Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Jan Straka**

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: **Air Traffic Control Simulation in Terminal Area**

Guidelines:

1. Get familiar with air traffic, air space, and procedures related to operations in a terminal area.
2. Design general architecture of a simulated air traffic controller behavior model that works in a terminal area.
3. Implement general configurable air traffic controller model in the AgentFly system.
4. Implement specific air traffic controller model for selected airport and study influence of the behavior to an air traffic.

Bibliography/Sources:

[1] Federal Aviation Administration: Air-traffic procedures
[2] M. S. Nolan. Fundamentals of Air Tra_c Control. Thomson Brooks/Cole, Belmont, CA, USA, 4th edition, 2004.
[3] D. Šišlák, P. Volf, M. Pěchouček, C.T. Cannon, D.N. Nguyen, W.C. Regli. Multi-Agent Simulation of En-Route Human Air-Traffic Controller. 24. IAAI Conference, 2012

Diploma Thesis Supervisor: Mgr. Přemysl Volf, Ph.D.

Valid until the end of the summer semester of academic year 2015/2016

L.S.

prof. Ing. Jiří Žára, CSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, November 4, 2014

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Master’s Thesis

Air Traffic Control Simulation in Terminal Area

Bc. Jan Straka

Supervisor: Mgr. Přemysl Volf, Ph.D

Study Programme: Open Informatics

Field of Study: Software Engineering

January 5, 2015

Acknowledgements

This work would not have been possible without help from my supervisor Přemysl Volf and other colleagues from the AgentFly team, namely to Lukáš Koranda and Tomáš Blovský.

I would also like to thank my family for their continuous support during my studies.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on January 5, 2015 ....

Abstract

The aim of this thesis is to extend the multi-agent simulation software AgentFly with a module adding the capability to simulate air traffic control in terminal area. First the air traffic control procedures in airspace near airports are analyzed and then the simulation module is designed to allow configurable simulation of approach phase of the flight and scheduling arriving airplanes to runways using different planning algorithms. Then the simulation module is implemented and finally different scheduling methods are evaluated to determine the one providing the best results for real-world simulation tasks.

Abstrakt

Tato práce se zabývá rozšířením multi-agentního simulačního software AgentFly o modul, který umožňuje simulovat řízení letového provozu v terminální oblasti. Nejdříve jsou analyzovány procedury pro řízení letového provozu v blízkosti letiště. Dále je navržen modul pro konfigurovatelnou simulaci přiblížení na letiště a rozvrhování přistání na několika ranvejích s využitím různých plánovacích algoritmů. Navržený modul je implementován a jednotlivé rozvrhovací metody jsou vyhodnoceny a je vybrána ta, která poskytuje nejlepší výsledky pro úlohy simulující reálný letový provoz.

Contents

1 Introduction **2**
  1.1 Motivation – Why Is Air Traffic Simulation Necessary? 2
  1.2 Thesis Goals 3

2 Introduction to Air Traffic Management **4**
  2.1 Brief History of Aviation 4
  2.2 Current ATM Equipment 6
  2.3 Air Traffic Control 6
    2.3.1 Air Traffic Control Services 6
      2.3.1.1 Area Control Service 7
      2.3.1.2 Approach Control Service 7
      2.3.1.3 Aerodrome Control Service 7
    2.3.2 Coordination Within ATC 8
      2.3.2.1 Transfer of Control Between Sectors 8
      2.3.2.2 Coordination Between ACC and APP 8
  2.4 Airspace Classes 9
  2.5 Flight Plan 10
    2.5.1 Fix 10
    2.5.2 Route 11
      2.5.2.1 STAR 11
      2.5.2.2 SID 11
  2.6 Used Units 11
    2.6.1 Altimetry 11
  2.7 Separation 12
    2.7.1 Vertical 12
    2.7.2 Horizontal 13
      2.7.2.1 Lateral 13
      2.7.2.2 Longitudinal 13
    2.7.3 Separation in Terminal Area 13
      2.7.3.1 Wake Turbulence Separation 14
  2.8 Holding Pattern 14
  2.9 Horizontal Diversion Manoeuvre 16
  2.10 Miles In Trails 16

3 Data 17
3.1 Hartsfield–Jackson Atlanta International Airport 17
3.2 Atlanta TMA 18
3.3 Flights 19
3.4 STARs 20
    3.4.1 Route Duration Estimation 21

4 Flights Scheduling 22
4.1 Term Definitions 22
4.2 Problem Definition 22
    4.2.1 Existing Algorithms Solving Similar Problems 23
    4.2.2 Evaluation Criteria 23
4.3 Runway Plan Visualization 24
4.4 Slot Allocation on Single Runway 25
    4.4.1 Algorithm 0 – Force Slot on ETA 25
    4.4.2 Algorithm 1 – Keep Order of Appearance on Radar 26
    4.4.3 Algorithm 2 – Find First Empty Space in Which The Slot Fits 26
    4.4.4 Algorithms 3a, 3b – Keep Order of ETA (With Local Optimization) . 27
    4.4.5 Algorithms 4a, 4b, 4c – Branch And Bound 28
4.5 Runway Selection 30

5 AgentFly 31
5.1 Air Traffic Model 31
    5.1.1 Pilot 31
    5.1.2 Air Traffic Controller 32
    5.1.3 Communication 32
5.2 Visualization 32

6 Implementation 33
6.1 Data Processing and Visualization 33
    6.1.1 Runway Plan Visualization 33
    6.1.2 Approach Routes 34
    6.1.3 Wake Turbulence Separation Intervals 34
    6.1.4 Airplane Trails 34
6.2 Scheduling 34
    6.2.1 STAR Selection 36
6.3 STAR Application 37
    6.3.1 Flight Plan Update 37
6.4 Sector Entry Check 38
6.5 Flow Control 38
    6.5.1 Vectoring 39
    6.5.2 Holding Pattern 40
6.6 Airplane Landing 41

7 Experiments **42**
7.1 Comparison of Scheduling Algorithms on Single Runway 42
7.1.1 Makespan 44
7.1.2 Maximal Delay 45
7.1.3 Total Delay 46
7.1.4 Number of Replanned Slots 46
7.1.5 Conclusion 47
7.2 Real World Flights on Single Runway 48
7.3 Real World Flights on Multiple Runways 49
7.4 Real World Flights With Miles In Trails 51

8 Conclusion **53**
8.1 Future Work 54

Bibliography **55**

A Enclosed Videos **57**

B Contents of Enclosed CD **58**

List of Figures

2.1 Airspace division and control [7, Chapter 2.10] 7
2.2 Airspace Classification Example [3] 9
2.3 Holding pattern with examples of entry 15
2.4 Schematic representation of the horizontal diversion manoeuvre in ATC plan 16

3.1 Section of the airport diagram of the Hartsfield–Jackson Atlanta International Airport [19] 17
3.2 Left: Atlanta Class B airspace in AgentFly 3D visualization, Right: The airspace as shown on radar screen in AgentFly with borders of neighboring en-route sectors 18
3.3 Left: histogram of number of airplanes appearing on screen in each hour of the simulated 24 hours traffic, Right: chart of representation of aircraft with different weight classes in the test air traffic 19
3.4 STARs near Atlanta Airport, Right: overall view, Left: detailed view of the configuration near runways 20

4.1 Example of runway plan with colliding slots 25
4.2 Example of runway with slots in the order of airplane's first appearance . . . 26
4.3 Example of runway plan with slots fixed in place 26
4.4 Example of runway plan that preserves the order of ETA 27
4.5 Comparison between the three variants of Algorithm 4 28
4.6 Example of adding later slot causing changes in plan before its ETA, first row shows plan before addition of slot `D`, second row shows plan after addition . . 28

6.1 Flowchart showing simplified processes of ATC in TMA sector as implemented by `EomAtaStarApplication`. 35
6.2 Continuation of the flowchart showing simplified processes of ATC in TMA sector from Figure 6.1. 36
6.3 Visual representation of horizontal diversion manoeuvre. Left: ATC plan with planned diversion. Right: trail of airplane that performed the diversion manoeuvre and landed. 39
6.4 Visual representation of holding pattern. Left: ATC plan with airplane completing first round of the holding. Right: trail of airplane that performed two rounds of holding (one full and one shortened). 40
6.5 Snapshot of visualization output during simulation with real-world flights. . . 41

7.1 Result of planning with different algorithms on single runway 43
7.2 Chart of makespan for plans created with different algorithms on single runway 45
7.3 Chart of maximal delay for plans created with different algorithms on single runway 45
7.4 Chart of total delay for plans created with different algorithms on single runway 46
7.5 Chart of cumulative sum of replanned slots for plans created with different algorithms on single runway 47
7.6 Result of planning with different combinations of algorithms on multiple runways 50
7.7 Visualization of MIT restriction application in en-route sectors. 52
7.8 Comparison between planning with and without MIT. 52

List of Tables

2.1 Wake separation minima [17] 14

3.1 Results of estimation of routes duration 21

4.1 Configuration of the scenario illustrating different outputs generated by individual algorithms 25
4.2 Run times of the three variants of Branch and bound algorithm 29

7.1 Configuration of the test scenario with alternating weight classes on single runway 42
7.2 Results of different scheduling algorithms for dense real-world traffic on single runway 48
7.3 Result of planning with different combinations of algorithms on multiple runways 49

List of Used Abbreviations

ATM	Air Traffic Management
ATC	Air Traffic Control
ATFM	Air Traffic Flow Management
AIS	Aeronautical Information Services
FP	Flight Plan
GPS	Global Positioning System
SID	Standard Instrument Departure
STAR	Standard Terminal Arrival Route
ACC	Area Control Service
APP	Approach Control Service
TWR	Aerodrome Control Service
CTA	Control Area
CTR	Control Zone
TMA	Terminal Control Area
IFR	Instrument Flight Rules
VFR	Visual Flight Rules
RWY	Runway
ETA	Estimated Time of Arrival
ICAO	International Civil Aviation Organization
FAA	Federal Aviation Administration
LOA	Letter of Agreement
RVSM	Reduced Vertical Separation Minima
VOR	VHF Omnidirectional Radio Range
DME	Distance Measuring Equipment
ILS	Instrument Landing System
NAS	National Airspace System
MIT	Miles in Trails
BADA	Base of Aircraft Data
ZTL	Atlanta Air Route Traffic Control Center
HITL	Human in The Loop
CSV	Comma Separated Values
NM	Nautical mile = 1.852 km
kt	Knot = 1 NM per hour
ft	Foot = 30.48 cm
FL	Flight Level = 100 ft

Chapter 1

Introduction

Aviation is the most time-effective form of widely available means of long distance transportation. The amount of people and freight transported grows every year and is expected to continue growing in years to come. For example, in 2014 some 3.2 billion passengers used air transport, which is a 5% increase compared to 2013, and this number is expected to reach over 6.4 billion by 2030. [1]

The growth of air traffic brings increased demands on the capacity of airspace. One way to address this issue is to build bigger aircraft capable of carrying more passengers and cargo at once, but there are physical limits on the size of airplanes. Therefore the utilization of available airspace must be improved in order to increase its capacity. At the same time, the safety of air traffic must be ensured.

The Federal Aviation Administration (FAA) of the United States in cooperation with International Civil Aviation Organization (ICAO) are working on new flight rules that would increase the capacity of airspace while still taking into account the safety restrictions, the aircraft dispositions, the workload of pilots and ground personnel and the airport capacities.

1.1 Motivation – Why Is Air Traffic Simulation Necessary?

The field of aviation is ever-evolving and the air traffic control needs to keep up with the changes, especially with the growing number of aircraft flying to and from large hub airports leading to congestion and potentially resulting in high fuel expenses for waiting airplanes or worse a risk of mid-air collisions between aircraft. The rules for air traffic control must evolve with growing traffic to accommodate it.

Air traffic simulation is used to test and evaluate the impact of increasing and changing traffic or any changes made to the rules of traffic control operation before their actual implementation in real world. Primarily the impact of the changes on safety of the flights and the workload of both pilots and ground controllers is observed. Testing changes in flight rules in real air traffic would be difficult and dangerous and therefore computer simulation of traffic is used.

Simulations are carried out with air traffic controllers involved in the process controlling the simulated traffic. These simulations are called human in the loop (HITL) and are quite time consuming and expensive especially when they involve inter-sector cooperation and therefore require the presence of several human controllers at once.

There is an effort to minimize the costs connected to HITL simulations by running the tests with computer simulated air traffic controllers first and possibly eliminating the changes in the flight rules that didn't turn out promising before the costly HITL simulation. This approach can significantly reduce the time and the financial demands of the testing leading to cheaper and faster implementation of new air traffic control rules.

AgentFly system provides the capability to simulate civilian air traffic in US National Airspace System (NAS) including the simulation of air traffic controllers in en-route sectors and their workload while performing the control duties. These include keeping the aircraft separated from each other at any moment, handing control over the aircraft from one sector to another, keeping the aircraft on route to their final destination and controlling the flow of air traffic.

1.2 Thesis Goals

The goal of this thesis is to design a model of air traffic controller in terminal area, to implement it in AgentFly simulation system and to conduct and evaluate experiments with real-world simulation scenarios. Specific goals for this thesis are:

1. **Problem analysis** – The goal is to understand and analyze the task of terminal air traffic controller. The procedures for controlled flight in the vicinity of airports are described in Chapter 2.

2. **Data gathering and processing** – The goal is to gather data needed to conduct simulations of terminal air traffic control. The data acquisition is described Chapter 3

3. **Scheduling algorithm design** – The goal is to design algorithms that simulate the controller's deliberation on scheduling arriving flights to land on controlled airport, each algorithm offering different approach for arranging the flights for landing. Chapter 4 is dedicated to description of proposed algorithms.

4. **Implementation in AgentFly** – The goal is to implement proposed scheduling algorithms as controller module in AgentFly simulation system. Also the current methods of simulating aircraft need to be extended to allow for simulation of the approach phase of the flight. The implementation is described in Chapter 6.

5. **Evaluation** – The goal is to conduct experiments testing the different scheduling methods in both specifically designed and real-world scenarios in order to determine their performance according to different criteria. The scheduling method demonstrating best results is then selected for future real-world simulation tasks. Experiments are described in Chapter 7.

Chapter 2

Introduction to Air Traffic Management

This chapter will provide basic introduction to the field of ATM (Air Traffic Management), starting with brief summary of the history of aviation and continuing with explanation of some basic terms and concepts that will be needed further.

The Air Traffic Management is an collective term for three services provided in aviation: Air Traffic Control (ATC), Air Traffic Flow Management (ATFM) and Aeronautical Information Services (AIS). The responsibility of ATC is to keep aircraft safely separated at all times, be it in the air or on the ground. ATFM prevents congestion in busy areas by predicting and controlling the flow of future air traffic. AIS is responsible for distribution of aeronautical information that are needed to maintain safe, efficient and regular air traffic. These information can contain for example current whether situation or notification on potential hazards at specific locations. [2]

This thesis focuses on Air Traffic Control and Air Traffic Flow Management in the vicinity of airport. The presence of relevant information in given place at appropriate time is assumed and not addressed in this thesis.

2.1 Brief History of Aviation

The history of aviation began on 17th December 1903 with the first flight of Wright brother's powered fixed-wing airplane heavier than air. [3] In first years after the maiden flight the aviation was considered only a dangerous pastime for daredevils, but year after year the early flying machines were becoming more capable and safe. By the end of World War I airplanes would prove themselves useful for observation and weapon delivery.

After war several uses were found for airplanes. One of them would be application of pesticides in agriculture, but the most important for further development of aviation would be probably delivery of mail. At this time most flights were conducted in daytime but with the rising demand for air mail delivery first experiments with flight in night were performed, first using bonfires for navigation and later replacing bonfires with gas and electrical lighting.

In 1930s commercial industry began to form in aviation and with increasing number of airplanes in air the need for some way of air traffic control became apparent. First new on-board instruments were designed to allow flight at certain altitude and direction without visual reference to the ground, and later system of ground radio navigational aids was introduced to assist pilots with navigation in low-visibility conditions. First air traffic controllers were located at airports where the traffic was most dense. They stood on well visible place on the airfield and would either permit the take-off or landing with a green flag or prohibit the pilot from proceeding with intended manoeuvre and hold position until it's safe to proceed.

This early control had many disadvantages and quickly evolved into system using light guns to guide the pilots near airfield. The basic principle was the same with green meaning "go" and red meaning "stop", only flags were replaced with focused light guns that allowed to better target the instructions to specific airplanes. Also the operation of light guns could be carried from elevated control tower which improved visibility for both pilots and controllers as well as provided more comfort for controllers.

Later two-way radio system between control tower and aircraft was introduced and allowed pilots to confirm issued instruction and controllers to transmit additional information regarding traffic, weather, etc. Unlike light guns that required direct line of sight between tower and aircraft, this system could be also used in low-visibility conditions.

Soon the airspace became more and more crowded and it was needed to control the traffic not only in the vicinity of airports but also on the routes between them. This was done by Air traffic control units (ATCUs) that would separate the air traffic on federal airways during instrument flying conditions. When the visibility was good it was still responsibility of pilots to separate their flight from surrounding traffic. The ATCUs were equipped with radio and airspace maps and would update the position of each aircraft on the map based on flight plan filled by the pilot before the flight and periodical position reports over the radio. This way the controllers at ATCU were able to keep the aircraft separated.

The World War II brought great advance in both aviation technology as well as the amount of air traffic. Approach control service was formed at the busiest airports, its task was to sequence the incoming and outcoming traffic so it will arrive to the airport in regular intervals and therefore making the task of tower control easier. The radar proved itself useful in military environment but needed some development to work well for civilian air traffic control. So in 1950s first air route surveillance radar became operational and the radar system was soon after improved with transponders on board of controlled aircraft that would allow to display the flight id and altitude on the radar screen.

New positioning system for aircraft was also developed around VHF omnidirectional range (VOR) and Distance measuring equipment (DME) using ground-based radio beacons. There were around 3000 VOR stations built around the world.[11] Because this system needs ground stations to operate it can't provide navigation over oceans or other places the stations cannot be built on.

The following years brought continuous evolution of the used equipment and procedures. Instrument landing system (ILS) was introduced to allow landing in adverse meteorological conditions. Computers soon found their way into control towers. They allowed the controllers to work with flight plans without printing them out and to transfer the FP between sectors automatically without using telephone which was used before. Computerized radar system was developed that would show additional information directly on the radar screen.

2.2 Current ATM Equipment

The current equipment of the traffic controllers is the result of the improvements made in the few previous decades. The computer system controllers use aggregates data from various sources and shows them conveniently on the screen. These data include aircraft positions computed from signals from multiple radar stations, flight id, altitude, speed and flight plan.

In order to be able to display the additional information the secondary radar system must be used.[3] Standard radar emits radio waves and measures the interval between the pulse and when the waves reflected from any solid objects arrive back. This way the position of objects can be determined but the system is prone to interference and reflections from tall buildings, mountains or even cloud formations. The secondary radar also emits an interrogation signal and aircraft equipped with transponder will respond according to the interrogation mode. This way the airplane's flight id and altitude can be shown on the radar screen. The aircraft speed is computed from a few previous positions of the aircraft and its altitude.

Another tool used in air traffic control nowadays is air traffic flow management (ATFM). This system predicts the air traffic density based on the available flight plans and if it reaches the capacity of the destination airport or sector the aircraft is delayed on the ground before it even takes of saving considerable amount of fuel. The process of the computing the airspace capacity utilization is very complex and influenced by many factors most important being the weather and is therefore automated and handled by computers.

2.3 Air Traffic Control

The main objective of Air Traffic Control is to prevent collisions between aircraft in air or on the ground and to expedite the flow of air traffic. [7, Chapter 2.2] In order to do so effectively, the airspace is divided to separate areas which are controlled by different ATC services depending on their characteristics.

2.3.1 Air Traffic Control Services

ATC services can be divided into Area Control Service (ACC), Approach Control Service (APP) and Aerodrome Control Service (TWR) [5, Chapter 1]

The airspace in which ATC service is provided can be divided into Control Area (CTA) and Control Zones (CTR). Within CTA, Terminal Control Areas (TMA) are established to help in arrival and departure at some airports.

Control Zones are normally situated below CTA and encompass airspace used by flights arriving at and departing from aerodromes. CTR extends from the ground at least to the lower limit of CTA, but may extend further. CTR may include several aerodromes situated close together. [7, Chapter 2.10] The Figure 2.1 shows how the controlled airspace can be divided and which services provide control in which areas.

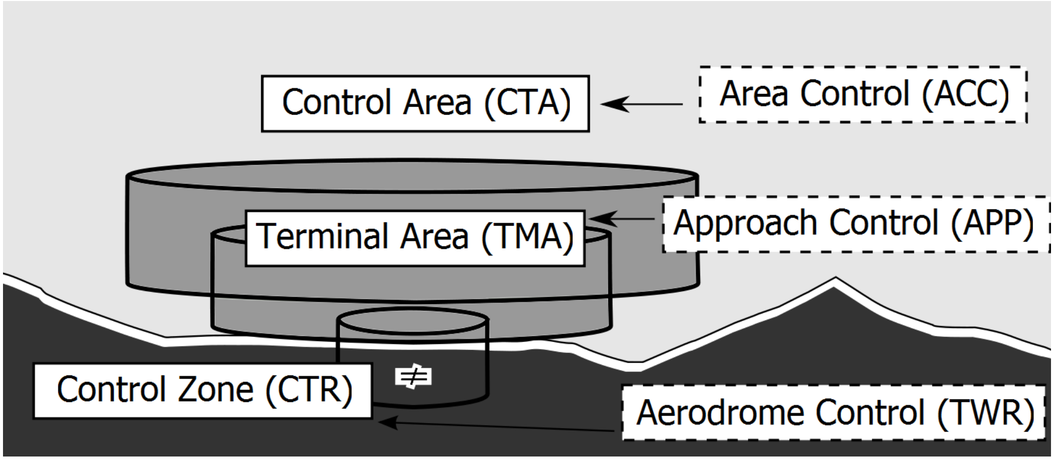

Figure 2.1: Airspace division and control [7, Chapter 2.10]

2.3.1.1 Area Control Service

Area Control Service is an ATC service provided by Area Control Center responsible for flights in Control Areas (CTA). Normally ACC is identified by the name of a nearby city, area or landmark. Smaller countries usually have one ACC, but many larger countries are controlled by several. ACCs usually control aircraft in their en-route phase of flight. The ACC may be also responsible for flights to and from smaller aerodromes with no separate approach control service. [7, Chapter 3.2]

Control Area contains airways, Terminal Control Areas (TMA) and other airspace. It extends upwards from specified altitude.

2.3.1.2 Approach Control Service

Approach Control Service (APP) is ATC service that is responsible for the part of CTA and CTR required by arriving or departing controlled flights. This area is called Terminal Control Area (TMA or TCA) in U.S. or Terminal Manoeuvring Area (TMA) in Europe. The primary functions of APP is sequencing arriving aircraft and assisting departing aircraft in becoming established on course. The arrival and departure functions can be divided into several positions on busy aerodromes. APP is usually identified by the name of the aerodrome which it is serving, but sometimes it's not collocated with TWR and is at distant ACC location away from the airport it serves. When no separate APP exists, approach control service is provided by ACC or TWR. [7, Chapter 3.2]

2.3.1.3 Aerodrome Control Service

Aerodrome Control Service is provided by a control tower (TWR) and is responsible for aircraft landing and taking off. It's also responsible for airplanes flying under Visual Flight Rules (VFR) in the CTR and for preventing collisions between aircraft on the manoeuvring area of the aerodrome. [7, Chapter 3.2]

2.3.2 Coordination Within ATC

The airspace in CTA is further divided to sectors to allow for effective control. The shape and size of the sectors is determined by usual traffic going through them. Near busy aerodromes lay smaller sectors and larger sectors are located in higher altitude or places with low traffic like Alaska.

TMA airspace isn't further divided, but rather controlled by single sector. TWR controlled airspace is located inside CTR sector.

The sectors are grouped within Air Route Traffic Control Centers. Each center has its controller who provides high-level view on the air traffic flow and manages overall traffic flow especially in cases like approaching storms that require traffic flow to be diverted across several sectors.

The coordination can take place between ATC centers, between ACC and APP, APP and TWR and between individual sectors inside ACC. The coordination between ATC sectors is usually conducted in relation to transfer of control of individual flights or while planing the air traffic flow. The detailed rules and procedures of the communication are often defined in local Letter Of Agreement (LOA) between the sectors.

2.3.2.1 Transfer of Control Between Sectors

Every flight must be controlled by single ATC unit at any time. When the flight plan of an aircraft crosses several sectors, they must transfer the responsibility for the flight along the way. The transfer of control must be approved by the accepting sector and the point of transfer agreed by both units.

The transfer can be divided into three stages: first the pilot is notified to prepare to change sectors, then the conditions of the transfer are negotiated and finally after agreeing the control is transferred to the accepting sector. This process can be achieved by automated means without the need of traditional telephone coordination between the sectors. [5, Chapter 10]

2.3.2.2 Coordination Between ACC and APP

After ACC releases a flight to APP, APP may apply clearances to the aircraft without reference to ACC. In case of missed approach, ACC is informed as soon as possible if aircraft will fly back to CTA.

The take-off time is specified by APP with respect to traffic in the TMA sector. The take-off time can be also specified by ACC in case it needs to coordinate the departure flow with other ACC traffic. APP and ACC can also agree on expiry time for departure clearance if delay of take-off would cause separation problems.

APP keeps ACC informed about used runway and type of instrument approach procedure, minimal time between successive arrivals, revision of estimated arrival times for transferring flights, departure times, missed approach times and other information about controlled traffic that may affect ACC. ACC informs APP about identification, type and point of transfer of incoming aircraft, anticipated delay to departures because of congestion and other information agreed on by both parties.[5, Chapter 10]

2.4 Airspace Classes

The controlled airspace can be also classified in one of the Classes A-G. The classification is mutually exclusive so Class A airspace can't be also classified as Class D at the same time and in the same place. The classes define the flight rules used in the airspace, interactions between pilot and Air Traffic Control and most importantly the responsibility for avoiding other aircraft which can be assigned either to the ATC or to the pilot himself. [7]

The definition of the classes is given by ICAO and is respected by most countries although some differences in the specification can exist. The use of the classes is nation specific, some countries classify their whole airspace into just one or few classes, some use all classes. Which classes are used in which parts of the airspace also varies greatly. [9] The typical use of airspace classes in USA can be seen in Figure 2.2.

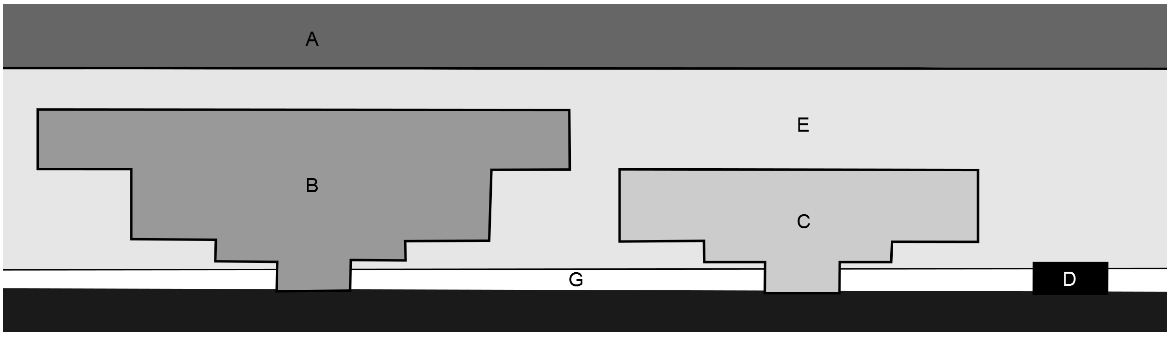

Figure 2.2: Airspace Classification Example [3]

Class A All flights in Class A airspace must be under ATC control and must operate under Instrument Flight Rules (IFR). Flights entering Class A airspace must receive clearance from ATC before the entry. ATC is responsible for separation of all flights in Class A airspace.

In USA Class A airspace extends upwards from approximately 5,500 m to 18,000 m. Above 18,000 m Class E airspace is used.

Class B In Class B airspace Visual Flight Rules flights are permitted in addition to IFR flights. All aircraft must have clearance from and are separated by ATC.

In USA Class B airspace usually surrounds largest and busies airports. The shape of the airspace differs from one airport to another, but usually reminds an upside down cake around the airport. The individual layers often don't form complete circle but their shape is adapted to the configuration of the airport, local topology and traffic patterns. Vertically Class B airspace normally reaches approximately 3,000 m. [4, Chapter 3]

All flights entering Class B airspace must ask for clearance before entry. Because Class B airspace is so busy and the traffic is very dense, there are some additional restrictions are placed on the aircraft equipment and pilot training and certification. VFR flights are restricted and in some Class B airspaces prohibited.

Class C Class C airspace also allows both IFR and VFR flights but does not provide separation for VFR flights. Only traffic information are provided for VFR flights. The structure of Class C airspace is similar to Class B airspace only in smaller scale. It is usually located around airports with moderate traffic. The vertical upper bound is at approximately 1,200 m.

Class D In Class D airspace IFR and VFR flights may be conducted but separation is provided only among IFR flights. Traffic information is provided to VFR flights and to to IFR flights regarding VFR traffic. The shape is usually cylinder circa 760 m high. Class D airspace is usually located around small airports and is in place only during operation hours of the airport. It reverts to Class E or G when the tower is closed.

Class E Class E airspace is controlled airspace that allows IFR and VFR flights. IFR flights are separated from each other, traffic information is provided to all flights in respect to VFR flights. In most parts of USA Class E airspace extends from 370 m to 5,500 m and then above 18,000 m. For airports with no tower Class E airspace can extends from the ground level in order to transition aircraft between terminal and en-route airspace. [3]

Class F Class F airspace is uncontrolled, separation is provided to IFR flights if possible. Class F airspace is not used in USA.

Class G Both IFR and VFR flights are allowed in Class G airspace but separation is not provided. Traffic information may be given, if requested. Class G airspace lays normally near ground under Class E airspace. Class G airspace is completely uncontrolled, there are no clearance requirements and even radio communication is not required.

2.5 Flight Plan

Every controlled aircraft has its flight plan (FP) that contains basic information about the flight. These information include aircraft identification, aircraft type, departure point, departure time, route of flight, destination, fuel on board and a few other. [4, Chapter 5] The purpose of the flight plan is to give all relevant data about the aircraft to the air traffic controller in a standardized format so it can be used efficiently during the flight control. The flight plan is handed over from one control to another as the aircraft progresses from its departure airport to destination along the flight route.

The flight route is a sequence of fixes and routes the airplane will fly through during its flight.

2.5.1 Fix

Fix is named point on Earth that aircraft use for their navigation. The position of fix can be determined by radio beacons or as GPS position for airplanes equipped with GPS receivers.

If aircraft has certain fix in its flight plan, it must fly over the position of the fix during the flight. This way the route of the aircraft is defined and automatic tools can be used to notify the pilot and air traffic controller if the airplane deflects too far from the trajectory defined by its flight plan.

2.5.2 Route

Route is named sequence of fixes. There can be some additional restrictions added on the route, for example altitude at given fixes or type of aircraft that can use the route. The flight plan does not need to include the whole route, the airplane can be ordered to follow only part of the route between two fixes.

2.5.2.1 STAR

Standard Terminal Arrival Route (STAR) is a special kind of route that is used to navigate airplanes approaching an airport. It links a significant fix with a point from witch the aircraft can safely land using instrument approach procedure. [5] This basically means the STAR leads the aircraft from its en-route flight to a touch-down on a runway.

2.5.2.2 SID

Standard Instrument Departure (SID) route works the other way around. It leads the aircraft from the runway to a specified significant fix from which the airplane can continue with its en-route flight.

2.6 Used Units

Heading is the direction in which the airplane's nose is pointing. It is measured in degrees in clockwise direction with 0° being north, 90° east, 180° south and 270° being west. Note that the direction of flight doesn't have to correspond to the airplane's heading, this happens typically in situations with crosswind when the the heading must be adjusted according to the wind's direction and strength in order to achieve desired direction of flight.

Distances are measured in nautical miles (NM) with 1 NM equal to 1.852 km.

Speed is measured in knots (kt) with 1 kt being 1 NM per hour. Indicated airspeed is the actual reading from airplane's instruments. Calibrated airspeed adjusts the value for instrument, installation and position errors. True airspeed describes the speed of the aircraft relative to the surrounding mass of air. At standard sea level conditions the calibrated airspeed and true airspeed are equal. Ground speed is the airplane's speed relative to the ground instead of the air, as the air itself can be moving.

2.6.1 Altimetry

Aircraft altitude is measured either in feet (ft) or in flight levels (FL). Flight level is defined by a specific barometric pressure and is expressed in hundreds of feet. The pressure is derived from the International standard sea-level pressure (1013.25 hPa) and therefore may not equal to the actual altitude. To determine the true altitude the pilot must know the atmospheric pressure at given space as the reading from the instruments depend on it. This value is provided by the air traffic control. The true altitude can be expressed either as altitude above sea level or above ground level, which value is used depends on agreement between pilot and air traffic controller.

Because the altitudes measured in feet and in flight levels don't equal the vertical separation cannot be ensured between aircraft using different measuring method. Therefore a transitional altitude is established. Above this altitude the altitude is measured in flight levels which removes the necessity to update the local air pressure. Below the transitional altitude the true altitude is used, because precise altitude values are needed for take-off and landing. The transition altitude and corresponding transition level are established by the air traffic control. Before crossing this altitude ground controller must provide the transitional altitude and local air pressure to the pilot. AgentFly currently uses altitudes derived from the International standard sea-level pressure.

2.7 Separation

Along with transferring control over aircraft from one sector to another and applying standard operating procedures for take-off or landing is keeping air traffic separated one of the main duties of air traffic controller. Proper separation ensures safety and eliminates risk of collision. [10, Chapter 2]

Vertical or horizontal separation must be provided for all flights in Class A and B airspaces, IFR flights in Class C, D and E airspaces and between IFR and VFR flights in Class C airspace. [5, Chapter 5]

The rules air traffic controllers use to keep aircraft separated are called separation minima. Separation can be achieved in two basic ways: vertical and horizontal. Their description follows.

2.7.1 Vertical

Vertical separation is reached when ATC controller assigns airplanes different cruising levels. Historically vertical separation minima were set to 1000 ft (10FL) below FL290 and 2000 ft above FL290. This was due to the fact that altimeter precision decreases with increasing altitude. Over time more precise altimeters and combination with other means of measuring aircraft altitude allowed to decrease the 2000 ft separation minima and introduce the Reduced vertical separation minima (RVSM) system. In the RVSM airspace the separation limit is 1000 ft up to FL410 and 2000 ft above this level. If the cruising altitude of two airplanes is greater or equal as the separation minima, the airplanes are considered as separated. [4]

Other rules can be applied, often different flight levels are reserved for aircraft flying in certain direction. Example of such rule can be that eastbound flights (with heading 0° – 179°) fly on even tenths flight levels (FL20, FL40 etc.) and westbound flights (with heading 180° – 359°) fly on odd tenths (FL10, FL30 etc.).

2.7.2 Horizontal

If two aircraft fly in the same altitude, they must be separated horizontally. The separation can be either lateral or longitudinal. The separation can be achieved both with or without using radar. When the radar is available, the controller sees the position of the aircraft directly on the radar screen. Without radar procedural control must be used which means that the controller must rely on position updates from the pilots over radio.

2.7.2.1 Lateral

Lateral separation presumes that the aircrafts follow different routes whose reserved airspace do not overlap. [3, Chapter 7] The width of the airway is 8NM. Aircraft are also considered laterally separated if holding over different fixes whose holding pattern airspace does not overlap. The lateral separation can be decreased to be less than 8NM in specified cases like two airplanes crossing ways over a fix flying in different directions.

2.7.2.2 Longitudinal

When two airplanes are flying along the same route or converging routes and are not separated vertically, they must use longitudinal separation. For the separation to be applied to two aircraft, both must fly in the same speed or the leading aircraft must fly faster than the following.

The longitudinal separation can be defined as a time delay between when two aircraft fly over a specified position or as minimal distance between the aircraft in nautical miles (NM). The separation minima differ if the airplanes are flying on the same track, or on reciprocal or crossing tracks.

For airplanes flying on the same track the basic separation minimum is 15 minutes, if the navigational aids allow for frequent position and speed updates the separation may be decreased to 10 minutes. If the preceding aircraft flies 20 kt or more faster than the following, the separation is 5 minutes, if the preceding aircraft flies more than 40 kt faster, the separation can be as low as 3 minutes. [5, Chapter 5]

For airplanes on crossing paths the separation minimum is 15 minutes or 10 minutes if frequent position updates are available. The same applies for airplanes flying on the same track crossing paths vertically

2.7.3 Separation in Terminal Area

In terminal area similar separation rules apply as in en-route sectors. The aircraft must be separated either vertically, laterally or longitudinally. Vertical separation is used for holding aircraft. STARs are defined in such way that aircraft flying along the routes are separated either laterally on the same altitude or vertically if the STARs cross. This way the separation of aircraft flying on different STARs is always ensured. Airplanes flying on the same STAR use longitudinal separation with reduced separation minima that reflect the possible hazard of wake turbulence.

2.7.3.1 Wake Turbulence Separation

Wake turbulence is the air vortex formed behind every aircraft as it moves through the air. The strength and size of the vortex depends on the parameters of the aircraft (mass, size, speed, wing configuration, etc.). It's not visible and can cause hazardous situation for the following aircraft especially if it is smaller and lighter and therefore more susceptible to be affected by the vortex.

Special alertness is needed in conditions with light wind, because the turbulence won't dissipate quickly and may drift to parallel runway or descend to lower altitude into path of following aircraft.

ICAO specifies three categories of the aircraft based on their maximum certificated take-off mass as follows [5, Chapter 4]:

- Heavy - 136 000 kg or more
- Medium - 7 000 – 136 000 kg
- Light - 7 000 kg or less

Super Heavy category is also used to provide separation behind very large aircraft like Airbus A380-800.

		Following aircraft			
		Super Heavy	Heavy	Medium	Light
Preceding	Super Heavy	2.5 NM	6 NM	7 NM	8 NM
	Heavy	2.5 NM	4 NM	5 NM	6 NM
	Medium	2.5 NM	2.5 NM	2.5 NM	4 NM
	Light	2.5 NM	2.5 NM	2.5 NM	2.5 NM

Table 2.1: Wake separation minima [17]

Table 2.1 shows the separation minima defined by FAA used for aircraft following behind another aircraft at the same altitude or less than 1000ft when both aircraft are using the same runway or parallel runways separated by less than 2500ft.

2.8 Holding Pattern

Holding procedure is a predefined manoeuvre that keeps the aircraft in predetermined airspace while waiting for clearance. The procedure is the same for VFR and IFR flights. Holding fix is a geographical location that serves as a reference point for holding procedure. The pattern itself is defined by the holding fix, heading of the inbound leg and length of the pattern. [5, Chapter 6] The pattern and some terms used for it's description are shown in Figure 2.3.

Reasons for holding can be traffic congestion, delays at destination airport or aircraft problems.

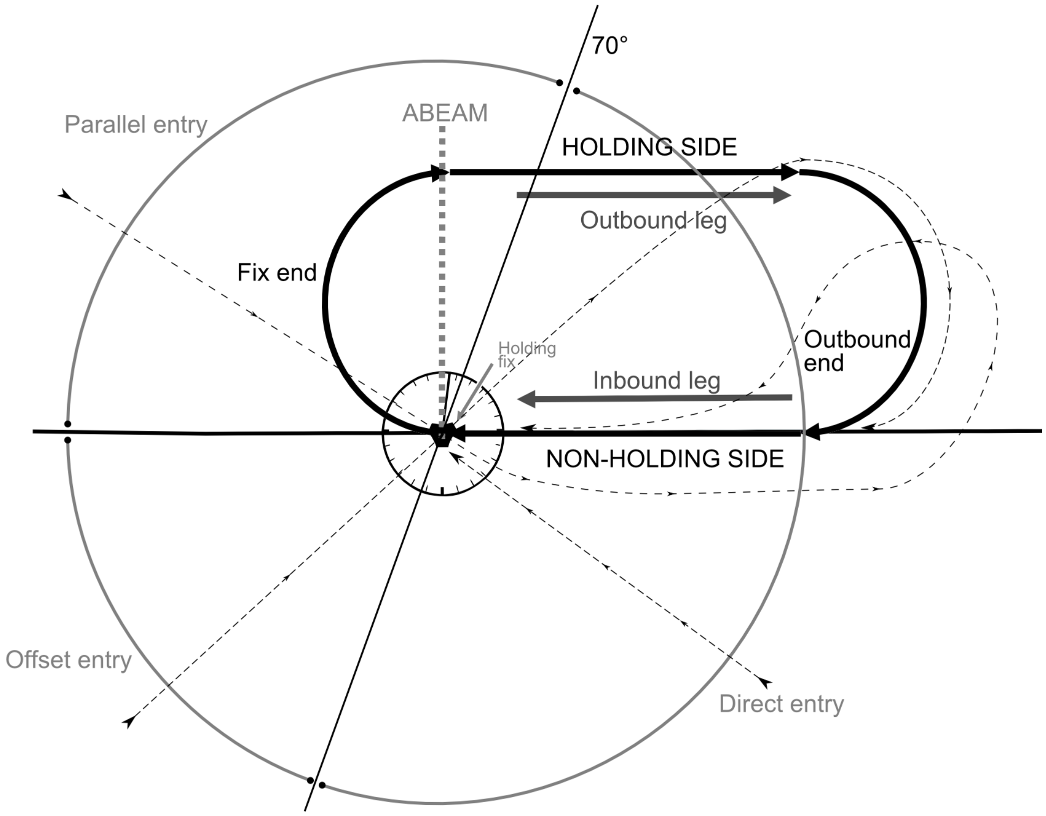

Figure 2.3: Holding pattern with examples of entry

The holding procedure is usually published beforehand but ATC can specify the details of an ad hoc holding pattern if the situation calls for it. The turn direction is usually right, but left-hand turn holding patterns can be used if needed.

Several airs can hold over the same holding fix, these airplanes must be separated vertically. Normally the airplane to arrive first holds on the lowest level with the following aircraft using successively higher levels. Jet aircraft can hold at higher levels to save fuel, but the order must be retained.[5, Chapter 6]

Maximum holding speeds are established by ICAO to keep the aircraft within the protected holding space. Aircraft can also have specific holding speed prescribed by the manufacturer. This speed is lower than typical cruising speed and is used in order to conserve fuel. The maximum speeds are defined as follows [6, Section 6]:

- up to 14000 ft: 230 kt
- 14000 ft to 20000 ft: 240 kt
- 20000 ft to 34000 ft: 265 kt
- more than 34000 ft: 0.83 Mach

There are three different entry procedures for the holding pattern depending on in which direction the aircraft arrives to the holding fix. Direct entry is straight forward, the airplane flies directly to the holding fix and turns outbound as soon as the holding fix is reached. In offset entry aircraft flies over the holding fix into the protected area and through the area and then turns back at the outbound end and continues the holding from there. In parallel entry the airplane flies over the holding fix and continues parallel to the inbound leg on the non-holding side. At the outbound end the airplane turns and continues back to the holding fix an holds from there. The entry procedures are indicated in Figure 2.3.

2.9 Horizontal Diversion Manoeuvre

The horizontal diversion manoeuvre can be used either to keep separation between aircraft in collision avoidance or to extend the aircraft flight plan in order to postpone its arrival to given location (typically airport).

The manoeuvre is defined by the diversion angle, length of the diversion and return point. The ATC orders the pilot to divert to given magnetic heading and await further instruction. The pilot turns as soon as possible abandoning the current flight plan. When the airplane reaches desired position, ATC orders the pilot to change direction back to a return point and resume flight on the original flight plan from the return point on.

Figure 2.4 shows the representation of diversion manoeuvre in the ATC plan. The diversion is represented by the whole area of the shape as the precise times of the turn applications are uncertain before the pilot actually performs them. The ATC plan is updated and the area of uncertainty gets smaller as the flight progresses through the manoeuvre.

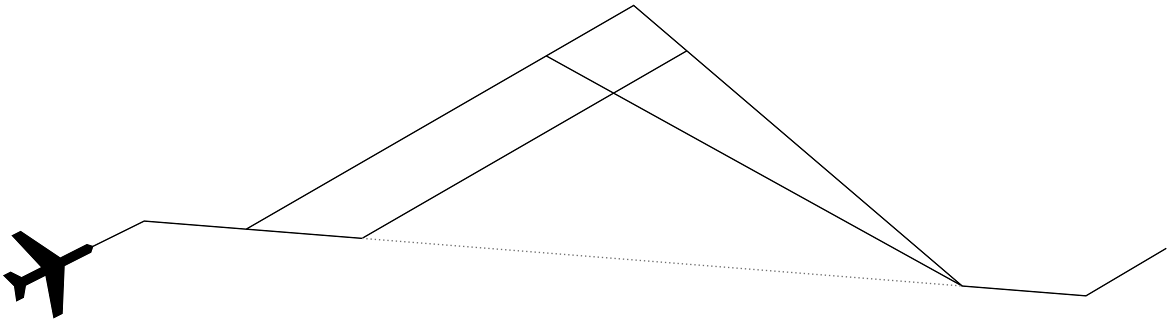

Figure 2.4: Schematic representation of the horizontal diversion manoeuvre in ATC plan

2.10 Miles In Trails

Miles in trails [12] is a set of restrictions that ensures that aircraft entering a sector are separated by desired time interval at the time they cross the sector border. The MIT restrictions are often used to manage arrival flows into TMA sectors and to protect the destination airport particularly when weather reduces the airport capacity or in cases of unexpected interval of high volume traffic.

The MIT separation intervals are designed so they allow merging of two streams of airplanes from different en-route sector into single stream leading to runway.

Chapter 3

Data

This chapter summarizes which data were used for implementation and testing of the module for air traffic control simulation in terminal area. The design and implementation of the module is general and can be used for simulation of any TMA sector but in order to test its functionality concrete sector description and air traffic scenario must be used. It was important to have all the data relevant to the test sector available because any of them missing would cause the whole scenario to be unusable.

3.1 Hartsfield–Jackson Atlanta International Airport

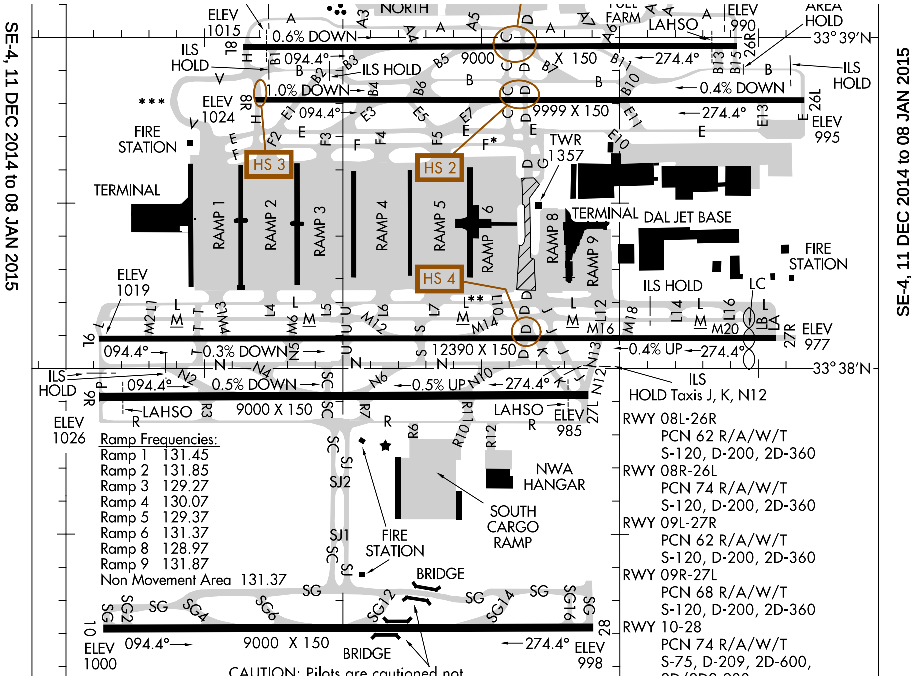

Figure 3.1: Section of the airport diagram of the Hartsfield–Jackson Atlanta International Airport [19]

Hartsfield–Jackson Atlanta International Airport (ICAO code KATL) has been the world's busiest airport by passenger traffic since 1998 with more than 94 million passengers in 2013. It has also the most landings and take-offs since 2005. The airport has 207 domestic and international gates which is the most at any airport. [18]

This airport was chosen because the amount of traffic would provide more interesting test-case in comparison to some other, less busy airport. Also all the other required data were available for Atlanta Airport.

The goal of this thesis is to simulate the approach phase of the flight, from the moment airplane leaves en-route sector to touchdown on the runway. The ground movement is not simulated and therefore the only information needed about the airport is the configuration of its runways.

The runway configuration is publicly available on the FAA website [19]. Airport runway configuration including positions and elevations of each runway was created in a form of XML file that can be used in AgentFly simulation.

The Atlanta Airport has five runways, the southernmost (`10/28`) is used for cargo aircraft and was therefore eliminated from the scenario. Out of the four remaining runways the inner two (`8R/26L` and `9L/27R`) are used exclusively for departures and only the outer two runways (`8L/26R` and `9R/27L`) are used for incoming flights. These two runways will be used in the test simulation scenarios. Figure 3.4 shows all the Atlanta Airport runways (in yellow) as they are shown in AgentFly visualization.

3.2 Atlanta TMA

Together with the description of the Atlanta Airport itself, the definition of surrounding TMA sector was needed. This sector describes the area of authority of approach control service. The name of the TMA sector is `A80`. As the Atlanta Airport is one of the busiest airports, the associated TMA sector is defined as Class B airspace. The definition of the airspace is provided by Federal Aviation Administration [20].

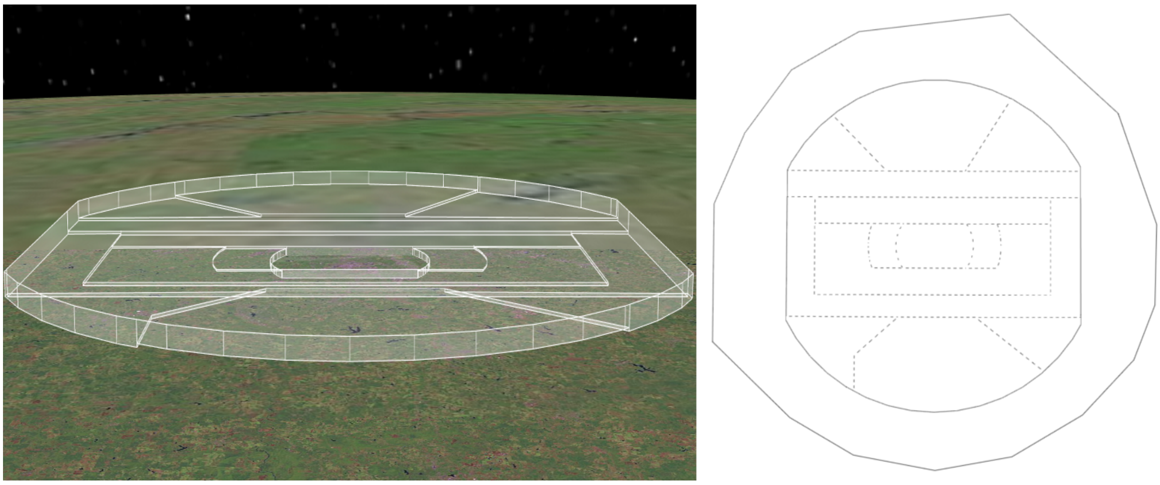

Figure 3.2: Left: Atlanta Class B airspace in AgentFly 3D visualization, Right: The airspace as shown on radar screen in AgentFly with borders of neighboring en-route sectors

The definition of the airspace isn't available in a computerized form ready for machine processing and had to be converted to XML sector definition manually from the available textual description [20]. The resulting sector as seen in agentFly visualization is shown in Figure 3.2. Problem with this TMA definition is that there is a gap between TMA border border and previously provided neighboring en-route sectors making it impossible to hand-off airplanes from one sector to the other. Upon consulting with FAA, for the purposes of testing the borders of surrounding en-route sectors were used for TMA definition instead of the Class B airspace itself. This way the TMA sector touches the en-route sectors allowing airplane hand-offs. The comparison between Class B airspace and used sector definition is shown in Figure 3.2 right. The TMA sector and some neighboring en-route sectors is also shown in Figure 3.4 left, note that the sectors do not overlap, the en-route sectors horizontally touch the TMA sector and partly extend above it.

3.3 Flights

Recording of real-world 24 hour traffic in Atlanta Air Route Traffic Control Center (ZTL) on 20th June 2013 was provided by FAA for testing. The log files contain records of 7832 flights. Out of those VFR flights were filtered out because only IFR flights are simulated in AgentFly. Also aircraft of unusual type that can not be simulated using BADA simulation models were removed from the test data. Also flights flying through the `A80` sector or taking of at the Atlanta Airport were filtered from the logs leaving 1308 flights arriving to the airport.

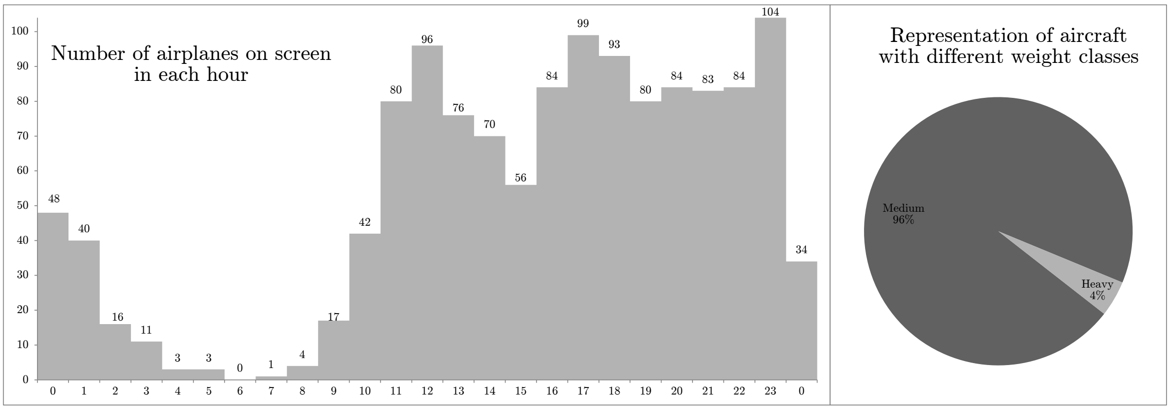

Figure 3.3: Left: histogram of number of airplanes appearing on screen in each hour of the simulated 24 hours traffic, Right: chart of representation of aircraft with different weight classes in the test air traffic

Figure 3.3 shows the histogram of number of airplanes appearing on screen in each hour and chart of representation of aircraft with different weight classes in the air traffic. The histogram demonstrates the usual progress of the traffic during the day with decline in early morning and peaks around noon and in the evening. The pie chart shows that the traffic going to the Atlanta Airport is rather monotonous 96% of medium weight aircraft, 4% heavy aircraft and no light or super heavy airplanes. Therefore additional test scenario was created to test the scheduling algorithms in situations with more variable composition of arriving flights.

3.4 STARs

STARs are routes that connect the en-route sectors and lead the airplanes to the runways. They define the path itself as well as safe intervals for altitude and speed for selected fixes on the route. The routes are designed in such way that if they cross horizontally, the separation is ensured vertically. This means that once an airplane is flying on a STAR route there is no risk of collision with aircraft flying on other routes.

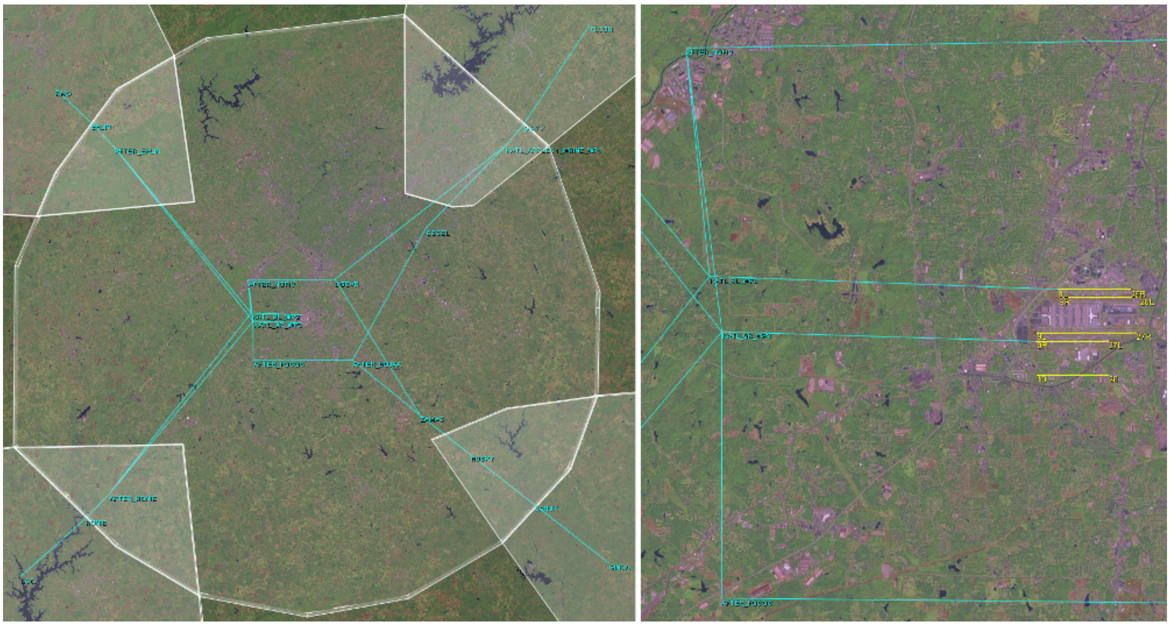

Figure 3.4: STARs near Atlanta Airport, Right: overall view, Left: detailed view of the configuration near runways

The STAR definitions are made publicly available by the FAA in non-machine readable format but were also provided for the purpose of TMA simulation in AgentFly in computerized proprietary format. The provided STAR descriptions were used in this thesis. Holding pattern definitions for the Atlanta TMA sector were manually obtained from the publicly available data, because the provided STAR descriptions didn't include them.

Figure 3.4 shows the loaded routes for eastern operation of Atlanta Airport in AgentFly visualization. Atlanta Airport can function in two configurations depending on the prevailing wind direction. The approaches and take-offs can be performed either in western or eastern direction. The route configuration is basically identical, only mirrored along the north-south axis. AgentFly doesn't simulate the wind at the moment and therefore static eastern direction was used for the testing purposes.

3.4.1 Route Duration Estimation

Estimated durations of STARs are an a priori knowledge of the air traffic controller who uses it to determine the approximate time of arrival for scheduling flights to runways. The durations of routes were not provided as part of the route description and therefore needed to be acquired experimentally.

Real-world flight data (presented in 3.3) were used to estimate the route durations. Three isolated runs were carried out in order to maximize the number of airplanes flying through each applicable route. The flights were launched separately one by one, STAR was applied by the neighboring en-route sector controllers and the duration of flight on the route was measured. No scheduling, holding or diversion manoeuvres were applied.

Route	# flights	Avg. route duration [m]	Accuracy margin [s]
SINCA → 8L	238	27:35	11.531
SINCA $\xrightarrow{AFTER\_FOGOG}$ 9R	238	24:46	11.493
SINCA $\xrightarrow{AFTER\_NOFIV}$ 9R	238	27:42	11.442
FLCON → 8L	391	25:55	10.678
FLCON $\xrightarrow{AFTER\_FOGOG}$ 9R	391	28:44	10.505
FLCON $\xrightarrow{AFTER\_NOFIV}$ 9R	391	26:02	10.568
LGC → 8L	303	19:59	6.562
LGC → 9R	303	19:27	6.580
RMG → 8L	353	17:21	6.969
RMG → 9R	353	17:26	6.887

Table 3.1: Results of estimation of routes duration

From the measured durations average route duration was computed to be used as the estimate in further experiments. Also an accuracy margin was determined to describe the precision of the estimate. It was chosen so that at least 90% of the test flights would arrive not sooner or later from the estimated duration. The resulting durations and accuracy margins are shown in Table 3.1.

Chapter 4

Flights Scheduling

Main goal of this thesis is to design and implement algorithm for scheduling flights arriving to airport controlled by approach controller. This chapter defines the scheduling task and presents several algorithms that provide variety of approaches to solve the task. All of the algorithms were then implemented and experiments were conducted to find the most appropriate for real-world simulation.

4.1 Term Definitions

First, let's define some terms that will be used in this chapter:

- **ETA** – Estimated Time of Arrival to the runway if the airplane will fly directly without delay along given STAR.
- **EETA** – Earliest ETA among all runways ariving flight may land on. If only one runway is available EETA = ETA. ETA is always bigger or equal to ETA.
- **SETA** – Scheduled ETA is the time at which the airplane is scheduled to arrive to runway. SETA is always bigger or equal to ETA.
- **Delay** – for single runway the delay of flight arrival is defined as SETA – ETA, for multiple runways the delay is defined as SETA – EETA.
- **Slot** – time interval assigned to arriving flight on a runway. It is defined by SETA surrounded by buffer given by accuracy of the estimate.

4.2 Problem Definition

The approach controller's main task is to schedule aircraft flying to an airport in controlled TMA sector, given following limitations:

- Individual flights arrive over time and the controller becomes aware of them at the time they appear on his/hers radar screen.

- Each flight needs to be assigned one of available runways to land on and time interval in which the assigned runway is free.
- The runway and slot assignment must be done at the time of their appearance on the radar screen, no flights are allowed to fly in the TMA sector not knowing which runway they will land on and when.
- The slot must be scheduled to given estimated arrival time (ETA) or later.
- Once the runway is assigned, it must not be changed.
- Assigned slot may be rescheduled to a later time, it may not be rescheduled earlier.
- The slots on single runway must be spaced at least by minimum wake separation interval length. The interval length depends on weight classes of both preceding and following airplane and on their order. This means that the separation interval of **Light** airplane behind **Heavy** airplane is different (longer) than interval of **Heavy** behind **Light**.

4.2.1 Existing Algorithms Solving Similar Problems

The problem of sequencing incoming flights is a task that TMA controllers face every day. The responsibility for safe and smooth approach of incoming flights lies on the human controllers, but there are computer algorithms that help them with the approach scheduling. These tools and algorithms are proprietary, non-public and often airport specific – they solve the task for a specific airport based on its size, configuration and complexity of the traffic.

The task the TMA sector controller deals with is a special case of online scheduling problem. There are several algorithms providing solutions for variety of online scheduling problems [21], but none were found, that would be applicable for this specific scheduling task. Some of the algorithms allow for jobs (corresponding to arrival slots) to wait before being assigned to one of available machines (corresponding to runways) which is not possible here. Other make the allocation of jobs to machines irreversible, in flight scheduling problem the machine (runway) selection is irreversible, but the job order on the runway is variable (with the restriction that jobs mustn't be rescheduled earlier). Some algorithms allow job preemption which is not allowed in flight scheduling. There are algorithms that allow temporal constraints between jobs, but the constraints are static, here the temporal constraints are dynamic, they only apply for jobs scheduled directly behind each other on single machine and depend on the job's order. Additionally the algorithms search for optimal solution according to single criterion, whereas in flight scheduling we look on several performance criteria and don't search for optimal result but try to find good solution by as many criteria as possible.

4.2.2 Evaluation Criteria

The criteria the TMA sector controller takes into account while scheduling approaching flights are following:

- **Makespan** – the length of scheduled runway plan, which is equal to the difference between SETA of last and first task on the runway. For multiple runways the makespan is defined as difference between SETA of last and first task among all runways. Makespan expresses the quality of the schedule in terms of system efficiency and airport throughput and safety, as plan with shorter makespan leaves more room for future arriving airplanes in case of unexpected increase in the incoming traffic.
- **Total delay** – sum of all delays of scheduled arrivals. It expresses the schedule quality in terms of social welfare as lower total delay means lower fuel consumption and thus lower costs for airlines and lower emissions.
- **Maximal delay** – maximal delay among all scheduled arrivals. It expresses the plan quality in terms of competitive welfare and safety as too long delay may cause the airplane to deplete it's fuel supply before landing and therefore lead to an accident. Also no airline want their flight to wait while other flights are landing and therefore more uniform delay distribution among all flights is desirable.
- **Number of replanned slots** – total number of replanned slots during the scheduling (one slot may be replanned multiple times after additional slot allocations). It expresses the plan quality in terms of controller workload and safety as high number of changes in planned slots require the controller to update the instructions given to pilots and thus increasing his/hers workload.

Because no existing algorithm providing desired properties was found, algorithms were designed specifically for the flight scheduling problem. The problem was divided to two parts. First, new slot is allocated on all available runways and then the resulting plans are compared and one runway is selected. This way the algorithm produces both slot and runway allocation. Algorithms for both parts of the problem are proposed in following sections but first visual representation of the runway plan is described so it can be used later to illustrate the outputs of individual algorithms.

4.3 Runway Plan Visualization

Runway Plan Visualization module was designed to provide representation of the runway plan that is quick and easy to comprehend. Figure 4.1 shows an example of such visual runway plan representation. The slots planned for approach are displayed horizontally with a timeline at the bottom with highlighted one and five minute marks.

Above the timeline there are rows representing individual runway plans. Headline of each runway plan is shown on the left side with the number of the runway on the top and names of every weight class in corresponding color shown below. Plans of different runways planned by the same algorithm are divided by gray dashed border. Plans of different runways planned by different algorithms are divided by solid black border.

Each slot in the plan is represented by a vertical green bar with a dark green line in it. The dark green line shows the exact time of SETA and the green bar surrounding it shows the interval of possible arrival as defined by the precision of arrival estimation. Flight ID is shown on the top, the color of the text indicates the weight class of the aircraft. Below the flight ID, gray wake turbulence intervals are shown for every weight class in the same order as in the plan headline. If the wake turbulence interval is active (there is an following airplane scheduled after the current slot), the corresponding interval is colored in the color of following airplane's weight class.

If there is a collision between the slots, active wake turbulence interval of the preceding slot and the slot itself of the following flight are shown in red color. If a slot is delayed, vertical orange line marks the time of original ETA and is connected to the delayed ETA with horizontal orange line. In multi-runway scenarios, earliest ETA among all runways (EETA) is marked by dotted orange line and is again connected with the delayed ETA with horizontal dotted line. This way the controller can distinguish which part of the delay is caused by the order of slot on the runway and which part of the delay is the runway selection responsible for.

Examples of algorithm outputs shown in this chapter are results generated by individual algorithms for scenario defined in Table 4.1. The scenario was chosen specifically to illustrate different results produced by the algorithms.

Flight id	Weight class	Appearance on screen	Estimated arrival time
01	MEDIUM	14:48	48:09
02	MEDIUM	17:36	44:23
03	JUMBO	19:00	46:11
04	JUMBO	19:24	44:11
05	MEDIUM	21:36	48:23

Table 4.1: Configuration of the scenario illustrating different outputs generated by individual algorithms

4.4 Slot Allocation on Single Runway

This section presents several algorithms designed to allocate arrival slot on single runway. The input of the algorithm is ETA, airpane's weight class and slot size, output is the runway plan with new allocated slot.

4.4.1 Algorithm 0 – Force Slot on ETA

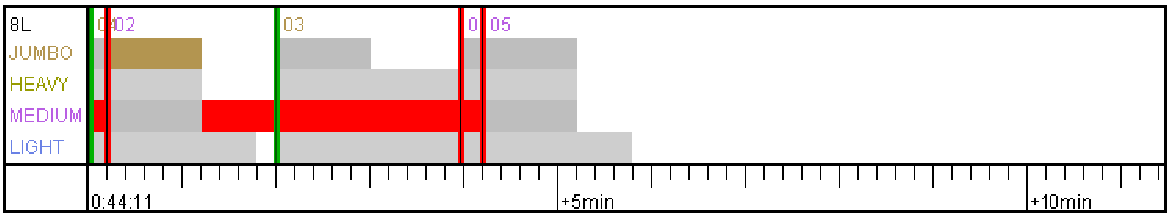

Figure 4.1: Example of runway plan with colliding slots

An example of a plan generated by Algorithm 0 is shown in Figure 4.1. This algorithm is the simplest of all implemented, it doesn't perform any deliberation on where to put the slot in the plan and simply places it in the time the aircraft is expected to arrive, ignoring possible collisions with other aircraft. This algorithm is not intended for actual use and serves only for comparison to other algorithms and to allow analysis of the traffic flow: it shows how many collisions there are or if the airplanes arrive to the runway periodically or in groups.

4.4.2 Algorithm 1 – Keep Order of Appearance on Radar

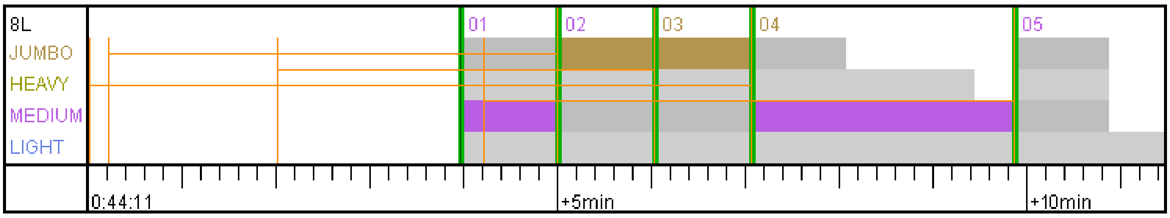

Figure 4.2: Example of runway with slots in the order of airplane's first appearance

Algorithm is a simple first-come, first-served algorithm that produces valid plans with no collisions between slots. When new airplane appears on radar screen it's slot is created after all previously planned slots and not sooner than at the airplane's ETA.

Figure 4.2 shows an example of a plan generated by this algorithm. It is clearly visible that unnecessary delays can occur when the interval between the time when the airplane shows on the radar and its ETA differ from airplane to airplane. This can happen when the arrival routes have different lengths. Approach route for airplane heading directly to runway will be much shorter than for airplane arriving from opposite direction, because such airplane must first fly around the airport before landing. See Figure 3.4 for example, the routes from south-west are significantly shorter than routes from south-east. In the example plan shown in Figure 4.2 flight `04` will arrive more than 7 minutes late because it appeared on the radar screen later than flights `01`, `02` and `03` even though its ETA is smallest.

4.4.3 Algorithm 2 – Find First Empty Space in Which The Slot Fits

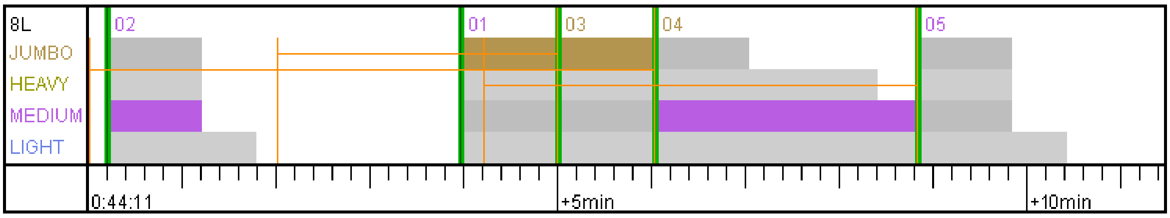

Figure 4.3: Example of runway plan with slots fixed in place

Algorithm 2 creates the slot for arriving airplane in the first empty space following airplanes ETA in which the slot fits. The wake turbulence separation minima is taken into account for both preceding and following slot so no collisions between slots occur. The advantage of this this algorithm is its simplicity and the fact that the planned slots are fixed. This means that if the controller assigns a slot to airplane and clears it for approach he/she doesn't have to alter the airplane's clearance and can focus on other arriving aircraft.

An example of a plan generated by Algorithm 3 is shown in Figure 4.3. In comparison to plan generated by previous algorithm (4.2) it is apparent that flight `02` can land sooner than `01` even though it entered the controlled area later. However `03` and `04` won't fit the void between `02` and `01`, and must be scheduled after. `05` would fit but arrives to the runway too late.

4.4.4 Algorithms 3a, 3b – Keep Order of ETA (With Local Optimization)

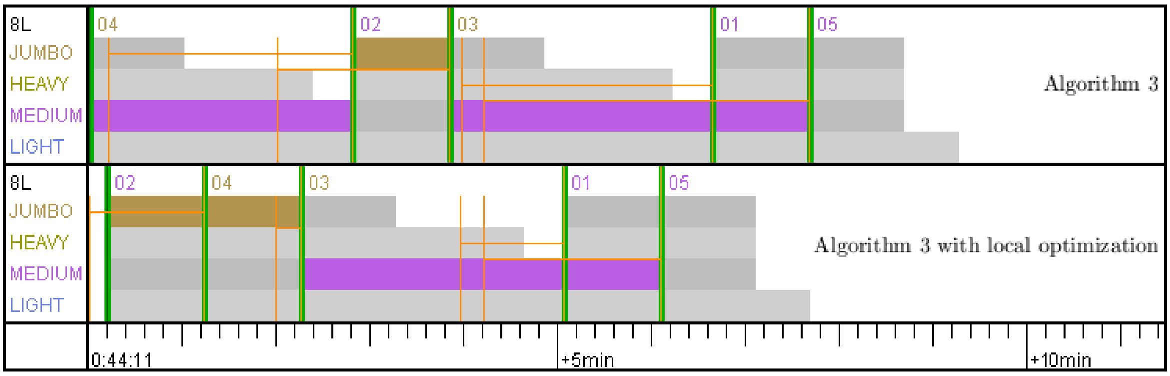

Figure 4.4: Example of runway plan that preserves the order of ETA

Third algorithm was implemented in two variants, the first one (*3a*) schedules the slots in a way that strictly preserves the order of estimated time of arrival. If the slot doesn't fit, any already planned slots following this one are delayed. If there is a continuous stream of airplanes scheduled for approach one after other and new, early arriving airplane appears on the radar screen the controller using this algorithm will squeeze the airplane in and postpone all airplanes following in the stream. This prevents a situation where the new airplane would wait for all the airplanes in the stream to land first. On the other hand the controller may need to postpone a significant number of previously planned aircraft which would take a considerable amount of time.

The second variant (*3b*) is not strict with the order according to ETA. It finds the place where the new slot would fit by the ETA, but inserts it only if the delay of the following slot after the new slot is inserted is smaller than the delay of the new slot would be if it was placed after the following slot. Otherwise it tries to insert the new slot after the slot following it and so forth until the condition is met. This serves as a local optimization of the maximal delay and helps the algorithm to cope with flights with alternating weight classes.

The example plans generated by this algorithm are shown in Figure 4.4, first row contains result of variant (*3a*), second row contains variant (*3b*). The difference between the algorithms is visible on the order of `02` and `04`. `02` is planned first (from these two) directly on its ETA. When the `04` was inserted into the plan, variant *3a* placed it before `02` because its ETA is smaller. Variant *3b* with the local optimization compared the delay of `02` behind `04` with delay of `04` behind `02` and find out that the second one is smaller and therefore pushed `04` behind `02`. Before placing `04` it compared it with `03` and when delay of `03` behind `04` was smaller than the other way around it placed the slot `04` between `02` and `03`.

4.4.5 Algorithms 4a, 4b, 4c – Branch And Bound

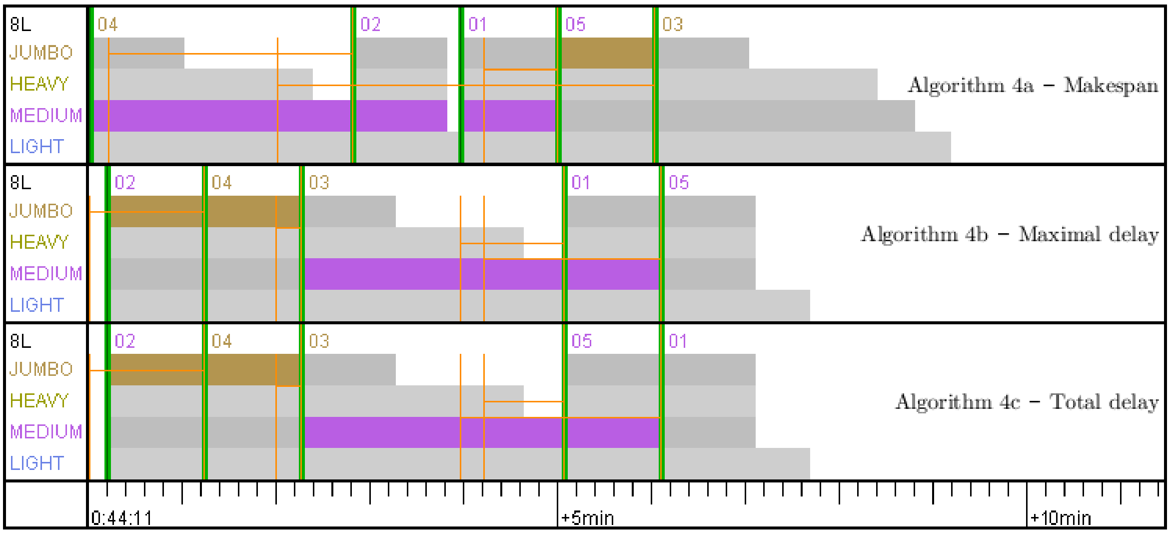

Figure 4.5: Comparison between the three variants of Algorithm 4

Algorithm 4 is an algorithm commonly used in combinatorial optimization and is called Branch and bound. [16] The algorithm is used here in three different variants, each optimizing different criterion. Variant *4a* minimizes makespan. Variant *4b* minimizes maximal delay. Variant *4c* minimizes the sum of delays of all planned slots.

Figure 4.5 shows a comparison between the three variants of Algorithm 4. It shows that the variant *4a* indeed produces plan with smallest makespan but with big delays for some slots. *4b* and *4c* produce plans with same total delay but *4b* has smaller maximal delay.

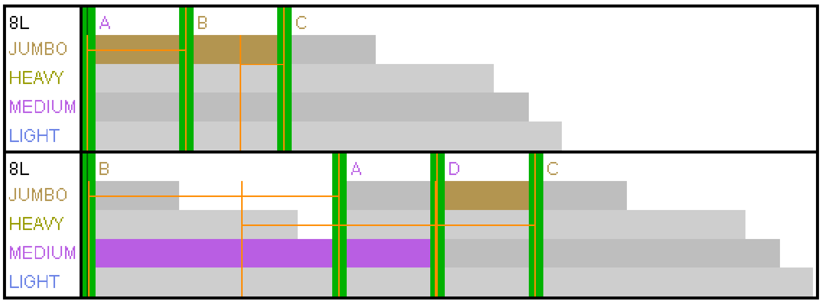

Figure 4.6: Example of adding later slot causing changes in plan before its ETA, first row shows plan before addition of slot `D`, second row shows plan after addition

There are two disadvantages of branch and bound algorithm that limit its use in real world applications.

The first one is that it doesn't keep the relative order of previously planned slots. This is caused by the fact that the algorithm is an offline algorithm and recomputes the whole plan from scratch every time new slot needs to be added. This is illustrated in Figure 4.6 where addition of slot for flight `D` caused the order of slots for `A`, `B` and `C` to change. Changing the order of flights scheduled on a route to one runway would result in need of special behavior that would allow the airplane to leave the route, wait in a separate area until the following aircraft flies past on the route and then return back to the route. And because the complete replanning takes place every time new airplane appears on a screen it could happen that two flights would switch their place in the sequence back and fort multiple times before reaching runway.

Additionally the planned delay may decrease for any given airplane in time (see `B` in Figure 4.6). But if the airplane already performed certain manoeuvre to slow it down to accommodate the delay prescribed by the previous plan, it may be impossible to speed up to reach the runway in time planned by the updated plan, even if it would be able to do so before the hold-up.

	10 slots	11 slots	12 slots	13 slots	14 slots
4a – Makespan	$< 1s$	$3s$	$37s$	–	–
4b – Maximal delay	$< 1s$	$< 1s$	$< 1s$	$7s$	$33s$
4c – Total delay	$< 1s$	$4s$	$45s$	–	–

Table 4.2: Run times of the three variants of Branch and bound algorithm

The second disadvantage of this algorithm is also linked to the need to recompute the whole plan with each slot addition and it is the computational complexity of the Branch and bound algorithm.

The algorithm enumerates possible solutions in a systematic way that ensures the optimum will be found eventually. To prevent searching through the whole state space, upper and lower bounds are used to prune the unpromising branches from the search tree. The minimal solution found so far can act as an upper bound pruning all branches with partial plans whose criterion value is already bigger or equal to the optimum. This is especially beneficial for the second variant minimizing maximal delay among all slots, because the pruning takes place early on in the search tree, eliminating many non-optimal solutions.

For optimizing makespan any empty voids in the plans can be used as a bound for the solution. This can be done if the slots are ordered according to their ETA. In such case if the ETA of the next added slot is later than the end of the previous slot and forms an empty void before it, the optimal solution lies in a tree rooted by the added slot. This is because rearranging the order of previous slots wouldn't allow the next slot to start sooner than it starts in the current partial solution. This bound cannot be used in variants 2 and 3, because rearranging previously planned slots can still improve maximal and total delay.

The sum of tasks that remain to be planned added to the length to the current partial plan can serve as an upper bound to the solution. If this value exceeds the value of the minimal solution found so far it is obvious that planning the remaining tasks cannot result into better result and current sub-tree can be pruned. This bound cannot be effectively used in this planning problem, because the size of the slots isn't constant and therefore the size of the sum depends on the order in which the slots are added to the plan. And finding the order of the slots that produces the smallest sum equals to the planning problem itself.

The previously mentioned restrictions limit the benefits of pruning and the algorithm must therefore search through a significant part of the state space which makes it slow. The Table 4.2 shows the runtime for rather small instances of the planning problem. The algorithm would not be useful for faster than real time simulation as is, but may serve as reference algorithm for small instances or in combination with online planning algorithm (for example for local optimization run on several neighboring slots around newly added one).

4.5 Runway Selection

When the configuration of STARs and the airport allows the airplanes to land on one of several runways, the air traffic controller must not only decide on the order in which the airplanes land on the runway but also on which airplane lands on which runway.

The procedure for runway selection goes as follows: For every runway the arriving airplane can land on, new plan is created incorporating the slot for the incoming airplane. The plans are created using the single-runway algorithms introduced in previous section. Then all the created plans are compared using one of the criteria presented below and the optimal plan is used for the corresponding runway.

- First criterion used to compare runway plans is the number of slots on each runway. The plan with **less slots** is selected. This will keep the number of slots on runways balanced leading to even distribution of flights between runways.
- Second criterion guides the airplane to the runway with the **smallest makespan** after the slot addition. This is another way of balancing the flights between the runways, this time taking into account the arrival times and not only number of slots.
- Third criterion selects the runway with smallest **maximal delay** of slots planned on the runway. Note, that for multiple runway scheduling delay is defined as difference between SETA and EETA.
- Fourth criterion for runway selection is **total delay** of all slots on the runway, again delay is defined as SETA minus EETA in this case.
- Last criterion used to compare runway plans is **SETA of the new slot**. The air traffic controller wants the airplanes to land as soon as possible and this criterion is focused on that. It compares the scheduled arrival times of the newly added slot and directs the airplane to the runway with lowest time. This way the airplane may fly to a runway which is further away if the runway is less utilized and allows for earlier landing than a runway that has the shortest route.

All algorithms were implemented in AgentFly system and then different combinations of slot allocation and runway selection algorithms were evaluated to determine which combination is best suitable for simulating real-world scenarios.

Chapter 5

AgentFly

The AgentFly is a large-scale high-fidelity distributed multi-agent simulation system. [14] The simulation framework consists of Aglobe middleware that provides infrastructure and communication between agents, simulation layer for managing virtual world and entities in it, and high level reasoning modules for simulation of airplanes and air traffic controllers [13].

5.1 Air Traffic Model

The AgentFly's simulation model consists four types of entities: pilot, airplane, visualization and ATC. The airplane agents model the flight itself, it computes it's movement in space using the Base of Aircraft Data (BADA) which is a database of mathematical models that describe the behavior and performance of modeled aircraft. [15] The visualization agents show the results of the simulation to the user. The air traffic itself is modeled using pilot and ATC agents.

5.1.1 Pilot

The aircraft in the simulation are operated by pilots who are modeled as pilot agents. These agents are event-driven and react to events sent from air traffic controllers. The pilot agent maintains internal flight plan that contains future intentions and actions of the pilot and therefore describes the precise future flight path of the airplane. It consists of waypoints defining the flight route which are connected with elements that describe how the airplane's position, altitude, speed and other characteristics change between the waypoints. When the pilot receives request from ATC, the agent applies the requested change to the flight plan and recomputes new flight states. If there are no instructions from the air traffic controller, the pilot just follows the precomputed flight plan.

The pilot's flight plan uses GPS coordinate system to describe the movement of the airplane because it corresponds to the real movement of the airplane in 3D space.

5.1.2 Air Traffic Controller

The ATC model in AgentFly is divided into two types of modules: ATA modules model complex computational tasks connected to the ATM computer and interaction with it – watching radar screen or typing on keyboard. RSide modules model the actions of controller and his/hers reactions to messages from radio or information from radar.

5.1.3 Communication

The air traffic controllers communicate with each another using a sector radio. The sector radio is half duplex – every party hears everything, but only single participant can broadcast at any time. The radio communication implementation includes waiting for free communication frequency, simulation of unreliable connection and repeated attempts to transmit the message.

5.2 Visualization

AgentFly system contains modules for advanced 2D and 3D real-time visualization that allows user to monitor and interact with the process of simulation. The visualization is implemented using `OperatorAgent`s. Every one of those agents represents one visualization window and can run on different machines in distributed environment. The information shown in `OperatorAgent` are shown using layers. Every layer is handled by `LayerProvider` that creates the representations of objects that will be rendered on the layer.

Two types of windows are usually used in the system, one of them shows 3D model of the Earth and all simulated aircraft flying above the surface at given moment, this serves as a view of real-world situation. Second type of visualization is 2D representation of radar screen of a particular controlled sector. The radar screen shows the sector boundary, aircraft flying in and around the sector and plenty additional information like log of radar communication, graphs of controller workload etc. There are usually multiple radar windows shown for multiple sectors that are being controlled by the simulated ATC.

Chapter 6

Implementation

This chapter describes the implementation of the air traffic controller module. The module provides the means for controlling aircraft in the approach phase of the flight. The controller schedules the incoming flights to available runways in an effort to land them quickly and safely. Several different scheduling algorithms are implemented each offering different approach for arranging the flights for landing. Also the current methods of simulating aircraft in the AgentFly system are extended to allow for simulation of the approach phase of the flight including two methods of delaying the flights before they can commence the final approach to runway. The first one is holding pattern that is used when the flight needs to be delayed by a greater amount of time. The second if horizontal diversion manoeuvre which delays the aircraft by a lesser amount of time by diverting the airplane from its flight plan and therefore prolonging its route.

6.1 Data Processing and Visualization

Before the implementation of the controller module itself additional supporting modules were implemented. Their responsibility is to load and provide external data for the simulation and visualize the results of scheduling and other data in relation to simulation of approach phase of the flight.

6.1.1 Runway Plan Visualization

The visualization of runway plans is implemented in `RunwayPlanVisualModule` and its inner class `GraphicRunwayPlan`. `VisualModule`s are used in AgentFly to display additional information on controllers radar screen. When the runway plans are updated by the simulated ATC controller, `RunwayPlanVisualModule` converts the updated plans to their graphical representation described by instances of `GraphicRunwayPlan` and draws them on the radar screen.

The implementation of visualization is general and can display plans for any number of runways created by any number of planning algorithms. There is a scrollbar at the bottom of the visualization window that allows plans that are longer than viewport to be shown. The visual output is described in 4.3.

6.1.2 Approach Routes

Approach routes (STARs) are provided in proprietary formatted CSV file as described in 3.4. The routes are defined by short segments and links connecting those segments. `ApproachRoutesCsvScenarioPlayer` reads the route definitions, creates their representation and publishes them using `SharedObjectHolder` so that they can be used through the whole simulation.

The routes are stored in a data structure `ApproachRoutesData` that allows the routes to be found according to their star fix or the runway they're leading to. The routes themselves are represented by `ApproachRoute`. Each route contains information about runway it leads to, individual waypoints on the route, defined holding patterns along the route and estimated route duration and accuracy interval of the estimation. Waypoints are represented by `ApproachRouteWaypoint` defining the waypoint's name, position and altitude and speed interval passing airplanes must fit in.

Rendering of the approach routes in real-world 3D visualization is provided by `ApproachRoutesLayerProvider`. It can show either only the routes, or routes with waypoint names or complete route information with the altitude and speed intervals. The output of the visualization is shown in Figure 3.4.

6.1.3 Wake Turbulence Separation Intervals

Wake turbulence separation intervals (shown in Table 2.1) are provided in a text file. This data file is processed by `WakeTurbulenceDataScenarioPlayer` and shared with the simulation environment through `SharedObjectHolder`. The data are stored in `WakeTurbulenceData` which provides methods that return the separation given the wight classes of the preceding and following airplanes.

6.1.4 Airplane Trails

`AirplaneTrailLayerProvider` serves as additional visual aid that allows the user to easily analyze the course of the aircraft's flight. It subscribes to aircrafts' position updates and gradually draws a line through all of them. After the airplane lands, entire trail of the flight is shown in the real-world simulation window. The number of trails shown on screen after their airplanes landed can be set through configuration file. The rendered airplane trails are shown in Figure 6.3 and Figure 6.4.

6.2 Scheduling

The TMA air traffic controller's behavior is implemented in module `EomAtaStarApplication`. The module keeps an internal plan of which airplane is scheduled to land on which runway at which time. When a new airplane appears on the sector radar screen, the landing is scheduled and the pilot is commanded to use selected STAR for approach. Then the controller monitors the progress of the flight and either updates the plan to correspond with the reality or orders the pilot to perform such manoeuvres to arrive to the runway at prescribed time.

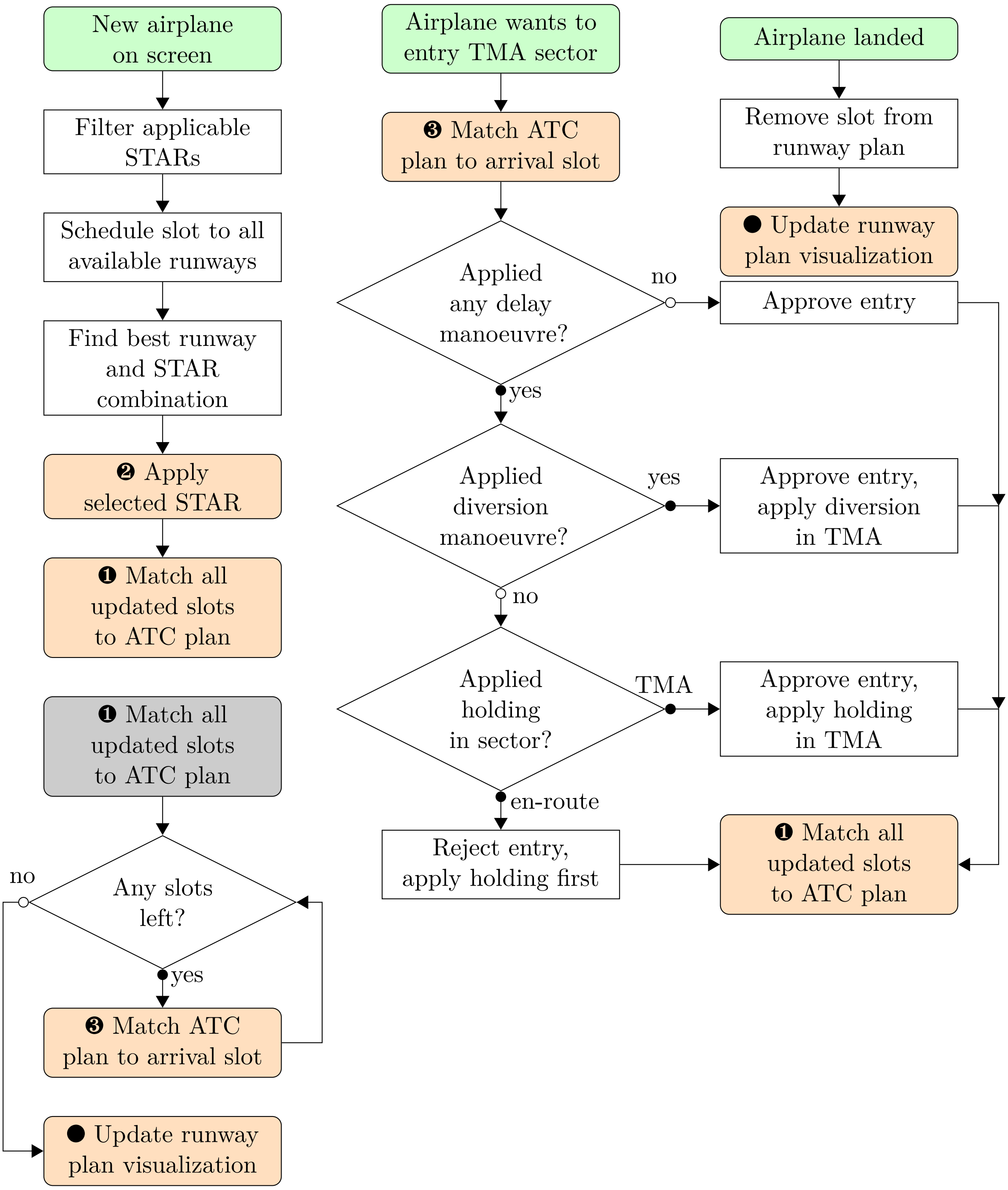

Figure 6.1: Flowchart showing simplified processes of ATC in TMA sector as implemented by `EomAtaStarApplication`.
Algorithm entry points responding to events from the simulation framework are shown in green. Internal subroutines' entry points are shown in gray and the subroutine calls are orange. The flowchart continues in Figure 6.2. The subroutine numbers don't represent order, they are only visual aid to match routine definition and call.

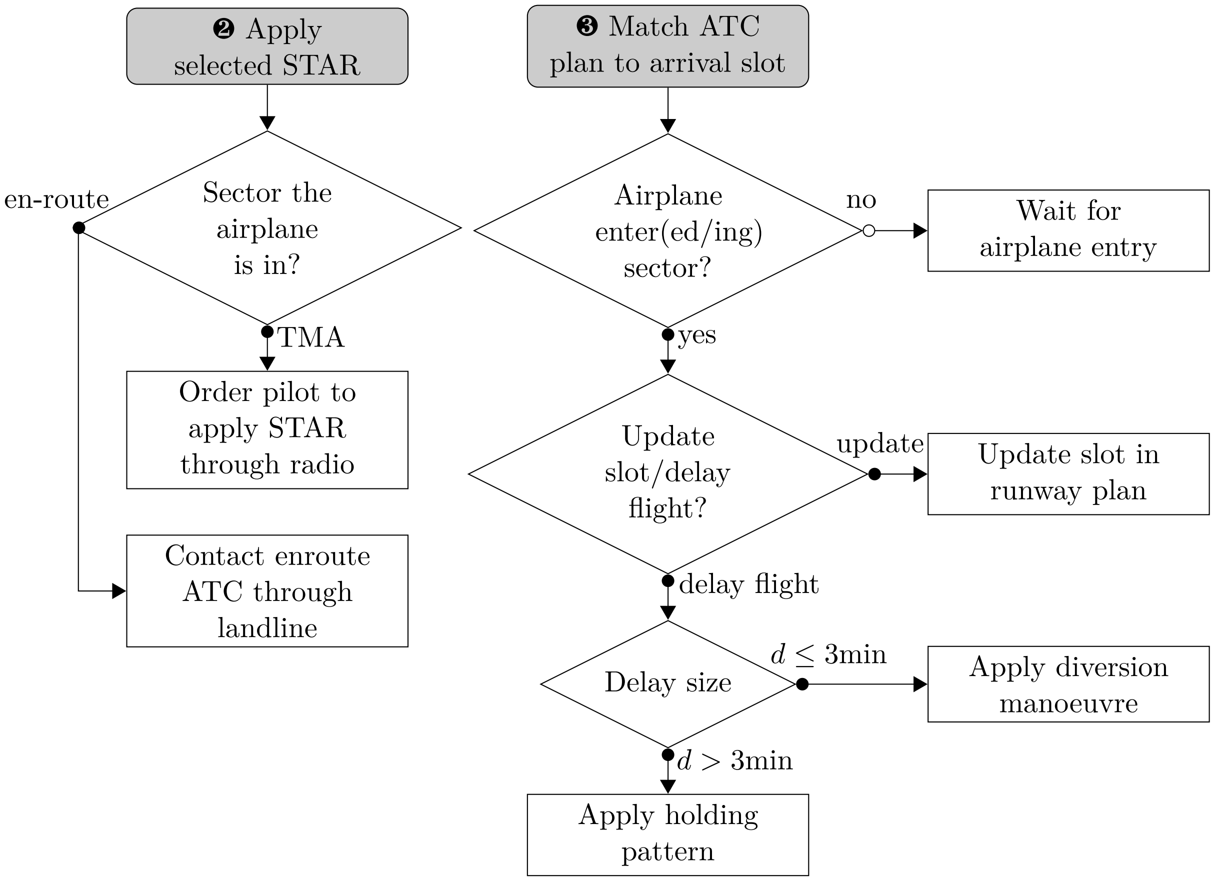

Figure 6.2: Continuation of the flowchart showing simplified processes of ATC in TMA sector from Figure 6.1.
Internal subroutines' entry points are shown in gray.

The following section describes the process of approach scheduling. See the corresponding flowchart in Figure 6.1.

When a new airplane appears on screen `EomAtaStarApplication` must select to which runway through which STAR the airplane should be heading and create a new slot in the runway plan. First, all routes that the airplane can take to the airport are filtered.

6.2.1 STAR Selection

There are several possible scenarios for the scheduling and STAR selection. If only single STAR is available, the situation is simple. Controller schedules the airplane to land on the runway the STAR leads to using one of the algorithms described in 4.4 and orders the pilot to use the STAR for approach.

If there are several STARS leading to different runways, controller must select one first. This is done using one of the multi-runway scheduling algorithms described in 4.5.

There are also cases where multiple routes lead to the same runway and the air traffic controller must decide which route to use. For example see Figure 3.4: airplane arriving from northeast through fix `FLCON` can land on runway `9R` flying along either route north of the airport or south of the airport.

The decision is made as follows: when there are multiple routes to one runway arriving flight can take, the one with shortest estimated duration is used for planning. Then scheduling on multiple runways is performed as was described in 4.5. If the runway with multiple routes leading to it is selected to be used, deliberation on which of the routes to take is carried out. If the slot assigned to the arriving flight has a delay from the ETA of the shortest STAR (because the runway is busy with other flights at the moment) and the delayed ETA is bigger than that of another route with longer duration, this route is used instead of the shortest.

This basically means that if the runway is free, the airplane takes the shortest route, but if the flight needs to be delayed, the controller prefers that the airplane takes a longer route to accommodate the delay rather than hold in one place and taking the short route. The airplane needs to be delayed anyway and this way it at least doesn't occupy the holding pattern and instead flies along a different route.

The plan of approaches of each runway is represented by an instance of `RunwayPlan`. It contains all scheduled slots on the runway and provides methods for new slot allocation based on the algorithms described in 4.4 as well as methods that provide information for comparison of the plans for multi-runway scheduling. These return plan makespan, maximal delay among the planned slots, total delay of all planned slots etc. Slots are instances of `RunwaySlot` and hold flight ID, airplane weight class, earliest ETA, target ETA, latest ETA and slot delay.

6.3 STAR Application

When the approach is planned, the controller must advise the pilot which route he/she has to use for landing. First the controller determines in which sector the airplane is currently located (see Figure 6.2). Typically this is one of the neighboring en-route sectors, but the airplane might also already be in the TMA sector. If the airplane is in en-route sector, the TMA controller sends event to en-route controller to contact the pilot through radio. If the airplane is in the TMA sector, the controller contacts the pilot directly.

The radio communication is provided by `EomRSideStarApplication` module. The module arranges for the whole process including for the communication itself, delay of the communication, repeated transmissions if the previous tries failed and displaying the message in radio communication window on the radar screen. The message transmitted through radio is represented by `StarApplicationData`.

6.3.1 Flight Plan Update

On the pilot's side, the radio message is received by module `EomPilotStarApplication` which applies the STAR to its current flight plan that is represented by `GpsFlightPlanWrapperPolyline`. The flight plan wrapper was extended so it can apply the whole STAR with single replanning. First a common waypoint of current flight plan route and applied STAR is found and the STAR is appended to the flight plan after the common waypoint replacing the original route. Then a list of clearances is created, with required altitudes and speeds at fixes along the STAR. Finally, flight plan is replanned through the added STAR with restrictions defined by the clearances using BADA model of the airplane's behavior.

When the STAR is applied, `EomAtaStarApplication` gathers all slots that have been updated during the scheduling, including the newly added slot, and ensures that each slot matches the estimated arrival according to the current airplane ATC plan. The process is shown in Figure 6.1. The ATC plan represents the airplane not from the pilots view, but from the controllers perspective. While doing so, more slots can be updated and the process continues until all slots and ATC plans match. When the matching is done, visualization of runway plans is updated.

6.4 Sector Entry Check

Sector entry check is a process that takes place before the hand-off of an airplane to new sector is performed. At this time the airplane's ATC plan is available for the first time. For TMA sector the entry check procedure is shown in figure 6.1 and works as follows:

Since the ATC plan of airplane in entry check is now known, the controller can match the ATC plan and planned slot. While doing so, more slots can be updated and the process continues until all slots and ATC plans match. When the matching is done, the result of matching of the entering airplane is examined. If no delay manoeuvre was performed and only the slot was updated, the entry is approved and hand-off will take place. If horizontal diversion manoeuvre was applied, the entry is approved and TMA controller waits till the airplane crosses to TMA sector to perform the vectoring. If holding was applied before the airplane crosses border to TMA sector, the entry is rejected and another entry check will take place once the airplane completes its holding in en-route sector and will be about to cross to TMA once again. If the holding pattern was applied after the airplane crosses the border of TMA sector, entry is approved and airplane holds inside TMA.

6.5 Flow Control

The duty of the flow control is to make sure, that the airplanes will reach the runway during its assigned time slot. To do that, `EomAtaStarApplication` must either update the slot to match the ATC plan or delay the airplane so the ATC plan will match the slot. The process is shown in Figure 6.2.

First, the controller checks whether the entry check took place and ATC plan is already available and if not, airplane wait for its entry.

If the ATC plan is available, ATC plan arrival interval is determined and compared with the runway plan slot. If the ATC plan interval is smaller than the slot and fits inside it, the slot is shrunk not to occupy unnecessary time on the runway.

If the ATC plan arrival interval is smaller than the runway slot and ends later (the estimated arrival was too optimistic), the slot is shifted later in time and shrunk to fit the arrival interval. This can lead to shifting of slots planned after the current one and these slots will have to be matched to their ATC plans after the matching of current slot is finished. The following slots can be only shifted further in time, the order of the slots is preserved.

If the ATC plan arrival interval is bigger than runway slot and the slot ends sooner than the arrival interval, the slot is enlarged and shifted so it fits the arrival interval. Again, this may lead to the need to shift following slots in the runway plan.

Finally if the ATC plan arrival interval starts sooner than the runway slot (the estimated arrival was pessimistic) no matter if the interval is bigger and smaller than the slot, the controller first tries to shift the slot back in time so it starts at the same time as the arrival interval. This is possible only if there is empty space in the runway plan before the slot. If there is no empty space or the slot can't be moved backwards all the way, the airplane must be delayed so the ATC plan arrival interval would fit in the slot. Once the airplane is delayed, the runway slot is shrunk or enlarged to fit the updated ATC plan arrival interval.

If the airplane needs to be delayed by more than three minutes, holding pattern is used, if the delay is smaller or equal than three minutes, horizontal diversion manoeuvre is used. The tree minute threshold is arbitrary and based on experience, and can be changed to different value.

6.5.1 Vectoring

Horizontal diversion manoeuvre is performed by `EomAtaVectoring` module using airplane vectoring. Unlike STAR or holding application, in which the controller tells the pilot what manoeuvre to use immediately (*"Hold 5 minutes at fix SINCA"*) and the pilot incorporates them into his/hers flight plan and performs them at the right time and place, vectoring is performed by the pilot at the exact time he/she received the order through radio (*"Turn to heading 270 immediately"*).

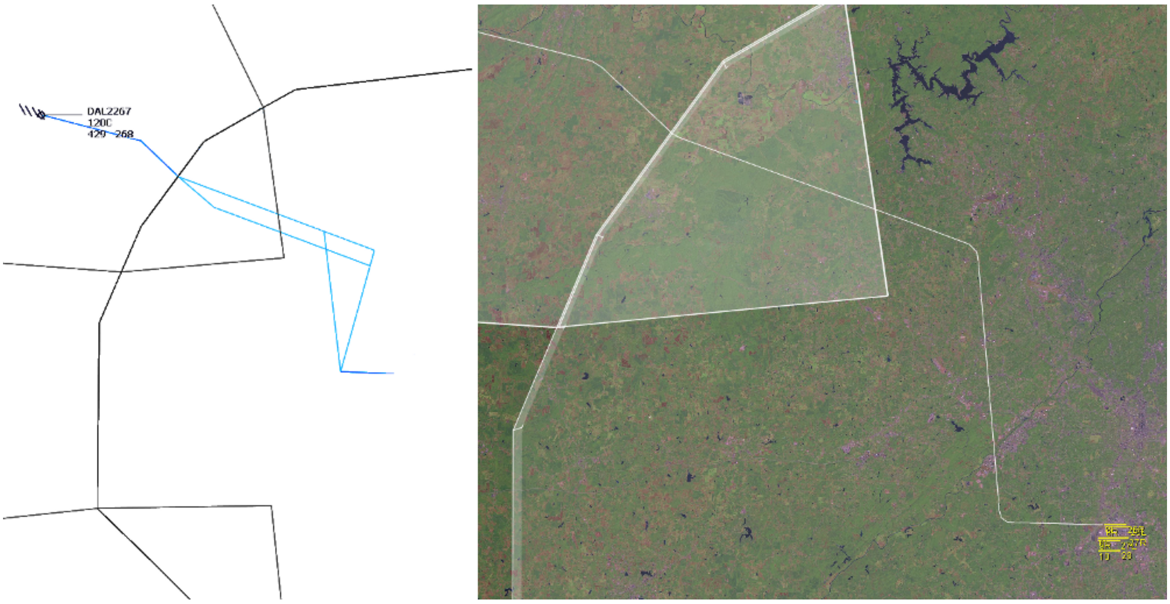

Figure 6.3: Visual representation of horizontal diversion manoeuvre. Left: ATC plan with planned diversion. Right: trail of airplane that performed the diversion manoeuvre and landed.

This means that the `EomAtaVectoring` module only computes the geometry of the manoeuvre (when and how much the airplane must divert from its flight plan to achieve required delay) and applies it to the ATC plan. The vectoring itself (radio communication with the pilot) takes place later at the time the airplane must divert. This is already implemented in AgentFly and used for example in collision avoidance.

You can see an example of planned and executed horizontal diversion manoeuvre in Figure 6.3.

6.5.2 Holding Pattern

If the holding pattern needs to be applied to delay the airplane's arrival, the `EomAtaStarApplication` first needs to decide which holding pattern to use. If there are any published holding patterns on the remaining part of STAR ahead of the airplane, the one defined for earliest fix will be used to delay the airplane early on and not at the latest possible moment just before landing.

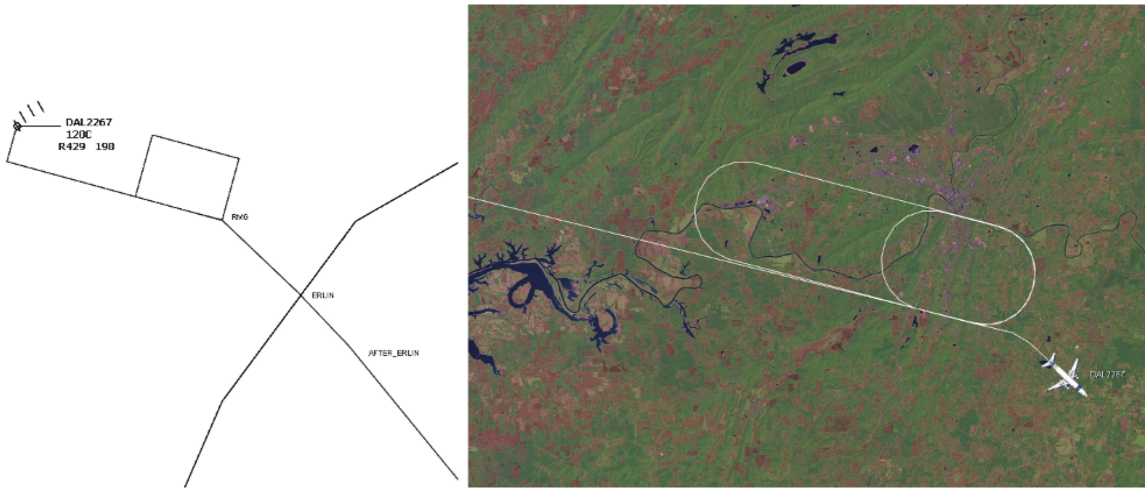

Figure 6.4: Visual representation of holding pattern. Left: ATC plan with airplane completing first round of the holding. Right: trail of airplane that performed two rounds of holding (one full and one shortened).

If there is no published holding pattern, the controller creates one ad hoc. He/she tries to create it as early as possible. If the holding pattern would cross the sector border and the airplane would therefore cross it back and forth multiple times, the controller first tries to decrease the holding pattern size (and make the airplane fly through it more times) and if that doesn't help, he/she then creates the holding further down the route inside the TMA sector.

The application of the manoeuvre itself is done by `EomAtaHoldingApplication` module. It calculates how much time the airplane needs to spend in the holding pattern to achieve desired arrival time at the runway and incorporates the holding pattern into ATC plan.

The communication with the pilot is ensured by `EomRSideHoldingApplication` and in much the same way as in STAR application, the whole process of the communication is arranged including delay of the communication, repeated transmissions if the previous tries failed and displaying the message in radio communication window on the radar screen. The message transmitted through radio is represented by `HoldingPatternData` and includes the holding fix information, holding altitude, pattern heading, holding direction, speed and time, when the holding is completed and airplane should continue with the approach.

On the pilot's side, the radio message is received by `EomPilotHoldingPatternApplication` module that applies the holding to its current flight plan, represented by `GpsFlightPlanWrapperPolyline`. New waypoints defining the shape of the the holding pattern are added as well as clearances ensuring the holding will be carried out at the right altitude and speed. Finally, flight plan is replanned using BADA model of the airplane's behavior.

You can see an example of planned and executed holding pattern in Figure 6.4.

6.6 Airplane Landing

The last task performed by `EomAtaStarApplication` module is checking if the airplane landed at the correct time (shown in Figure 6.1). When event marking that an airplane landed, the controller checks if the touchdown time fits inside. If it doesn't, there has been probably some error during one of the manoeuvres performed by the pilot before arrival. The airplane arriving outside its slot is logged to be further investigated.

When the check is completed, the controller removes the slot from the corresponding runway plan to minimize the plan's size leading to faster allocation of future slots. Also the runway plans visualization is updated. Visualization of a simulation scenario progress is shown in Figure 6.5. There is runway plan window shown in top-left part. Under it is ATC plan with planned holding pattern and diversion manoeuvres. The right part contains real world visualization with trails of completed approaches.

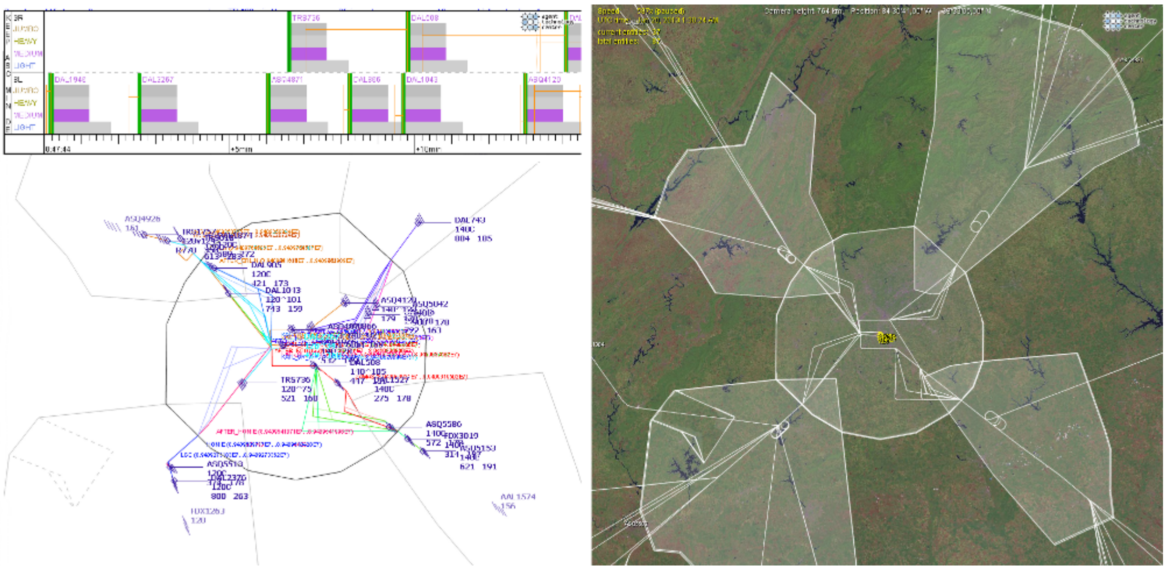

Figure 6.5: Snapshot of visualization output during simulation with real-world flights.

Chapter 7

Experiments

This chapter is a summary of conducted experiments. Four experiments that well represent the behavior of individual algorithms were selected and are presented here. The first one studies the performance of slot allocation algorithms in scenario with alternating weight classes of the arriving airplanes. The second one studies the performance of slot allocation algorithms in real-world air traffic. The third one analyses combinations of slot allocation and runway selection algorithms in scenario with real-world air traffic. The last one studies the influence of use of miles in trails (MIT) restrictions in neighboring en-route sectors on airplane delays in TMA sector.

7.1 Comparison of Scheduling Algorithms on Single Runway

This experiment compares the performance of implemented scheduling algorithms. The scenario contains twelve flights with alternating weight class landing on one runway arriving in close succession. This scenario should be particularly demanding to be scheduled optimally because the alternation in weight classes make it sensitive to the order in which the slots are scheduled.

Flight id	Weight class	Appearance on screen	Estimated arrival time
01M	MEDIUM	17:36	42:30
02J	JUMBO	19:24	42:29
03J	JUMBO	19:00	44:15
04M	MEDIUM	22:00	44:49
05M	MEDIUM	21:36	46:30
06J	JUMBO	23:24	46:29
07J	JUMBO	23:00	48:15
08M	MEDIUM	26:00	48:49
09M	MEDIUM	25:36	50:30
10J	JUMBO	27:24	50:29
11J	JUMBO	27:00	52:15
12M	MEDIUM	30:00	52:49

Table 7.1: Configuration of the test scenario with alternating weight classes on single runway

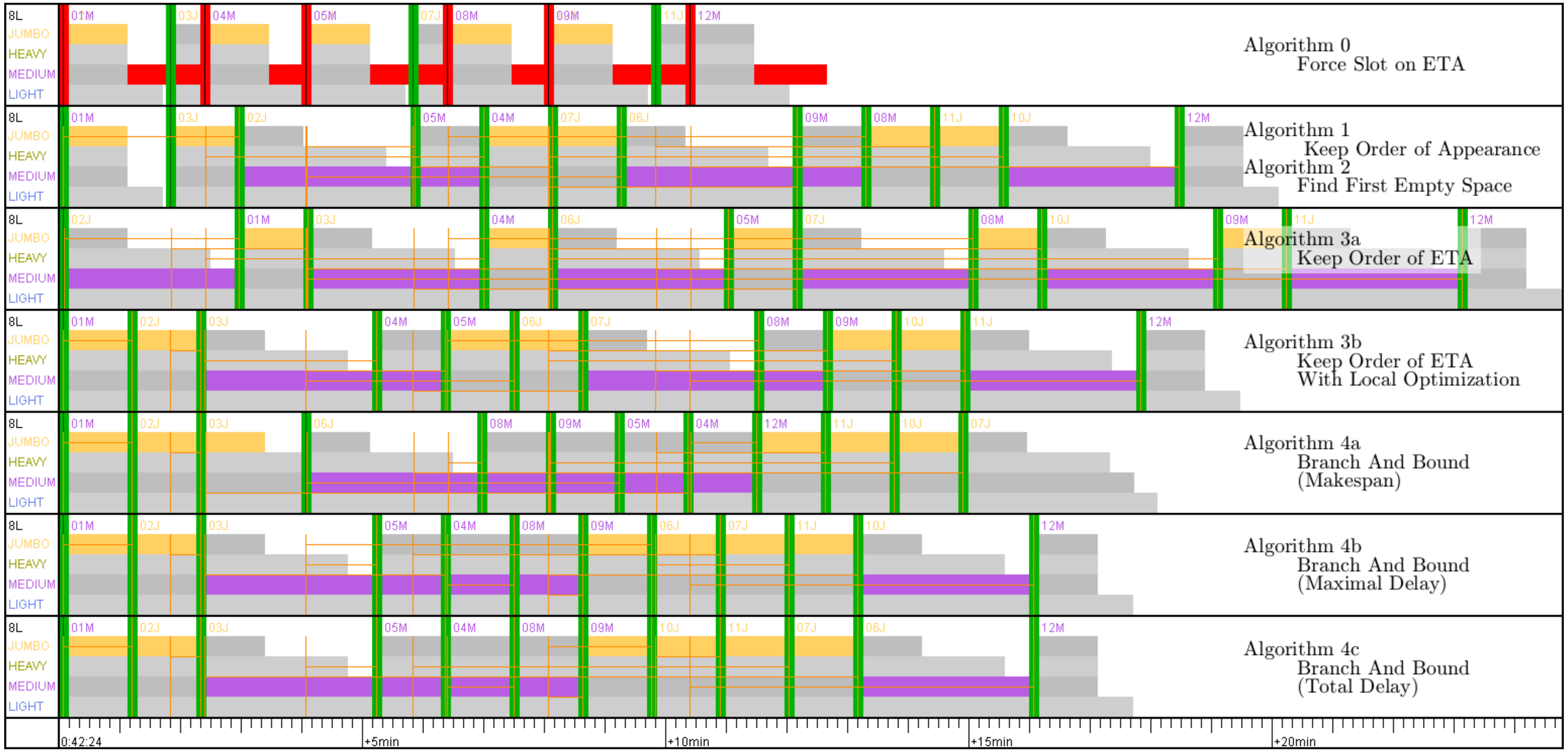

Figure 7.1: Result of planning with different algorithms on single runway

Table 7.1 shows the configuration of the test scenario. For the `JUMBO` category the Airbus A380-800 was used and the `MEDIUM` category was represented by Boeing 737. Airplanes headed to the airport from two separate flows to create dense traffic at the runway.

The resulting plans are shown in Figure 7.1. The first row shows the result of *Algorithm 0* which places the slots in the exact time the arrival was estimated. It demonstrates how many collision would occur if no planning was present and serves as the lower bound on the total plan time.

The plans generated by *Algorithm 1* and *Algorithm 2* are identical in this case and are depicted in the second row. The slots are ordered in the same succession the corresponding aircraft appeared on the controller's screen. The reason the plans are identical is that the stream of arriving airplanes is very dense and there are no voids present between the aircraft that the *Algorithm 2* could utilize to produce better schedule than *Algorithm 1*.

Third and fourth rows show the results of both versions of *Algorithm 3*. The third row contains the plan generated by the simple version of the algorithm, that keeps the order of aircraft's expected time of arrival. In this instance the algorithm produced the poorest result with the longest overall time as well as longest maximal and total delays. This is due to the specific configuration of the test scenario that alternates between weight classes. The second version doesn't rigorously keep the order of estimated arrival but tries to locally minimize the delays and therefore produces much better result.

The last three rows contain plans generated by the three versions of *Algorithm 4*. Each one shows optimal plan according to selected criterion. The one in the fifth row has the shortest total makespan. The plan in sixth row has the shortest maximal delay among all slots in the plan. And the last plan has minimal sum of delays of all slots. Note, that the last two plans are very similar and even have the same total length but the order of the slots determines whether the solution minimizes one or the other criterion.

Let's look at the plans produced by individual algorithms from the point of view of different evaluation criteria introduced in 4.2.2.

7.1.1 Makespan

The first criterion by which the quality of a plan can be determined is the total duration of the schedule, also called makespan. The shorter the plan is, the better. In reality this criterion is problematic, because the flow of airplanes to the airport is infinite, only the density changes in time. But even so, this criterion can give a notion of the quality of immediate plan.

Makespan times in the progress of planning are shown in Chart 7.2. The result given by *Algorithm 0* is a lower bound that is given by the configuration of the testing scenario, no plans can be shorter than this value. The longest plans are generated by *Algorithm 3a*. Optimal plan with minimal makespan is generated by *Algorithm 4a* because this algorithm optimizes the makespan, with *Algorithm 4c* having very similar results (the values for 12 planned slots differ by less than one second) and *Algorithm 4b* also being close. The results given by *Algorithm 3b* are still near optimum with plans 22% longer at most. *Algorithm 1* and *Algorithm 2* produce identical, fairly good plans.

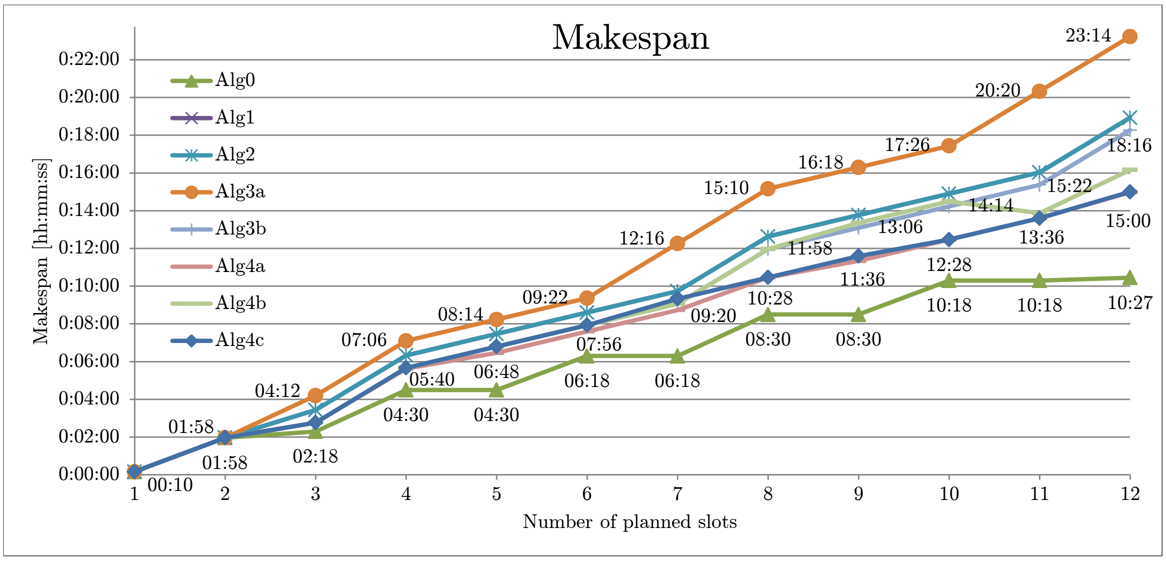

Figure 7.2: Chart of makespan for plans created with different algorithms on single runway

7.1.2 Maximal Delay

Another criterion that can describe the quality of a runway plan is the maximal delay among all slots. The smaller the delay is, the better the plan is. This criterion can be used to prevent the situation in which one airplane would give the priority to all others and would wait until its supply of fuel is depleted. The sum of all delays can be small, but the fact that the critical delay time for one aircraft was exceeded renders such plan potentially dangerous.

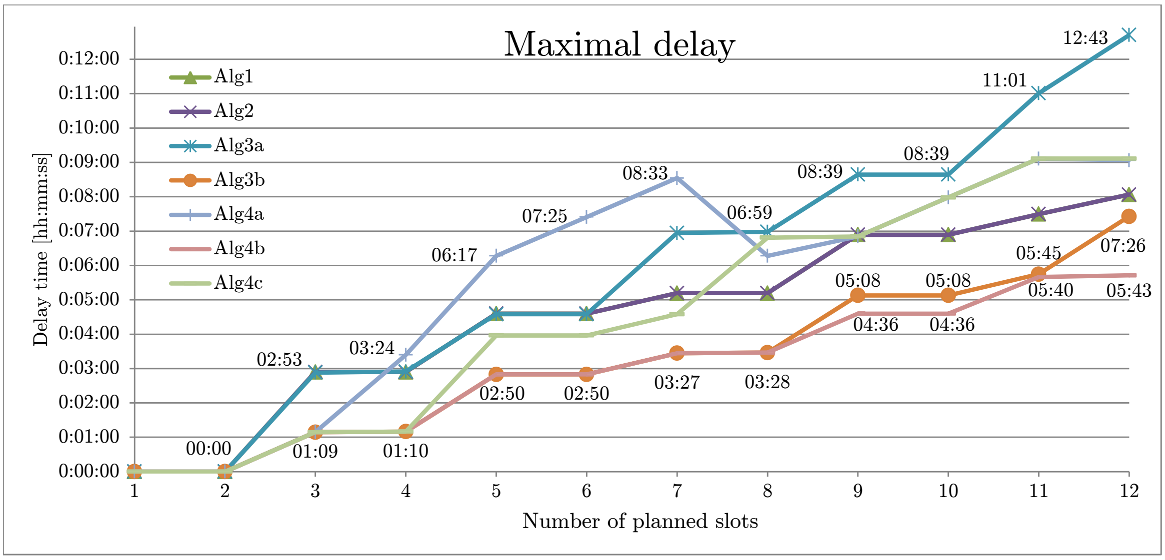

Figure 7.3: Chart of maximal delay for plans created with different algorithms on single runway

Chart 7.3 shows the development of values of this criterion during the planning. The optimal value for the final plan is 5 minutes and 43 seconds and is achieved by *Algorithm 4b* because this algorithm optimizes the maximal delay. *Algorithm 3a* has the poorest performance with the value of 12:43. *Algorithm 4a* optimizes the total makespan and in order to do that, it can delay some slots by a significant amount of time. This can lead to high maximal delay values as shown here for plans with 4 – 7 slots. *Algorithm 3b* performs very well with its result at or very near the optimal value. For the final plan, the maximal delay for this algorithm is 1 minute 43 second longer than the optimal value.

7.1.3 Total Delay

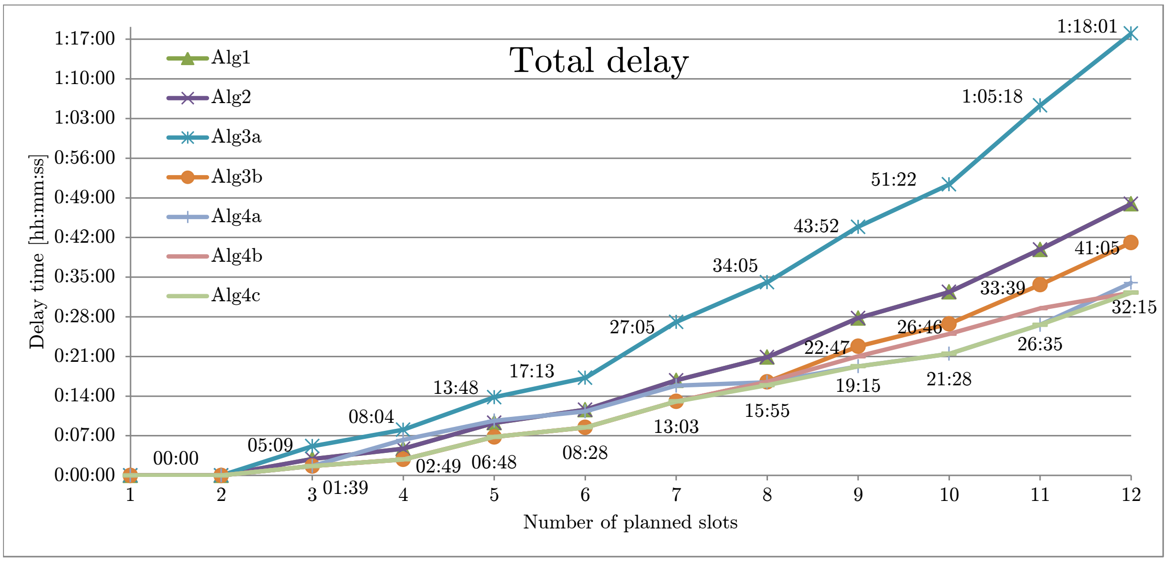

Figure 7.4: Chart of total delay for plans created with different algorithms on single runway

The sum of all slot delays is another criterion that measures runway plan's quality. It describes the total time spent waiting in the terminal area. More importantly is also linked to the total amount of fuel burnt during the waiting. Both time and used amount of fuel affect the costs of the flight.

The total delay times in the progress of planning are shown in Chart 7.4. The poorest results are again given by *Algorithm 3a* with the total delay more than 2.4 times the optimum. The optimal result is produced by *Algorithm 4c* (32:25 for the final plan) with other two versions of the *Algorithm 4* very near. *Algorithm 3b* performs well with the total delay at 41:05 which is less than 30% more than the optimum. *Algorithm 1* and *Algorithm 2* still perform good and could produce usable results but are not as good as *Algorithm 3b*.

7.1.4 Number of Replanned Slots

The number of replanned slots shown in Chart 7.5 expresses how often the controller must interfere with the schedule of previously planned aircraft after a new one has been added to the plan. This criterion has a relation to the controllers workload, because he/she must contact each replanned aircraft and give the pilot updated instructions.

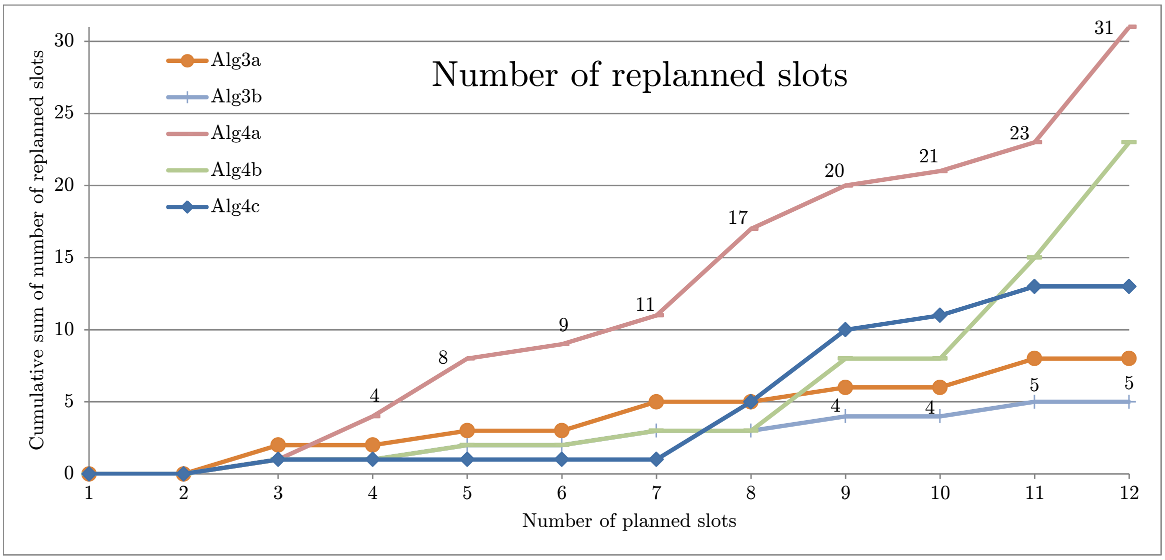

Figure 7.5: Chart of cumulative sum of replanned slots for plans created with different algorithms on single runway

Algorithm 0, *Algorithm 1* and *Algorithm 2* place the new slots in a way that doesn't affect the previously planned and therefore don't cause the need to replan old slots. *Algorithm 3b* with five replanned slots during the creation of final plan produces very good results according to the aforementioned criteria but is still undemanding of frequent replanning of old slots. All three versions of *Algorithm 4* have high numbers of replanned slots because the fact that they look for optimal solution causes them to change the plan significantly with each slot addition. This is especially true for *Algorithm 4a* which minimizes the total makespan. Its 31 replanned slots during the creation of plan for 12 airplanes render it practically unusable for real world application.

7.1.5 Conclusion

Algorithm 1 and *Algorithm 2* behave identically in this scenario providing usable results in every measured aspect with the main advantage being that previously planned slots are fixed and there is no need to update the pilots instruction when new airplane flies in. *Algorithm 3a* performs poorly and doesn't handle the alternating weight classes well. On the other hand, similar *Algorithm 3b* that adds the local optimization of delay, gives good results not too far from optimal solution with a low number of slots being replanned. *Algorithm 4a–c* produce the optimal results in the categories they are optimizing for but can have inferior results in other categories (e.g. versions a and c for maximal delay). The fact that the algorithms strive for the optimum result also means that these algorithms tend to replan previous slots often causing high workload for the controller. This and their computational complexity makes their usage in real-world application impractical.

7.2 Real World Flights on Single Runway

The next experiment compared implemented scheduling algorithms on real-world air traffic. The test scenario contained 764 flights going through ZTL Center on 20th June 2013 between 12:00 a.m. and 6:00 p.m. All flights were arriving to runway `8L` of Atlanta airport.

The flights were scheduled for approach using four algorithms. *Algorithm 0* was not used because it produces invalid plan with possible collisions. All three variants of *Algorithm 4* were also not used because their computational complexity makes them unsuitable for planning in real-world scenario with hundreds of flights.

Scheduling algorithm	Makespan	Maximal delay	Total delay	Updated slots
Algorithm 1	22:22:25	4:25:13	53d 16:08:22	0
Algorithm 2	22:20:12	4:23:00	52d 13:56:08	0
Algorithm 3a	22:18:54	4:15:23	51d 21:34:16	2038
Algorithm 3b	22:19:18	4:15:47	51d 21:42:57	2025

Table 7.2: Results of different scheduling algorithms for dense real-world traffic on single runway

The results of the experiment are shown in Table 7.2. Because only the final plan is studied in this case, only the results for entire plan with all slots are shown, not their progress during the planning. It can be clearly seen from the table that directing all traffic comming to Atlanta airport will result in extremely dense traffic and therefore great delays.

The lower bound on makespan given by *Algorithm 0* defined by the configuration of the test scenario is 18:05:23, no plan can be shorter than that. The actual plans were about 4 hours longer. The differences between the algorithms are small with the longest makespan no more than 4 minutes longer than the shortest.

Differences between maximal delays of the schedules are slightly bigger with the interval between smallest and biggest value spanning 10 minutes. The values for both variants of *Algorithm 3* are virtually identical. In real-world scenario, delay of more than four hours would certainly cause the airplane to deplete its fuel reserve and crash since the commercial flights are in usual conditions required to have 30 minutes reserve of fuel for holding at the destination airport.[8, Chapter 4]

The total delay results show how even small differences of the algorithms performance can have significant impact when large number of airplanes is affected. *Algorithm 3a* generated schedule in which the total waiting time is 1 day 18 hours and 34 minutes smaller than in schedule made by *Algorithm 1*. More than 42 hours worth of fuel and passenger time is very significant saving. The results of both variants of *Algorithm 3* are again virtually same, the total delays differ only by 8 minutes and 41 second (less than 0.012%).

Algorithm 1 and *2* don't update slots once they're planned. The second version outperforms the first variant of *Algorithm 3* by 13 updated slots during the planning. The overall number of updates is fairly hight but with average 2.7 updates per airplane not unrealistically demanding for the air traffic controller. Also with lower density of the traffic, this number would be lower.

In conclusion, this experiment shows that all four algorithms produce similar and usable results in scenarios with high density real-world traffic. The best results were planned by

Algorithm 3a. On the other hand, results by *Algorithm 3b* were nearly identical and this algorithm is more robust in situations with alternating weight classes of the arriving airplanes as shown in previous experiment. Now let's look how will the results look for multi-runway planning with real-world data.

7.3 Real World Flights on Multiple Runways

This experiment is aimed to compare the quality of different combinations of slot allocation and runway selection algorithms. The scenario configuration contained 1308 flights of 24 hour traffic as described in 3.3 and the flights were scheduled to land on one of the runways `8L` and `9R` on Atlanta airport. In total, 20 combinations of scheduling algorithms were evaluated.

	Alg1	Alg2	Alg3a	Alg3b	Criterion
Less slots	24:24:49	24:24:49	24:24:49	24:24:49	Makespan
	109:47:19	58:27:18	37:39:31	37:37:12	Total delay
	0:19:36	0:16:30	0:11:03	0:11:03	Maximal delay
	0	0	2012	2008	Updated slots
	Alg1	Alg2	Alg3a	Alg3b	Criterion
Makespan	24:22:38	24:21:29	24:21:12	24:21:12	Makespan
	77:16:46	44:40:44	27:55:17	27:45:45	Total delay
	0:13:33	0:11:48	0:07:07	0:07:07	Maximal delay
	0	0	1405	1372	Updated slots
	Alg1	Alg2	Alg3a	Alg3b	Criterion
Maximal delay	24:17:30	24:25:08	24:18:10	24:18:04	Makespan
	80:14:43	57:28:01	53:41:49	54:18:14	Total delay
	0:09:25	0:08:24	0:06:57	0:06:57	Maximal delay
	0	0	2090	2080	Updated slots
	Alg1	Alg2	Alg3a	Alg3b	Criterion
Total delay	24:25:08	24:25:08	24:25:08	24:25:08	Makespan
	89:32:04	39:07:20	20:22:59	19:33:21	Total delay
	0:14:38	0:11:16	0:06:08	0:06:14	Maximal delay
	0	0	1746	1715	Updated slots
	Alg1	Alg2	Alg3a	Alg3b	Criterion
New slot SETA	24:22:38	24:18:27	24:18:10	24:18:10	Makespan
	77:17:41	40:09:58	19:47:41	19:39:17	Total delay
	0:13:33	0:11:32	0:05:36	0:05:36	Maximal delay
	0	0	1963	1973	Updated slots

Table 7.3: Result of planning with different combinations of algorithms on multiple runways

The results of the experiment are shown in Figure 7.6 and Table 7.3. Figure 7.6 shows that all slot allocation algorithms provide plans with similar makespans, but differ in the maximal and total delays. *Algorithms 3a,b – Keep Order of ETA (With Local Optimization)* provide results superior to *Algorithm 1 – Keep Order of Appearance* and *Algorithm 2 – Find*

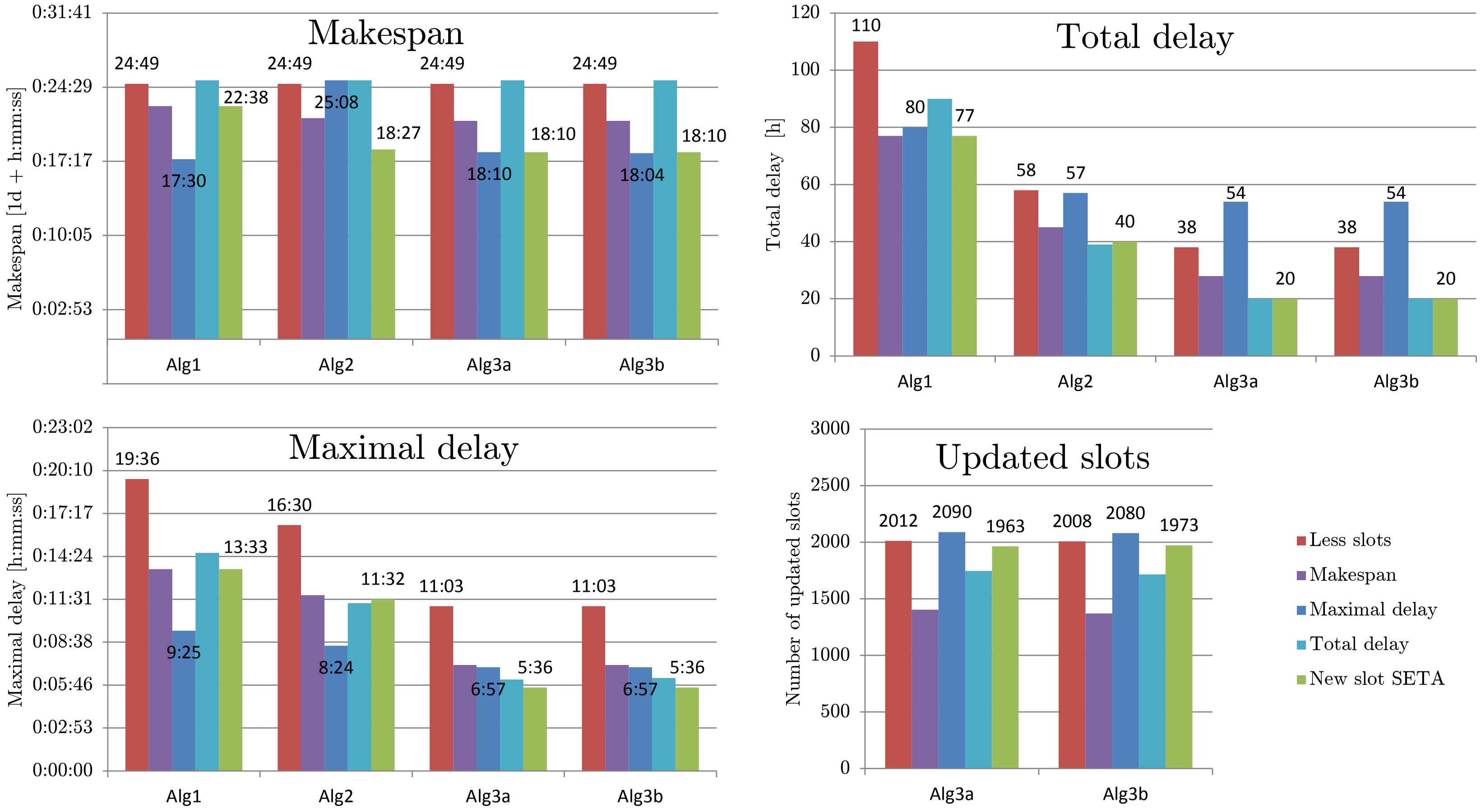

Figure 7.6: Result of planning with different combinations of algorithms on multiple runways

First Empty Space according to maximal and total delays criteria but unlike *Algorithm 1* and *Algorithm 2* pay for their performance by updating previously planned slots.

Best performing runway selection algorithms are *Total delay* and *New slot SETA* with *Total delay* producing plans with longer makespan but less updated slots.

Table 7.3 contains all results from all algorithm combinations for all evaluation criteria. Best results in every category are highlighted in dark green, good results close to optimal value are green, average results are yellow and the poorest results are red. This allows easy analysis on overall performance of the algorithm combinations.

Algorithm 3b – Keep Order of ETA With Local Optimization for slot allocation and *New slot SETA* for runway selection were selected as the best combination for future real-world simulations. They produce plans with the smallest maximal delay and makespan and total delay very close to optimum.

Algorithm 3a – Keep Order of ETA performs also very well, but produces plans with slightly bigger total delay than *Algorithm 3b* and is also less robust to alternating weight classes of airplanes as shown in 7.1. Another good algorithm is *Total delay* for runway selection, which has with *Algorithm 3b* the smallest total delay but creates plans with longest makespan.

7.4 Real World Flights With Miles In Trails

Last experiment studies the impact of using miles in trails restrictions in the neighboring en-route sectors. The capability to use MIT was already implemented in AgentFly and was used in this test scenario with minimum separation set to 90 seconds and optimum separation 120 seconds. First 226 flights from the scenario with real-world flights described in 3.3 were used for this experiment. Both runways of Atlanta airport were active. *Algorithm 3b* and *New slot SETA* were used for scheduling approaching flights.

Figure 7.7 shows airplane trails of flights with active MIT restrictions, note that the horizontal diversion manoeuvres are applied not only in the TMA sector as a standard way to delay flights before approach, but also in neighboring en-route sectors to ensure regular flow of airplanes to the TMA sector.

The results of the experiment are shown in Figure 7.8. Use of MIT significantly decreases the maximal and total delay caused by scheduling in TMA sector. On the other hand, MIT produced plan with longer makespan and more updated slots (94 vs. 93 without MIT).

The experiment shows that use of MIT has significant impact on the way airplanes are scheduled for approach in the TMA sector but further experiments are needed to analyze the impact more deeply. For example in this experiment only delay caused by scheduling in TMA was measured. It may be interesting to find out the influence of MIT on the overall flight delay over its whole flight plan.

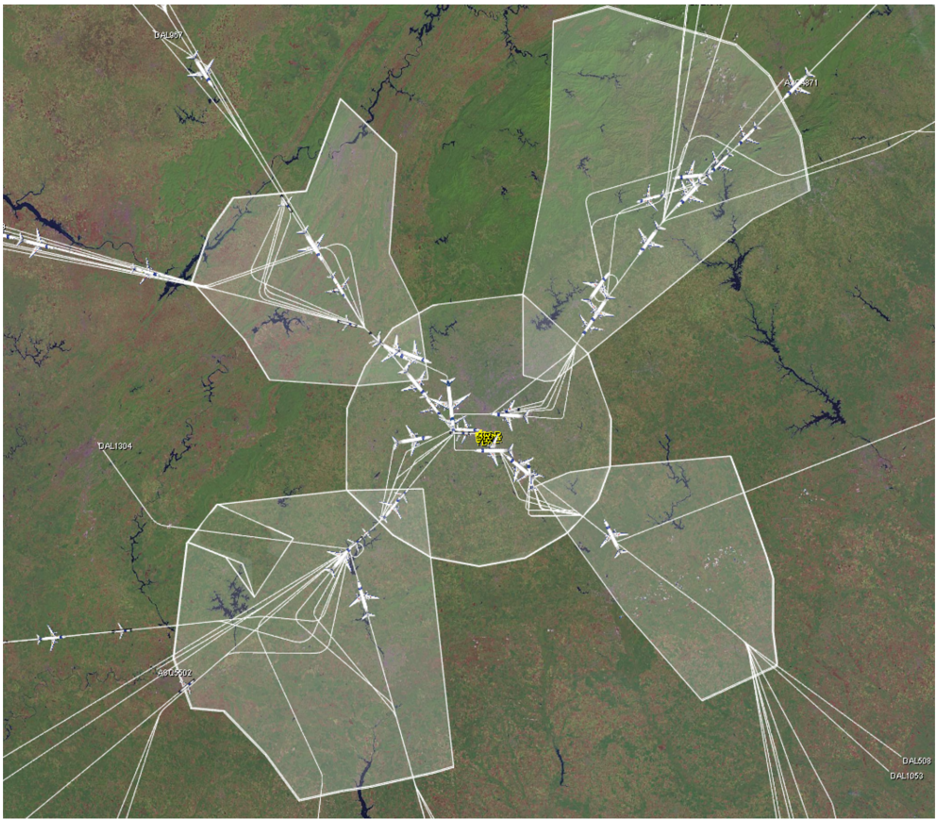

Figure 7.7: Visualization of MIT restriction application in en-route sectors.

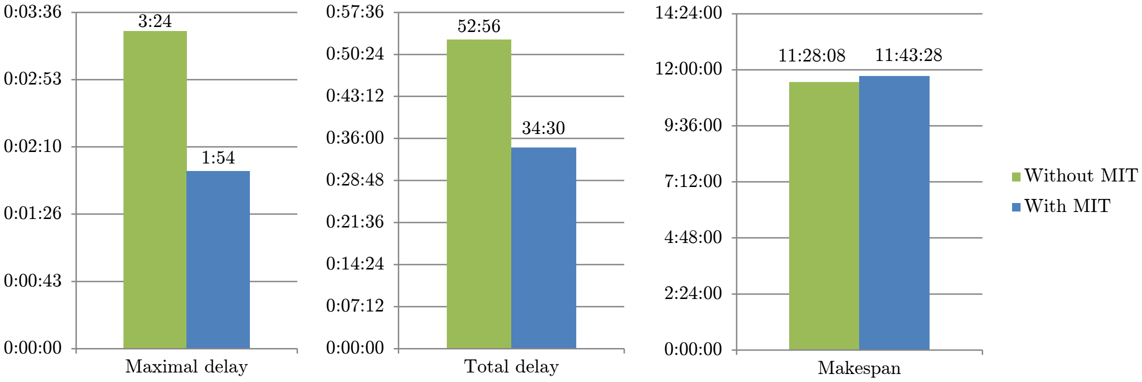

Figure 7.8: Comparison between planning with and without MIT.

Chapter 8

Conclusion

The overall goals of this thesis were satisfied, the module simulating air traffic controller behavior was designed and implemented and series of evaluate experiments with real-world scenarios was conducted and evaluated and best performing algorithm combination was selected based on these experiments. This work accomplishes the individual goals as follows:

1. The duties of air traffic controller in terminal area and procedures for controlled flight in the vicinity of airports are studied in Chapter 2.
2. Data needed for implementation and evaluation were analyzed and acquired or experimentally estimated. The data acquisition turned out to be more difficult and time consuming than expected, as many of the data sources were in proprietary format or had to be processed manually. The process of data acquisition is described Chapter 3
3. Variety of algorithms that simulate the controller's deliberation on scheduling arriving flights to land on controlled airport was designed. No existing algorithms that are in use on airports were available and therefore the algorithms have been designed from scratch based only on the knowledge of procedures of air traffic control in TMA sectors. Each algorithm offers different approach for arranging the flights for landing and has different advantages and disadvantages. The algorithms are described in Chapter 4.
4. The proposed scheduling algorithms were implemented as a controller module in AgentFly simulation system. Apart from implementation of the scheduling module, existing methods of simulating aircraft implemented in AgentFly needed to be extended to allow for simulation of the approach phase of the flight. The implementation is described in Chapter 6.
5. Series of experiments testing the different scheduling methods in both specifically designed and real-world scenarios was conducted in order to determine their performance according to different criteria. *Algorithm 3b – Keep Order of ETA With Local Optimization* for slot allocation together with runway selection based on *minimal scheduled arrival time* of the new slot were selected as best performing and robust combination for use in real-world simulation scenarios. Experiments are described in Chapter 7.

8.1 Future Work

There are several areas in which the current work can be extended:

- The designed module for terminal air traffic control can be extended so it will take into account the configuration of STARs and their common segments during the arrival slot scheduling to prevent possible collisions on the shared parts of the routes. Current solution does detect these collisions and allows for their prevention by delaying one of the flights, but does not schedule the flights with this in mind.
- Other additional information about the airport configuration can be employed to plan the arriving flights on airports cross or otherwise influence each other. The configuration of taxiways and the fact if they allow the arriving airplane to leave the runway at high speed may also have impact on the frequency in which the airport is able to accept arriving flights.
- The implementation of holding patterns can be extended so it would allow intelligent stacking of multiple airplanes on single holding pattern at different heights. The order of the airplanes, holding heights and speeds can be assigned in a way that would minimize the fuel consumption.
- Communication between the TMA and en-route sector controllers can be improved and extended. En-route sectors can provide information about arriving flights sooner than when they appear on the TMA radar screen leaving the TMA controller more time to schedule and manipulate the traffic to achieve better results. TMA can also compute parameters of the incoming traffic and provide the neighboring en-route sectors with dynamically generated parameters for Miles In Trails which would mean that the traffic entering TMA sector would already be optimally spaced and allow the TMA sector controller to schedule the flights easily and with unnecessary delays.
- Further, more detailed experiments on the impact of individual algorithms and other terminal control procedures on the controllers workload can be conducted to learn more about how the procedures used affect the TMA controller as well as the controllers of near en-route sectors. Additional experiments on the impact of MIT use in neighboring en-route sectors on the overal airplane delays may be also conducted.
- Additionally, the AgentFly simulation system can be extended with the capability to schedule and simulate flights taking of the same runway that is used for arrivals.

Bibliography

[1] ICAO. *Strong Passenger Results and a Rebound for Freight Traffic in 2014.* International Civil Aviation Organization, Montréal, 18 December 2014. Available at: `http://www.icao.int/Newsroom/NewsDoc2014/COM.48.14.EN.pdf`

[2] Eurocontrol. *What is air traffic management* [online]. European Organisation for the Safety of Air Navigation [cited 2014-11-16]. Available at: `http://www.eurocontrol.int/articles/what-air-traffic-management`

[3] M. S. Nolan. *Fundamentals of air traffic control.* Thomson–Brooks/Cole, Belmont, CA, 4th edition, 2004.

[4] United States Department of Transportation, Federal Aviation Administration. *Aeronautical Information Manual - Official Guide to Basic Flight Information and ATC Procedures.* United States Department of Transportation, Federal Aviation Administration, 2014. Available at: `https://www.faa.gov/air_traffic/publications/ATpubs/AIM/`.

[5] ICAO. *Doc 4444 – Air traffic management.* International Civil Aviation Organization, Montréal, 15th edition, 2007.

[6] ICAO. *Doc 8168 – Aircraft Operations Volume I, Flight Procedures.* International Civil Aviation Organization, Montréal, 5th edition, 2006.

[7] ICAO. *Annex 11 to the Convention on International Civil Aviation: Air Traffic Services.* International Civil Aviation Organization, 2001.

[8] ICAO. *Annex 6 to the Convention on International Civil Aviation: Operation of Aircraft (Part I: International Commercial Air Transport — Aeroplanes).* International Civil Aviation Organization, 9th edition, 2010.

[9] *Airspace class. in: Wikipedia: the free encyclopedia* [online]. San Francisco: Wikimedia foundation, 2001- [cited 2014-11-18]. Available at: `http://en.wikipedia.org/wiki/Airspace_class`

[10] United States Department of Transportation, Federal Aviation Administration. *ORDER JO 7110.65U: Air Traffic Control.* Available at: `http://www.faa.gov/air_traffic/publications/atpubs/ATC/index.htm`.

[11] *VHF omnidirectional range. in: Wikipedia: the free encyclopedia* [online]. San Francisco: Wikimedia foundation, 2001- [cited 2014-11-26]. Available at: `http://en.wikipedia.org/wiki/VHF_omnidirectional_range`

[12] P. Ostwald, T. Topiwala and J. DeArmon. *The Miles-in-Trail Impact Assessment Capability.* 2007. Available at: `http://www.mitre.org/sites/default/files/pdf/06_0967.pdf`.

[13] D. Šislák, P. Volf, D. Pavlíček and M. Pěchouček *AGENTFLY: Multi-Agent Simulation of Air-Traffic Management.* 2012.

[14] D. Šišlák, P. Volf, M. Pěchouček, Ch. T. Cannon, D. N. Nguyen and W. C. Regli *Multi-Agent Simulation of En-Route Human Air-Traffic Controller.* 2012.

[15] Eurocontrol. *Base of Aircraft Data (BADA)* [online]. European Organisation for the Safety of Air Navigation [cited 2014-11-30]. Available at: `https://www.eurocontrol.int/services/bada`

[16] J. Clausen. *Branch and Bound Algorithms – Principles and Examples.* 1999. Available at: `http://www.imada.sdu.dk/Employees/jbj/heuristikker/TSPtext.pdf`

[17] T. Kolos-Lakatos. *The Influence of Runway Occupancy Time and Wake Vortex Separation Requirements on Runway Throughput.* MIT International Center for Air Transportation (ICAT) Department of Aeronautics & Astronautics Massachusetts Institute of Technology, 2013.

[18] *Hartsfield–Jackson Atlanta International Airport. in: Wikipedia: the free encyclopedia* [online]. San Francisco: Wikimedia foundation, 2001- [cited 2014-12-03]. Available at: `http://en.wikipedia.org/wiki/Hartsfield%E2%80%93Jackson_Atlanta_International_Airport`

[19] United States Department of Transportation, Federal Aviation Administration. *Airport Diagram – Hartsfield-Jackson Atlanta Intl. (ATL).* 11 December 2014. Available at: `http://aeronav.faa.gov/d-tpp/1413/00026ad.pdf`

[20] United States Department of Transportation, Federal Aviation Administration. *Document 2013-00287 – Amendment to Class B Airspace; Atlanta, GA.* 9 January 2013. Available at: `https://www.federalregister.gov/articles/2013/01/09/2013-00287/amendment-to-class-b-airspace-atlanta-ga`

[21] S. Albers. *Online scheduling.* In Introduction to Scheduling, edited by Yves Robert and Frederic Vivien. Chapman and Hall/CRC Press, 57-84, 2009. Available at: `http://www14.in.tum.de/personen/albers/papers/sched_chapter.pdf`

Appendix A

Enclosed Videos

01-multiple_runways.mp4 – The video shows separation of arriving airplanes for landing on two runways using horizontal diversion manouevres, the window on the left shows radar screen with runway plans and ATC plans of individual airplanes, the window on the right shows airplane trails in real world visualization.

02-single_runway.mp4 – The video shows planning of arriving airplanes on single runway using all implemented slot selection algorithms.

03-diversion.mp4 – The video shows application of horizontal diversion manoeuvre, the window on the left shows radar screen with ATC plan of the en-route sector, the window on the right shows radar screen with ATC plan of the TMA sector. Communication between controller and pilot is shown in the SECTOR RADIO window.

04-holding.mp4 – The video shows application of holding pattern, the window on the left shows radar screen with ATC plan of the en-route sector, the window on the right shows radar screen with ATC plan of the TMA sector. Communication between controller and pilot is shown in the SECTOR RADIO window.

Appendix B

Contents of Enclosed CD

```
|-- sources   - source codes of implemented modules
|-- videos    - videos described in Appendix A
|-- text      - this thesis in PDF format
\-- tex       - source code of this thesis in LaTeX with images in PDF and PNG
```