 | 43 |
| LSGL67C | 45 | 40.22% | 31.84% | 70% | 20 | 43 |
| LSGL67C | 48 | 32.40% | 17.32% | 70% | 20 | 43 |
| LSGL67C | 51 | 28.49% | 21.79% | 70% | 20 | 43 |
| LSGL67C | 54 | 12.85% | 8.38% | 70% | 20 | 43 |
| LSGL67C | 57 | 13.97% | 7.26% | 70% | 20 | 43 |

(peaks) of workload. For instance, the excess of workload at [5:21, 5:24[, i.e., epoch interval [1533100860, 1533101040[ is caused by conflict supervisions and interventions, and by close to sector boundary conflicts. The epoch, referred to in the third column at Table in Appendix A.3, is relative to the universal epoch which started on $1^{st}$ January 1970.

As one can see, there are other time intervals with a significant amount of tasks to be performed. But, the fact that it is not caused by conflicts (just monitoring, entries and exits), reduces the workload to 41.9% for the EC and 33.52% for the PC, at epoch interval 1533101220 to 1533101400 (05:27 to 05:30 $1^{st}$ August 2018).

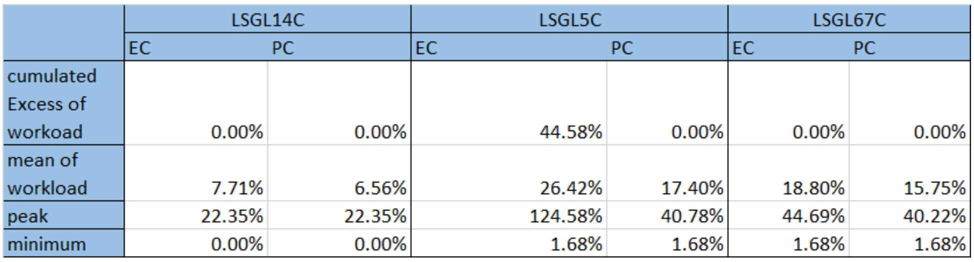
Table 4.12 is a summary of some of the most important values of sectors from $U3K$, as a guideline to ATCOs. Even though $LSGL5C$ has a high peak for the EC, the mean of workload is 26.42%. The sector configuration with lower peak value is at $LSGL14C$, with 22.35% for both controllers.

Table 4.11: HEC and complexity values LSGL5C for the time interval 5 a.m. to 6 a.m.



| sector_id | minutes | EC_workload | PC_workload | workload threshold | HEC | peak HEC |
|---|---|---|---|---|---|---|
| LSGL5C | 0 | 8.38% | 5.59% | 70.00% | 24 | 43 |
| LSGL5C | 3 | 1.68% | 1.68% | 70.00% | 24 | 43 |
| LSGL5C | 6 | 13.97% | 8.94% | 70.00% | 24 | 43 |
| LSGL5C | 9 | 1.68% | 1.68% | 70.00% | 24 | 43 |
| LSGL5C | 12 | 13.41% | 14.53% | 70.00% | 24 | 43 |
| LSGL5C | 15 | 27.37% | 27.37% | 70.00% | 24 | 43 |
| LSGL5C | 18 | 16.76% | 17.88% | 70.00% | 24 | 43 |
| LSGL5C | 21 | 124.58% | 40.78% | 70.00% | 19 | 43 |
| LSGL5C | 24 | 65.36% | 35.20% | 70.00% | 19 | 43 |
| LSGL5C | 27 | 41.90% | 33.52% | 70.00% | 19 | 43 |
| LSGL5C | 30 | 18.44% | 19.55% | 70.00% | 19 | 43 |
| LSGL5C | 33 | 15.64% | 11.73% | 70.00% | 19 | 43 |
| LSGL5C | 36 | 13.97% | 11.73% | 70.00% | 19 | 43 |
| LSGL5C | 39 | 16.20% | 11.73% | 70.00% | 20 | 43 |
| LSGL5C | 42 | 20.67% | 16.76% | 70.00% | 20 | 43 |
| LSGL5C | 45 | 11.73% | 12.85% | 70.00% | 20 | 43 |
| LSGL5C | 48 | 8.38% | 8.38% | 70.00% | 20 | 43 |
| LSGL5C | 51 | 18.99% | 13.97% | 70.00% | 20 | 43 |
| LSGL5C | 54 | 22.91% | 19.55% | 70.00% | 20 | 43 |
| LSGL5C | 57 | 66.48% | 34.64% | 70.00% | 20 | 43 |

Table 4.12: Some important statistic data for the suggested sector configuration at time interval 5 a.m. to 6 a.m.



| | LSGL14C | | LSGL5C | | LSGL67C | |
|---|---|---|---|---|---|---|
| | EC | PC | EC | PC | EC | PC |
| cumulated Excess of workoad | 0.00% | 0.00% | 44.58% | 0.00% | 0.00% | 0.00% |
| mean of workload | 7.71% | 6.56% | 26.42% | 17.40% | 18.80% | 15.75% |
| peak | 22.35% | 22.35% | 124.58% | 40.78% | 44.69% | 40.22% |
| minimum | 0.00% | 0.00% | 1.68% | 1.68% | 1.68% | 1.68% |

### 9 a.m. - 10 a.m. period

The optimal sector configuration, between 9 a.m. and 10 a.m., suggested by the complexity estimation algorithm is $U4J$ (see Figure 4.13). This time interval has one more open sector than the optimal sector configuration obtained between 5 a.m. and 6 a.m., which corresponds to the division of $LSGL67C$ into $LSGL6C$ and $LSGL7C$. The analysis of the graphs at Figures 4.14 and 4.15 will explain the algorithm's suggestion.

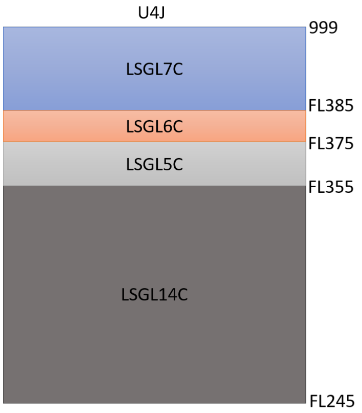


Figure 4.13: simple representation of sector configuration $U4J$.

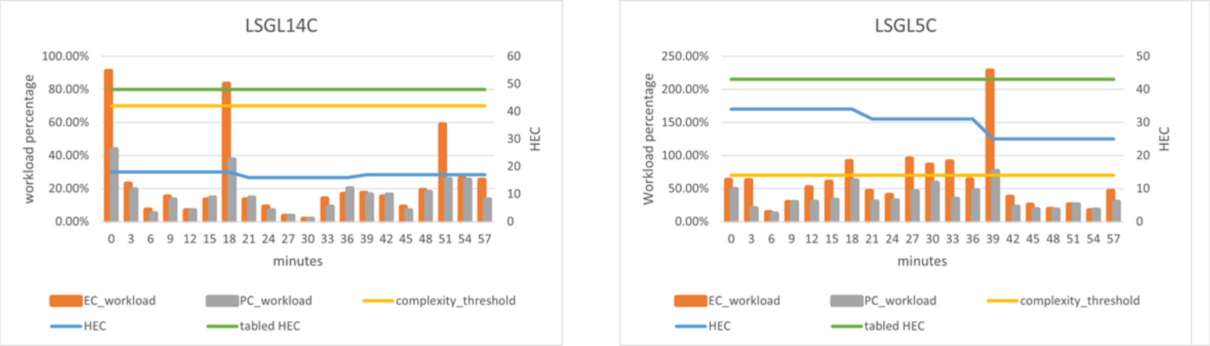


Figure 4.14: HEC and controller workload at sectors $LSGL14C$ and $LSGL5C$ between 9 a.m. and 10 a.m.

$LSGL14C$ has two time intervals which overcome the safety threshold of 70% workload for the EC. This happens at the first three minutes (91%) and between 9:18 and 9:21 (83%). The HEC are very below the HEC accepted value (48), with HEC between 16 and 18. In contrast, $LSGL5C$ has a very high peak, mostly due to conflicts. The most advisable action to do would be to re-route the conflicting flights. A study of this high peak will be performed in section 4.2.3. Nonetheless, the other complexity values are all below one hundred, with five peaks that overcome safety threshold. However, this sector cannot be divided into more, it is the smallest possible. According to the reduced vertical separation minima (RVSM) [12] this sector takes on two flight levels where it is feasible to fly (FL360, FL370). The sector's HEC, with values between 34 and 25, is greater than at $LSGL4C$ but still below the sustain value of 43.

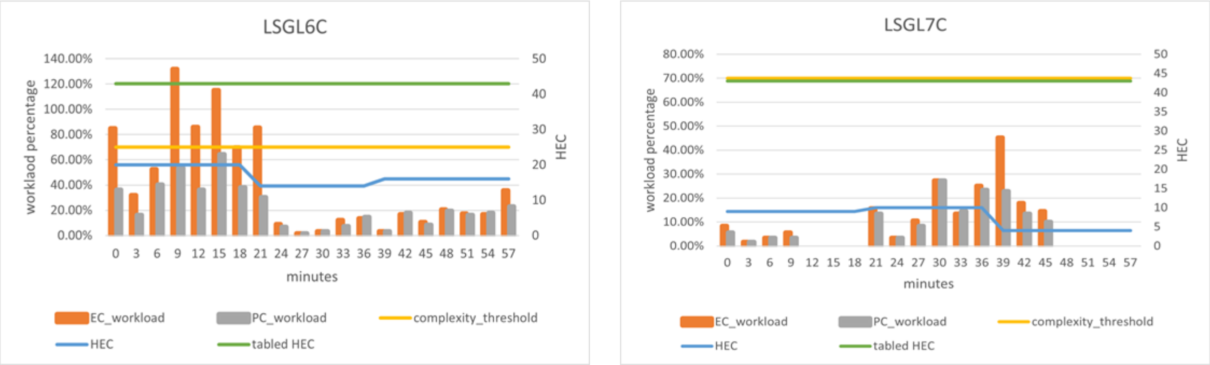


Figure 4.15: HEC and controller workload at sectors $LSGL6C$ and $LSGL7C$ between 9 a.m. and 10 a.m.

$LSGL6C$ has five peaks above safety threshold, where the one at 09:09 to 09:11 is the highest one, with 135% of workload for the EC. The hourly entry count however is lower than the one observed at $LSGL5C$ with HEC between 20 and 14. $LSGL7C$ does not have much complexity associated being the highest peak at 45% ( [09:39, 09:41[ ) for the EC all the others are below 30%.

One might ask why is $LSGL14C$ considered instead of using two sectors to avoid the two peaks above the threshold. As Figure 4.16 shows, apart from $LSGL4C$, the other sectors do not have considerable amounts of workload and sector $LSGL1C$ has no workload. The separation into one only sector $LSGL4C$ does not eliminate or reduce the highest peak shown at Figure 4.14 even though the other peak (83% at 09:18 to 09:21) does get below threshold at 29%, but the algorithm considers it is better to still join the four lower sectors (the excess of complexity stays bellow 20% threshold for sector configuration $U4J$ compared with the sector configuration with five opened sectors with best_delta - Equations 3.15 and 3.16).

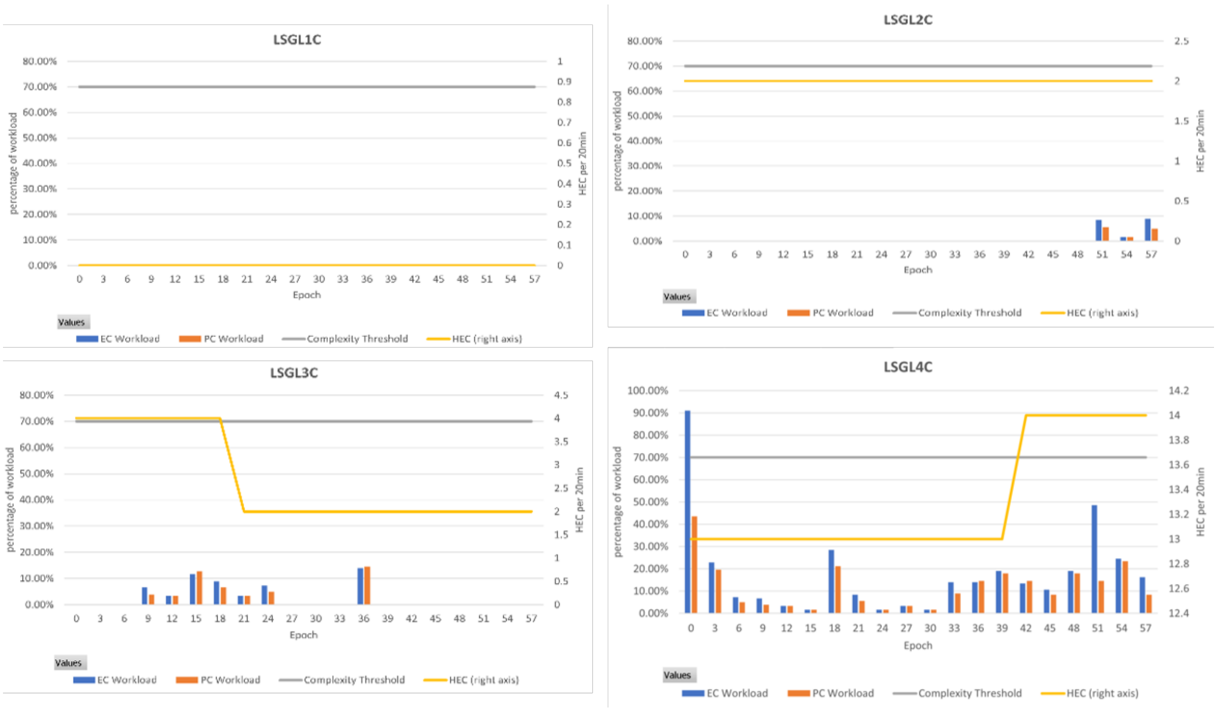


Figure 4.16: Separate analysis of sectors LSGL1C, LSGL2C, LSGL3C and LSGL4C.

Table 4.13 summarizes the main complexity values and corresponding average values, as well as the mean of HEC for the $U4J$ sectors, between 09 a.m. and 10 a.m.. Even though LSGL5C has a very high peak, the mean of workload for the EC is 60%, and is also the sector with higher mean.$LSGL7C$ is the only sector without cumulative excess of workload and the one with lower HEC mean.

Table 4.13: Summary of the main values taken from analysis of sectors part of configuration $U4J$.

| | LSGL14C | | LSGL5C | | LSGL6C | | LSGL7C | |
|---|---|---|---|---|---|---|---|---|
| | EC_workload | PC_workload | EC_workload | PC_workload | EC_workload | PC_workload | EC_workload | PC_workload |
| cumulated Excess of workoad | 34.30% | 0% | 240.50% | 6.57% | 153.35% | 0% | 0% | 0% |
| mean of workload | 23.41% | 15.84% | 59.55% | 34.53% | 40.92% | 22.74% | 9.61% | 7.54% |
| peak | 91.06% | 43.58% | 227.93% | 64.25% | 131.84% | 64.25% | 45.25% | 27.37% |
| minimum | 1.68% | 1.68% | 13.97% | 11.73% | 1.68% | 1.68% | 0.00% | 0.00% |
| mean HEC | 17.05 | | 29.95 | | 16.8 | | 7.55 | |

### 4.2.2 Example Complexity by Flight

The complexity estimation through CAPAN parameters allows the estimation of complexity given by each individual flight, which may help decide which flight would be better to re-route. To understand the level of detail this algorithm undergoes and the tasks assign to the flights, two examples will be presented, flight DLH47E and flight EWG1YF. The first flight (DLH47E) has not much complexity associated to conflicting tasks, but comes in Geneva's ACC climbing, this will demand extra taskload accompanied with the normal tasks. The second flight (EWG1YF) already presents conflicting tasks, which will demand higher controller workload.

### Flight DLH47E

Figure 4.17 shows the trajectory flight DLH47E performs, going from South France to Germany. Accompanied with Table 4.14, it is observed the the flight enters at sector $LSGL3C$ in climb from sector's lower bound, demanding tasks 1, 2 and 10, as any entering flight from outbound would demand, with two more extra tasks (task 16 and task 13) associated to entering from lower bound (from another ACC) and in climb, respectively. The, flight does not stays in sector $LSGL3C$ for long. Fifty seconds later, it comes in $LSGL4C$, now from inbound (same ACC), which demands tasks 1, 4 and 11, as well as tasks 14 and 8, because the flight corresponds to an atypical route and enters from lower bound at same ACC, respectively. When it reaches a specific flight level it changes to cruise (task 17) since the controller has to communicate with the aircraft's pilot saying that it reached the flight level given at the flight's plan. It stays in sector $LSGL4C$ long enough to be monitored four times, until it leaves Geneva's ACC heading South France (tasks 12, 6 and 3).

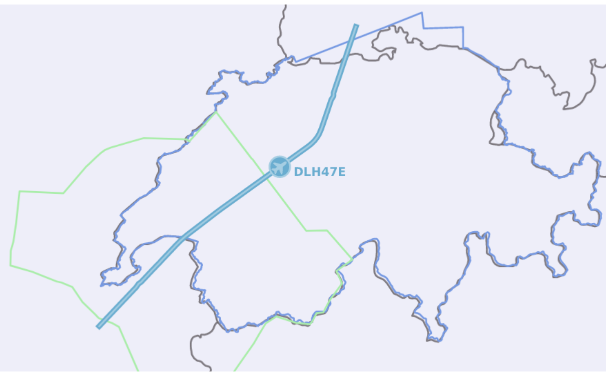
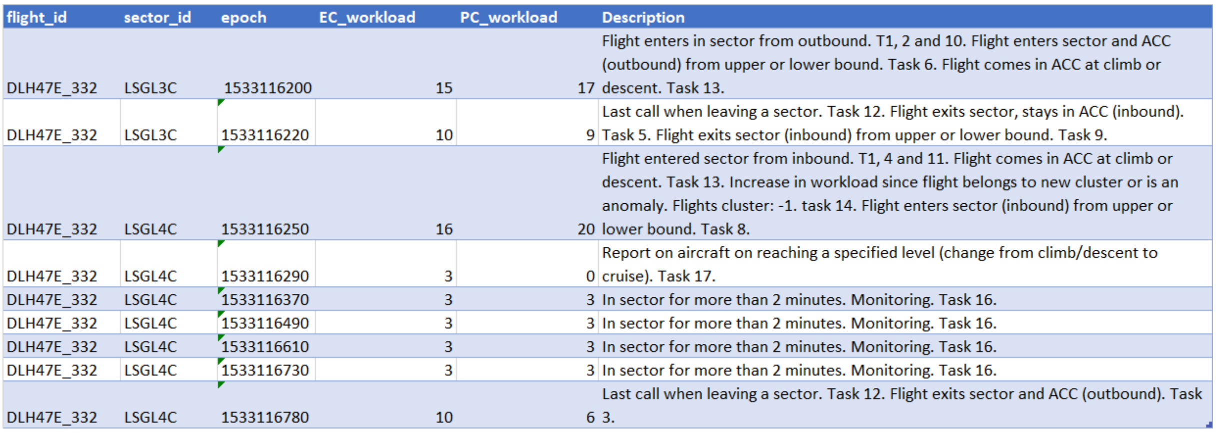


Figure 4.17: Flight DLH47E crossing Swiss Upper-airspace

Table 4.14: DLH47E_332 flight in Geneva's ACC entering at 09:36:40 in LSGL3C and leaving ACC at 09:46:20 from sector LSGL4C



| flight_id | sector_id | epoch | EC_workload | PC_workload | Description |
|---|---|---|---|---|---|
| DLH47E_332 | LSGL3C | 1533116200 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Flight enters sector and ACC (outbound) from upper or lower bound. Task 6. Flight comes in ACC at climb or descent. Task 13. |
| DLH47E_332 | LSGL3C | 1533116220 | 10 | 9 | Last call when leaving a sector. Task 12. Flight exits sector, stays in ACC (inbound). Task 5. Flight exits sector (inbound) from upper or lower bound. Task 9. |
| DLH47E_332 | LSGL4C | 1533116250 | 16 | 20 | Flight entered sector from inbound. T1, 4 and 11. Flight comes in ACC at climb or descent. Task 13. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14. Flight enters sector (inbound) from upper or lower bound. Task 8. |
| DLH47E_332 | LSGL4C | 1533116290 | 3 | 0 | Report on aircraft on reaching a specified level (change from climb/descent to cruise). Task 17. |
| DLH47E_332 | LSGL4C | 1533116370 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| DLH47E_332 | LSGL4C | 1533116490 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| DLH47E_332 | LSGL4C | 1533116610 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| DLH47E_332 | LSGL4C | 1533116730 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| DLH47E_332 | LSGL4C | 1533116780 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |

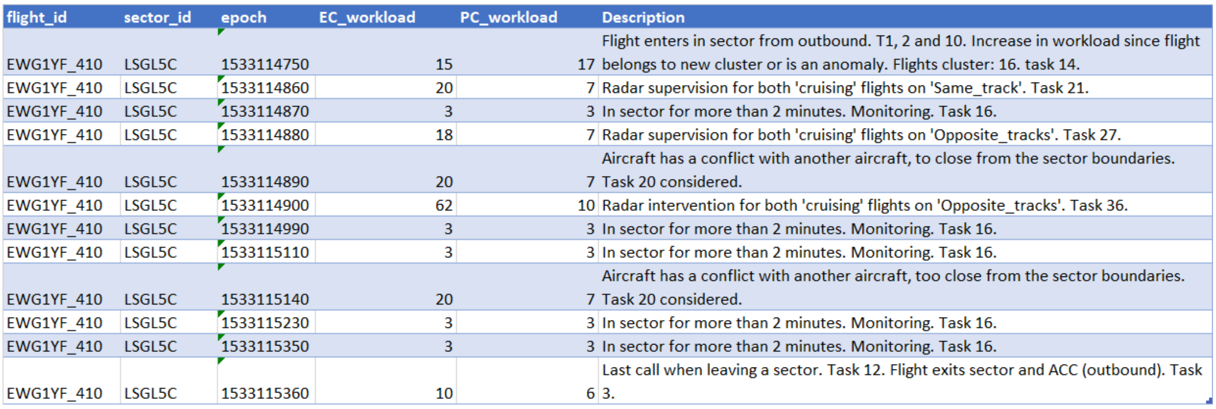
Throughout almost 10 minutes in Geneva's ACC a number of tasks need to be performed demanding a total of 66 seconds from EC and 64 seconds from PC and 10 different tasks. [1]

### Flight EWG1YF

The flight presented above did not have any problems of conflict, even though it was an interesting example due to the change of flight phase that could be analysed, and the tasks associated.

FLight EWG1YF already has some conflicts that are interesting to analyse. Table 4.15 shows the diversity of tasks inherent to this flight, demanding a total of 180 seconds workload from EC and 76 seconds from PC. While this flight crosses Geneva's ACC, it does not change sector, staying at $LSGL5C$, and keeping its cruise phase.

Table 4.15: EWG1YF flight entering Geneva's Airspace at 09:12:30 and exiting at 09:22:40, tasks and description.



| flight_id | sector_id | epoch | EC_workload | PC_workload | Description |
|---|---|---|---|---|---|
| EWG1YF_410 | LSGL5C | 1533114750 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 16. task 14. |
| EWG1YF_410 | LSGL5C | 1533114860 | 20 | 7 | Radar supervision for both 'cruising' flights on 'Same_track'. Task 21. |
| EWG1YF_410 | LSGL5C | 1533114870 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| EWG1YF_410 | LSGL5C | 1533114880 | 18 | 7 | Radar supervision for both 'cruising' flights on 'Opposite_tracks'. Task 27. |
| EWG1YF_410 | LSGL5C | 1533114890 | 20 | 7 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. |
| EWG1YF_410 | LSGL5C | 1533114900 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| EWG1YF_410 | LSGL5C | 1533114990 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| EWG1YF_410 | LSGL5C | 1533115110 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| EWG1YF_410 | LSGL5C | 1533115140 | 20 | 7 | Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered. |
| EWG1YF_410 | LSGL5C | 1533115230 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| EWG1YF_410 | LSGL5C | 1533115350 | 3 | 3 | In sector for more than 2 minutes. Monitoring. Task 16. |
| EWG1YF_410 | LSGL5C | 1533115360 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |

In this situation there is a large discrepancy between the EC workload and PC workload. This is mainly because of conflicts, intervention and supervisions, which demand a lot more time and responsibility from the EC. There are two types of supervision conflicts associated to this flight and one intervention. One, is supervision between cruising flights at same track (task 21), the other is supervision between cruising flights on opposite tracks (task 27), there is also an addition of workload associated to the fact that one of the flights' samples in conflict is at less than 10NM from sector boundary. There is also a radar intervention (task 36). The radar interventions do take more controller effort than the supervisions.

It is interesting, because during the study of complexity indicators (at the Background section) there were studies that considered aircraft flying a certain sector for a short period of time was also an indicator of complexity. However, when a flight stays for longer it has to be checked every 2 minutes, as shown in this case.

### 4.2.3 Example Conflict analysis

At section 4.2.1 one hotspot of 125% was identified at sector $LSGl5C$ between 05:21 and 05:24 and another hotspot was identified at same sector but higher (230%) between 09:39 and 09:42. During these two hotspots conflicts were identified, which may be the cause of such peaks.

[1] The epoch in third column is related to the Universal epoch which started 1$^{st}$ January 1970.

### Conflict between 05:23 and 5:27

This conflict corresponds to the peak at $LSGL5C$, illustrated in Figure 4.12, and takes the the three flights depicted in Figure 4.18.

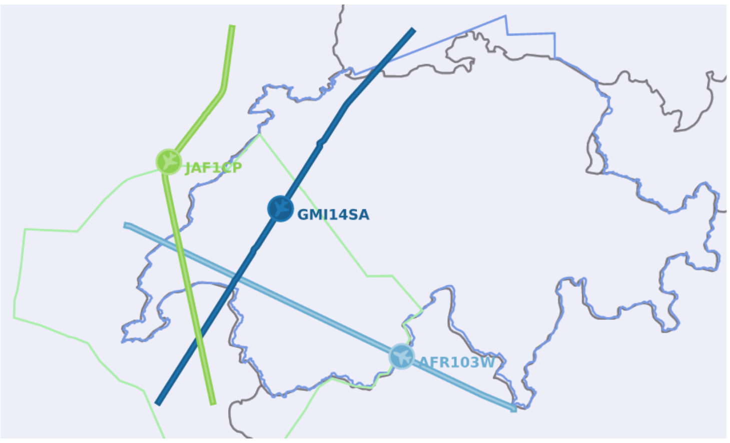


Figure 4.18: Flights JAF1CP, GMI14SA, AFR103W which present conflicts between each other.

As Table 4.16 confirms, this example presents some supervision conflicts. Two are from type *crossing tracks* (JAF1CP, AFR103W) and (GMI14SA, AFR103W) while the other type is from *Same Track* (between JAF1CP and GMI14SA) which are at a heading difference $\leq 45^o$. The three flight are cruising. Pair (GMI14SA, AFR103W) calls of an intervention conflict and the conflict takes place close to sector boundaries. Then, pair (JAF1CP, AFR103W) suffers a supervision conflict also close to sector. Ten seconds later at 1533101020, JAF1CP and GMI14SA present a supervision conflict, requiring another an intervention and again, couple of seconds later, a supervision. During this event, there is again a distance of supervision between GMI14SA and AFR103W.

The distances selected at the experimental results, for supervision and intervention between cruising flights at same track is 15NM and 10NM, respectively; between crossing tracks (both cruising) is 18NM and 12 NM, respectively.

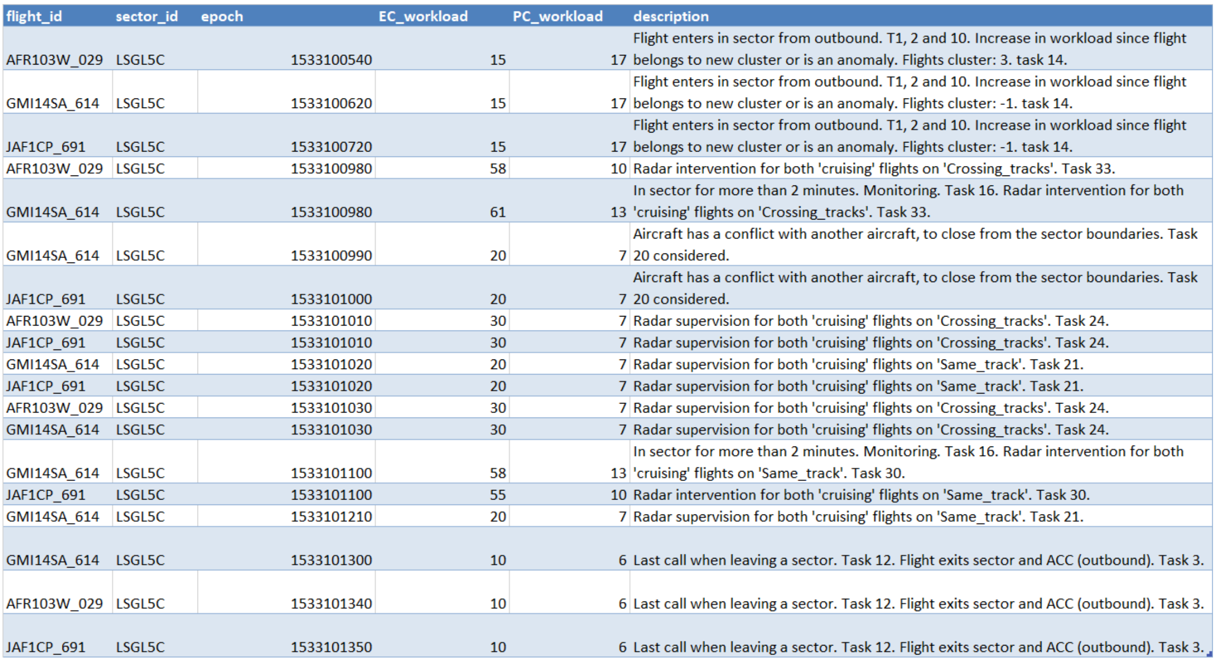
### Conflict from 09:39 to 09:41

Another conflict between three flights was found during time interval between 9 a.m. and 10 a.m., illustrated in Figure 4.19, this coincides with the high peak at $LSGL5C$ (Figure 4.14) because it takes place at time interval 09:39 to 09:40.

In Figure 4.19, it can be seen two aircraft with similar headings and one aircraft flying opposite to them at very close time intervals. Table 4.17 details the tasks associated to these conflicts. Many times, the intervention and the supervision tasks between 'opposite_tracks' are called and also the fact that the conflict happens close to sector boundary, calls task 20 a considerable amount of times, increasing complexity. Since Table 4.17 does not show conflicts between aircraft at same track, there are no conflicts between the two aircraft heading south. This is explained by the entry of one being 3 minutes (180seconds) before, and at exit they are still separated by 2 minutes and 50 seconds.

First, the conflict happens between EWG8HZ and RYR34UU, the flight coming from North which

Table 4.16: Tasks assigned to conflict and entries and exits of flights that are part of this conflict. Other tasks such as monitoring every 2 minutes are not here considered, the focus is at the conflict tasks.



| flight_id | sector_id | epoch | EC_workload | PC_workload | description |
|---|---|---|---|---|---|
| AFR103W_029 | LSGL5C | 1533100540 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 3. task 14. |
| GMI14SA_614 | LSGL5C | 1533100620 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14. |
| JAF1CP_691 | LSGL5C | 1533100720 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14. |
| AFR103W_029 | LSGL5C | 1533100980 | 58 | 10 | Radar intervention for both 'cruising' flights on 'Crossing_tracks'. Task 33. |
| GMI14SA_614 | LSGL5C | 1533100980 | 61 | 13 | In sector for more than 2 minutes. Monitoring. Task 16. Radar intervention for both 'cruising' flights on 'Crossing_tracks'. Task 33. |
| GMI14SA_614 | LSGL5C | 1533100990 | 20 | 7 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. |
| JAF1CP_691 | LSGL5C | 1533101000 | 20 | 7 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. |
| AFR103W_029 | LSGL5C | 1533101010 | 30 | 7 | Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24. |
| JAF1CP_691 | LSGL5C | 1533101010 | 30 | 7 | Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24. |
| GMI14SA_614 | LSGL5C | 1533101020 | 20 | 7 | Radar supervision for both 'cruising' flights on 'Same_track'. Task 21. |
| JAF1CP_691 | LSGL5C | 1533101020 | 20 | 7 | Radar supervision for both 'cruising' flights on 'Same_track'. Task 21. |
| AFR103W_029 | LSGL5C | 1533101030 | 30 | 7 | Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24. |
| GMI14SA_614 | LSGL5C | 1533101030 | 30 | 7 | Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24. |
| GMI14SA_614 | LSGL5C | 1533101100 | 58 | 13 | In sector for more than 2 minutes. Monitoring. Task 16. Radar intervention for both 'cruising' flights on 'Same_track'. Task 30. |
| JAF1CP_691 | LSGL5C | 1533101100 | 55 | 10 | Radar intervention for both 'cruising' flights on 'Same_track'. Task 30. |
| GMI14SA_614 | LSGL5C | 1533101210 | 20 | 7 | Radar supervision for both 'cruising' flights on 'Same_track'. Task 21. |
| GMI14SA_614 | LSGL5C | 1533101300 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |
| AFR103W_029 | LSGL5C | 1533101340 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |
| JAF1CP_691 | LSGL5C | 1533101350 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |

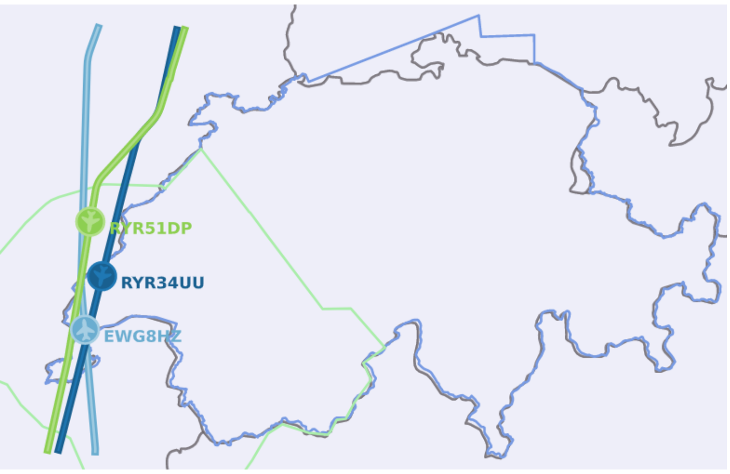


Figure 4.19: Flights R51DP, RYR34UU, EWG8HZ which have 4D trajectories very close to one another. It would be advisable that flight $EWG8HZ$ suffered a level-capping or re-routing.

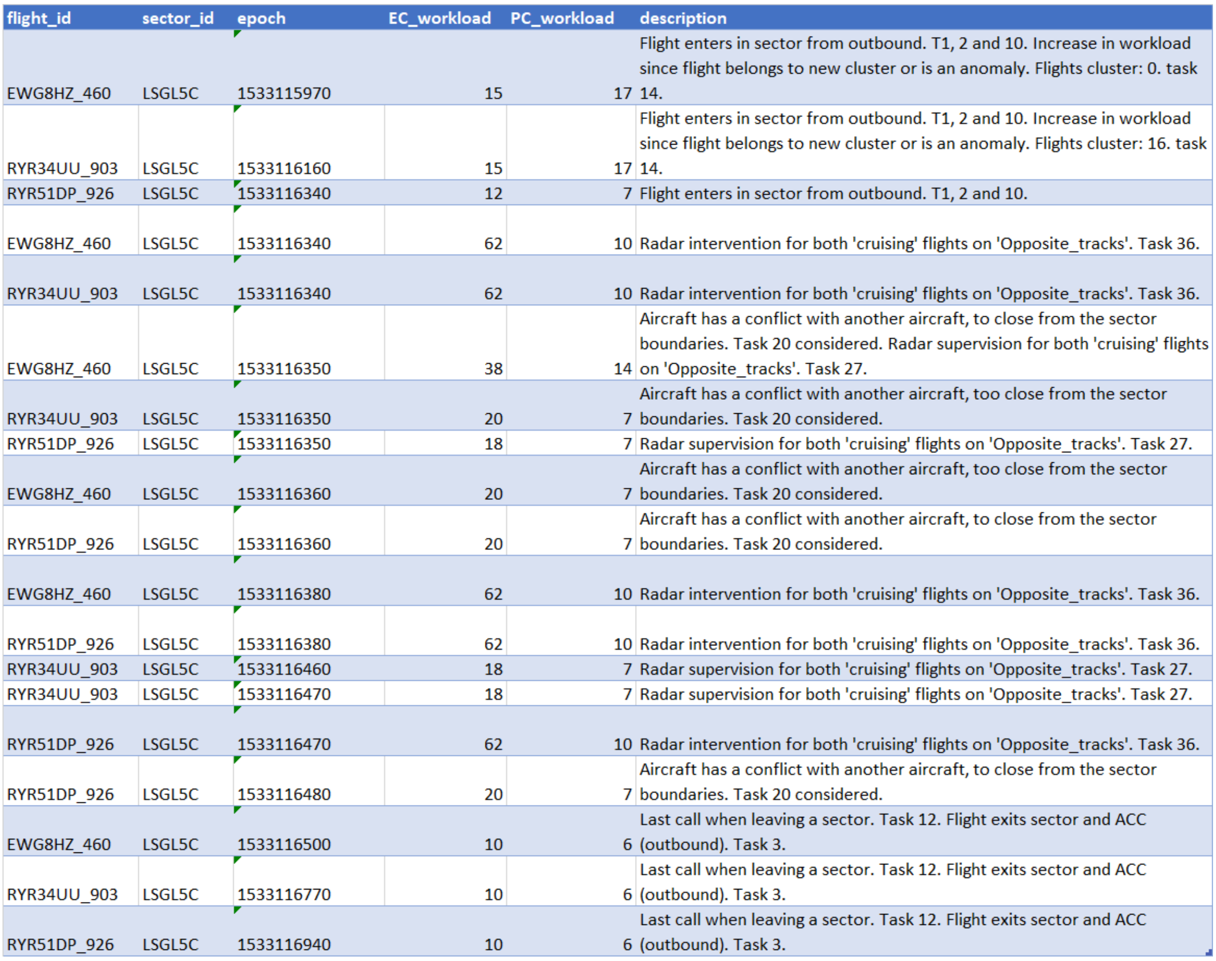
enters first. In the beginning, it is observed an intervention, followed by a supervision (close to sector boundaries). At the same time instant (epoch 1533116350) EWG8HZ starts a supervision conflict with RYR51DP (where one of the flights is still close to sector boundary) followed by a radar intervention between EWG8HZ and RYR51DP.

Then, RYR34UU has another supervision with a flight not identified in this conflict (153316460 and 1533116470). It is not with RYR51DP because, if they had a conflict, it would be of the same track type of conflict, as said before, and because aircraft RYR51DP does not present any conflict at 153316460. RYR51DP does present another conflict at epoch 1533116470 but still not with RYR34UU, because it is a different task, it is an intervention (task 36).

If a level capping of aircraft EWG8HZ was performed to sector four ($LSGL4C$) this identified conflict would vanish and the high peak could be avoided, which corresponds to the hotspot (230%) depicted at Figure 4.14 on the complexity associated to sector $LSGL5C$. Looking at the complexity analysis of sector $LSGL4C$, at the time interval of analysis (minute = 39), sector four could continue below safety threshold if there were not any conflicts with the flights already there (right now both EC and PC workloads are below 20%).

Summing the complexity inherent only to the conflicting tasks between pairs (EWG8HZ, RYR51DP) and (EWG8HZ, RYR34UU) and dividing by the *time_workload_resolution* (see Equation 3.13), it is concluded that a demand of 202% EC workload and of 49.4% PC workload come from the conflict of the identified two crossing pairs.

Table 4.17: Data from flights EWG8HZ, RYR34UU, RYR51DP extracted during conflict



| flight_id | sector_id | epoch | EC_workload | PC_workload | description |
|---|---|---|---|---|---|
| EWG8HZ_460 | LSGL5C | 1533115970 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 0. task 14. |
| RYR34UU_903 | LSGL5C | 1533116160 | 15 | 17 | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 16. task 14. |
| RYR51DP_926 | LSGL5C | 1533116340 | 12 | 7 | Flight enters in sector from outbound. T1, 2 and 10. |
| EWG8HZ_460 | LSGL5C | 1533116340 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| RYR34UU_903 | LSGL5C | 1533116340 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| EWG8HZ_460 | LSGL5C | 1533116350 | 38 | 14 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. Radar supervision for both 'cruising' flights on 'Opposite_tracks'. Task 27. |
| RYR34UU_903 | LSGL5C | 1533116350 | 20 | 7 | Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered. |
| RYR51DP_926 | LSGL5C | 1533116350 | 18 | 7 | Radar supervision for both 'cruising' flights on 'Opposite_tracks'. Task 27. |
| EWG8HZ_460 | LSGL5C | 1533116360 | 20 | 7 | Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered. |
| RYR51DP_926 | LSGL5C | 1533116360 | 20 | 7 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. |
| EWG8HZ_460 | LSGL5C | 1533116380 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| RYR51DP_926 | LSGL5C | 1533116380 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| RYR34UU_903 | LSGL5C | 1533116460 | 18 | 7 | Radar supervision for both 'cruising' flights on 'Opposite_tracks'. Task 27. |
| RYR34UU_903 | LSGL5C | 1533116470 | 18 | 7 | Radar supervision for both 'cruising' flights on 'Opposite_tracks'. Task 27. |
| RYR51DP_926 | LSGL5C | 1533116470 | 62 | 10 | Radar intervention for both 'cruising' flights on 'Opposite_tracks'. Task 36. |
| RYR51DP_926 | LSGL5C | 1533116480 | 20 | 7 | Aircraft has a conflict with another aircraft, to close from the sector boundaries. Task 20 considered. |
| EWG8HZ_460 | LSGL5C | 1533116500 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |
| RYR34UU_903 | LSGL5C | 1533116770 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |
| RYR51DP_926 | LSGL5C | 1533116940 | 10 | 6 | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. |

The monitoring (task 6) was eliminated from Table 4.17 to focus more on conflict tasks.

# Chapter 5

# Conclusions

Air traffic control is usually a demanding and stressful job, since air traffic controllers are responsible for thousands of passengers. To support their tasks, computer-aided systems were developed, in order to help them: identify typical aircraft routes and airspace volume spots most prompted to risk (intersection of typical routes); have a better understanding and awareness of the airspace where FMPs and ATCOs work (most importantly now that Free Route Airspace is implemented in many, many countries) and consequently better management; estimate air traffic complexity more accurately than the method currently used, and predict time periods of excess of workload for the ATCOs and, therefore, automatically provide the best sector configuration for a given time interval.

## 5.1 Achievements

According to the presented results, OPTICS with PCA 2, is the best clustering technique for the studied case, followed by DBSCAN with t-SNE. The fact that DBSCAN occupies the second and third places is very surprising. It was expected R-DBSCAN, HDBSCAN and OPTICS ranked before DBSCAN, because of their adaptive nature. However, DBSCAN could have surpassed at this data analysis with the default value epsilon=0.5, defined at scikit learn, if the typical routes identification was performed at other airspaces, perhaps the hierarchical method would have overcome the density-based clustering, this is why the comparison method is still suggest since results may vary from air-data to air-data. Another fact is that the best pre-processing algorithms are PCA 2 and t-SNE, occupying the first nine places in ranking. An interesting observation is that DBSCAN may occupy the second and third places with t-SNE an PCA 2, but DBSCAN with PCA 6 is qualified as the worst among the tested.

Another expected observation, is that the clustering only works when there is a considerable amount of data, just like all the other machine learning mechanisms, the clustering results decrease its quality bellow 900 flight trajecories. If the person in charge wants to analyse the typical routes of a small set of flights, the best practice is to apply Best-Fit-Cluster. The Best-fit-cluster is a fast algorithm conceived to use the centroids of the clustering proposed system (made from a great amount of data) to analyse flights that were not inserted in the clustering input.

Hence, combining Clustering (to analyse past data) with Best-Fit (to analyse the next x hours of flight plans) is a robust way to detect typical and atypical routes. It provides valuable knowledge to FMP and ATCOs, since it identifies spots more prompted to danger (conflicts, where two typical routes cross) and provides them a better knowledge an consequently management of the airspace where they work. Specially nowadays that FRA is implemented in most European countries.

Regarding Complexity Estimation, the proposed algorithm has very much potential in determining optimal sector configurations. When increasing the number of opened sectors does not improve complexity, it understands that the sector with excess of complexity is as divided as possible, or a greater sector compared to the possibility of it being split does not decrease that much the complexity (traffic concentrates in a specific airspace volume and the rest of volume has very low traffic demand), as it was observed at time interval 9 a.m. to 10 a.m..

The algorithm is robust and identifies a diversity of events and tasks such as different kinds of entries (inbound, outbound, upper/lower sector), exits, monitoring tasks, conflicts, etc. It does not only compute the workload in each designed sector but the complexity associated to each flight which is an added-value. With this information it is easy to subtract the complexity associated to this flight, and study hypothetical re-routes of the flight in question, such as level capping and heading change, and analyse the change in complexity and time to the sectors it may concern. This provides detailed information to the decision maker.

The identification of hotspots for controllers and such a detailed complexity analysis provides precision and anticipation of problems to both FMPs and ATCOs. Also, the recommendation of sector configurations are a very useful tool, specially to help the FMP decide the best sector configurations for the next day.

To conclude there must be pointed out that for the good results at complexity analysis and consequently optimal sector configurations three to six hours in advance, there is a high dependency with accurate trajectory predictions.

## 5.2 Future Work

This dissertation motivates some Future Work both at the Clustering side and at the the Complexity Estimation side.

At Clustering, the Vertical Evolution is still dependent on the time. The suggestion is to make it dependent on latitude and longitude with some degree limits and test. In fact, even though the dependence on time still gave very good correlations and results, it would be better that the dependence was spacial, and not time relative, because flights were identified flying at considerably different ground--speeds. Besides, it would be convenient to take advantage of the *applicability column* addressed to all trajectories to disconsider trajectories that were wrongly assign to clusters. Another application would be to eliminate the quality indicators TAC and TAVC and have another indicator measuring the reason of trajectories with *applicability value* 1 or 2 (section 3.1.3). Furthermore, it would be very interesting to perform clustering over one week and test the results to compare with the one performed at this dissertation. It would be very good to see the influence of much more data.

Concerning Complexity Estimation, there is a couple of future works to suggest. One would be the insertion of weather and restricted areas, that change the airspace volume capacity and trajectories usual routes. The last is to propose a study implementing neural networks or a linear correlator with a set of complexity indicators that learn, are tuned or correlated with the reference value given by the algorithm developed at this dissertation. Nine parameters are suggested:

1. Occupancy count;
2. fraction of main-flow flights;
3. Horizontal Interactions;
4. Vertical interactions;
5. Speed Interactions;
6. Aircraft close to sector borders;
7. Fraction of climbing/descending aircraft;
8. Fraction of outbound handoffs;
9. volume density.

Nonetheless, another approach is to build a complexity estimator based only on a subset of *CAPAN* tasks (those that had a higher correlation with the *reference value* this thesis tackles, and less correlation between each other). Then, perform a correlation analysis (linear or non-linear correlation) or teach an Artificial Intelligent (AI) (Neural Network (NN)) algorithm. Whatever application is chosen, those algorithms must be continuously tuned.

# Bibliography

[1] K. Schickel. Balancing sector volume workload by using air traffic controller click inputs. 2013.

[2] P. Debels. Ceats research development and simulation centre, budapest. https://slideplayer.com/slide/753330/. last assessed Dec 2020.

[3] H. Helmke, O. Ohneiser, J. Buxbaum, and C. Kern. Increasing atm efficiency with assistant based speech recognition. In *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, USA*, 2017.

[4] S. R. Proud. Go-around detection using crowd-sourced ads-b position data. *Aerospace*, 7(2):16, 2020.

[5] D. K. Schmidt. Queuing analysis of air traffic controller's workload. Technical report, 1978.

[6] M. W. Hurst and R. M. Rose. Objective job difficulty, behavioural response, and sector characteristics in air route traffic control centres. *Ergonomics*, 21(9):697–708, 1978.

[7] E. Stein. Air traffic controller workload: An examination of workload probe (report faa/cttn90/60). *Atlantic City, New Jersey: FAA*, 1985.

[8] F. of Transport and U. o. Z. Traffic Sciences. Sector complexity study - sesar 2020, July 2018. Study Comissioned by Croatia Control Ltd.

[9] P. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhodes. Airspace complexity measurement: An air traffic control simulation analysis. In *7th USA/Europe Air Traffic Management R&D Seminar*, pages 295–315, 2007.

[10] E. A. Group et al. Complexity metrics for ansp benchmarking analysis. *EUROCONTROL, April*, 2006.

[11] P. EUROCONTROL. Performance review report. *Performance Review Commission*, 2018.

[12] Reduced vertical separation minima (rvsm). URL `https://www.skybrary.aero/index.php/Reduced_Vertical_Separation_Minima_(RVSM)`.

[13] I. V. Laudeman, S. G. Shelden, R. Branstrom, and C. Brasil. Dynamic density: An air traffic management metric. 1998.

[14] R. H. Mogford, J. Guttman, S. Morrow, and P. Kopardekar. The complexity construct in air traffic control: A review and synthesis of the literature. Technical report, CTA INC MCKEE CITY NJ, 1995.

[15] B. Hilburn. Cognitive complexity in air traffic control: A literature review. *EEC note*, 4(04), 2004.

[16] N. Suarez, P. López, E. Puntero, and S. Rodriguez. Quantifying air traffic controller mental workload. *Fourth SESAR Innovation Days*, 220, 2014.

[17] P. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhodes. Airspace complexity measurement: An air traffic control simulation analysis. US/Europe 7th Air Traffic Management Seminar; Barcelona, Spain, July 2007.

[18] T. Radišić, D. Novak, and B. Juričić. Reduction of air traffic complexity using trajectory-based operations and validation of novel complexity indicators. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3038–3048, 2017.

[19] P. Andraši, T. Radišić, D. Novak, and B. Juričić. Subjective air traffic complexity estimation using artificial neural networks. *Promet-Traffic&Transportation*, 31(4):377–386, 2019.

[20] E. O. P. Raffaele Russo. Capan methodology sector capacity assessment. Air Traffic Services System Capacity Seminar/Workshop, June 2016. https://www.icao.int/ESAF/Documents/meetings/2016/CAPAN_Sector_Capacity_Assessment.pdf (available on June 2020).

[21] G. M. Flynn, C. Leleu, and B. Hilburn. A complexity study of the maastricht upper airspace centre. *EEC Report*, (403), 2006.

[22] skybrary. Rams plus. skybrary website, last edited 2017. last accessed November 2020.

[23] J. G. P. D. A. Coelho. Analysis of new arrival operational procedures in terminal airspace. 2016.

[24] J. Rodrigues. *Lisbon Airport Capacity Enhancement: Airspace Capacity Estimation and Enhancement*. PhD thesis, MS Thesis, Instituto Superior Técnico, Lisbon, 2014.

[25] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[26] G. Hamerly and C. Elkan. Learning the k in k-means. In *Advances in neural information processing systems*, pages 281–288, 2004.

[27] M. Gariel, A. N. Srivastava, and E. Feron. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1511–1524, 2011.

[28] C. E. V. Gallego, V. F. G. Comendador, F. J. S. Nieto, and M. G. Martinez. Discussion on density-based clustering methods applied for automated identification of airspace flows. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2018.

[29] A. Eckstein. Automated flight track taxonomy for measuring benefits from performance based navigation. In *2009 Integrated Communications, Navigation and Surveillance Conference*, pages 1–12. IEEE, 2009.

[30] M. Enriquez and C. Kurcz. A simple and robust flow detection algorithm based on spectral clustering. In *ICRAT Conference*, 2012.

[31] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[32] J. Almeida, L. Barbosa, A. Pais, and S. Formosinho. Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering. *Chemometrics and Intelligent Laboratory Systems*, 87(2):208–217, 2007.

[33] I. T. Jolliffe. Springer series in statistics. *Principal component analysis*, 29, 2002.

[34] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[35] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[36] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[37] A. Segarra Torne. Route clustering for strategic planning in air traffic management. Master's thesis, UC Irvine, 2015.

[38] T. Dubot. Clustering aircraft trajectories with recursive dbscan, October 2019. URL `https://github.com/thomasdubdub/recursive-trajectory-clustering`.

[39] L. Basora, V. Courchelle, J. Bedouet, and T. Dubot. Occupancy peak estimation from sector geometry and traffic flow data. *8th SESAR Innovation Days*, 2018.

[40] D. I. Ville Satopa, Jeannie Albrecht and B. Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior, last updates May 2020. Williams College, Williamstown, MA, University of Massachusetts Amherst, Amherst, MA, International Computer Science Institute, Berkeley, CA.

[41] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[42] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[45] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

[46] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[47] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[48] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[49] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 839–847. SIAM, 2014.

[50] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[51] D. Ulyanov. Multicore-tsne. `https://github.com/DmitryUlyanov/Multicore-TSNE`, 2016.

[52] F. Machanic. Radio navigation – collision avoidance systems. flight-mechanic.com/radio-navigation-collision-avoidance-systems/. last access December 2020.

[53] M. Dalichampt and C. Plusquellec. Hourly entry counts versus occupancy count–relationship, definitions and indicators, 2007.

[54] M. Dalichampt and C. Plusquellec. Hourly entry counts versus occupancy count–robustness, calibration, application (ii), 2007.

[55] X. Olive. traffic, a toolbox for processing and analysing air traffic data. *Journal of Open Source Software*, 4:1518, 2019. ISSN 2475-9066. doi: 10.21105/joss.01518.

[56] L. Basora. sectflow, June 2019. URL `https://github.com/lbasora/sectflow`.

[57] E. O. P. Raffaele Russo. Capacity planning and assessment additional considerations. Air Traffic Services System Capacity Seminar/Workshop, June 2016.

# Appendix A

## A.1 Review of High-Complexity Occurrences reported in 2017 at Croatia Control

At [8], chapter 6, mentions occurrences reported in 2017 during the Summer season at Croatia Control, here are summarized these events, since it is important to understand the causes of complexity in past events. Notice due to safety recommendations sectors are opened at one sector increment.

| Occurrence Report | Analysis |
|---|---|
| **"HA" problematic,<br>5 July 2017** | Time of maximum overload occurred from 9:40 until 10:00, there is a significant leap in traffic entry in 20-minute entry count: There are 21 flights entering reported while published capacity is only 11.There was an overload of 8 flights before new sector configuration opened in 20-minute period and 11 flights overload in 60 minute period. In 60-minutes entry counts 52 flights, while declared capacity is 33 flights. The new sector configuration substituted the High and Top Adria (which were horizontal divisions) into Top High West and Top High South (vertical divisions).<br>If the increase of traffic that this timespan could have been predicted, the proposition of a new sector configuration before the overload would have prevented the overload of ATCOs workload. |
| **Overdelivery LDNX, 11 July 2017** | There was an active regulation at LDNX from 17:40 to 21:20, from 18:00 until 19:00, by observing a 60-minute capacity and entry-rate an overload of 20 flights can be seen, even though regulation was active, when looking at the 20-minute frame there is an overload of 10 flights entering the sector. Stated peak happened right before the opening of a new sector configuration where the North sector was divided into Top Hight North and Upper Lower North.<br>Most of the overloading planes migrated from Serbian airspace to avoid sever weather. SIGMET&IR and MSG Sandwich Product weather reports, show thunder strikes and clusters of Cumulonimbus clouds. |
| **High peak at ULW, 15 July 2017** | 15 flights were not predicted to enter the reported airspace in the mentioned time period and 7 flights did not enter airspace even though it was initially planned. Out of 15 flights (11 overflights, 3 arrivals and 1 departure) that were not planned for the given time period, 12 entered later than planned(delayed), 2 were earlier than planned and 4 were planned the fly over the reported area. |
| **Meteo situation/called Hungarian ATCOs to put regulation** | Severe weather resulted in regulation between 7:00 to 8:00, besides the regulation there were 16 flights above the capacity defined to the sector LDNX, by the time of closing the regulation it had accumulated 222 minutes of delay. Even though reported item was based on weather conditions, there were no significant deviations from the initial and actual tracks. ATCOs called Hungarian ANSP to activate regulation due to weather and, according to NEST, it was activated from 8:20 to 13:00, which accumulated 1027 minutes of delay. In SIGMAT&IR report, it can be seen at 5:00 UTC severe Cumulonimbus clouds just passing Croatia-Hungary border (cluster of thunder strikes), this cluster travels on to Hungary airspace. |
| **Overdelivery LDTHWX** | The entry count has increased 25%. Most of extra entries represent flights that deviate from original course due to severe weather leaving sector from the same FIR LDNX and migrating to sector LDTHW (lateral rerouting). In period from 17:50 to 18:10 it can be seen that there are 18 actual entries of which 16 were not initially planned to enter in states timespan, most of initial and actual flights that were supposed to enter stated airspace did not enter in initial time due to delays. SIGMET&IR and MSG Sandwich Product weather reports confirm weather which caused mentioned deviation in initial and actual traffic flows. |
| **HW Regulations** | According to NEST data regulation was active from 16:00 to 17:38 and that caused 238 minutes of delay. Between 16:10 and 16:30 there is a hight peak with an overload of 8 flights which rapidly drops to 8 flights between 16:30 and 16:40. Possible reason for such high fluctuation is delay and deviation from initial route. Initial count was 15 planes per 20 minutes for both timestamps. |
| **LDUW – ATCOs heavily loaded but airspace occupation was 8-10 aircraft** | Between 16:10 and 16:30, there is significant deviation in actual track for few entering flights, probably to fulfil separation from other traffic, however 2 occurances of high risk happen in an interval of 30 minutes, 2 converging flights at the same flight level(FL350) and two flight heading with opposite directions and the difference in flight levels corresponds to 600ft. |

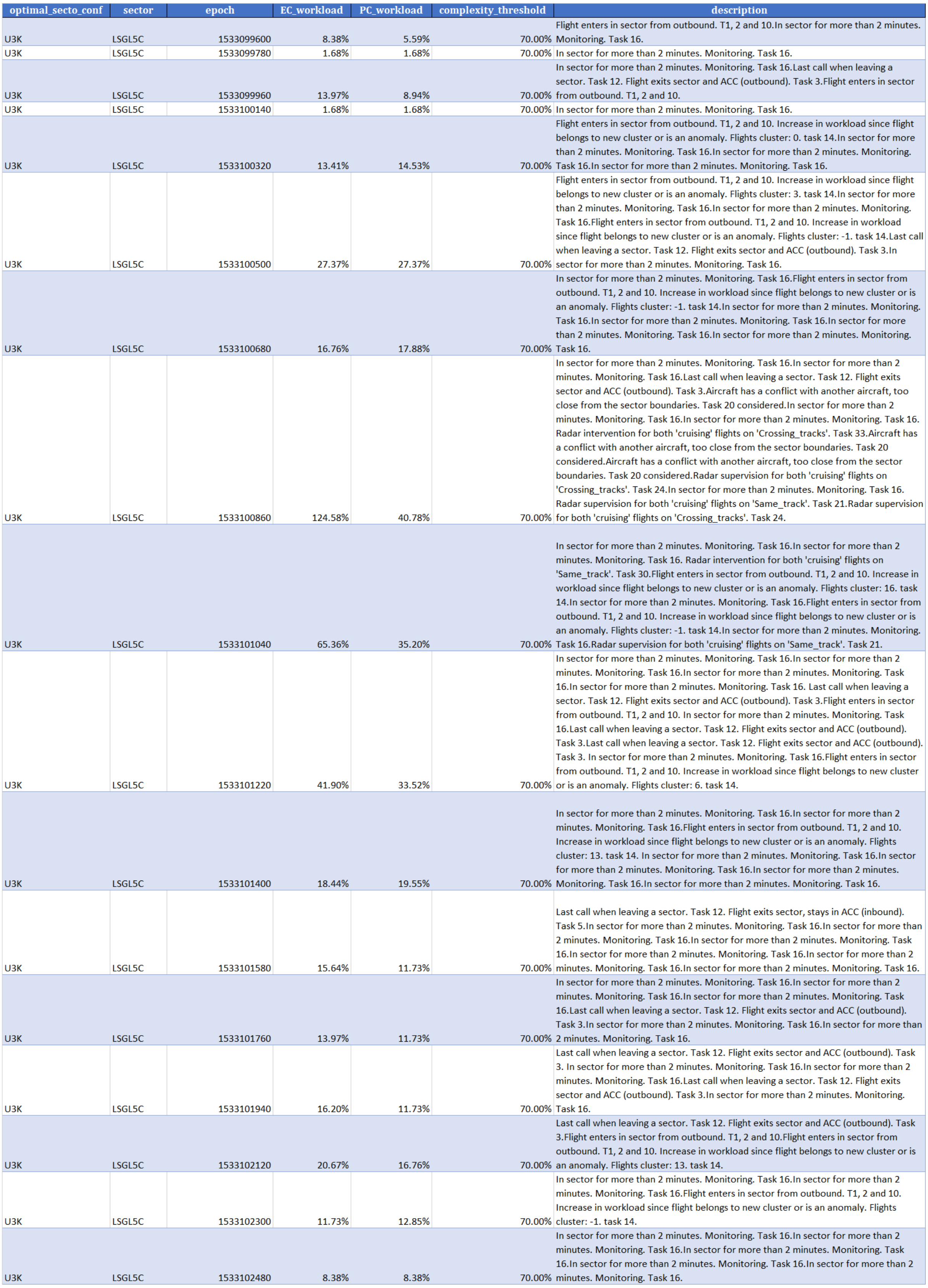
## A.2 CAPAN parameters and corresponding computing event

EC - executive controller, PC - planning controller

| | Task | Corresponding Computation | EC [s] | PC [s] |
|---|---|---|---|---|
| 1 | receive flight information (RFI) | entry of any aircraft | 0 | 1 |
| 2 | receive time and level estimate from a different ACC (RTLEDA) | entry outbound | 0 | 6 |
| 3 | transmit time and level estimate to different ACC (TTLEDA) | exit outbound | 0 | 6 |
| 4 | receive time and level estimation from same ACC (RTLESA) | entry inbound | 0 | 3 |
| 5 | transmit time and level estimation to same ACC (TTLEDA) | exit inbound | 0 | 3 |
| 6 | receive information from an aircraft that comes from upper or lower sector to different ACC | entry upper lower bound different ACC | 0 | 10 |
| 7 | transmit information from an aircraft that comes from upper or lower sector to different ACC | exit upper lower bound different ACC | 0 | 10 |
| 8 | receive information from an aircraft that comes from upper or lower sector in same ACC | entry upper lower bound same ACC | 0 | 6 |
| 9 | transmit information from an aircraft that comes from upper or lower sector in same ACC | exit upper lower bound same ACC | 0 | 6 |
| 10 | first call aircraft that comes from different ACC | entry outbound | 12 | 0 |
| 11 | first call aircraft that comes from same ACC | entry inbound | 10 | 0 |
| 12 | last call when leaving a sector | any aircraft exiting | 10 | 0 |
| 13 | conflict search to establish initial level clearance for flight entering the sector in climb or descent | climbing or descending aircraft coming in | 3 | 0 |
| 14 | conflict search to establish sector planning clearance | an aircraft that belongs to a new cluster or is an anomaly coming in | 3 | 10 |
| 15 | additional conflict search when more than 10 AC within sector | an aircraft comes in and the setor has an aircraft number greater than 10 | 0 | 2 |
| 16 | monitoring of aircraft that stay in sector for more than 2 minutes (every 2 minutes) | aircraft that stays in the sector for more than 2 minutes | 3 | 3 |
| 17 | report on an aircraft on reaching a specified level | a climbing or descending aircraft that starts cruising | 3 | 0 |
| 18 | additional monitoring for long climb/descent phase every 2 minutes | an aircraft at climbing or descent in sector for more than 2 minutes | 3 | 0 |
| 19 | instruction to climb or descent | an aircraft that starts climbing or descending at middle of sector | 10 | 0 |
| 20 | additional radar supervision when closest point of approach is close to boundary | if CPA between 2 aircraft is 10NM from sector boundaries | 20 | 7 |
| 21 | radar supervision of two aircraft at same track both in cruise at same flight level | two aircarft FL= in cruise with 15>=CPA >=10NM, vertic. dist.<1000ft and smallest angle <= 45˚ | 20 | 7 |
| 22 | radar supervision of two aircraft same track one of which is descending or climbing | two aircraft one in cruise other in climb/descent with 15>=CPA >=10NM, vertic. dist.<2000ft and smallest angle <= 45˚ | 24 | 7 |
| 23 | radar supervision of two aircrat at same track both climbing or descending | both aircraft in climb/descent with 15>=CPA >=10NM, vertic. dist. < 3000ft and smallest angle <= 45˚ | 24 | 7 |
| 24 | radar supervision of two aircraft at crossing tracks both in cruise at same flight level | two aircarft FL= in cruise with 18>=CPA >=12NM, vertic. dist.<1000ft and 45˚< smallest angle < 135˚ | 30 | 7 |
| 25 | radar supervision two aircraft on crossing tracks one of which is in climb or descent | two aircraft one in cruise other in climb/descent with 18>=CPA >=12NM, vertic. dist.<2000ft and 45˚<smallest angle < 135˚ | 36 | 7 |
| 26 | radar supervision two aircraft on crossing tracks, both of which are in climb or descent | both aircraft in climb/descent with 18>=CPA >=12NM, vertic. dist. < 3000ft and 45˚<smallest angle < 135˚ | 36 | 7 |
| 27 | radar supervision two aircraft on opposite tracks, both of which are in cruise at the same flight level | two aircraft in cruise with 22>=CPA>=15NM, vertic. dist <1000ft and 135˚<= smallest angle <=180˚ | 18 | 7 |
| 28 | radar supervision two aircraft on opposite tracks, one of which is in climb or descent. | two aircraft one in cruise other in climb/descent with 22>=CPA >=15NM, vertic. dist.<2000ft and 135˚<smallest angle <= 180˚ | 27 | 7 |
| 29 | radar supervision two aircraft on opposite tracks both of which are in climb or descent. | both aircraft in climb/descent with 22>=CPA >=15NM, vertic. dist. < 3000ft and 135˚< smallest angle <= 180˚ | 36 | 7 |
| 30 | radar intervention of two aircraft at same track both in cruise at same flight level | two aircarft FL= in cruise with 10NM > CPA , vertic. dist.<1000ft and smallest angle <= 45˚ | 55 | 10 |
| 31 | radar intervention of two aircraft same track one of which is descending or climbing | two aircraft one in cruise other in climb/descent with 10NM > CPA, vertic. dist.<2000ft and smallest angle <= 45˚ | 55 | 10 |
| 32 | radar intervention of two aircrat at same track both climbing or descending | both aircraft in climb or descent with 10NM > CPA, vertic. dist. < 3000ft and smallest angle <= 45˚ | 55 | 10 |
| 33 | radar intervention of two aircraft at crossing tracks both in cruise at same flight level | two aircarft FL= in cruise with 12 > CPA , vertic. dist. < 1000ft and 45˚< smallest angle < 135˚ | 58 | 10 |
| 34 | radar intervention two aircraft on crossing tracks one of which is in climb or descent | two aircraft one in cruise other in climb or descent with 12 > CPA , vertic. dist.<2000ft and 45˚<smallest angle < 135˚ | 58 | 10 |
| 35 | radar intervention two aircraft on crossing tracks, both of which are in climb or descent | both aircraft in climb or descent with 12 > CPA, vertic. dist. < 3000ft and 45˚<smallest angle < 135˚ | 58 | 10 |
| 36 | radar intervention two aircraft on opposite tracks, both of which are in cruise at the same flight level | two aircraft in cruise with 15NM > CPA, vertic. dist < 1000ft and 135˚<= smallest angle <=180˚ | 62 | 10 |
| 37 | radar intervention two aircraft on opposite tracks, one of which is in climb or descent. | two aircraft one in cruise other in climb or descent with 15 > CPA, vertic. dist. < 2000ft and 135˚<= smallest angle <= 180˚ | 62 | 10 |
| 38 | radar intervention two aircraft on opposite tracks both of which are in climb or descent. | both aircraft in climb or descent with 15NM > CPA, vertic. dist. < 3000ft and 135˚<= smallest angle <= 180˚ | 62 | 10 |

## A.3 Sector $LSGL5C$ between 5 a.m. and 5:51a.m. analysis, sector configuration $U3K$

Table A.1: HEC, COmplexity and Description of Tasks, LSGL5C, $U3K$, between 05:00 and 05:51.



| optimal_secto_conf | sector | epoch | EC_workload | PC_workload | complexity_threshold | description |
|---|---|---|---|---|---|---|
| U3K | LSGL5C | 1533099600 | 8.38% | 5.59% | 70.00% | Flight enters in sector from outbound. T1, 2 and 10.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533099780 | 1.68% | 1.68% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533099960 | 13.97% | 8.94% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.Flight enters in sector from outbound. T1, 2 and 10. |
| U3K | LSGL5C | 1533100140 | 1.68% | 1.68% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533100320 | 13.41% | 14.53% | 70.00% | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 0. task 14.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533100500 | 27.37% | 27.37% | 70.00% | Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 3. task 14.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533100680 | 16.76% | 17.88% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533100860 | 124.58% | 40.78% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. Radar intervention for both 'cruising' flights on 'Crossing_tracks'. Task 33.Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered.Aircraft has a conflict with another aircraft, too close from the sector boundaries. Task 20 considered.Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24.In sector for more than 2 minutes. Monitoring. Task 16. Radar supervision for both 'cruising' flights on 'Same_track'. Task 21.Radar supervision for both 'cruising' flights on 'Crossing_tracks'. Task 24. |
| U3K | LSGL5C | 1533101040 | 65.36% | 35.20% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. Radar intervention for both 'cruising' flights on 'Same_track'. Task 30.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 16. task 14.In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14.In sector for more than 2 minutes. Monitoring. Task 16.Radar supervision for both 'cruising' flights on 'Same_track'. Task 21. |
| U3K | LSGL5C | 1533101220 | 41.90% | 33.52% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.Flight enters in sector from outbound. T1, 2 and 10. In sector for more than 2 minutes. Monitoring. Task 16.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 6. task 14. |
| U3K | LSGL5C | 1533101400 | 18.44% | 19.55% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 13. task 14. In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533101580 | 15.64% | 11.73% | 70.00% | Last call when leaving a sector. Task 12. Flight exits sector, stays in ACC (inbound). Task 5.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533101760 | 13.97% | 11.73% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533101940 | 16.20% | 11.73% | 70.00% | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3. In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.In sector for more than 2 minutes. Monitoring. Task 16. |
| U3K | LSGL5C | 1533102120 | 20.67% | 16.76% | 70.00% | Last call when leaving a sector. Task 12. Flight exits sector and ACC (outbound). Task 3.Flight enters in sector from outbound. T1, 2 and 10.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: 13. task 14. |
| U3K | LSGL5C | 1533102300 | 11.73% | 12.85% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.Flight enters in sector from outbound. T1, 2 and 10. Increase in workload since flight belongs to new cluster or is an anomaly. Flights cluster: -1. task 14. |
| U3K | LSGL5C | 1533102480 | 8.38% | 8.38% | 70.00% | In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16.In sector for more than 2 minutes. Monitoring. Task 16. |