UNIVERSITÀ DEGLI STUDI DI TRIESTE

DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA

Corso di Studi in Ingegneria Informatica

Tesi di Laurea Magistrale

OPTIMAL DECISION MAKING FOR AIR TRAFFIC SLOT ALLOCATION IN A COLLABORATIVE DECISION MAKING CONTEXT

Laureando:
Stefano Pacchiega

Relatore:
prof. Lorenzo Castelli

Correlatore:
prof. Sergio Ruiz

ANNO ACCADEMICO 2015–2016

Contents

List of Figures

List of Tables

Nomenclature

ACDM	Airport Collaborative Decision Making
ANSP	Air Navigation Service Providers
ATC	Air Traffic Control
ATFCM	Air Traffic Flow and Capacity Management
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
AU	Airspace User
CCS	Capacity Constraint Situation
CDM	Collaborative Decision Making
CI	Confidence Index
DCB	Demand & Capacity Balancing
DST	Decision Support Tools
EC	European Commission
ESFP	Enhanced Selective Flight Protection
eSS	enhanced Slot-Swapping
FAA	Federal Aviation Administration
FCFS	First Come First Served
FPFS	First Planned First Served
GDP	Ground Delay Programs
GHP	Ground Holding Problem
IFR	Instrument Flight Rules
ILS	Instrument Landing System
LOC	Leftover Operational Credit
LP	Linear Programming
MUMH	Multiple Users Multiple Hotspots
MUSH	Multiple Users Single Hotspot
NM	Network Manager
NOP	Network Operations Plan
SES	Single European Sky
SESAR	Single European Sky ATM Research
SFP	Selective Flight Protection
SUMH	Single User Multiple Hotspots
SUSH	Single User Single Hotspot
TEN-T	Trans-European Transport Network
UDPP	User Driven Prioritisation Process
VFR	Visual Flight Rules

Abstract - English

The purpose of this research is to design an optimization model for User Driven Prioritisation Process - Selective Flight Protection (UDPP - SFP): a collaborative procedure developed within the framework of the Single European Sky ATM Research (SESAR) programme to support Airspace Users (AUs) to manage their delayed flight during conditions in which the actual airport capacity is unexpectedly reduced and below the nominal value. During these periods, referred to as Capacity Constraint Situations (CCSs), it may happen that the demand for using the available resources (the slots) is larger than the actual capacity. The occurrence of such event is called hotspot. The SFP procedure, under certain constraints, enables airlines to decrease the delay of some higher-priority flights as much as possible in exchange of suspending (i.e. send to the last position of the hotspot's queue) other flights of less tactical importance, thus allowing a higher flexibility in scheduling and a better management of delay costs.

The first phase of this work involves the study of a centralized system in which all AUs work together to achieve the common welfare. Results show a total cost delay reduction (on average) of 24% with respect to the First Planned First Served (FPFS) solution. The second phase addresses how to manage and merge into a single flight sequence the different AUs decisions which are the output of a distributed SFP procedure. The final phases involves the development of a simulator in JAVA which creates many instances of the hotspot problem. Therefore it has been possible to highlight statistically which are the pros and cons of all the mechanisms considered in order to reduce the costs of delay.

Keywords: User Driven Prioritisation Process, Selective Flight Protection, cost of delay, Capacity Constraint Situation, hotspot, Collaborative Decision Making, slot allocation.

Abstract - Italiano

Lo scopo di questa ricerca è progettare un modello di ottimizzazione per lo "User Driven Prioritisation Process - Selective Flight Protection" (UDPP - SFP): una procedura ideata da "Single European Sky ATM Research" (SESAR) per aiutare gli utenti dello spazio aereo (AU) a gestire i propri voli in ritardo durante una Capacity Constraint Situation (CCS): una condizione per gli aeroporti di capacità inaspettatamente ridotta dovuta a cattivo tempo, incidenti aerei, etc. Durante una CCS può accadere che la domanda di utilizzo delle piste di atterraggio sia maggiore della capacità effettiva; questo evento è chiamato hotspot. La procedura SFP, attraverso determinati vincoli, permette alle compagnie aeree di diminuire, per quanto possibile, il ritardo di alcuni voli scelti e di sospenderne altri fino alla fine dell'hotspot, consentendo così una maggiore flessibilità nella programmazione aerea e una migliore gestione dei costi legati al ritardo.
La prima fase del progetto prevede lo studio di un sistema centralizzato in cui tutti gli AU lavorano insieme per raggiungere il bene comune. I risultati delle simulazioni mostrano una riduzione dei costi legati al ritardo in media del 24% rispetto la schedulazione First Planned First Served (FPFS), normalmente usata. La seconda fase concerne come diversi utenti cerchino di minimizzare i propri costi inerenti al ritardo individualmente. Sono stati studiati diversi algoritmi per interpolare i risultati dei vari AU, quali ad esempio l'iterazione delle procedure e l'unione delle varie preferenze. Nella fase finale è stato sviluppato un simulatore in JAVA che crea molte istanze del problema hotspot. È stato così possibile studiare statisticamente quale siano pregi e difetti delle varie procedure analizzate.

Parole chiave: User Driven Prioritisation Process, Selective Flight Protection, costo del ritardo, Capacity Constraint Situation, hotspot, Collaborative Decision Making, slot allocation.

Chapter 1

Introduction

In recent years the large amount of air traffic in Europe has led to a significant increase in flight delays that become large costs for airlines (fuel, crew, taxes, discontent of passengers, ...).

In 2014, around 880 million of passengers used the air transportation system in Europe [1]. The United Kingdom reported the highest number of air passengers in this period, with 220 million or an average of 3.4 passengers per inhabitant (which was the double the EU-28 of passengers average). In Table 1.1 is shown the number of passengers in the busiest European airports for that year (2014).

Table 1.1: The busiest airports in 2014

AIRPORT	PASSENGERS
London Heathrow	73 million
Paris' Charles de Gaulle	64 million
Frankfurt	59 million
Amsterdam's Schiphol	55 million

After 2016, the traffic growth in Europe is estimated at around 2.6% increase rate per year [2]. The 2008 peak of traffic of 10.1 million flights is forecasted to be reached again in 2017. The new forecast is for 11.4 million IFR (Instrument Flight Rules) movements in 2021, accounting for 19% more than in 2014.

These huge numbers are expected to keep growing exponentially in the incoming years and they represent the complexity of Air Traffic Management (ATM) in Europe.

SESAR (Single European Sky ATM Research), with the aim of increasing capacity but also to use the available capacity in a more efficient manner, designed some solutions for this growing problem given the large amount of air passengers.

This research is focused on creating an optimization model of one of the mechanisms introduced by SESAR to manage the excess demand in reduced capacity situation: User-Driven Prioritisation Process (UDPP).

The air traffic is planned and scheduled from several weeks ago up to a few hours before the time of operations. However, close to the flight execution, the air transportation system may be subject to unexpected capacity reductions caused by:

- bad weather (i.e. snow, reduced visibility, thunderstorms, ...);
- large volume of aircraft going to the airport;
- closed runways;
- aircraft incidents;
- conditions that require increased spacing between aircraft (Instrument Landing System (ILS) vs. Visual Flight Rules (VFR)).

The reduced capacity period forces to re-plan different tactical scheduling for the air traffic usually by applying FPFS policies. After such process, some flights may be assigned considerable delay, which in turn represents an increase of costs for the Airspace Users (AUs). AUs would like to have extras flexibility to redistribute the delay and associated costs among their flights.

The periods in which the capacity available is reduced below the nominal are called **Capacity Constraint Situations (CCS)**[1]. During a CCS, it can happen that the number of requested resources (the slots) are larger than the reduced actual capacity. The period in which the demand exceeds the available capacity is called **hotspot**. During a hotspot all flights increments their delay and their request is handled as shown in Fig. 1.1.

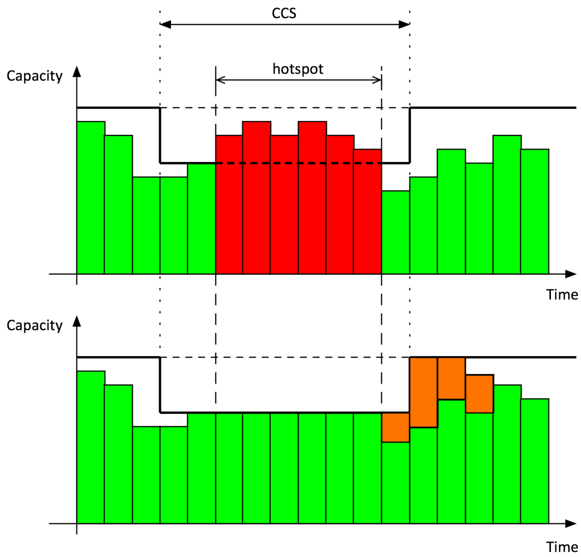


Figure 1.1: CCS and hotspot

FPFS is a common policy due to the fact that it is widely considered by the AUs a fair policy in the event of a hotspot. It is important to note that the FPFS scheduling in reduced capacity condition is not the best choice to reduce

[1] The definitions of CCS, hotspot and slot will be formalized in chapter 2, section 2.3.1

the delay cost of flights. Each flight indeed may have (and indeed usually does have) a different cost than others for the same delay. The difference of the delay costs depends on many factors, e.g. the number of passengers, possible connections with other flights, etc., and it may determine (together with the incomes) the relative tactical operational importance of each flight.
In the following example it can be seen how a scheduling FPFS can be bad for the final sum of costs.

Example Consider a situation in which 5 flights are involved. In Table 1.2 there is their initial scheduling.

Flights	Arrival Time
Flight A	12.00
Flight B	12.01
Flight C	12.02
Flight D	12.03
Flight E	12.04

Table 1.2: Initial Arrival Time - Motivational Example

Time	12.00	12.01	12.02	12.03	12.04
Flights	A	B	C	D	E

Figure 1.2: Initial Arrival Time - Motivational Example

Each flight has a different income and cost function. It will be assigned a different value of cost per minute of delay to each flight. These costs are represented in Table 1.3.

Flights	Delay Cost (€)
Flight A	2
Flight B	3
Flight C	1
Flight D	2
Flight E	11

Table 1.3: Different Costs - Motivational Example

Subsequently there is a reduction of 50% of the sector capacity. Thus, a delay occur to most flights. In Figure 1.3 it is shown how the flights are affected by the capacity reduction, their current delay and the related cost due to the delay.

Time	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	Σ
Initial	A	B	C	D	E						
FPFS	A		B		C		D		E		
Delay (minutes)	0		1		2		3		4		10
Costs	0		3		2		6		44		55

Figure 1.3: FPFS Scheduling - Motivational Example

In Table 1.3 it can be seen that the flight E has a huge cost of minute of delay, more than the other flights. Its delay of 4 minutes leads to a large final cost (indeed it is the 80% of the total cost). It can be presumed that the decrease of flights E delay through swapping procedure allows to decrease the total cost.

Applying a slots swapping between flight E and C we got the result shown in Figure 1.4.

Time	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	Σ
Initial	A	B	C	D	E						
Final	A		B		E		D		C		
Delay (minutes)	0		1		0		3		6		10
Costs	0		3		0		6		6		15

Figure 1.4: Swapping Scheduling - Motivational Example

Thanks to this slot swapping, costs can be notably decreased: from 55€ to 15€. In this simple example the FPFS solution is not the best solution for the system in a reduced-capacity situation.

1.1 Motivation

In order to improve the management of air traffic delays during reduced capacity conditions, SESAR has introduced a mechanism that is based on collaboration between users called Airport Collaborative Decision Making (A-CDM). All the stakeholders (airport operators, aircraft operators, ATC and NM) can work together in a more transparent manner, sharing all their data and strategies to create a common plan for all the network operations. There will be a larger interactivity of the airlines in the system and they can autonomously take decisions to change trajectories and schedules of their flights in accordance to certain procedures and constraints. One of the new plans set up by the SESAR to manage delays during reduced capacity conditions is the **User-Driven Prioritisation**

Process (UDPP) that will be explained in detail in Chapter 2. One of the mechanisms used in UDPP is the Selective Flight Protection (SFP): a procedure that enables airlines to suspend some flights, increasing their delay until the end of the congestion. Due to the suspension, the operators earns credits that can be spent to reduce the delay of some of its other flights with the constraint that all flights of the system must not increase their initial delay. This mechanism allows a higher flexibility in scheduling and a better management of delay costs than FPFS scheduling.

1.2 Objectives

The main objective of this research is to design an optimization model for UDPP-SFP to help AUs to manage the flights in CCSs: i.e., in the event of a hotspot, which flights to suspend and which ones to prioritize in order to get the optimal solution for the user or overall in terms of costs.

In order to achieve the main objective other secondary important goals are defined:

1. Studying the UDPP-SFP mechanism in a single hotspot.
2. Understand the problem dynamics and solution bounds in specific cases:
    - Single User Single Hotspot (SUSH): the global optimal solution;
    - Multiple Users Single Hotspot (MUSH): UDPP-SFP Algorithm.
3. Develop the mathematical model of SUSH and MUSH;
4. Develop the optimization suite that can find the optimal solution of the problem from the network perspective and/or for the participant perspective (i.e., AUs);
5. Develop an agent-based simulator to reproduce representative scenarios;
6. Analyze the results in each specific case and extract conclusions;
7. Process the individual user solutions and their various combinations (i.e. iterations, union) in order to understand which is the best procedure to get the closest solution to the global optimal one (SUSH).
8. Identify the limitations of the knowledge generated and put solid Milestones for future work.

1.3 Methodology

In order to find the optimal decision making for AUs with respect to flight prioritisation request in the context of SFP, a mathematical programming model is developed in Section 3 in order to reflect the optimal operator behavior during a hotspot.

The methodology is described in Fig.1.5: once the problem is observed, it is followed by the development of a mathematical model. With the model, a set of simulations can be performed in order to make predictions of the various cases of the problem instances.

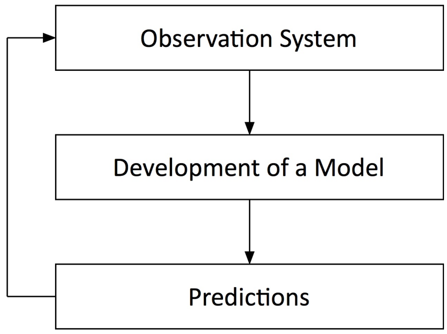


Figure 1.5: Research methodology

The main methodological steps in this research are:

1. Study of the Literature inherent to the UDPP;
2. Study of the Ground Delay Program (GDP) [4][11];
3. Development of linear programming model;
4. Design of the optimization mathematical model of UDPP-SUSH and UDPP-SFP using the graph theory.
5. Implementation of the optimization mathematical model in FICO Xpress-Optimizer [12], a commercial optimization solver for linear programming (LP), thus obtaining a small software suite that manages the different cases studied (FPFS - SUSH - MUSH).
6. Design and implementation of an agent-based simulator in JAVA in order to study thousands of instances of the problem (Monte Carlo simulation[2]).
7. Analyze the results and extract conclusions.

[2]Monte Carlo methods is a class of computational algorithms that use large random sampling to obtain numerical results.

1.4 Structure of the document

The document is divided into six main chapters.

In this **Chapter 1** there is the introduction to the project, the research motivation, the final objectives, the methodology to follow and the description of the document structure.

In **Chapter 2** the Literature Review is presented together with the background of the topic of this research. After describing SES (Single European Sky) and the SESAR Programme, fundamental concepts of UDPP are introduced: the ideas behind, actors of the system and main rules.

In **Chapter 3** the Selective Flights Protection (SFP) is presented with its principal definitions, assumptions, the algorithm and examples. It will consider the following two cases:

- Single User and Single Hotspot (SUSH): in this configuration of the problem, assuming that all the different airlines operate for the common goal of the network as a single-user, a decision maker in a hotspot who wants to find the best solution in terms of minimum costs for the entire set of flights, without differentiation between the various airlines and with a global perspective (global optimization). It's the global optimal solution for the UDPP problem.
- Multiple Users Single Hotspot (MUSH): there are many airlines in a single hotspot. Each airline tries to minimize the costs for its flights using UDPP and ignoring other companies tactical interests. The second part of this section will deal with how the various MUSH results (one from every user of the system) can be combined into a single solution with the purpose of obtaining a pseudo-optimal solution expectedly composed of several sets of local optimal solutions.

In **Chapter 4** the Simulator in JAVA is presented: the design and the implementation.

Chapter 5 describs the methods to study the data obtained from the simulator and the results (which procedure has a positive impact, such as a negative impact, limits and bounds).

Chapter 6 presents the conclusions and the limitation of this research and all possible related future work.

Chapter 2

Background and current state of SFP

2.1 Current air traffic tactical slot allocation mechanisms

In Europe the Air Traffic Management (ATM) authority imposes a regulation when there is imbalance between the requested demand and available capacity which consists in forcing delays (Air Traffic Flow Management (ATFM) delays) to some flights ([5],[25]). ATFM delays are imposed on a concept of equity but without considering the Airspace User (AU) costs[1]. The ATM authority then manages flights through a First Planned First Served (FPFS) policy, which represent a fair procedure.

The Air Traffic Management authority focuses primarily on safety and capacity to manage the various Airspace Users. Subsequently secondary objectives are related to efficiency and costs and they are the ATM and flight cost-efficiency.

In order to manage these goals, Single European Sky ATM Research (SESAR) has designed several procedures and mechanism which aim to manage the European ATM system (see Section 2.2.1), such as the Network collaborative management, dynamic demand, capacity balancing, airport integration in the ATM network (Airport Collaborative Decision Making (A-CDM)), traffic synchronization and 4D trajectory management.

2.1.1 Collaborative Decision Making in USA: Ground Delay Program

Interesting from a comparison viewpoint, the paradigm of Collaborative Decision Making in the USA is based on improving data exchange and communication between the Federal Aviation Administration (FAA) and the airlines, which lead to better decision making. The primary focus of the CDM at the beginning was the implementation of Ground Delay Program (GDP) enhancements. This

[1]The UDPP-SFP mechanism instead manages the various costs for users while maintaining the concept of fairness and increasing flexibility for the AUs.

procedure is usually instantiated when one of the following events occur at the airport:

- bad weather;
- an aircraft incident;
- closed runways;
- a large volume of aircraft going to an airport;
- a large volume of aircraft going en route to another airport in the same line of flight;
- a condition that requires increased spacing between aircraft (such as the Instrument Landing System (ILS) approaches vs. Visual Flight Rules (VFR) approaches).

In the Ground Delay Program flights that are scheduled to arrive during periods of congestion are assigned delays prior to their departure (it is assumed that is less expensive and safer to delay a flight on the ground than in airborne). This type of allocation can be viewed as a resource allocation problem, i.e.: the available arrival capacity (arrival slots) is allocated to flights (and a new departure slot is allocated to flights). The central idea of GDP is to manage delays and cancellations of flights to optimize the scheduling with special procedures.

2.3.3.2 GDP procedures

Before the implementation of GDP flights were assigned to slots by a First Come First Served (FCFS) algorithm known as Grover Jack. The affected flights were ordered according to their most recent estimated arrival times. This method of assigning flights to slots penalized the airlines for providing accurate information (if a flight delay increases and this is not communicated, the aircraft preserves the same slot).

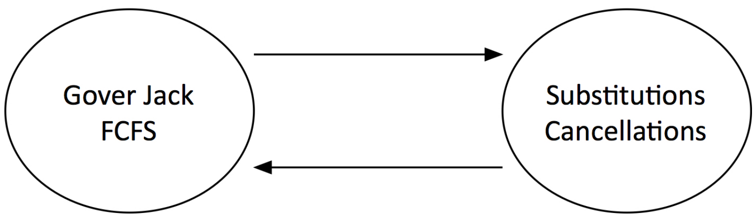


Figure 2.1: Gover Jack procedure, referring [4]

The new idea of the GDP paradigm is to consider the allocation of capacity to an assignment of slots to airlines. This has led to two new mechanisms, called Ration-By-Schedule (RBS) and Compression. The RBS algorithm generates the initial allocation of slots to airlines, based on the recognition that airlines have claims on the arrival capacity through the original flight schedules. FAA periodically execute Compression: an algorithm that updates the schedule by an inter-airline slot exchange, which aims to provide airlines with an incentive to report flight cancelations and delays (unlike the Gover Jack procedure). Airlines can adjust their schedules by substituting and canceling flights.

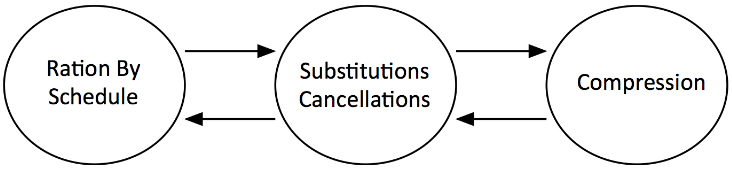


Figure 2.2: Ration by Schedule and Compression procedures, referring [4]

2.2 SESAR

The European Commission (EC) developed in 1999 a legislative framework for European aviation to manage the dramatic growth in the air travel. The previous airspace design driven by national boundaries are expected to merge into airspace blocks to maximize the efficiency of the airspace. Air Traffic Management will be driven by the requirements of the airspace user and the need to provide for increasing air traffic beyond the primary objective of safety.
In 2004 Single European Sky sets up the **Single European Sky ATM Research (SESAR)** project to define ATM systems through definitions, developments and deployments of new solutions, called SESAR Solutions.

SESAR is the EU-wide framework whose High-Level Goals are described in Table 2.1 [3].

Table 2.1: SESAR goals

GOAL	DESCRIPTION
CAPACITY	enable a threefold increase in capacity which will also reduce delays both on the ground and in the air
SAFETY	improve safety by a factor of 10
ENVIRONMENT	10% reduction in the effects flights have on the environment
COST	provide ATM services to airspace users at a cost of at least 50 % less

SESAR contributes to these High-Level Goals managing the resources of the entire ATM community, including:

- the Network Manager (NM)[2];
- civil and military air navigation service providers (ANSPs);
- airports and military aerodromes opened to civil air traffic;
- civil and military airspace users;
- staff associations;
- academia;
- research centers.

[2]EUROCONTROL, https://www.eurocontrol.int

SESAR project is composed by 3 different phases [6]:

1. **Definition phase** (2005 – 2008). This phase was characterized by different goals :
   - define European air transport system performance requirements up to 2020 and beyond;
   - identify globally interoperable ATM solutions;
   - produce the detailed research and technology work programme;
   - propose the legislative, financial and regulatory framework;
   - establish a detailed deployment plan.

   This phase was co-financed by the European Commission (through the Trans-European Transport Network (TEN-T) programme) and EUROCONTROL. Its total cost was 60 million Euro (50% European Commission, 50% EUROCONTROL).

2. **Development phase** (2008 – 2016). This phase is managed by the SESAR Joint Undertaking [3]: a joint undertaking created under European Community law on 27th of February 2007. SESAR Joint Undertaking produced the required new generation of technological systems, components and operational procedures as defined in the SESAR ATM Master Plan and Work Program. Its total cost is 2.1 Billion Euro (1/3 European Commission, 1/3 EUROCONTROL, 1/3 Industries).

3. **Deployment phase** (2014 – 2020). It will be a large-scale production and implementation of the new air traffic management infrastructure composed of interoperable components guaranteeing high performance air transport activities. It will cost 20 Billion Euro paid by the Industry.

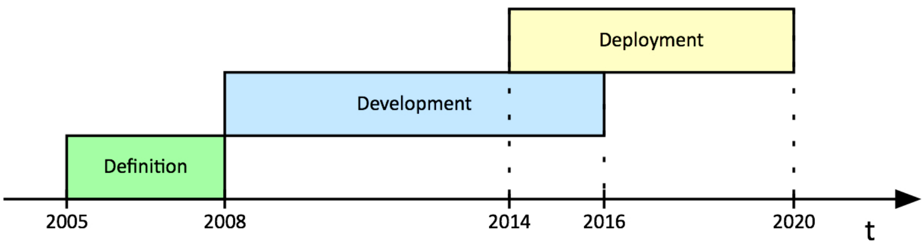


Figure 2.3: Phases of SESAR project

The SESAR Solution are operational and technological improvements which aim to manage the European ATM system. The concept of operations is based on the following topics:

- **Moving from airspace to 4D trajectory management** allows the sharing of aircraft trajectories between all the ATM users. Airspace Users (AU) and Air Navigation Service Providers (ANSP) have the same view of all the flights in the system. They can have the the most up-to-date data available to perform their tasks therefore it let dynamic adjustment aircraft trajectories or scheduling.

[3]SESAR Joint Undertaking, http://www.sesarju.eu.

- **Traffic synchronization** aims to reach an optimum traffic sequence: optimize the sequencing of aircraft surface movements, arrivals and departures at airports so to avoid the Air Traffic Control (ATC) tactical intervention.
- **Network collaborative management, dynamic demand and capacity balancing** aim different phases of operation planning from long to medium/short term. All users of the system progressively share more and more precise data to build a common traffic and operational environment picture: the Network Operations Plan (NOP)[14]. This topic is closely related to the first argument: 4D Trajectory which is a first example of sharing precise data in the entire system. The **Collaborative Decision Making** (CDM) and the **User Driven Priorisation Process** (UDPP) are part of this topic. During periods of imbalance between demand and capacity in particular sector of ATM, the process UDPP will be activated: users can negotiate together and make proposal to change the scheduling of flights to manage the critical condition in an efficient and equitable way.
- **Airport integration and throughput** aims a full integration of airports into the ATM network. An important topic of SESAR is the implementation of Airport Collaborative Decision Making (A-CDM) which improves Air Traffic Flow and Capacity Management (ATFCM) at airports reducing delays and improving the predictability and efficiency.
- **Conflict management and automation** aims at reducing controller task load per flight through a increment of integrated automation support. Air Navigation Service Provider (ANSP) and pilots will be assisted by new automated functions to handle decision-making processes.
- **System Wide Information Management (SWIM)** consists of new standards, infrastructure and governance enabling the management of ATM information. It is a complete change in the way of how information is managed along its lifecycle: the future European ATM intranet.

2.3 Cost of delay

One of the main objectives of SESAR is to reduce the various costs for Airspace Users, as shown in Table 2.1. One of the major costs that SESAR aims at reducing is the cost of delay.

The actual cost of delay of a particular flight depends on many factors and it is difficult to know it with exactitude even for the operator of the flight. However it is possible to approximate these costs through statistical characterization. In [19] some reference values are given.

The cost of delay can be split in two sides: hard and soft costs.
Hard are those costs resulting from passenger delay, such as those due to passenger rebooking, compensation and care. Table 2.4 shows reference values (ref. [19]) of different care costs per passenger for different hours of delay.

Delay duration	Care provision	€ (2005-2006)
Up to 2 hours	Refreshment (bottle of water)	1.5
2-3 hours	Tax-free voucher and phonecard	7.0
3-5 hours	Tax-free voucher, meal voucher, phone card and frequent-flyer miles	17.2
Over 5 hours (no hotel)	Tax-free voucher, meal voucher, phonecard, frequent-flyer miles and ticket discount voucher	19.2
Over 5 hours (with hotel)	Tax-free voucher, meal voucher, phone car, frequent-flyer miles, ticket discount voucher and hotel accommodation	75.0

Figure 2.4: Average care costs per delayed passenger, ref. [19]

Soft costs manifest in several ways, for example as follow:

- A passenger with a flexible ticket may arrive at an airport and decide to take a competitors on-time flight instead of a delayed flight.
- Passengers may perceive the airline unpunctual and choose another.
- Due to a delay, a passenger may defect from an unpunctual airline as a result of dissatisfaction.

Soft costs are rather more difficult to quantify, but may even overcome the hard costs. Longer delays have higher associated costs per minute. This fact, together with the different absolute values of the delay cost for each individual flight, can be described by a particular function of costs, similar to that in Figure 2.5 (these three different functions are made using the data in Table 2.6 that represents the costs of delay in three different scenarios (low, normal and high cost scenario), [19]).

Delay minutes range (minutes)	1	16	31	46	61	76	91	120	180	240	300
Low cost Scenario	0,06	0,17	0,26	0,35	0,42	0,47	0,58	0,75	0,89	0,92	1,15
Base cost scenario	0,13	0,36	0,63	0,89	1,11	1,24	1,47	1,75	1,98	2,03	2,40
High cost scenario	0,15	0,43	0,72	1,03	1,27	1,42	1,69	2,03	2,31	2,38	2,82

Figure 2.5: Total passenger costs of delay per minute, ref. [19]

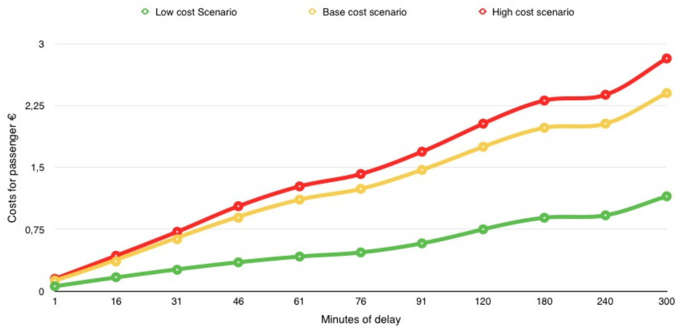


Figure 2.6: Graph passenger costs of delay per minute

2.4 UDPP

2.4.1 UDPP Philosophy

UDPP is the Airspace User Driven Process that contributes to manage more efficiently a Capacity Constrained Situation (CCS) in Airport or in Airspace. It is designed to provide flexibility to the AUs so they can prioritize their flights in case of critical CCS situation according to their relative tactical importance. It will become the key for Airspace Users to reduce the delay/cancellation induced cost. The costs of delay caused by the reduction of the capacity can be better controlled and managed.

The aim of UDPP is to provide to the Airspace Users the possibility to rearrange their own flight sequence through an AU driven prioritisation process. This procedure can be described as follows:

- **UDPP Departure** allows the AUs to change in the pre-departure sequence the priority order of flights. Provisions will be introduced in NM to take into consideration the CDM airport delay.
- **ATFM Slot Swapping Enhancement**: the current slot-swapping of regulated flights between AUs will be extended to allow an enhanced service with the aim to increase flexibility. The enhanced Slot-Swapping (eSS) will include the development of new processes and services addressing multi ATFM slot-swapping.

UDPP should not be used to the detriment any other AU. It will include a set of principles and rules to ensure fairness and equity.

2.4.2 UDPP Actors

In UDPP there are three different actors:

- **The Airspace User (AU)** is the central entity of UDPP and his participation is voluntary. AU will have the opportunity to provide Prioritization requests to improve their flight operations.

- **The Network Manager (NM)** is informed of changes resulting from the development and implementation of the UDPP-Departure.
- **Airport CDM (A-CDM)** is informed of changes resulting from the development and implementation of the ATFM Slot Swapping Enhancement.

2.4.3 UDPP Principles and Rules

In UDPP there are 8 important principles that aim at ensuring fairness and equity in the system.

1. The methodology and process of UDPP must at all times remain transparent to all stakeholders.
2. When UDPP is used, it is currently assumed that there shall be no negative impact on the network.
3. AUs have the right not to participate in UDPP without prejudice.
4. The process and infrastructure will be easily and readily accessible, and be user-friendly to all actors.
5. Before UDPP is considered between AUs there must be binding agreements between the actors involved.
6. When a request for prioritisation is made, a response shall be required.
7. The UDPP process transactions information shall be made available to AUs by an automatic system.
8. The UDPP process transactions shall be recorded to allow for post-operations tracking of potential fraudulent tactics.

2.5 SFP

The main idea of the UDPP is to provide more flexibility to AUs to manage the delay (and costs) of their flights in the event of a hotspot. One of the mechanisms that have been proposed is a collaborative decision-making process called Selective Flight Protection (SFP) which uses ATFM slot swapping and departure reordering techniques. The core of the algorithm is to suspend some flight thus earning credits called **Operating Credits** (OC). These credits can be re-allocated/re-distributed to minimize the delay of other flight. The number of flights that the operator can prioritize depends on the severity of system congestion and on his amount of credits. SFP respects the concept of fairness: flights that do not participate can not increase their lagging behind the First Planned First Served (FPFS) scheduling delay.

2.5.1 Definitions

Slots The right to use a physical (and scarce) resource for a specific period of time. Slots are the principal resources in the context of UDPP. They represent airstrips at given time. Each slot is characterized by a precise start, end time and a duration which is often referred as the size of the slot (usually in minutes).

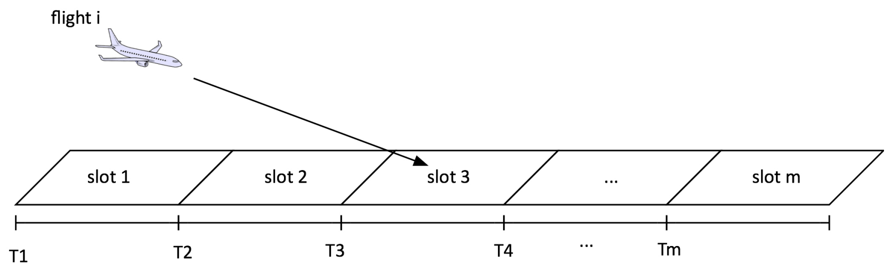


Figure 2.7: Example of slots

Capacity Amount of the slots available in a period of time (usually defined as the number of the flights that can be processed per hour).

Capacity Constraint Situation (CSS) A CCS indicates the reduction of the nominal capacity of the airport. It can create troubles for users of the airspace respect landing/departing runway capacity. The CCS period is defined by a list of capacity constraint elements given by:

1. Start Time;
2. End Time;
3. % of remaining capacity respect the nominal capacity;
4. Confidence Index (CI).

The Confidence Index represent the level of confidence of the CCS in the predicted evolution of the capacity and demand.

$$\begin{array}{cc} \text{CI} = 1 & \text{maximum certainty} \\ \text{CI} = 0 & \text{maximum uncertainty} \end{array}$$

Example Consider a situation like in Figure 2.8. There are 4 different capacity situations described in Table 2.2. It is assumed that CI = 1 (no uncertainly).

Table 2.2: Example of different capacity reductions

Start Time	End Time	% of remaining capacity
0	2000	100%
2000	3000	50%
3000	4000	75%
4000	...	100%

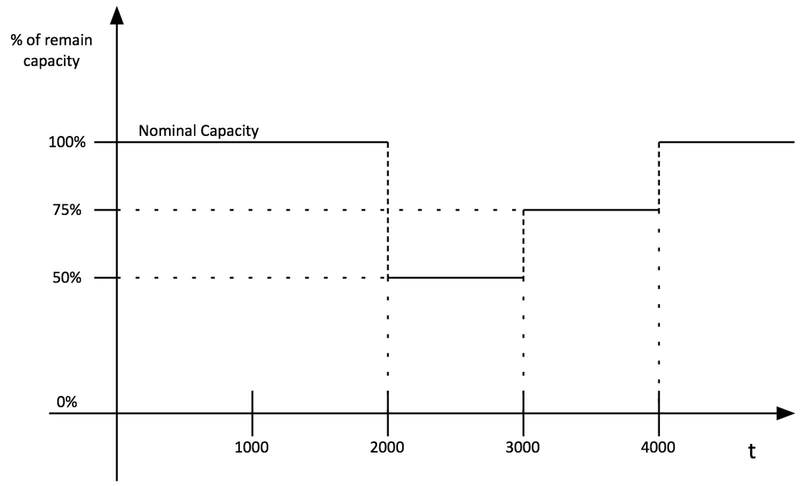


Figure 2.8: Example of different capacity reductions

Hotspot During a CCS it can happen that the traffic demand might be larger than the available capacity. These conditions would not have happened under normal situation (in this research it is assumed that the demand and capacity were already balanced before the CCS) and this particular state is called hotspot (see Fig. 2.9). In a single CCS there may be multiple hotspots.

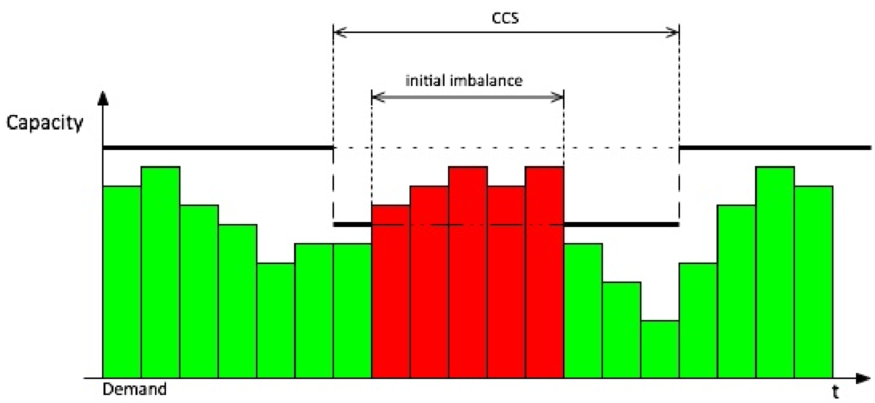


Figure 2.9: Example of a hotspot

The excess demand during hotspots must be managed and spread (by applying delays to individual flights) to handle all the requests for slot allocation. There are therefore two different periods during hotspots [22]:

1. **Stress Period**: period of time in which there is an imbalance between demand and capacity (there is an excess of demand) and the trend of the average of the delay is growing.
2. **Recovery Period**: period of time in which there is a free slot where the last flight of the excess demand can be allocated and therefore the sector is no longer under stress (the excess demand is fully absorbed).

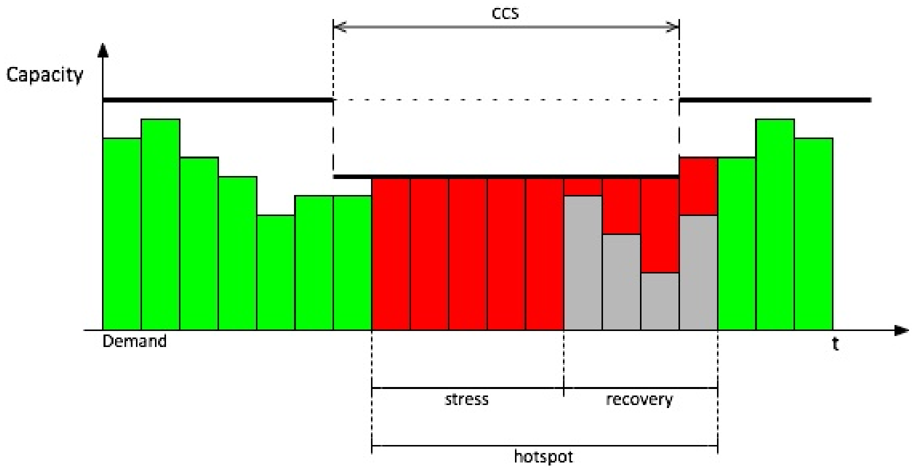


Figure 2.10: Stress and Recovery Period of a hotspot

It can be considered also a situation with multiple hotspots. These can be included in one or more CSSs, as it can be seen in Figure 2.11. In this research only single hotspot will be considered. Furthermore will not be studied the potential interdependencies between hotspots.

There are three different connections between hotspots and CSSs:

(A) There may be two or more hotspots in the same CCS;

(B) There may be two or more hotspots in different CSSs in the same geographical area (eg. the same track, the same airport, ...).

(C) There may be two or more hotspots in different CCSs in different territorial areas (eg. two or more different airports).

Suspension One of the basic SFP procedure tasks is the suspension of a flight. An airline can decide to increase the delay of one of its flights until the end of the hotspot area; for each suspended flight, the flight operator can manage 100 credits for prioritization rights.

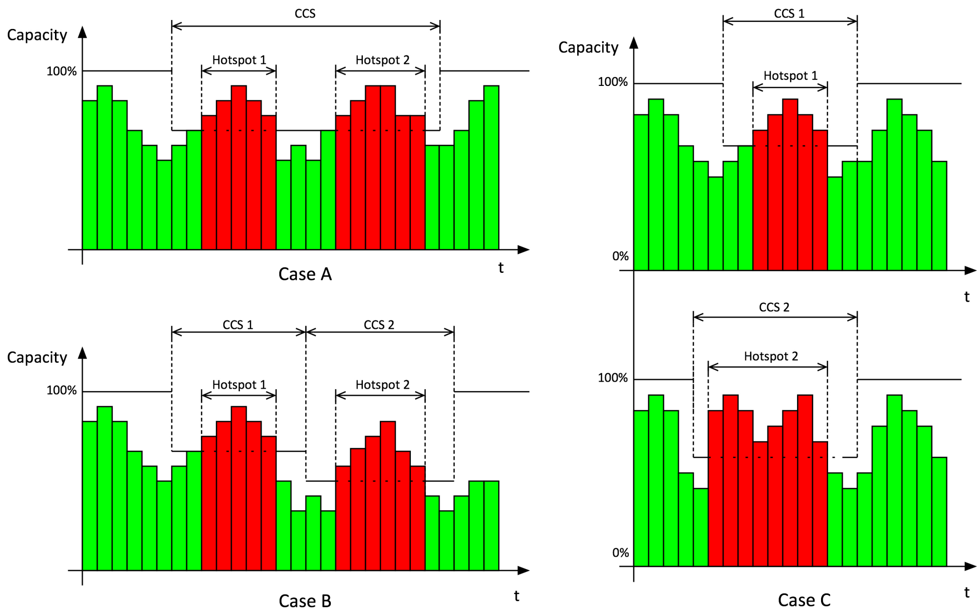


Figure 2.11: Different cases of Hotspots

Protection With the credits obtained by the suspension of flights, the airline may decide to reduce to the minimum the delay for a few selected flights (that might be tactically more important for the operator), which will be defined protected. The price of the delay reduction is based on the system's congestion level.

The major characterization indexes of a hotspot Through three indices it can be studied the hotspot severity and the possible actions that the user can engage in the system.

The **Operating Index (OI)** represents the severity of a hotspot for demand and capacity.

$$OI_{Hotspot} = \frac{100 \cdot Demand_{Hotspot}}{Capacity_{Hotspot}} \tag{2.1}$$

The number of **credits required** for protect a flight is:

$$credits \quad requirement \quad = \quad OI \quad - \quad 100 \tag{2.2}$$

Thus the maximum number of flights that an AU can protect with each suspension (MFP) is

$$MFP \quad = \quad \left\lfloor \frac{OI}{credits \quad requirement} \right\rfloor \tag{2.3}$$

The flight delay can be calculated in two different ways (see Fig. 2.12):

- from the flight actual time request;

- from the flight slot time request.

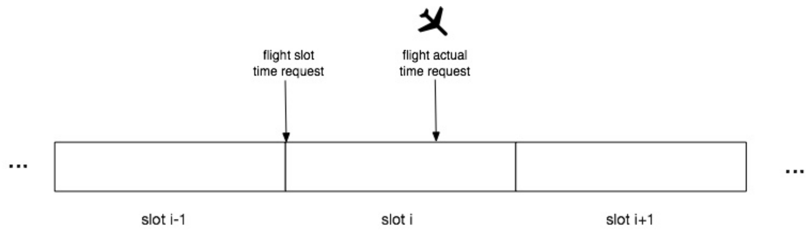


Figure 2.12: Different time requests

In this research it is assumed that a flight whose its initial scheduling time is in the middle of a slot due to a capacity reduction, it can be associated to that slot with a delay of zero minutes (no negative delays). For example in the Fig. 2.12 the considered flight will be allocated in the slot i.

These two calculation methods lead to different consequences regarding the change of the total sum of the minutes delay of the flights, explained in Theorems 2.4.1, 2.4.2, 2.4.3, 2.4.4 and 2.4.5.

Theorem 2.5.1 *Consider a hotspot in a CCS in which flights $f_1, f_2, ..., f_n$ are involved. The sum of the delays for these flights does not change after the UDPP suspension of one or more flights if the delay is calculated using the slot time request.*

Proof Notice that the delay of a single flight is defined as the sum of the actual flight arrival time minus its scheduled arrival (slot) time so:

$$\sum_{i=1}^{n} delay(i) = \sum_{i=1}^{n} (Time_{actual} - Time_{planned}) \tag{2.4}$$

Consider the two different part that form the delay

$$\sum_{i=1}^{n} delay(i) = \sum_{i=1}^{n} (Time_{actual}) - \sum_{i=1}^{n} (Time_{planned}) \tag{2.5}$$

The summation of the planned flight (slot) time doesn't change after a suspension because it is constant in time. Instead the sort of the actual flight times is different but the sum remains unchanged. So the final sum of the delays remains the same after a suspension procedure.

Example Consider the following example. Such scenario is characterized by a set of 9 flights, $F = \{A, B, C, D, E, F, G, H, I\}$, that configures the slot demand over a period of $t = [1, 9]$ with a capacity reduction of the 50% for the period (slot size of 2 minutes). In this example the sum of all the total delays in FPFS scheduling is 40 minutes (see equations 2.6 and 2.7). In Figure 2.13 is shown the different flights slots allocation between the initial situation and FPFS scheduling.

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Σ
Initial alloc.	A	B	C	D	E	F	G	H	I										
FPFS alloc.	A		B		C		D		E		F		G		H		I		
Delay	0		+2		+2		+4		+4		+6		+6		+8		+8		+40

Figure 2.13: Initial condition - example 1

$$\sum_{i=1}^{9} delay(i) \;=\; d_A \;+\; d_B \;+\; ... \;+\; d_I \;=\; 0 \;+\; 2 \;+\; ... \;+\; 8 \;=\; 40 \tag{2.6}$$

$$\begin{aligned} \sum_{i=1}^{9} delay(i) = & \sum_{i=1}^{9} (Time_{actual}(i) - Time_{planned}(i)) \;= \\ & \sum_{i=1}^{9} Time_{actual}(i) - \sum_{i=1}^{9} Time_{planned}(i) = \\ & (Time_{actual}(A) + Time_{actual}(B) + ... + Time_{actual}(I)) - \\ & (Time_{planned}(A) + Time_{planned}(B) + ... + Time_{planned}(I)) = \\ & (1 + 3 + ... + 17) - (1 + 1 + ... + 9) = 81 - 41 = 40 \end{aligned} \tag{2.7}$$

The user decides to suspend flight B. In the Fig. 2.14 is shown how all the flights are affected by this decision. The equations 2.8 and 2.9 point out that the sum of the total delay is unchanged after the suspension procedure.

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Σ
Initial alloc.	A	B	C	D	E	F	G	H	I										
SFP alloc.	A		C		D		E		F		G		H		I		B		
Delay	0		0		+2		+2		+4		+4		+6		+6		+16		+40
Diff.	0		-2		-2		-2		-2		-2		-2		-2		+14		0

Figure 2.14: Suspension condition - example 1

$$\sum_{i=1}^{9} delay^{'}(i) \;=\; d_A{}^{'} \;+\; d_B{}^{'} \;+\; ... \;+\; d_I{}^{'} \;=\; 0 \;+\; 16 \;+ 0 \;+\; ... \;+\; 6 \;=\; 40 \tag{2.8}$$

$$\sum_{i=1}^{9} delay'(i) = \sum_{i=1}^{9}(Time_{actual}{}'(i) - Time_{planned}{}'(i)) =$$
$$\sum_{i=1}^{9} Time_{actual}{}'(i) - \sum_{i=1}^{9} Time_{planned}{}'(i) =$$
$$(Time_{actual}{}'(A) + Time_{actual}{}'(B) + ... + Time_{actual}{}'(I)) -$$
$$(Time_{planned}{}'(A) + Time_{planned}{}'(B) + ... + Time_{planned}{}'(I)) =$$
$$(1 + 17 + 3 + ... + 15) - (1 + 1 + ... + 9) = 81 - 41 = 40 \tag{2.9}$$

In this example it can bee seen that the order of flights can be changed while delay remains unchanged.

Theorem 2.5.2 *Consider a hotspot in a CCS in which flights $f_1, f_2, ..., f_n$ are involved. The sum of the delays for these flights does not change after the UDPP protection of one or more flights if the delay is calculated using the slot time request.*

Proof The proof is analogous to that of Theorem 2.5.1

Example Continuing the previous example, the user decides to protect the flight I (see Figure 2.15 and equations 2.10 and 2.11).

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	∑
Initial alloc.	A	B	C	D	E	F	G	H	I										
SFP alloc.	A		C		D		E		I		F		G		H		B		
Delay	0		0		+2		+2		0		+6		+6		+8		+16		+40
Diff.	0		-2		-2		-2		-8		0		0		0		+14		0

Figure 2.15: Protection condition - example 1

$$\sum_{i=1}^{9} delay'(i) = d_A{}' + d_B{}' + ... + d_I{}' = 0 + 16 + 0 + ... + 0 = 40 \tag{2.10}$$

$$\sum_{i=1}^{9} delay''(i) = \sum_{i=1}^{9}(Time_{actual}{}''(i) - Time_{planned}{}''(i)) =$$
$$\sum_{i=1}^{9} Time_{actual}{}''(i) - \sum_{i=1}^{9} Time_{planned}{}''(i) =$$
$$(Time_{actual}{}''(A) + Time_{actual}{}''(B) + ... + Time_{actual}{}''(I)) -$$
$$(Time_{planned}{}''(A) + Time_{planned}{}''(B) + ... + Time_{planned}{}''(I)) =$$
$$(1 + 17 + 3 + ... + 9) - (1 + 1 + ... + 9) = 81 - 41 = 40 \tag{2.11}$$

It has obtained the same result than in previous example.

More generally it can be started:

Theorem 2.5.3 *Slot swapping and departure reordering techniques cannot decrease the overall delay (over slot time) of the traffic, but only to allocate the individual delays to flight in a different manner*[4].

Flight can use any instant of time within the slot allocated, not necessary the starting time of the slot. Due to that, the previous theorems must be modified partially if the delays are calculated instantaneously (i.e. using the flight slot time request).

Theorem 2.5.4 *Consider a hotspot in which flights $f_1, f_2, ..., f_n$ are involved. If the delay is calculated using the flight actual time request, the suspension of a flight can lead to an increase of one or more temporal units (the size of the slot) the total sum of the delays.*

Theorem 2.5.5 *Consider a hotspot in which flights $f_1, f_2, ..., f_n$ are involved. If the delay is calculated using the flight actual time request, the protection of a flight can lead to an increase of one or more temporal units (the size of the slot) the total sum of the delays.*

Only the flights suspended or protected are subject to the increase of delay; for all other planes, the operating times of the resource does not change.

These counterintuitive theorems can be observed in the examples in Figures 2.16, 2.17 and 2.18.

In this research the potential increase of actual delay as a consequence of the slot allocation processes will be considered negligible due to the order of magnitude.

[4]The flight delay is calculated using the slot time flight request.

Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
FLIGHTS	A	B	C	D	E	F	G								
FLIGHTS	A		B		C		D		E		F		G		
DELAY	0		1		2		3		4		5		6		21
FLIGHTS	A		C		D		E		F		G		B		
DELAY	0		0		1		2		3		4		11		21
FLIGHTS	B		C		D		E		F		G		A		
DELAY	0		0		1		2		3		4		12		22 (+1)
FLIGHTS	B		D		F		G		A		C		E		
DELAY	0		0		0		0		8		8		8		+24 (+3)

Figure 2.16: Changing of delay in suspension - slots size of 2

Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	SUM
FLIGHTS	A	B	C	D	E	F	G															
FLIGHTS	A			B			C			D			E			F			G			
DELAY	0			2			4			6			8			10			12			42
FLIGHTS	B			C			D			E			F			G			A			
DELAY	0			1			3			5			7			9			18			43 (+1)
FLIGHTS	A			C			D			E			F			G			B			
DELAY	0			1			3			5			7			9			17			42
FLIGHTS	C			D			E			F			G			A			B			
DELAY	0			0			2			4			6			15			17			44 (+2)
FLIGHTS	C			F			G			A			B			D			E			
DELAY	0			0			0			9			11			12			14			46 (+4)

Figure 2.17: Changing of delay in suspension - slots size of 3

Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
FLIGHTS	A	B	C	D	E	F	G								
FLIGHTS	A		B		C		D		E		F		G		
DELAY	0		1		2		3		4		5		6		21
FLIGHTS	A		C		D		E		F		G		B		
DELAY	0		0		1		2		3		4		11		21
FLIGHTS	A		C		E		D		F		G		B		
DELAY	0		0		0		3		3		4		11		21
FLIGHTS	A		C		F		D		E		G		B		
DELAY	0		0		0		3		4		4		11		22 (+1)
FLIGHTS	A		D		C		E		F		G		B		
DELAY	0		0		2		2		3		4		11		22 (+1)

Figure 2.18: Changing of delay in protection - slots size of 2

2.6 Selective Flight Protection optimization problem

This Selective Flight Protection optimization problem can be thought as a graph problem [17]. The entities that fill the system are the flights, slots and the costs of each flight for each slot. The entire system can be thought as a graph with two types of vertices: flights and slots. The weight of the arcs that connect that vertices are the costs[5] (see Fig. 2.19).

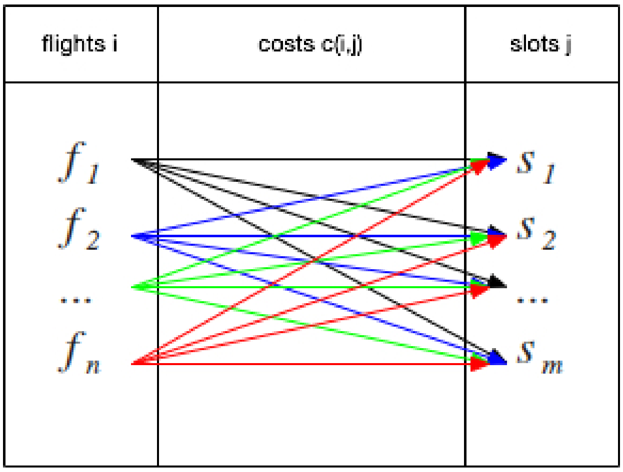


Figure 2.19: Graph of flights, slots and costs

[5] i.e. the cost $c_{i,j}$

Table 2.3: Initial arrival time - example 1

FLIGHTS	TIME
A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9

Each arc has a weight and the goal of the optimal procedure is to find the right set of arcs that minimizes costs for the entire system and connect each flight to a different slot. It is important to highlight that usually not all flights can be connected to every slots: this link depends on the flight arrival time and the slot actual time. This concept will be further discussed in the Chapter 3.

2.6.1 Example 1

Using the above representation, a scenario will be described in which 9 flights $F = \{A, B, C, D, E, F, G, H, I\}$ are involved over a period of $t = [1, 9]$ (see Fig. 2.20). Their scheduled time is in Table 2.3. Assume there is a capacity reduction of the 50% so the flights can no longer use the resources every minute. The operator, using SFP mechanism, decides to suspend the flight B and to reduce as much as possible the delay of the flight I.

<table>
<tr><th>T</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th></tr>
<tr><td>Initial alloc.</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>FPFS alloc.</td><td colspan="2">A</td><td colspan="2">B</td><td colspan="2">C</td><td colspan="2">D</td><td colspan="2">E</td><td colspan="2">F</td><td colspan="2">G</td><td colspan="2">H</td><td colspan="2">I</td></tr>
<tr><td>Delay</td><td colspan="2">0</td><td colspan="2">+1</td><td colspan="2">+2</td><td colspan="2">+3</td><td colspan="2">+4</td><td colspan="2">+5</td><td colspan="2">+6</td><td colspan="2">+7</td><td colspan="2">+8</td></tr>
</table>

Figure 2.20: Initial condition - example 1

In Figure 2.21, there is the graph representation of this problem example. The blue nodes are the flights (in the parenthesis there is the scheduled time) and the black nodes instead represent the slots (in the parenthesis there is the new allocated time).

In Table 2.4, Ts is the set of slots that flights are willing to accept in exchange for the actual slot.

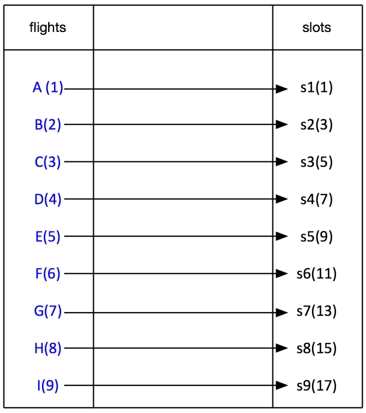


Figure 2.21: Initial condition graph - example 1

The value of the slots linked to each flight has the following bounds (the function $f : Slot(T)$ identifies the initial time of the slot related to the considered time T):

$$\forall\ flight\ i \quad Slot(T_{Initial\ Allocation}(i)) \leq Slot(Ts(i)) \leq Slot(T_{FPFS\ Allocation}(i)) \tag{2.12}$$

The highlighted green cells are the slots associated with the flights (the set of arcs), the black cells are the slots that can be feasibly allocated to each flight and the red ones with trimmed values are not possible links.

Table 2.4: Slot scheduling FPFS - example 1

Ts	SLOTS								
Ts(A)	1	~~3~~	~~5~~	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(B)	1	3	~~5~~	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(C)	~~1~~	3	5	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(D)	~~1~~	3	5	7	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(E)	~~1~~	~~3~~	5	7	9	~~11~~	~~13~~	~~15~~	~~17~~
Ts(F)	~~1~~	~~3~~	5	7	9	11	~~13~~	~~15~~	~~17~~
Ts(G)	~~1~~	~~3~~	~~5~~	7	9	11	13	~~15~~	~~17~~
Ts(H)	~~1~~	~~3~~	~~5~~	7	9	11	13	15	~~17~~
Ts(I)	~~1~~	~~3~~	~~5~~	~~7~~	9	11	13	15	17

After the decision of suspension of the flight B, there is a change in the order of the arcs and in the Ts table. It will be obtained the graphs in Figure 2.23 and the Table 2.22. It can be noticed that in the graph, arcs pointing up (in green) will be those that can bring a reduction of the flight delay and the ones pointing down (in red) will bring larger delay instead.

In Table 2.5 it can be seen how the slots, linked to all the flights, are affected by the decision of the suspension of the flight B.

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Initial alloc.	A	B	C	D	E	F	G	H	I									
SFP alloc.	A		C		D		E		F		G		H		I		B	
Delay	0		0		+1		+2		+3		+4		+5		+6		+15	

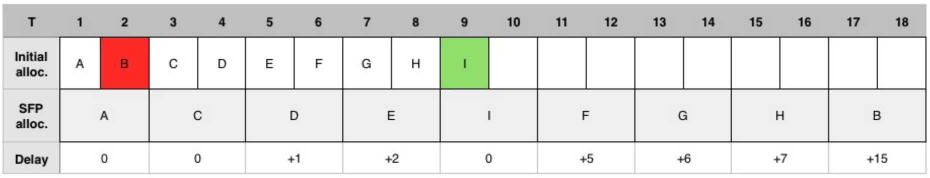
Figure 2.22: Suspension flight B - example 1

Figure 2.23: Suspension condition graph - example 1

Table 2.5: Suspension B - example 1

Ts	SLOTS								
Ts(A)	1	~~3~~	~~5~~	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(B)	SUSPENDED								17
Ts(C)	~~1~~	3	5	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(D)	~~1~~	3	5	7	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
Ts(E)	~~1~~	~~3~~	5	7	9	~~11~~	~~13~~	~~15~~	~~17~~
Ts(F)	~~1~~	~~3~~	5	7	9	11	~~13~~	~~15~~	~~17~~
Ts(G)	~~1~~	~~3~~	~~5~~	7	9	11	13	~~15~~	~~17~~
Ts(H)	~~1~~	~~3~~	~~5~~	7	9	11	13	15	~~17~~
Ts(I)	~~1~~	~~3~~	~~5~~	~~7~~	9	11	13	15	17

With the choice of protecting (i.e. reducing the delay) the flight I, there is another changing in the order of the edges in the graph (see Figs. 2.24 and 2.25). It is important to notice that there is not an increase for any delay of individual flight (compared to the initial situation), except for the suspended flight. Flights C,D and E are better off (less delay), while F,G and H are neutral to the protection decisions over flights B and I.



T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Initial alloc.	A	B	C	D	E	F	G	H	I									
SFP alloc.	A		C		D		E		I		F		G		H		B	
Delay	0		0		+1		+2		0		+5		+6		+7		+15	

Figure 2.24: Protection of flight I - example 1

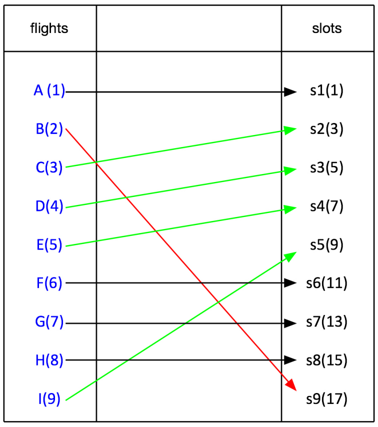


Figure 2.25: Protection condition graph - example 1

Table 2.6: Protection of flight I - example 1

TS	SLOTS								
TS(A)	1	~~3~~	~~5~~	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
TS(B)	SUSPENDED								17
TS(C)	~~1~~	3	5	~~7~~	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
TS(D)	~~1~~	3	5	7	~~9~~	~~11~~	~~13~~	~~15~~	~~17~~
TS(E)	~~1~~	~~3~~	5	7	9	~~11~~	~~13~~	~~15~~	~~17~~
TS(F)	~~1~~	~~3~~	5	7	9	11	~~13~~	~~15~~	~~17~~
TS(G)	~~1~~	~~3~~	~~5~~	7	9	11	13	~~15~~	~~17~~
TS(H)	~~1~~	~~3~~	~~5~~	7	9	11	13	15	~~17~~
TS(I)	~~1~~	~~3~~	~~5~~	~~7~~	9	11	13	15	17

The SFP mechanism can be thought as the problem of finding the right set of weighed arcs that minimizes the cost of the system and that associates

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13	12.14
Initial alloc.	A	B	C	D	E	F	G	H	I						
FPFS alloc.	A		B		C		D		E		F		G	H	I
Delay	0		+1		+2		+3		+4		+5		+6	+6	+6

Figure 2.26: FPFS scheduling - example 2

each flight to a different slot. This problem can be solved through a linear programming model.

2.6.2 Example 2

In this second example there is an approach to a more realistic case in terms of actor and SFP usage.

Consider a situation of 9 flights: $F = \{A, B, C, D, E, F, G, H, I\}$. The initial scheduling of these flights is in the Table 2.7.

Table 2.7: Initial arrival time - example 2

FLIGHTS	TIME
A	12.00
B	12.01
C	12.02
D	12.03
E	12.04
F	12.05
G	12.06
H	12.07
I	12.08

The system has subsequently a reduction of 50% of the slot capacity from 12.00 to 12.12. The FPFS scheduling is shown in Figure 2.26. This reduction in capacity creates a hotspot where demand is greater than the actual capacity of the system and all flights increase their delay. Using the definitions in (2.4), (2.5) and (2.6):

$$\begin{array}{rlcc} & OI & = & 166.\bar{6} \\ credits & requirement & = & 66.\bar{6}\ \text{OC} \\ & MFP & = & 2 \end{array}$$

OI has been calculated as follows:

$$OI_{Hotspot} = \frac{100 \cdot Demand_{Hotspot}}{Capacity_{Hotspot}} = \frac{100 \cdot 15}{9} = 166.\bar{6} \tag{2.13}$$

In this case with a single suspension two flights can be protected.

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13	12.14
Initial alloc.	A	B	C	D	E	F	G	H	I						
SFP alloc.	A		C		D		E		F		G		H	I	B
Delay	0		0		+1		+2		+3		+4		+5	+5	+12
Diff.	0		-2		-2		-2		-2		-2		-1	-1	+12

Figure 2.27: Suspension of flight B - example 2

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13	12.14
Initial alloc.	A	B	C	D	E	F	G	H	I						
SFP alloc.	A		C		E		D		I		F		G	H	B
Delay	0		0		0		+3		0		+5		+6	+6	+12
Diff.	0		-2		-4		0		-6		0		0	0	+12

Figure 2.28: Protection of flight E and I - example 2

The operator may decide to suspend the flight B (Fig. 2.27) and to protect two flights: E and I (Fig. 2.28).

In this example it can be immediately noticed that the flight C, which does not participate in UDPP, gets an improvement of its initial delay. All delays of other flights remain neutral and it is important to note that the delays for flights that do not participate do not increase. SFP mechanism allows a higher flexibility in scheduling during a hotspot for the AUs.

Chapter 3

Models for making optimal decision in SFP context

In this chapter the mathematical formulation to find optimal solutions in the context of SFP will be described. Two different case are considered:

- Single User Single Hotspot (SUSH): it is assumed that there is a single operator that can manage all the flights in the hotspot using the SFP procedure. The operator can choose which flights to suspend and which ones to protect in order to find the best solution for the reduction of the costs. Analyzing this case will be useful to find the best social/global optimal solution in terms of total cost of flights.
- Multiple Users Single Hotspot (MUSH): there are many airspace users in a hotspot, each one with a particular set of flights. Analyzing this case will be useful to reproduce a more realistic environment in which each AU tries to optimize its own operations without regarding the wellness of other AUs.

3.1 Single User Single Hotspot

The optimization model for the User-Driven Prioritisation Process (UDPP) in the particular case which there is a single moderator in a hotspot is described in this section. The manager of the network wants to find the best solution of **minimum costs for the entire system**, without differentiation between the various airlines.

3.1.1 Example of the UDPP - Single User Single Hotspot

This same example will be considered also in the chapter 3.2.1 for the case of Multiple Users.

Consider a situation where 7 flights $F = \{A, B, C, D, E, F, G\}$ are involved. The initial scheduling of these flights is represented in the Table 3.1.
The system has subsequently a reduction of 50% of the sector capacity. This reduction in capacity generates a hotspot where demand is larger than the actual

capacity of the sector and all flights increase their delay as a consequence (see Fig. 3.1).

Table 3.1: Initial arrival time - example 3

FLIGHTS	TIME
A	12.00
B	12.01
C	12.02
D	12.03
E	12.04
F	12.05
G	12.06

Using the previous definitions (2.1, 2.2 and 2.3) it will be get:

$$\begin{array}{rcl} OI & = & 200 \\ credits\ \ requirement & = & 100\ \mathrm{OC} \\ MFP & = & 2 \end{array}$$

In this particular case, if the user suspends one flight, he can protect up two flights.

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13
Initial alloc.	A	B	C	D	E	F	G							
FPFS alloc.	A		B		C		D		E		F		G	
Delay	0		+1		+2		+3		+4		+5		+6	

Figure 3.1: FPFS scheduling - example 3

The unique user decides to suspend the flight B (see Fig. 3.2).

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13
Initial alloc.	A	B	C	D	E	F	G							
SFP alloc.	A		C		D		E		F		G		B	
Delay	0		0		+1		+2		+3		+4		+11	
Diff.	0		-2		-2		-2		-2		-2		+10	

Figure 3.2: Suspension of flight B - example 3

After the suspension of B, the operator decides to protect the flights E and G (see Fig. 3.3) .

It can be immediately noticed in this example that the flight C, while not participating in UDPP, gets an improvement of its initial delay. Furthermore, the flight G, even if protected, can not reduce its delay up to zero minutes (since the flights not suspended can not increase their delay compared to the FPFS

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09	12.10	12.11	12.12	12.13
Initial alloc.	A	B	C	D	E	F	G							
SFP alloc.	A		C		E		D		G		F		B	
Delay	0		0		0		+3		+2		+5		+11	
Diff.	0		-2		-4		0		-2		0		+10	

Figure 3.3: Protection of flight E and G - example 3

allocation, G can not be allocated to the slot that start at 12.06 because it is the unique possibile scheduling choice for flight D).

3.1.2 Optimization model of the UDPP - SUSH

The optimization model of the problem is necessary to show what is the best sequence solution, that is, which flights to protect and which ones to suspend in order to get the lowest total cost.

3.1.2.1 Initial data

Consider a system formed by $f_1, f_2, ..., f_n$ flights which want to use $s_1, s_2, ..., s_m$ resources (slots).

Each flight $f_1, f_2, ..., f_n$ is characterized by:

- $CMD_1, CMD_2, ..., CMD_n$: the different cost for the first minute of delay;
- $TSched_1, TSched_2, ..., TSched_n$: the scheduled arrival time for each flight;
- $TActual_1, TActual_2, ..., TActual_n$: the actual time to use the slot according to the FPFS.
- $TTarget_1, TTarget_2, ..., TTarget_n$: the slot associated to the scheduled arrival time for each flight;

Each slot $s_1, s_2, ..., s_m$ has:

- $SSize$: the duration (or size) of a slot (assumed the same for all slots);
- $Slot_1, Slot_2, ..., Slot_m$: the slot start time.

One of the most important indicators for the entire model is the Operating Index (OI) which provides representation of the severity of the imbalance of the hotspot is defined as:

$$\begin{aligned} OI_{Hotspot} &= \frac{100 \times \sum (Demand \quad of \quad flights \quad during \quad the \quad Hotspot)}{\sum (Capacities \quad during \quad the \quad Hotspot)} \\ &= 100 \frac{Demand}{Capacity} \end{aligned} \tag{3.1}$$

Through OI index, it can be found the number of credits required to protect a flight with the equation (2.2) and also to obtain the max number of flights that can be protected from a single suspension MFP.

Each flight has a different delay for every sequence position in the hotspot. The matrix $d(i,j)$ represent these values.

$$\underset{i\ in\ 1..n,\ j\ in\ 1..m}{d(i,j)} \quad = \quad delay \quad for \quad flight \quad f_i \quad in \quad slot \quad s_j \tag{3.2}$$

3.1.2.2 The decisional variables

The binary decisional variable is $x(i,j)$.

$$\underset{i\ in\ 1...n,\ j\ in\ 1...m}{x(i,j)} \quad = \quad \begin{cases} 1 & if \quad flight \quad f_i \quad use \quad slot \quad s_j \\ 0 & otherwise \end{cases} \tag{3.3}$$

All the flights can be divided in *normal* $[n]$, *protected* $[p]$ and *suspended* $[s]$ ones so three new binary variables are introduced.

$$\sum flights \quad = \quad \begin{cases} normal \quad flights \\ protected \quad flights \\ suspended \quad flights \end{cases} \tag{3.4}$$

3.1.2.3 The objective function

The objective function of the model is to minimize the costs for all the AUs.

$$minimize \qquad \sum_{i\ in\ 1...n,\ j\ in\ 1...m} x(i,j) \cdot d(i,j) \cdot f(d(i,j)) \cdot CMD(i) \tag{3.5}$$

$f(d(i,j))$ is a function used to calculate the flight cost due to the delay. In the implementation of the simulator it will be used a pseudo-real cost function found through Cook, Tanner, Jovanovic, Lawes studies about the cost of delay to air transport in Europe [19] (see Section 5.1.2).

3.1.2.4 The constraints

Constraint c1: "Each flight must have one and only one slot allocated".

$$\underset{i\ in\ 1..n}{\forall i} \sum_{j\ in\ 1..m} x(i,j) \;=\; 1 \tag{3.6}$$

Constraint c2: "Each slot cannot be allocated to more than a single flight".

$$\underset{j\ in\ 1..m}{\forall j} \sum_{i\ in\ 1..n} x(i,j) \;\leq\; 1 \tag{3.7}$$

Constraint c3: "Each flight is *normal* $[n]$ or *protected* $[p]$ or *suspended* $[s]$".

$$\underset{i\ in\ 1..n}{\forall i} \; n(i) + p(i) + s(i) = 1 \tag{3.8}$$

Constraint c4: "The delay of each flight, except for the suspended ones, must be between 0 minutes and the FPFS scheduling delay".

$$\begin{aligned} &if\ delay_{actual}(i)\ >\ delay_{FPFS}(i)\ \Rightarrow\ x(i,j)\ \leq\ s(i) \\ &for\ every\ i\ in\ 1...n,\ j\ in\ 1...m \end{aligned} \tag{3.9}$$

Constraint c5: "The suspended flights are scheduled in the last position of the hotspot".
In order to highlight which are the suspended flights it will be checked if the flight increases its delay more than the FPFS baseline delay. If so, it must be a suspended flight.

$$\begin{aligned} &if\ delay_{actual}(i)\ >\ delay_{FPFS}(i)\ \Rightarrow\ s(i)\ =\ 1 \\ &for\ every\ i\ in\ 1...n \end{aligned} \tag{3.10}$$

There is a vector r indicating the possible position of the flights suspended at the end of the optimization. It will be formed as follows: $r : [0, ..., 0, 1, ..., 1]$ (suspended flights can be placed only on the last hotspot position). The number of 1 in the vector is equal to the sum of flights suspended in the sector.

$$\begin{aligned} &if\ s(i)\ =\ 1\ \Rightarrow\ x(i,j)\cdot s(i)\cdot r(j)\ =\ 1 \\ &for\ every\ i\ in\ 1...n,\ j\ in\ 1...m \end{aligned} \tag{3.11}$$

Constraint c6: "The number of credits for every operator cannot be negative (i.e., proportional effort is required first before any flight protection)".

$$Cr \geq 0 \tag{3.12}$$

Constraint c7: "The protected flights will be scheduled in their best possible position (least possible delay)".

To find out which flights are protected compared to non-prioritized flights, it will be introduced a new vector h. It represents the amount of delay improved due to the suspension for the individual flights. In the following example there will be a practical representation of how the vector h is made and how it behaves due to flight suspensions.

Example Consider a case in which 5 flights $F = \{A, B, C, D, E\}$ are involved in a situation of 50% reduction of sector capacity. In Figure 3.4 is shown the FPFS scheduling.

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09
Initial alloc.	A	B	C	D	E					
FPFS alloc.	A		B		C		D		E	
Delay	0		+1		+2		+3		+4	

Figure 3.4: FPFS scheduling - example 4

The operator decides to suspend the flights B and D, as shown in Fig. 3.5.

T	12.00	12.01	12.02	12.03	12.04	12.05	12.06	12.07	12.08	12.09
Initial alloc.	A	B	C	D	E					
SFP alloc.	A		C		E		B		D	
Delay	0		0		0		+4		+4	
Diff.	0		-2		-4		+4		+2	

Figure 3.5: Suspension of B and D - example 4

The vector h, that represents in a system without protection the change of the non-prioritized flight delay, in this example is: $h : [0, s, 2, s, 4]$ (the letter s represents the suspended flights).
The non-prioritized flights in this particular instance are A, C and E, respectively the first, the third and the fifth position of the vector h.

If a flight has a larger reduction of the delay than the corresponding value in h, it has been protected: its delay reduction is larger than the non-prioritized flights reduction.

$$\begin{aligned} &if(-\Delta_{delay}(i) \; > \; h(i)) \; \Rightarrow \; p(i) = 1 \\ &for \; every \; i \; in \; 1...n \end{aligned} \tag{3.13}$$

3.1.2.5 The definitions

Definition d1: with a flight suspension, the user gets 100 OC and these credits are included in its total budget.

$$Budget \; = 100 \; \cdot \sum_{i \; in \; 1,...,n} s(i) \tag{3.14}$$

Definition d2: The amount of credits to protect the flights are:

$$Credits \; spent \; = [OI \; - \; 100] \; \cdot \sum_{i \; in \; flights} p(i) \tag{3.15}$$

These credits can be used in order to prioritize certain flights.

Definition d3: d1 and d2 can be combined to calculate the residual value of the user's credits .

$$Cr \; = 100 \; \cdot \sum_{i \; in \; flights} s(i) \; - [OI \; - \; 100] \; \cdot \sum_{i \; in \; flights} p(i) \tag{3.16}$$

3.1.2.6 The rules

Rules r1: AUs can suspend flights at any place in the sector;

Rules r2: AUs can protect flights at any place in the sector, irrespective of whether there has been a previous suspension;

3.1.2.7 Example

Using the above representation of the model, a numerical example is described below.

Consider a hotspot in which 8 flights are involved. Due to constraints 3.8 and 3.9, flights can be allocated only in slots delimited between the slot request time and final time achieved by FPFS scheduling. The Fig. 3.6 shows a table where the value 1 feasibility of flight(i) to use slot(j), otherwise it is 0.

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	1	0	0	0	0	0	0	0
Flight 2	1	1	0	0	0	0	0	0
Flight 3	0	1	1	0	0	0	0	0
Flight 4	0	1	1	1	0	0	0	0
Flight 5	0	0	1	1	1	0	0	0
Flight 6	0	0	0	1	1	1	0	0
Flight 7	0	0	0	1	1	1	1	0
Flight 8	0	0	0	0	1	1	1	1

Figure 3.6: The possibilities of flights to use slots - example 5

Consider that flight 1 and flight 3 are suspended (i.e., sent to the slot 8 and 7 respectively). Figure 3.7 shows the allocation (green circles) for the two flights suspended (highlighted in red).

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	1	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	1	1	0	0	0	0
Flight 5	0	0	1	1	1	0	0	0
Flight 6	0	0	0	1	1	1	0	0
Flight 7	0	0	0	1	1	1	1	0
Flight 8	0	0	0	0	1	1	1	1

Figure 3.7: The possibilities of flights to use slots after suspension - example 5

Using the bond that links flights and slots (c1 and c2) and taking advantage of the constraint propagation, the results of the Figures 3.8 can be found.
Note that after the constraint propagation, some slot allocations are already determined, while other (in the yellow rectangle) need an extra search to find the optimal combination. Figure 3.9 shows the updated values once the constraint has been applied: the slots that are not feasible have been set to 0.

If the SFP procedure would end without prioritizations (the FPFS policy would be applied), it would lead to the result shown in Fig: 3.10; instead if the operator decided to protect a flight (flight 7), then the flight will get the slot with the minimum possible delay (see Fig. 3.11).

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	~~1~~	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	~~1~~	~~1~~	0	0	0	0
Flight 5	0	0	1	~~1~~	~~1~~	0	0	0
Flight 6	0	0	0	1	1	1	0	0
Flight 7	0	0	0	1	1	1	~~1~~	0
Flight 8	0	0	0	0	1	1	~~1~~	~~1~~

Figure 3.8: Applying constraints after the suspension - example 5

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	0	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	0	0	0	0	0	0
Flight 5	0	0	1	0	0	0	0	0
Flight 6	0	0	0	1	1	1	0	0
Flight 7	0	0	0	1	1	1	0	0
Flight 8	0	0	0	0	1	1	0	0

Figure 3.9: Final table of the feasible allocations for flights after suspensions - example 5

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	0	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	0	0	0	0	0	0
Flight 5	0	0	1	0	0	0	0	0
Flight 6	0	0	0	1	1	1	0	0
Flight 7	0	0	0	1	1	1	0	0
Flight 8	0	0	0	0	1	1	0	0

Figure 3.10: Final allocation without protections - example 5

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	0	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	0	0	0	0	0	0
Flight 5	0	0	1	0	0	0	0	0
Flight 6	0	0	0	~~1~~	1	1	0	0
Flight 7	0	0	0	1	~~1~~	~~1~~	0	0
Flight 8	0	0	0	0	1	1	0	0

Figure 3.11: Final allocation with protections - example 5

Finally in the Figure 3.12 is shown the value of the decisional variable $x(i, j)$ after allocating all the slots to flights and after updating the value of the cells corresponding to non-allocated flight-slot pairs to zero. It can be seen that there is only one cell active (i.e. set to 1) in every row and in every column: each flight is associated with one and only one slot and vice versa.

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Flight 1	0	0	0	0	0	0	0	1
Flight 2	1	0	0	0	0	0	0	0
Flight 3	0	0	0	0	0	0	1	0
Flight 4	0	1	0	0	0	0	0	0
Flight 5	0	0	1	0	0	0	0	0
Flight 6	0	0	0	0	1	0	0	0
Flight 7	0	0	0	1	0	0	0	0
Flight 8	0	0	0	0	0	1	0	0

Figure 3.12: Final values of the decisional variables x(i, j) - example 5

3.1.2.8 Other protection issues

Continuing with the previous graphical representation, a more general case will be considered. In the Figure 3.13 the initial FPFS scheduling is described for some flights during a hotspot.

	...	Slot i	Slot i+1	...	Slot j-1	Slot j	Slot j+1	...
...				...				
Flight k	0	1	1	1	1	0	0	0
Flight m	0	1	1	1	1	1	0	0
Flight n	0	1	1	1	1	1	1	0
...				...				

Figure 3.13: Initial FPFS Scheduling - Protection problem

In the Figure 3.14 there is the actual situation of the sector after some flights suspension.

	...	Slot i	Slot i+1	...	Slot j-1	Slot j	Slot j+1	...
...				...				
Flight k	0	1	1	1	1	0	0	0
Flight m	0	1	1	1	1	1	0	0
Flight n	0	1	1	1	1	1	1	0
...				...				

Figure 3.14: Scheduling after suspension - Protection problem

The Figures 3.15 and 3.16 show how the protect constrain behaves correctly and wrongly in several cases. The optimizer sometimes can find that the best solution for protecting a determinate flight is not allocating it in its first possible position. The solution is better or equal than the real SFP one with the right protection constraint. This not accurate result can be evaluated as an upper bound for the SFP mechanism.

	...	Slot i	Slot i+1	...	Slot j-1	Slot j	Slot j+1	...
...				...				
Flight k	0	1	1	1	1	0	0	0
Flight m	0	1	1	1	1	1	0	0
Flight n	0	1	1	1	1	1	1	0
...				...				

Figure 3.15: Right scheduling after protection - Protection problem

	...	Slot i	Slot i+1	...	Slot j-1	Slot j	Slot j+1	...
...				...				
Flight k	0	1	1	1	1	0	0	0
Flight m	0	$\boxed{1}$	1	1	1	1	0	0
Flight n	0	1	1	1	1	1	1	0
...				...				

Figure 3.16: Wrong scheduling after protection - Protection problem

3.1.3 Adaptation of the SUSH model for implementation

In order to make the implementation FICO Xpress-Optimizer has been used, a commercial optimization solver for linear programming (LP). The language of modeling of Xpress is Xpress-Mosel.

3.1.3.1 Linear programming problem

One of the first problems encountered during the implementation of the mathematical model was the need to linearize the non-linear constraints, in order to transform the problem studied in a **linear problem**[13],[10]. New decision variables were introduced to solve the linearization and it has made extensive use of the mathematical structure of the **big O**[1], that was used to manage all edge cases in the constraints.

The problematic polynomial constraints are for suspension c4 - 3.10 and c5 - 3.11 and for the prioritization c7 - 3.13.

Constraint c4 - 3.10 This first constraint highlights the suspended flights: if the flight initial delay increases then it is suspended.
During the implementation it will be created a new vector that represents the difference of the delay between the initial (FPFS) and the final situation.
Two new values will be introduced: very large $BigO$ ($\gg 1$) and a very little $SmallO$ ($0 < SmallO \ll 1$).

$$\begin{aligned} &\forall\, i\; in\; 1...n \\ &\qquad s(i) \;\geq\; -\frac{diffDelay(i)}{Big\,O} \;-\; Small\,O \\ &\qquad s(i) \;\leq\; (1 \;-\; Small\,O) \;-\; \frac{diffDelay(i)}{Big\,O} \end{aligned} \tag{3.17}$$

Constraint c5 - 3.11 The second constraint to linearize is used to place the flights suspended into the last positions of the hotspot. It can be seen that it is a multiplication of three decision variables.
Two new binary decision variables have been introduced in order to solve this polynomial problem: y and k. k represent the multiplication of $x \cdot r$. The constraint can then be represented by the following disequations:

$$\forall\, i\; in\; 1,...,n \quad y(i) \;\geq\; s(i) \tag{3.18}$$

$$\begin{aligned} &\forall\, i\; in\; 1,...,n \quad j\; in\; 1,...,m \\ &\qquad k(i,j) \;\geq\; x(i,j) \;+\; r(j) \;-\; 1 \\ &\qquad k(i,j) \;\leq\; \frac{x(i,j) \;+\; r(j)}{2} \end{aligned} \tag{3.19}$$

$$\forall\, i\; in\; 1,...,n \quad y(i) \;\leq\; \sum_{j:in\; 1,...,m} k(i,j) \tag{3.20}$$

[1]The formal explanation of the meaning of big O is in Appendix A at the end of this document

Constraint c7 - 3.13 The last problematic constraint that required linearization highlights the protected flights.
It's very important for the prioritization process to find which flights reduce their delays due to the suspensions compared which are protected ones. So it will be created the vector h that represents the amount of delay improved in individual flights thanks to each suspension.

$$\begin{gathered} h(1) \;=\; s(1) \,\cdot\, size\,of\,the\,slot \\ \\ \forall\, 2\; in\; 1,...,n \\ h(i) \;=\; h(i-1) \,+\, s(i) \,\cdot\, size\,of\,the\,slot \end{gathered} \tag{3.21}$$

It will be created a new vector Δ that represents the variation of the delay between FPFS and actual scheduling.

$$\begin{gathered} \forall\, i\; in\; 1,...,n \\ \Delta(i) = \sum_{j\; in\; 1,...,m} d(i,j) \cdot (x_{FPFS}(i,j) - x(i,j)) \end{gathered} \tag{3.22}$$

The constraint can be represented as follows:

$$\begin{gathered} \forall\, i\; in\; 1,...,n \\ p(i) \;\geq\; \frac{\Delta(i) \;-\; h(i)}{Big\;O} \end{gathered} \tag{3.23}$$

3.1.4 Illustrative instance of the UDPP - Single User Single Hotspot solved with Mosel

In this section it will be examined a single instance of the problem to illustrate the concept. In Section 5 larger number of instances with realistic traffic scenarios will be presented and the solution found will be studied through Monte-Carlo simulation and with a statistical approach to conclude on some underlying characteristics of the optimization model.

3.1.4.1 Initial data

This case considers 7 flights. Their arrival time is in Table 3.2

In the Table 3.3 it can be found the different initial cost per minute of delay for each flight.

The importance of a flight is assumed in this example as being directly proportional to the cost per minute of delay for the same flight. The most important flights are C, F and G, for which every minute of delay brings a higher cost to the system.

The sector has a reduction of 50% of the capacity. The scheduling, following the First Planned First Served rule, is represented in Fig. 3.17.

Table 3.2: Initial arrival time - instance SUSH

FLIGHTS	**TIME**
A	00
B	01
C	02
D	03
E	04
F	05
G	06

Table 3.3: Cost of a minute of delay - instance SUSH

FLIGHTS	**CMD (€)**
A	1
B	1
C	14
D	1
E	1
F	12
G	15

T	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**	**1**	**2**	**3**
Initial alloc.	A	B	C	D	E	F	G							
FPFS alloc.	A		B		C		D		E		F		G	
Delay	0		+1		+2		+3		+4		+5		+6	

Figure 3.17: FPFS scheduling - instance SUSH

3.1.4.2 Results

In Figure 3.18 it can been seen a screenshot with the output of the the results after the computation with FICO Xpress. According to the optimization model, it is recommended to the operator to suspend flights B and D and to protect flights F and G carrying a huge reduction of the costs in comparison with the original schedule sequence (186€ to 22€). It is important to notice that the flight C that does not participate to the SUSH mechanism, decreases the delay due to the flight B suspension.

```
****************FPFS****************
Total Cost: 186
Slot 1 - Time 0 - Flight A(delay 0 minutes)
Slot 2 - Time 2 - Flight B(delay 1 minutes)
Slot 3 - Time 4 - Flight C(delay 2 minutes)
Slot 4 - Time 6 - Flight D(delay 3 minutes)
Slot 5 - Time 8 - Flight E(delay 4 minutes)
Slot 6 - Time 10 - Flight F(delay 5 minutes)
Slot 7 - Time 12 - Flight G(delay 6 minutes)
************************************

****************UDPP****************
Total Cost: 22
Slot 1 - Time 0 - Flight A (delay 0 minutes)
Slot 2 - Time 2 - Flight C (delay 0 minutes)
Slot 3 - Time 4 - Flight F (delay 0 minutes)
Slot 4 - Time 6 - Flight G (delay 0 minutes)
Slot 5 - Time 8 - Flight E (delay 4 minutes)
Slot 6 - Time 10 - Flight D (delay 7 minutes)
Slot 7 - Time 12 - Flight B (delay 11 minutes)
************************************
PRIORITIZED FLIGHTS: F G
************************************
SUSPENDED FLIGHTS: B D
************************************
```

Figure 3.18: FICO Xpress results - instance SUSH

T	0	1	2	3	4	5	6	7	8	9	0	1	2	3
Initial alloc.	A	B	C	D	E	F	G							
SFP alloc.	A		C		F		G		E		D		B	
Delay	0		0		0		0		+4		+7		+11	
Diff.	0		-2		-6		-6		0		+4		+10	

Figure 3.19: SUSH scheduling - instance SUSH

3.1.5 Final thoughts on UDPP - SUSH

It is interesting to note that, in the various cases tested, it was found that the unique user of SUSH is often led to suspend more flights than to protect others flights. This is due to the fact that a suspension in early positions of the sequence generate a positive impact in the rest of the flights in the hotspot, thus reducing their delay (i.e., complex dynamics were identified). Another curiosity/consequence of the presence of complex dynamics is that sometimes the proposed solutions suggested by the optimization model may ask the user to protect ”important flights” (those with a large CMD) but not necessarily the ”most important” ones.

3.2 Multiple Users Single Hotspot

In this section it will be described the optimization model for the User - Driven Prioritization Process (UDPP) in the particular case in which there are many AUs in a single hotspot. This configuration of the model is more representative of the real scenarios in which SFP will be applied, under the assumption of the de-centralized and collaborative decision making process.

3.2.1 Example of the UDPP - Multiple User Single Hotspot

This example will show how the relations between two users in a MUSH instance are different and more complex compared to the previous case considered (SUSH) in the example 3.1.1. Consider then the situation in which 7 flights of two different airlines $F = \{A, B, C, D, E, F, G\}$ are involved. The initial scheduling of these flights is in the Table 3.4. In this case the considered user (shadowed in blue) only owns 3 flights: $\{B,\ E,\ F\}$. In Table 3.4, consequently, the user cannot apply prioritisation decision on flights A, C, D and G which belongs to other users (MUSH problem).

Table 3.4: Initial arrival time - example 6

FLIGHTS	TIME
A	12.00
B	12.01
C	12.02
D	12.03
E	12.04
F	12.05
G	12.06

Let us consider a reduction of 50% of the sector capacity. This reduction capacity causes a hotspot where demand is larger than the actual capacity of the system. All flights invalued are delayed as a consequence. (see Table 3.5).

It is assumed that flight F has a relative large cost minute of delay. In order to improve the situation for the airline costs it is proposed to suspend a flight with lower priority to be able to protect F (see Table 3.6).

Now the user can protect flight F with the purpuse to reduce as much as possible its delay (Table 3.7). Some flights, which do not participate in UDPP-SFP, can also reduce (indirectly) their delay, or in the worst-case, keep the same delay with respect the FPFS scheduling.

Table 3.5: FPFS - example 6

FLIGHTS	FPFS TIME	INCREASE THE DELAY
A	12.00	0
B	12.02	+1
C	12.04	+2
D	12.06	+3
E	12.08	+4
F	12.10	+5
G	12.12	+6

Table 3.6: Suspension flight B - example 6

FLIGHTS	TIME	DELAY	DIFFERENCE FPFS
A	12.00	0	0
B	12.12	+11	+10
C	12.02	0	-2
D	12.04	+1	-2
E	12.06	+2	-2
F	12.08	+3	-2
G	12.10	+4	-2

Table 3.7: Protection flight F - example 6

FLIGHTS	FPFS TIME	DELAY	DIFFERENCE FPFS
A	12.00	0	0
B	12.12	+11	+10
C	12.02	0	-2
D	12.06	+3	0
E	12.08	+4	0
F	12.04	0	-6
G	12.10	+4	-2

3.2.2 Optimization model of the UDPP - MUSH

The mathematical model of UDPP Multiple Users Single Hotspot (MUSH) does not differ too much with respect the SUSH model (Section 3.1.2). In order to focus in the relevant differences, only the variations of MUSH respect SUSH model will be presented (incremental approach). The major difference is that the user can't handle the scheduling of all the flights in the sector but he can only manage its own flights.
The following section will describe only the new variables, constraints and objective function present in MUSH model, whereas the explanation of the common elements between SUSH and MUSH can be found in Section 3.1.2.

3.2.2.1 Initial data

For MUSH model a new binary vector is required, called $MyFlight$ that represents which is the set of flights that the user can manage in the sector.

$$MyFlight = \begin{cases} 1 & Considered\,flights \\ 0 & Flights\,of\,other\,users \end{cases} \tag{3.24}$$

The vector of CMD will be 0 for all flights not considered because the user does not care about the flights of other airlines. In the real world, airlines do not usually know the CMD of other companies.

This new vector is complementary to the initial data for SUSH, described in Section 3.1.2.1.

3.2.2.2 The objective function

The objective function of MUSH model is to minimize the costs for the specific airline, with no attention to the import of other users (assuming "selfish behaviour).

$$minimize \sum_{i\ in\ 1...n,\ j\ in\ 1...m} MyFlights(i)\ \cdot\ [\,x(i,j)\ \cdot\ d(i,j)\ \cdot\ f(d(i,j))\ \cdot\ CMD(i)\,] \tag{3.25}$$

f(d(i,j)), as in the SUSH model, is a function related to the delay of the flights than the slots (see Section 5.1.2).

3.2.2.3 The constraints

There is only one different constraint from the previous model. This new boundary condition puts in a "normal" status the flights that do not belong to the selected user.

Constraint c8: "The flights of other airlines can not be managed".

$$\begin{array}{l} \forall \quad i \quad in \quad 1,...,n \\ if \quad (MyFlight(i) = 0) \quad \Rightarrow \quad n(i) = 1 \end{array} \tag{3.26}$$

The new bound is added to the constrains c1, ..., c7 described in Section 3.1.2.4.

3.2.3 Illustrative instance of the UDPP - Multiple Users Single Hotspot solved with Mosel

3.2.3.1 Initial data

The case studied has 7 flights belonging to 2 different airlines. In Table 3.8 it can be seen the different airline flights (airline 1 (YA) flights are in yellow and airline 2 (BA) flights are in blue) together with their arrival time.

Table 3.8: Initial arrival time - instance MUSH

FLIGHTS	Airline	TIME
A	YA	00
B	BA	01
C	YA	02
D	YA	03
E	BA	04
F	BA	05
G	YA	06

In the Table 3.9 and 3.10 there is the different cost per minute of delay for each flight and for each user.

Table 3.9: Flights of the yellow airline (YA) - instance MUSH

FLIGHTS	CMD (€)
A	1
C	14
D	1
G	15

Table 3.10: Flights of the blue airline (BA) - instance MUSH

FLIGHTS	CMD (€)
B	4
E	9
F	12

There is a reduction of 50% of the sector capacity and the new scheduling, following the First Planned First Served rule, is represented in Table 3.11.

Table 3.11: FPFS scheduling - instance MUSH

FLIGHTS	Airline	FPFS TIME	INCREASE THE DELAY
A	YA	00	0
B	BA	02	+1
C	YA	04	+2
D	YA	06	+3
E	BA	08	+4
F	BA	10	+5
G	YA	12	+6

3.2.3.2 Results

Blue Airline (BA) According with the Table 3.12, the Blue Airline, with FPFS scheduling, has an initial cost of 100€. After the application of SFP mechanism it can decrease the cost of delays to 80€, by suspending flight B and protecting E. In the same table it can be seen also that the Yellow Airline (YA) could also decrease its costs significantly (from 121€ to 63€) without participating actively in the UDPP procedure.

Table 3.12: SFP Blue Airline (BA) - instance MUSH

	Blue Airline (BA)	Yellow Airline (YA)
FPFS Cost	100€	121€
SFP Cost	80€	63€
Difference in costs between SFP and FPFS	-20€	-58€
Flights Suspended	B (BA)	-
Flights Protected	E (BA)	-
Final Scheduling	A (YA) C (YA) E (BA) D (YA) F (BA) G (YA) B (BA)	

Yellow Airline (YA) Table 3.13 presents the situation initially observed by YA before applying the prioritisation decisions made by BA. The Yellow Airline, with FPFS scheduling, has an initial cost of 121€. After the application of the SFP mechanism, it finds that can decrease the cost of delays to 13€ by suspending flight A and protecting flight G. In the Table 3.13 it can be also observed that the Blue Airline's costs would remain substantially unchanged (from

100€ to 96€) if only YA prioritisation request were applied. Note, however, that both airlines could request their preferred sequence in parallel, and thus the combination of flight prioritisation request must be analyzed (see Section 3.3).

Table 3.13: SFP Yellow Airline (YA) - instance MUSH

	Yellow Airline (YA)	Blue Airline (BA)
FPFS Cost	121€	100€
SFP Cost	13€	96€
Difference in costs between SFP and FPFS	-108€	-4€
Flights Suspended	A (YA)	-
Flights Protected	G (YA)	-
Final Scheduling	B (BA) C (YA) D (YA) G (YA) E (BA) F (BA) A (YA)	

3.2.4 Consequence of Optimal Solution UDPP - MUSH

It is interesting to study the difference results between the centralized mechanism SUSH (global decision making) and the de-centralized SFP (parallel decision making) in the same instances with respect to global and local solution of the problem. Table 3.14 shows how the results of the previous example changes after the application of the SUSH mechanism. The total cost decreases significantly (from 221€ to 55€).

Table 3.15 shows the distribution of costs between the two companies for the SUSH sequence.

It can be seen that there is a significant improvement for the Blue Airline (BA) (from 80€ to 19€ according to Table 3.12) and a notable (almost 3 times) worsening for the yellow company (according to Table 3.13, the local optimal solution for the Yellow Company is 13€ while in the case SUSH sequence is 36€).

The optimal choices of the SFP-SUSH mechanism are quite different than to the SFP-MUSH ones. For example, the flight E, protected in the (BA) MUSH instance, has no final delay due the suspension of the flights A and D, thus is not necessarily to protect it.

In the Table 3.16 there is the difference between the optimal solution SUSH and MUSH for the entire sector and how the cost is distributed among AUs in each case. The global solution of the yellow case (YA - MUSH) is two times worse than SUSH; that one of the blue case (BA - MUSH) is almost three times

FPFS Cost	221€
SFP SUSH Cost	55€
Difference in costs between SUSH and FPFS	-166€
Flights Suspended	A (YA) D (YA)
Flights Protected	G (YA)
Final Scheduling	B (BA) C (YA) E (BA) G (YA) F (BA) D (YA) A (YA)

Table 3.14: SUSH scheduling in the previous instance MUSH

User	Delay Costs
Blue Airline (BA)	19
Yellow Airline (YA)	36
SUSH Total Cost	55

Table 3.15: Distribution of the delay costs between users in SUSH

	Yellow Airline	Blue Airline	Σ
FPFS	121€ ↓	100€ ↓	221€ ↓
SUSH	36€	19€ ↑	55€ ↑
MUSH Yellow Airline	13€ ↑	96€	109€
MUSH Blue Airline	63€	80€	143€

Table 3.16: Difference between FPFS/SUSH/MUSH costs - instance MUSH

worse. In the Table the green cells (also highlighted by the symbol ↑) represent the best results while those in red (↓) the worst ones.

Figure 3.20 shows how the costs are distributed between the AUS in the three different scheduling mechanisms considered (FPFS, SFP-SUSH and SFP-MUSH) in this particular instance. This histograms points out that the solution with lower total costs (SUSH) is not necessarily the preferred by everyone (the Yellow Airline delay costs are almost three time worse than the YA - MUSH). Perhaps SFP is only useful in the presence of "dominant/Pareto-efficient" solution (i.e., everyone is better-off). In addition, the user, through the SFP-MUSH procedure, normally gets a better reduction of the delay costs than the other AUs.

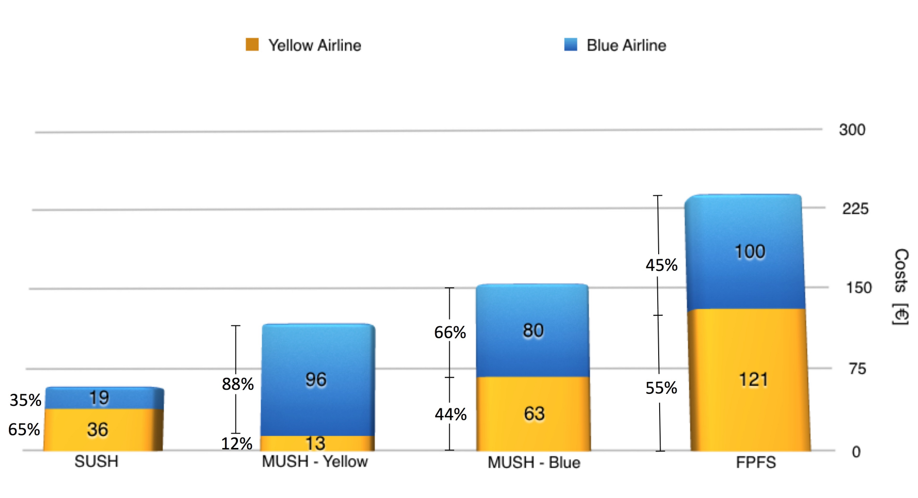


Figure 3.20: Chart of different costs between FPFS/SUSH/MUSH - instance MUSH

3.2.4.1 Boundaries of the MUSH Solution

In this sub-section it is explained how the limits of the found MUSH solutions are located between the FPFS and SUSH solution in the global and local cases.

With the help of the simulations, it can be highlighted very important results that describe the upper and lower bounds of the solution space.

Lemma 3.2.1 *The cost of the delay in a hotspot in which the flights are allocated with the FPFS technique is the upper bound of the global delay cost solution of the UDPP-SFP mechanism.*

Lemma 3.2.2 *The cost of the delay in a hotspot in which the flights are allocated with the UDPP-SUSH (one iteration) method is the lower bound of the global delay cost solution of the UDPP-SFP (one iteration) mechanism.*

Lemma 3.2.2 only applies if the condition of the single iteration is true. In fact, as will be seen (see Section 3.3.3 and 6.1.1), SFP mechanism (SUSH and MUSH) can be repeated several times in a sector in order to obtain a further reduction in costs.

Lemma 3.2.3 *The cost of the delay in a hotspot in which flights are allocated with the FPFS mechanism can be the upper bound of the global delay cost solution in the case that FPFS sequence is the global optimal solution.*

As it can be seen in Fig. 3.21, from the point of view of the global solution, the MUSH procedure is placed in the middle between SUSH and FPFS.

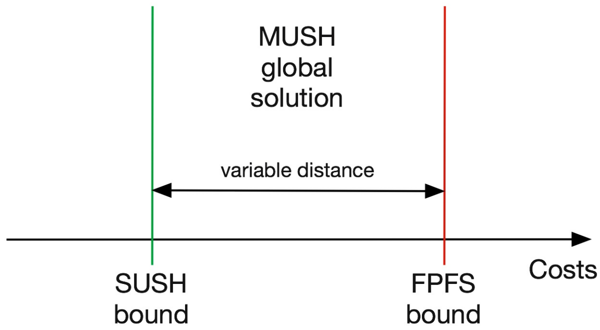


Figure 3.21: Bounds of MUSH global solution

Note that the global cost aren't the best solution for all the AUs. In fact, it is interesting to understand how these costs are distributed among the various users and how the AUs preferences are managed.

Lemma 3.2.4 *The cost of the delay in a hotspot in which the flights are allocated with the FPFS technique is the upper bound of the local delay cost solution of the UDPP-SFP mechanism for every AUs.*

Lemma 3.2.5 *The cost of the delay in a hotspot in which the flights are allocated with the FPFS mechanism can be the lower and upper bounds of the local delay cost solution, i.e. it is the local optimal solution.*

The local counterpart of the Lemma 3.2.2 is not valid. In fact it can happen that, in the local case, the preferences made by an individual user can give locally a larger cost reduction than the SUSH mechanism. In order to understand this concept, see Figure 3.22, which represents the results of the previous example. For the Yellow Airline (YA) the MUSH solution is almost three time better than the SUSH solution (its local optimal solution is 13€ while in the case global optimal one is 36€).

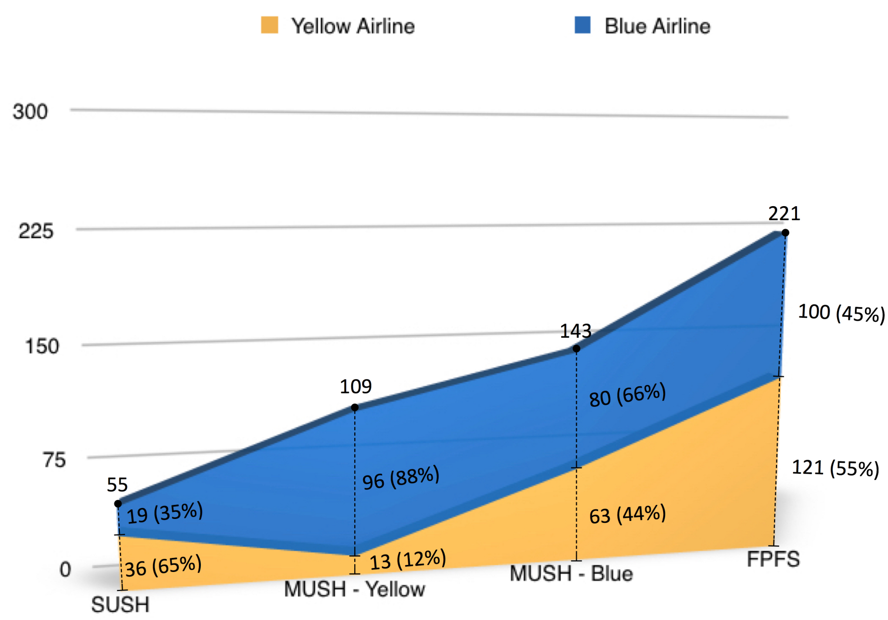


Figure 3.22: Bounds of MUSH global solution - instance MUSH

3.3 Interpolation of optimal results

Once studied the results in individual case MUSH, the Network Manager goal is to unify the various preferences of the airlines respecting the UDPP initial rules. It is important to know that the NM does not know the costs structure of the flights, so it will try to unify in a heuristical manner the various prioritisation requests without having information on actual costs. In this section it is explored which can be the effect of these heuristics on the distribution of costs though the following mechanism:

- FPFS solution;
- SUSH solution;
- MUSH solutions;
- Union of the two MUSH solutions;
- Conditional solutions (As an user performs knowing the preferences of other AUs);
- Iterative conditional solutions.

The goal is to find how far the different solutions obtained are from the optimal solution (SUSH) and from the worst one (FPFS) as implicit in the Figure 3.21. To evaluate these different cases it will be used the previous MUSH example.

3.3.1 Union of the solutions

Now let us consider that the Network Manager can unify the MUSH solutions requested by the two airlines.
In our previous example the following flight prioritisations were requested:

Suspend:	A & B
Protect:	E & G

Note that if flight A and B were both suspended, the first slot would become a hole. This is a simplified example of how two independent solutions (from both airlines) can be incompatible. In such a case, the NM will need to decide which of the two flights suspend and keep the other. The Network Manager doesn't know the different flight costs so it can not choose in an accurate way which scheduling has been chosen.
It will be evaluated both previous solutions:

1. Suspend B in Figure 3.23;
2. Suspend A in Figure 3.24;

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SUM
Arrival Time	A	B	C	D	E	F	G								
Land Time	A		C		E		D		G		F		B		
Delay	0		0		0		+3		+2		+5		+11		
Cost	0		0		0		3		30		60		44		137

Figure 3.23: Case 1 suspension flight B - Union MUSH

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SUM
Arrival Time	A	B	C	D	E	F	G								
Land Time	B		C		E		D		G		F		A		
Delay	0		0		0		+3		+2		+5		+12		
Cost	0		0		0		3		30		60		12		105

Figure 3.24: Case 2 suspension flight A - Union MUSH

In the Table 3.17 it can be seen that in the Case 1 (the suspension of flight B), the Blue Airline has a worsening in the costs respect FPFS scheduling (from 100€ to 104€). From the local optimal, the union solutions can lead to a deterioration from the initial condition. It may happen, as in this example, that the airline might be suspending a flight without achieving the desired benefits.

It is very important to emphasize in conclusion that there is a close interdependence between the local decisions of the various AUs, as described in the shown example. These dependencies can lead to the creation of holes, hardly optimally managed by the Network Manager.

	Case 1	Case 2
AY Cost	33€	45€
AB Cost	104€	60€
Total Cost	137€	105€

Table 3.17: Costs of the two cases - Union MUSH

3.3.2 Conditional solutions

The second way to process the various solutions is to influence combinatorially the choices made by the operators: what would happen when a user decides after another. Each AU waits for the scheduling proposed by competitors (MUSH solutions) and then decides the strategy to be carried out.

Continuing with the considerations of the previous example, two different cases can be considered:

- A1 performs knowing the preferences of A2. As it can be seen in Figure 3.25, the Yellow Airline decides to suspend flight B and to protect flight G thus lowering its costs. In fact, as shown in the Fig. 3.26, the local costs for YA are reduced from 63€ to 9€ (-86%) and also BA achieved a reduction in costs (from 80€ to 72€);
- A2 performs knowing the preferences of A1. In our example the Blue Airline does not make any choice and does not change the scheduling proposed by A1.

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SUM
Arrival Time	A	B	C	D	E	F	G								
Land Time	A		C		E		G		F		B		D		
Delay	0		0		0		0		+3		+9		+9		
Cost	0		0		0		0		36		36		9		81

Figure 3.25: Decisions of A1 knowing A2 strategy

	MUSH(A1)	MUSH(A2 I MUSH(A1)	MUSH(A2)	MUSH(A1 I MUSH(A2)
Cost A1 Yellow Airline	13	13	63	9
Cost A2 Blue Airline	96	96	80	72
Total Cost	109	109	143	81

Figure 3.26: Different costs between MUSH and conditional MUSH

A negative aspect of conditional solution is the high number of the case combinations, which increase linked to the amount of users, in particular is $n \cdot (n-1)$, where n is the number of airlines considered. The order of complexity[2] can be approximated to $O(n^2)$.

[2]In appendix A it is explained the concept of orders of complexity.

3.3.3 Iterative conditional solutions

The reasoning in the previous section can be extended. It can be figured out what every user would do in response to each different scheduling made previously by competitors in several steps (iteratively).
Continuing with the previous example, it would be considered what BA would decide after YA: a case referred as $MUSH(A1|MUSH(A2))$.
In this manner it can be obtained $MUSH(A2|MUSH(A1|MUSH(A2)))$. In the previous case this does not make improvements, but especially with multiple users, this procedure could lead to big enhancements of the final solution.
It can be thought a global market where alternately every airline says its scheduling. Due to the rule of fairness, the application of prioritisations of other companies can only improve the results at every step: the delay of the planes that do not actively participate in UDPP in a particular round can not increase, it can only stay the same or decrease. The various companies can only get delay reductions at every step of the algorithm.
It can be observed that the combinatorial entities of the system implies an explosive combinatorial that result in an intractable complexity for the mathematical model. For instance, let us show what would happen if three airlines decide iteratively conditionally. The computational explosion is shown in Figure 3.27.

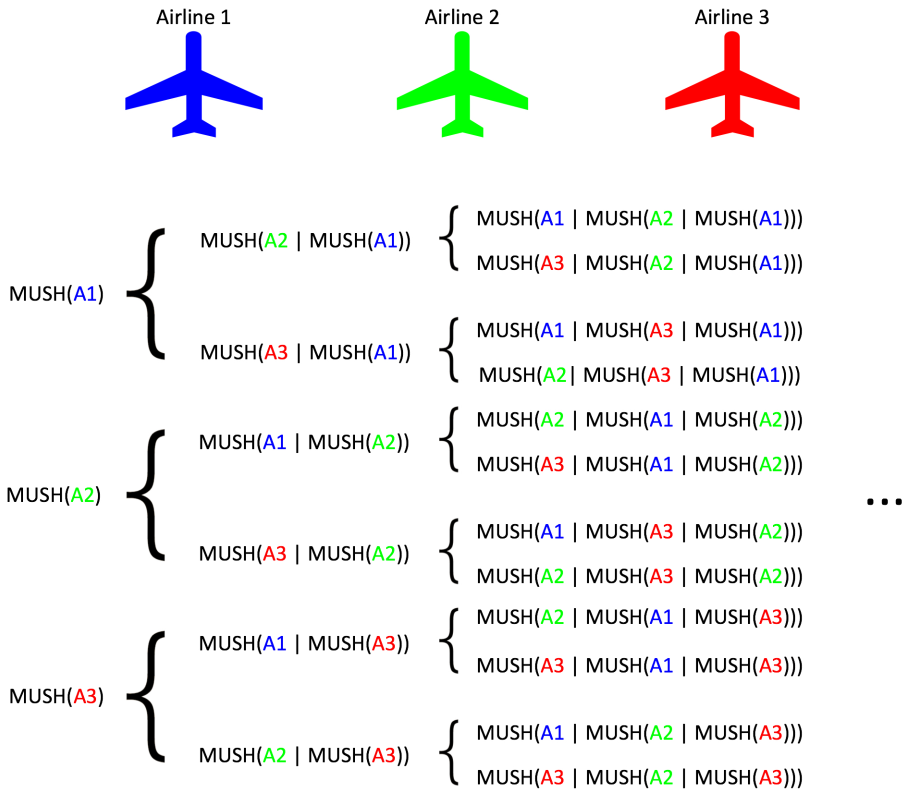


Figure 3.27: Complexity of iterative conditional solution - three airlines

The number of operations depends on the interpolation level that it want to be obtained. In the Figure 3.27 the level of iterative conditional progression stops at the second step, but it can be easily seen how it is possible, indeed it is almost a must, get more iterations between the various users for a better solution.
Considering n the number of airlines involved and h is the number of the steps, the result is n times n-ary complete tree and the total number of nodes (which are the MUSH operations) is:

$$number \quad of \quad nodes \quad = \quad n \cdot \left[(n-1)^{h+1} - 1 \right] \tag{3.27}$$

The number of calculations can then be approximated to $O(n^h)$. This type of complexity makes the problem intractable for combinatorial exploration.

3.3.4 Final thoughts on UDPP - MUSH

Due to the complexities present in the context of MUSH, an approach to the problem based on Monte Carlo simulations is proposed. Figures 3.28 and 3.29 show a summary of results of different strategies tested in the previous examples. The aim of introducing Monte Carlo simulations is to find the best strategy to solve the MUSH problem, based on statistical description of the different strategies simulated thousand of times:

- FPFS is the worst global solution;
- SUSH is the optimal global solution (1 iteration);
- $SUSH_{global\ (1\ step)}(solution) \leq MUSH_{global\ (1\ step)}(solution)$;
- $MUSH_{global}(solution) \leq FPFS_{global}(solution)$
- FPFS can not be the worst local solution;
- SUSH can not be the optimal local solution;
- iterative conditional solution can be a pseudo-optimal solution;
- the complexity of iterative conditional solution is exponential;
- the UNION of different MUSH solution is a pseudo-optimal solution;
- it can happen that there is no solution to the UNION (the suspension of flights can create an hole in the scheduling) and how often it can happen will be quantified.

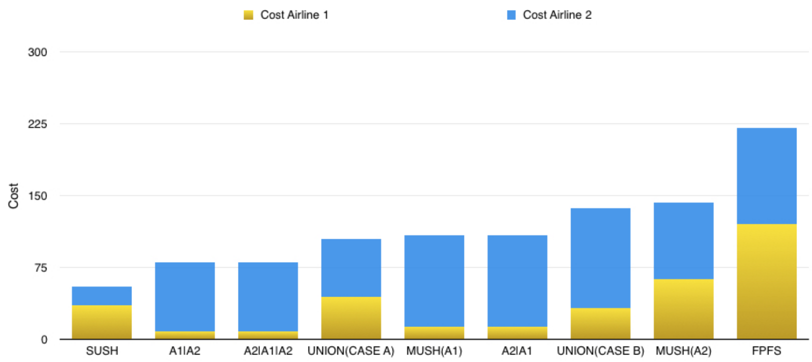


Figure 3.28: Comparison of results in the previous example - MUSH

Mechanisms	FPFS	SUSH	MUSH(A1)	MUSH(A2)	UNION$_{case\ 1}$	UNION$_{case\ 2}$	MUSH(A2 | MUSH(A1))	MUSH(A1 | MUSH(A2))	MUSH(A1 | MUSH(A2 | MUSH(A1)))	MUSH(A2 | MUSH(A1 | MUSH(A2)))
Scheduling	A (YA) B (BA) C (YA) D (YA) E (BA) F (BA) G (YA)	B (BA) C (YA) E (BA) G (YA) F (BA) D (YA) A (YA)	B (BA) C (YA) D (YA) G (YA) E (BA) F (BA) A (YA)	A (YA) C (YA) E (BA) D (YA) F (BA) G (YA) B (BA)	A (YA) C (YA) E (BA) D (YA) G (YA) F (BA) B (BA)	B (BA) C (YA) E (BA) D (YA) G (YA) F (BA) A (YA)	B (BA) C (YA) D (YA) G (YA) E (BA) F (BA) A (YA)	A (YA) C (YA) E (BA) G (YA) F (BA) B (BA) D (YA)	B (BA) C (YA) D (YA) G (YA) E (BA) F (BA) A (YA)	A (YA) C (YA) E (BA) G (YA) F (BA) B (BA) D (YA)
Yellow Airline (YA) Costs	121 € ↓	36 €	13 €	63 €	33 €	45 €	13 €	9 € ↑	13 €	9 € ↑
Blue Airline (BA) Costs	100 €	19 € ↑	96 €	80 €	104 € ↓	60 €	96 €	72 €	96 €	72 €
Total Costs	221 € ↓	55 € ↑	109 €	143 €	137 €	105 €	109 €	81 €	109 €	81 €

Figure 3.29: Summary of the MUSH instance considered

Chapter 4

Simulator

During the last phases of this research, a simulator in JAVA was designed and implemented to evaluate the different impacts of the various algorithms introduced in Chapter 3.
To program this simulator, the software development platform NetBeans 8.1 has been used[10].

4.1 Concept of the Simulator

The simulator uses the optimization model written in FICO Mosel Xpress to find the optimal solutions of the instances created and the data is exported in Microsoft Excel for analysis in an easier and faster way.
Java was considered as the programming language, since it is independent of the platform and also contains many useful tools and libraries for the purpose of research.
The core idea of the simulator is to create a connection bridge between JAVA, FICO Mosel and the final output format: Excel spreadsheet.

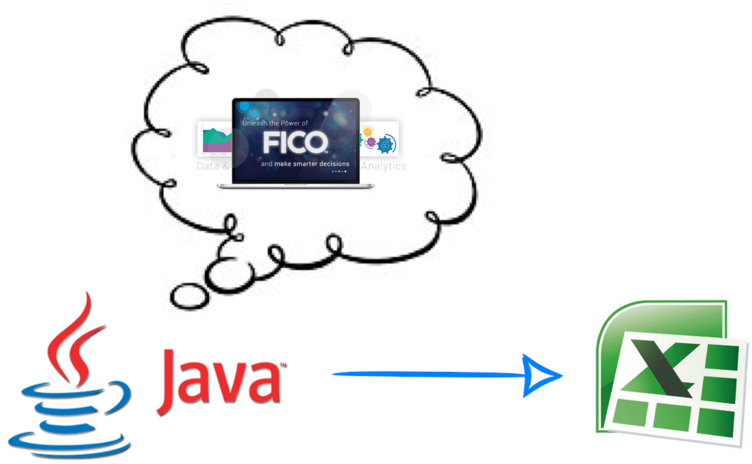


Figure 4.1: Concept of the simulator

In order to make the connections between the different layers two different libraries were used two different libraries:

- Apache POI - the Java API for Microsoft Documents [16], made by The APACHE Software Foundation;
- XPRD, XPRS, XPRB, XPRM - the JAVA API for FICO Xpress Mosel.

In this software, there are three different layers:

1. the top level: FICO Xpress;
2. the middle level: JAVA;
3. the output level: Apache POI - Excel.

4.2 Design of the Simulator

The simulator recreates hotspots of 15 aircraft that belong to two different airlines. The cost of the first minute of delay[1] of each flight and of which airline is each plane are chosen randomly.

After an initial phase of setting (as to set the number of simulations), the simulator starts to perform by communicating with the various layers as shown in Figure 4.2.

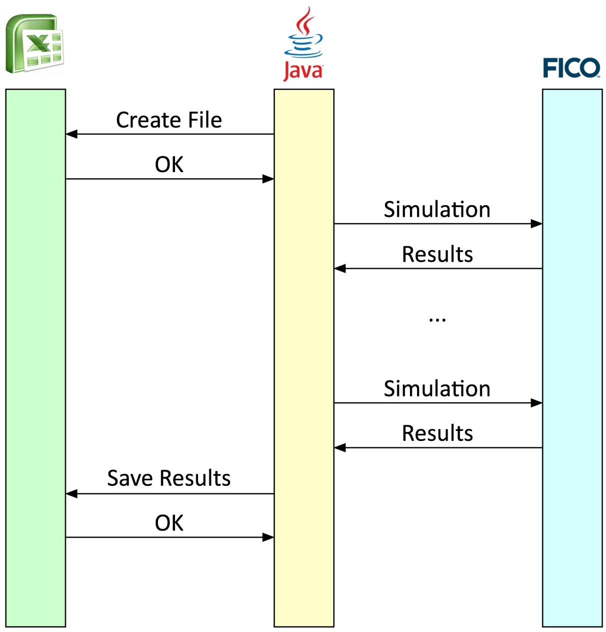


Figure 4.2: Connection between layers in the simulator

[1]The CMD is chosen in a range of possibile values due to the studies of cost-delay function, see Section 5.1.2.

Each simulation consists of several steps that are the various algorithms studied in the Chapter 3 and it can be schematically represented by the Figure 4.3.

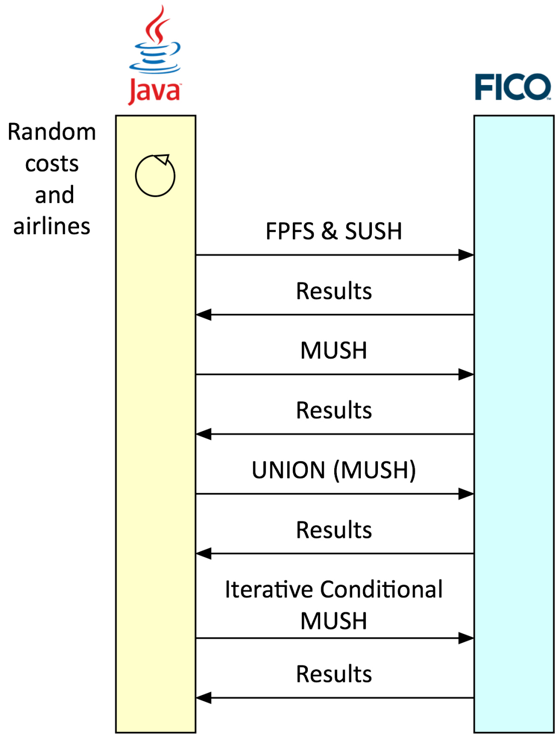


Figure 4.3: Single simulation graph

The UNION procedure is more complex than the others. In fact it may cause some complexities given 2 main situations:

1. If one of the two airlines decided not to suspend flights, the final scheduling will be the other airline scheduling without modification;
2. The union of the suspension of airlines can generate holes in scheduling. See the following example.

Example Consider a situation in which 6 flights are involved.
The first airline owns $F_1 = \{A_1, B_1, C_1\}$.
Instead the second airline owns $F_2 = \{A_2, B_2, C_2\}$.
The FPFS scheduling is shown in the Figure 4.4.

After the MUSH algorithm the first airline decides to suspend A_1 and the second airline suspends A_2. The union of these two MUSH solutions generate a hole in the scheduling, as shown in Figure 4.4.

This situation is quite rare (less than 1% of the time, as shown in Ch. 5), but this cases require special treatment by the Network Manager.

To solve this issue, a new object was introduced into the simulator model: the Network Manager, who handles this kind of particular issues. In the UML

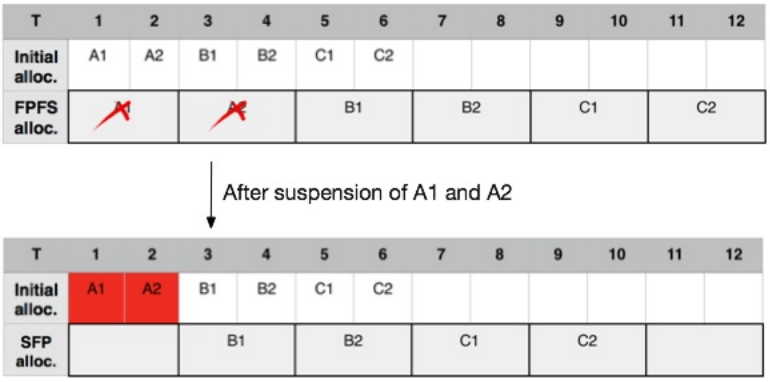


Figure 4.4: Hole - UNION Hole example

activity diagram, in Fig. 4.5, is shown how the NM interacts with the layer of the simulator and the optimization suite MOSEL.

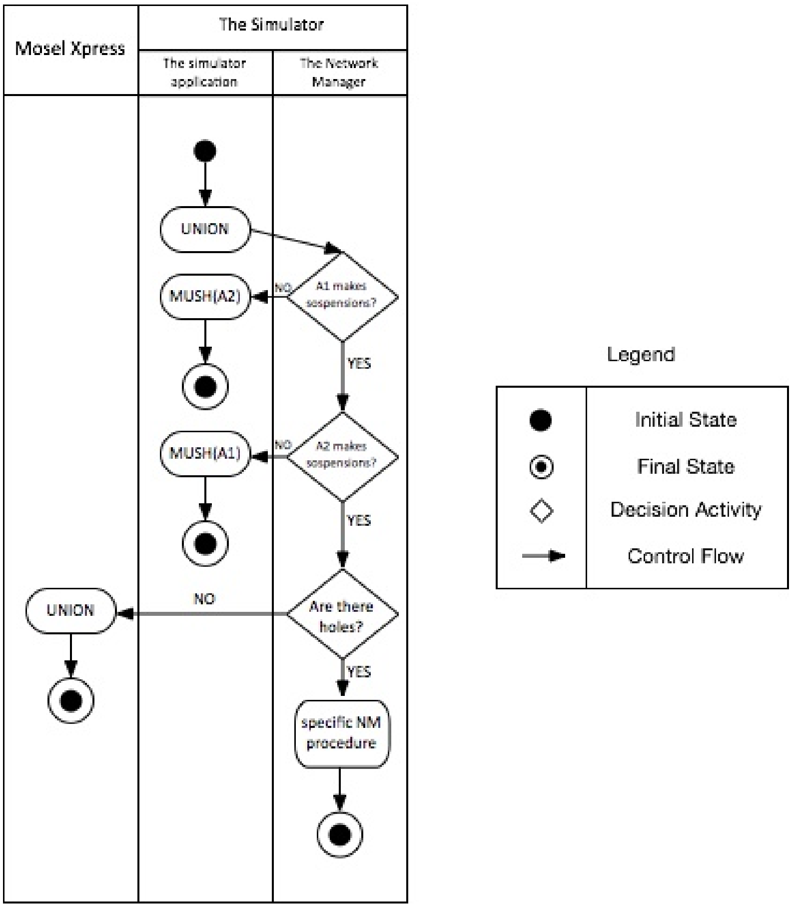


Figure 4.5: UML - Activity Diagram - Union Function

In the presence of a hole in the sequence, the NM can choose between different procedures to find a final scheduling. In this research, FPFS and MUSH mechanisms were been applied for those cases.

4.3 Final output of the Simulator

The final output of the software is a file in Excel format which will represent all the values of the simulations so as to carry out statistical studies on the different algorithms. The file will be made up of 27 columns: some of them representing the various algorithms. An example of content is in Fig. 4.6[2].

Simulazion	FPFS - Total Cost	FPFS - A1 Cost	FPFS - A2 Cost	SUSH - Total Cost	SUSH - A1 Cost	SUSH - A2 Cost	MUSH(A1) - Total Cost	MUSH(A1) - A
1	1934,96	789,01	1145,95	1065,12	349,97	715,16	1492,50	510,50
2	3265,47	1894,79	1370,68	1230,39	868,40	361,99	1757,29	836,85
3	1594,08	782,74	811,34	1086,00	196,71	889,29	1351,20	557,55
4	2482,51	876,54	1605,97	936,94	459,84	477,10	1812,66	542,35
5	2748,34	2148,72	599,62	1015,10	885,46	129,64	1031,23	953,19
6	2699,76	1141,73	1558,03	1111,31	667,90	443,41	1731,92	550,72
7	2522,79	715,94	1806,86	1410,04	863,06	546,99	2150,73	471,08
8	2381,88	1772,69	609,19	1763,65	1637,76	125,90	1936,95	1327,76
9	3433,40	1924,66	1508,74	2276,40	908,55	1367,84	3277,76	1769,02

Figure 4.6: Excel file example

From thousands of simulations different tables have been filled. The methods of analysis of the simulations and the results obtained will be described in the Chapter 5.

[2]The colors were added later to the Excel file to get better visibility of the different content.

Chapter 5

Simulations & Results

The purpose of this Chapter is to describe the characterization of the Monte-Carlo simulations and the results obtained through the agent based simulator (see Chapter 4) in order to highlight which are the pros and cons of the different mechanisms considered to manage the delay costs in the event of a hotspot.

This Chapter is divided into three main sections:

- the simulations setting (assumptions, parameterizations and particularities in models used);
- the methods to analyzing the results;
- the final results.

5.1 Simulations settings

The agent based simulator sets up a hotspot situation in which 15 flights of two different airlines are involved. The sector capacity is reduced to the half of its nominal capacity: the demand is the double than the capacity and then the OI index considered is equal to 200. This index value implies that for each suspension, the user can only protect a maximum of two flights. The slot size is of 2 minutes for a total duration of the hotspot of half an hour.

5.1.1 General assumptions considered during the simulations

In the contest of the SFP optimization model and the simulator implementation, the following assumptions are made:

- the ultimate goal of SFP is to minimize costs instead of maximize profits or Utility. However any of these problems are closely related and their representation in the mathematical model can be easily adopted. The conclusions reached could be different than the ones provided in this research, however.

- no negative delay is allowed ($d \geq 0$). It is assumed that the flights cannot be ready before the time in which the flight was originally planned to use the slot. In real operations there might be some tolerance. This is a parameter that can be changed in the model;
- a flight whose initial scheduling time is in the middle of a slot due to a capacity reduction, it can be associated to that slot with a delay of zero minutes (no negative delays).
- the granularity of time is in the order of minutes.
- the scheduling of the flights is accurate to the minute;
- the cost of the delay for each plane is calculate precisely (see Section 2.2.1). In the real world flight operator might have no perfect information about the actual costs of a flight.
- slots have a constant time size accurate to the minute;
- the demand and capacity are considered to be balanced before the CCS;
- the potential interdependencies between hotspots are not considered;
- each flight has on average 150 passengers;
- the flight delays do not exceed 240 minutes.

5.1.2 Objective Function parametrization

In Section 2.2.2 it has been introduced the concept of different cost per minute of delay. Cook, Tanner, Jovanovic and Lawes studied about the cost of delay to air transport in Europe ([19]) and found a trend of the function similar to that pointed out in Fig 2.6 (Sec. 2.2.2).
Assuming that each flight has on average 150 passengers, through MATHEMATICA Software[1] and the data in the Fig. 2.5 (Sec. 2.2.2), it has been found a mathematical function that has been used to characterized the typical delay cost structure of flights. The curve that represents (on average) the cost of the minute delay (not considering delays larger than 240 minutes) is:

$$y = -0.00691019x^2 + 2.85533x + 14.4519 \tag{5.1}$$

This function is a parabola and in Fig. 5.1 it can be seen his value between 0 and 240 minutes of delay.

Through the data in Table 2.5 and the assumption about the average number passengers per flight (150 passengers per flight), it can be argued that the cost of the first minute of delay (CMD) per flight in a normal cost scenario is 19.5€.

The equation 5.1 can therefore be seen as follows:

[1] WOLFRAM MATHEMATICA, http://www.wolfram.com/mathematica

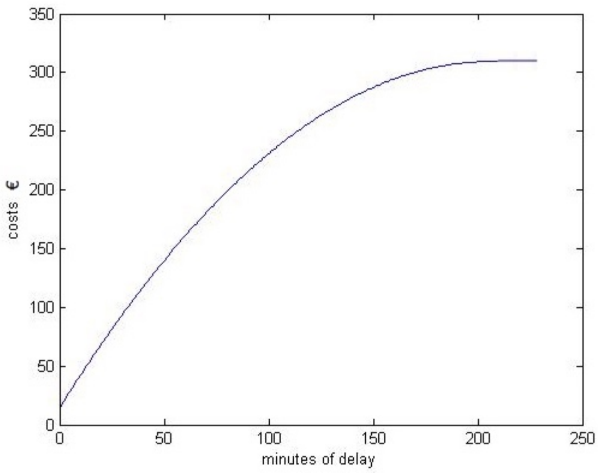


Figure 5.1: Parabola of costs of minute of delay

$$\begin{aligned} y \; &= \; -0.00691019\, x^2 + 2.855331\, x + 14.4519 \\ &= 19.5 \cdot (-0.000354359\, x^2 + 0.146427\, x + 0.741123) \\ &= CMD \cdot (-0.000354359\, x^2 + 0.146427\, x + 0.741123) \end{aligned} \tag{5.2}$$

This function can be split in two parts:

1. the cost for the first minute of delay for each flight (variable part of the cost function);
2. a constant part of the cost function related to the delay.

The cost of each minute of flight delay is as follow:

$$\begin{aligned} y \; &= \; CMD \cdot (-0.000354359\, x^2 + 0.146427\, x + 0.741123) \\ &= CMD \cdot f(delay) \end{aligned} \tag{5.3}$$

Eventually the total cost of the flight delay can be represented as follow:

$$Cost\,of\,flight\,delay = delay\; \cdot\, CMD \cdot f(delay) \tag{5.4}$$

The goal of each simulation is to minimize the following (see Fig. 5.2) Objective Function[2] (where i are the flights, j are the slots).

[2]see the Objective Function of SUSH and MUSH models respectively in Sections 3.1.2.3 and 3.2.2.2

$$\sum_{ij} \underbrace{x(i,j)}_{\text{decisional variable}} \cdot \{\underbrace{delay(i)}_{\text{delay of the flight}} \cdot \underbrace{[CMD(i)\,(-0.000354359\, delay(i)^2 + 0.146427\, delay(i) + 0.741123\,)\,]}_{\text{different cost of the minute of delay}}\}$$

Figure 5.2: Models objective function

5.1.3 Some model implementations issues

As introduced in Section 3.1.2.8, the implementation of the constraint of the protection (see Section 3.1.2.4 - Constraint c7) has led to the development of two different models: $SFP_{UpperBound}$ and $SFP_{LowerBound}$.

The only difference in these models is the implementation of c7, which was made as follows, respectively:

$SFP_{UpperBound}$: *"the user is free to allocate the protected flights in any slot between that one obtained in flight initial allocation (before the reduction of the sector capacity) and that one resulting by the sequence that follow the FPFS policy, choosing the best slot for reducing its delay costs"*.
This constraint does not ensure that the protected flights are placed in their first position possible (flights are allocated in the best slot for minimizing the delay costs which is not necessarily their first possible slot). This is a soft constraint and the final solution will have slightly lower or equal delay cost than the SFP optimal solution;

$SFP_{LowerBound}$: *"the flights protected will be placed only in their initial allocation slot (before the reduction of the sector capacity) if this is possible, otherwise they will not be protected"*. This solution will have the same or slightly larger cost than the optimal solution.

All simulations were handled with these two models simultaneously in parallel in order to evaluate the various differences. The final optimal solution will be therefore in the range [$SFP_{UpperBound}$, $SFP_{LowerBound}$].

5.2 Methods for result analysis

The main method used to analyze the simulation results is the linearization of the solution space within the gap between SUSH and FPFS solutions (i.e., the best and worst case), as shown in Figure 5.3.

The purpose is to find "how far" are in percentage the solutions of different MUSH strategies with respect the best and worst case (see Section 5.4 where the discussion about the need and interest of such analysis can be found), with respect the total delay cost in a sector (total delay cost), but also for each AU

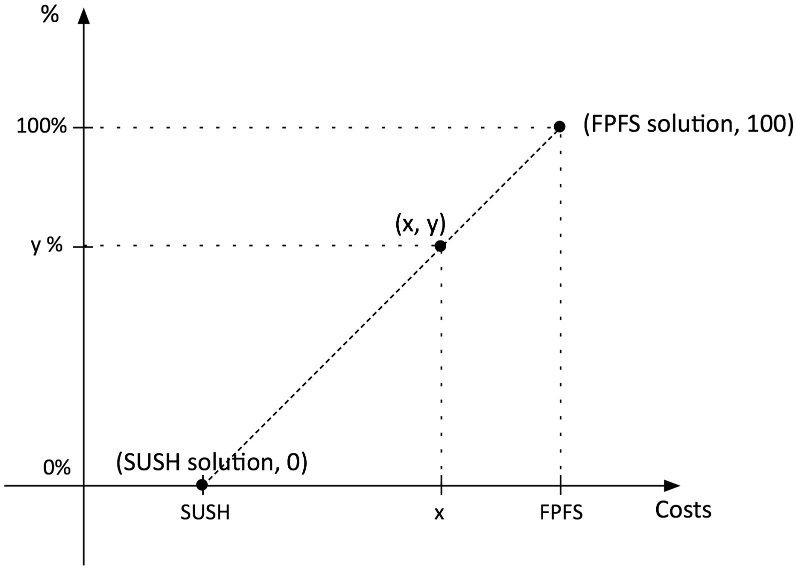


Figure 5.3: Linearization of the solution space between SUSH and FPFS

involved individually (local delay cost).

To find the value of the percentage of the function it is used the value of the line through the two extreme points: SUSH(x_1, y_1) and FPFS (x_2, y_2) solutions where $y_1 = 0\%$ and $y_2 = 100\%$ (as shown in Figure 5.3).

The linear equation in the two variables x and y in two-point form is:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) \tag{5.5}$$

where (x_1, y_1) and (x_2, y_1) are two points on the line with $x_2 \neq x_1$. The straight line is thus obtained:

$$y = 100 \cdot \frac{x - x_1}{x_2 - x_1} \tag{5.6}$$

Eventually the grade of a solutions (gap percentage) from different MUSH strategies can be expressed as follows:

$$GAP = 100 \cdot \frac{x - SUSH_{solution}}{FPFS_{solution} - SUSH_{solution}} \tag{5.7}$$

The gap value leads to the following conclusions on the quality of the strategy solution:

1. $0\% < GAP < 100\%$ normal solution;
2. $GAP = 0\%$ the solution is optimal (the same SUSH algorithm);
3. $GAP = 100\%$ the solution cannot improve the worst-case (i.e., the solution from the FPFS policy);

4. $GAP < 0\%$ the solution is better than the SUSH solution (i.e., in Section 3.2.3.2 it has been shown that it can happen occasionally that the SUSH local AU solution can be larger than the MUSH local one);

5. $GAP > 100\%$ the solution is worse that one made with the FPFS policy (i.e., in Section 3.3.1 it has been shown that it can happen occasionally that the local AU solution with the FPFS policy can be smaller than the local solution in the mechanism after unifying the MUSH results with the UNION method in the hole problem condition with a random suspension choice made by the Network Manager (see the example in Section 3.3.1)).

Steps 4 and 5 of the previous list are possible only in the local solution condition (prioritization actions taken by individual AUs), never globally.

Using the value of the gap as the only final solution index does not show the actual distance that incurs between SUSH and FPFS solutions. Since the data is totally random, it can happen situations where SUSH solution FPFS can coincide or be very close and this does not put in evidence significant results. Moreover, as reported in the explanation of the concept of GAP in steps 4 and 5, it can occur in local AU solutions conditions that lead to a critical particular gap values, as shown in the next example.

Example Consider a situation in which two airlines (A1 and A2) are involved in a hotspot. The A1 delay cost resulting from sequence with FPFS policy is 800 € while using the SUSH mechanism the A1 delay cost is 790 €. It is assumed that the final value of the MUSH solution is 400 € (the solution made by the MUSH mechanism is better than the SUSH one; it will be expected a negative GAP value).

$$\begin{aligned} GAP &= 100 \cdot \frac{MUSH_{local\ solution} - SUSH_{local\ solution}}{FPFS_{local\ solution} - SUSH_{local\ solution}} \\ &= 100 \cdot \frac{400 - 790}{800 - 790} = -3900\% \end{aligned}$$

From the statistical analysis of the various cases, the occasional presence of such large negative values leads to inaccuracies regarding the averages of the local AU final results.

To handle these issues, the percentage GAP is always associated with the objective value of the actual distance between the optimal solution and the worst-case.

5.3 Results

With the Monte-Carlo simulations it was possible to obtain statistical informations about the reduction of delay costs using as a baseline the costs of the flights sequence obtained with the FPFS policy.

The average of the total delay cost with the FPFS policy is 3923.45€ for a hotspot situation. Using the SFP-SUSH mechanism, the costs on average vary between 2998.77€ and 2966€ (using the two different models introduced in Section 5.1.3).

Table 5.1 shows the differences between the maximum and minimum costs in the sector using the SUSH and FPFS methods for allocating flights. Note how all of the costs that will be considered in all the mechanisms with respect to all AUs will be in the range of 1748.53€ and 5505.36 €.

	FPFS	$SUSH_{Lower\ Bound}$	$SUSH_{Upper\ Bound}$
min	1944.78€	1748.53€	1748.53€
media	3923.45€	2998.77€	2966.00€
max	5505.36€	5046.31€	5023.38€

Table 5.1: FPFS and SUSH methods - final results

On average, **SUSH mechanism brings the saving of the delay costs of about 24% compared to the FPFS policy**.

$$23.6\%_{\ Lower\ Bound} \geq SUSH_{Optimal} \geq 24.4\%_{\ Upper\ Bound} \tag{5.8}$$

The difference (on average) between the SUSH mechanism result in lower bound and upper bound case is 32.8€ (about 1% compared with the size of costs), it might be neglected given the magnitude of the costs.

Approximately in the 93% of the simulations, the SUSH mechanism reduces costs by at least 400€ (> 10% of total costs).

In Figure 5.4 it can be considered how the costs are in average distributed between the two users during the initial FPFS allocation policy and after the method SFP-SUSH: both AUs get a similar distribution (about 50%) of the costs. Due to this fact, it can be argued that both **users get about the same benefits using the SFP-SUSH mechanism**.

Let us consider the statistical results of the SFP-MUSH mechanism. Using the GAP function[3] introduced in the section 5.2, the MUSH (one iteration) method is between 73% and 65%. This fact implies that **the SFP-MUSH (one iteration) mechanism brings the saving of the delay costs of about 8% compared to the FPFS policy**.

Figure 5.5 shows the distribution of the costs between the two airlines with the MUSH mechanism (one iteration). **The AU that uses the SFP-MUSH (one iteration) mechanism, will get, on average, a slightly larger reduction (about +2%) of costs related to the delay than the other**.

As introduced in Section 3.3.2, one of the way to process the MUSH solutions is to iterate the MUSH mechanism in order to consider what results the AU would get if it makes prioritization decisions after the competitors. The GAP function obtained with two MUSH steps is 36.7%, while with three iterations is of 35.4%. Other steps in the MUSH mechanism does not bring improvements.

[3] 100% indicates the flight delay cost using FPFS policy and 0% the flight delay cost using SFP-SUSH mechanism

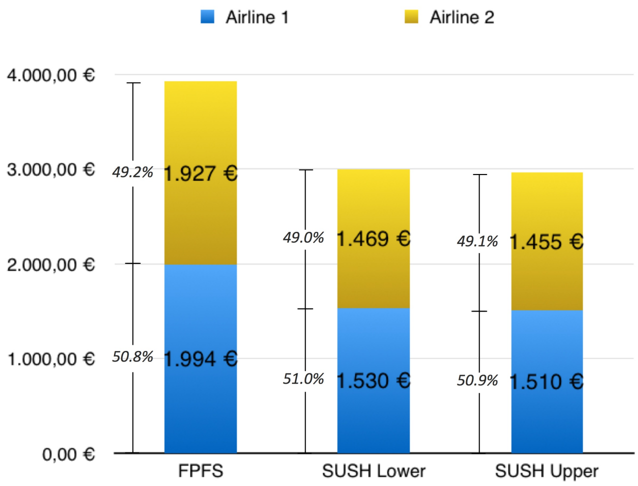


Figure 5.4: Distribution of the costs between the AUs with FPFS and SUSH mechanisms

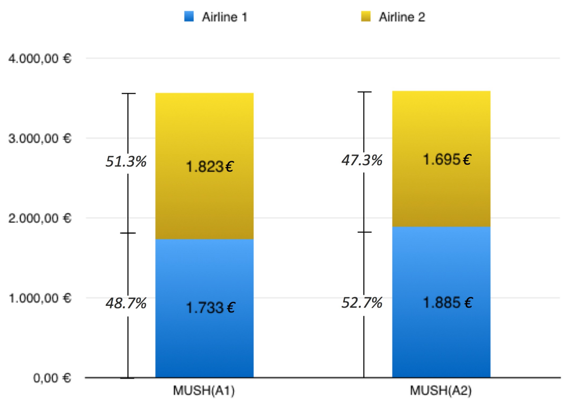


Figure 5.5: Costs distribution between the AUs in the case of SFP-MUSH (one iteration) mechanism

The iteration of the MUSH mechanism brings the saving of the delay costs of about 14% compared to the FPFS policy. For every iteration the grow rate of the improvement decreases and the running time for finding the solution increases exponentially.

Figure 5.6 shows the distribution of the costs between the two airlines with the MUSH mechanism (two iteration). The AUs get a similar distribution (about 50%) of the costs. Also the percentage remains almost the same by increasing the number of iterations. Due to this fact, it can be argued that both **AUs shall get about the same benefits using the iterative MUSH mechanism**. This conclusion supports the initial concept of Iterative MUSH strategy: the idea of a kind of market mechanism where every AU can handle its flights schedules in response to other users choices.

Figure 5.6: Costs distribution between the AUs in the case of SFP-MUSH (two iteration) mechanism

The second considered way to process the MUSH solutions is to unify the prioritisation requests of the two airlines (see Section 3.3.1).

In the merge process were used some combinations of models introduced in Section 5.1.3. More specifically it was studied how the protection constraint (c7) changed the results in the initial solution MUSH and the merging procedure. There were considered three different models:

1. $Union_U(MUSH_U)$: the Upper Bound rule to unify the $MUSH(1\ step)_{Upper\ Bound}$ protections;
2. $Union_U(MUSH_L)$: the Upper Bound rule to unify the $MUSH(1\ step)_{Lower\ Bound}$ protections;
3. $Union_L(MUSH_L)$: the Lower Bound rule to unify the $MUSH(1\ step)_{Lower\ Bound}$ protections;

The actual optimal solution of the union of the SFP-MUSH prioritisations is in the range of $Union_U\,(MUSH_L)$ and $Union_L\,(MUSH_L)$ procedures solutions.

The operation for unifying the MUSH prioritisations is affected by the potential formation of holes in the flights scheduling sequence (see Section 3.3.1). Furthermore the procedure $UNION_L$ might not admit solution in critical case due the particular structure of the flights protection constraint (see Section 5.1.3).

Table 5.2 shows the GAP function in the three different procedures (between 32% and 52.7%). **The problem of the holes formation present in the UNION mechanism of the MUSH preferences may be almost neglected as the percentage of occurrence is very small (< 0.1%).** It should be considered instead in $UNION_L$ the handling of the case "no solution" which is considerably large (17.4%).

	$Union_U(MUSH_U)$	$Union_U(MUSH_L)$	$Union_L(MUSH_L)$
GAP	32.07%	45.36%	52.69%
holes or no solution	0.1%	0.1%	17.4%

Table 5.2: Union of MUSH preferences - GAP, hole and solutions results

The Network Manager can choose which mechanism to use in the case of holes or not solutions:

1. FPFS policy. Table 5.3 shows the final GAP results using this strategy. The actual optimal solution of the union of the SFP-MUSH prioritisations will be between 45.41% and 60.92%. **The mechanism of unifying the MUSH prioritisations using FPFS policy to manage the critical situations brings the saving of the delay costs of about 9% - 13% compared to the FPFS policy**. Futhermore the AUs get a similar distribution (50%±1%) of the costs. **AUs get about the same benefits using the strategy of unify the MUSH prioritisations using FPFS policy to manage the critical situations**.

	$Union_U(MUSH_U)$ and FPFS	$Union_U(MUSH_L)$ and FPFS	$Union_L(MUSH_L)$ and FPFS
GAP	32.14%	45.41%	60.92%

Table 5.3: Union of MUSH preferences + FPFS policy for critical situations - GAP results

2. MUSH one iteration mechanism. Table 2 shows the final GAP using this strategy, which is between 45.39% and 56.22%. **The mechanism of unifying the MUSH prioritisations using MUSH (one iteration) mechanism to manage the critical situations brings the saving**

of the delay costs of about 10.5% - 13% compared to the FPFS policy. The users get a similar distribution of the delay costs (50% ± 1%). It can be concluded that both **AUs get about the same benefits using the strategy of unifying the MUSH prioritisations using MUSH (one iteration) mechanism to manage the critical situations**.

	$Union_U(MUSH_U)$ and MUSH	$Union_U(MUSH_L)$ and MUSH	$Union_L(MUSH_L)$ and MUSH
GAP	32.11%	45.39%	56.22%

Table 5.4: Union of MUSH preferences + MUSH (one iteration) mechanism for critical situations - GAP results

Figure 5.7 shows the summary of the distribution (in average) between the two airlines of the delay cost in all the considered mechanisms[4].

The Iteration MUSH method seems to be the best way in terms of reaching the global optimal solution (after SFP-SUSH strategy) that can be achieved (14% below FPFS costs) but its enormous complexity reduces its applicability. In the real world, the large number of flights and airlines involved in the hotspot problem shall generate difficulties to find the optimal solution with Iteration MUSH.

Unifying the AUs MUSH preferences using MUSH (one iteration) mechanism to manage the critical situations seems another excellent solution (10.5% - 13% below FPFS costs, worse than SFP-SUSH mechanism and Iteration MUSH), and with the additional shortcoming that the Network Manager has to choose one of the two airline MUSH prioritisation in a totally randomly manner. This type of policy makes the process unfair: the NM might be always giving higher priority (unintentionally) to some AU preferences rather than the others.

Consequently the "best" solution in order to reach the global reduction of delay costs is to unify the AUs MUSH preferences using FPFS policy to manage the critical situations (9% - 13% below FPFS costs). This strategy does not require the intervention of the Network Manager and it might be considered equitable by the AUs due to the FPFS policy and flexible enough because it allows a considerable reduction of flights delay costs.

[4]The Union of MUSH preferences mechanism is represented without considering the holes and "no solutions" problems

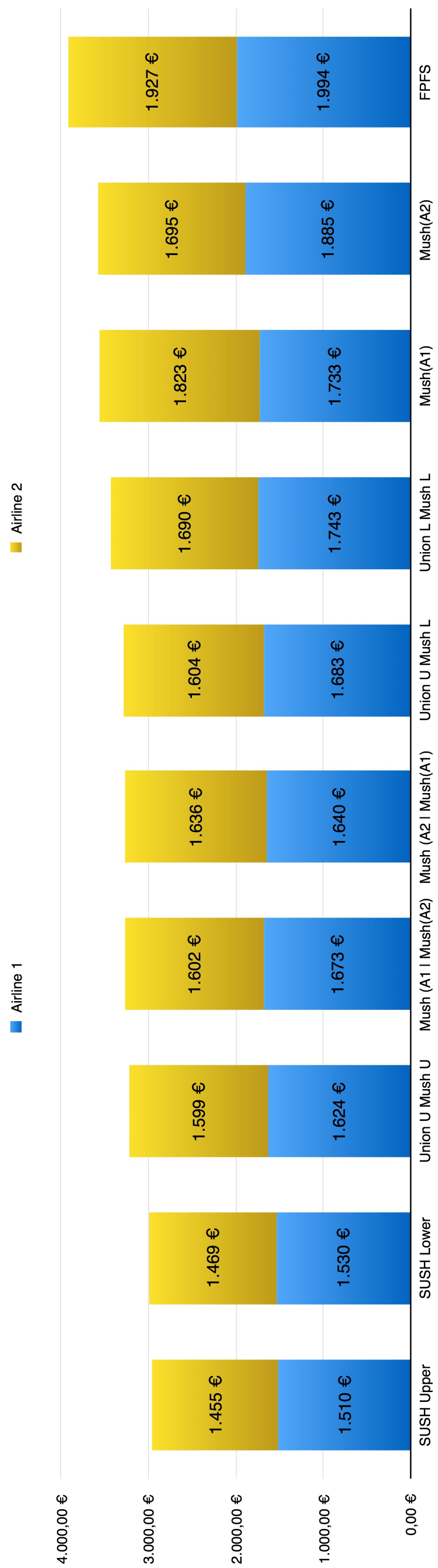


Figure 5.7: Summary of the distribution (in averange) between the two airlines of the delay cost in all the considered mechanisms

Chapter 6

Conclusions and future work

6.1 Conclusions

Through the use of the simulator and therefore the study of many instances, it was possible to find the positive aspects and the limits of the considered mechanisms: SUSH and MUSH.

The Single User Single Hotspot (SUSH) method brings the entire sector to save on average 24% of costs related to the delay. The idea of this strategy is to optimize the flight sequence with the SFP mechanism as a unique user (that could be the NM) in order to achieve a reduction of the total delay costs in the sector of all the AUs. Among the methods considered, this is the best because it seeks the total (and non-local) optimal solution.
SUSH, however, is affected by limitations for its application in real cases:

- The AUs should share all cost data. Airlines are reluctant to disclose sensitive data (especially to the competitors).
- This procedure aims to find the optimal solution for the entire sector. That can lead, by evaluating local solutions for each user, a local worsening (e.g. the suspension of all flights of a single airline and the protection of flights of an another rival company).

The Multiple User Single Hotspot (MUSH) mechanism, single iteration, brings the sector to save on average of about 8% of costs related to the delay. MUSH is most appropriate to represent the real decentralized decision making: the AUs seek to minimize their own delay costs only and they would not communicate their costs data which is sensitive information.
One of the goals of this research was to find ways to interpolate the MUSH results, studying the compatibility of several de-centralized and co-existing flight prioritisation decisions, improving the delay/priority management and getting the global solutions as closest as possible to SUSH solutions.

Two different methods have been studied for that purpose:

1. ITERATIVE MUSH. This procedure reflects the idea of a market where every AU can handle its flights schedules in response to other users choices. At each iteration, the solution improves (the value of GAP in average passes indeed from 65% (iteration 1) to 36.7% (iteration 2) and to 35.4% (iteration 3)) and this strategy converges to the optimal value of SUSH. For every iteration the grow rate of the improvement decreases and the running time for finding a solution increases exponentially. At the third step this procedure brings to save on average 15% of costs related to the delay (more iterations do not bring substantial solution enhancements).

   In some cases it was noticed that the final iteration MUSH solution is better than the optimal SUSH solution. This is due to the concept of iteration, that changes the initial conditions of the instance and leads to considerable margin of improvement (more combinations that can lead to better solutions). However, it is understood that if the same amount of iterations was applied to SUSH (Iterative SUSH), then the Iterative MUSH could not improve the Iterative SUSH.

2. UNION MUSH. This procedure seeks to unify (impartially) the various MUSH results of the various users. Each user expresses its MUSH results to the network manager who will try to compose the de-centralized prioritisation choices in order to obtain the final flight sequence. In this way the airlines do not give their sensitive data but only information about the own preferences, i.e., their protected and suspended flights. During the union it may occur the emerging of holes due a lot of flights suspended in the first slots. This problem, however, according to the simulator results, only is likely to occur in less than 1% of the cases. When it occurs, the NM has to take action in order to find a final solution that can be compatible with the prioritisation decisions of all the AUs. The NM has two options available to fix the "empty slot" problem:

   - applying FPFS scheduling. This is the most equitable solution. It reduces the delay costs of the system between 9% and 13%.
   - applying one MUSH solution. The NM chooses one of the two solutions MUSH (unfair) randomly. This reduces the delay costs of the system between 10.5% and 13%.

In the Table 6.1 are described the pros and cons of the various strategies.

Mechanism	**Average reduction of costs (Baseline FPFS)**	**Complexity**	**Equitable**[1]
SUSH	$\approx$ 24%	polynomial	NO
MUSH	$\approx$ 8%	polynomial	YES
Iteration MUSH	$\approx$ 15%	exponential	YES
$Union_{MUSH} + FPFS$	$\approx$ 9% − 13%	polynomial	YES
$Union_{MUSH} + MUSH$	$\approx$ 10.5% − 13%	polynomial	NO

Table 6.1: Final considerations

Ideally the MUSH iteration is the best method in terms of reaching the global optimal solution that can be achieved with MUSH ($\approx$ 15% below FPFS costs). However, its enormous complexity , due to combinatorial explosion, reduces its real applicability to small-scale case studied. The union of the MUSH solution with the use of one random MUSH solution in the case of incompatible solutions (holes, for 1% of the cases), is almost a good solution but it is inequitable for an user. The recommended mechanism then is the **union of the MUSH strategy**(which can be applied effectively in a 99% of the times) with the use of **FPFS scheduling in the case of incompatible solutions (holes)**: it's equitable and it has a polynomial complexity.

In the Table 6.1 is shown the summary of the results obtained with the simulator.

6.1.1 SUSH iterations

It is important to note that SUSH can lead to a more efficient flight position in the sequence than FPFS and SFP (1 step) scheduling. However, it does not necessary mean that SUSH can reach the optimal flight sequence. Nonetheless each time the SFP mechanism is iteratively used, it changes the initial condition for the next run. There is some likeliness to improve the previous situation, thus converting to the global optimum sequence solution.

Figure 6.1 shows a graph that represents an example of different flights scheduling algorithms in a sector. The graph root is the initial flights allocation. The ultimate goal is to find the path (the mechanism) that comes closer to the optimal solution. The higher the considered node, the better will be the solution.

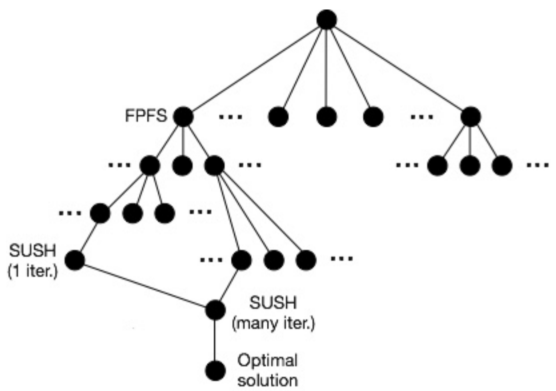


Figure 6.1: Example of different scheduling solutions

6.2 Limitations

In this research it is also important to highlight the limitations:

[1]The term *equitable* means here that all individual users do not receive the same amount of benefits relative to their effort.

1. All the instance of UDPP - MUSH are focused only the case with **only two different Airspace Users**. A larger number of users could bring totally different results using the same studied mechanisms.

2. During the simulations, a **limited constant number of flights (15)** is used while realistic cases have hundred of flights.

3. In the mathematical model, **the size of the slot is a constant quantity** of 2 minutes.

4. Difficulty to obtain and characterized **realistic cost values**.

6.3 Future work

Many future works related to this research are proposed:

1. Find new functional ways to interpolate the various MUSH results in order to obtain a more efficient reduction of the delay costs respect the mechanism studied in this research. One of the main proposal is to iterate several steps of the UNION of MUSH solution, as shown in Fig. 6.2.

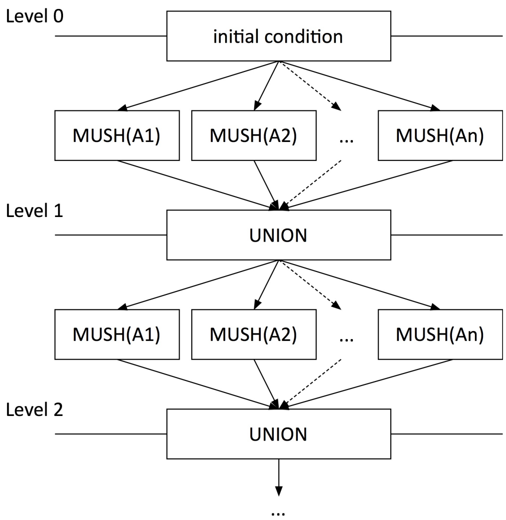


Figure 6.2: Proposal of Union iteration

2. Demonstrate the validity of SFP via testing with real data (real hotspot situations taken from European airports).

3. Evaluate how the SUSH solution improves with some iterations and study its link with iterations MUSH.

4. Several improvements on the simulator:

   (a) Model Part: the insertion of new parameters to make the simulation closer to reality: the rate of the cost than the delay, the number of airlines, the change of size of the slots, ... ;

(b) View Part: a graphic layer with buttons in order to directly manage the simulator without the use of terminal or IDE.

5. Another work directly related to this research is the creation of the mathematical model of the **Enhanced Selective Flight Protection (ESFP)**: the same problem treated in this research in the case of several hotspots occurred in different periods. ESFP proposes to enable the use of credits obtained by suspensions in one hotspot to protect flights that may be tactically important in future hotspots. It will increase the complexity of the problem and the number of variables considered. This is one of the current goals in UDPP SESAR project. This research could be extended to understand how the AUs could make their decisions of flight prioritisation and analyze their impact on other users.

6. The development of a mathematical model in which the size of the slot is not constant. This type of evolution approaches the theoretical model to a situation closer to reality and can therefore study possible improvements to the current system.

7. The mathematical modeling of a sale and purchase market of minutes of delay in the case of Capacity Constraint Situation (CCS): a potential evolution of SFP and ESFP. Airlines will now be able not only to prioritize and decide to suspend their flights since the end of the hotspot (the idea behind SFP/ESFP), but also to buy or sell minutes of delay in a market monitored by the Network Manager.

APPENDIX A - ORDERS OF COMPLEXITY

During this research it has been treated several times the orders of complexity in the algorithms found. In this appendix, it will be explained how that kind of orders are structured and it will be introduced the Big O notation [15].

Big O Notation

Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. It is how fast a function grows or declines. It is a member of a family of notations invented by Edmund Landau and Paul Bachmann (indeed it is called Bachmann-Landau notation (or asymptotic notation)). The letter O is used because the rate of growth of a function is called order.

Example To describe the complexity of an algorithm it is defined a function which evaluates the total number of steps (or the amount of time) given an initial problem of size n.
E.g. this value is $T(n) = 15n^3 - 6n^2 + 3n + 2$.
The constants and the slower growing terms will be ignored and it can be said that $T(n)$ grows at the order of n^3 and write: $T(n) = O(n^3)$.

For the formal definition, suppose $f(x)$ and $g(x)$ are two functions defined on some subset of the real numbers.

$$f(x) = O(g(x)) \quad for \quad x \to \infty \tag{6.1}$$

if and only if there exist constants N and C such that

$$|f(x)| \leq C|g(x)| \quad for\ all \quad x > N \tag{6.2}$$

f does not grow faster than g.

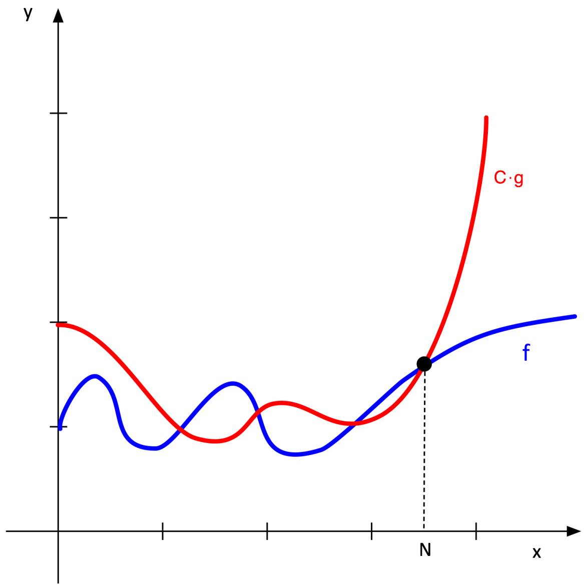


Figure 6.3: Example of Big O notation

Orders of complexity of common functions

Here is a list of classes of functions that are commonly encountered when analyzing the running time of an algorithm. c is a positive constant and n increases without bound. The complexity are inserted in the Table 6.2 in ascending order.

Table 6.2: Orders of complexity of common functions

Notation	Name
$O(1)$	constant
$O(log(n))$	logarithmic
$O(n)$	linear
$O(n^2)$	quadratic
$O(n^c)$	polynomial
$O(c^n)$	exponential
$O(n!)$	factorial

Family of Bachmann - Landau notations

Table 6.3: Family of Bachmann - Landau notations

Notation	**Name**	**Intuition**	**Definition**
$f(n) = O(g(n))$	Big O	f is bounded above by g	$\exists k > 0 \quad \exists n_0 \quad \forall n > n_0 \quad \lvert f(n)\rvert \leq k \cdot \lvert g(n)\rvert$
$f(n) = \Omega(g(n))$	Big Omega	f is bounded below by g asymptotically	$\exists k > 0 \quad \exists n_0 \quad \forall n > n_0 \quad f(n) \geq k \cdot g(n)$
$f(n) = \Theta(g(n))$	Big Theta	f is bounded both above and below by g asymptotically	$\exists k_1 > 0 \quad \exists k_2 > 0 \quad \exists n_0 \quad \forall n > n_0$ $k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n)$
$f(n) = o(g(n))$	Small O	f is dominated by g asymptotically	$\forall k > 0 \quad \exists n_0 \quad \forall n > n_0 \quad \lvert f(n)\rvert \leq k \cdot \lvert g(n)$
$f(n) = \omega(g(n))$	Small Omega	f dominates g asymptotically	$\forall k > 0 \quad \exists n_0 \quad \forall n > n_0 \quad \lvert f(n)\rvert \geq k \cdot \lvert g(n)$
$f(n) =\sim g(n)$	On the order of	f is equal to g asymptotically	$\forall \varepsilon > 0 \quad \exists n_0 \quad \forall n > n_0 \quad \left\lvert \frac{f(n)}{g(n)} - 1 \right\rvert < \varepsilon$

Bibliography

[1] Eurostat, *Passenger transport statistics*, data from January 2016.

[2] EUROCONTROL, *Seven-Year Forecast*, February 2015.

[3] SESAR, *European ATM Master Plan*, 2015.

[4] T. Vossen, M. Ball, *Optimization and Mediated Bartering Models for Ground Delay Programs*, Naval Research Logistics 51, pp. 75-90, 2006

[5] L. Castelli, R. Pesenti, A. Ranieri, *The design of a market mechanism to allocate Air Traffic Flow Management slots*, Transportation research part C: Emerging technologies 19 (5), 931-943, 2011.

[6] europa.eu *Welcome to the SESAR project*, 2015.

[7] EUROCONTROL Lexicon, *UDPP*, www.eurocontrol.int/lexicon/lexicon/en/index.php/UDPP.

[8] S. Ruiz, *Strategic Trajectory De-confliction to Enable Seamless Aircraft Conflict Management*, Ph.D Thesis, 2013

[9] SESAR Joint Undertaking. http://www.sesarju.eu.

[10] NETBEANS, https://netbeans.org

[11] R. Hoffman, *Integer Programming Models for Ground-Holding in Air Traffic Flow Management*, Ph.D Thesis, 1997

[12] FICO Xpress Optimisation Suite, www.fico.com

[13] D. G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, Third Edition, Springer, 2008

[14] EUROCONTROL, *European Network Operations Plan 2015 - 2019*, 2015

[15] MIT - Introduction to Computer Science and Programming - Lecture 8: Efficiency and Order of Growth

[16] Apache POI - the Java API for Microsoft Documents, https://poi.apache.org

[17] C. Demetrescu, I. Finocchi, *Algoritmi e strutture dati*, McGraw-Hill, 2008

[18] $FICO^{TM}$ Xpress Optimization Suite, *MIP formulations and linearizations*, 2009

[19] A. Cook, G. Tanner, R. Jovanovic, A. Lawes, *The cost of delay to air transport in Europe - quantification and management*, 13th Air Transport Research Society (ATRS) World Conference, 27-30 June 2009, Abu Dhabi, United Arab Emirates, Paper No. 107

[20] S. Martello, *Fondamenti di ricerca operativa*, progetto Leonardo, 2006

[21] P. Serafini, *Ricerca Operativa*, Springer, 2008

[22] SESAR, *UDPP Credits feasibility report*, 2015

[23] Computing in Science and Engineering, volume 2, no. 1, 2000

[24] M. Ball, C. Bamhart, G. Nemhauser, A. Odoni, *Air Transportation: Irregular Operations and Control*, in G. Laport and C. Barnhart, editors, Handbooks in Operations Research & Management Science: Transportation, Handbooks in Operations Research and Management Science. North Holland (8 Dec 2006), 2006.

[25] L. Castelli, R. Pesenti, A. Ranieri, *Allocating Air Traffic Flow Management Slots*, International journal of revenue management 6 (1-2), 28-44, 2011