

Implémentations d'optimisation-simulation pour l'harmonisation des opérations dans les grands aéroports

Paolo Maria Scala

▶ **To cite this version:**

Paolo Maria Scala. Implémentations d'optimisation-simulation pour l'harmonisation des opérations dans les grands aéroports. Infrastructures de transport. Université Paul Sabatier - Toulouse III, 2019. Français. NNT : 2019TOU30120 . tel-02329839v2

HAL Id: tel-02329839

https://theses.hal.science/tel-02329839v2

Submitted on 29 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par
Paolo Maria SCALA

Le 7 octobre 2019

Implémentations d'optimisation-simulation pour l'harmonisation des opérations dans les grands aéroports

Optimization-Simulation implementations for harmonizing operations at big airports

Ecole doctorale : **AA - Aéronautique, Astronautique**

Spécialité : Mathématiques Appliquées

Unité de recherche : ENAC Lab

Thèse dirigée par
Daniel DELAHAYE et Pierre MARECHAL

Jury

M. Michael Schultz, Rapporteur
M. Miquel Angel Piera Eroles, Rapporteur
Mme Judith Rosenow, Examinatrice
M. Daniel Delahaye, Directeur de thèse
M. Pierre Marechal, Co-directeur de thèse
M. Miguel Mujica Mota, Invited member

To my beloved grandmother.

Abstract

The constant growth of air traffic, especially in Europe, is putting pressure on airports, which, in turn, are suffering congestion problems. The airspace surrounding airport, terminal manoeuvring area (TMA), is particularly congested, since it accommodates all the converging traffic to and from airports. Besides airspace, airport ground capacity is also facing congestion problems, as the inefficiencies coming from airspace operations are transferred to airport ground and vice versa. The main consequences of congestion at airport airspace and ground, is given by the amount of delay generated, which is, in turn, transferred to other airports within the network. Congestion problems affect also the workload of air traffic controllers that need to handle this big amount of traffic.

This thesis deals with the optimization of the integrated airport operations, considering the airport from a holistic point of view, by including operations such as airspace and ground together. Unlike other studies in this field of research, this thesis contributes by supporting the decisions of air traffic controllers regarding aircraft sequencing and by mitigating congestion on the airport ground area. The airport ground operations and airspace operations can be tackled with two different levels of abstractions, macroscopic or microscopic, based on the time-frame for decision-making purposes. In this thesis, the airport operations are modeled at a macroscopic level.

The problem is formulated as an optimization model by identifying an objective function that considers the amount of conflicts in the airspace and capacity overload on the airport ground; constraints given by regulations on separation minima between consecutive aircraft in the airspace and on the runway; decision variables related to aircraft entry time and entry speed in the airspace, landing runway and departing runway choice and pushback time. The optimization model is solved by implementing a sliding window approach and an adapted version of the metaheuristic simulated annealing. Uncertainty is included in the operations by developing a simulation model and by including stochastic variables that represent the most significant sources of uncertainty when considering operations at a macroscopic level, such as deviation from the entry time in the airspace, deviation in the average taxi time and deviation in the pushback time. In this thesis, optimization and simulation techniques are combined together by developing two methods that aim at improving the solution robustness and feasibility. The first method acts as a validation tool for the optimized solution, and it improves the robustness of solution by iteratively fine-tuning some of the optimization model input parameters. The second method embeds the optimization in a simulation environment by taking full advantage of the sliding window approach and creating a loop for a continuous improvement of the optimized solution at each window of the sliding window approach. Both methods prove to be effective by improving the performance, lowering the total amount of conflicts up to 23.33% for the first method and up to 11.2% for the second method, however, in contrast to the deterministic method, the two methods they are not able to achieve a conflict-free scenario due to the effect of uncertainty.

In general, the research conducted in this thesis highlights that uncertainty is a factor that affects to a large extent the feasibility of optimized solution when applied to real-world instances, and it, moreover, confirms that using simulation together with optimization has the potentiality to

deal with uncertainty. The framework developed can be potentially applied to similar problems and different optimization solving methods can be adapted to it.

Keywords: Optimization, Simulation, Integrated airport operations, Uncertainty

Resumé

L’augmentation constante du trafic aérien, spécialement en Europe, exerce une pression sur les aéroports, qui en conséquence sont souvent congestionnés. La zone aérienne entourant les aéroports, l’aire de manœuvre terminale (TMA), est particulièrement encombrée, puisqu’elle accueille tout le trafic aéroportuaire. Outre la zone aérienne, le partie sol fait aussi face à des problèmes d’encombrement, ainsi l’inefficacité des opérations en zone aérienne est transférée au sol. Cet encombrement des zones aériennes et terrestres des aéroports a pour conséquence de générer des retards, qui sont ensuite reportés sur les autres aéroports du réseau. Le problème d’encombrement affecte également la charge de travail des contrôleurs aériens qui doivent gérer ce large trafic.

Cette thèse porte sur l’optimisation des opérations intégrées aux aéroports, en considérant l’aéroport d’un point de vue holistique et en incluant les activités aériennes et terrestres. Contrairement aux autres études dans ce domaine, cette thèse apporte sa contribution en appuyant les décisions des contrôleurs aériens en terme de séquencement des avions et en atténuant l’encombrement de la partie sol des aéroports. Les activités terrestres et aériennes peuvent être abordées avec deux différents niveaux d’abstractions, macroscopique, ou microscopique, en raison de différent délais de prise de décision. Dans cette thèse, les activités sont modélisées au niveau macroscopique.

Le problème est formulé comme un modèle d’optimisation en identifiant une fonction objective qui prend en compte le nombre de conflits dans l’espace aérien et la surcharge au sol des aéroports; contraintes données par la régulation sur le minimum de séparation entre des avions consécutifs dans la zone aérienne et sur la piste de décollage; variables de décision liées au temps d’entrée de l’avion et à la vitesse d’entrée dans l’espace arien, au choix de la piste d’atterrissage et de la piste au départ et à l’heure de push-back. Le modèle d’optimisation est résolu en implémentant une approche par fenêtre glissante et par une version adaptée de la métaheuristique de recuit simulé. Des incertitudes sont ajoutées dans les activités en développant un modèle de simulation et en incluant des variables stochastiques représentant des sources d’incertitudes comme une variation de l’heure d’entrée dans l’espace aérien de l’aéroport, une variation de l’heure moyenne de temps du roulage ou encore une variation dans l’heure de push-back des avions. Dans cette thèse, les techniques d’optimisation et de simulations sont combinées en développant deux méthodes qui visent à améliorer la robustesse et la faisabilité des solutions. La première méthode se comporte comme un outil de validation pour la solution optimisée, elle améliore la robustesse de la solution en ajustant finement à plusieurs reprises plusieurs paramètres d’entrée du modèle d’optimisation. La deuxième méthode intègre l’optimisation dans un environnement de simulation en tirant pleinement parti de l’approche par fenêtre glissante et en utilisant une itération pour améliorer continuellement la solution optimisée à chaque fenêtre temporelle. Les deux méthodes ont prouvé leur efficacité en améliorant les performances, en diminuant le nombre de conflit jusqu’à 23.33% pour la première méthode et jusqu’à 11.2% pour la deuxième méthode, néanmoins, contrairement aux résultats dans les cas déterministes, elles ne sont pas capables de créer un scénario sans aucun conflit à cause de l’incertitude.

En général, l’étude menée dans cette thèse souligne que l’incertitude est un facteur qui affecte dans une large mesure la faisabilité d’une solution optimisée quand elle est appliquée à des instances du monde réel, et cela, confirme que l’utilisation combinée de simulation et d’optimisation a le potentiel de faire face aux incertitudes. Le système développé peut être potentiellement appliqué à autres problèmes similaires à d’autres méthodes d’optimisation peuvent être utilisées.

Mots clés : Optimisation, Simulation, Opérations aéroportuaires intégrées, Incertitudes

Acknowledgements

First, I would like to express all my gratitude to Prof. Daniel Delahaye, as my thesis supervisor. He enlightened me with his wisdom and made me always comfortable with his kindness.

Second, but first in my personal rank, I would like to thank Dr. Miguel Mujica Mota, my co-supervisor. You know me since already 6 years, when I was still a master student, young and with a lot of enthusiasm. Later, you guided me in this long and difficult journey that a PhD is. Now, I am still young, but with a bit of more experience, thanks also to you. As I often call you "Master", you have been a Master for me not only in the subject of Modeling and Simulation but also in the difficult subject of Life. I will be always grateful to you, and you will be always my Master.

A special thanks to the jury that agreed on evaluating my PhD thesis, especially Prof. Miquel Angel Piera and Dr. Michael Schultz, whose comments have been precious for me to improve my thesis manuscript. Dr. Judith Rosenow that kindly accepted to be examiner and Prof. Marechal who kindly supported my PhD candidature through these years.

I would like to mention all the different institutions that in these years have supported me, morally and financially. Without your support I could have not reached this far. Amsterdam University of Applied Science, Aviation Academy and Amsterdam International School of Business; Ecole Nationale de l'Aviation Civile; Ecole Doctorale Aeronautique Astronautique; Universite Toulouse 3 - Paul Sabatier.

Now, I would like to thank all the people that have had an influence in my life during these years of PhD, most of them are close friends, some of them acquaintances, many of them are still in my life, some of them not anymore. The list is long, I hope not to forget anyone, and in case I do, my sincere apologies.

I would like to thank my friends at ENAC. Every time I came there, you have welcomed me in the warmest and best way possible, I will always keep nice memories of our lunches together, coffees and card games. Among the others, I would like to mention Maji, the best PhD mate and one of the nicest person I have ever met, you will always be a friend for me. My dear friend Tambeth, I will remember our Toulouse football matches watched together; Jun, Man, Imen, Romaric, Florian, Vincent, Isabelle, Georges and Andrija.

A special mention to Dr. Richard Wu, and some of his PhD students Yimeng, Agnes and Dragon, at the UNSW in Sydney. With you I have spent very nice three months, both working and exploring a bit Australia.

Living in Amsterdam for the past years has been a great experience mostly for my special friends, the "Cocoteros". Alen, Mehdi, Humberto, Jayme and Joseba. I am sure that there will be many other adventures ahead of us! I want to mention also Simge, Nadine, Nico, Margarita and Daiki. The people from Koning Wilhelminaplein 318, Alen (again), Jakub, Eva and Robert.

Another special mention to Abdel el Makhloufi, a very kind-hearted person, a good friend and a supreme researcher; Ann and Idalia my Mexican(half-Belgian) colleagues/friends; my other Mexican friends Omar, Pedro, Alma, Tania, hopefully I will come to visit you again in Mexico!

A special mention also to the Dutch Benelux Simulation Society and the EUROSIM, I am proud to be part of this community.

My childhood friends from Crotone, even from far, your support was critical for my achievements, I owe you many. Dor, Tiger, Salvatore, Marco, you are the closest friends I have, and you know how much you all mean to me. I hope to celebrate many other achievements with you. Masto Arturo, Angela, Franco, Jessica, Giovanni, Giulia, Anastasia, il Bomber, Mimmo, Simone, Franci Fra, Ilaria, Sara, Isabella, Maria Grazia, you are my best friends as well, thanks for being part of my life. I cannot forget to mention all my dear childhood friends from Tufolo: Marco, Tommaso, Alessio, Luigi, Carlo, Demetrio, Scudo, Mattia, Dario, Danilo etc.

Finally, I want to say a big thank to my family, who was always supportive, my brothers Luca and Mario, my many cousins, uncles and aunts, my two sisters-in-law, my young “little pest” nephew Gioele and the other two little ones that are about to come to this world. I love you all.

Last but not least, the person who made all this possible, providing me education and teaching me the principles and values of life. Nonetheless, providing me support in all my new challenges, my beloved Nonna Franca. You are always in my heart and I know I am in yours as well. Dear Nonna, I hope you are proud of me, even just a 1% as much as I am proud of you would be already a lot. I love you.

Contents

Chapter 1 1

Introduction 1

1.1. Background and motivation 1

1.2. Airport characteristics and related operations 5

1.2.1. Airport definition and role 5

1.2.2. Airport classifications 5

1.2.3. Airport stakeholders 6

1.2.4. Airport sections 8

1.2.4.1. Landside 9

1.2.4.2. Airside 10

1.3. TMA airspace 16

1.4. Thesis contributions 18

1.5 Thesis outline 19

Chapter 2 20

Literature review 20

2.1. Optimization for airport operations 20

2.1.1. Aircraft scheduling and sequencing problem 20

2.1.2. Conflict detection and resolution (CD&R) problem 22

2.1.3. Airport surface management 22

2.1.4. Integration of airspace and ground operations 24

2.1.5. Gate assignment problem 25

2.2. Simulation approaches for airport operations 27

2.2.1. Specific-purpose simulation applications 28

2.2.2. General-purpose simulation applications 29

2.3. Optimization-Simulation solutions for airport operations 30

Chapter 3 33

Problem description, modeling and general solution approach 33

3.1. Airport operations at a macroscopic level 33

3.1.1. Operations description and modeling 33

3.1.2. Model assumptions 36

3.1.3. Optimization model 37

3.1.3.1. Decision variables 37

3.1.3.2. Objective and constraints 39

3.1.4. Model uncertainty 42

3.2. General architecture of the Opt-sim approach 44
3.2.1. Sliding window approach 44
3.2.2. Optimization by simulated annealing 46
3.2.3. Simulating by discrete event simulation 51
3.2.3.1. Simulation model of airport operations at a macroscopic level 53
3.3. Methodological approach for combining simulation together with optimization 56
3.3.1. Opt-sim method 1: ex post optimization 57
3.3.2. Opt-Sim method 2: Algorithmic approach for a continuous improvement of the solution 58
3.4. Conclusion 62
Chapter 4 63
Experiment on a real case study: Paris Charles de Gaulle Airport 63
4.1. Implementation of Opt-Sim method 1 65
4.1.1. Scenarios and results 65
4.1.1.1. Default scenario results 67
4.1.1.2. Results from scenario 1 and 2 68
4.2. Implementation of Opt-Sim method 2 79
4.2.1. Parameter setting for the Opt-Sim Method 2 79
4.2.2. Scenarios 80
4.2.3. Results 80
4.2.3.1. Default scenario 81
4.2.3.2. Alpha scenario 83
4.2.3.3. Separation minima increase scenario 85
4.2.3.4. Objective function weight scenario 87
4.2.3.5. Comparison between the scenarios 90
4.3. Conclusion 95
Chapter 5 97
Conclusion and future work 97
5.1. Conclusion 97
5.2. Future work 98
Bibliography 100
Appendix A: Simulation model of the Opt-sim method 1 (SIMIO) 110
Appendix B: Simulation model of the Opt-sim method 2 (Anylogic) 116

List of Figures

Figure 1.1: Air traffic trend for the last decades to date (a); Air traffic forecasts for the next 20 years (b) (Airbus GMF 2017) 1
Figure 1.2: Generated aircraft demand (Airbus GMF 2017) 2
Figure 1.3: Comparison between air traffic in 2016 and 2017 (EUROCONTROL 2018b) 2
Figure 1.4: Top five European airports in terms of passenger traffic and aircraft movements (EUROCONTROL 2018b) 3
Figure 1.7: Centralized airport terminal layout, Amsterdam Schiphol Airport (source: *https://www.ifly.com/amsterdam-schiphol/terminal-map*) 9
Figure 1.8: Decentralized airport terminal layout, Paris Charles de Gaulle Airport (source: *https://www.ifly.com/paris-charles-de-gaulle-airport/terminal-map*). 10
Figure 1.9: A schematic representation of contact and remote gate configuration. 11
Figure 1.10: Airport airside layout: parallel runways (source: AIP France). 13
Figure 1.11: Airport airside layout: crossing runways (source: PD-USGov-FAA). 13
Figure 1.12: Airport airside layout: hybrid configuration (source: AIP Netherlands). 13
Figure 1.13: Typical single runway capacity envelope. (Ball et al. 2007, Mirkovic 2015) 15
Figure 1.14: TMA representation 16
Figure 1.15: Schematic representation of the landing routes (STAR ad Instrument approach procedure). 17
Figure 1.16: lateral separation based on different geographic location 18
Figure 1.17. lateral separation based on different navigation aid 18
Figure 3.1: Airspace network modeling (2D): STARs of Paris Charles de Gaulle airport, west configuration 34
Figure 3.2: Low detail modeling of the airport ground components. 35
Figure 3.3: Schematic representation of the macroscopic approach for integrating airspace and ground operations. 35
Figure 3.4: Node conflict detection area 40
Figure 3.5: Max Overload and Average overload 41
Figure 3.6: Schematic representation of the sliding window approach 45
Figure 3.7: Simulated annealing algorithm 47
Figure 3.8: Initial temperature algorithm 49
Figure 3.9: Neighbor solution generation example 51
Figure 3.10: Main window of a general purpose simulation software (SIMIO) 52
Figure 3.11: Screenshot representing the experiment window of a general-purpose simulation software (SIMIO). 53
Figure 3.12: Screenshot of the flight schedule with the data input of the model. Each column corresponds to a data input record (SIMIO). 56
Figure 3.13: Schema of the opt-sim framework of method 1. 57
Figure 3.14: Architecture of the Opt-sim Loop 60
Figure 3.15: Comparison between Opt-Sim method 1 and 2 regarding the time horizon considered. 60
Figure 4.1: Landing routes at Paris Charles de Gaulle Airport, west configuration. 63
Figure 4.3: Average number of aircraft involved in airspace conflicts for scenario 1 (separation minima increase/red line) and scenario 2 (alpha increase/blue line) 69
Figure 4.4: Average number of airspace conflicts for scenario 1 (separation minima increase/red line) and scenario 2 (alpha increase/blue line). 70
Figure 4.5: Number of airspace conflicts that are in a specific conflict size interval (scenario 1). 71
Figure 4.6: Number of airspace conflicts that are in a specific conflict size interval (scenario 2). 71
Figure 4.7: Average airspace conflict size percentage. 72

Figure 4.8: *"runway in"* number of conflicts. 72
Figure 4.9: Number of "*runway in*" conflicts that are in a specific conflict size interval (scenario 1)...73
Figure 4.10: Number of "*runway in*" conflicts that are in a specific conflict size interval (scenario 2).73
Figure 4.11: Average *"runway in*" conflict size percentage. 74
Figure 4.12: "*runway out*" number of conflicts. 74
Figure 4.13: Number of *"runway out"* conflicts that are in a specific conflict size interval (scenario 1) 75
Figure 4.14: Number of *"runway out"* conflicts that are in a specific conflict size interval (scenario 2). 76
Figure 4.15: Average *"runway out"* conflict size percentage. 76
Figure 4.16: Number of taxiway network conflicts. 77
Figure 4.17: Number of terminal conflicts. 77
Figure 4.18: Total number of conflicts. 78
Figure 4.19: Number of conflicts for each time window for the scenario without schedule optimization 79
Figure 4.20: Results from the default scenario; time frame from 5.30AM to 10AM, windows 12 to 17. 82
Figure 4.21: Results from the default scenario; time frame from 6PM to 10.30PM, windows 37 to 42. 82
Figure 4.22: Results from the alpha scenario; time frame from 5.30AM to 10AM, windows 12 to 17. 84
Figure 4.23: Results from the alpha scenario; time frame from 6PM to 10.30PM, windows 37 to 42. .84
Figure 4.24: Results from the sep. min. increase scenario; time frame from 5.30AM to 10AM, windows 12 to 17. 86
Figure 4.25: Results from the sep. min. increase scenario; time frame from 6PM to 10.30PM, windows 37 to 42. 87
Figure 4.26: Results from the O.F. weight scenario; time frame from 5.30AM to 10AM, windows 12 to 17. 89
Figure 4.27: Results from the O.F. weight scenario; time frame from 6PM to 10.30PM, windows 37 to 42. 89
Figure 4.28: Comparison between the different scenarios and the non-optimized one. (time frame from window 12 to 17) 91
Figure 4.29: Comparison between the different scenarios and the non-optimized one. (time frame from window 37 to 42) 92
Figure 4.30: Airspace conflicts, O.F. weight scenario. Extended time frame (windows: 36-37-38-39-40-41-42-43) 95
Figure A.1: Schema of the simulation model of the Opt-sim Method 1 110
Figure A.2: Airspace model structure (SIMIO) 110
Figure A.3: Path object properties (SIMIO) 111
Figure A.4: Node object properties (SIMIO) 111
Figure A.5: Airport ground side model structure (SIMIO) 112
Figure A.6: Server object properties (SIMIO) 112
Figure A.7: Source object properties (SIMIO) 113
Figure A.8: Sink object properties (SIMIO) 113
Figure A.9: Processes in SIMIO 114
Figure B.1: Schema of the simulation model of the Opt-sim Method 2 116
Figure B.2: Airport airspace and ground model structure (Anylogic) 116
Figure B.3: Airspace visualization (Anylogic) 117
Figure B.4: Delay object properties (Anylogic) 117
Figure B.5: Queue object properties (Anylogic) 118

Figure B.6: Ground side visualization (Anylogic)....118
Figure B.7: Service object properties (Anylogic)....119
Figure B.8: Source object properties (Anylogic)....120
Figure B.9: Sink object properties (Anylogic)....120
Figure B.10: Function in Anylogic....121

List of Tables

Table 1.1. Commercial Airport Classifications (FAA 2016) 5
Table 1.2. Airport revenues (de Neufville et al. 2013) 7
Table 1.3. Longitudinal separation minima between two consecutive aircraft (ICAO 2016) 14
Table 2.1. General-purpose vs specific-purpose simulation software 29
Table 3.1. Summary of the main components and data input/instruction given to the flights based on their type. 37
Table 3.2. Separation minima ICAO (NM) 39
Table 3.3: Separation minima between consecutive aircraft on the runway (seconds) 42
Table 3.4. Operations affected by uncertainty and their value 43
Table 3.5. Summary of the object characteristics of the simulation model 54
Table 4.1. Number of movements classified based on the type of movement 63
Figure 4.2: Paris Charles de Gaulle Airport airside layout. 64
Table 4.2. Airside component capacity 64
Table 4.3. Average taxi time (s) 64
Table 4.4. Values of the input parameters of the optimization model for the two scenarios. In the first run the input parameters are set as default, while in the three iterations separation minima constraint relaxation gradually increases from 10% to 30%, and the cooling schedule parameter α gradually increases from 0.97 to 0.99. 65
Table 4.5. Value of the objective function broken down in all its components. Original schedule 66
Table 4.6. Uncertainty sources and their respective values. 67
Table 4.7. Value of the objective function broken down into the different components using the optimized schedule 68
Table 4.8. Values of the fine-tuned parameters of the Opt-sim method 2 81
Table 4.9. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17. 83
Table 4.10. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42. 83
Table 4.11. Comparison between the alpha scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17. 85
Table 4.12. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42 86
Table 4.13. Comparison between the sep. min. increase scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17. 88
Table 4.14. Comparison between the sep. min. increase scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42 88
Table 4.15. Comparison between the O.F. weight scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17. 90

Table 4.16. Comparison between the O.F. weight scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42 91
Table 4.17. Objective function value broken down into the main components for the most congested windows. Comparison between the different scenarios. (windows 12-13-14-15-16-17). 93
Table 4.18. Objective function value broken down into the main components for the most congested windows. Comparison between the different scenarios,. (windows 37-38-39-40-41-42). 94
Table 4.19. Comparison between the optimized scenarios and the non-optimized scenario considering the sum of the conflicts for all the windows. 95
Table A.1. List of processes implemented in the model 115
Table B.1. List of functions implemented in the model 122

Acronyms

ABS Agent Based Simulation

ACI Airports Council International

ACS Ant Colony System

AMAN Arrival Manager

ANOVA Analysis of Variance

ANSP Air Navigation Service Provider

ASS Aircraft Sequencing and scheduling

ATC Air Traffic Control

ATCO Air Traffic Controllers

ATM Air Traffic Management

ATS Air Traffic Service

CD&R Conflict Detection and Resolution

CPS Constrained Position Shifting

DES Discrete Event Simulation

DMAN Departure Manager

ECD&R Extended Conflict Detection and Resolution

FAA Federal Aviation Administration

FAF Final Approach Fix

FCFS First Come First Served

IAF Initial Approach Fix

IF Intermediate Fix

IATA International Air Transport Association

ICAO International Civil Aviation Organization

IFR Instrument Flight Rules

LCC Low Cost Carriers

LoS Level of Service

MAP Missed Approach Point

PCDG Paris Charles de Gaulle Airport

OR Operational Research

RHC Receding Horizon Control

RHG Receding Horizon Guidance

RPK Revenue Passenger Kilometer

SID Standard Instrument Departure Route

SMAN Surface Manager

SPSA Simultaneous Perturbation Stochastic Approximation

STAR Standard Arrival Route

TMA Terminal Manoeuvering Area

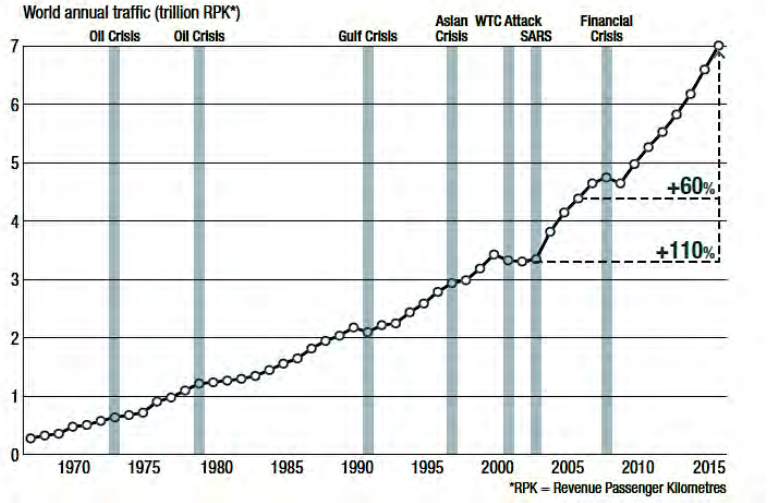
VFR Visual Flight Rules

Chapter 1

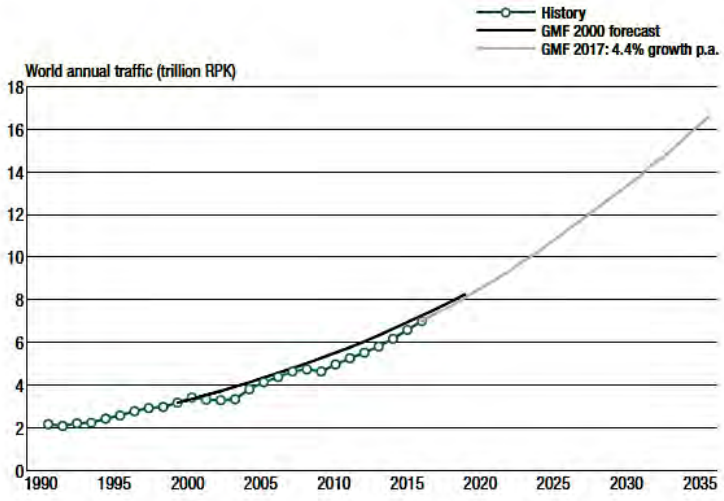
Introduction

1.1. Background and motivation

The air traffic has proven to be resilient and able to recover from many negative events such as economic and political crisis or terrorist attacks, keeping a constant growth through the years. As it is showed in the graph of Figure 1.1(a), looking back from 15 years ago until now, the air traffic has doubled. In the graph of Figure 1.1(a), it can be noticed that since the last financial crisis in 2008, the air traffic, in terms of revenue passenger kilometer (RPK), has grown with a steady rate. In the graph of Figure 1.1(b), it is shown the forecasted traffic for the next 20 years where it can be seen that the air traffic growth will continue with an average yearly growth rate of 4.4%, leading to double the current traffic (Airbus GMF 2017).



(a)



(b)

Figure 1.1: Air traffic trend for the last decades to date (a); Air traffic forecasts for the next 20 years (b) (Airbus GMF 2017)

The main aircraft producers Airbus and Boeing, have reported net commercial orders of 1109 and 912 aircraft in 2017, respectively. They had an order increase compare to 2016 of 52% and 37%, respectively (EUROCONTROL 2018a). Figure 1.2 represents the generated aircraft demand for the next 20 years based on the forecasted air traffic, in total there will be 34900 new delivered aircraft, 40% of which will be needed for replacement and 60% for growth (Airbus GMF 2017).

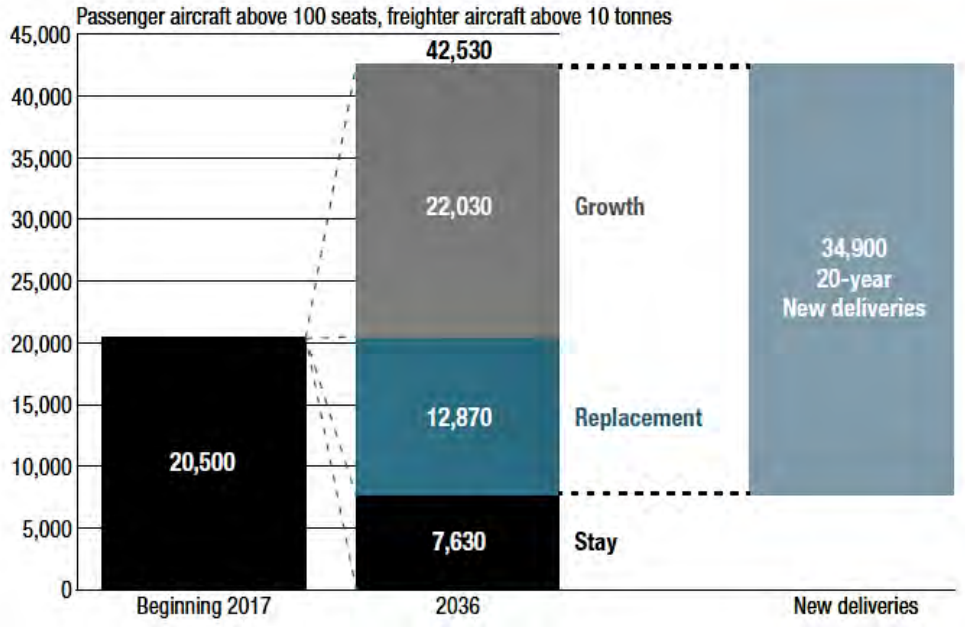


Figure 1.2: Generated aircraft demand (Airbus GMF 2017)

In 2017, Europe air traffic saw a big growth, reaching 10.6 million flights, surpassing the last record of 10.2 million in 2008. Figure 1.3 shows a comparison between 2016 and 2017 in terms of daily flights movements. Comparing with 2016, in 2017 there were 4.3% increase of the average daily flights.

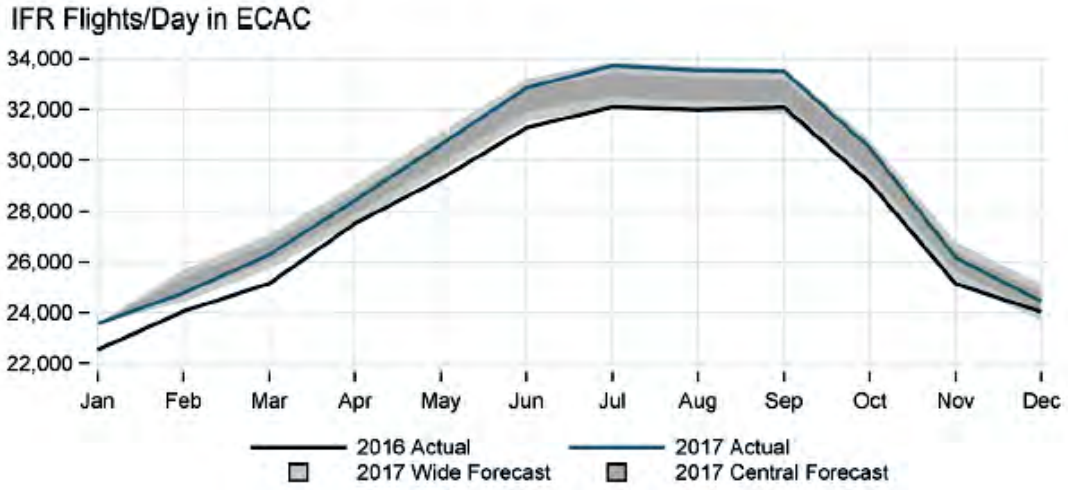


Figure 1.3: Comparison between air traffic in 2016 and 2017 (EUROCONTROL 2018b)

Passengers at European airports in 2017, were 8.5% higher in comparison with 2016. Figure 1.4 shows the major airport in Europe in terms of passengers and aircraft movements, with the top five airports having an amount of passengers above 63 million and aircraft movements above 449 thousand, each of them had an increase in both passenger traffic and aircraft movements between 2016 and 2017. As it can be noticed in Figure 1.4, in some airports, although the number of aircraft movements is equal or lower than other airports, the number of passengers is higher. As instance Frankfurt has 78 million passenger traffic, however, it is only fourth in the ranking, which is based on air traffic movements. This is explained by the difference in the number of wide-body aircraft movements at the airport, as these types of aircraft can carry a bigger amount of passengers than narrow-body aircraft. Looking into the air traffic movements of Frankfurt and Amsterdam airports in more detail, it could be discovered that in 2017, Frankfurt airport had a share of 22% of movements provided by wide-body aircraft (Fraport, 2018), while Amsterdam Airport had a share of 15% (Schiphol, 2017).

Rank	Airport	Passenger traffic	Aircraft movements
1	Amsterdam	68.5 million (+ 7.7%)	497K * (+ 3.7%)
2	Paris Charles De Gaulle	69.5 million (+ 5.4%)	476K (+ 0.6%)
3	London Heathrow	64.5 million (+ 6.1%)	476K (+ 2.7%)
4	Frankfurt	78.0 million (+ 3.1%)	474K (+ 0.2%)
5	Istanbul Atatürk	63.7 million (+ 5.5%)	449K (+ 0.3%)

*excluding general aviation

Figure 1.4: Top five European airports in terms of passenger traffic and aircraft movements (EUROCONTROL 2018b)

On the other hand, this constant growth has put pressure on airport, to which for some of them, capacity limit has already been reached. Environmental constraints, societal, technical and land use restrictions hinder the expansion of airports to accommodate more capacity, and one of the main indicator of congestion is the delay experienced by the flights. In December 2017, the average delay on departure of European flights was reported as 16.9 minutes, 3.3 minutes higher than the value of the same month in 2016. In the same period, 51% of flights were delayed on departures (>=5 minutes), 4% higher than the value reported in December 2016 (EUROCONTROL 2017a, EUROCONTROL 2018b). In general, July and December are the two busiest months over the year that are affected the most by delays, December mainly due to bad weather and July due to the higher traffic because of the summer season.

Airports are always looking for ways to cope with the increasing air traffic, in this context, long-term and short-term solutions are potentially able to solve the problem, however often they have the disadvantage of not being feasible in real-world operations. For instance, long-term solutions imply the expansion of the airport infrastructure such as terminals, runways and taxiways, the downside of these solutions is that they require long times of implementation and big investments. In this scenario, the risk involved is high due to the big investment and to the variability of external and internal environments such as financial markets and/or political scenarios. Short-term solutions imply the use of existing facilities in a more efficient way, and these solutions are the most effective given their application within the short time period. Short-term solutions have lower risk than long-term solutions due to the fact that they do not require

big investments, and moreover, a short horizon allows to make more reliable prediction of the future state of the internal/external factors affecting the operations. However, in both cases, due to the variability that affects any real-world scenario, the risk of implementing solutions (short- or long-term) which will not be effective is high. To this purpose, when proposing long/short-term solutions, uncertainty becomes a critical factor and needs to be considered. In this context, programs such as Next-Gen (FAA 2018) established by the FAA in the US, Single European Sky ATM Research (SESAR) established by EUROCONTROL in Europe (EUROCONTROL 2018a), and the Collaborative Action for Renovation of Air Transport System (CARATS) established by Japanese Civil Aviation Bureau (JCAB) (MLIT 2010), represent real examples of projects that stimulate the development of new procedures and applications aiming at improving the overall capacity and efficiency of the air transportation, at a given safety level. The FAA, EUROCONTROL, and JCAB, with their projects, play an important role as leading institutions in terms of development and deployment of applications with the objective of harmonizing the air traffic in the safest and most efficient way. Inside and outside the umbrella these programs, many researchers have tackled the problem of optimizing airport operations, including terminal 4aneuvering area (TMA) airspace operations (Beasley et al. 2000, Hu and Chen 2005, Michelin et al 2011, Furini et al. 2015, Sama et al. 2013) and airside operations (Montoya et al. 2010, Kjenstad et al 2013, Lee 2014, Simaiakis and Balakrishnan 2015, Guepet et al. 2017), while another part of research has been conducted for problems related to the optimization of the air traffic flow at large scale (continental and intercontinental routes) (Marzuoli et al. 2014, Allignol et al. 2016, Dhief 2018, Courchelle et al. 2019). Most of these studies have come up with the development of mathematical models providing exact solutions, employing techniques that belong to operational research (OR). In this context, two main knowledge gaps are identified: first, tackling the airport capacity management problem from a holistic point of view, by considering airspace and airside operations together; second, by considering the uncertainty related to the real-world operations. By filling this gap, it will be possible to come up with solutions which will be more resilient to the variability of the real-world operations. In this thesis, a resilient solution is defined as a solution which is less sensitive to perturbations, where perturbations are represented by uncertainty inherent to the operations. This thesis aims at filling these knowledge gaps, specifically, it aims at developing a decision support system to help air traffic controllers (ATCOs) in managing the traffic at major airports. It addresses the problem of making tactical decision for helping ATCOs in managing the traffic both in the TMA and on the ground, where tactical decisions are defined as the ones taken on the day of operations. Tactical decisions include the update of daily plans and the continuous capacity optimization according to the real traffic demand (EUROCONTROL 2017b). The methodology developed in this thesis imply the use of two techniques coming from OR such as optimization using a metaheuristic algorithm and discrete event simulation. Combining these two approaches, the solution of the problem will be addressed by considering also the uncertainty involved. For instance, considering time deviation of scheduled operations such as aircraft sequencing in the airspace or aircraft departures on the ground, would lead to the violation of technical constraints, which in turn would cause conflicts. The aim of this thesis, is to develop such methodology which will be able to minimize the occurrence of these conflicts while including uncertainty in the system.

1.2. Airport characteristics and related operations

1.2.1. Airport definition and role

The air transport system, has the main objective of connecting cities, countries, continents and therefore people from different parts of the world. It can be seen as a network of nodes and links, where the nodes are the airports and the links are the routes connecting them. In this context, the airport can be defined as the infrastructure intended to accommodate the flow of aircraft within the air transport network with the purpose of transporting passengers and/or freight. Airports play an important role as logistic means for the transportation of various goods, furthermore, they have also a positive impact in the development of the region or country that they serve. Their presence within the territory has the potential of attracting businesses and increasing the tourism (Khadaroo 2008), therefore, improving the overall regional/national economic performance. Studies show the positive correlation between airport development and employment rate growth (Goetz 1992, Green 2007, Van de Vijver 2015).

1.2.2. Airport classifications

The Federal Aviation Administration (FAA) classifies the airports into several categories based on the activities and the number of passenger boardings each year in: Commercial, Cargo, Reliever and General Aviation. Commercial airports, are defined as "publicly owned airports that have at least 2,500 passenger boardings each calendar year and receive scheduled passenger service" (FAA 2016). They are furthermore classified into: primary airport, with more than 10,000 passengers boardings, and non-primary, between 2,500 and 10,000 passenger boardings each year. In Table 1.1 a more detailed classification of commercial airports is given, classifying primary airports in: Large, medium, small and non-hub.

Table 1.1. Commercial Airport Classifications (FAA 2016)

Airport Classifications	Hub Type: Percentage of Annual Passenger Boardings	Common name
Primary	Large: 1%* or more	Large Hub
	Medium: At least 0.25%*, but less than 1%*	Medium Hub
	Small: At least 0.05%*, but less than 0.005%*	Small Hub
	Non-hub: More than 10,000, but less than 0.05%*	Non-hub Primary
Non-primary	Non-hub: At least2.500 and no more than 10,000	Non-primary Commercial Service

*percentage of the total annual passenger boardings in the United States.

Cargo service airports are the ones that besides any other transportation service, transport more than 100 million of pounds of cargo annually. Reliever airports are airports designated by the FAA to relieve other airport from congestion and to provide improved general aviation. General aviation airports are public-use airport without scheduled service and with less than 2,500

annual passenger boardings. In Europe, the Airports Council International (ACI) categorizes airports according to the passenger and freight traffic in four different groups: airports welcoming more than 25 million passengers per year (Group 1), airports welcoming between 10 and 25 million passengers per year (Group 2), airports welcoming between 5 and 10 million passengers per year (Group 3), airports welcoming less than 5 million passengers per year (Group 4) (ACI EUROPE, 2018). A similar classification is made by OAG (OAG 2018), who classifies airports based on the number of departing scheduled seats per year, having: small airports 2.5-5 million seats; medium airports 5-10 million seats; large airports 10-20 million seats; major airports 20-30 million seats; and mega airports 30+ million seats. Airports can be furthermore classified based on the proximity to large catchment areas, in primary and secondary airports (Dziedzic and Warnock-Smith 2016). In the recent years, with the development of low cost carriers (LCC), secondary airports have gain a lot of interest and have expanded their traffic. Secondary airports, due to their characteristics, fit best for the LCC business models, so they are preferred over primary airports. Usually, primary airports are chosen by specific airlines to be their hub, in a "hub and spoke" transport network, while secondary airports are used for a "point-to-point" transport network (Cook and Goodwin 2008). In this thesis, the focus is on big size airports which can be identified as Large Hub, Group 1 and mega airports according to the FAA, ACI and OAG classifications, respectively.

1.2.3. Airport stakeholders

There are many stakeholders acting within the airport system, each of them with its own function and scope. Airports are usually owned by the local/regional/national government, for any of these alternatives, there is an entity that is appointed to manage them and acting as interface between them and the other stakeholders such as airlines, passengers, and so on. This entity is usually known as the airport operator. The main function of the airport operator is to manage the airport resources, maintaining, developing and operating the airport on a daily basis (de Neufville et al. 2013). The airport operator can be established by the government as an ad hoc entity or it can be a private entity which is appointed by the government, usually with the formula of a contract, either ways, the local/regional/national government has always the right to claim its ownership. Generally, the sources of revenues for the airport are divided in: aeronautical and non-aeronautical. The former are related to the services directly provided to aircraft, passengers and cargo, while the latter are related to ancillary commercial services usually available at the airport. Moreover, there is also a third category of revenues which is called non-airport revenues, they are related to all the activities that generate income and are not directly related to aeronautical and non-aeronautical revenues. According to the International Air Transport Association (IATA) (IATA 2017), in 2015, airlines and passengers were estimated to have paid to airport and air navigation infrastructure $118.9 billion, while in 2016 $125.9 billion. In table 1.2 there is a summary of all the charges and fees that constitute that aeronautical and non-aeronautical revenues.

Airlines represent also an important stakeholder within the airport system, due to their route network and business model, they influence the air traffic at the airport and, therefore, they affect the airport capacity. They try to balance the air traffic demand, and in some cases they induce the demand, for example by applying low fares for tickets. As already mentioned before in section 1.2.2, the airlines choose some airports to be their hub, meaning that most of their routes will pass through that airport.

Table 1.2. Airport revenues (de Neufville et al. 2013)

Revenue source	Charge/fee
Aeronautical	Landing fees
	Terminal area air navigation fees
	Aircraft parking and hangar charges
	Airport noise charges
	Emission-related charges
	Passenger service charges
	Cargo service charges
	Security charges
	Ground handling charges
	En route aviation fees
Non-aeronautical	Concessions fees for aviation fuel and oil
	Concessions for commercial activities
	Revenues from car parking and car rentals
	Rental of airport land, space in buildings, and assorted equipment
	Fees for charged airport tours, admissions
	Fees from provision of engineering services and utilities
Non-airport revenue	Consulting, educational, and training services
	Management contract for operating terminal building or even other airports elsewhere
	Real-estate ventures and holdings
	Subsidiaries operating duty-free shops, hotels or restaurants
	Equity investments in into various commercial ventures
	Acquisition of shares in other airports

An airline, in its hub airport, has a big influence in the management of the space and resources of the airport, in fact it has often a reserved area for aircraft parking positions, and in some cases, they even manage an entire terminal or part of it. Usually, airlines are present in their hub airports also with their own resources such as ground handling facilities and maintenance areas. The hub-base airlines or home carriers, achieve several advantages due to the centralized provision of maintenance facilities, personnel and back-up aircraft at the hub (DLR 2008), other advantages derive from other factors such as route, frequency, fare and frequent flyer programs (Lijesen et al. 2006). Airlines have also influence in the airport slot management, which in most cases, shapes the daily traffic trend at the airport. Another stakeholder within the airport system are companies providing services such as: ground handling, aircraft maintenance and cargo handling. These entities are usually located in the airport airside and work as third party providing services to airlines and passengers. At secondary airports these types of services are provided mostly by these entities, since these airports are not under the influence of a major airline. However, also at hub airports, private service providers, as they were described before, are present with a big share, providing services to the other many airlines present at these airports. The Air Navigation Service Provider (ANSP) is an organization present at the airport with the purpose of managing the air traffic en route or in the TMA and airside of airports. It can be private or public, and their function is to provide the following services (EU 2011):

- Air traffic management,
- Communication, navigation and surveillance systems,
- Meteorological service for air navigation,
- Search and rescue,
- Aeronautical information services.

Especially for big airports that accommodate a big daily amount of air traffic movements, and for particular airspaces that are at the boundaries between different countries, the work of the ASNP - specifically ATCOs - becomes very demanding and difficult to fulfill (Majumdar and Ochieng 2002, de Oliveira et al. 2006, Hah et al. 2006).

Concessionaires are private entities that occupy and manage a part of the airport terminal and airside, under concession agreements with the airport operator. These concession agreements are represented by contracts of limited time period. For all the duration of the contract the concessionaires will operate their businesses inside the assigned portion of the airport. The most common example of concessionaires regards commercial activities (retailers) inside the airport terminal, and other commercial activities outside the terminal like rental cars, car parking and oil and fuel suppliers. Concessionaires are the ones that contribute to the non-aeronautical revenues to the airport; in the United States car parking and rental cars are the largest generators of non-aeronautical revenues. Last but not least, passengers are the principal stakeholders of the airport system, since they are end users of the airport services. Airports facilities should ensure the passengers a smooth access and connection to their various areas, in fact, each part of the airport terminal and of the airside is designed to provide the best passenger experience (ICAO 2006). To this regard, IATA developed the Level of Service concept, which consists in a set of values able to assess the ability of supply to meet demand by measuring passenger waiting times and level of comfort (available space per passenger) within the different areas of the airport terminal (IATA 2019). Recently airport have become a place not only for travel purposes but also for business and commercial purposes and even for leisure. Airports provide space for congresses and business meetings, but also, they reserve big area for shop retailers. The facilities intended only for the use of passengers such as check-in desks, security control, passport control and gate areas are designed to give the passengers the best comfort, providing enough space and including recreational areas. Regarding the airside, gate areas and taxiway layout are designed to provide the passengers the best quality service in terms of journey comfort and duration. Due to all these aspects, airports terminal and airside design is influenced by the passenger's experience, which in turn is a crucial factor for the airport success and attractiveness (Potgieter et al. 2014, Chen et al. 2014, Carballo-Cruz and Costa 2014).

1.2.4. Airport sections

The airport infrastructure can be divided into two different sections: landside and airside. The landside includes the portion of the airport within the boundaries of the terminal. The airside or commonly known as ground side, is identified as the portion of the airport outside the terminal, therefore, it includes runways, taxiway network, apron areas, and other areas used for other aviation purposes such as maintenance hangars, freight warehouses, military areas, training areas and so on. In the following sections these areas will be described with more detail.

1.2.4.1. Landside

The landside area includes the airport passenger terminal, the main function of the terminal is to receive the passengers and, based on their purpose, convey the flow properly. There are different flows inside a terminal, passengers flow and baggage flow. Passengers flow can be classified, in turn, in arrival, transfer and departing. Baggage flow can be inbound and outbound. Airport terminals can have different layouts, usually they are designed following two different schemes: centralized or decentralized. Centralized terminal layout consists in one terminal that is connected to all the gates, usually it is constituted by several piers where the gates are located. The main advantages coming from this type of layout is, for example the access to the terminal, which favors a unique access by a rail or other forms of public transport. On the other hand, the walking distance measured for reaching one side from another can be long, therefore, in this type of terminals it is important to manage the transfer passengers, so to allocate them not too far from their arrival gate to the transfer gate. In figure 1.7 it can be seen an example of centralized airport terminal layout from Amsterdam Schiphol airport.



Figure 1.7: Centralized airport terminal layout, Amsterdam Schiphol Airport (source: *https://www.ifly.com/amsterdam-schiphol/terminal-map*).

Another type of airport terminal layout is the decentralized one. It is usually constituted by several terminals, where in each terminal there are some resource that are replicated such a: check-in desks, security checks, passport controls and so on, therefore, each terminal operates as an independent one. The main advantage coming from this layout is that passengers walking distance is reduced, on the other hand, it might be not ideal for transfer passengers to switch from one terminal to another one, and it does not ensure a unique access by rail or other forms of public transport (Ashford et al., 2013). In Figure 1.8 it can be seen a typical decentralized airport terminal layout as the one in Paris Charles de Gaulle Airport.



Figure 1.8: Decentralized airport terminal layout, Paris Charles de Gaulle Airport (source: *https://www.ifly.com/paris-charles-de-gaulle-airport/terminal-map*).

1.2.4.2. Airside

The airside area is a secured area where aircraft and ground vehicles have access. In this area, passengers are passive entities since at this stage they are already inside the aircraft, while ground handlers have an active role due to their interaction with the aircraft. At large-sized airports, the airside can cover a vast amount of land and can present a complex layout, which makes difficult for the ground traffic controllers to manage (Mogford et al. 1995, Chua et al. 2015, Taurino et al. 2017).

- **Gate operations**

The apron area is the airside airport area where aircraft are parked for an amount of time that is between the arrival phase and the departing phase. The apron area includes not only the parking stands but also the adjacent area that is used by the aircraft to make their parking maneuvers. The parking maneuvers, both for arriving and departing aircraft are assisted by the support of the ground handlers, who guide the aircraft to make a proper parking maneuver at the right parking stand. The apron area and the relative gate can be designed following a different layout, resulting in contact and remote gates. Contact gates are the ones adjacent to the terminal and usually passengers are transferred from the terminal gate to the aircraft using bridges. At contact gates, ground handlers use towing vehicles in order to push back aircraft and trail them until they reach a part of the apron area where they can turn their engines on. Remote gates are the ones that are located in an area of the airside distant from the terminals; with this configuration passengers will be transferred to the aircraft by means of buses. This solution is often seen in airports with a small terminal where the only contact gates are not enough for the given traffic. At remote gates, usually, ground handling towing operations are not necessary, since the aircraft

can start moving by themselves by turning the engines on as soon as they are cleared to leave the gates. In figure 1.9 the two different apron layouts are shown.

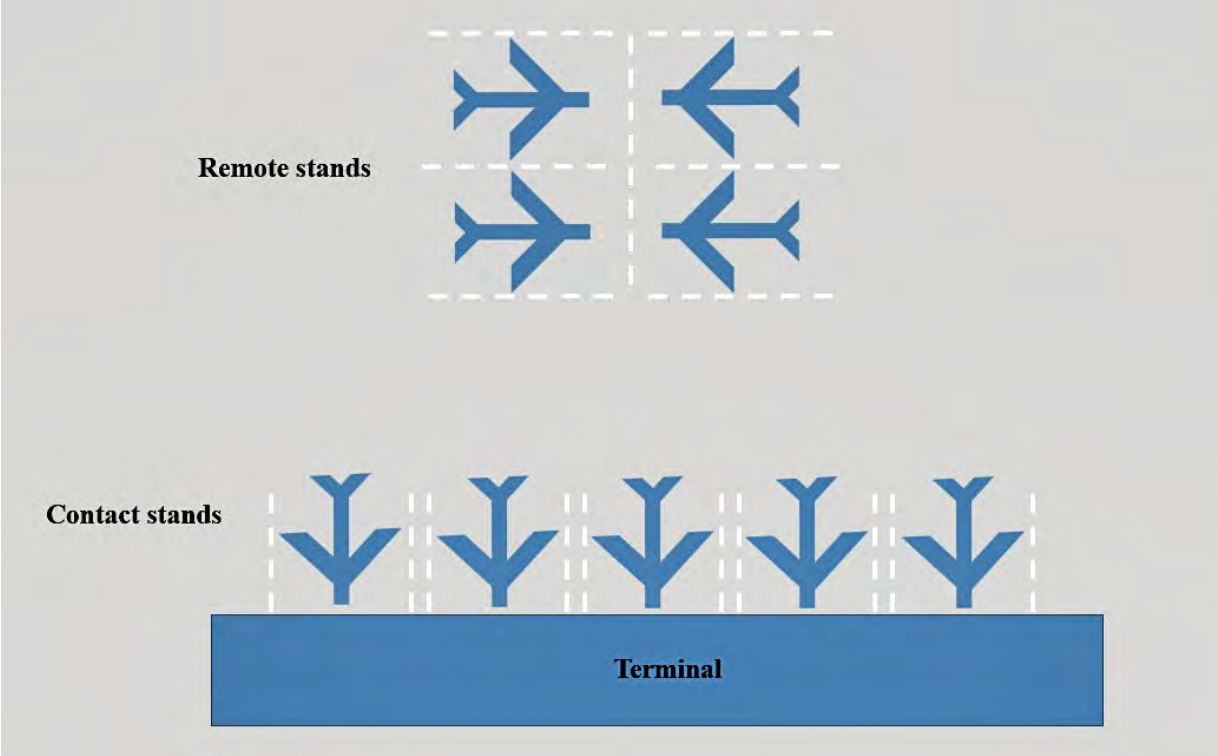


Figure 1.9: A schematic representation of contact and remote gate configuration.

When an aircraft is parked at the gate (contact or remote), it undergoes a sequence of operations called turnaround operations. Turnaround operations involved mainly aircraft passengers boarding/deboarding, baggage loading/unloading, water service, refuel service, catering service, and cleaning service among others. These operations are subjects to changes and also can require different times, depending on some factors such as the type of aircraft, the type of flight (short or long haul) and the airline business model (full service or low cost). For instance, low cost carriers for short haul flights aim at achieving maximum 30 minutes of turnaround time, while full service carrier in intercontinental routes can have a turnaround time up to 3 hours. The turnaround time is an indicator of efficiency of the terminal gate capacity, and airlines find very important to keep it as low as possible in order to minimize the costs and be competitive within the market. Each turnaround operation implies the use of specific vehicles, therefore the management of these resources impacts the gate capacity. In the literature we can find several studies about this topic proving its relevance in the field (Wu and Caves 2004, Adeleye and Chung 2006, Fricke and Schultz 2009, Norin et al. 2012, Makhloof et al. 2014, Airbus 2019, Boeing 2019).

- **Taxiway operations**

The taxiway network has the objective of linking the gate apron area with the runways. Aircraft that cross the taxiway network are usually guided by the ground controller from the gate to the assigned runway and vice versa. While crossing the taxiways, aircraft must respect some speed limit and minimum distance between consecutive aircraft due to safety reasons (ICAO 2018). Currently there are not regulations about these speed and distance limits, so they are ensured by

the ground controllers and pilots decisions based on direct visualization (Gotteland et al. 2009, Lee 2014). (Ground controllers give instructions to aircraft by addressing them to the taxiway routes, which are identified as the ones most utilized. Especially for complex taxiway network, taxiway junctions are identified as hot-spots, which are areas where traffic gets congested with high likelihood. It is important to consider these hot-spot when managing the traffic on the ground since they can form potential bottleneck in the system. Taxiway operations are fulfilled by the cooperation between aircraft pilots and ground controllers, where the latter give instructions to the pilots about which route to follow. At airports where the gate area is close to the runway, there are few taxiway routes connecting a specific gate with a specific runway, therefore, resulting in short taxiway times. On the other hand, at major airports, given a specific gate and a specific runway to connect, there can be several alternative taxiway routes which increase the complexity in managing the traffic by the ground controllers; due to these reasons, the likelihood of having long taxiway times is high. In order to mitigate this effect, the topic of ground management, especially concerning taxiway routings, has been extensively studied by researchers as an individual problem and also within the context of the airport departure problem. The main objectives pursued within this specific type of problem are: achieving a smooth flow of aircraft on the ground, reducing taxiway times, minimizing delays for departures and maximizing the departure throughput (Montoya et al. 2010, Kjenstad et al 2013, Lee 2014, Simaiakis and Balakrishnan 2015, Guepet et al. 2017).

- **Runway operations**

The runway system represents the main feature of the airport airside, since its function is to accommodate landings and departures, that together define the maximum capacity of the airport in terms of air traffic movements. The runway system is considered as the main bottleneck of the airport (Idris et al. 1999, Lieder et al. 2014, Simaiakis and Balakrishnan 2015), therefore, when designing an airport, the number of runways and their layout, it assumes a crucial aspect to consider. Regarding the layout of the runway system, we can have parallel runways, crossing runways and both. Building parallel runways represents an effective way of increasing the capacity of airports, since they can be used independently for different operations (landings and takeoffs). Parallel runways layout, as already adopted by several airports (e.g. Paris Charles de Gaulle, Los Angeles International, Delhi Indira Gandhi), represents the next stage of capacity evolution (de Neufville et al., 2013). The number of runways is based on several criteria, the main ones can be summarized as follows: forecasted air traffic demand, land availability and noise constraints. Within the runway system, a factor which is important besides the number of runways, is the number of *active* runways. Due to the layout of the runways, not all the runways can be used at the same time, so the layout of the runway system plays an important role for the computation of its capacity. For example, crossing runways cannot be used for independent operations at the same time, as well as close spaced parallel runways. In figure 1.10, 1.11 and 1.12 are depicted three different runway system layouts.

Figure 1.10, depicts a parallel runway system; in this specific example there are two northern and two southern parallel runways. Each of the two parallel runways are spaced between 2.500 and 4.300 miles, which is enough for them to work as independent runways only for the simultaneous departures and simultaneous departures/landings. In order to be able to operate simultaneous landings they need to be spaced by more than 4.300 miles.

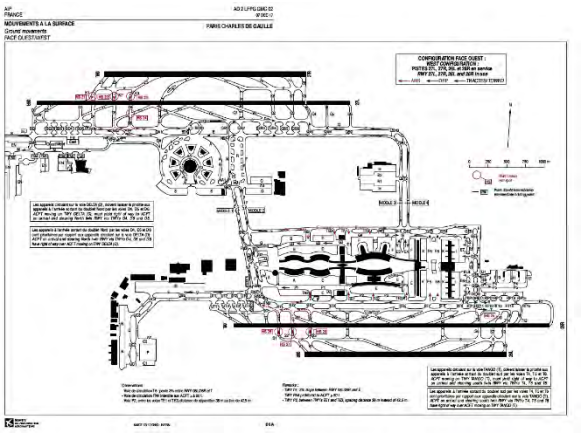


Figure 1.10: Airport airside layout: parallel runways (source: AIP France).

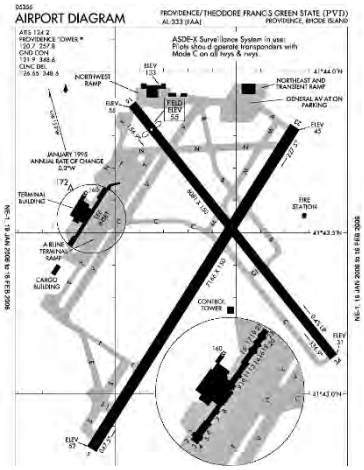


Figure 1.11: Airport airside layout: crossing runways (source: PD-USGov-FAA).

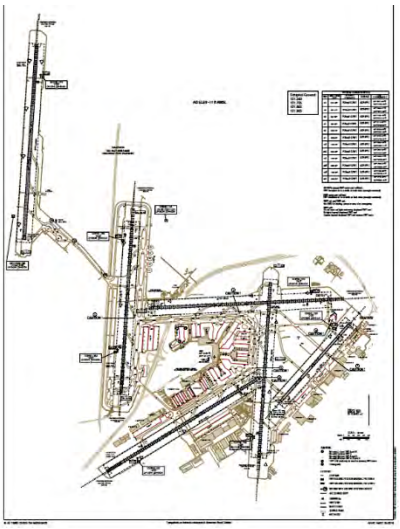


Figure 1.12: Airport airside layout: hybrid configuration (source: AIP Netherlands).

On the other hand, if they were spaced by less than 2.500 miles (close spaces runways), they could not operate any movement simultaneously. Figure 1.11 shows two crossing runways; in this case only one movement at a time is allowed, while Figure 1.12 shows a hybrid layout composed by both independent parallel runways and crossing runways. In the latter case, it is worth noting that although there are 6 runways, only 3 runways can be used simultaneously (active runways). The operations carried out on the runway are named as landings and take offs. The landing phase starts when the aircraft is in its final approaching phase, continues when it touches down, and ends when it exits the runway. The take off phase starts when the aircraft enters the runway, crosses it, starts to climb, and ends when it reaches a visual flight rules (VFR) pattern or when reaching 1000ft, whichever comes first. Since the runway system is considered as the bottleneck of the airport system (Idris 1999, Lieder et al. 2015, Simaiakis and Balakrishnan 2015), it is important to define and evaluate its capacity together with the factors that influence it. A common definition of runway capacity is the hourly runway capacity, which is the number of movements (landings and takeoffs) that are performed in one hour. The capacity

of a runway changes based on the type of movement that it accommodates, such as runway only for landings, only for departures or for both. As stated by regulation, on the runway it is allowed only one movement at a time, and moreover, two consecutive movements need to be separated by minimum separation due to safety reasons (ICAO 2016). In Table 1.3, these separation minima are presented, they depend on which category the leading and trailing aircraft belong. Aircraft are categorized by ICAO with the so called wake turbulence category, based on their maximum take off mass (MTOM) (ICAO 2019). The wake turbulence category are defined as follows:

- *Heavy*, aircraft types of 136000 Kg or more,
- *Medium*, aircraft types less than 136000 Kg and more than 7000 Kg,
- *Light*, aircraft types of 7000 Kg or less.

Table 1.3. Longitudinal separation minima between two consecutive aircraft (ICAO 2016)

		Leading aircraft		
	Category	Heavy	Medium	Light
Trailing aircraft	Heavy	4	3	3
	Medium	5	3	3
	Light	6	5	3

Runways can accommodate two types of movements: Arrival (A) and Departure (D), depending on the sequence of movements performed on the runways (arrival followed by arrival, A-A; arrival followed by departure, A-D; departure followed by departure, D-D; departure followed by arrival, D-A), different separation minima are applied, and they can be derived by the equations (1), (2), (3) and (4) (de Neufville et al. 2013).

In the specific, for the A-A case:

$$s_{ij} = d_{ij} \tag{1}$$

For the A-D case:

$$s_{ij} = l_i^r \tag{2}$$

For the D-D case:

$$s_{ij} = \begin{cases} min\left\{120, \frac{d_{ij}}{v_i}\right\}, i = Heavy, \\ max\left\{l_i^r, \frac{1}{v_i}\right\}, i = Medium, \\ max\left\{l_i^r, \frac{0.75}{v_i}\right\}, i = Light, \end{cases} \tag{3}$$

For the D-A case:

$$s_{ij} = max\left\{l_i^r, \frac{2}{v_j}\right\} \tag{4}$$

Where s_{ij} is the required separation minima between two consecutive aircraft i and j. In the A-A case (1), s_{ij} is equal to the ICAO standard minimum separation distance based on the aircraft type at the runway threshold between two consecutive aircraft i and j, d_{ij}. In the A-D case (2), s_{ij} is equal to the runway occupancy time of the aircraft i at the runway r, l_i^r. In the D-D case (3), s_{ij} is calculated according to the following rules: if the leading aircraft i is of type *Heavy*, the minimum separation must be at least 120 seconds or the time separation calculated according to the standard separation minima d_{ij} and the speed of the leading aircraft i at the runway threshold, v_i; if the leading aircraft i is of type *Medium*, the minimum required separation must be at most equal to the runway occupancy time of aircraft i, l_i^r, or calculated as time distance by assuming 1 nautical mile as the required distance and v_{ij}, as the speed of the leading aircraft i at the runway threshold; if the leading aircraft i is of type *Light*, the minimum required separation must be at most equal to the runway occupancy time of aircraft i, l_i^r, or calculated as time distance by assuming 0.75 nautical miles as the required distance and v_{ij}, as the speed of the leading aircraft i at the runway threshold. In the D-A case (4), s_{ij} is calculated as the maximum value between the runway occupancy time for aircraft i at the runway r, l_i^r, and the time distance by assuming 2 nautical miles as the required distance and v_{ij}, as the speed of the leading aircraft i at the runway threshold. In Figure 1.13 it is showed a typical capacity envelope for a single runway.

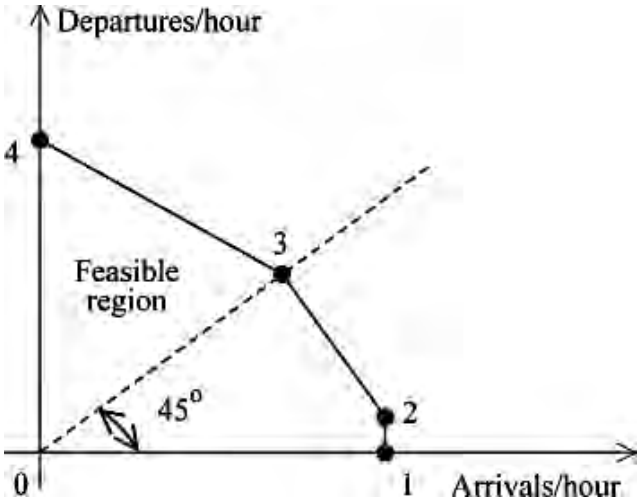


Figure 1.13: Typical single runway capacity envelope. (Ball et al. 2007, Mirkovic 2015)

Figure 1.13 denotes four vertexes that represent a different configuration of arrivals departures. In point 1 arrivals are maximized while departures are zero, in point 4 it can be seen the opposite phenomenon. Point 2 shows that even if there are some departures, the arrivals are still at a maximum value, meaning that, this point represents the maximum number of departures that can be processes without lowering the number of arrivals. From point 2 to point 3 as soon as the number of departures increases, the number of arrivals decreases until reaching point 3 where there are the same number of departures and arrivals. From point 3 to point 4, the same phenomenon continues until point 4 where the maximum number of departures is reached while having zero arrivals. At some airports, given the configuration of the runways and taxiway network, aircraft might need to cross the runway and use it as a taxiway; these types of movements are commonly named as “runway crossings”. Runway crossings have also an impact

on the runway capacity since the runway can be occupied by only one aircraft at a time. However, runway crossings influence the runway capacity to a less extent since the runway occupancy time for crossings operations is lower than normal runway occupancy time for arrivals/departures operations.

1.3. TMA airspace

The Terminal Maneuvering Area is a control area normally established at the confluence of ATS (Air Traffic Service) routes in the vicinity of one or more major aerodromes (ICAO 2005). Especially for busy airports, it can be a congested area due to the traffic converging to the runways and also due to the outbound traffic. In this context, the ANSP has the role of managing the traffic, having as main objective to ensure safety, and besides that, their aim is to make an efficient flow with the purpose of minimizing delays and increasing the airspace capacity. The TMA can be seen as an upside-down wedding cake shape portion of airspace surrounding the airport, its extension varies according to the specific airport. Figure 1.14 helps to better understand the TMA airspace.

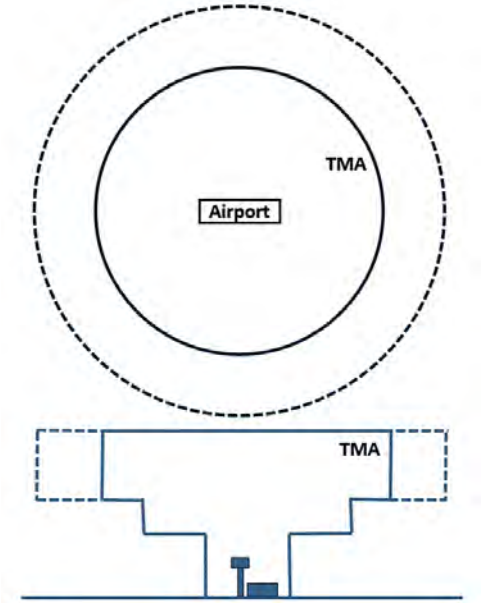


Figure 1.14: TMA representation.

When the aircraft fly under Instrument Flight Rules (IFR), their flight paths are facilitated by the implementation of Standard Arrival Routes (STARs) and Standard Instrument Departure Routes (SIDs). STARs and SIDs are standard routes that expedite the safe and efficient flows of air traffic operating to and from the same or different runways. STARs and SIDs are published routes that can be followed by the flight crew unless the air traffic controllers give different instructions. Each runway can have one or more STAR/SID, each of them ensures aircraft to fly at a certain altitude level, under speed restrictions and following some significant points (waypoints). The last descending path before landing, under IFR, is called instrument approach procedure. This procedure includes a series of predefined maneuvers that leads to landing at a predefined runway. The instrument approach procedure can be divided into two main segments, named initial approach segment and final approach segment. In some cases, another additional

segment can be part of the instrument approach procedure, the intermediate approach segment. Figure 1.15 shows a schematic representation of them.

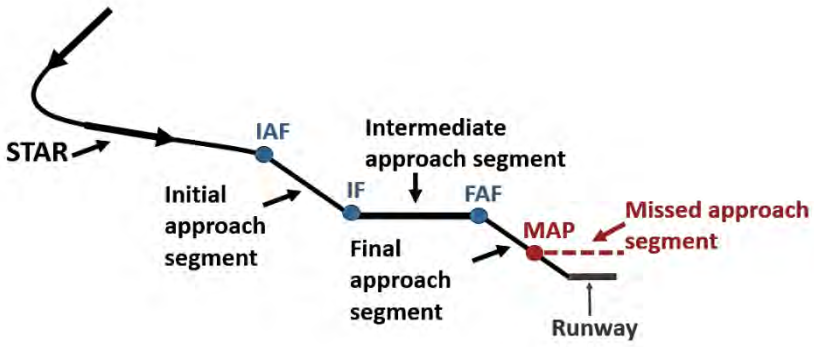


Figure 1.15: Schematic representation of the landing routes (STAR ad Instrument approach procedure).

The initial approach segment starts at the initial approach fix (IAF) and ends at the final approach fix (FAF), or when not present, ends at the intermediate fix (IF). The intermediate approach segment, when present, starts from the IF and ends at the FAF. Finally, the last segment of the instrument approach procedure, the final approach segment, starts from the FAF until reaching a designed missed approach point (MAP) or when reaching the runway. In case of aircraft unable to complete the final approach segment, a missed approach procedure can start by flying the missed approach segment from the MAP until reaching the IAF or another en route waypoint (FAA 2017).

The operations carried out in the TMA, under IFR, are restricted by predefined constraints that involve speed, and separation between consecutive aircraft. Speed limits can be consulted in the STARs and SIDs routes that are published by the air traffic service (ATS). Usually, an aircraft entering the TMA from one of the STARs until reaching the IAF, is not allowed to fly at a speed that exceeds 250 kts. Other specific speed limitations can be found in other waypoints along the STAR, and in the main waypoints of the instrumental approach procedure (IAF, IF and FAF). At the FAF, the final descending speed is reached and kept until touching down to the runway. The same concept is applied to the SIDs, where aircraft fly at a certain speed based on the speed limits defined on each waypoint placed along each SID. Concerning the separation minima, there are two different separations that must be respected due to safety reasons: horizontal and vertical. Horizontal separations are ensured in terms of longitudinal and lateral separation. Longitudinal separation minima between two consecutive aircraft depends on the aircraft type of the leading and trailing aircraft; in this context the International Civil Aviation Organization (ICAO) has defined the different separation minima, as they are shown in Table 1.3. Recently Eurocontrol (EUROCONTROL 2015) has revised the longitudinal separation minima and re-categorized them in order to reduce the separation minima for some types of aircraft and therefore increasing the capacity.

The lateral separation between two consecutive aircraft is ensured by referencing to different geographic locations and/or by referencing to the same navigation aid as it is represented by Figures 1.16 and 1.17. Vertical separation is ensured by leaving at least 1000 ft (300m) of vertical distance between two aircraft. In the literature there a many studies about the

optimization of the aircraft sequence for arrivals and departures, most of them employing mathematical models where speed and separation minima are considered as the main constraints (Beasley et al. 2000, Hu and Chen 2005, Michelin et al 2011, Furini et al. 2015, Sama et al. 2013).

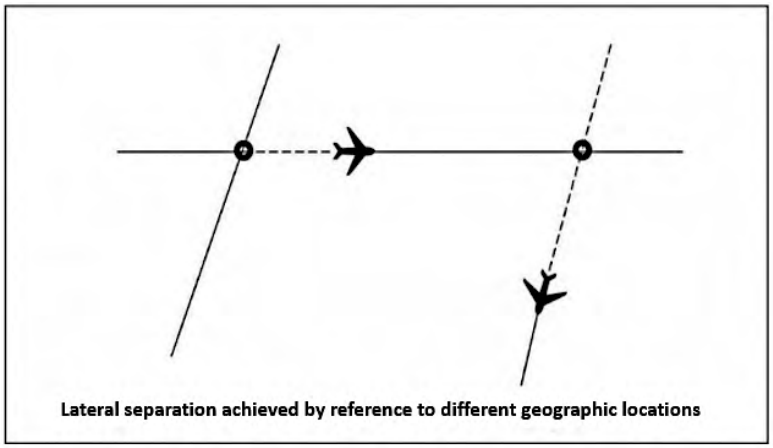


Figure 1.16: lateral separation based on different geographic location

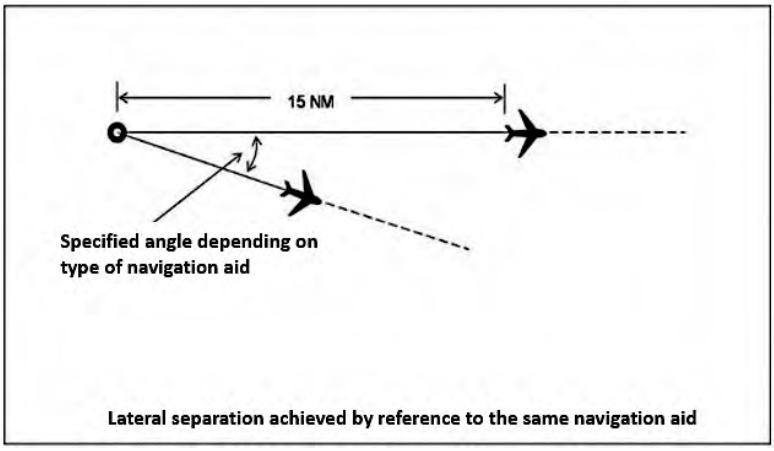


Figure 1.17. lateral separation based on different navigation aid

1.4. Thesis contributions

This thesis focuses on the development of a methodology that combines the use of optimization and simulation techniques for dealing with the integrated airport operations under uncertainty. From the literature review presented in the previous sections some knowledge gaps were highlighted, these knowledge gaps will be filled in this thesis by the following contributions:

- Considering airport operations in an integrated way, where both TMA and airport airside operations are modelled, optimized and simulated within a certain time-horizon (macroscopic level), addressing:

    - The optimization of the sequence for both arrival and departure aircraft flow in the TMA to and from the runways, ensuring safety (conflict free) and smoothness of operations.
    - The optimization of the capacity for the airport airside components. In this context, taxiways and terminals are considered as airside components, and the objective is to avoid any congestion (overload of aircraft) within a certain time-horizon.
- Including the uncertainty that characterizes the airport operations and evaluate its effect.
- Developing a methodology that combines optimization and simulation for the analysis of systems that are affected by uncertainty, with the objective of improving the robustness of the optimized solution.
- Investigating the different uses of optimization combined with simulation and evaluate the potential benefit:
    - Simulation used as a validation tool for the optimized solution.
    - Optimization fully integrated in a simulation framework for a continuous improvement of the robustness of the optimized solution.

1.5. Thesis outline

The thesis is organized as follows: Chapter 2 presents the state of the art concerning problems related to aircraft scheduling in the airspace and airport airside capacity management. For each of these problems it has been made a review concerning problems tackled with optimization techniques, simulation techniques and the combination of both techniques. In Chapter 3 a detailed description of the problem tackled in this thesis is given. The operation involved and the level of abstraction used for modelling these operations are described, together with the definition of the sources of uncertainty considered in this problem. Moreover, it introduces the methodology developed in this thesis, which addresses the use of optimization and simulation techniques together. This chapter includes the description of the implementation of two different ways of using optimization and simulation. Chapter 4 introduces the case study applied for testing the proposed methodology, together with the description of the experiments conducted and the relative results. Finally, in Chapter 5, the conclusions of the thesis are drawn by discussing and interpreting the results and making an overall evaluation of the methodology. Furthermore, future research directions related to the topic are outlined.

Chapter 2

Literature review

In the last decades, airport operations, have emerged as a consolidated field of study, as they have been addressed by many researchers. Nowadays, it continues to be an attractive field of research due to the big development of different techniques which are mainly included in the subject of operation research. In this context, studies about airport operations focus on improving their efficiency and effectiveness with the aim of improving airport capacity, both for airspace and ground areas. Recently, simulation has also emerged as a technique for evaluating airport performance. Even more recent studies have coupled the use of simulation together with optimization, trying to benefit from the advantages of both techniques. In the following sections, a literature review of the most relevant techniques that use optimization, simulation and optimization together with simulation applied to airport operations is presented.

2.1. Optimization for airport operations

In this section are mentioned the most relevant works which are related to the use of optimization techniques for solving various airport capacity problems.

2.1.1. Aircraft scheduling and sequencing problem

Concerning the problem of the scheduling of landing aircraft, Bennel et al. (2013), made a review about the airport runway scheduling, including the main solution approaches such as dynamic programming, branch and bound, heuristics and meta-heuristics. Beasley et al. (2000) developed a model for tackling the static case, where they formulated a mixed-integer non-zero mathematical model solved optimally using linear programming based tree search and also with a heuristic algorithm based on an "upper bound and restart" strategy. In this context, static means that decisions for flights are made off-line, where the set of aircraft that is going to land is known in advance. In another work of Beasley et al. (2001), the landing aircraft scheduling problem was solved by applying a population heuristic (genetic algorithm). The dynamic case of the landing aircraft scheduling problem was also studied by Beasley et al. (2004); in this context, dynamic, or on-line, means that decisions such as landing time and landing runway were taken by considering the environment changes as time passes. This problem was solved optimally as in their previous work, and furthermore, employing the two heuristic algorithms proposed in their previous works (Beasley et al. 2000, Beasley et al. 2001). The objective of the optimization model was to minimize the total deviation from the scheduled landing time, indicated as delay. Results showed that an optimal exact solution could be found only for small instances of the problem, while applying the two heuristics they could solve the problem effectively. In the work of Bianco et al. (2006), the aircraft scheduling problem was modeled as a deterministic job shop scheduling problem. Moreover, its dynamic version was solved by implementing a heuristic algorithm. The algorithmic approach was based on a fast local search which included also the constrained position shifting (CPS) concept[1]. The job-shop scheduling ensured the respect of constraints such as longitudinal and vertical separation between aircraft and other trajectories, while with the implementation of the heuristic, the dynamicity of the problem was taken into

[1] CPS technique consists in optimizing the sequence of the aircraft on the runway, ensuring that each aircraft was not deviated more than N shifting positions from the initial First Come First Served (FCFS) sequence.

account. Another relevant work is the one of Balakrishnan and Chandran (2010) where the problem of scheduling runway operations under constrained position shifting (CPS) was solved by applying a dynamic programming approach. The paper presented different algorithms to solve the aforementioned problem, different objectives were considered such as the minimization of the makespan, minimization of the maximum delay over all aircraft and the minimization of the sum of the delay over all aircraft. Moreover, discrete time models were developed in order to deal with more complex scenarios such as aircraft-dependent cost functions and cases where the triangle inequality[2] was violated. The paper extended the methodology also to the case of simultaneous scheduling of arrivals and departures cases.

Hu and Chen (2005), solved the sequencing and scheduling of landing aircraft problem introducing the Receding Horizon Control (RHC) approach. RHC is a N-step ahead search, where a solution for the N-step intervals is taken but it only takes place in the current interval, after that, the same search is repeated for all the other intervals. Results showed that the implementation of this procedure reduced sensibly the computational time and the airborne delay. In the literature, we can find other works that proposed the RHC as a solution approach for aircraft sequencing and/or scheduling problems, Zhan et al. (2010), solved the aircraft sequencing and scheduling (ASS) problem by implementing an ant colony system (ACS); Sama et al. (2013) solved the aircraft scheduling problem by using a branch and bound algorithm; Furini et al. (2015) implemented a rolling horizon approach using a tabu search heuristic to solve the aircraft sequencing problem; in their work they tested different rules for dividing the whole horizon in several windows. Toratani et al. (2015), proposed a receding horizon guidance (RHG) approach to solve the trajectory and sequencing aircraft problem; they solved the problem by proposing a mixed integer linear program. Other works included uncertainty in the sequencing and scheduling problem, like Kapolke et al. (2016), Heidt et al. (2016) and Hong et al. (2017). In Kapolke et al. (2016), the pre-tactical runway scheduling under uncertainty was studied. In pre-tactical planning, decisions for aircraft are taken several hours in advance or up to 30 minutes in advance. Due to this level of abstraction, decision variables of the problem were not about the assignment of the exact arrival times but, instead, they were about the assignment of aircraft into a certain time window of a given size. The uncertainty was included as deviation in the earliest time of arrival windows, the latest time of arrival windows and the maximal latest time of arrival windows. The objective was to minimize the deviation from scheduled times. They adopted three solution approaches: a stochastic, a strict robust, and a recovery to a strict robust. While the strict robust solution appears to be a conservative one, as it represent the most robust solution but obtained at the price of delay/advance time increase, the recovery to strict robust solution proved to be a good trade-off between robustness and conservativism. A similar work was conducted by Heidt et al (2016), in this case they solved the exact runway scheduling problem. They include the uncertainty by considering deviation from estimated time of arrival and departure. The objective and solution approach were similar to Kapolke et al (2016), as they applied a strict robust solution approach and a light version of it by allowing the value of the objective cost function (deviation from scheduled time) to increase by a certain percentage. In

[2] The triangle inequality refers to the case of a sequence of three consecutive aircraft that need to be properly separated. For three aircraft in sequence a, b and c, the separation between the first and last in the sequence has to be bigger or equal than the sum of the separation between the first and the second in sequence and the separation between the second and the last in sequence. Formally is stated as: $\delta_{ac} \leq \delta_{ab} + \delta_{bc} \; \forall a, b, c$.

the work of Hong et al. (2017), we find a dynamic robust aircraft sequencing and scheduling problem. They applied this to the Point Merge System (PMS) technique, considering uncertainty in the flight time of aircraft. The objective function minimizes the time required for aircraft to reach the merge point and the amount of delay during the final approaching path. After having found a solution by applying a mixed integer linear program to a deterministic version of the problem, uncertainty is included in the solution, and then a heuristic algorithm is employed to find a robust solution. In this work a sliding time window approach is employed in order to reduce the computational time. Results confirmed the effectiveness of the method in dealing with uncertainty, however, the experiments were conducted on a small instance of traffic.

2.1.2. Conflict detection and resolution (CD&R) problem

The conflict detection and resolution (CD&R) is another problem that has been extensively studied; in this context, a conflict is detected anytime the separation minima between two consecutive aircraft is violated. The CD&R problem can be applied to airspace operations (en-route and TMA) and also to ground operations (taxiways, runways). Many researchers have focused on en route conflicts resolution, as we found several studies related to this topic such as Allignol et al. (2016), Hong et al. (2016), Lehouillier et al. (2017), and Hao et al. (2018). In many of these works uncertainty was taken into account as pilot intent and navigation (Hao et al. 2018), wind effect and speed imprecision (Allignol et al. (2016), Lehouillier et al. (2017). The solution approaches for these works span from mix integer program (Lehouillier et al. (2017)), mix integer linear program (Hong et al. (2016)), to metaheuristics such as evolutionary algorithm (Allignol et al. (2016)). To the best of our knowledge, few works focused on conflict detection and resolution in the TMA airspace. Jones et al. (2012), proposed a CD&R algorithm for potential conflicts on runway, taxi and low altitude airspace operations. They evaluated the performance of the algorithm by simulating separately the different airport areas, namely runway, taxiway and low altitude airspace. The work proved to be insightful as it revealed that unnecessary maneuvers were operated by aircraft, as they were caused by unwanted alert triggered due to position inaccuracies. However, one main limitation of this work is that it considers a small number of aircraft involved in the operations. In the work of Zuniga et al. (2011), a genetic algorithm has been applied to solve the CD&R problem in the TMA, taking as control decisions the speeds and the trajectories of aircraft. The objective was to reduce the conflicts to zero while minimizing the deviation of the decision control compared to their initial values. Ruiz et al. (2013), developed medium term conflict detection and resolution, where medium term refers to the time-window for making decisions which is between 20-30 minutes. They considered 4D trajectories, and the solution approaches employed a conflict detection algorithm using time based distances and a conflict resolution algorithm based on vectorizations. In Kuchar and Yang (2000), can be found a review of various modeling methods for the CD&R problem.

2.1.3. Airport surface management

Another problem extensively addressed is the surface management. Atkins et al. (2010) presented a review of the main models implemented for solving various airport ground problems, highlighting the main areas for improvements and future research. Pujet et al. (1999) solved this problem by developing a queueing model, testing different control strategies with the objective of alleviating the congestion on the ground. Taxi times were modelled by a probability distribution, and aircraft were held at the gate during congested period in order to

reduce taxi-out times. Simiakis and Balakrishnan (2014) developed a stochastic and dynamic queuing model for the departure process, predicting expected runway schedule and takeoff times, taxi out times and queuing delay. In a similar work, Simaiakis et al. (2014), developed a pushback rate control strategy. The push back rate control strategy consists in identifying the number N of aircraft cleared to push back that corresponds to the maximum runway departure throughput. Beyond this value (N), the departing runway throughput will not increase and, as a consequence, the congestion on the taxiway will increase. The objective of this control strategy is to come up with a push back rate that will not affect negatively the runway throughput and at the same time, that will reduce the congestion on the taxiway. Field tests demonstrated that a good calibration of the control parameter could guarantee reduction in taxi delays and fuel burn. In Evertse and Visser (2017), and Adacher et al. (2017), the airport surface problem was tackled by focusing on environmental aspect, where the objective function was set to minimizing aircraft emissions. This objective is linked with the minimization of ground delays which leads to unnecessary time spent by aircraft with engine on. In Adacher et al. (2017), the solution approach was split into two phases, in the first, taxiway routes are obtained so that delays are minimized, in the second, given the previously calculated routes, the waiting time of aircraft with engine on are minimized. The latter objective is achieved by shifting the waiting time from taxiways to parking areas where engines could remain off. Evertse and Visser (2017) work, is based in a mixed integer linear programming model which is implemented within a time window decomposition approach. In the work of Kang and Hansen (2018), the impact of surface congestion management (SCM) on airlines' scheduled block times (SBT) was evaluated. SBT is the time in between an aircraft leaves the gate for departing from its origin airport and the time when it arrives at the airport of destination and parks at the gate. The main findings of this work revealed that by applying SCM, SBT could be reduced as a consequence of reductions of taxi out delays. SCM, in this context, was conducted by developing a predicting model by using historical data regarding taxi in/out times and flight times.

Some works focused on the taxi out process with the objective of reducing the congestion on the airport surface. Khadilkar and Balakrishnan (2014) proposed a control algorithm based on the aircraft entry times in the network, trying to minimize the taxi out times of aircraft. The algorithm tried to balance the congestion on the surface while, at the same time, keeping a good throughput performance. In their model, they used actual surface surveillance data and they modelled the travel time on the links of the taxi network in a stochastic way.

In Bohme (2005) it was presented the DMAN concept. DMAN is a tactical departure management tool that was developed by EUROCONTROL and DLR. Its objective is to predict the take off time and to derive from it, the pushback time and/or the engine on start-up. DMAN architecture is based on a modular approach which enables to model different airport scenarios; it includes data-based models for operations, so it can be applied to different airports. The benefits coming from DMAN application are delay reduction, better environmental impact, harmonization of the traffic flow, improvement in the prediction and reliability of schedules. A mixed integer linear program model was presented in the work of Gupta et al. (2009) for the deterministic runway departure scheduling. The model showed good improvements when compared with a FCFS queuing rule and also a fast computational time, although, it has been tested on small samples of aircraft (15 to 20) and for a time horizon of 15 minutes.

Regarding the problem of taxi routing, we can find relevant works that used mainly mixed integer linear program models. Smeltink et al. (2004) developed a mixed-integer program for an optimal taxi scheduling, where for each aircraft were calculated the time for crossing a specific node of the taxiway network. The model considered constraints about minimum and maximum speed on the links, capacity on holding positions, and separation minima between two consecutive aircraft. The objective was to reduce the conflicts and to minimize the sum of taxi times for all aircraft. Due to the size of the problem, the solution approach consisted in a rolling horizon approach where a window of 30 minutes was used for each optimization. A similar work is the one of Roling and Visser (2008), where the taxi process was modeled as a mixed-integer linear programming model, with the objective of minimizing taxi time and holding times. The main difference between this model and the previous one is that it considered taxi rerouting and holdings, and it included also some sources of uncertainty in the expected landing time and pushback time. The work of Montoya et al. (2009) extended the taxi scheduling and routing problem to a multiple route taxi scheduling problem, where besides deciding for each aircraft at which time to be on a certain node, they added an extra decision about the route to take within a predefined set of alternative routes. The problem was solved by a mixed-integer linear program model, and two objective functions were implemented into one objective that minimizes the taxi time and another one that maximizes aircraft departure throughput. In Zou et al. (2018), a two stage scheduling strategy for multiple runways model was developed. The novelties of this work are the inclusion of runway exit availability (REA) in the model, so that it could be decided to which runway exit a landing aircraft would take; and the block-and-hold decision, which allows aircraft to hold along taxiway segment and not only on taxiway intersections. The problem was solved by an iterative integer programming model, with the objective of minimizing the total taxi time of the flights.

Some studies focused on the integration of ground operations such as runway and surface operations together, as in the work of Deau et al. (2009) where it is stressed the connection between runway operations (departure manager-DMAN) and surface operation (surface manager-SMAN). In their tests, they found out that the ground delay performance when considering only runway operations is less impacted when considering also constraints coming from the ground operations. In their work, they highlighted the benefits from the integrated implementation of these operations and also the complexity derived from them. In the work of Benlic et al. (2016) an integrated runway sequencing and taxiway routing problem was proposed. The applied a receding horizon approach and solved the problem by applying a local search heuristic. A follow up of this work is given in Brownlee et al. (2018), where they addressed the same problem by including uncertainty. Uncertainty was included by an adapted fuzzy method to estimate taxi times, and the problem was solved by the quickest path problem with time windows (QPPTW) algorithm. Results revealed that, although taxi times increased due to a more conservative solution, taxi delays have been reduced compared with a solution obtained by a QPPTW algorithm without uncertainty included.

2.1.4. Integration of airspace and ground operations

Bertsimas and Frankovich (2015), tackled the problem of unifying TMA and ground operations. They divided the integrated problem into two phases, where in a first phase they took decision about runway configuration and aircraft sequencing, and in the second phase decision regarding ground operations such as gate-holding and taxi routing. They solved both problems using a

binary optimization model, achieving good results in terms of computational performance and operational performance. One limitation of this work was that they used deterministic data, and therefore, they were not considering the uncertainty that affects the problem. Another limitation resides in the fact that all the decisions were taken inside the same time frame, which in reality might be unfeasible due to the nature of the different operations. To be more precise, airspace and ground operations require a different decisional time frame, since airspace decision are more tactical (runway configuration, aircraft sequencing) and ground decision are more operative (gate-holding, taxi routing). In the work of Bosson et al. (2015), taxiway and runway operations were integrated with an arrival and departure operations in the terminal airspace. In their model, they modeled uncertainty as pushback delay and arrival gate delay; the objective of the problem was to maximize on-time performance by reducing gate waiting time and the total travel time in the air and on the surface. The problem was modelled as a mixed-integer linear program and solved by a an algorithm-based scheduler. In Guepet et al. (2017) the same problem was tackled by proposing a heuristic sequential approach where first, the earliest take off times and arrival time at the stand were estimated, second, the take off sequence was created based on the takeoff times given by the first step, and third, aircraft were routed on the taxiway based on the sequence given by the second step. Another contribution of this work was that it considered in the operation also conflicts at the stand area. Sama et al. (2017), addressed the problem of coordinating ground and airspace operations, by developing and solving an mixed integer linear programming model and by applying two different heuristics: one based on a branch and bound algorithm, and another one based on a fast greedy algorithm. They tested different policies for air traffic controllers for landing and departing aircraft, and assessed different indicators such as: runway throughput, approach time (landing), air traffic controllers' workload and taxi time. The authors stressed the importance of coordinating airspace and ground operations in order to avoid inefficiencies and to improve overall performance. Serhan et al. (2018), tackled the same problem but the put focus on airline and passenger delay cost in their optimization model. They solve the problem by applying a mix integer nonlinear programing model and swap position violating aircraft heuristic model. In their work they evaluated the impact of changing cost parameters on flight schedules, delays and runway makespan. Badrinath et al. (2019), developed a queuing model for the ground operations and extended it to the departure phase, by incorporating airspace departing routes. The queuing model was able to predict congestion in different areas of the ground such as ramps, taxiways and runways. In their model, they considered different factors in their model such as: runway configurations, weather and en route traffic.

2.1.5. Gate assignment problem

The gate assignment is another problem that was studied by researchers. In this problem, the main objective is to make a robust assignment, where robust assignment is defined as the capability of making the gate assignment insensitive to variations in flight schedules (Bolat, 2000). One effective way of making the gate assignment robust is to maximize the idle time between two consecutive aircraft assigned to the same gate. Bolat (2000), solved this problem by formulating a mixed-quadratic binary model with the objective of minimize the variance of the idle time at gates. The model was solved by applying a branch and bound algorithm and then with different heuristics for bigger instances of the problem. The heuristics were based on dynamic priority functions to guide the gate assignment. Ding et al. (2005), solved the over-constrained gate assignment problem, where we have more aircraft than gates. The first

objective is to minimize the number of aircraft non assigned to a gate, and the second objective is to minimize the passenger' walking distance inside the terminal. They solved the problem applying first a greedy algorithm to minimize non assigned flights, and then employed two heuristics for improving the second objective. The two heuristics are the simulated annealing and a hybrid simulated annealing with tabu search approach. A similar objective was set in the work of Dell'Orco et al. (2017), where the minimization of assign aircraft to remote gates was proposed together with the minimization of the passengers' walking distance. The problem was solved by the use of metaheuristic, the fuzzy bee colony Optimization (FBCO). In Kim et al. (2011), the robust gate assignment was solved by setting as objective, the expected level of disturbance. In this framework, the level of disturbance was defined as the time overlap due to delayed departure or earlier arrivals at the gate, for two consecutive aircraft that are assigned to the same gate at different times. The problem was modelled as a linear mixed 0-1 problem and was solved optimally and also by the implementation of a tabu search meta-heuristic. Genc et al. (2012), focused on the maximum gate employment, which aims at maximizing the time spent by aircraft at gates (increase gate utilization), and minimize non assigned aircraft to gates. They developed a method that uses heuristics in combination with a stochastic approach. With the former, they can obtain a solution in a fast time, while with the latter they can improve the quality of the solution. Van Schaijk and Visser (2017), developed a robust flight-to-gate assignment with the objective of minimizing the occurrences of re-planning due to flight uncertainties, They included in set of the constraints a new stochastic one, which defines that the probability of having two aircraft scheduled at the same gate and at the same gate does not exceed a certain value. A binary integer programming model was developed in order to solve the problem. Gate assignment robustness was evaluated also in the work of Zhang et al. (2017), where they estimated the probability of having conflicts at gates, analogously as van Schaijk and Visser (2017). They solved the problem by applying the biogeography-based optimization (BBO) algorithm. In the review of Dorndorf et al. (2007) there is a large variety of mathematical models and solution approaches for the gate assignment problem.

The topic of this thesis covers the integration of airport airspace and ground operations. Similar to the work of Bertsimas and Frankovich (2015), Sama et al. (2017) and Serhan (2018), the operations involved are the ones in the TMA airspace and the ones on the ground. The main difference with their work is that operations are not split in two phases as in Bertsimas and Frankovich (2015), but they are integrated. Moreover, the two phases, airspace and ground, are not modeled as microscopic models as in Sama et al. (2017) and Serhan et al. (2018), but as a macroscopic model. For instance, in the two-phase model of Bertsimas and Frankovich (2015), in the first phase, decisions are generated for the runway configuration selection and aircraft sequencing in the TMA, while in the second phase decisions are generated for the ground. In this thesis, the operations were integrated and all the decision were taken within the same time frame. The main limitation of the work of Bertsimas and Frankovich (2015) is linked to the way they considered the decision making time frame, for airspace and ground. Considering that tactical decisions for the aircraft sequencing in the TMA can be done even two hours in advance, it is clear that within this time frame it is not relevant to know in detail what happens on the ground due to the high uncertainty. On the other hand, if operative decisions such as aircraft taxiing must be made, a smaller time frame, even five minutes in advance should be applied so that there is less uncertainty. These two different type of decision making, tactical and operative, and the different time frame considered, can help defining two abstraction levels for modelling

airport operations, having macroscopic and microscopic level (Sama et al. (2017) and Serhan et al. (2018)), respectively. In this thesis, a macroscopic model was developed for which both decisions for airspace and ground were made within the same time frame, and the same level of abstraction was implemented. In this macroscopic level of abstraction, the performance of the ground are not evaluated considering flight-by-flight, but by considering the performance of the overall system. In this context, ground operations are modeled in low detail, by identifying the main ground components and considering them as resources characterized by limited capacity. The main objective here becomes the mitigation of the overall ground congestion. In this way, uncertainty related to ground operation can be reduced compared to Bertsimas and Frankovich (2015), and a more consistent and realistic unified approach for airspace and ground operations can be implemented. Moreover, another gap found in Bertsimas and Frankovich (2015) was that the operations were modeled in a deterministic way, by not addressing the uncertainty related to these kinds of operations. This thesis tries to close this gap by including different sources of uncertainty by coupling optimization techniques together with simulation techniques. The problem tackled in this thesis, falls also in the category of the CD&R problem, since the main objective of the problem is to find a sequence of aircraft in the TMA without any conflict, however, in this thesis, the concept of conflict has been extended also to the ground operations by defining ground conflicts as well.

In general terms, the studies reviewed so far presented the following assumptions/limitations: some of them considered only small instances of the problem, and this is due to the complexity of the problem for bigger instances; some of them considered the problem as deterministic. These assumptions represent limitations for the application of these models to real world instances, this fact motivates the research of this thesis to focus on overcoming these limitations.

2.2. Simulation approaches for airport operations

The progress of technology regarding computer science has allowed researchers and practitioners to develop advanced simulation models that serve as support in training, decision making, planning, and scenario evaluation for strategic, tactic and operative operations in many different fields. As evidence of this trend, Brunner et al. (1998), pointed out that the use of simulation in transportation increased dramatically, and in their publications, they included a review regarding opportunities and barriers for its further development. Concerning the application of simulation techniques in the aviation field, and specifically to airport operations, many airport simulation models were developed through the years. Many of these models represent specific operations and areas of airports such as airspace, airside and terminal operations. Recently, research efforts have focused on the integration of some of these airport operations in order to take into account also their interactions, which, on the one hand permits to make a more precise and reliable analysis, but on the other hand, it increases the complexity of the model. In the report of Odoni et al. (1997), there was an extensive survey on the existing airport simulation models. In their report, they made a classification between airport models, which was based on the level of detail and purpose. They classified airport simulation models into: macroscopic (strategical), mesoscopic (strategical) and microscopic (tactical). A more recent review of simulation application for airport system was made by Ran et al. (2013). In this review they highlighted the potential developments and shortages for simulation applications. In their work, they concluded that simulation models for airport systems should integrate operations from different areas, such as terminal and airfield, and do not limit the scope to the

analysis of specific areas. Moreover, there is a need of a more user friendly software interface and the development of virtual reality to make the simulation closer to reality and to improve the man-machine interaction.

Another classification of simulation models is made based on the type of simulation software used, having models created with specific purpose simulation software or with general purpose simulation software. The first category implies that the model is developed and used only for a specific field of interest. General purpose simulation software, on the other hand, are constituted by a simulation engine, it can be discrete event simulation (DES) or agent based simulation (ABS), and include some predefined general objects that can be used by the developers to create any kind of model for any kind of field of interest. DES is a simulation paradigm where time passes based on specific events, and not based on fixed time intervals. In this paradigm, the state of the system changes based on the occurrences of these specific events. ABS is a simulation paradigm, where time passes according to a specific time interval and the entities (agents) are built with their own logic, so that the system state changes based on the actions and interactions of the agents. Usually, models for airspace and airside are simulated using DES due to the fact that in these models, the sequence of actions for the entities (events), in this case aircraft, is already defined (e.g. airspace routes, taxiway routes). Models for airport terminals, where passengers are the entities simulated (agents), generally use ABS because the models need to recreate the different logic of each passenger that behave according to his own plan and also according to the interactions that he/she has with the surrounding environment. Generally, building a model by using specific purpose simulation software is easier due to the fact that the simulation framework offers already the objects needed for the specific type of operations; on the other hand, building the same model by using a general purpose simulation software might be more difficult and time consuming due to the fact that the developer needs to create all the objects and logic from scratch. General purpose simulation software are more flexible because the developer has more freedom in building the system logic, while for specific purpose simulation software, the developer is limited to the properties and functionalities of the predefined objects included in the software. However, nowadays, many specific simulation software are being developed as open source, in such a way that developers are free to get access to the code behind the software and be able to extend them based on their requirements. In table 2.1 the main advantages of using general-purpose and specific-purpose are summarized.

In the next two sections, a literature review of airport simulation models using general and specific purpose simulation software is presented.

2.2.1. Specific-purpose simulation applications

One of the most relevant and used airport modeling tool is SIMMOD, developed by the FAA. SIMMOD is a simulation software that allows to represent and model the operations of airport airspace and ground side. It is classified as a microscopic airport simulation model, since it models the airport operations in detail (SIMMOD PRO, 2018). SIMMOD has been widely used in many applications and studies (Delcaire and Feron, 1997), (Kleinman and Hill, 1998), and (Wei and Siyuan, 2010). Other airport simulation software that can be found on the market are: FACET (Bilimoria et al., 2000) and BlueSky (Hoekstra and Ellerbroek, 2016), focusing mainly on airspace operations, AirTop (Transoft solutions, 2018) and CAST (Airport Research Center, 2018) which include different modules for modelling airspace, airside and terminal operations,

Table 2.1. General-purpose vs specific-purpose simulation software

	General-purpose	Specific-purpose
Field of application	Any field	A specific field of interest, e.g. logistics, manufacturing, etc.
Ease of use	More difficult to use since the developer has to create its own objects and its own model variables. Often programming is required.	Models are usually created by dragging and dropping predefined objects. Often programming is not required.
Modeling flexibility	Highly flexible, since it offers general object libraries that can be used to model any type of process.	Low flexibility due to the fact that there are already predefined objects with certain characteristics that cannot be changed.

TAAM (Jeppesen, 2018) and RAMS (Innovation for Sustainable Aviation, 2018) that simulate airspace and airside operations and AirSim (Productivity Apex, 2018) that is limited to the modeling and simulation of passenger flow inside airport terminal. Some of these airport simulation models are based on DES or ABS. Other airspace simulation model applications can be found like in the work of Zheng et al. (2015), where an ABS model was developed to simulate the main actors in the air traffic management such as aircraft, air traffic controller and ATC automation system. In their model, the interactions between the agents were modelled such as communications, surveillance and instructions and it results showed the ability of the model to work as a planning tool for air route network, sectors and flight plans, and for evaluating the ATC workload. Klein (2017) proposed a simulation model to evaluate the impact of the weather for strategic and tactical airspace applications. Based on different weather forecast scenarios, they were able to give instructions to aircraft for trajectory change in order to minimize ground and airspace delays, cancellations and diversions. Simulation models for ground applications can be found in the work of Chua et al. (2014), where a simulation model of taxing operations was developed within the framework of the project Modern Taxiing. The model was used as decision support system to assist ground traffic controllers in taxiing path suggestions and conflict detection. Kern and Schultz (2016), developed a model for evaluating the capacity of a standard single runway considering different arrival-departure rates, aircraft mix, and runway infrastructure.

2.2.2. General-purpose simulation applications

In the literature, we can find many studies about the modeling of airport operations that are developed by using general-purpose simulation software; in the following paragraph the most relevant are mentioned. In the work of Zuniga et al. (2013), a DES model was developed using the Colored Petri Net CPN formalism. In their model, they simulated the airport TMA with the objective of detecting and resolving conflicts (CD&R). The CD&R was based on an algorithm able to give instruction about trajectory change (heading angle), speed change and vertical manoeuvers. Ground operations were also subject to the development of simulation model, for instance, we can find many taxiway routing models such as the one presented in Khoury et al. (2007) where they developed a DES model of the airside operations (runway and taxiway). In

their work, they highlighted the importance of using visual tools such as simulation models, in order to evaluate the performance of airport airside operations and also in planning phase, moreover they highlighted the use of general-purpose simulation software as tool for modeling such operations. Runway operations were modelled in Martinez et al. (2014), where they proposed a DES model for runway operations using a general-purpose simulation software, with the objective of estimating runway capacity and delay. In other studies, turnaround operations were analyzed by using of modeling and simulation techniques, such as: Wu and Caves (2004), Adeleye and Chung (2006), and Norin et al. (2012). In Wu and Caves (2004), a simulation model based on Markovian and Monte Carlo simulation was developed. The Markovian simulation combined with Monte Carlo Simulation allowed the authors to deal with uncertainties coming from the schedule and also from the turnaround operations themselves. In Adeleye and Chung (2006), turnaround operations were simulated, and the sources of delay were analyzed and classified in terms of maintenance, logistical and operational. They used the ARENA (ARENA, 2018) general purpose simulation software to model and simulate the turnaround operations, the same simulation software has been used in the work of Norin et al. (2012), where they tested different scheduling alternatives for de-icing equipment and evaluated their impact on the overall airport performance in term of delays and waiting times. In the work of Scala et al. (2017), a DES model was developed by using the general-purpose simulation software SIMIO (SIMIO, 2018). In their work, they proposed a modular approach to simulate airspace and airside airport together. They built four different modules: airspace module; runway module; taxiway module; and turnaround operations module. One of the main advantages deriving from this approach is that the different modules could work separately or together, depending on the objective and the type of analysis to be conducted.

In this thesis, the author has developed a simulation model using a general-purpose simulation software based on DES. The simulation model was built according to the macroscopic level previously introduced, furthermore, it works in together with an optimization model for evaluating the performance of the solution obtained from it. In the simulation model, TMA operations were modeled similarly as the work of Scala et al. (2017), making sure that all the main rules about speed, altitude, and separation minima between aircraft were properly implemented. An additional feature was the development of conflict detection within the simulation model, allowing it to detect not only conflicts in the TMA, similarly as Zuniga et al. (2013), but also detecting conflicts for ground operations. Another aspect that distinguishes the simulation model developed in this thesis from the previously mentioned models is the inclusion of uncertainty, which allows the evaluation of the effect of uncertainty in the operations.

2.3. Optimization-Simulation solutions for airport operations

The combination of simulation together with optimization techniques has been proven to be a good approach for tackling problems were uncertainty plays a crucial role. Simulation on the one hand, allows to represent the uncertainty that affects the system under study that is often not possible by using analytical models, on the other hand, optimization allows to achieve an optimal or near-optimal solution even for complex systems. When coupling the two techniques, models can benefit from their advantages, leading to find solutions that are more feasible when applied to real-world instances. In the work of Odoni (1994), it is highlighted the importance of considering uncertainty when modeling airport and air traffic control operations. He classified uncertainties based on the time horizon in long-, medium-, and short-term. For long-term

scenarios the uncertainty that needs to be considered is related to the demand, while for medium- and short-term, the uncertainty is represented by delays, cancellations and weather conditions. In order to cope with uncertainty, he advised the use of probabilistic concepts in the planning, design and management of infrastructures and operations. Moreover, he points out that in the future, the development computer-based models with visualization features will increase and will be beneficial for training and evaluating purposes.

In the last years, many applications were developed using simulation and optimization combined in many fields, as a reference, in the book of Mujica and Flores (2017), various examples can be found including aeronautical, logistics and industrial case study. In the aviation field, concerning airport operations, we can find applications of simulation combined with optimization, and especially in the recent years this approach has been exploited extensively. Kleinman et al. (1998) tackled the airport network delay problem by using a simultaneous perturbation stochastic approximation (SPSA) optimization algorithm in order to have an optimum ground-holding policy, with the objective of minimizing cost of delays in the air and on the ground. The simulation part was implemented by using SIMMOD simulation software, being able to measure in more detail the performance of the system in terms of ground and air delay. However, in this work, a synthetic traffic and airport layout were considered, while airport ground and airspace operations were simplified by including many assumptions. In the work of Confessore et al. (2005), an airport ground simulation model was developed for simulating taxiway operations, including an optimization module for the minimization of the ground delay. In this context, the simulation and optimization module could communicate between each other and update information in order to address the problem in a dynamic way. The simulation is based on a DES model using a general-purpose simulation software called ARENA, while the optimization module is based on an algorithm for solving the shortest path problem. In the PhD thesis of Lee (2014), airport surface optimization problem was tackled and uncertainty sources such as pushback time, runway exit times, taxi speeds, were included by simulating the operations using SIMMOD simulation software. Mujica (2015) proposed a methodology for the optimization of the check-in desk allocation inside airport terminal, with the objective of improving the level of service (LoS) for passengers. The optimization part was based on a genetic algorithm, including different operational restrictions, while the simulation part was based on DES using the general-purpose simulation software SIMIO. The simulation part allowed to take into account some stochastic aspects like passenger arriving profile, opening times and physical facility configurations. In Mujica et al. (2017) an optimization search algorithm, OptQuest (OptTek, 2018), was embedded in a DES model (SIMIO) for improving the turnaround performance at an airport in terms of time and vehicles utilization. In their work, airport ground and airspace operations were modelled in an integrated way, having airspace arriving and departing routes, runway, taxiway and turnaround operations. The methodology included the use of simulation, a statistical approach, and an optimization search algorithm. Simulation was used for the evaluation of the turnaround performance, while with the application of the statistical technique Analysis Of Variance (ANOVA), it was possible to identify the main components impacting the turnaround performance. Based on the identification of these impacting components, the state space was limited to a smaller instance, so that the optimization search algorithm could efficiently find an optimal solution.

Similarly to the aforementioned studies, in this thesis, simulation techniques are coupled with optimization in order to take advantage of their characteristics and to be able to tackle the stochastic aspect of the problem. The contribution of this thesis to this regard, is the evaluation of the robustness of a solution obtained by making it more resilient to perturbations. Moreover, in the thesis were developed two different ways of exploiting simulation and optimization in order to test their effectiveness and efficiency. In the first method, simulation was used as a validation tool to test the feasibility of the optimized solution and then it was attempted to improve the robustness of the optimized solution by tuning some input parameters. In the second method, simulation and optimization were fully integrated, establishing a direct communication between the two, working as a feedback loop where the robustness of the optimized solution is continuously evaluated and then improved.

In the next chapter, the problem addressed in this thesis and the general solution approach implemented will be introduced.

Chapter 3

Problem description, modeling and general solution approach

In this chapter, the problem tackled in this thesis will be described, together with the general approach employed for solving it. The operations involved will be described in detail, considering the assumptions made for modelling them. A description of the optimization model will be given as well, including decision variables, objective function and constraints. A section will be address the uncertainty that has been considered in the problem. The second part of this chapter focuses on the general approach employed for solving this problem, which is a framework combining an optimization model with a simulation model. In the last part of the chapter, two methods that combine optimization and simulation for solving the problem are described.

3.1. Airport operations at a macroscopic level

This thesis deals with the optimization and simulation of airport operations at a macroscopic level. One of the contributions of this thesis is the airport operations modelling at a macroscopic level by integrating both airspace and ground operations together. To the best of the author knowledge, no one has dealt with this before. In the study of Bertsimas and Frankovich (2015), the airspace and ground operations were unified, however, implementing a two-phase optimization approach, where tactical decisions were taken first followed by operative decisions. In order organize the operations in an integrated way, the main challenge is about the choice of the planning time-frame and consequently of the level of abstraction for modeling the operations. If we consider this problem from an ATCO perspective, decisions regarding aircraft sequencing in the airspace can be made around two hours in advance, however, planning ground movements at this stage would be too early and not effective, since uncertainty would considerably affect such operations. Therefore, for planning purposes, airspace and ground operations should consider a different time-frame, which explains why, in this thesis, airspace and ground operations are modeled using two different levels of abstraction, implementing a macroscopic level approach. In this context, macroscopic level means that the main airport ground components are identified, such as runways, taxiway network and terminals, and modelled as resources with specific capacities. Airspace operations, on the other hand, are identified as the ones related to the TMA, such as aircraft sequencing along the STARs, and merging in the final approach segment before landing to the runway. In the next sections these operations, and the way they have been modeled, will be described more in detail.

3.1.1. Operations description and modeling

In this thesis, the focus is put on ATCOs operations, specifically, regarding the planning activities at a tactical level. Planning at tactical level, means to consider operations within a time-frame ranging from one day to few hours before the real scheduled operations. Tactical decisions consider changes in trajectories, gates, landing/departing runways. At a tactical level, decisions concerning airspace operations such as trajectory change, route change, holding etc., can be planned by ATCOs for the following two hours time-frame. Ground operations such as taxiway routing, clearance for leaving the gate or runway crossing, need a short planning time

frame. For example, planning two hours in advance taxiway routing and/or runway crossings, would be too early, because there would be too much uncertainty involved. For these reasons, in this thesis, the integration of airspace and ground operations has been done by implementing two different level of abstractions, coming up with the modeling of airport operations at a macroscopic level. Thus, modeling airport operations at a macroscopic level consider, on the one hand, airspace operations like aircraft sequencing at flight-by-flight detail. A more detailed analysis is done by including components such as fixed landing routes in the TMA (STARs), speed, altitude and separation between aircraft. The airspace landing routes (STARs), are modelled as a network of links and nodes. Each node represents a waypoint of the route, and each link connects two adjacent waypoints. In Figure 3.1, there is a 2D representation of the waypoint network, taking as example the landing routes at Paris Charles de Gaulle when the direction of the runways is toward west (west configuration).

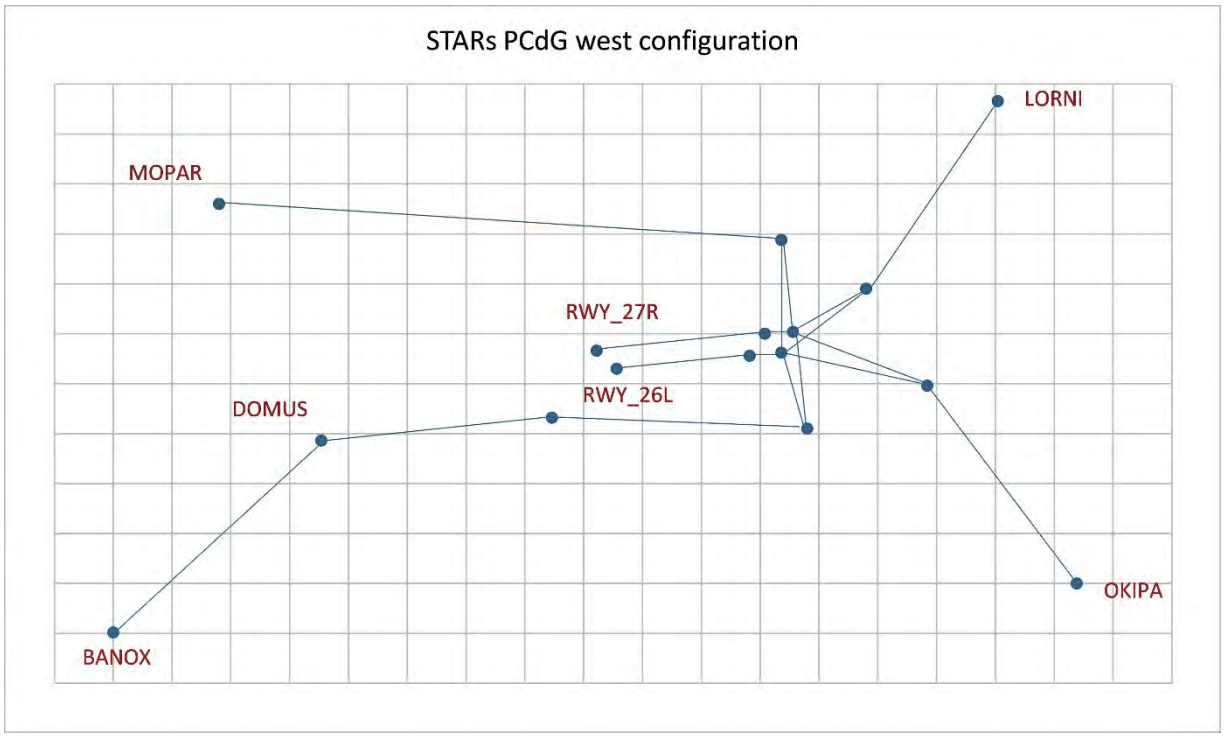


Figure 3.1: Airspace network modeling (2D): STARs of Paris Charles de Gaulle airport, west configuration

Within this macroscopic approach, ground operations were modeled in a low detail, considering the main components of the ground like runways, taxiway network and terminals as nodes characterized by a certain capacity and an occupancy time (landing/departure occupancy time for runway components, taxi in/out time for taxiway network components, and turnaround time for terminal components). In Figure 3.2 there is a representation of the ground components as they were modelled, while Figure 3.3 shows a schematic representation of the integration of airspace and airside operations within the macroscopic approach.

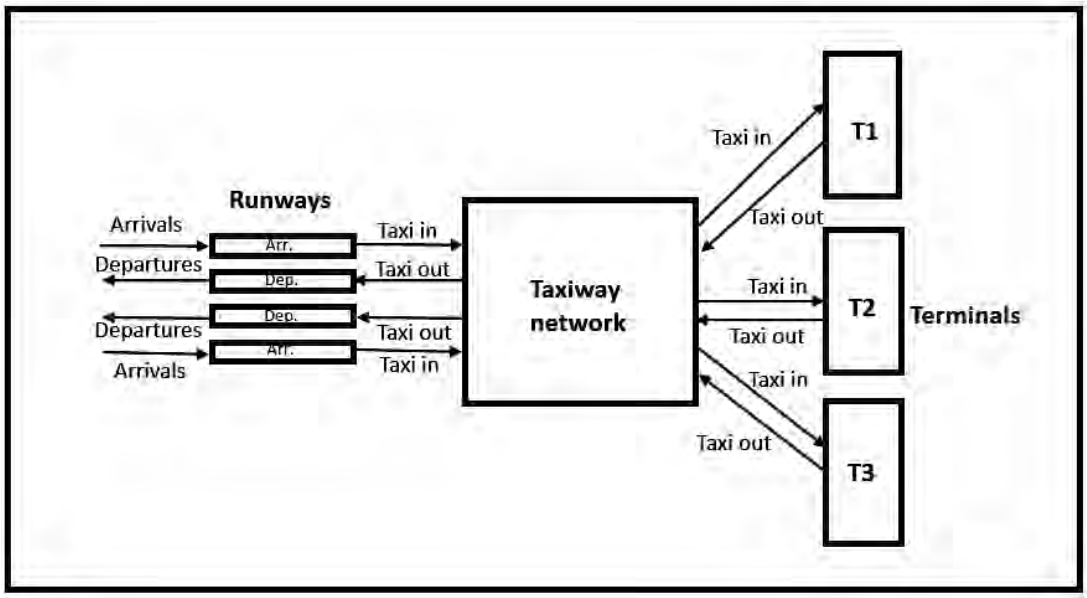


Figure 3.2: Low detail modeling of the airport ground components.

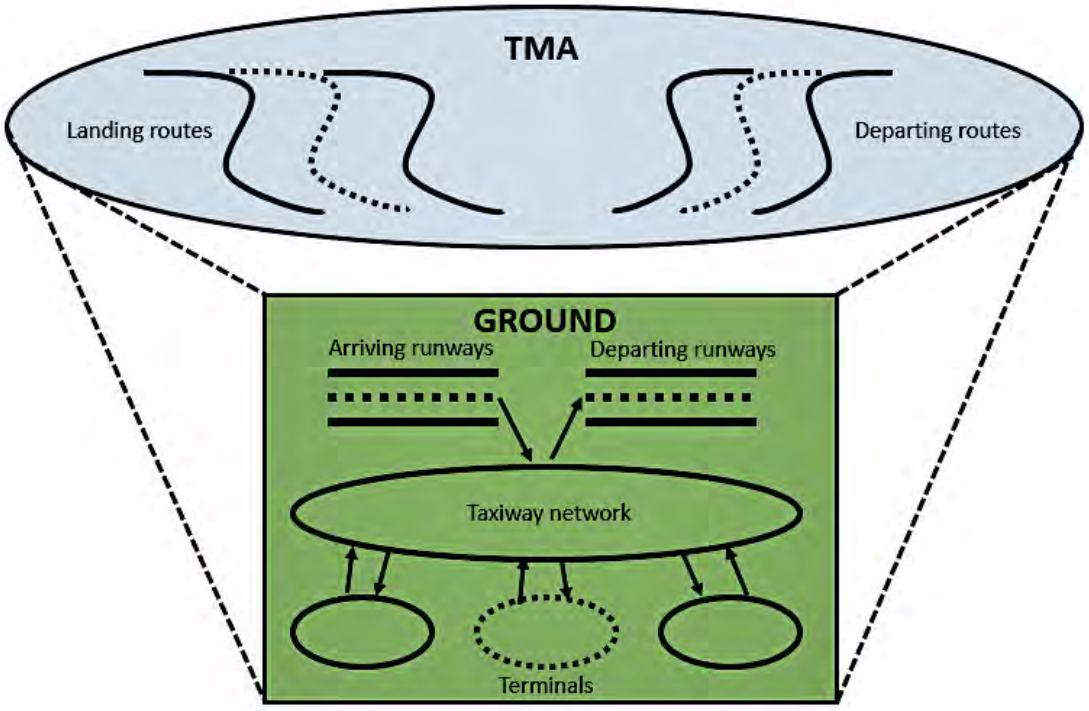


Figure 3.3: Schematic representation of the macroscopic approach for integrating airspace and ground operations.

In this model, flights are the entities which are tracked and manipulated by changing some decision variables. Flights are classified into three different types: Flights arriving only (ARR), flights departing only (DEP), and flights arriving and then departing (ARR_DEP). The first type are those flights that usually arrive late in the evening at the airport and stay overnight, or flights

that need to stay in the hangar for long maintenance; the second type are the ones that are already at the airport from the night before and they are usually scheduled during the morning; the third type of flights are regularly scheduled for arriving and then departing during the same day.

Based on the flight type, the entities of our model will get different input data and will simulate the different operations by undergoing different logic and constraints. Flights of arrival (ARR) type, will start by entering one of the arrival route in the TMA, and continue their flight course according to an entry time in the TMA, an entry speed in the TMA and by choosing the landing runway. While flying toward the runway, they will undergo logic such as speed control, acceleration (negative), vectoring and they will be subject to constraints such as separation minima between any previous (leading) and/or following (trailing) consecutive aircraft. Flights of departure (DEP) type, will start their operations from the moment when they leave the gate, and will continue taxiing out toward to departing runway. These operations are done exclusively on the ground, and due to the different level of abstraction implemented, these operations will be modeled with low detail. In other words, taxiing operations will be modelled only by the average taxi time duration, as well as the departure operations from the runway, which is modelled by its average runway occupancy time. Factors such as speed and taxiway and runway layout are not taken into account due to the level of abstraction chosen. Instruction given to the aircraft of departure (DEP) type, are the pushback time (the time when aircraft are cleared to leave the gate), and the choice of the departing runway. The main constraint is represented by the value of capacity given to terminals, taxiway network and runways components, which represents a threshold up to where the system can still guarantee smooth operations, and beyond whose level congestion situations will be identified. Flights of arrival and departure (ARR_DEP) type, undergo the same logic, constraints, and receive the same instructions and data input as the previous two types, additionally they are involved in the turnaround operations. However, due to the level of abstraction chosen, turnaround operations will be modelled only by considering the turnaround time, which is given by the real flight schedule. In table 3.1 there is a summary of the components and data input/instructions in which the different flight types are involved.

3.1.2. Model assumptions

In order to develop a model able to represent airspace and ground operations in an integrated fashion, some assumptions have been made. Assumptions are classified into airspace and ground operations. Next, a description is given for each of them.

- Airspace assumptions:
    - Aircraft fly along fixed standard arrival routes (STAR), these STARs are published routes and they are assumed to have a proper vertical separation between each other.
    - The departing routes, standard instruments departure routes (SID), are not considered, assuming that once aircraft take off, they will take one of the SIDs without incurring in any congestion/conflict.
    - Aircraft flying their arrival routes, fly starting with an entry speed and then they slow down at a fixed acceleration (negative) rate.
- Ground assumptions:

- Runway, taxiway network, terminals components are modelled as nodes of a network characterized by a certain capacity (number of simultaneous aircraft able to occupy each component: taxiway network, runway and terminal) and a service time (taxi and runway occupancy time, and turnaround time at the gate).
- Aircraft are assumed to cross the taxiway network (in, out), with an average taxi (in, out) time, based on the combination terminal-runway.
- Aircraft are assumed to cross the runway (landing, departing), with an average runway occupancy time.
- Aircraft can park at any gate of the assigned terminal.

Table 3.1. Summary of the main components and data input/instruction given to the flights based on their type.

		Flight type		
Airport area	**Input data/Instructions**	**ARR**	**DEP**	**ARR_DEP**
Airspace (TMA)	Arrival route (STAR)	Yes	No	Yes
	Entry time in the TMA	Yes	No	Yes
	Entry speed in the TMA	Yes	No	Yes
Runway system	Landing Rwy choice	Yes	No	Yes
	Departing Rwy choice	No	Yes	Yes
Taxiway network	Taxi In time	Yes	No	Yes
	Taxi Out time	No	Yes	Yes
Terminal/Gate	Terminal number	Yes	Yes	Yes
	Pushback time	No	Yes	Yes
	Turnaround time	No	No	Yes

3.1.3. Optimization model

In this section, the optimization decision variables, objective and constraints are presented:

3.1.3.1. Decision variables

In the optimization model, some of the ATCOs common instructions are considered as decision variables. In this section, these decision variables are listed and described, explaining their meaning under an ATC point of view and also defining them using a mathematical form.

- **Entry time in the TMA.** The entry time in the TMA is the time instructed by ATCOs for aircraft to reach a specific node (waypoint) of the landing route (STAR). ATCOs can give a delay or an advance to aircraft based on the initial flight plan depending on the level of congestion in the airspace. Delays are achieved by slowing down, or by speeding up the aircraft in the en-route airspace. When aircraft are requested to speed up, it results in a very expensive maneuver in terms of fuel burn, therefore, airlines usually tend to

avoid it as much as possible. Due to this reason, the intervals of delays and advances are not symmetric, meaning that it can be given a bigger delay than an advance. Formally, given a set of aircraft $f \in A \cup AD$, where A is the set of arrival flights and AD is the set of arrival and departing flights, the entry time in the TMA is defined as $t_f \in T_f$, where $T_f = \left\{T_f^0 + j\Delta T \mid \frac{\Delta T_{min}}{\Delta T} \leq j \leq \frac{\Delta T_{max}}{\Delta T}, j \in Z\right\}$. ΔT_{min} and ΔT_{max} are the extremes of the decision interval, representing the maximum advance and maximum delay, respectively, while ΔT is a discretized time increment. For this problem, the values of ΔT_{min}, ΔT_{max}, and ΔT are -5 minutes, +30 minutes and 5 seconds, respectively.

- **Entry speed in the TMA.** The entry speed in the TMA is a speed instructed by ATCOs for aircraft to be reached at a specific node (waypoint) of the landing route (STAR). In this case, as it was the case for the entry time in the TMA, ATCOs can request the aircraft to speed up or to slow down, based on the circumstances. Formally, given a set of aircraft $f \in A \cup AD$, where A is the set of the arrival flights and AD is the set of the arrival and departing flights, the entry speed in the TMA is defined as $s_f \in S_f$, where $S_f = \left\{S_{fmin} + j\Delta S \mid |j| \leq \frac{(S_{fmax} - S_{fmin})}{\Delta S}, j \in Z\right\}$. S_{fmin} and S_{fmax} are the extremes of the decision interval, representing the minimum slowing down rate and the maximum speeding up rate, respectively, and ΔS is a discretized increment. For this problem, the value of S_{fmin}, S_{fmax} and ΔS are $0{,}9S_f^0$, $1{,}1S_f^0$, and $0.01S_f^0$, respectively.
- **Landing runway choice.** At big airports, the runway system can be constituted by several runways. Arrival routes (STARs), are often assigned to a specific runway or to a set of specific runways, then ATCOs can instruct aircraft to land on a specific runway in order to balance the traffic on the arrival routes and specially to balance the runway load. In this problem, given a set of aircraft $f \in A \cup AD$, and a set of active landing runways $lr_f \in LR_f$, lr_f is the landing runway that can be chosen. In this problem the set of active landing runways is $LR_f = \{0{,}1\}$, meaning that there are only two landing runways available.
- **Pushback time.** The pushback time is a time instructed by the ATCOs or by the ground controller, where aircraft get clearance for leaving the gate and starting the taxiing out maneuvers to reach the departing runway. Depending on the congestion on the ground, ATCOs/ground controllers, can hold the aircraft at the gate for a while before letting it leave the gate. This procedure is applied often, and it is more efficient than holding aircraft at the runway, however, there are constraints regarding departure slots and gate occupancy in the choice of the delay to apply. In other words, the delay to apply should be reasonable, so airlines do not lose their departure slots and gate capacity does not reduce. Formally, given a set of flights $f \in D \cup AD$, where D is the set of departing flights and AD is the set of arrivals and departing flights, the pushback time is defined as $pb_f \in PB_f$, where $PB_f = \left\{PB_f^0 + j\Delta T \mid 0 \leq j \leq \frac{\Delta PB_{max}}{\Delta T}, j \in Z\right\}$. $\Delta PB_{max}$is the extreme side of the decision interval, and represents the maximum value of delay that can be given to an aircraft; its value is +15 minutes. The left extreme of the decision interval is set to zero, meaning that it cannot be given an advance to the aircraft. In real instances, aircraft might receive clearance to leave the gate even before their original pushback time, for example, when passengers have already boarded and the aircraft is

ready to leave. In this specific case, we decided to leave it as zero, assuming that aircraft would not leave the gate before their scheduled pushback time. ΔT is a discretized time increment, and its value is 5 seconds.

- **Departing runway choice.** Similarly to the landing runways, several departing runways can be found at airports. ATCOs instruct aircraft to depart on a specific runway in order to balance the runway load. In this problem, given a set of aircraft $f \in D \cup AD$, and a set of active departing runways $dr_f \in DR_f$, dr_f is the departing runway that can be chosen. In this problem, the set of active departing runways is $DR_f = \{0,1\}$, meaning that there are two departing runways available.

3.1.3.2. Objective and constraints

The objective of the model is to ensure smooth operations, for what concerns the aircraft arrival flow sequencing and merging, and the mitigation of the congestion on the ground. This objective is translated into a mathematical form and then optimized. The smoothness of arrival aircraft flow sequencing and merging operations is achieved by ensuring a conflict-free scenario. In this context, airspace conflicts are defined as a violation of separation minima detected between any consecutive pair of aircraft. Separation minima regulations are given by ICAO (ICAO, 2016), and are shown in table 3.2. The value of separation minima is in nautical miles and they are based on the aircraft wake turbulence category (light, medium, heavy) of the leading (previous) and trailing (following) aircraft. In the airspace network, separation minima conflicts are detected on nodes and links.

Table 3.2. Separation minima ICAO (NM)

		Leading aircraft		
	Wake turbulence category	Heavy	Medium	Light
Trailing aircraft	Heavy	4	3	3
	Medium	5	3	3
	Light	6	5	3

Formally, given a set of landing aircraft $f \in A \cup AD$, for each consecutive pair of aircraft *i* and *j*, we define S_{ij} as the separation minima between the leading aircraft type i and the trailing aircraft type j and D_{ij} as the distance between aircraft type i and aircraft type j. In this way conflicts are detected and calculated as:

$$Airspace\ separation\ minima\ conflict\ (link\ and\ node) = \begin{cases} 1,\ if\ D_{ij} < S_{ij}, \forall i,j \in A \cup AD \\ 0, otherwise \end{cases} \quad (1)$$

This condition is checked at any entry/exit of each link in the network by using (1). Furthermore, when aircraft are not on the same link but converge toward the same node one must check conflict at that node. To check such conflict, we assume a virtual circular protection area around each node with a radius equal to S_{ij}, separation minima between two consecutive aircraft i ad j, and we check that only one aircraft is inside this area at a given time, as it is shown by Figure 3.4. For such case, the separation minima, S_{ij}, refers to the horizontal separation requirement,

which in the TMA is 3 NM. Each time such constraint is violated, the number of extra aircraft in the protection area is counted as conflict.

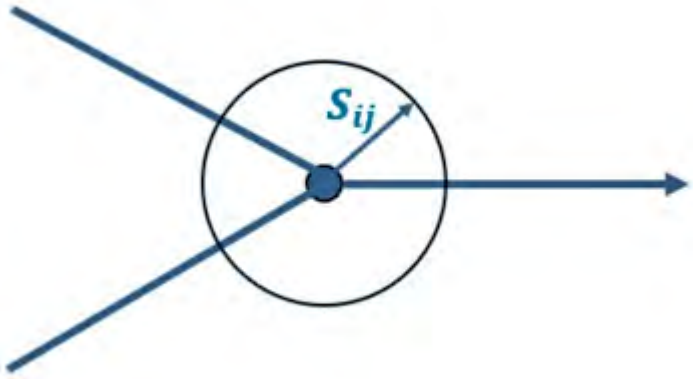


Figure 3.4: Node conflict detection area

Another type of conflict is detected along the links, and it ensures that the aircraft sequence order is maintained at the link entry and at the link exit, thus, making sure that no aircraft overtakes another one. Formally, given a set of links $l(u,v), l = 1,..,n$; $order_{fu}^{l}$ and $order_{fv}^{l}$ are the positions of the aircraft i at the link entry u and at the link exit v, respectively. The order of sequence conflicts are calculated as:

$$Order\ of\ the\ sequence\ conflict = \sum_{l=1}^{n}(\sum_{i\in A} order_{iv}^{l} - order_{iu}^{l}) \quad (2)$$

Then, total airspace conflict number is computed as follows:

$$Airspace\ conflicts = Airspace\ separation\ minima\ conflict\ (link\ and\ node) + Order\ of\ the\ sequence\ conflict \quad (3)$$

Regarding the airside of the airport, congestion has been calculated as capacity overload for taxiway network and terminals, and as separation minima violation at the runway. Given a defined capacity for the taxiway network and for terminals, a capacity overload is defined as the number of aircraft that exceeds this capacity. Given the level of abstraction applied to the ground operations and the assumptions made, the taxiway network was considered as virtual facility able to accommodate a limited number of aircraft simultaneously, while the terminals were considered as a virtual facility with a capacity equal to the number of available gates. Once the number of aircraft simultaneously occupying the taxiway network and/or terminals exceeds this given capacity, congestion is expected to appear and affects operations. The objective of the model is to avoid this congestion to happen, keeping the occupancy level for taxiway network and terminals always lower or equal to their capacity. Capacity overload is computed by measuring two indicators: maximum overload and average overload. The first one is an indicator of the time at which the maximum congestion appears, and the second one is an indicator of the level of congestion. In Figure 3.5, there is a representation of how these two metrics are computed. The maximum overload, is the maximum amount of aircraft exceeding the capacity at a certain moment in time, as it is depicted by the green bar of Figure 3.4; while the average

overload is the amount of aircraft exceeding the capacity over time, as it is represented by the red dashed-line area of Figure 3.5. Formally, given the capacity *C*, aircraft occupancy O_t for each discrete time increment $t \in T$; where T as the entire time frame considered, the max overload is given by:

$$Max\ overload = \begin{cases} max_{t \in T}(O_t - C),\ if\ O_t > C \\ 0,\ otherwise \end{cases} \quad (4)$$

and average overload is given by:

$$Avg\ overload = \frac{overload_t}{T} \quad (5)$$

where,

$$overload_t = \begin{cases} O_t - C,\ if\ O_t > C\ \forall\ t \in T \\ 0,\ otherwise \end{cases} \quad (6)$$

Then, the total overload value for each airside component is:

$$Airside\ conflicts = Max\ overload + Average\ overload \quad (7)$$

This formulation is the same for both taxiway network and terminals.
Conflicts on the runway are calculated in a similar way as the conflicts for the airspace, since they are detected as separation minima violations between two consecutive aircraft. Depending on the runway movement (landing or departure), and on the aircraft wake turbulence category (light, medium, heavy), there are different separation minima to be respected, as shown in Table 3.3 (Frankovich, 2012).

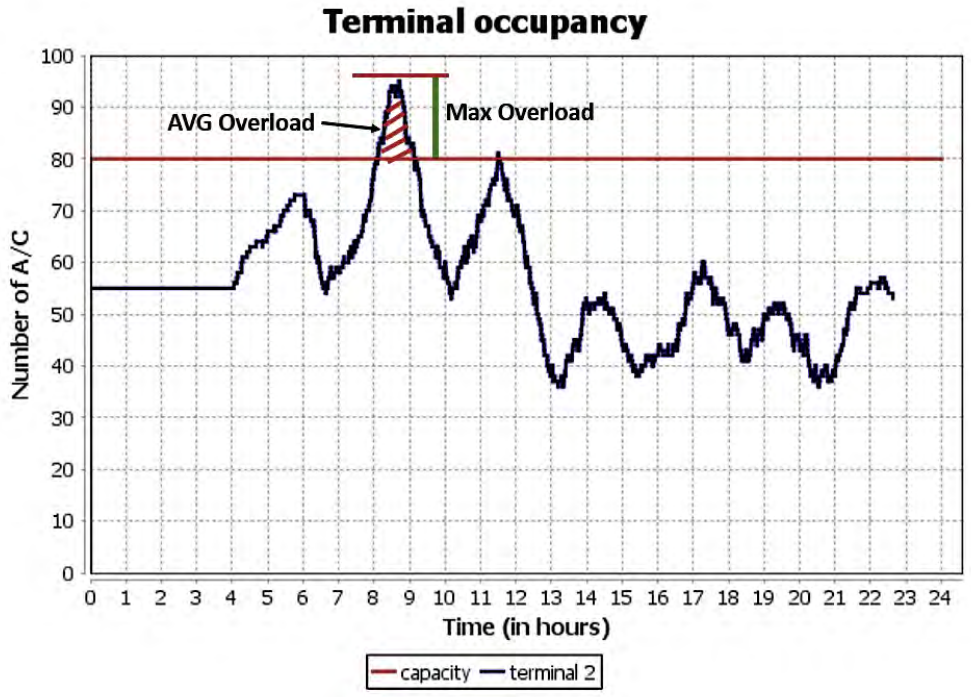


Figure 3.5: Max Overload and Average overload.

Table 3.3: Separation minima between consecutive aircraft on the runway (seconds)

		Leading aircraft					
	Operation – Wake turbulence category	A - Heavy	A - Medium	A - Light	D - Heavy	D - Medium	D - light
Trailing aircraft	A - Heavy	96	157	207	60	60	60
	A - Medium	60	69	123	60	60	60
	A - Light	60	69	82	60	60	60
	D - Heavy	60	60	60	96	111	120
	D - Medium	60	60	60	60	60	60
	D - Light	60	60	60	60	60	60

*A=arrival, D=departure.

Formally, given a set of aircraft $f \in A \cup D \cup AD$, for each consecutive pair of aircraft *i* and *j*, we define S_{ij} as the time separation requested between aircraft type *i* and aircraft type *j* and T_{ij} as the current time separation between aircraft type *i* and aircraft type *j*. In this way conflicts are formalized with the following function:

$$Runway\ conflict = \begin{cases} 1,\ if\ T_{ij} < S_{ij}, \forall i,j \in A \cup D \cup AD \\ 0,\ otherwise \end{cases} \quad (8)$$

The objective function of the problem is a weighted sum of two components, airspace conflicts and airside overload, where, $\gamma_{airspace}$ and $\gamma_{airside}$ are the two weight coefficients for airspace and airside, respectively. The optimization model aims at minimizing such objective function, depending on which component the optimization process should focus on, these weights can be adjusted accordingly.

$$Objective\ function = \gamma_{airspace} * Airspace\ conflicts + \gamma_{airside} * (Airside\ conflicts + Runway\ conflicts) \quad (9)$$

3.1.4. Model uncertainty

In real-world instances, operations are far from ideal scenarios, and this is due to the inherent uncertainty that affect them. Uncertainty plays a major role in the airport system performance. In this thesis, two types of uncertainties have been defined: internal and external. Internal uncertainties are directly related to aircraft performance, while external uncertainties are related to dynamics that happen externally and affect the aircraft performance. For example, internal uncertainties are based on human factor (pilots maneuvers, ATCOs instructions), or technical performance of the aircraft. External uncertainties are related mainly to weather conditions, airport facility availability, and interactions with other actors inside the airport. The airport system is a complex system where different actors interact with each other and use/exchange different resources. Due to these characteristics, modelling all these operations together is a very complex task and requires to make some assumptions in order to simplify it. One of the limitation of similar studies found in the literature is that in their assumptions they do not consider uncertainty, assuming ideal operational conditions and creating a deterministic model

like in Bertsimas and Frankovich (2015) and Guepet et al. (2017). This limitation motivated one of the main contribution of this thesis, which represents the modeling of the uncertainty related to airport operations and the evaluation of the impact on the airport system performance. This contribution is fulfilled by using simulation techniques, which allows to model stochastic variable within the system. In the following list, the consequences on airport operations due to uncertainties that have been considered for this model, are described:

- Airspace uncertainty consequence:
  - Entry time in the TMA deviation. When ATCOs instruct the pilot to approach to the first waypoint of the STAR at a certain time and speed, internal uncertainty like miscommunication and pilot reaction time can play a role in the precision of the operations, leading to differences in the requested entry time in the TMA and the actual entry time in the TMA. Moreover, external factors such as weather can affect aircraft performance and contribute to the on time performance of the aircraft. In this context, a time interval around the ATCOs instruction on the entry time in the TMA has been defined, as it represents the uncertainty related to these operations.
- Airside uncertainty consequence:
  - Push back time deviation. ATCOs or ground controllers, communicate the pilot and ground handlers to set the pushback time, which is the time when it is possible to leave the gate. However, when the aircraft is parked at the gate and undergoes all the turnaround operations, ground handlers performance efficiency plays a major role in the on time performance of the push back time. Communication between ATCOs/ground controllers, pilots and ground handlers is also crucial, as bad communication can lead to delay operations. These factors are related to internal and external uncertainty, and can affect the push back time of the aircraft. In this thesis, a time interval around the pushback time has been defined in order to model this uncertainty.
  - Taxi (in/out) time deviation. When aircraft are taxiing toward the departing runway (taxi out), or towards the gate (taxi in), internal and external uncertainty come into play and affect the taxi times. Usually, the most uncertainty comes from taxi network layout complexity, level of congestion of the taxiway network, weather condition, ATCOs/ground controller instructions, which can affect the speed and the taxiway route of the aircraft. In this thesis, a deviation, positive or negative, from the average taxi time has been defined as a percentage of the average taxi time. Table 3.4, shows the values of the uncertainties that were considered for this thesis.

Table 3.4. Operations affected by uncertainty and their value.

Uncertainty consequence on operations	**Value**
Entry time in the TMA deviation	[-30 sec., +30 sec.]
Pushback time deviation	[-30 sec., +30 sec.]
Taxi (in/out) time	[-10%, +10%]

3.2. General architecture of the Opt-sim approach

The general approach implemented for solving this problem, consists in framework where optimization and simulation techniques are employed. By optimizing the system, it is possible to achieve an optimal or near optimal solution by changing the values of some decision control (variables), however, this remains a deterministic approach since the variables and parameters of the problem are considered as deterministic ones. Since most of the real-world operations are affected by variability, when we want to apply these optimized solutions to real cases it is likely to obtain unfeasible implementations. In order to overcome this flaw, simulation comes in help, allowing us to model also some of the stochastic variables that affect the system. Under this point of view, optimization and simulation are complementary, this fact encourages their use in combination. This section will describe how the optimization and the simulation processes were conceptualized and developed. The optimization process is obtained by the implementation of a sliding window approach and by the use of a metaheuristic (simulated annealing) for solving the problem. The optimization model was developed by using the programming language Java. The simulation process is given by the development of a discrete event simulation model, which was able to include some source of uncertainty and to evaluate the objective function of the problem. In this thesis, two different ways of combining optimization and simulation were investigated, and to this purpose, two simulation models were developed by using two different general purpose simulation software: SIMIO and Anylogic. In Appendix A and B these two models are described in more detail.

3.2.1. Sliding window approach

The optimization process is based on the implementation of a sliding window approach. This approach has been employed in other studies obtaining good results (Hu and Cheng 2005, Zhan et al. 2010, Furini et al. 2015, Toratani et al. 2015, Liang et al. 2015, Ma et al. 2016). The sliding window approach allows to explore the whole time horizon by considering at each iteration a small time frame (window) and by shifting it along the entire time horizon. By using this approach, a small instance of the problem is solved considering only the elements within the window. The main advantages coming from the use of this method are: less computational time required; the possibility of treating the problem in a dynamic way, by considering, as time passes, new information updates due to changes of the external environment and due to the interactions between entities with the surrounding environment; the reduction of the uncertainties that affect the decision variables. In the next paragraph are listed some useful notations for understanding the sliding window approach:

- W = Window length
- S = Shift length
- T_{start} = Starting time of the total time horizon
- T_{end} = Ending time of the total time horizon
- t_s = Starting time of the window
- t_e = Ending time of the window
- t_s^f = Starting time of the flight $f \in F$, for flight type "*arrival*" where $f \in A \in F$, and flight type "*arrival and departure*" where $f \in AD \in F$, the starting time is the entry time in the TMA; for flight type "*departure*" where $f \in D \in F$, the starting time is the push back time.

- t_e^f = Ending time of the flight $f \in F$, for flight type "*departure*" where $f \in D \in F$, and flight type "*arrival and departure*" where $f \in AD$, the ending time is the take off time; for flight type "*arrival*" where $f \in A \in F$, the ending time is the in-block time, which is the time when the aircraft arrives at the gate.

The main parameters to set are the window length W and the shift length S. It is crucial to choose the right values for the window and shift length, since big values of window length might lead to worsen the overall performance, as decisions for entities that are too far ahead in time are made with a lot of uncertainties. Moreover, there will also be an increase in the computational time. On the other hand, small values of window length might lead to not consider enough information about aircraft, jeopardizing the dynamicity of the approach. In the same way, a big value of the shift length will reduce the effect of the dynamicity of the method, since less decisions will be updated from one window to another. Small values of shift length, on the other hand, will increase the computational time performance.

As it can be appreciated on the Figure 3.6, depending on the relative time position of the aircraft entities, t_s^f and t_e^f, with respect to the current active window starting time t_s and ending time t_e, each entity is categorized into different categories of flights: *completed*, *on-going*, *active*, and *planned*;

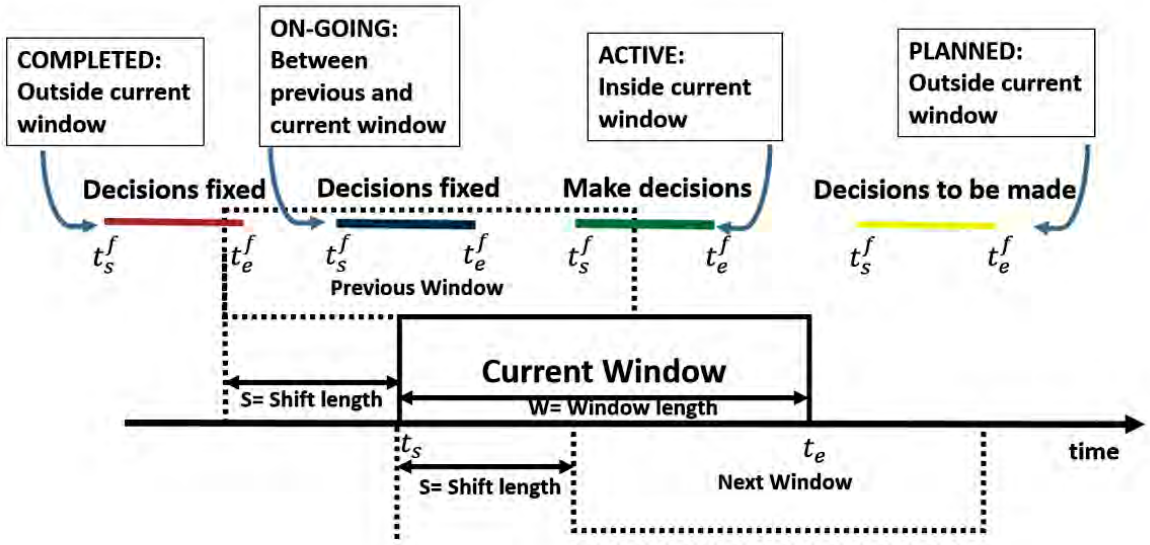


Figure 3.6: Schematic representation of the sliding window approach

- **Completed flights.** This category of flights is for the ones that fall entirely behind the current window, for them, decisions are already taken and fixed, thus there is no possibility to update their decisions. Formally, $t_e^f \leq t_s$, meaning that the ending time of the flight $f \in F$ is lower than starting time of the current window.
- **On-going flights.** These category of flights is for the ones that fall partly before and partly inside the current window, formally, $t_s^f \leq t_s \leq t_e^f$, meaning that the starting time of the flight $f \in F$ is lower than the starting time of the window t_s and the ending time

of the same flight is greater than the starting time t_s of the window. These decisions were already made in the previous window (when they were active flights), however, they are considered in the optimization of the current window affecting the decisions for the active current flights.

- **Active flights.** This category of flights involves the ones that are fully inside the current window as $t_s \leq t_s^f \leq t_e$, meaning that the starting time of the flight $f \in F$ falls between the starting t_s and ending time t_e of the current window. Decisions for this flight category will be made in the current window.
- **Planned flights.** This category of flights is for the ones for which the starting time is greater than the window ending time having $t_e \leq t_s^f$, and for this flight category decisions are not yet made.

As it has been described, the sliding window approach allows the optimization to make decisions only on specific entities (flights), the ones who are directly involved in the current window. *On-going* and *active* flight categories are the ones that are considered in the optimization process of the current window. In the current window, the optimization will make decisions only for *active* flights, however, the operations of the *on-going* flights inside the window will be considered, in this way the dynamicity of the approach is kept. Flights of type *completed* and *planned* are not considered by optimization of the current window, reducing the computational burden.

3.2.2. Optimization by simulated annealing

The problem under study has been proved to be NP-Hard (Beasley et al. 2000), for that reason, a metaheuristic has been implemented in order to solve the optimization problem. The metaheuristic that has been implemented is an adapted version of the well-known simulated annealing (Kirkpatrick et al. 1983). The main characteristic of this metaheuristic is that it allows to escape from local optima by doing hill-climbing moves in the state space, accepting solutions which are worse in terms of objective. In this way, it allows to explore the state space more broadly, leading to find a global optimum. In other similar studies, this metaheuristic has been proved to be very efficient (Ma et al. 2016, Liang et al. 2017). The name simulated annealing comes from the analogy with the process of physical annealing with solids, where a crystalline solid is heated and then cooled down until it achieves its most possible regular crystalline lattice configuration (Henderson et al. 2003). The concept behind this thermo-dynamic process has been translated to optimization, where given an initial temperature T_0, a cooling process is done in the search of an optimal solution. At each value of temperature T, two solutions are found and compared, the algorithm accepts all the solutions which are better in terms of objective function, but also worse solution based on an acceptance probability, in this way it avoids to get trapped in local optima. During the cooling schedule, if the temperature decreases sufficiently slowly, there are high chances to converge to a global optimum. The main parameter of the simulated annealing is the temperature T, from its value it depends most of the algorithm efficiency and effectiveness, since it is directly connected with the probability of acceptance of worse solutions, the cooling schedule and the stopping condition. In general, as the temperature value is high, hill-climbing moves are more likely to happen, while, when the temperature decreases to zero, less hill-climbing moves happen, and solution converges to global optimum. In **Algorithm** 1 the steps of the algorithm are listed, while Figure 3.7 depicts a flow-chart explaining the general algorithm of the simulated annealing (Eglese 1990).

Algorithm 1 Simulated annealing

1: **procedure** simulatedAnnealing
2: $\omega \leftarrow$ select an initial solution $\omega \in \Omega$
3: k $\leftarrow$ select the initial temperature change counter to 0
4: $t_k \leftarrow$ select a temperature cooling schedule
5: $t_0 \leftarrow$ select an initial temperature $t_0 \geq 0$
6: $M_k \leftarrow$ select a repetition schedule, number of iterations executed at each temperature t_k
7: **repeat**
8: $m \leftarrow$ set repetition counter to 0
9: **repeat**
10: $\omega' \leftarrow$ generate a neighbor solution $\omega' \in N(\omega)$
11: $\Delta_{\omega,\omega'} \leftarrow$ calculate $\Delta_{\omega,\omega'} = f(\omega') - f(\omega)$
12: **if** $\Delta_{\omega,\omega'} \leq 0$ **then**
13: $\omega \leftarrow \omega'$
14: **if** $\Delta_{\omega,\omega'} > 0$ **then**
15: $\omega \leftarrow \omega'$ with probability $\exp(\Delta_{\omega,\omega'}/t_k)$
16: $m \leftarrow m + 1$
17: **until** $m = M_k$
18: $k \leftarrow k = k + 1$
19: **until** stopping criterion is met
20: **end procedure**

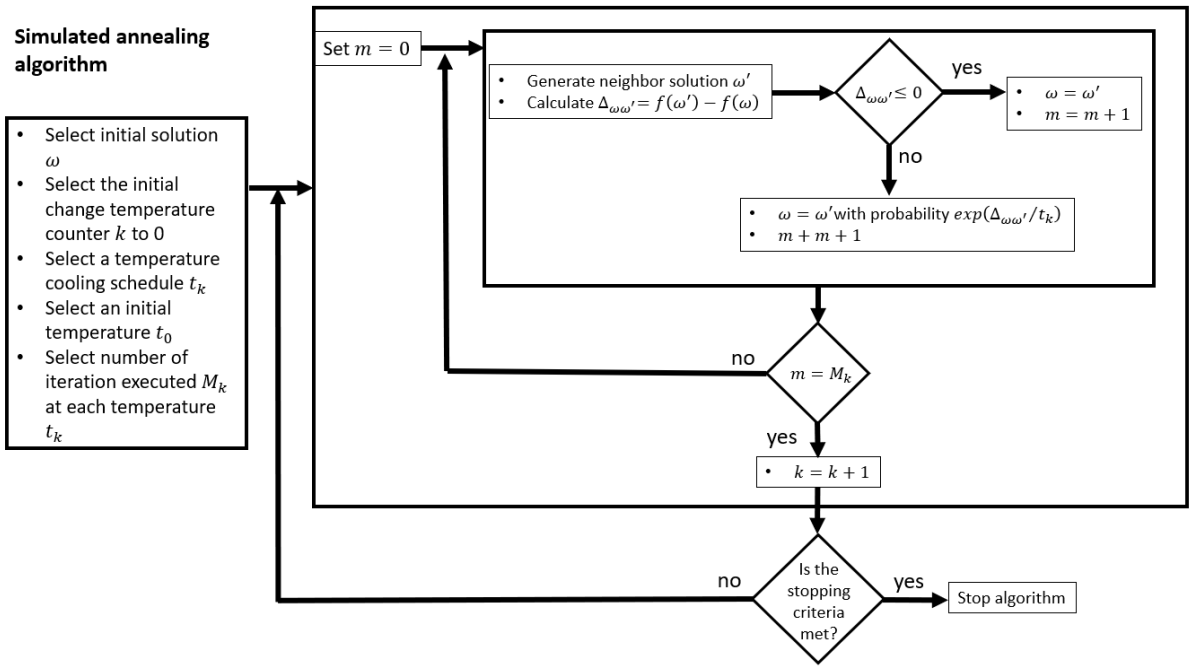


Figure 3.*7*: Simulated annealing algorithm

In line 4 of **Algorithm 1** after a first solution is generated, the cooling schedule for the temperature needs to be defined. Usually there are two cooling schedule approaches that can be applied, static and adaptive cooling schedules. The former assumes that a cooling schedule is defined before running the algorithm, while the latter assumes that the temperature decrease rate

will change based on the information provided during the run of the algorithm. Some of the most used cooling schedule are the following:

- Geometric cooling schedule. At each iteration k, we decrease the temperature t_k multiplying it by a coefficient α, where $0 \leq \alpha \leq 1$.
$$t_{k+1} = t_k \times \alpha$$
The choice of this coefficient is very important because having a coefficient α too big will slow down a lot the convergence of the algorithm, while a value too small will decrease the temperature too fast, putting at risk the capability of finding a global optimum.
- Linear cooling schedule. At each iteration k, we decrease the temperature t_k by:
$$t_k = t_0 - \beta \times k, \beta > 0 \; and \; t_k > 0$$
Where β is predefined coefficient, in this case the same considerations for the coefficient α of the geometric cooling schedule apply for the coefficient β.
- Logarithmic cooling schedule. Each iteration k, the temperature $t_k$decreases following a logarithmic law:
$$t_k = t_0/\log(i)$$

In this thesis the cooling schedule implemented was the geometric, and the coefficient α was set equal to 0.99.

In line 5 of **Algorithm 1**, the initial temperature t_0 must be set. In this work the way this temperature was chosen is explained in **Algorithm 2** and in Figure 3.8.

Algorithm 2 Initial temperature

```
1: procedure initialTemperature
2: T_0 ← set initial temperature
3: nbTransitions ← set number of transitions
4: do
5:     acceptCount ← 0
6:     T ← T × 1.1
7:     for i=0 to nbTransitions do
8:         ω ← generate solution ωϵΩ
9:         ω′ ← generate neighbor solution ω′ϵΩ
10:        if ω′ is accepted then
11:            acceptCount ← acceptCount + 1
12:        end if
13:    end for
14: φ ← acceptCount/nbTransitions
15: while φ < 0.85
16: T_0 ← T
17: return T
18: end procedure
```

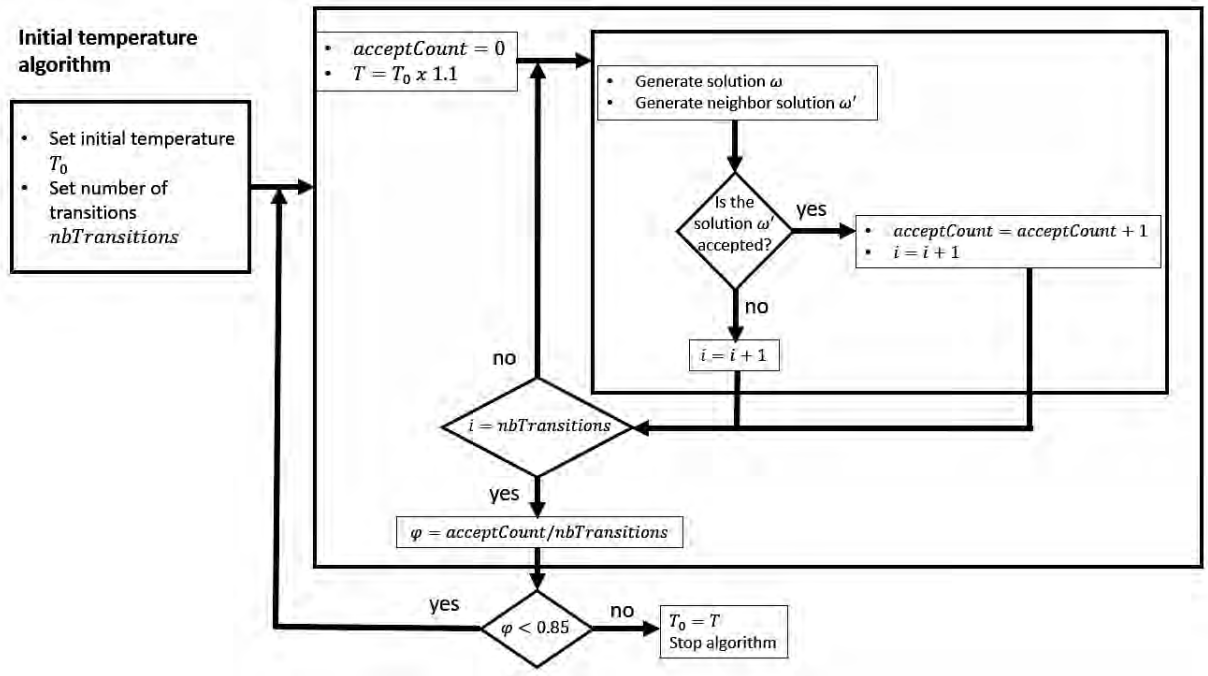


Figure 3.8: Initial temperature algorithm

The initial temperature was set as 0.01, and the number of transitions "*nbTransitions"* was set as 150. In line 6 of **Algorithm 2,** the temperature is increased by 10% at each iteration until the rate of accepted solution is equal or greater than 0.85, which is the 85% of the number of transitions, as it is seen in line 15 of **Algorithm 2**. The number of transitions of **Algorithm 2** is equal to the number of repetitions M_k in **Algorithm 1**. Regarding computation of the neighborhood solution, instead of randomly changing the decision variables for each entity, the algorithm targets the areas which are most affected by the overload or conflicts. Two main areas have been defined: airspace and ground, and for each entity performance for the two areas were computed. Airspace performance are given by conflicts on links and nodes, while ground performance are given by runway conflicts, taxiway network and terminal overloads. Depending on these performances, it was decided which decision to change for each flight. Decisions involving airspace performance are the entry time in the TMA, and the entry speed in the TMA, while decisions involving ground performance are the landing and departing runway choices and the pushback time. In **Algorithm 3** the neighborhood solution computation is explained.

For instance, as it is depicted in the example of Figure 3.9, given a total performance of 100, and the target value set to 30, the flight index is 3. According to the algorithm the flight number 3 will be selected and its decisions will be modified. Flight $f_3 \in A$, has an airspace performance equal to 20 and ground performance (runway) equal to 2, this means that the next solution will affect this flight only on decision involving the airspace performance (entry time in the TMA and/or entry speed in the TMA).

The last feature of the simulated annealing metaheuristic is the stopping criterion. Typically, maximum computational time, maximum number of transitions without improving the objective function value, minimum temperature value are some of the most used stopping criteria. In this study, the stopping criterion was a combination of the aforementioned stopping criteria, having

the following stopping condition: if the objective function reaches a global minimum (zero), or if the temperature t_k is less than $t_0 \times 0.0001$.

Algorithm 3 Neighborhood solution

Input: for each flight *f*, airspace performance P_f^a, runway performance P_f^r, ground performance P_f^g, total performance $P_f^t = P_f^a + P_f^r + P_f^g$, total number of conflicts $P_t = \sum_{f \in F} p_f^t$.

1: **procedure** neighborhoodSolution
2: $v \leftarrow$ random number random(0,1)
3: **if** $P_t > 0$ **then**
4: sum ← 0
5: target ← $P_t \times v$
6: i ← 1
7: **while** sum < target **do**
8: sum ← sum + P_f^t
9: i ← i + 1
10: **end while**
11: **else**
12: choose randomly one flight *i* from the flight set
13: **end if**
14: **if** i∈ A **then**
15: **if** $P_f^a > 0$ **then**
16: choose with equal probability between the entry time and the entry speed in the TMA, then choose randomly one value within their decision interval range
17: **else if** $P_f^r > 0$ **then**
18: choose with equal probability among the entry time in the TMA, entry speed in the TMA, and the landing runway
19: **else**
20: choose randomly the entry time in the TMA
21: **end if**
22: **else if** $i \in D$ **then**
23: **if** $P_f^g > 0$ **then**
24: choose randomly the push back time within the its interval decision range
25: **else**
26: choose with equal probability between the pushback time and the departing runway
27: **end if**
28: **end if**

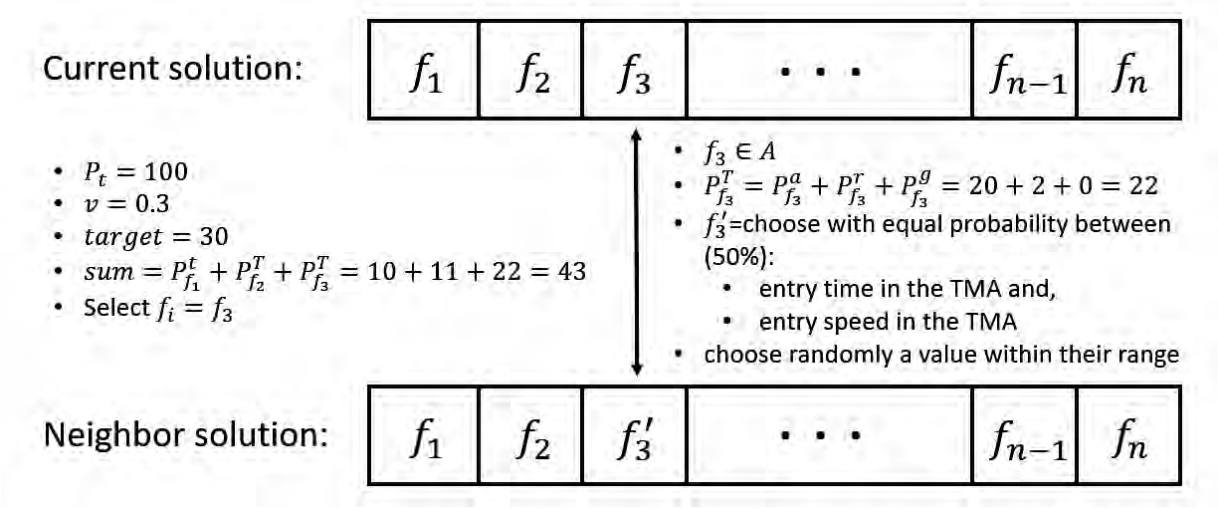


Figure 3.9: Neighbor solution generation example

3.2.3. Simulating by discrete event simulation

The simulation model for the airspace and airside operations has been developed with the use of DES. DES has been widely used and proved as a powerful tool for conducting analysis based on different scenarios and as a decision support tool in different fields. This approach has been consolidated over the last decades since it has been implemented with success by other authors in many fields (Brunner et al. 1998; Banks et al. 2000; Negahban and Smith 2014). This type of approach fits best for the type of systems that are process driven, where entities follows some predefined path, undergo predefined processes and perform predefined actions. One of the main feature of DES is that the time changes based on specific events, and not in a continuous manner as it happens for continuous simulation approaches. Moreover, this approach is different than an agent-based simulation (ABS) due to the fact that the entities involved in the system do not have their own logic, as it happens for ABS, but they follow the logic of the processes in which they are involved. An airport system, as it was described in section 3.1, implies all the characteristics described for DES, and that is why, for this problem, DES approach was chosen rather than other simulation approaches. The main advantage of using simulation in this work, is that it includes sources of uncertainties in the system. Uncertainty can derive from different causes like external and internal factors such as the surrounding environment or the human factor. Simulation in this context, is able to overcome the flaws deriving from deterministic optimization, where exact solutions are generated without considering the stochastic nature of systems.

The use of a DES gives some advantages such as: visualization, time compression and expansion, use of stochastic variables, running experiments with several replications. Moreover, by using general purpose simulation software, users and developers can take advantage from some of their common features such as: modeling by using predefined object, extending logic by using advanced features.

- **Simulation visualization.** By using simulation, it is possible to visualize the dynamics of a system and get more insights about its behavior, conducting a thorough analysis of the system. Figure 3.10 shows a screenshot of a general purpose simulation software, with its animation and main features such as the command bar and the library with predefined objects. The centered panel shows the airspace network developed for the

case study considered in this thesis (Paris Charles de Gaulle Airport, west configuration).

- **Time compression and expansion.** This feature allows analysts and users to make different kind of analysis, like concentrating on details when slowing down the simulation time or being more generic by speeding up the simulation time. Moreover, simulation can be paused and replayed again, when a certain instant of the simulation needs to be checked. Especially for planning purposes, it is really important to manipulate the time and to be able to simulate a long time-horizon in a small amount of time. For instance, the simulation of one day of operation at an airport can be done in few seconds.

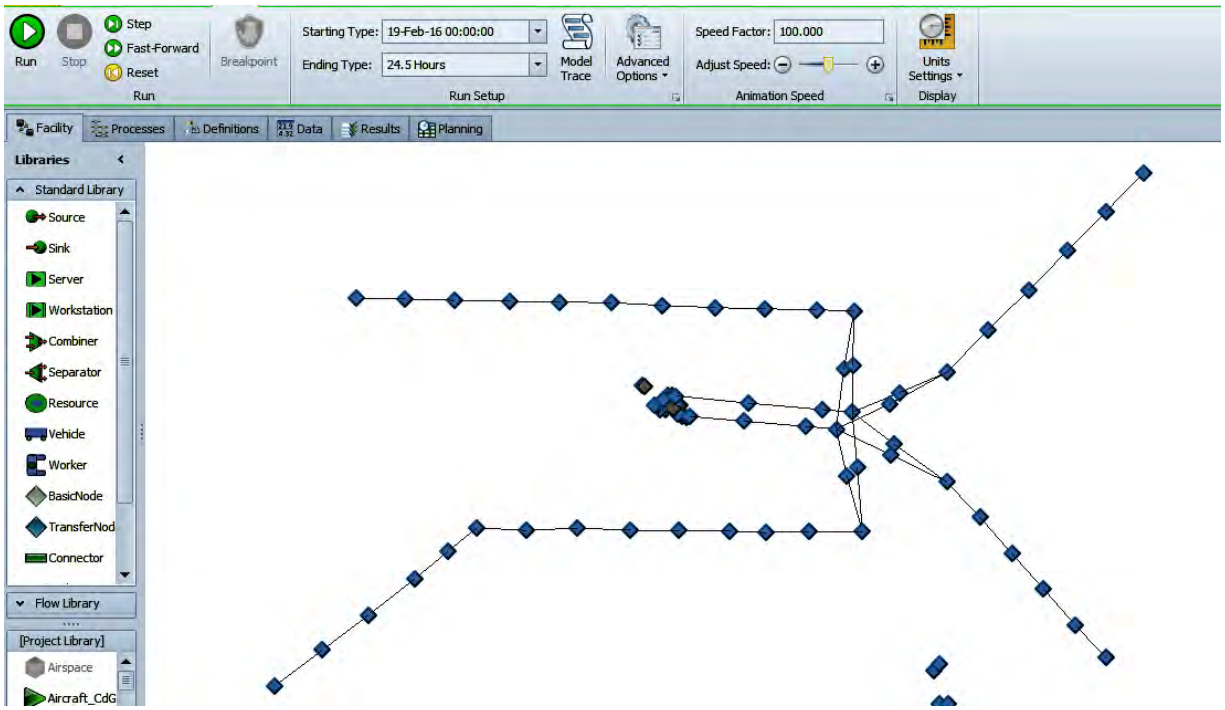


Figure 3.10: Main window of a general purpose simulation software (SIMIO)

- **Stochastic variables.** The use of stochastic variables represents one of the main advantages of using simulation. By setting stochastic values to variables, it is possible to model the uncertainty of the system. Most of the commercial simulation software have available libraries with many probability distributions that can be used to represent stochastic variables.
- **Running experiments.** One of the main feature of a simulation software is the capability of running simulation experiments. A simulation experiment consists in running a specific scenario of the system, based on a given parameter setting. A simulation model containing stochastic variables, is defined as stochastic simulation model. The output variables of a stochastic simulation model are estimates that contain random error. For this reason, several replications are required when running experiments so that a statistical analysis can be conducted in order to obtain precise output estimates. Figure 3.11 shows the experiment window of a general purpose simulation software, containing the main components such as experiment name, number of replications and parameters of the model.

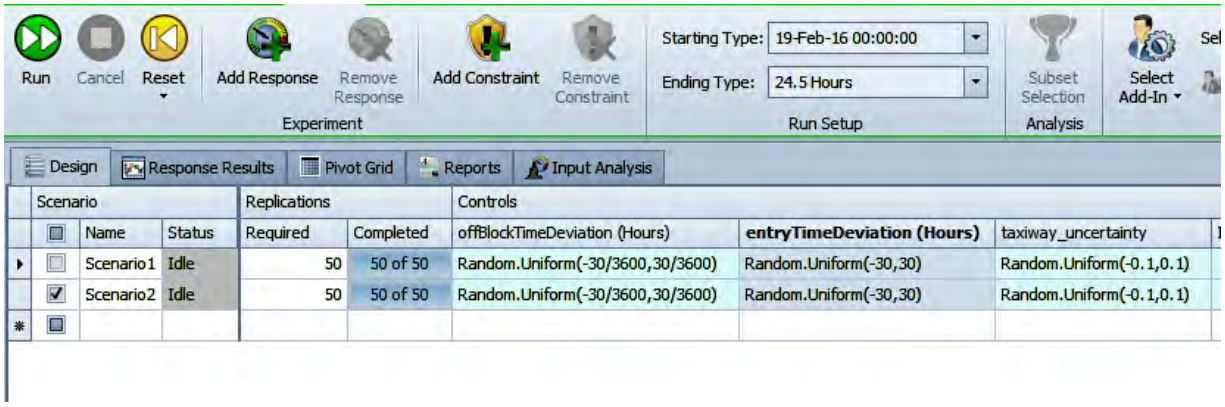


Figure 3.11: Screenshot representing the experiment window of a general-purpose simulation software (SIMIO).

- **Modelling using predefined objects.** General purpose simulation software, provide the developers with libraries of predefined objects that can be used to model any kind of system. This feature is very powerful since it does not restrict the developers to model a limited amount of systems. The most common objects, present in every DES general purpose simulation software are: source, server and sink.
    - Source object. The purpose of this object is to create entities. Entities can be created based on a predefined schedule, based on an inter-arrival time (deterministic or stochastic), moreover, it is also possible to set the number of entity to be created each time.
    - Server object. This object is characterized mainly by two properties: server capacity, and server processing time. Entities entering this object stay there for the whole processing time which has been defined in the server, moreover, the capacity of the server restricts the amount of entities that can be processed simultaneously.
    - Sink object. This object represents the end of the system operations. When entities enter this object, they are automatically destroyed.
    - Nodes and links. These objects are usually used for representing network and/or for connecting objects. In the nodes and links objects different properties can be set such as: capacity, for nodes and links, and direction, maximum speed, among others. In each node and link, is possible to call some functions, these are useful when advanced logic need to be implemented in the system.
- **Modelling advanced logic.** Having libraries with predefined objects relieves from the modelling effort, but on the other hand, when developers need to model more advanced dynamics, these objects are limited. In order to overcome this problem, usually even for commercial simulation software, it is possible to extend the logic by using advanced tools. Depending on the simulation software, the logic can be extended by programming advanced functions and implementing them in the software as an external application or as an external library. The programming language to be used depends on the simulation software, as well as the effort for developing external applications/libraries.

3.2.3.1. Simulation model of airport operations at a macroscopic level

The simulation model of the airport operations at a macroscopic level, as they were described in section 3.1, have been modelled using a general purpose simulation software. In order to

develop the model, the main simulation software features have been used such as entities, object library, variables and control parameters. In the following paragraphs, the main feature of the model are explained:

- **Model structure.** The structure of the model relies mainly on the use of source, server, sink, nodes and links objects. The source object was used for generating flights based on a specific flight schedule, while the sink object was used for ending the model after aircraft had departed. The airport airspace was modelled as a network of node objects and link objects, representing the waypoints and segments of the landing routes. In each node, logic such as speed-check and separation minima-check were implemented. Regarding the ground, the main components such as runways, taxiway network and terminals were modelled by server objects connected to each other. In table 3.5 there is a summary of the main characteristics for each object used in the development of the simulation model.

Table 3.5. Summary of the object characteristics of the simulation model

Object	Characteristics
Model entity: aircraft	• Initial speed [m/s] • Acceleration [m/s^2] • Flight type • Wake turbulence category • Entry point in the TMA • Entry time in the TMA [s] • Terminal number • Landing runway • Push back time [s] • Departing runway
Source	• Creates aircraft model entities according to a flight schedule given as input. • Generates deviation from the scheduled entry time [s] of the entities in the model for modeling uncertainty.
Server	• Models runways, taxiway network and terminals ground components • Capacity: component-specific • Processing time [s]: component-specific
Sink	• Destroys entities (end of the model)
Node	• Calls a function for detecting airspace conflicts between aircraft
Link	• Length of the segment [m]

- **Model parameters.** These parameters are static, in the sense that they are set before the simulation is run and they cannot be changed while the simulation is running. In the following, the main static parameters are listed:

- Terminals initial occupancy. It represents the capacity of the terminals at the beginning of the operations. Some aircraft are already parked at the terminal when the model starts, for example flights of departure (DEP) type.
- Terminals and taxiway network capacity. It is the maximum capacity given to the taxiway network and to the terminals.
- Landing speed. Depending on the wake turbulence category, a specific value is given. This parameter is used for the calculation of the aircraft acceleration during the landing phase.
- Uncertainty parameters. These parameters are related to the sources of uncertainty considered in the model. Entry time in the TMA, taxi (in/out) time and push back time are the parameters set in the simulation.

- **Model logic.** These logic are created by developing some functions. Below, the main logic implemented are described:
  - Speed. This logic is created by implementing the mathematical formula of the speed based on a specific acceleration. The Speed is updated at each second in the simulation.
  - Conflict detection on nodes. In each node, speed, time and wake turbulence category of the leading and trailing flights are stored, and also the actual time separation between them. After that, based on these information, the minimum separation between these aircraft is derived and compared to the actual time separation for detecting potential conflicts.
  - Conflict detection on runways. Conflicts on the runway are calculated in the same way as node conflicts, the only difference is linked to the are different separation minimum values.
  - Overload detection on taxiway network and terminals. Taxiway network and terminals were modelled by using server object, here a function for computing overload is called every time an entity enters. The function compares the actual number of entities inside the server and the capacity parameter set for the corresponding airside component (taxiway network, terminals). If the actual number of entities inside the server is bigger than the capacity value, an overload is detected, applying the formulas 4 and 5 of section 3.1.3.2.

- **Model data input.** The data input is given by a flight schedule as it is depicted in Figure 3.12. This flight schedule is imported in the simulation model as an excel file and contains all the information needed by the entities. The flight schedule contains information such as: flight type, wake turbulence category, terminal number, entry point in the TMA, entry time in the TMA, entry speed in the TMA, landing runway, pushback time, and departing runway. In the simulation model, for each aircraft, a number of variables equal to the data input information have been created in order to store and use these information during the simulation.
- **Model performance indicator.** In the model, some performance indicators have been created and used for evaluating the performance of the objective function of the problem. Therefore, indicators such as airspace conflicts, runways conflicts, taxiway network/terminal overload were implemented. In the simulation model, it was also possible to get more information about the performance, such as the number of aircraft

involved in airspace conflicts and also about the severity of airspace and runway conflicts.

Table1 | Table3 | Table4 | Table4_Decisions | Table1_before_decisions | Table1_Decisions | Table2

Bound to Excel: E:\Wintersim 2017\test 2 optimized\third try\Data Input Simio test 2_3.xlsm, Worksheet: Sheet2

	type AD	w TCat	terminal Number	entry Point	entry Time	rwy Out Dep	entry Speed	rwy In
1	2	2	3	0	19-Feb-16 00:08:18	2	0	0
2	2	2	3	0	19-Feb-16 00:19:19	2	0	0
3	0	1	3	3	19-Feb-16 00:45:46	0	282.54954	1
4	2	2	3	0	19-Feb-16 00:33:23	2	0	0
5	0	1	3	4	19-Feb-16 01:20:17	0	461.54878	1
6	2	2	3	0	19-Feb-16 01:25:46	3	0	0
7	0	2	2	3	19-Feb-16 03:51:46	0	282.1275	0
8	0	2	2	1	19-Feb-16 03:52:48	0	344.9888	0
9	0	2	2	2	19-Feb-16 03:57:10	0	240.5241	1
10	0	2	2	1	19-Feb-16 04:09:11	0	377.47978	1
11	0	1	2	3	19-Feb-16 03:48:53	0	340.17	1
12	0	2	3	1	19-Feb-16 04:10:55	0	367.13184	1
13	0	2	2	2	19-Feb-16 03:54:47	0	312.016019...	1
14	1	2	3	3	19-Feb-16 03:53:23	0	335.31086	1
15	0	2	3	3	19-Feb-16 04:08:31	0	312.521569...	0

Figure 3.12: Screenshot of the flight schedule with the data input of the model. Each column corresponds to a data input record (SIMIO).

3.3. Methodological approach for combining simulation together with optimization

The approach combining optimization and simulation introduced in section 3.2 has been investigated and studied in deep in this chapter. Two ways of combining optimization and simulation have been developed and they will be described in the next sections. The first method uses simulation as a validation tool for solution obtained from an optimization model. In this context, validating a solution means testing its feasibility in a close-to-reality environment, which is provided by the simulation. Furthermore, this first method aims at improving the robustness of the solution by fine-tuning some of the input parameters at each iteration. In the second method, optimization is fully integrated in a simulation environment. In this method, an algorithmic approach is developed, where optimization and simulation continuously work together for finding a better solution at each time window of the sliding window approach. The main differences between the two methods are the followings: first, while the first method involves iterations between the two components (optimization and simulation) during the solution process, the second method proposes a framework that embeds the simulation component in the optimization part. Second, in the first method, the optimization model and the simulation model were considered as two distinct and separate entities, where the simulation part was used essentially to validate the solution of the optimization model. In the second method, simulation contributes to the improvement of the optimized solution. In this context, the simulation component is tightly coupled within the solution improvement algorithm in each sliding window when solving the optimization model. Therefore, the second method fully benefits from the dynamic characteristics of the simulation results that are generated during the process of solving the optimization model.

3.3.1. Opt-sim method 1: ex post optimization

In the Opt-Sim method 1, optimization and simulation techniques are combined in order to test optimized solution for the airport capacity management at macroscopic level. In this method, the uncertainty is introduced after the optimization has found an optimized solution, with the objective of testing and validating the robustness of the optimized solution. As already explained in chapter 3, simulation allows to include the uncertainty present in the ATM framework, in an optimization process, the use of both techniques provides more robust and feasible solutions. In this context, robustness is defined as the capability of the system of being resilient to perturbations. Perturbations are defined as sources of variability that affect the ideal airport operations, these perturbations can be represented as flight inefficiencies, human factor, miscommunication, weather conditions, and so on. This method aims at finding a more robust solution by tuning the input parameters of the optimization. Figure 3.13 shows a schematic representation of this methodology, where first, an optimization model based on a sliding window approach and solved by the simulated annealing metaheuristic, is run, providing an (sub)optimal solution evaluated by its objective function value Z_opt. The description of the optimization model, can be found in section 3.1.3, while the description of the sliding window approach and the adapted simulated annealing metaheuristic can be found in sections 3.2.1 and 3.2.2. Then, a simulation model (Appendix A) is used in order to evaluate the same objective function and include uncertainty in the problem, obtaining as output, the objective function Z_sim. The robustness of the solution is measured by the value of Z_sim. Since the optimization provides always a conflict-free solution, the absolute value of the objective function evaluated by the simulation, Z_sim, is taken as indicator of the robustness. The objective is to keep the value of Z_sim as much close as possible as Z_opt, in order not to deviate too much from the optimal solution. The deviation is linked to the robustness of the decision proposed by the optimization algorithm. In chapter 3 the sources of uncertainty that were included in the simulation model have been described in detail. The simulation model was capable of including these perturbations in the form of stochastic variables so that the objective function value calculated can be used for measuring the robustness of the optimized solution. In this approach, the solution robustness is measured a posteriori.

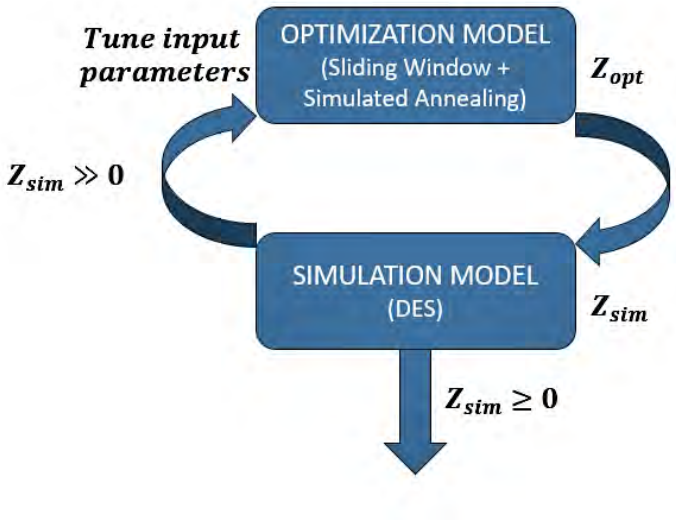


Figure 3.13: Schema of the opt-sim framework of method 1.

An important aspect of this method is the choice of the input parameters that have to be fine-tuned. Some of them can influence more than others the improvement of the solution robustness. In this work two categories of input parameters have been defined: problem-related and metaheuristic-related.

- Problem-related parameters. These parameters are related to the main features of the problem, such as decision variables, constraints and objective function. The idea behind this choice is that, by changing some of these parameters, the optimization process might be able to provide a new solution that might be more robust. For example, fine-tuning the value range of the decision variables (expanding or restricting), modifying the weights of the objective function, or relaxing constraints are ways of fine-tuning problem-related input parameters. In the implementation of the opt-sim method 1 the problem-related parameter taken into consideration was:
    - The separation minima increase; at each iteration, we increase the separation minima constraint (i.e. constraint relaxation). In theory we shall obtain an optimized solution which will be more conservative. However, when we simulate the solution, it can potentially absorb the effect of uncertainty.
- Metaheuristic (simulated annealing)-related parameters. These parameters are related to the metaheuristic. In this specific case, the metaheuristic implemented is the simulated annealing, however, this concept can be extended to other metaheuristics. By fine-tuning metaheuristic parameters, the method tries to find a new solution without impacting operational variables. Examples of parameters of the simulated annealing metaheuristic that can be fine-tuned are: the parameter α regarding the cooling schedule; the number of transitions M_k per each iteration; the rate of accepted solution for calculating the initial temperature t_0. The metaheuristic-related parameter considered in this method was:
    - The cooling schedule parameter α; we increase the simulated annealing cooling schedule parameter, α, so that the optimization model will generate a solution by exploring the state space more in depth.

3.3.2. Opt-Sim method 2: Algorithmic approach for a continuous improvement of the solution

In this method, for each time window, simulation is combined together with optimization in a loop that aims at improving the intermediate solution of that window. In order to achieve that, an algorithm has been developed and coded within the simulation software. The algorithm developed for this framework is summarized in **Algorithm 4** and **5**.

At the beginning of the algorithm some variables need to be initialized:

- M: objective function tolerance. It is the maximum value of objective function that is accepted, even though it does not give an optimal (conflict-free) solution. Beyond this value a new iteration (loop) of the method is implemented.
- N: maximum number of iterations (loops). This value sets the maximum number of iterations (loops) that can be implemented in the method, for each window.
- nbRep: number of replications. This value represents the number of replications run by the simulation model at each experiment. The simulation model run several replications

Algorithm 4 Opt-sim method 2 Loop

```
1: procedure optSimLoop
2:    M ← max objective value tolerance;
3:    N ← max number of loops;
4:    nbRep ← number of replications;
5:    i ← 0;
6:    objective ← 10000;
7:    minObjective ← 10000;
8:    do (starting of the loop iteration/start running the optimization for each window)
9:       run optimization (Sliding window + Simulated Annealing);
10:      do (start running the simulation for each window)
11:        update simulation database with the optimized schedule;
12:        run simulation replications;
13:        if number of replications run is equal to nbRep then
14:           objective ← objective value calculated by the simulation;
15:         storeBestSolution(objective); (pick the best solution provided by the loop iterations
so far for each window)
16:        end if
17:         while replication number is equal to nbRep; (end of simulation run/end of loop
iteration)
18:      i ← i + 1; (update the loop number)
19:    while objective is greater than M and i less than N (stopping condition for the loop
iterations);
20: end procedure (go to the next window)
```

Algorithm 5 Store Best Solution

```
1: procedure storeBestSolution(objective)
2:   currentObjective ← objective;
3:   if currentObjective is less than minObjective then
4:      minObjective ← currentObjective;
5:      store the schedule;
6:   end if
7: end procedure
```

due to the stochastic nature of some of the variables, and in order to produce statistically significant results.

- i: loops counter. This variable stores the number of iterations (loops) implemented. At each iteration this value is compared with N maximum number of loops. When "i" is higher than N the method stops the iterations (loops) for the current window and it continues to the next window.
- Objective. This variable stores the objective function value of each iteration (loop).
- Minimum objective. This variable stores the minimum value of the objective function between the iterations of each window.

Figure 3.14 gives a visual description of the **Algorithm 4**. The chronological order of the actions between the optimization and the simulation is the following: the optimization process, as described in sections 3.2.1 and 3.2.2, looks at the database for the original schedule and generates an optimized schedule; this optimized schedule is then used by the simulation model (see Appendix B) to run the replications; in the simulation model, variability is implemented by including stochastic variables that represent the uncertainty of the system; the simulation process then identifies the best schedule according to the function *storeBestSchedule(objective)*

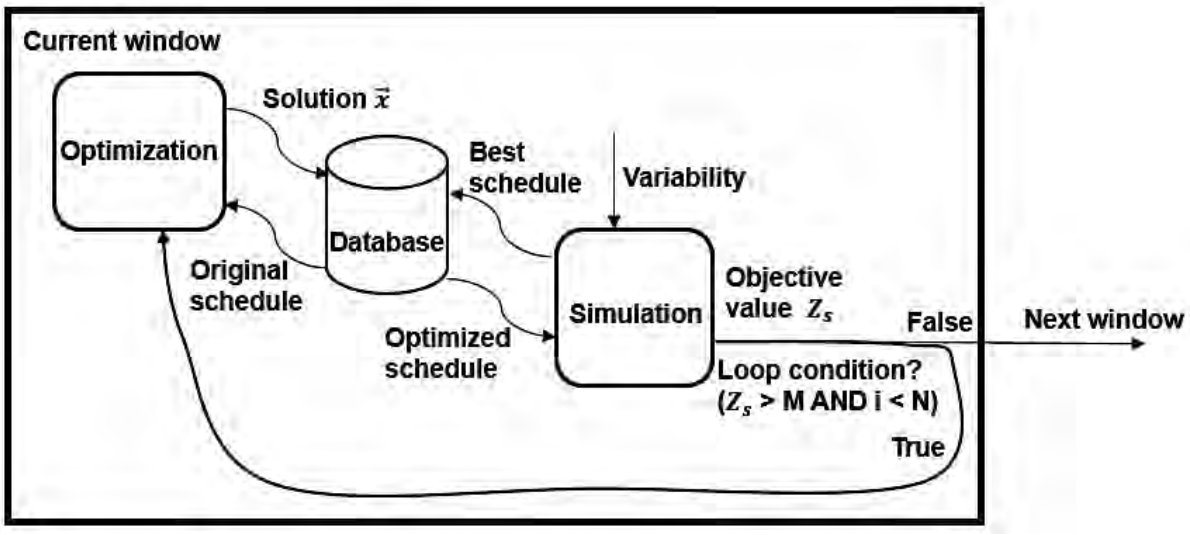


Figure 3.14: Architecture of the Opt-sim Loop

and stores it in the database; the cycle repeats for each iteration (loop) of each window. The database plays an important role since it allows the two models, the optimization and the simulation, to communicate between them and to share data. In this opt-sim framework, every time a new loop is triggered, a new optimized solution is provided, and because of the random nature of the optimization metaheuristic (simulated annealing) of changing decision variables, the new solution will be different than the previous one. As in the Opt-sim method 1, the Opt-sim method 2 tries to improve the solution at each loop by running the optimization model and obtaining a new solution. The new solution is driven either by the randomness of the metaheuristic in finding a new solution or by the fine-tuning of some specific input parameters. The main difference between the Opt-sim method 1 and the Opt-sim method 2 is that the Opt-sim method 2 is implemented to each window, while in the opt-sim method 1, the optimization process considers the time windows, and the simulation process considers the whole time horizon. Figure 3.15 depicts this main difference between the two methods.

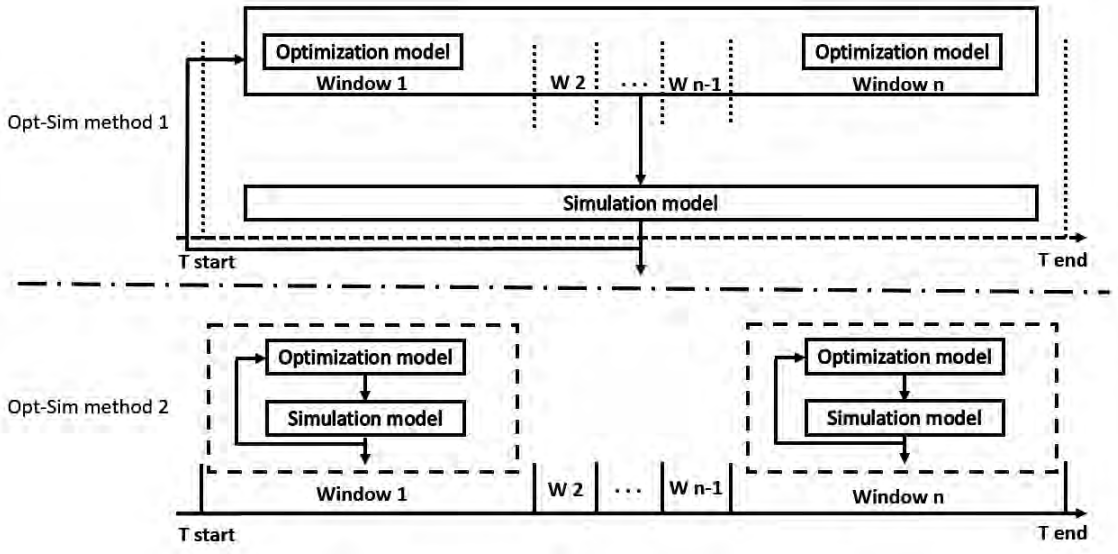


Figure 3.15: Comparison between Opt-Sim method 1 and 2 regarding the time horizon considered.

The parameters M and N of the algorithm play a crucial role as they drive the algorithm efficiency and effectiveness. The **Algorithm 4**, is implemented for a single window; in this algorithm, the storeBestSolution(objective) function, as it is summarized in **Algorithm 5**, is used for choosing the solution with the minimum objective value between the different loops of each window, so eventually the algorithm will come up with an optimized schedule composed by the best schedule of each window. As it was mentioned before, the two parameters M and N play a crucial role in the algorithm, since they are part of the stopping condition at each iteration. Once the stopping criteria is met, the algorithm will move to the next window. For simplicity we refer to this condition as *loop condition.* The *loop condition* checks if the objective value of the current window is greater that M and if the number of loops is less than N. The parameter M defines the objective value tolerance that can be accepted by the algorithm, meaning that a certain number of conflicts can be accepted by a solution. Parameter N defines the maximum number of loops that can be implemented for a window. It is important to choose the right values for these two parameters as they will affect the quality of the solution and also the computational feasibility. For example, a big value of M will accept solution with many conflicts, meaning that that the algorithm will not be effective in finding a good solution, while a small number of M will force the algorithm to keep searching for another solution which might be costly in terms of computing performance. In the same way, choosing a big value for the parameter N will increase the chances of obtaining better solutions, so at least a minimum number of loops should be kept. On the other hand, too many loops might be hampering the performance time of the algorithm.

In the opt-sim method 2, these input parameters are both problem-related and metaheuristic-related:

- Problem-related parameter
    - separation minima increase: applies the same for the opt-sim method 1, see the description in the previous section.
    - objective function weights: we modify the objective function weights $\gamma_{airspace}$ and $\gamma_{airside}$ relative to airspace and ground performance. At each iteration, depending on the value of the airspace conflicts or ground side overload in the objective function, we will increase its weight so that the next solution will focus more on that conflicts. The rationale behind it is that if most of the conflicts/overloads are generated by a specific area (airspace or ground side), then it is better to focus the optimization process on that specific area (by increasing the weight of that specific area). This scenario was implemented only in the Opt-sim method 2, as it is believed that in the window-based iterations, the traffic dynamics are different, as some windows can differ from others in terms of airspace and ground congestions. In a long-term iteration, when simulating the entire day of operations, keeping the objective function weights values fixed for the two airport components would not be beneficial in terms of final results, since some windows would not be affected by these weights and some other windows might be affected negatively. For these reasons, in the Opt-sim method 2, we introduced this extra scenario, so that we can take advantage of its dynamic behavior.

- Metaheuristic (simulated annealing)-related parameter:
    - cooling schedule parameter α: applies the same for the opt-sim method 1, see the description in the previous section.

3.4. Conclusion

In this chapter, it has been introduced the problem tackled in this thesis. This problem has been defined as the airport integrated operations at a macroscopic level. The operations involve airspace and ground. In the specific, regarding airspace operations, landing routes (STARs) were considered, while, regarding ground operations, runway operations (landings and take-offs), taxiway network and terminals were considered. The macroscopic approach addressed mainly the ground components such as taxiway network and terminals, since they were modelled in lower detail, considering them as nodes of a network characterized by a capacity and an average occupancy changing with time. The main assumptions concerned mainly the macroscopic approach, mainly, the way the taxiway and terminals were modelled. Other assumptions concerned the omission of the departing routes in the airspace, the landing routes (STARs) which are enough vertically-spaced between each other and they do not interfere with departing routes (SIDs), aircraft flying in the airspace have fixed acceleration. The optimization model was presented with its features such as decision variables, constraints, and objective function. Moreover, the sources of uncertainty were introduced and described.

In the second part of the chapter, the solution approach has been presented, introducing the optimization process composed by the sliding window approach and the adapted simulated annealing metaheuristic for solving this specific problem. Finally, the simulation approach based on discrete event simulation has been introduced, providing description about the approach itself and about its advantages, and giving a general view of the main features of general purpose simulation software. Then, the simulation model of the airport macroscopic operations was presented.

In third part of the chapter, two methods for combining optimization and simulation have been proposed and described. In the first method optimization was used in order to find a (sub)optimal solution and simulation was used for validating the feasibility of this solution applied in a system where uncertainty is also considered. The second method aims at continuously improving the robustness of the optimized solution in each window of the sliding window approach by running the optimization and the simulation. At each window, many iterations (loops) are implemented until an acceptable solution is found. In the two methods, at each iteration, some of the input parameters are fine-tuned, for which, two categories of input parameters have been defined.

In the next chapter, the case study considered for this thesis will be introduced. Moreover, experiments will be conducted, and results will be showed.

Chapter 4

Experiment on a real case study: Paris Charles de Gaulle Airport

Paris Charles de Gaulle (PCDG) Airport is one of the major airports in Europe, for its size, passengers transported and air traffic movements. It is the hub of the French legacy carrier Air France, and according to EUROCONTROL in 2017, it transported 69.5M of passengers and carried 476K air traffic movements (EUROCONTROL, 2018b). The data used in the experiments refers to a flight schedule of an entire day of operations at the airport. In total there were 1116 movements; in Table 4.1 they are classified based on the type of movement.

Table 4.1. Number of movements classified based on the type of movement

Type of movement	Number of movements
Arrival	554
Departure	563
Total movements	1116

- *Airport TMA and runway system.* PCDG Airport has 4 parallel runways, that operate as independent runways, i.e. the operations of one runway does not interfere with the operations of the other runway. In real operations two of the runways are used only for landings (runway 26L and 27R) and other two only for departures (runway 26R and 27L). In the TMA, four landing routes are identified, two coming from the south and two coming from the north. These routes are identified based on airspace surveillance records, and refer to west configuration landing routes, as it can be seen in figure 4.1. Depending on which of the two landing runways is used, in any of the four available landing routes, it is possible to take two different routes. These routes are spaced by a safe vertical distance, so to have a total of 8 trajectories available.

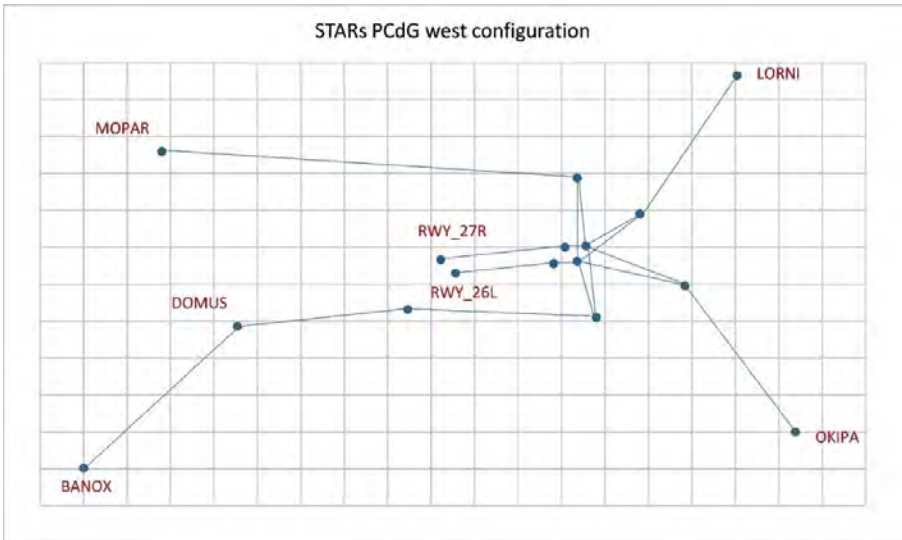


Figure 4.1: Landing routes at Paris Charles de Gaulle Airport, west configuration.

- *Airport airside.* PCDG Airport is constituted by three terminals and by a complex taxiway network, as it is depicted in figure 4.2. Following the macroscopic approach for modeling the airport airside, terminals and taxiway network are modeled considering only their capacity and service time. The main assumption for ground operations concerns the taxiway routes, where potential aircraft conflicts, due to physical position on the taxiway, were not detected; another assumption is that aircraft could park at any gate of the terminal, regardless of the aircraft size. Tables 4.2 and 4.3 show the capacity of these components and the average taxiway times used in both optimization and simulation models. The values of capacity for the airside components, taxiway network and terminals, imposed by the users, while the values of taxi time are calculated based on data surveillance. Regarding the runways, it was assumed an occupancy time for both landing flight and departing flight of 60 seconds.

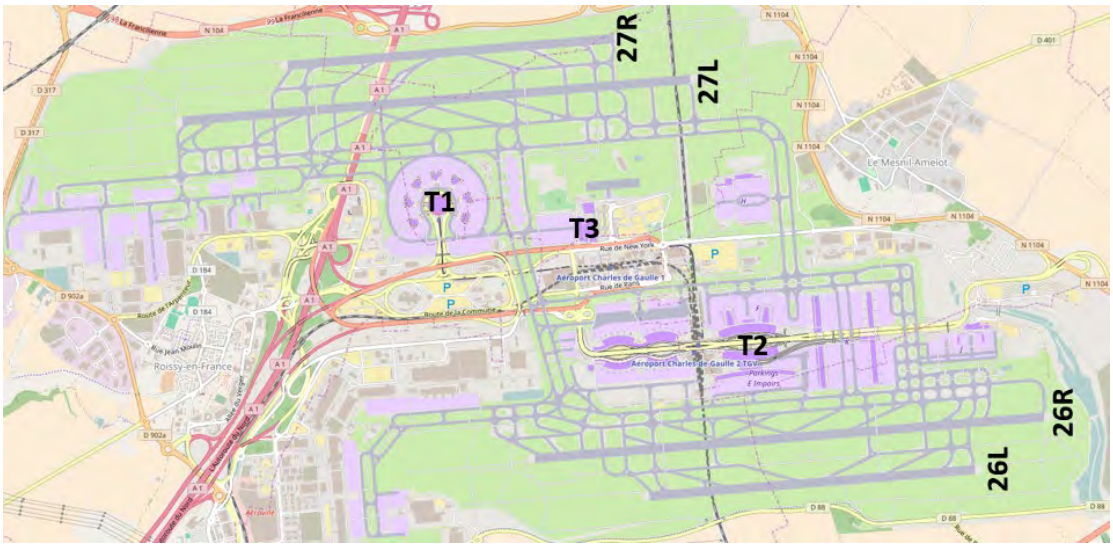


Figure 4.2: Paris Charles de Gaulle Airport airside layout.

Table 4.2. Airside component capacity

Ground component	Capacity value
Terminal 1	11
Terminal 2	91
Terminal 3	57
Taxiway network	18

Table 4.3. Average taxi time (s)

	Landing runways (taxi in time)		Departing runways (taxi out time)	
Terminals	26L	27R	26R	27L
Terminal 1	651	439	859	1197
Terminal 2	523	764	861	835
Terminal 3	599	569	1240	1139

4.1. Implementation of Opt-Sim method 1

In this section, scenarios and results of the Opt-Sim method 1 are presented. For each scenario three iterations were run by setting different values for the optimization model input parameters. In this method, the simulation model was implemented for evaluating the objective function by including uncertainty inherent to the operations. The purpose of the method is to validate the feasibility of the optimized solution and then improve it. The simulation model, in the Opt-Sim method 1, does not consider the sliding window approach, instead it runs the entire horizon time of the problem.

Results are represented in terms of average total objective function value. In order to make a thorough analysis, the performance of each component of the objective function such as airspace, ground components were evaluated. The simulation model runs 50 replications for each iteration, thus, results show average values in order to reveal the trend and standard deviation values in order to have insights about the impact of the uncertainty.

4.1.1. Scenarios and results

For testing the Opt-sim method 1, different scenarios were defined, and they were based on the input parameters of the optimization model to be fine-tuned. Two different parameters were used, one related to the metaheuristic, the cooling schedule, α; and one related to the optimization model formulation, the separation minima in the airspace and at the runway. The cooling schedule parameter, α, is responsible for the convergence toward the global optimal solution. At each iteration this parameter is increased so that the algorithm will search broadly in the state space. Regarding the constraint relaxation, at each iteration, the separation minima between consecutive aircraft in the airspace and at the runway was enlarged. By relaxing this constraint, the optimization model will generate a solution which will be potentially conservative, however, due to the larger separation between aircraft, the effect of the uncertainty on airspace and runway conflicts could be absorbed. The values used for the two parameters are shown in Table 4.4.

Table 4.4. Values of the input parameters of the optimization model for the two scenarios. In the first run the input parameters are set as default, while in the three iterations separation minima constraint relaxation gradually increases from 10% to 30%, and the cooling schedule parameter α gradually increases from 0.97 to 0.99.

Scenario/Parameter	First run	1st Iteration	2nd Iteration	3rd Iteration
1st Scenario/Separation minima constraint relaxation	0%	+10%	+20%	+30%
2nd Scenario/Cooling schedule parameter α	0.96	0.97	0.98	0.99

Three scenarios were tested:

- Default scenario. In this scenario, the optimization was run without modifying any of its input parameters, in Table 4.4 in the column “First run”, the values of the parameters used in the optimization are shown.
- Separation minima constraint relaxation scenario (Scenario 1). In this scenario, we tried to improve the optimized solution by fine tuning one of the input parameters of the

optimization model. In this case, the value of separation minima in the airspace and at the runway was changed. This value represents a constraint in the optimization model formulation, therefore, by fine tuning it, we are relaxing a constraint. In this case, at each iteration, the optimization model was run by relaxing the constraint according to the values given in the second row of Table 4.4. Following this approach, in the first iteration the separation minima will be enlarged by 10%, in the second iteration by 20%, and in the third iteration by 30%. By relaxing this constraint, we will achieve a more conservative solution because we will obtain a sequence of aircraft which will be spaced with a separation larger than the standard one, however, this might reduce the effect of uncertainty on the airspace and runway conflicts.

- Cooling schedule parameter, α, scenario (Scenario 2). In this scenario, the input parameter of the optimization model that was fine-tuned was the cooling schedule parameter, α, of the simulated annealing metaheuristic. For each iteration, this value was increased by 0.01, starting from the default value of 0.96 and reaching the values of 0.97, 0.98, and 0.99 in the first, second and third iteration, respectively. Increasing the cooling schedule, α, as already explained in section 3.3.1, will improve the search in the state space, and as a consequence, it will generate a new optimized solution that can potentially handle the uncertainty. In this scenario, the separation minima values between consecutive aircraft are kept as the original ones shown in Table 3.2.

First, the original flight schedule was run without the implementation of the optimization model in order to utilize it as a baseline for the comparison with the optimized scenarios. In Table 4.5 the result for the non-optimized scenario in terms of objective function is shown by breaking it down into all its components.

Table 4.5. Value of the objective function broken down in all its components. Original schedule.

Objective function component		Objective function subcomponent	Individual value	Aggregated value
Airspace conflicts		Link conflict	1114	1575
		Node conflict	461	
Runway conflicts				381
Taxiway network overload		Max Overload	8	8.14
		Avg Overload	0.14	
Terminal overload	T1	Max Overload	2	2.12
		Avg Overload	0.12	
	T2	Max Overload	4	4.3
		Avg Overload	0.3	
	T3	Max Overload	2	2.2
		Avg Overload	0.2	
Total conflicts				1972.76

Looking at the last row of Table 4.5, the total value of the objective function is 1972.76. Most of the conflicts were detected in the airspace, where with 1114 conflicts on links and 461 conflicts on nodes. Runway was also affected, with 381 conflicts. Taxiway network had an overload peak of 8 and an average overload of 0.14, which shows that the system is always at the edge of its capacity. Regarding the terminals, there were observed overload peaks of 2, 4 and 3 for terminal 1, 2 and 3, respectively. The average overloads were 0.12, 0.3 and 0.2 for

terminal 1, 2 and 3, respectively. From these results, it can be seen that terminal 2 was the most congested in terms of both peak and average overload, however, terminal 1 and 3 suffered from congestion as well. Overall, the results revealed that the airspace was more congested than the ground, meaning that the capacity of the ground was able to absorb the traffic better than the capacity of the airspace. Conflicts in the airspace routes (links and nodes) were transferred to the runway, which was also highly overloaded. Therefore, the optimization process needs to pay a lot of its effort in solving airspace conflicts, being a critical area to focus on.

4.1.1.1. Default scenario results

In the first scenario of the methodology, the optimization model is set with default values, e.g. cooling schedule α, separation minima for airspace and runway, among others. It generates an optimized solution in the form of a flight schedule that is used as input for the simulation model. The simulation model, in turn, runs the optimized schedule by including sources of uncertainty to obtain a new value of the objective function. The sources of uncertainties involved in the operations are:

- entry time in the TMA deviation, due to speed and human (pilots) factors,
- taxiway time deviation, due to layout complexity, congestion, speed and human (pilot) factor,
- off-block time deviation, due to ground handling procedures and communications with the aircraft cockpit.

In table 4.6 the values attributed to them are shown.

Table 4.6. Uncertainty sources and their respective values.

Uncertainty source	Value
Entry time in the TMA deviation	[-30s., +30s.]
Taxiway time deviation	[-10%, +10%]
Push-back time deviation	[-30s., +30s.]

The optimization model generates a conflict-free solution, however, when this optimal solution is simulated, conflicts appear again. Table 4.7 shows the conflicts of the different components of the problem such as airspace, runway, taxiway network and terminals. Moreover, the results in Table 4.7 shows also the number of aircraft involved in conflicts in the airspace, which was possible to extract thanks to the simulation model. It shows also the conflicts at runways differentiating it by landing runway (runway in) and departing runway (runway out). The number of aircraft involved in conflicts could be used as an indicator of the congestion in the airspace. As it can be seen from Table 4.7, when including uncertainties in the system, we create new conflicts, especially for the airspace and the runway out components. In Table 4.7 are also shown minimum, average, maximum and standard deviation values since there were run 50 replications for each scenario. Looking at the average values, it can be noticed that terminals are not affected by uncertainty since they do not have any overload, while the taxiway network is slightly affected with a max overload of 2 and an avg overload of 0.005. This can be explained by the fact that the turnaround time, which ranges approximately between 30 minutes and 2 hours, absorbs the deviation of taxi times and pushback times, reducing the effect of the uncertainty. The component which is the most affected is the runway out with 81.32 conflicts,

followed by the airspace with 31.62 conflicts and then the runway in with 9.42 conflicts. Aircraft involved in airspace conflicts are counted as an average of 13.26, comparing this value with the value of airspace conflicts, it can be deducted that each aircraft is subject to roughly 2.5 airspace conflicts. The total number of conflicts generated from the simulation model were 124.36 in average. These initial experiments pointed out that using optimization techniques alone to deal with such problems would not be enough for supporting decision makers in real operations. The deterministic nature of the optimization model represents a relevant limitation, as it was highlighted by the number of conflicts found after including uncertainty in the system by the simulation model.

Table 4.7. Value of the objective function broken down into the different components using the optimized schedule.

Objective function component	Objective function subcomponent	Values			
		Min	Avg	Max	St. Dev
Aircraft involved in airspace conflicts		7	13.26	19	2.67
Airspace conflicts		13	31.62	53	8.02
Runway conflicts	Runway in	4	9.42	16	2.77
	Runway out	61	81.32	93	7.2
Taxiway network overload	Max overload	2	2	2	0
	Agv overload	0.0005	0.005	0.01	0.003
Terminal overload	Max overload	0	0	0	0
	Avg overload	0	0	0	0
Total conflicts		80.00	124.36	164.01	18

4.1.1.2. Results from scenario 1 and 2

In this section, the results from scenario 1 and scenario 2 are presented and discussed. In order to make a thorough analysis, we consider two additional indicators besides the number of conflicts: the average conflict size violation (%), which express the magnitude of the conflicts detected and the number of conflicts falling within a certain interval of separation violation percentage, which represent the severity of the conflicts. Moreover, the results are divided into the different components of the objective function: airspace, runway in, runway out, taxiway network and terminals.

The average conflict size violation, is an indicator which expresses, in percentage, to which extent the separation minima was violated. For instance, if the separation minima, S_{ij}, between two consecutive aircraft i ad j, is 120 seconds, and the actual separation minima, D_{ij}, between them is 100 a conflict is detected. The violation of the separation minima is 20 seconds, which represents the 16.6% of the total separation minima. In the average conflict size indicator, we calculate the average of all the conflict violation, so that we can have information about the severity of the conflicts, obviously, the smaller the percentage the less severe are the conflicts.

The number of conflicts falling within a certain interval of separation violation, expresses how frequent big or small conflict violations happen. Linking to the previous example, the conflict detected between aircraft i and j, falls within the separation violation interval of 10% and 20%.

We have defined a separation violation interval scale that classifies three different level of conflict severity in: *"low severity"* for intervals between 0% and 5%, *"medium severity"* for intervals between 5% and 30%, and *"high severity"* for intervals between 30% and 100%.

The aircraft involved in airspace conflict is a measure of the level of congestion in the airspace, thanks to the simulation model which was able to track this type of information. In Figure 4.3 the number of aircraft involved in airspace conflicts is shown for both scenarios 1 and 2. The red line represents scenario 1 (separation minima increase), while the blue line represents scenario 2 (α increase). The first point from left, displayed in the graph of Figure 4.3, represents the default scenario. The rest of the values are displayed according to the scenario and according to the iteration, having three iterations for each scenario. In the default scenario the average number of aircraft involved in airspace conflicts is 13.26. Between scenario 1 and 2, the best is the first, since it is able to lower the value of the performance of the default scenario from 13.26 to 6.1. In each iteration of scenario 1, the number of aircraft involved in airspace conflicts progressively decreases as the separation minima relaxation increases as well. We obtain values of 8.58, 7.44 and 6.1 for scenario 1 when separation minima have been increased by 10%, 20%, and 30%, respectively. The performance of scenario 2 are not as good as scenario 1. The iteration with α equal to 0.97 has the worst performance reaching 14.78. The next two iterations, with α equal to 0.98 and 0.99, have better performance as they are able to lower the values up to 11.68 and 12.02. Overall, scenario 1 gained its best improvement when compared to the default scenario in the third iteration, with 30% of separation minima increase, achieving a performance improvement by 53.99%, while the best performance of scenario 2 was detected in the second iteration, achieving an improvement by 11.91%.

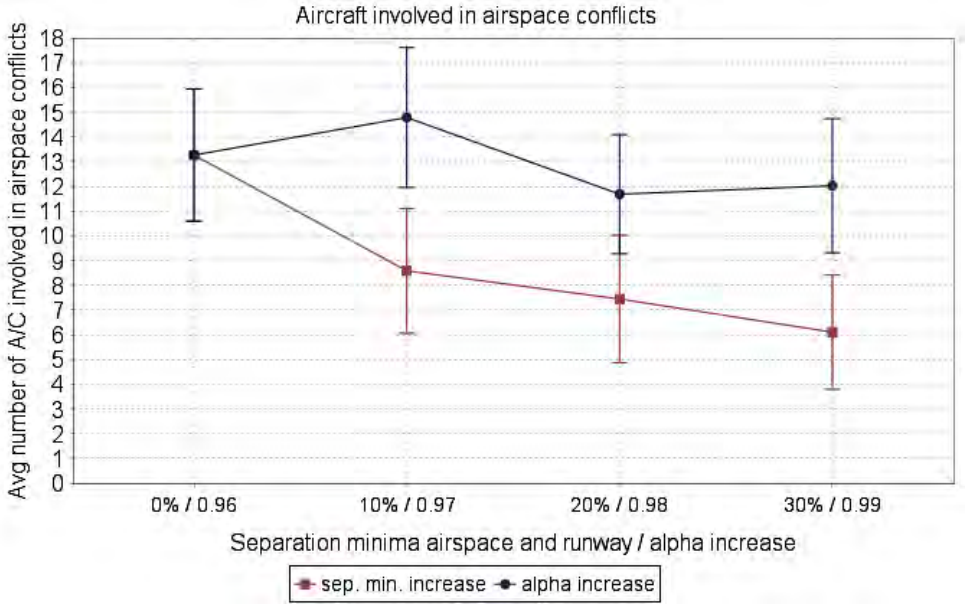


Figure 4.3: Average number of aircraft involved in airspace conflicts for scenario 1 (separation minima increase/red line) and scenario 2 (alpha increase/blue line)

Figure 4.4 shows the number of airspace conflicts for the default scenario, scenario 1 and 2; here we can see that scenario 1 outperforms scenario 2 at every iteration. The default scenario has 31.62 conflicts, while scenario 1 is able to lower the conflicts up to 9.08 in the third iteration, and scenario 2 lowered the airspace conflicts to 28.92 in the second iteration. Regarding airspace

conflicts, scenario 1 was more effective than scenario 2, obtaining an improvement by 71.28%, while scenario 2 obtained an improvement by 8.53%.

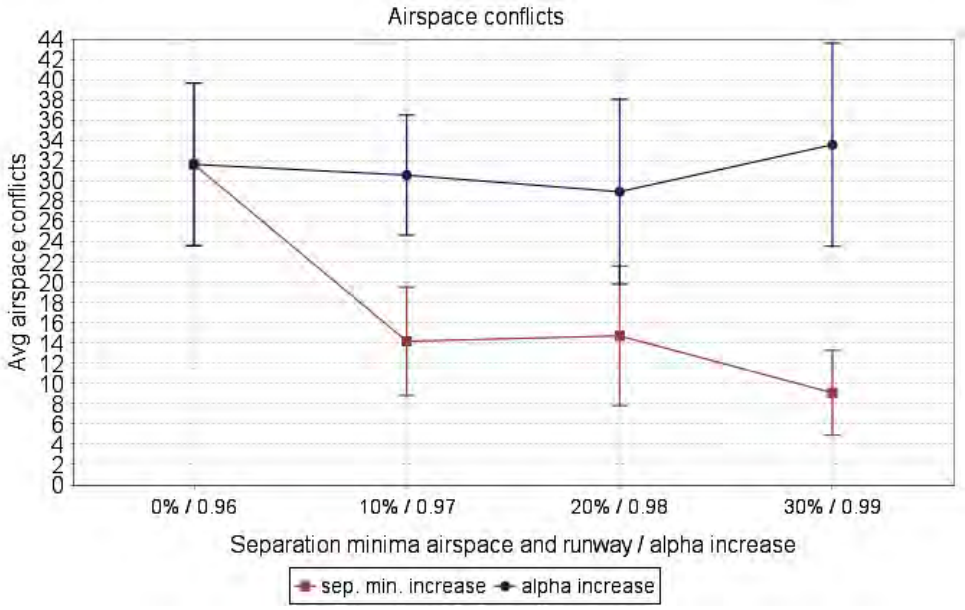


Figure 4.4: Average number of airspace conflicts for scenario 1 (separation minima increase/red line) and scenario 2 (alpha increase/blue line).

In Figures 4.5 and 4.6 are shown the number of conflicts falling into one of the conflict size intervals for both scenario 1 and 2 compared to the default scenario. This performance indicator serves as a measure of severity of conflicts. This measure is based on the separation minima violation extension of conflicts. In a close-to-ideal scenario, conflict would violate the separation minima to a small extent, so you might find values concentrated on the left end side of the chart. On the other hand, the concentration of values to the right side of the chart means that conflicts are severe since they violate the separation minima to a big extent. In Figure 4.5, scenario 1 is depicted, where the red line represents the default scenario, the blue line the first iteration, the green line the second iteration and the yellow line the third iteration. From this graph it can be seen that the default scenario is always outperformed at every iteration, and that the best iteration is the third. In Figure 4.6, the second scenario is depicted, here, as it was for the first scenario, the default scenario is always outperformed at every iteration. Among the three iterations, no one of them clearly outperforms the other, because they are all at the same level. Comparing scenario 1 with scenario 2, we can find, in scenario 2, more conflicts falling into 5-10%, 10-20% conflict size intervals than in scenario 1. It can be noticed also that scenario 2, for all the three iterations, has an average of 3.5 conflicts that fall in the conflict size interval 75-100%, the most severe, while scenario 1 has one conflict falling in the 75-100% interval in the first and second iteration, and zero conflicts in the third iteration.

Scenario 1 confirmed to be better than scenario 2 also in terms of average conflicts size percentage. In this context, the average conflict size is calculated as the average of all the airspace separation violation conflicts, measured as a percentage. This performance indicator provides a general measure about the magnitude of the conflicts. In Figure 4.7, it can be seen that scenario 1 improved the performance from the default scenario, first point from left, which has a value of 32.24%, up to 10.68% found in the third iteration.

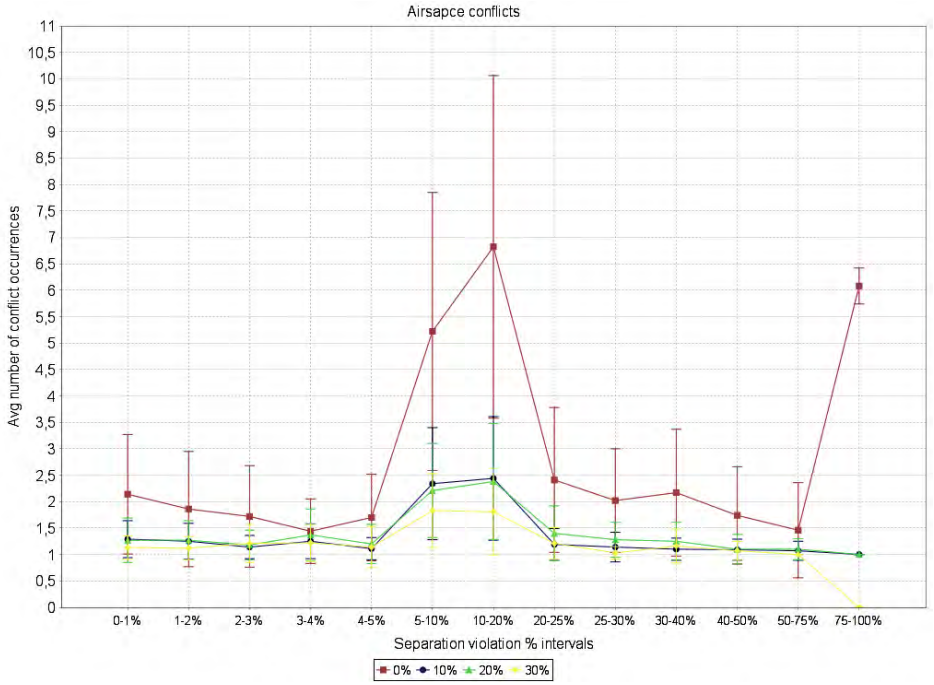
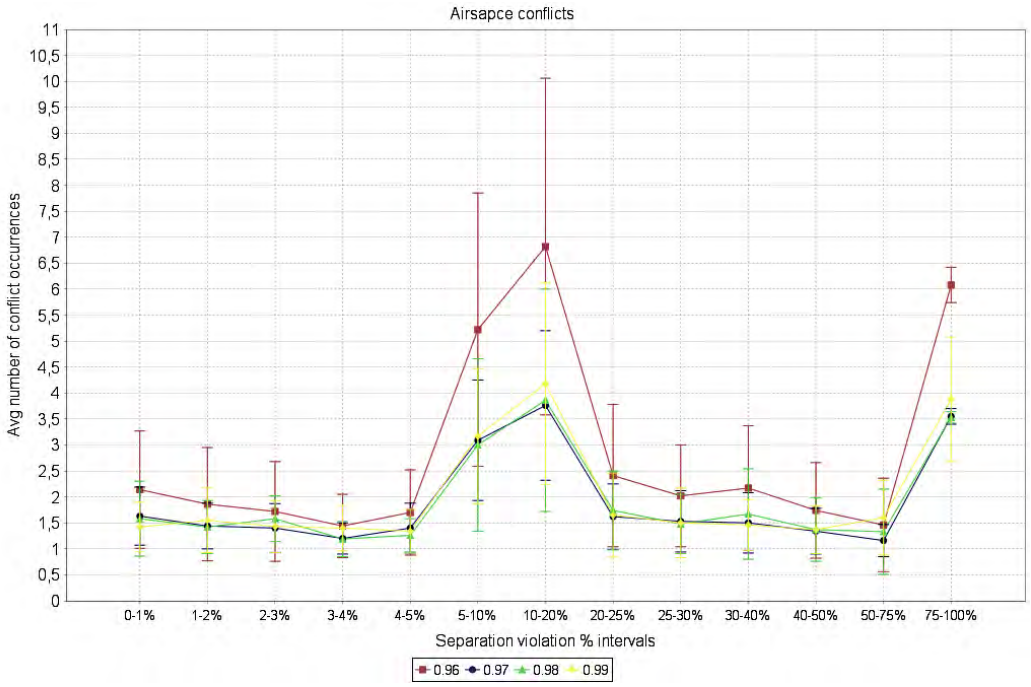


Figure 4.5: Number of airspace conflicts that are in a specific conflict size interval (scenario 1).



(b)

Figure 4.6: Number of airspace conflicts that are in a specific conflict size interval (scenario 2).

Scenario 2 is not effective in improving the performance, since the best performance is found in the first iteration with a value of 31.79%.

In Figure 4.8 average values of *"runway in"* conflicts are displayed; the default scenario has 9.42 conflicts, while scenario 1 obtains 6.28, 4.42, and 3.06 conflicts, for the first, second and third iteration, respectively. Scenario 2 has 11.44, 8.54 and 8.64 average values of conflicts in

the first, second and third iteration, respectively. Scenario 2 confirmed to be not effective in improving the performance from one iteration to the other, even worsening the performance in the second iteration. Scenario 1, on the opposite, improved the performance from the first to the third iteration, proving to be effective.

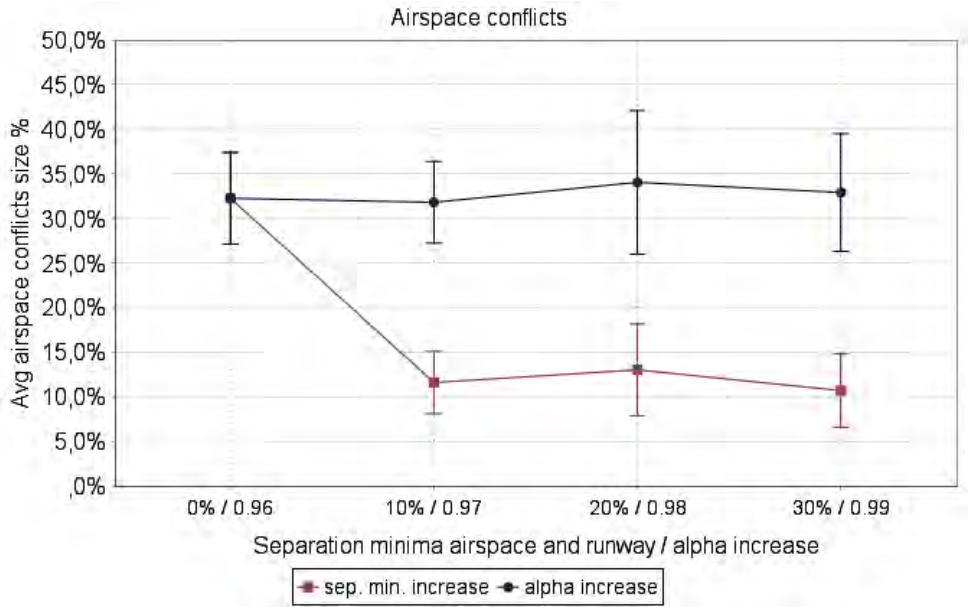


Figure 4.7: Average airspace conflict size percentage.

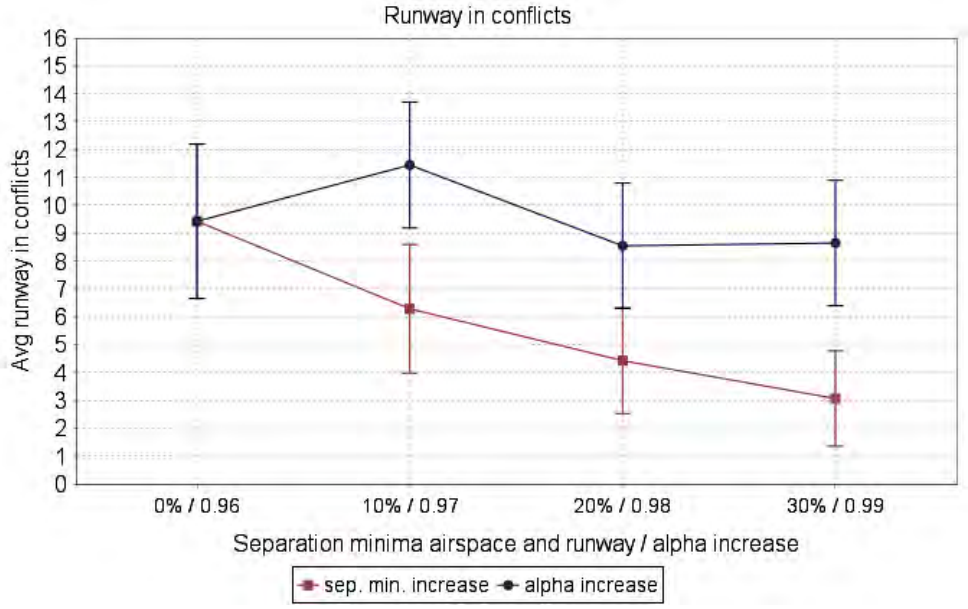


Figure 4.8: *"runway in"* number of conflicts.

Figures 4.9 and 4.10 show the number of *"runway in"* conflicts that fall within the different conflict size intervals for scenario 1 and scenario 2. As it can be noticed, both scenarios outperform the default scenario, and both scenario seems very close in terms of performance. *"Runway in"* conflicts do not seem to be too severe, as there are no conflicts the fall within the range of 50-75% and 75-100% separation minima violation, and the rest of the conflicts are spread among the other conflict size intervals. Furthermore, in scenario 1 there are no conflicts

falling in the conflict size interval 40-50%, while the rest of the conflicts in the other intervals range from 1 to 1.6. In scenario 2, conflicts are spread among the different conflict size intervals, ranging between 1 and 2.08, however, in the highest conflict size intervals (50-75% and 75-100%) there are not conflicts.

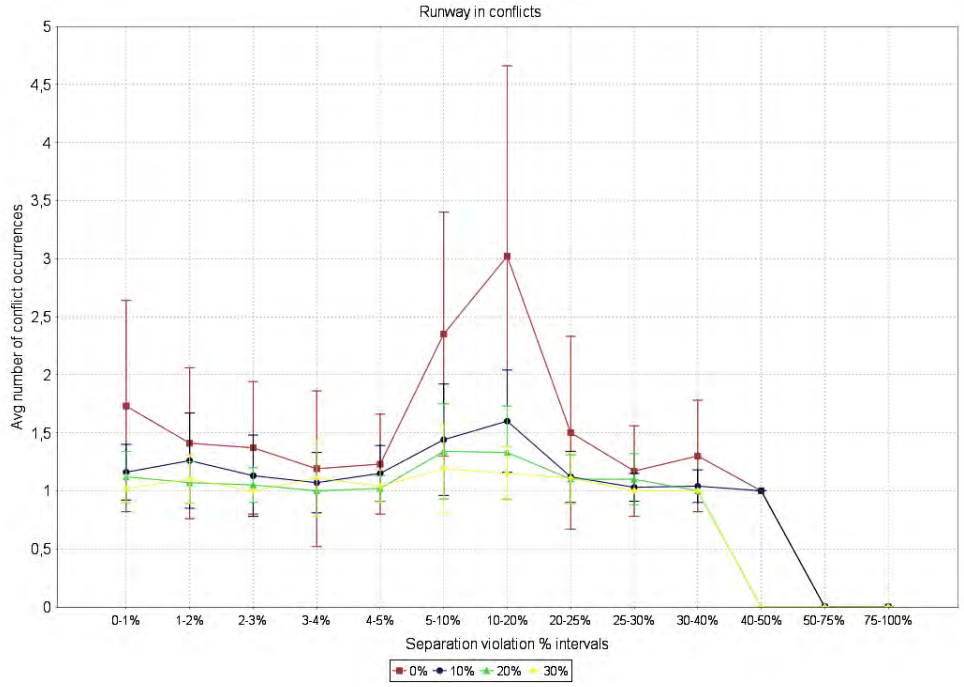


Figure 4.9: Number of "*runway in*" conflicts that are in a specific conflict size interval (scenario 1).

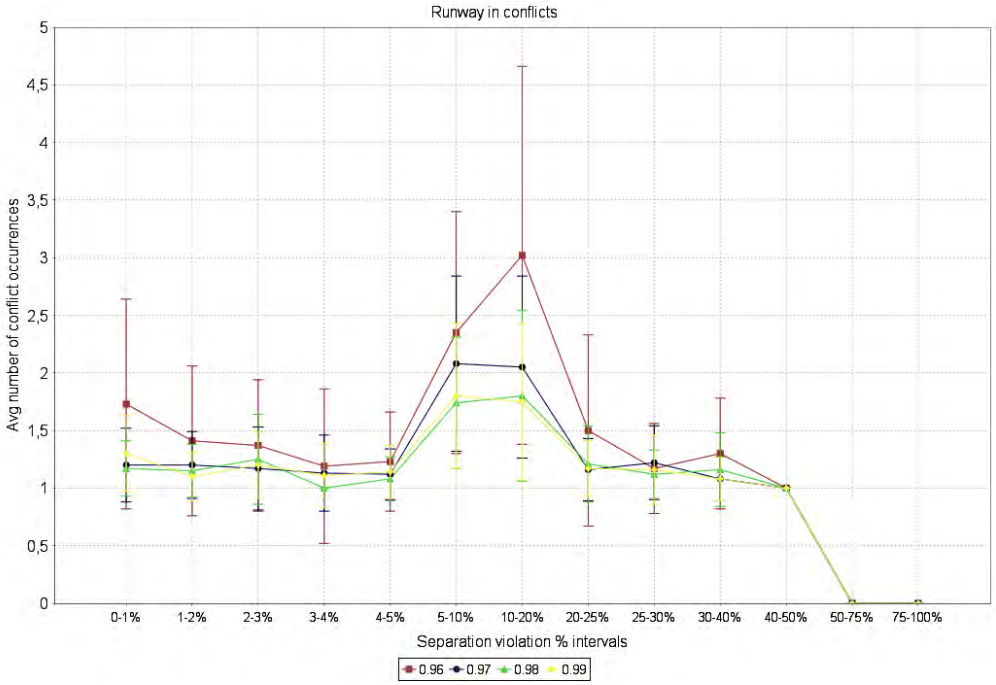


Figure 4.10: Number of "*runway in*" conflicts that are in a specific conflict size interval (scenario 2).

"*Runway in*" average conflict size are shown in Figure 4.11 for both scenario 1 and 2 and also for the default scenario. The improvements obtained by the two scenarios compared with the

default scenario are not big, as scenario 1 improves the performance by about 2%, while scenario 2 is not able to improve the performance at all.

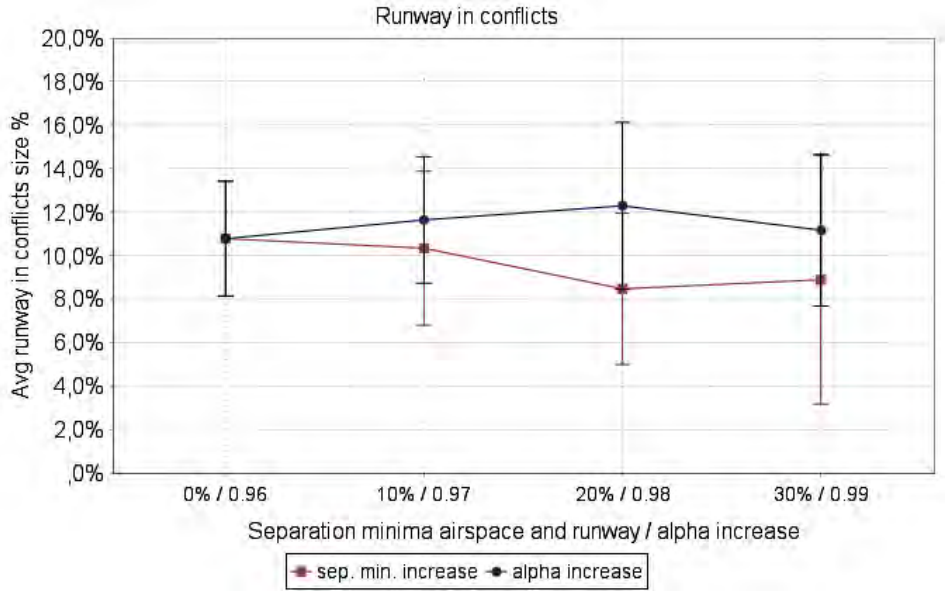


Figure 4.11: Average "*runway in*" conflict size percentage.

When analyzing "*runway out*" conflicts, it can be noticed that they have the biggest share in the total number of conflicts. Figure 4.12 shows the average number of conflicts for the "*runway out*" comparing scenario 1 and scenario 2. The scenario that performs the best is scenario 1, however, the improvements compared to the default scenario is not big. The default scenario obtained 81.32 conflicts, while scenario 1 obtained its best results in the second iteration with 78.98 conflicts. Scenario 2 is not able to improve the performance compared to the default scenario, on the opposite, it worsens the performance, achieving as its best result in the second iteration with 84.4 conflicts.

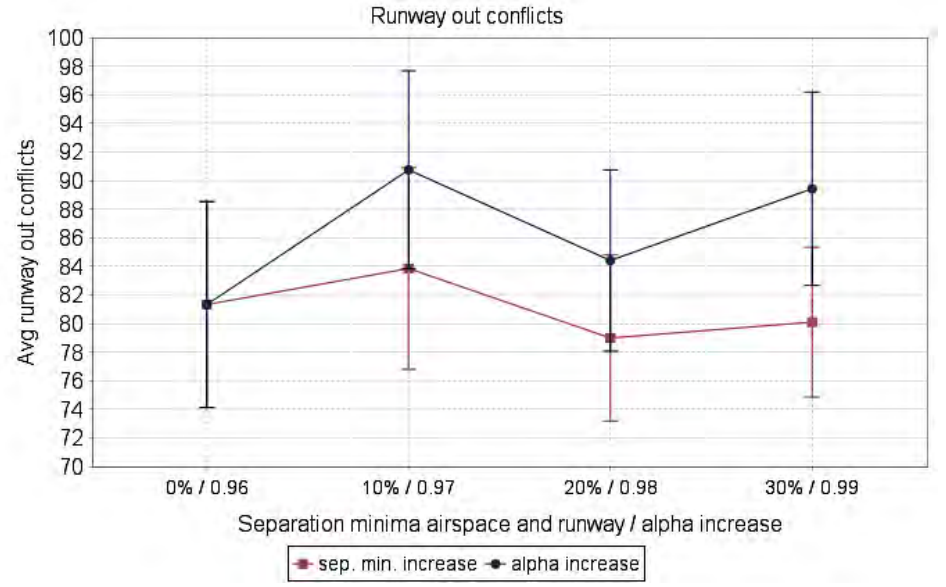


Figure 4.12: "*runway out*" number of conflicts.

Figures 4.13 and 4.14 show, for both scenarios, the number of "*runway out*" conflicts falling in the different conflict size interval. These results highlight the severity of "*runway out*" conflicts, as most of the conflicts violate the separation minima by 50 to 100%. The average conflicts size percentage for "*runway out*" conflicts, as it can be seen in Figure 4.15, lies around 50% for both scenarios, which is very high if compared with "*runway in*" and airspace conflicts where conflict size percentages were around 10% as it was obtained by their best scenarios. The main reason for this big number of conflicts and also the high conflict severity can be found in the magnitude of the uncertainty that affects ground side, identified in the simulation model as deviation from off block time and taxi time out. The values of separation minima for the "*runway out*" range between 60 and 111 seconds, while values of deviation from off block time range between minus and plus 30 seconds. Considering also that the deviation in taxi out time ranges between minus and plus 10% of the average taxi out time, which in turn ranges between 835 and 1240 seconds, it can be clearly understood how this can affect severely the "*runway out*" conflicts performance. In real operations, the consequence of these high number of conflicts and conflicts size for the "*runway out*", are translated into number of aircraft queuing up and waiting at the departing runway entry. This results implies that the "*runway out*" is very sensitive to uncertainty and, on the opposite, it is not sensitive to the different scenarios, therefore the output does not change much between the different scenarios. Moreover, it is important to notice that in the optimization model, the severity of conflicts indicator is not considered, as conflicts are computed by counting them every time a conflict is detected, however, the optimization model is able to provide a conflict-free solution at each iteration.

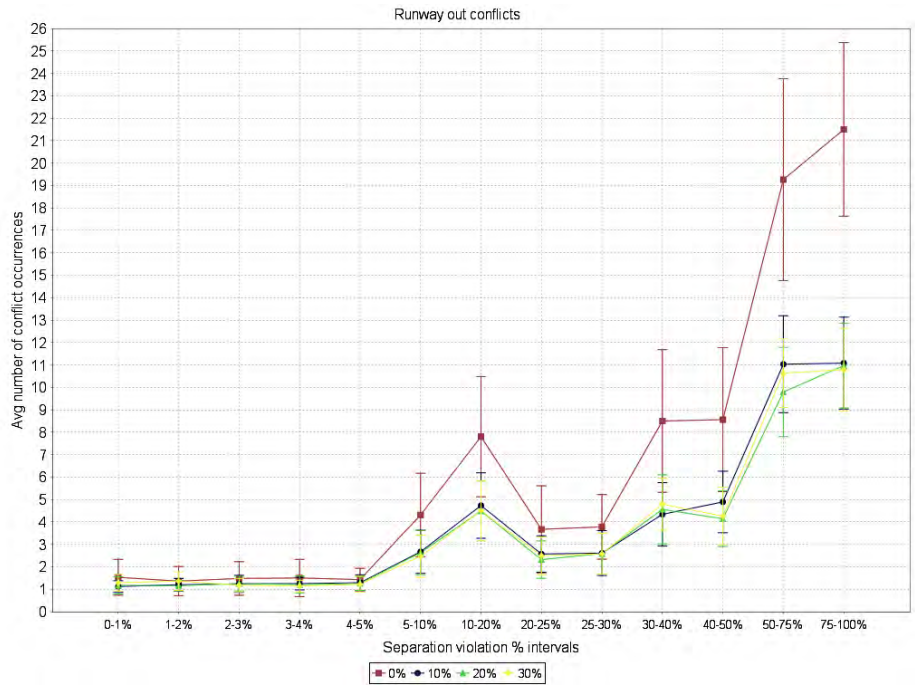


Figure 4.13: Number of "*runway out*" conflicts that are in a specific conflict size interval (scenario 1)

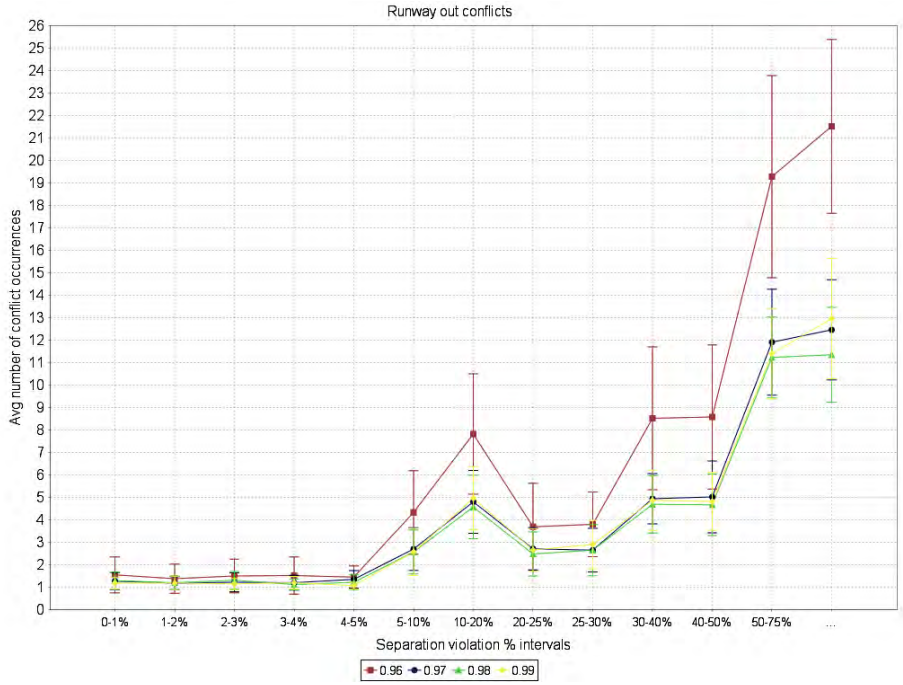


Figure 4.14: Number of *"runway out"* conflicts that are in a specific conflict size interval (scenario 2).

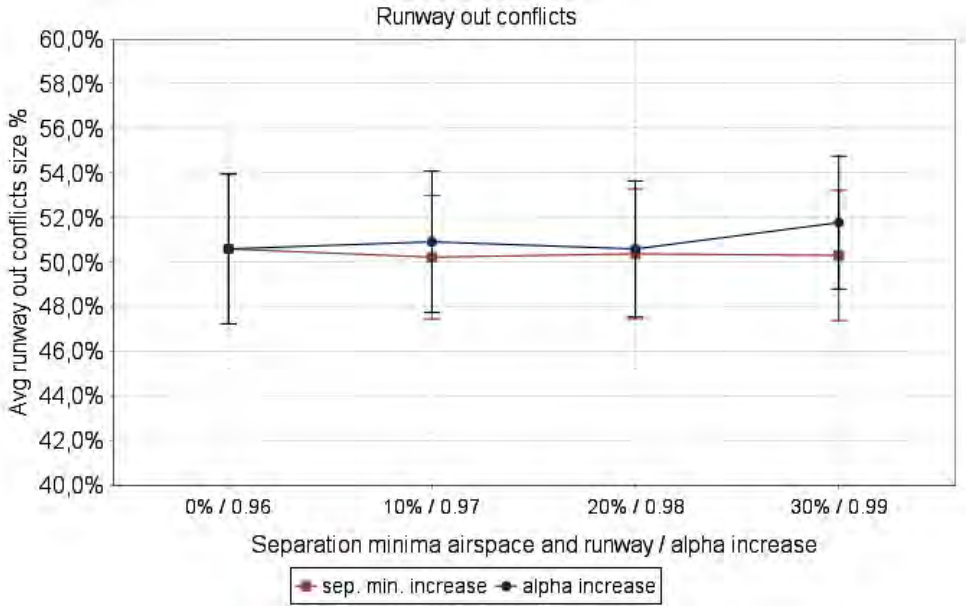


Figure 4.15: Average *"runway out"* conflict size percentage.

Results for taxiway network and terminals are depicted in the two charts of Figures 4.16 and 4.17. Concerning taxiway and terminals conflicts, values are given by the sum of the maximum overload (the integer part), and the average overload (the decimal part), as explained in equations (4), (5), and (6) in section 3.1.3.2[3]. Regarding taxiway network overload, the default

[3] $Max\ overlad = \begin{cases} max_{t \in T}(O_t - C),\ if\ O_t > C \\ 0, otherwise \end{cases}$ (4)

$Average\ overload = \frac{overload_t}{T}$ (5)

$overload_t = \begin{cases} O_t - C,\ if\ O_t > C\ \forall\ t \in T \\ o, otherwise \end{cases}$ (6)

scenario with 2.005 overload value was the best scenario. Scenario 1 and 2 were not able to reduce conflicts, and on the contrary their performance were slightly worse, with results ranging between 2.07 and 2.29 in scenario 1 and ranging between 2.09 and 2.13 in scenario 2. The uncertainty included in the taxi times, generates overloads in all the scenarios, at a small extent. In real operations, taxiway network overload, means that the taxiway is congested and, as a consequence, the flow of aircraft on the taxiway will not be smooth, and bottleneck are most likely to happen in some of the hot spots of the taxiway network. From these results, it can be also noticed that the average overload for taxiway network is small, which means that the overload has taken place for a small amount of time. Regarding terminal capacities, all scenarios except the second iteration of scenario 1, show no overload, however, both scenarios are able to remove overload. In general, as the results reveal, taxiway network and terminals performance are affected by uncertainty to a small extent.

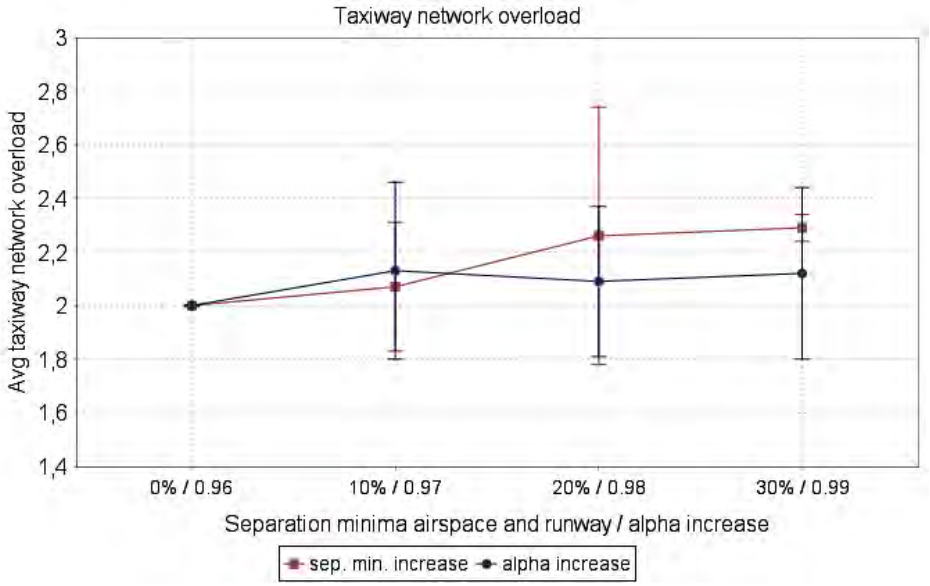


Figure 4.16: Number of taxiway network conflicts.

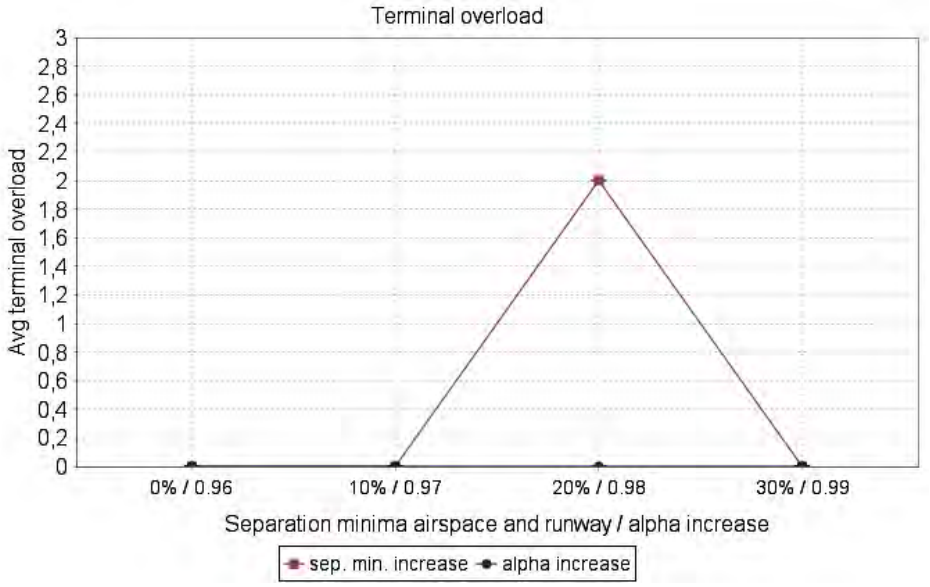


Figure 4.17: Number of terminal conflicts.

In Figure 4.18 the total number of conflicts are depicted. From this graph, it can be noticed that scenario 1 outperforms the default scenario and scenario 2, obtaining the best results in the third iteration, by enlarging the separation minima by 30%. Results show that scenario 1 progressively improve the performance at each iteration, compared to the default scenario with 124.36 conflicts, by reaching 106.35, 102.35 and 95.34, in first, second and third iteration, respectively. Overall, scenario 1 is able to reach its best improvement in the third iteration, by 23.33%, when compared to the default scenario. On the other hand, scenario 2 reveals not to be able to improve the performance, as the total number of conflicts in two of the three iterations are bigger that the default scenario. Scenario 2 outperforms the default scenario, only in the second iteration, by obtaining the value of 123.95, however, this does not represent a big improvement.

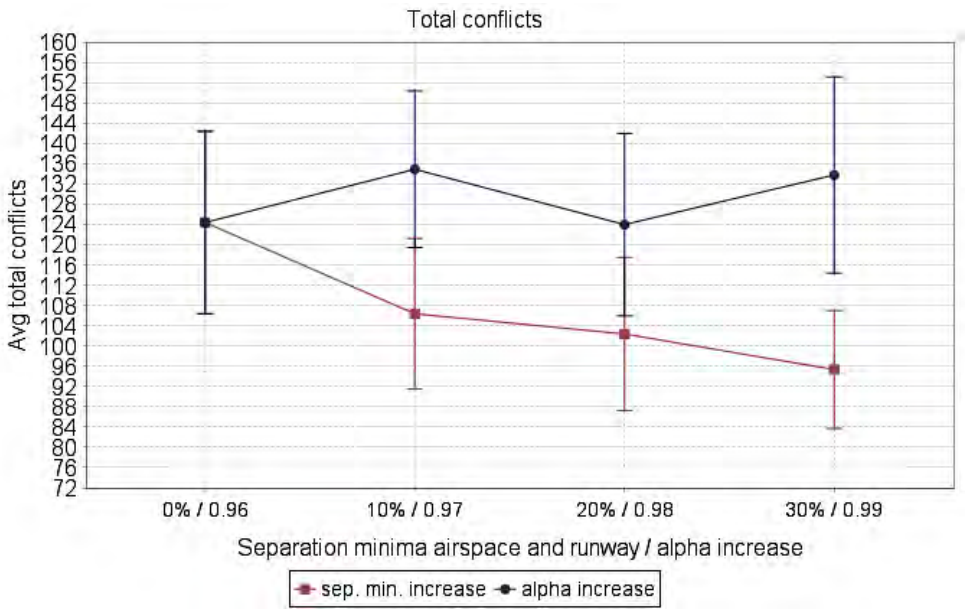


Figure 4.18: Total number of conflicts.

By analyzing the standard deviation of our average values (estimates) for all the airport components, we can find that in the default scenario, the standard deviation is higher than in scenarios 1 and 2. This can be clearly seen in the graphs related to the number of conflicts and to the conflict size interval, especially for airspace, *"runway in"* and *"runway out"* performance. In taxiway network and terminals performance, deviation is reduced to zero in almost all scenarios, and, as the average results shows, these two components are not affected significantly by the uncertainty. Concerning the total sum of conflicts, as depicted in Figure 4.18, we can see that the deviation from average values is higher in scenario 2 than in scenario 1; this is a recursive behavior which can be found among all the other airport components performance results. These findings provide insights in the analysis of the estimation of the conflicts related to each of the airport components. This means that by enlarging separation minima (scenario 1 implementation), we can obtain more accurate results, while by fine-tuning the parameter alpha of the metaheuristic, we obtain less accurate estimates.

4.2. Implementation of Opt-Sim method 2

The Opt-Sim method 2 described in chapter 3.3.2, implies to run an optimization model within a simulation environment, and then testing the optimized solution by running a simulation model. The two models are linked in order to form a feedback loop to continuously improve the optimized solution. Following the sliding window approach, the feedback loop aims at obtaining the best solution for each window until the entire time-horizon is covered. Different scenarios were implemented based on the optimization model input parameters to be fine-tuned. In this case, three different scenarios were evaluated and compared with a default scenario. Furthermore, the parameters of the feedback loop, M and N, must be set before running the scenarios; depending on the computer performance, the number of replications, as well as the parameters M and N can be changed accordingly. In the same way as in the Opt-sim method 1, the simulation model includes some source of uncertainty for evaluating the optimized solution under the close-to-real world scenario. The sources of uncertainty are described in section 4.1.1.1 and their values are shown in Table 4.6. For each window, the simulation model within the feedback loop is run 10 times to avoid being biased by the stochasticity of the variables.

4.2.1. Parameter setting for the Opt-Sim Method 2

Regarding the objective value tolerance parameter M, it was calculated as 10 percent of the average number of conflicts for each window for the scenario run without schedule optimization. This means that in every window, we want to keep the conflicts below this threshold, and in general we accept an amount of conflicts up to the 10 percent of the average number of conflicts for each time window. This criterion was arbitrarily chosen and could be further tested in future studies. Figure 4.19 presents the average objective function value for the schedule run without optimization for each time window. The average number of conflicts for each window, rounded to the nearest integer, was 168. Hence, the parameter M was chosen as 17 (16.8 rounded to the nearest integer).

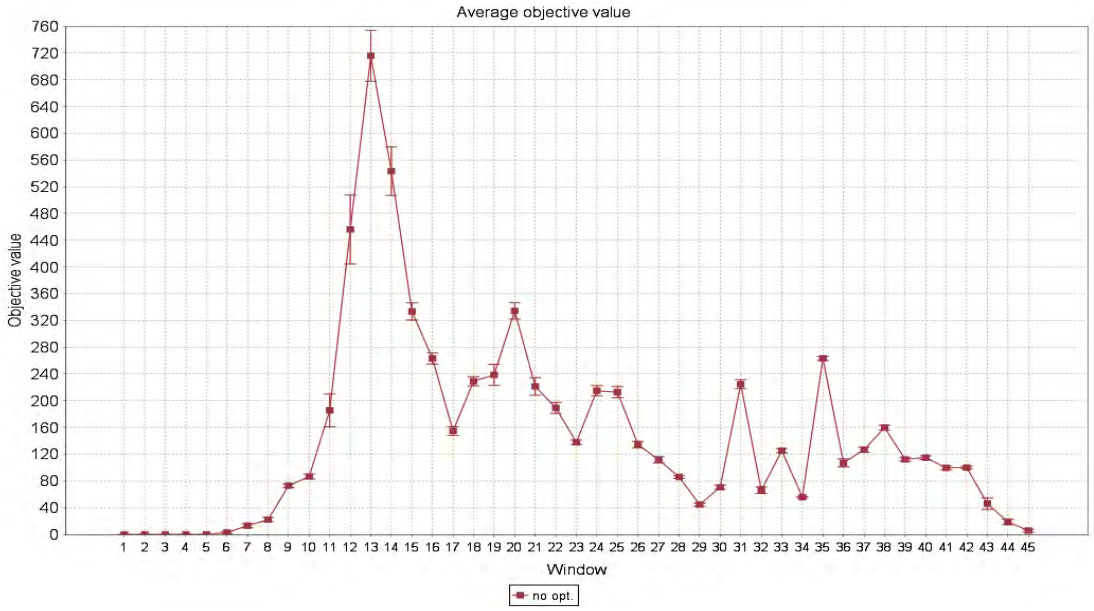


Figure 4.19: Number of conflicts for each time window for the scenario without schedule optimization

Regarding the maximum number of loops parameter N, the value that was chosen was three. The value of N impacts the computational aspect of the framework, and it can be adjusted based on the availability of hardware of the computer. However, a minimum of loops must be kept in order to evaluate the performance of the framework.

4.2.2. Scenarios

The proposed Opt-sim method 2 was tested by building three different scenarios where input parameters of the optimization process were fine-tuned at each loop, plus a default scenario. In the default scenario, no parameter is fine-tuned so the new solution generated is guided by the randomness of the metaheuristic. In the first scenario, the parameter of the metaheuristic is fine-tuned, and in the second one, the separation minima between aircraft in the airspace and at the runway, is fine-tuned. The third scenario is conducted by fine-tuning the objective function weights.

- Default scenario: at each loop, the optimization model generates a new solution based on the randomness of the metaheuristic when searching in the state space. We will refer to this scenario as: "Default"
- Scenario 1(S1): at each loop, we increase the simulated annealing cooling schedule parameter, α, so that the optimization model will generate a solution by exploring the state space more in depth. We will refer to this scenario as: "Alpha"
- Scenario 2(S2): at each loop, we increase the separation minima constraint (i.e. constraint relaxation). In theory we shall obtain an optimized solution which will be more conservative. However, when we simulate the solution, it can potentially absorb the effect of the uncertainty. We will refer to this scenario as: "Sep. min. increase"
- Scenario 3(S3): at each loop, we modify the objective function weights $\gamma_{airspace}$ and $\gamma_{airside}$. At each loop, depending on the value of the airspace or ground side performance in the objective function, we will increase its weight so that the next solution will focus more on that conflicts. Then, if most of the conflicts are generated by a specific area (airspace or ground side), then it is better to focus the optimization process on that specific area (by increasing the weight of that specific area). We will refer to this scenario as: "O.F. weight"

Table 4.8 shows the values of the parameters in various scenarios for each simulation loop in the Opt-sim method 2. Each row of the first column refers to the scenarios from S1 to S3, the other columns contain the values of the parameters used in each loop of these scenarios. The default scenario, is not displayed in this table since the fine-tuned parameters are kept with their default values among the various loops, thus, the cooling schedule parameter, α, is kept to 0.96, separation minima are kept with their original values, and the objective function weights, $\gamma_{airspace}$ and $\gamma_{airside}$, are kept both to 1. It is important to point out that the values for the objective function weights of Table 4.8 refer to the component of the objective function that impact the most the overall objective function value (it can be airspace or ground).

4.2.3. Results

In this section, results are presented and explained. The results are presented by analyzing first the different scenarios, and then by comparing them with each other. In order to show the potential and relevance of the method, we applied it to two different time frames. These two

time frames refer to the busiest hours during the day, when the airspace and airside capacity is stressed the most. The first time frame considered is from 5.30 AM until 10 AM, and it refers to windows 12 to 17; the second time frame considered is from 6 PM until 10.30 PM, and it refers to windows 37 to 42.

Table 4.8. Values of the fine-tuned parameters of the Opt-sim method 2

Scenario/Parameter	Loop 1	Loop 2	Loop 3
S1/Alpha	0.97	0.98	0.99
S2/Separation minima increase	+10%	+20%	+30%
S3/Objective function weight	+10%	+20%	+30%

4.2.3.1. Default scenario

In the default scenario, for each loop, the optimization process finds a new solution driven by the randomness of the metaheuristic applied (simulated annealing), and not by fine tuning input parameters. Figures 4.20 and 4.21 show the results for the two time frames considered in this analysis.

For each window, three loops were necessary, since in none of them the value of the objective function was lower than the threshold M (17). The Maximum number of loops N, was set to 3, therefore we implemented at maximum three loops for each window. Due to the stochasticity included in the system, specifically in the simulation model, we found high number of conflicts. The red line displayed in these figures, connects, for each window, the best results achieved which will be part of the overall final solution. As it can be noticed, in each window, the loops do not necessarily improve the solution of the first loop, this is due to the fact that in this scenario, the optimization is driven by the randomness of the metaheuristic, therefore, it is not guided by a specific criteria. Nevertheless, the method, is able to store for each window, the schedule that provides the best solution in terms of objective function value, before moving to the next window. The red line in Figures 4.20 and 4.21, represents the most robust solution obtained by the method, as for each loop the most robust solution was picked. In this context, robustness is defined as the capability of a solution to be resilient to perturbations. In the simulation model, perturbations are modeled as stochastic variables in order to represent the uncertainty of the operations. The robustness is then evaluated as the deviation of objective function value between the simulation and the optimization. Considering that the optimization provides for each window a free-conflict solution, the method aims at lowering the objective function value as much as possible.

In tables 4.9 and 4.10, the best values of the default scenario, for each window, are compared with the ones of the non-optimized scenario. The objective function is shown by its components: airspace; runways; taxiway; terminals. Regarding the airspace conflicts, it can be noticed that the default scenario is outperformed in almost all the windows by the non-optimized scenario, this results suggest that the optimized schedule is not effective in improving the airspace performance, and that the uncertainty plays a major role in the airspace congestion.

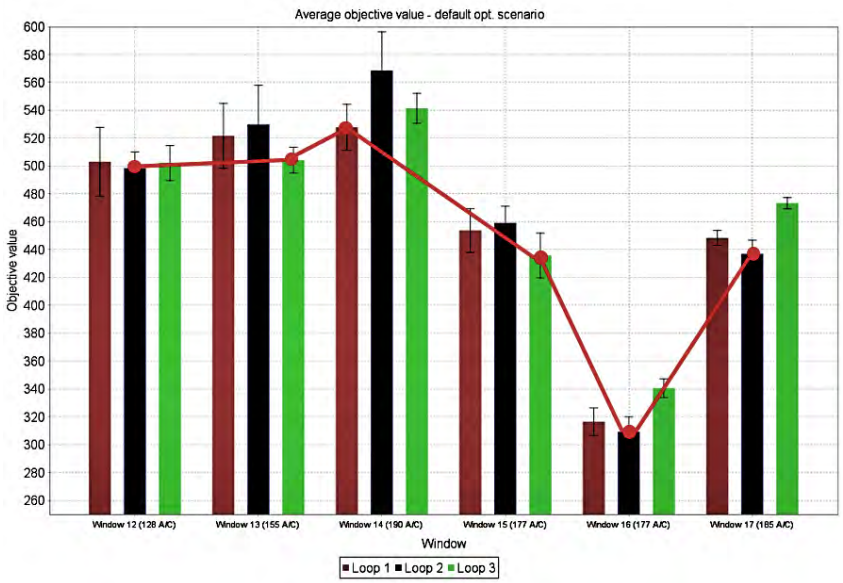


Figure 4.20: Results from the default scenario; time frame from 5.30AM to 10AM, windows 12 to 17.

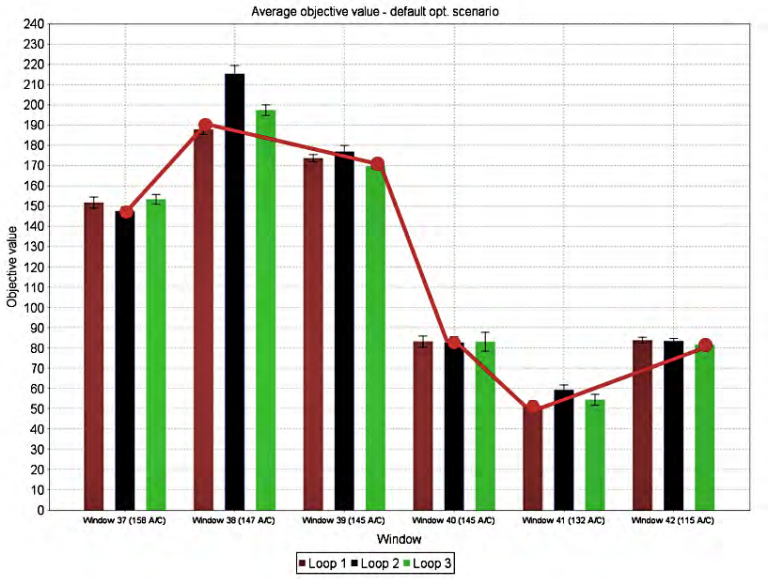


Figure 4.21: Results from the default scenario; time frame from 6PM to 10.30PM, windows 37 to 42.

On the other hand, for taxiway performance as shown in table 4.10, the default scenario outperforms the non-optimized one in almost every window, proving to be effective in lowering the congestion on the ground, moreover, taxiway operations proved to be not sensitive to the uncertainty. In these two tables, for each window, are shown the number of aircraft inside the window, this value is important to better understand the level of congestion in the airspace and on the ground. In the busiest time frames analyzed, the number of aircraft is always bigger than 100, reaching a maximum of 190 in window 14. Considering that in the default scenario, for both time frames, taxiway and terminal congestion are kept with values lower or slightly higher than their capacities, we can assume that most of the aircraft are in the airspace. Having the airspace with many aircraft, makes it congested, and exposes it to the risk of being affected by

the uncertainty to a big extent. This explains why airspace performance, in terms of conflicts, was sensitive to the stochastic perturbations triggered in the simulation model. In the non-optimized scenario, airspace performance are better than the default scenario, while airside performance are worse that the default scenario. This suggests that in the non-optimized solution, congestion was transferred to the airside, leading to have a smaller congestion in the airspace, therefore, uncertainty affected the airspace performance to a less extent.

Table 4.9. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17.

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 12 (128 /C)	No opt.	409.06	46.5	0	0
	Default	471	27.4	0	0
Window 13 (155 A/C)	No opt.	641.4	53.3	0	21.07
	Default	444	45.1	0	0
Window 14 (190 A/C)	No opt.	464.1	51.9	0	27.23
	Default	486.2	27.3	0	0
Window 15 (177 A/C)	No opt.	280.6	36.7	0	16.11
	Default	399.2	25.5	0	0
Window 16 (177 A/C)	No opt.	224.1	33.9	1	4
	Default	290.4	18.8	0	0
Window 17 (185 A/C)	No opt.	102.3	38.3	11.03	3.01
	Default	427.6	9	0	0

Table 4.10. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42.

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 37 (158 /C)	No opt.	67.2	43.4	16.11	0
	Default	134.9	8.5	4.01	0
Window 38 (147 A/C)	No opt.	97.4	38.1	24.16	0
	Default	171.8	12	4	0
Window 39 (145 A/C)	No opt.	61.1	37.1	14.1	0
	Default	156.6	8	22.04	0
Window 40 (145 A/C)	No opt.	49.4	45.1	20.23	0
	Default	58.1	13	20.13	0
Window 41 (132 A/C)	No opt.	30	38.1	31.42	0
	Default	17.2	13.9	18.12	0
Window 42 (115 A/C)	No opt.	34.4	33.8	31.83	0
	Default	43	19.2	5.01	0

4.2.3.2. Alpha scenario

In the alpha scenario, for each loop, the optimization process is driven by the fine-tuning of a parameter of the metaheuristic applied (simulated annealing), the cooling schedule, α. By tuning this parameter, the metaheuristic will find a new solution by searching more in depth into the state space. Figures 4.22 and 4.23 show the results for the two time frames considered in this analysis.

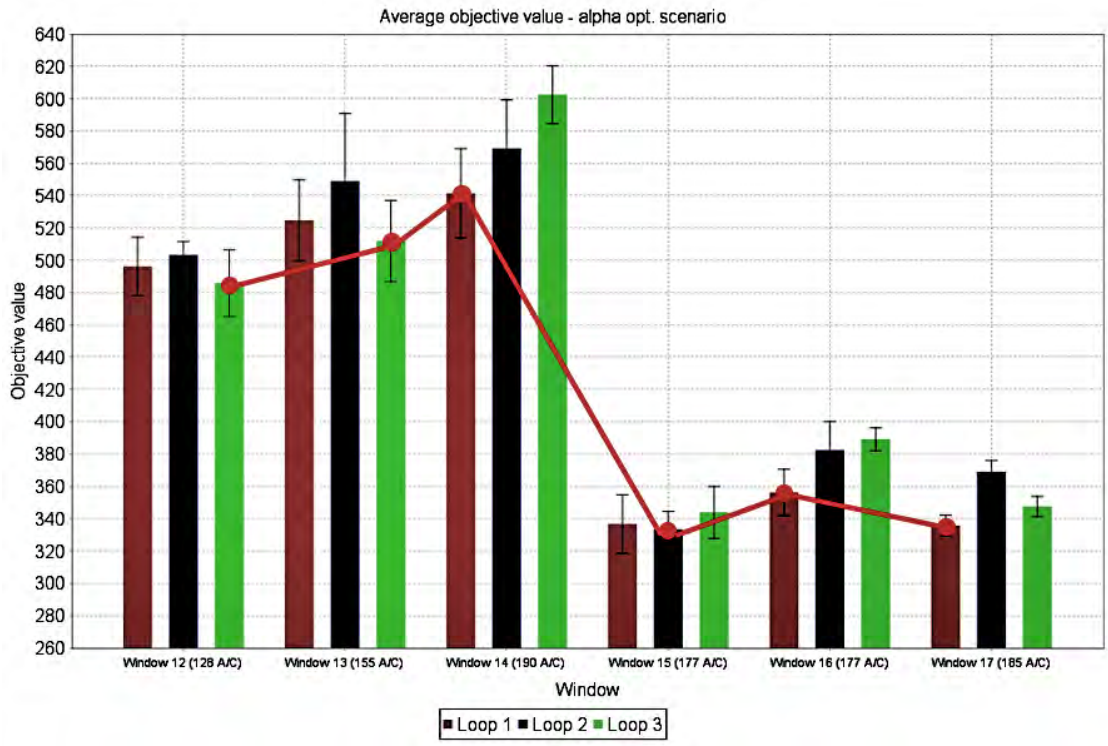


Figure 4.22: Results from the alpha scenario; time frame from 5.30AM to 10AM, windows 12 to 17.

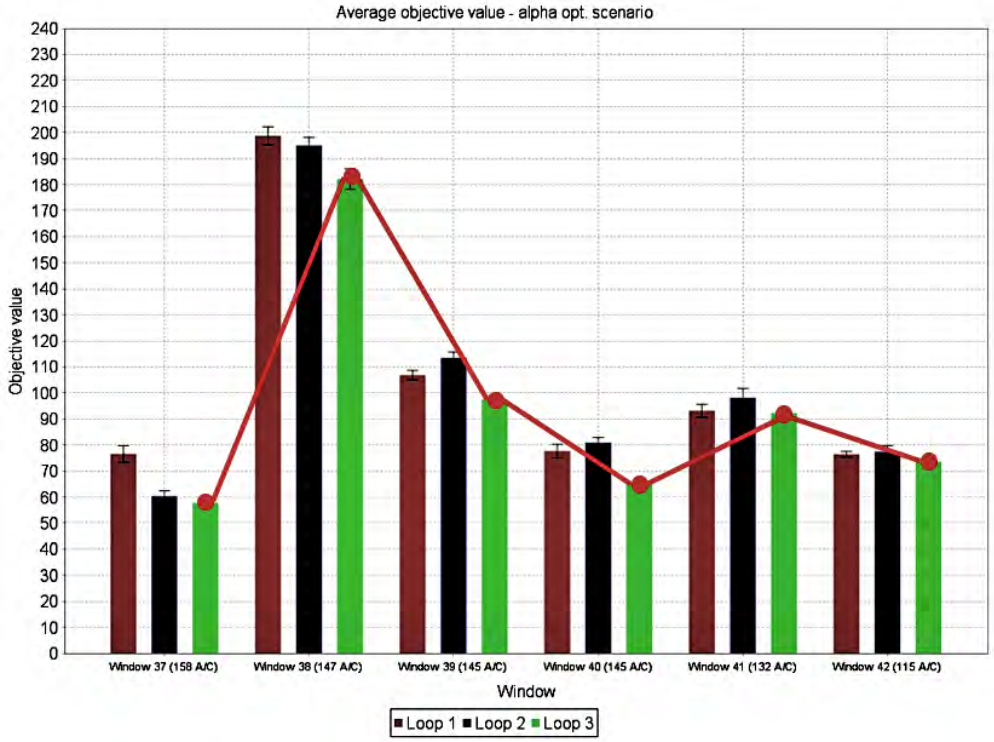


Figure 4.23: Results from the alpha scenario; time frame from 6PM to 10.30PM, windows 37 to 42.

For this scenario, the method was forced to implement three loops at each window, since in none of them the value of the objective function was lower than the threshold M (17). In this scenario, we found high values of objective function, revealing that the scenario found hard to

obtain a robust solution due to the uncertainty affecting the system. The red line displayed in these figures, shows how the best solution was built, by taking into consideration the best value among the three loops for each window. The alpha scenario improves the solution robustness, by lowering the objective function value, when more loop are implemented. Especially for the time-frame form window 37 to 42, as it shows in Figure 4.22, where all the best solution were found in the third loop. This results suggests that the cooling schedule parameter, β, impacts the efficiency of the method.

In tables 4.11 and 4.12, the best values of the alpha scenario, for each window, are compared with the ones from the non-optimized scenario. Similar to the default scenario, the airspace conflict are very high and even worse than the non-optimized scenario, except for windows 13 and 42. Regarding runway performance, the alpha scenario is able to lower the conflicts in all the windows of both time frames. Table 4.11, shows that terminal performance are improved by the alpha scenario, moreover, Table 4.12 shows that taxiway performance are improved as well. From these results we can derive similar conclusions as the default scenario, with the method being able to improve airside performance but not able to improve airspace conflicts for these busy windows. Once again, the stochasticity of the simulation model plays a critical role for the airspace performance, especially for windows with many aircraft involved in the airspace.

Table 4.11. Comparison between the alpha scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17.

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 12 (128 /C)	Non opt.	409.06	46.5	0	0
	Alpha	459.3	26.4	0	0
Window 13 (155 A/C)	Non opt.	641.4	53.3	0	21.07
	Alpha	453.8	42	0	13.69
Window 14 (190 A/C)	Non opt.	464.1	51.9	0	27.23
	Alpha	496.9	29.5	0	13.79
Window 15 (177 A/C)	Non opt.	280.6	36.7	0	16.11
	Alpha	295.7	21.9	0	15.1
Window 16 (177 A/C)	Non opt.	224.1	33.9	1	4
	Alpha	336.5	19.7	0	0
Window 17 (185 A/C)	Non opt.	102.3	38.3	11.03	3.01
	Alpha	295.7	16	0	0

4.2.3.3. Separation minima increase scenario

In the separation minima increase scenario, for each loop, the optimization process is driven by the fine-tuning of one of the constraints of the optimization problem, the separation minima between aircraft. At each loop, these values are incremented by 10%, up to 30% when the third loop is run. By fine-tuning this parameter, the optimization process will find a solution which will be more conservative because the aircraft will be spaced with each other by a bigger separation, at the same time, this might lead to find a more robust solution. Figures 4.24 and 4.25 show the results for the two time frames considered in this analysis.

Table 4.12. Comparison between the default scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 37 (158 /C)	Non opt.	67.2	43.4	16.11	0
	Alpha	39	11.7	7.02	0
Window 38 (147 A/C)	Non opt.	97.4	38.1	24.16	0
	Alpha	163	12.3	6.02	0
Window 39 (145 A/C)	Non opt.	61.1	37.1	14.1	0
	Alpha	87.8	4.8	4	0
Window 40 (145 A/C)	Non opt.	49.4	45.1	20.23	0
	Alpha	52.6	6.5	6.02	0
Window 41 (132 A/C)	Non opt.	30	38.1	31.42	0
	Alpha	70.2	7.6	13.08	0
Window 42 (115 A/C)	Non opt.	102.3	33.8	31.83	0
	Alpha	23	18.1	16.1	0

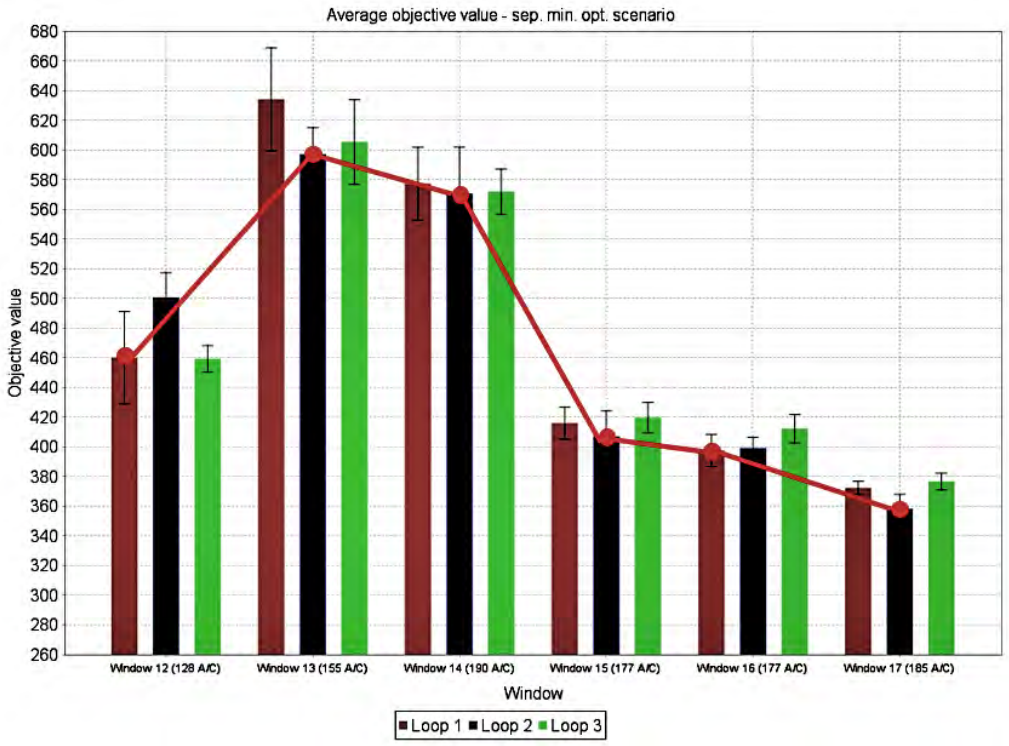


Figure 4.24: Results from the sep. min. increase scenario; time frame from 5.30AM to 10AM, windows 12 to 17.

For this scenario, as in the previous scenarios, three loops were necessary to be implemented for each window, due to the high value of objective function obtained. In this scenario, the value of the objective function was lowered for almost all the windows in the second loop as it can be seen from the red line that connects the best results, for each window, in Figure 4.24 and 4.25. In general, it can be noticed that the method is not too effective in finding a better solution, especially by looking at Figure 4.24, where it can be seen that there is not a big difference

between three loops. This results highlights that there is no a relevant difference by extending the separation minima from 10% to 30%.

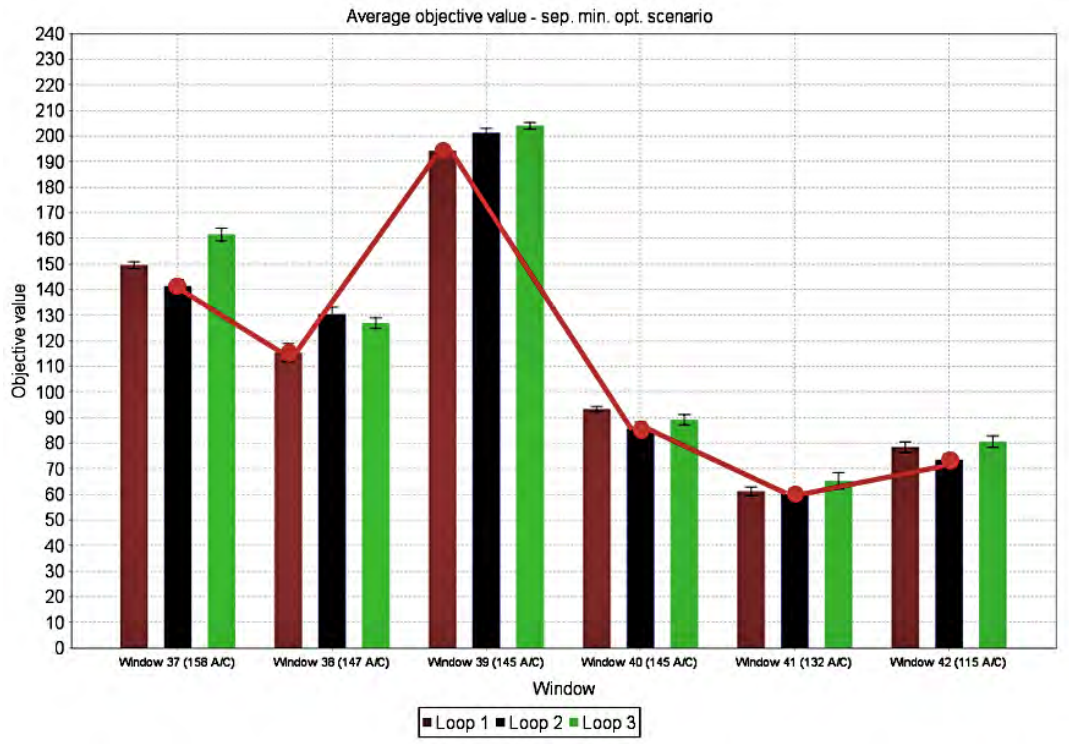


Figure 4.25: Results from the sep. min. increase scenario; time frame from 6PM to 10.30PM, windows 37 to 42.

In tables 4.13 and 4.14, the best values of the alpha scenario, for each window, are compared with the ones of the non-optimized scenario. Similar to the previous scenarios, the airspace conflict are very high and even worse than the non-optimized scenario, except for windows 13, 38 and 42. Regarding the airside components, runway, taxiway and terminals, the performance are similar to the previous scenarios, default and alpha. In both time frames, runway conflicts are lowered in all the windows, as well as taxiway and terminals. Regarding the efficiency and effectiveness of the method in improving the solution, we can draw the same conclusions as the two previously described scenarios, default and alpha. Airspace performance was hard to be improved as the opposite to the performance of the airside components, which proved to be less sensitive to the uncertainty.

4.2.3.4. Objective function weight scenario

In the O.F. weight scenario, for each loop, the optimization process is driven by the fine-tuning the objective function weights of the optimization problem. Specifically, the weights of the objective function, $\gamma_{airspace}$, and $\gamma_{airside}$, were fine-tuned at each loop. In this scenario, at each loop, depending on the value of the airspace and airside component, one of the two weight is increased by 10%, so that the optimization will focus more into one of them. For instance, if the performance of the airspace is bigger than the performance of the airside, then the weight related to the airspace will be increase of 10% so that the optimization will find a new solution by focusing more on solving the conflicts in the airspace. Figures 4.26 and 4.27 show the results for the two time frames considered in this analysis.

Table 4.13. Comparison between the sep. min. increase scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17.

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 12 (128 /C)	Non opt.	409.06	46.5	0	0
	Sep. Min. increase	434.2	23.3	0	0
Window 13 (155 A/C)	Non opt.	641.4	53.3	0	21.07
	Sep. Min. increase	544.3	40.4	0	9.64
Window 14 (190 A/C)	Non opt.	464.1	51.9	0	27.23
	Sep. Min. increase	526.5	23.9	0	12.27
Window 15 (177 A/C)	Non opt.	280.6	36.7	0	16.11
	Sep. Min. increase	374.6	17.6	0	13.08
Window 16 (177 A/C)	Non opt.	224.1	33.9	1	4
	Sep. Min. increase	378.7	16.7	0	0
Window 17 (185 A/C)	Non opt.	102.3	38.3	11.03	3.01
	Sep. Min. increase	345.7	10.6	0	0

Table 4.14. Comparison between the sep. min. increase scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 37 (158 /C)	Non opt.	67.2	43.4	16.11	0
	Sep. Min. increase	121.9	11.3	8.03	0
Window 38 (147 A/C)	Non opt.	97.4	38.1	24.16	0
	Sep. Min. increase	87.4	16.8	11.05	0
Window 39 (145 A/C)	Non opt.	61.1	37.1	14.1	0
	Sep. Min. increase	177.8	11	5.01	0
Window 40 (145 A/C)	Non opt.	49.4	45.1	20.23	0
	Sep. Min. increase	66.7	8.1	9.04	0
Window 41 (132 A/C)	Non opt.	30	38.1	31.42	0
	Sep. Min. increase	31.1	15.1	13.08	0
Window 42 (115 A/C)	Non opt.	102.3	33.8	31.83	0
	Sep. Min. increase	46.9	13.4	13.08	0

In Figure 4.26, given the time frame considered, it can be seen that the performance were improved in loop 1 in window 12, 14 and 15, in loop 3 in window 13, 16 and 17. The second loop, as it is displayed in Figure 4.26 was not capable to decrease the performance from the loop 1. Figure 4.27, shows the performance for the time frame from window 37 to 42. Here we can notice that the performance of loop 1 are improved by both loop 2 and 3 in almost all the windows, except for windows 37 and 41.

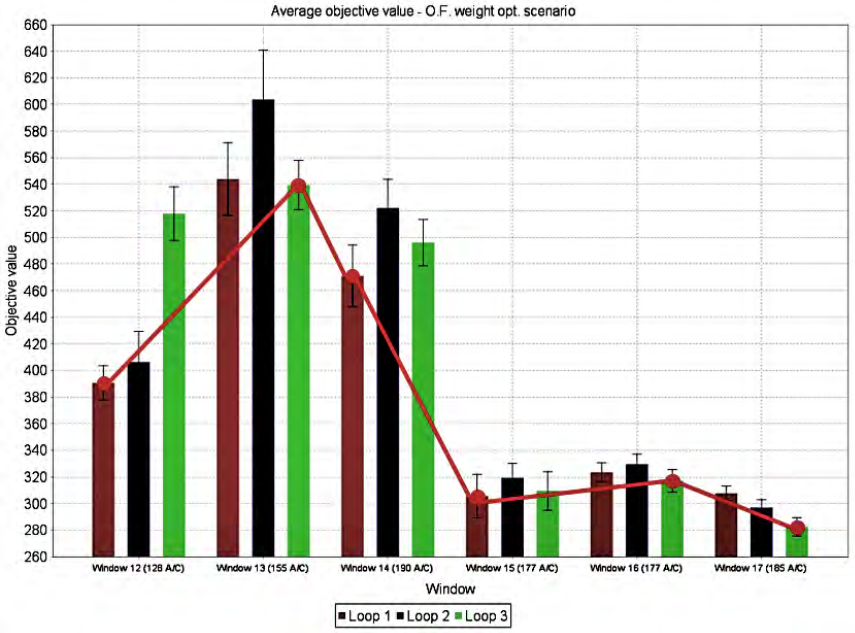


Figure 4.26: Results from the O.F. weight scenario; time frame from 5.30AM to 10AM, windows 12 to 17.

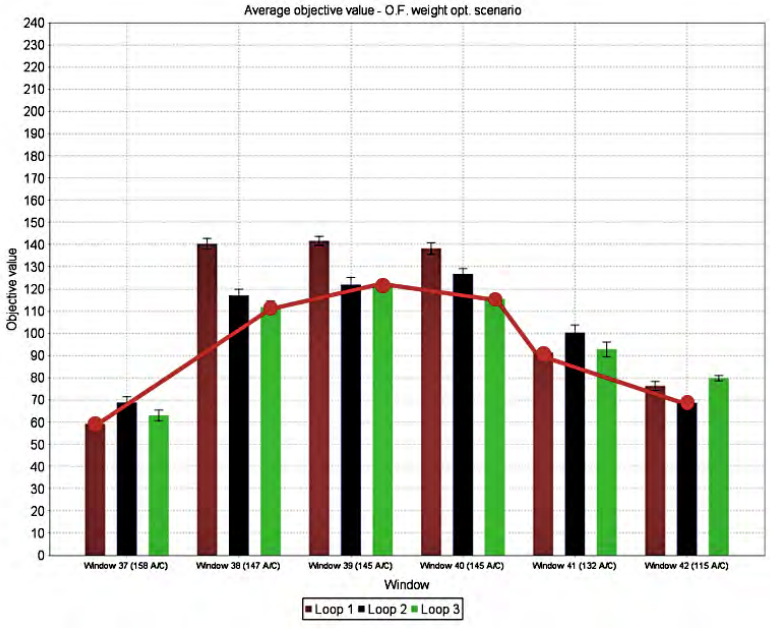


Figure 4.27: Results from the O.F. weight scenario; time frame from 6PM to 10.30PM, windows 37 to 42.

In tables 4.15 and 4.16, the best values of the O.F. scenario are compared with the ones of the non-optimized scenario. In this scenario, we can notice that the airspace conflicts are reduced compared to the non-optimized scenario in most of the windows, except for window 16, 17 as it is shown in Table 4.15. In the time frame from window 37 to 42, the airspace performance is improved in windows 37 and 38; it gets worse in the next three consecutive windows: 39, 40 and 41; finally the method improves the solution in the last window, 42. Looking at the two Tables 4.15 and 4.16, we can notice that airspace conflicts represent the biggest share of the total objective function value, as runway, taxiway and terminal performance are much lower than airspace performance. The O.F. weight scenario, in this context, will focus more on resolving airspace conflicts, by modifying decision variables such as: entry time in the TMA, entry speed in the TMA, and therefore, applying a delay to the aircraft. At first, these type of decisions have the effect of obtaining more robust solutions, as it can be seen in the first window of both time frames, but then they have the countereffect of moving the aircraft to the next windows, making them more congested and more sensitive to uncertainty, as the higher number of conflicts found in the following windows suggests. Regarding the airside components, Tables 4.15 and 4.16 show that the method is able to improve performance compared to the non-optimized scenario. Overall, for these two time frames, which represent the busiest hours of the day, the O.F. weight scenario seems to perform better than the other scenarios, being able to reduce airspace conflicts.

Table 4.15. Comparison between the O.F. weight scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 5.30AM to 10AM, windows 12 to 17.

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 12 (128 /C)	Non opt.	409.06	46.5	0	0
	O.F. weight	365.8	24.6	0	0
Window 13 (155 A/C)	Non opt.	641.4	53.3	0	21.07
	O.F. weight	484.8	42.5	0	4.21
Window 14 (190 A/C)	Non opt.	464.1	51.9	0	27.23
	O.F. weight	424.8	27	0	14.7
Window 15 (177 A/C)	Non opt.	280.6	36.7	0	16.11
	O.F. weight	271.5	22.4	0	11.58
Window 16 (177 A/C)	Non opt.	224.1	33.9	1	4
	O.F. weight	303	11.4	0	0
Window 17 (185 A/C)	Non opt.	102.3	38.3	11.03	3.01
	O.F. weight	272.4	10	0	0

4.2.3.5. Comparison between the scenarios

In this section, the different scenarios have been compared with the non-optimized one in order to analyze the efficiency of the method and to evaluate which scenario perform best. Two different time frames have been chosen to evaluate the scenarios, as they represent the busiest hours of operations at the airport during the day. Figure 4.28 and 4.29 show the best solution of the different scenarios along the windows for the two time frames. Looking at Figure 4.28, we can notice that the optimized scenarios not always outperform the non-optimized one, especially for window 16 and 17, where the non-optimized scenario perform the best. Among the optimized scenarios the one performing the best is the O.F. weight (S3) scenario, as it performs

the best in three windows, 12, 14, and 16. Regarding the second time frame, displayed in Figure 4.29, is not clear to establish which scenario perform best, since we find for the different windows different scenarios that outperforms the others. In window 37, 38 and 42, the best scenario is the O.F. weight (S3); in windows 39 and 40, the best scenario is alpha (S1); and in window 41 the best scenario is the default one.

Table 4.16. Comparison between the O.F. weight scenario and the non-optimized scenario. Objective function value broken down into its components. Time frame from 6PM to 10.30PM, windows 37 to 42

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 37 (158 /C)	Non opt.	67.2	43.4	16.11	0
	O.F. weight	43	7.8	8.02	0
Window 38 (147 A/C)	Non opt.	97.4	38.1	24.16	0
	O.F. weight	86.8	14.6	10.04	0
Window 39 (145 A/C)	Non opt.	61.1	37.1	14.1	0
	O.F. weight	102.7	13.7	5.02	0
Window 40 (145 A/C)	Non opt.	49.4	45.1	20.23	0
	O.F. weight	104.8	5.7	5.01	0
Window 41 (132 A/C)	Non opt.	30	38.1	31.42	0
	O.F. weight	63.6	14.4	12.06	0
Window 42 (115 A/C)	Non opt.	102.3	33.8	31.83	0
	O.F. weight	41.2	13.4	14.08	0

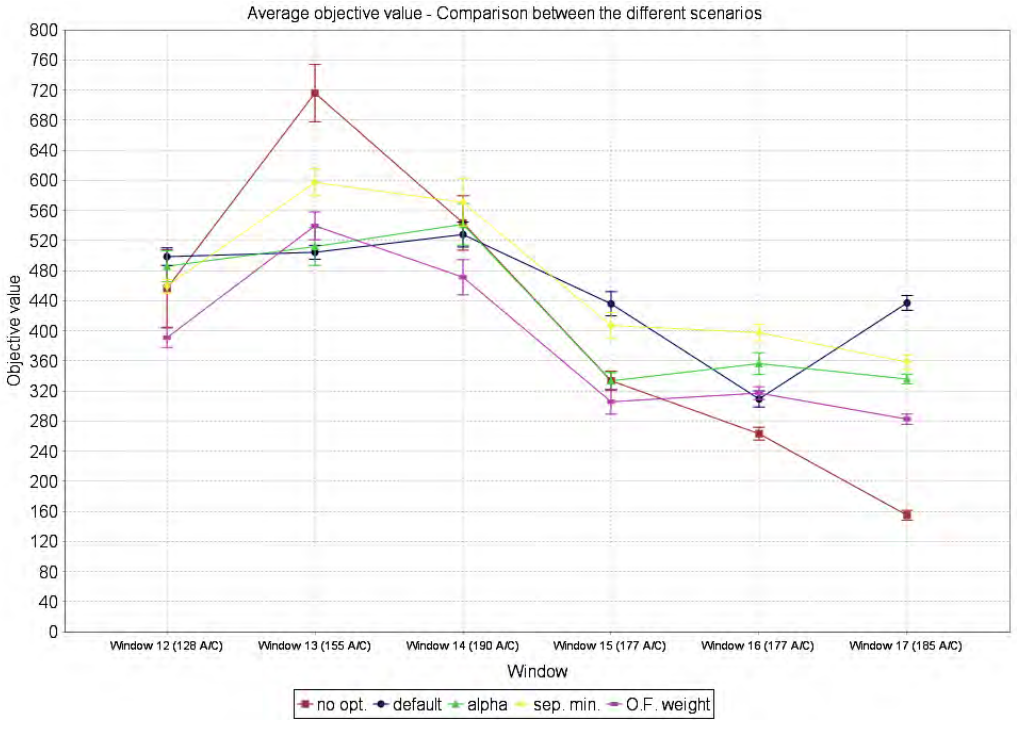


Figure 4.28: Comparison between the different scenarios and the non-optimized one. (time frame from window 12 to 17)

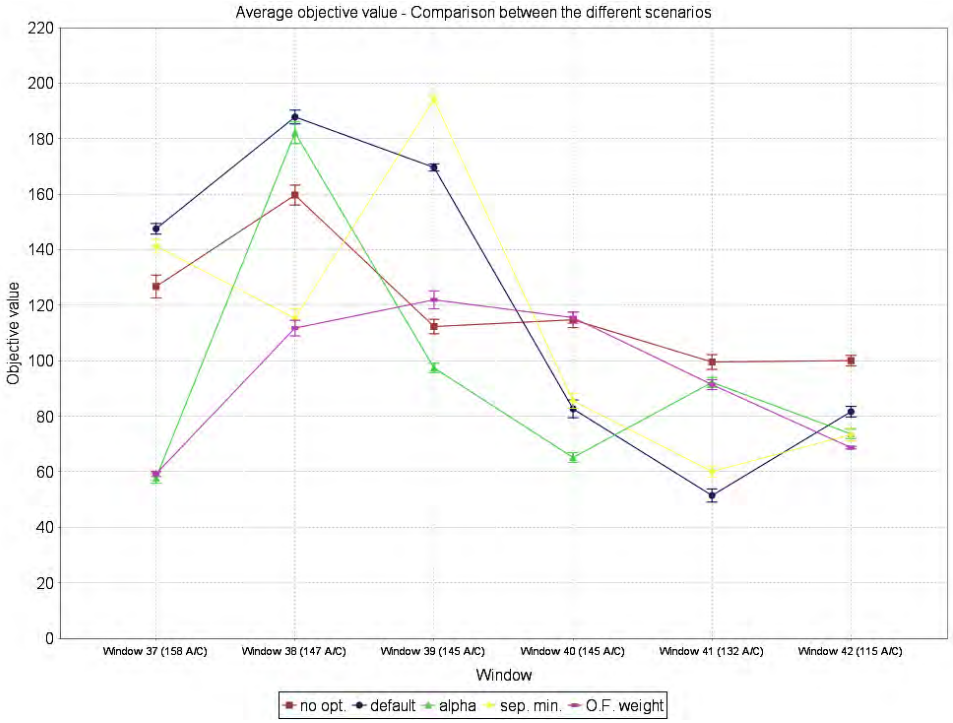


Figure 4.29: Comparison between the different scenarios and the non-optimized one. (time frame from window 37 to 42)

Table 4.17 and 4.18 summarize the results for the different scenarios, for the two different time frames. In these tables, for each window, the values of the objective function are shown by each of its component. Regarding the time frame from window 12 to 17, Table 5.17 show that the airside components, runway, taxiway and terminals, are always improved by all the different scenarios compared with the non-optimized one. Especially for the taxiway, we can see that there is no congestion even for the non-optimized scenario, this highlights the fact that the taxiway components is not sensitive to the uncertainty, proving to be robust. The area that needs more focus is, the airspace, since it contributes to a big extent to the overall objective function value. Among the different scenarios, the one that performs the best in terms of airspace performance, is the O.F. weight (S3) scenario, as it improves windows 12, 13, 14, and 15. In this scenario, the optimization process is clearly driven toward solving the airspace conflicts, since at each loops it will increase the airspace weight, $\gamma_{airspace}$, of the objective function. The scenario S3, however, is not able to improve all the windows, as it is the case for windows 16 and 17. The cause is attributed to the fact that, by focusing on airspace conflicts, the decision variables that are involved are related mainly to change in speed and entry time in the TMA, therefore, the optimization might tend to shift aircraft to the following windows resulting in accumulating them in specific windows and making the airspace too congested in such windows. A too congested airspace will be more sensitive to uncertainty, which explains the reason why at some windows this scenario are not able to effectively improve the airspace performance. In the second time frame considered, from window 37 to 42, the results show a similar trend as the first time frame previously analyzed. Airside components are not sensitive to uncertainty, as their performance are improved for each of the scenario implemented. Terminal, are not congested at all, showing no conflict in any of the scenarios, even for the non-optimized one.

Table 4.17. Objective function value broken down into the main components for the most congested windows. Comparison between the different scenarios. (windows 12-13-14-15-16-17).

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 12 (128 /C)	No opt.	409.06	46.5	0	0
	Default	471	27.4	0	0
	Scenario 1	459.3	26.4	0	0
	Scenario 2	434.2	23.3	0	0
	Scenario 3	365.8	24.6	0	0
Window 13 (155 A/C)	No opt.	641.4	53.3	0	21.07
	Default	444	45.1	0	13.69
	Scenario 1	453.8	42	0	13.69
	Scenario 2	544.3	40.4	0	9.64
	Scenario 3	484.8	42.5	0	4.21
Window 14 (190 A/C)	No opt.	464.1	51.9	0	27.23
	Default	486.2	27.3	0	12.68
	Scenario 1	496.9	29.5	0	13.79
	Scenario 2	526.5	23.9	0	12.27
	Scenario 3	424.8	27	0	14.7
Window 15 (177 A/C)	No opt.	280.6	36.7	0	16.11
	Default	399.2	25.5	0	11.05
	Scenario 1	295.7	21.9	0	15.1
	Scenario 2	374.6	17.6	0	13.08
	Scenario 3	271.5	22.4	0	11.58
Window 16 (177 A/C)	No opt.	224.1	33.9	1	4
	Default	290.4	18.8	0	0
	Scenario 1	336.5	19.7	0	0
	Scenario 2	378.7	16.7	0	0
	Scenario 3	303	11.4	0	0
Window 17 (185 A/C)	No opt.	102.3	38.3	11.03	3.01
	Default	427.6	9	0	0
	Scenario 1	319.7	16	0	0
	Scenario 2	345.7	10.6	0	0
	Scenario 3	272.4	10	0	0

having scenario S3, O.F. weight, performing the best in most of the windows, (37, 38 and 42), while not being effective in windows 39, 40 and 41. Particularly, as it can be seen in Figure 5.28, from window 37 to 39, the trend of airspace conflicts is uphill, while from window 39 to 42 it shows a downhill trend. In order to understand better this behavior a bigger spectrum of windows should be analyzed. Figure 4.30. shows airspace performance over an extended time frame for scenario S3. In this figure, we can see that window 36 has a higher number of airspace conflicts, afterwards the method, by focusing on airspace performance, is able to decrease airspace conflicts in window 37. For the same reason that was discussed before, aircraft are shifted toward the next windows, and that is why airspace performance starts to degrade until window 39. After window 39, the system starts to be less congested and airspace performance starts to improve as it can be noticed also in window 43.

Table 4.18. Objective function value broken down into the main components for the most congested windows. Comparison between the different scenarios,. (windows 37-38-39-40-41-42).

Window	Scenario	Airspace	Runway	Taxiway	Terminal
Window 37 (158 A/C)	No opt.	67.2	43.4	16.11	0
	Default	134.9	8.5	4.011	0
	Scenario 1	39	11.7	7.02	0
	Scenario 2	121.9	11.3	8.03	0
	Scenario 3	43	7.8	8.02	0
Window 38 (147 A/C)	No opt.	97.4	38.1	24.16	0
	Default	171.8	12	4	0
	Scenario 1	163	12.3	6.02	0
	Scenario 2	87.4	16.8	11.05	0
	Scenario 3	86.8	14.6	10.04	0
Window 39 (145 A/C)	No opt.	61.1	37.1	14.1	0
	Default	156.6	8	5.01	0
	Scenario 1	87.8	4.8	4	0
	Scenario 2	177.8	11	5.01	0
	Scenario 3	102.7	13.7	5.02	0
Window 40 (145 A/C)	No opt.	49.4	45.1	20.23	0
	Default	58.1	13	11.04	0
	Scenario 1	52.6	6.5	6.02	0
	Scenario 2	66.7	8.1	9.04	0
	Scenario 3	104.8	5.7	5.01	0
Window 41 (132 A/C)	No opt.	30	38.1	31.42	0
	Default	17.2	13.9	20.13	0
	Scenario 1	70.2	7.6	13.08	0
	Scenario 2	31.1	15.1	13.08	0
	Scenario 3	63.6	14.4	12.06	0
Window 42 (115 A/C)	No opt.	102.3	33.8	31.83	0
	Default	43	19.2	18.12	0
	Scenario 1	39.3	16.6	16.1	0
	Scenario 2	46.9	13.4	13.08	0
	Scenario 3	41.2	13.4	14.08	0

Table 4.19 shows the total sum of the average value of the objective function for all the windows. In the last row, the improvement in objective function for each scenario, compared to the non-optimized scenario, are displayed. We can see that, when considering all the windows of the entire day, which are 45 in total, all the optimized scenarios outperform the non-optimized one. The best results came from scenario S1 and S3, in which aircraft conflict reduction were by 11.1% and 11.2%, respectively. The scenario that performed the worst was S2, with 6.6% of conflict reduction over the non-optimized scenario. These scenarios were not able to reduce the conflicts to a big extent. This can be attributed to the fact that the optimization process was often moving aircraft from one window to another, leading to obtain high congested windows more sensitive to uncertainty. This suggests that this method might be more effective when used for a shorter time frame instead of a longer time frame, like an entire day of operations.

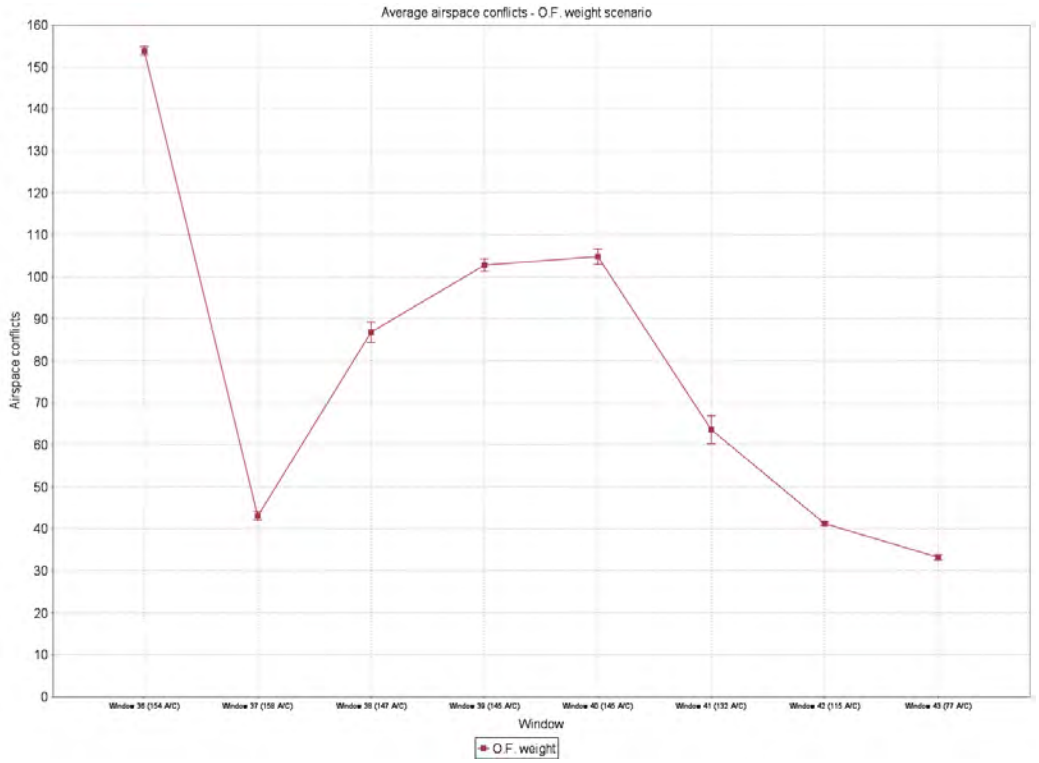


Figure 4.30: Airspace conflicts, O.F. weight scenario. Extended time frame (windows: 36-37-38-39-40-41-42-43)

Table 4.19. Comparison between the optimized scenarios and the non-optimized scenario considering the sum of the conflicts for all the windows.

	Scenarios				
	No Opt.	Default	S1	S2	S3
Total average value of objective function	6701	6208	5956	6256	5950
Improvement over the non-opt. scenario	-	7.3%	11.1%	6.6%	11.2%

The architecture of the method, can be more effective and efficient for smaller windows due to the possibility of extending some crucial parameter of the method such as N number of loops, which affect the effectiveness and efficiency of the method.

4.3. Conclusion

In this chapter, two different methods that combine optimization and simulation have been experimented. Two methods were applied to a real case study at Paris Charles de Gaulle airport. The first method, named as Opt-sim method 1, does not consider the sliding window approach for the simulation model, and utilizes the simulation model to validate the feasibility of the optimized solution from the optimization model. The second method, named as Opt-sim method 2, considers the sliding window approach also for the simulation model, and tries to continuously improve the optimized solution for each window by using the simulation model in order to provide a feedback to the optimization model.

In the Opt-sim method 1, two scenarios were built. These scenarios were based on the optimization model input parameters to be fine-tuned, which were: separation minima for airspace and runway increase, and simulated annealing cooling schedule, α. The performance of the different components of the airport such as airspace, runways taxiway network and terminals, were analyzed in detail. Results pointed out that some of the components were more sensitive to uncertainty, like the departing runway, while other components were less sensitive to it, like taxiway and terminals. Among the two scenarios, the one with separation minima increase is the best one, as it reduced the total conflicts by 23.33%, while the scenario with the cooling schedule parameter, α, was not effective as it was not able to obtain relevant improvements. However, considering the scenario with separation minima increase, the performance in terms of conflicts are considerably improved, confirming the benefits of the methodology.

In the Opt-sim method 2, three scenarios were implemented plus a default scenario. In the default scenario, none of the optimization model input parameters were fine-tuned. The other three scenarios were: the 'alpha' scenario, where the simulated annealing cooling schedule parameter, α, was fine-tuned; the 'separation minima' scenario, where airspace and runway separation minima were increased; and the 'objective function weights' scenario, where the weights of the objective function (airspace or airside) were fine-tuned. The results proved that the Opt-sim method 2 was able to improve the overall solution by reducing aircraft conflicts for the overall time horizon, however not to a big extent. Furthermore, during some of the busiest windows of the airport, the main findings still showed a high presence of conflicts. Although the methodology was not able to reach a conflict-free solution, the best scenarios were identified as the 'alpha' scenario and the 'objective function weight' scenario, capable of improving the average objective function value by 11.1% and 11.2%, respectively; while the default scenario and the 'separation minima' scenario improved the original solution by 7.3% and 6.6%, respectively, confirming the potential of the methodology. These results reveal that it would be better to drive the reduction in aircraft conflicts in the system by using parameters that are directly and indirectly involved in the optimization process. The 'alpha' scenario utilizes a fine-tuned parameter of the metaheuristic, while the 'objective function weights' scenario utilizes a fine-tuned parameter of the objective function which is, in turn, a part of the implemented metaheuristic. An aspect that is important to stress is that this method is not metaheuristic-driven, meaning that another metaheuristic can be applied to it, leaving room for further tests and implications on the efficiency and effectiveness of the method.

Chapter 5

Conclusion and future work

5.1. Conclusion

In this thesis, the problem of integrating airport operations under uncertainty was tackled. The aforementioned problem consisted in the optimization of the sequencing operations of landing aircraft in the terminal airspace and the mitigation of potential aircraft movement congestion on the airport ground. The first contribution of this thesis was the development of such an optimization model taking into account airspace and airport ground operations together in an integrated way. This optimization model was solved combining an adapted version of the simulated annealing metaheuristic and a time sliding window approach. Moreover, uncertainty of affecting real world operation was included in the model by means of a simulation model, and the effect of the uncertainty was quantified by comparing the value of the objective function with the solution provided by the optimization model. Lastly, two methods for combining optimization and simulation were developed for improving the robustness of the optimized solution.

- *Benefits from the integrated airport operations problem.* By modeling terminal airspace and airport ground operations together, it was possible to make decisions for aircraft still involved in the landing phase, but also looking ahead at the ground operations making sure to avoid aircraft movement congestion. This approach was possible by considering two different levels of abstraction for terminal airspace and airport ground operations. Terminal airspace operations were modeled in detail, while ground operations were modeled using a macroscopic approach. According to this macroscopic approach, the main components of airport ground operations were identified such as runways, taxiway network and terminals, and modeled considering their declared capacities and service times. In this way, it is possible to balance the congestion on the ground even when decision for aircraft are made quite some time in advance before landing.
- *Impact of uncertainty on the operations.* By simulating the optimized solution, it was possible to include some source of uncertainty such as deviation in the entry time in the airspace, deviation in the taxi time and deviation in the push back time. Results from the simulation model revealed that uncertainty was affecting the feasibility of the optimized solution and highlighted the need for a methodology that uses optimization and simulation in order to improve the feasibility and robustness of the optimized solution.
- *Benefits from the optimization and simulation framework.* By developing an approach combining optimization and simulation, it was possible to reach improvements in the solution feasibility and robustness. Although it was not possible to reach conflict-free solution, the approach proved its potential benefit by obtaining improvements in reducing the objective function value. The developed approach is general enough to be applied to other airports, however, it is more adapted to airports with large amount of traffic.

- *Comparison between the two opt-sim methods.* The two methods improved the optimized solution objective function value when uncertainty was considered, although without being able to reach a conflict-free scenario. The opt-sim method 1 was able to obtain relevant improvements in a relatively short time. On the other hand, this method fell short in treating the problem in a dynamic way, indeed it resulted to be static as it did not consider the sliding window approach while simulating the operations. The opt-sim method 2, was able to fully benefit from the sliding window approach also while simulating the operations, by improving the performance at each window, and by providing more accurate results. In the opt-sim method 2, simulation results provide, at each window, a feedback to the optimization model for the continuous improvement of the solution. This method was developed as a general framework that can potentially be applied to similar problems; moreover, it resulted to be non-metaheuristic driven since the framework allows to implement different optimization approaches. Although results were not satisfying, the method was able to reduce the conflicts up to a certain extent, proving its potential use. All these advantages were at the expense of the computational time, which represented the main hurdle for the Opt-sim method 2 compared to the Opt-sim method 1.

5.2. Future work

In view of future research, three main lines for improvements have been identified; the first one regards the integrated airport operations problem, the second one, the study of the sources of uncertainty, and the third one, the refinement of the optimization-simulation methods.

- *Extend the total airport capacity management at a microscopic level*. In this thesis, a different level of abstraction was chosen for airspace operations and airport ground operations. In the specific, airport ground operations were modelled broadly, considering only the capacity of the airport ground main components. In future work, the airport ground operations could be modelled more into details, at a microscopic level, by considering taxiway routing and gate assignment. A two-stage optimization model can be derived by connecting the macroscopic level and the microscopic level models.
- *Extensive study about the uncertainty sources.* By simulating the optimized solution, it was possible to include some source of uncertainty such as deviation in the entry time in the airspace, deviation in the taxi time and deviation in the push back time. Results from the simulation model revealed that uncertainty was affecting the feasibility of the optimized solution, and highlighted the need for a methodology that uses optimization and simulation in order to improve the feasibility and robustness of the optimized solution.
- *Further development for improving the opt-sim framework.* The opt-sim method 1 can be improved by choosing and testing different optimization input parameters to be fine-tuned. Moreover, an alternative optimization solution method can be tested, such as Genetic Algorithms. The opt-sim method 2, on the other hand, offers more potential for improvements than the opt-sim method 1, since there are more aspects that can be improved besides the once already mentioned. Further improvements can derive from the choice of the main parameters of the framework M and N, with the objective of reaching a balance between solution quality and computational effort. The results have

shown that some windows may need more loops to obtain an improvement in reducing aircraft conflicts. Therefore, future work could focus on the implementation of a dynamic rule for setting the parameter N at each window.

Bibliography

ACI EUROPE. (2019). Airport Traffic Report. https://www.aci-europe.org/media-room-airport-traffic-report/member.html Accessed: 29 January 2019.

Adacher, L., Flamini, M., Romano, E. (2017) Airport Ground movement Problem: Minimization of Dealty and Pollution Emission. IEEE Transaction on Intelligent Transportations Systems, Vol. 19, No. 12, pp. 3830-3839.

Adeleye, S., Chung, C. (2006). A simulation based approach for contingency planning for aircraft turnaround operation system activities in airline hubs. Journal of Air Transportation, Vol. 11, No.1, pp. 140-155.

Allignol, C., Barnier, N., Durand, N., Alliot, J-M. (2016). A new framework for solving en route conflicts. Air Traffic Control Quarterly, Vol. 21, No. 3, pp. 233-253.

Airbus. (2017). Airbus Global Market Forecast 2017.

Airbus. (2019). Aircraft Characteristics Airport operations & tech data. https://www.airbus.com/aircraft/support-services/airport-operations-and-technical-data/aircraft-characteristics.html accessed online the 29th of July 2019.

Airport Research Center. (2018). https://airport-consultants.com/ accessed online the 23rd of July 2018.

Ashford, N. J., Stanton, H. P. M., Moore, C. A., Coutu, P., Beasley, J. R. (2013). Airport Operations. Third edition. McGraw Hill. 2013.

Atkin, J.A., Burke, E.K., Ravizza, S. (2010). The airport ground movement problem: Past and current research and future directions. In the Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary, 2010, pp. 131–138.

Badrinath, S., Li, M.Z., Balakrishnan, H. (2019). Integrated Surface-Airspace Model of Airport Departures. Journal of Guidance, Control, and Dynamics, Vol. 42, No. 5, pp. 1049-1063.

Balakrishnan, H., Chandran, B.G. (2010). Algorithms for scheduling runway operations under constrained position shifting. Operations Research, Vol.58, No. 6, pp. 1650-1665

Banks, J., Carson, J.S., Nelson, B.L., Nicol D.M. (2000). Discrete-Event System Simulation. 3rd ed. Upper Saddle River, Nw Jersey: Prentice-Hall, *Inc.*

Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D. (2000). Scheduling aircraft landings: the static case. Transportation Science, Vol. 34, Issue 2, pp. 190-197.

Beasley, L.E., Sonander, J., Havelock, P. (2001). Scheduling aircraft landings at London Heathrow using a population heuristic. Journal of the Operational Research Society, Vol. 52, No. 5, pp. 483-493.

Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D. (2004). Displacement problem and dynamically scheduling aircraft landings. Journal of the Operational Research Society, Vol. 55, pp. 54-64.

Benlic, U., Brownlee, A.E.I., Burke, E.K. (2016). Heuristic search for the coupled runway sequencing and taxiway routing problem. Transportation Research Part C, Vol. 71, pp. 333-355.

Bennell, J.A., Mesgarpour, M., Potts, C.N. (2014). Airport runway scheduling. Ann. Oper. Res., (2013), Vol. 2014, pp. 249-270.

Bertsimas, D., Frankovich, M. (2015). Unified optimization of traffic flows through airports. Transportation Science, Vol. 50, No. 1, pp. 77-93.

Bianco, L., Dell'Olmo, P., Giordani, S. (2006). Scheduling models for air traffic control in terminal areas. J. Scheduling, (2006), Vol. 9, pp. 223-253.

Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., Grabbe, S. (2000). FACET: Future ATM Concepts Evaluation Tool. In the Proceedings of the 3rd USA/Europe Air Traffic Management R&D Seminar, 13-16 June 2000, Napoli, Italy.

Boeing. (2019). Airplane Characteristics for Airport Planning. https://www.boeing.com/commercial/airports/plan_manuals.page accessed online the 29th of July 2019.

Bohme, D. (2005). Tactical departure management with the Eurocontrol/DLR DMAN. In the Proceedings of the 6th USA/Europe Air Traffic Management Research and Development Seminar, Baltimore, MD.

Bolat, A. (2000). Procedures for providing robust gate assignments for arriving aircraft. European Journal of Operational Research, Vol. 120, No. 1, pp. 63-80.

Bosson, C.S., Xue, M., Zelinski, M. (2015). Optimizing integrated arrival, departure and surface operations under uncertainty. In the proceeding of the 10th USA/Europe ATM R&D Seminar, Lisbon, Portugal.

Brownlee, A.E.I., Weiszer, M., Chen, J., Ravizza, S. (2018). A fuzzy approach to addressing uncertainty in Airport Ground Movement optimization. Transportation Research Part C, Vol. 92, pp. 150-175.

Brunner, D.T., Cross, G., McGhee, C., Levis, J., Whitney, D. (1998). Toward Increase Use of Simulation in Transportation. In Proceedings of 30th Conference on Winter Simulation, edited by D. Medeiros, E. F. Watson, J. Carson, and M. Manivannan, 1169-1176. Washington, DC.

Carballo-Cruz, F., Costa, V. (2014). Success Factors of regional airports: The case of Oporto airport. Tourism and Management Studies Vol. 10, No. 1, pp. 37-45.

Chen, J.K.C., Yu, Y-W., Batnasan, J. (2014). Services Innovation impact to customer satisfaction and customer value enhancement in airport. In the Proceedings of the PICMET '14 Conference: Portland International Center for Management of Engineering and Technology, Infrastructure and Service Integration. 27-31 July 2014, Kanazawa, Japan.

Chua, Z.K., Cousy, M., Andre, F., Causse, M. (2014). Simulating air traffic control ground operations: preliminary results from project modern taxiing. In the Proceedings of the 4th SESAR Innovation Days 2014, November 2014, Madrid, Spain.

Chua, Z., Causse, M., Cousy, M., Andre, F. (2016). Modulating Workload for Air Traffic Controllers during Airport Ground Operations. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 59, N0. 1, pp. 16-20.

Confessore, G., Liotta, G., Grieco, R. (2005). Tactical decisions in the apron of Rome-Fiumicino airport. In the Proceeding of the 2005 Winter Simulation Conference, pp. 1596-1605, 4-7 December 2005, Orlando, FL, USA.

Cook, G., Goodwin, J. (2008). Airline networks: a comparison of hub-and-spoke and point-to-point systems. Journal of Aviation/Aerospace Education & Research. Vol. 17, No.2, pp. 51-60.

Courchelle, V., Soler, M., Gonzalez-Arribas, D., Delahaye, D. (2019). A simulated annealing approach to 3D strategic aircraft deconfliction based on en-route speed changes under wind and temperature uncertainties. Transportation Research Part C. Vol. 103 (2019), pp. 194-210.

Deau, R., Gotteland, J.B., Durand, N. (2009). Airport surface management and runways scheduling. In the Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar, June 2009, Napa, USA.

Delcaire, B., Feron, E. (1997). Development of an on-site ground operations model for Logan International Airport. FAA Air Transportation Center of Excellence in Operations research, research rep. RR-97-09. Dec. 1997, Cambridge, MA, USA.

Dell'Orco, M., Marinelli, M., Altieri, M.G. (2017). Solving the gate assignment problem through the Fuzzy Bee Colony Optimization. Transportation Research Part C, Vol. 80, pp. 424-438.

de Neufville, R., Odoni, A. R., Belobaba, P., Reynolds, T. (2013). Airport Systems: Planning, design, and management. Second edition. McGraw Hill Education. 2013.

de Oliveira, I.R., Galvao Texeira, R.J., Cugnasca, P.S. (2006). Balancing the air traffic control workload through airspace complexity function. IFAC Proceedings Volumes. Vol. 39, No. 20, pp. 64-69.

Dhief, I. (2018). Optimization of aircraft trajectories over the North Atlantic Airspace. Ph.D. dissertation, Université Paul Sabatier (Toulouse 3), Toulouse, France.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y. (2005). The over-constrained airport gate assignment problem. Computer & Operations Research, Vol. 32, pp. 1867-1880.

DLR. (2008). Topical Report Airline Business Model. https://ec.europa.eu/transport/sites/transport/files/modes/air/doc/abm_report_2008.pdf Accessed online the 29th of July 2019.

Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E. (2007). Flight gate scheduling: State-of-the-art and recent developments. Omega, Vol. 35, No. 3, pp. 326-334.

Dziedzic, M., Warnock-Smith, D. (2016). The role of secondary airports for today's low-cost carrier business models: the European case. Research in Transportation Business & Management. 21 (2016), pp. 19-32.

Eglese, R.W. (1990). Simulated annealing: a tool for operational research. European Journal of Operational Research, Vol. 46, pp. 271–281.

EU. (2011). Commission implementing regulation (EU) No 1035/2011. Official Journal of the European Union. 18/10/11, L 271, pp. 23-41.

EUROCONTROL. (2015). RECAT-EU, European Wake Turbulence Categorisation and Separation Minima on Approach and Departure. September 2015.

EUROCONTROL. (2017a). Industry monitor. Issue No. 190. 02/17.
EUROCONTROL. (2017b). Strategic, pre-tactical, tactical and post-ops air traffic flow and capacity management. http://www.eurocontrol.int/service/strategic-pre-tactical-tactical-and-post-ops-air-traffic-flow-and-capacity-management Accessed: 30 May 2018.

EUROCONTROL. (2018a). https://www.eurocontrol.int/sesar-research. Accessed: 31 May 2018.

EUROCONTROL. (2018b). Industry monitor. Issue No. 200. 14/02/2018.

Evertse, C., Visser, H.G. (2017). Real-time surface movement planning: Minimizing aircraft emissions. Transportation Research Part C, Vol. 79, pp. 224-241.

FAA. (2016). Airport categories. https://www.faa.gov/airports/planning_capacity/passenger_allcargo_stats/categories/. Accessed online the 29th of May 2018.

FAA. (2017). Instrument Procedures Handbook (IPH), Chapter 4: Approaches. https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/instrument_procedures_handbook/media/FAA-H-8083-16B_Chapter_4.pdf accessed online the 29th of July 2019.

FAA. (2018). https://www.faa.gov/nextgen/. Accessed online the 31st of May 2018.

Frankovich, M.J. (2012). Air traffic flow management at airports: a unified optimization approach. Ph.D. dissertation, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA.

Fraport. (2018). Frankfurt Airport Air Traffic Statistics 2018.

Fricke, H., Schultz, M. (2009). Delay impacts onto turnaround performance. 8th USA/Europe Air Traffic Management R&D Seminar, Napa, CA.

Furini, F., Kidd, M.P., Persiani, C.A., Toth, P. (2015). Improved rolling approaches to the aircraft sequencing problem. Journal of Scheduling, Vol. 18, Issue 5, pp. 435-447.

Genc, H.M., Erol, O.K., Eksin, I., Berber, M.F. (2012). A stochastic neighborhood search approach for airport gate assignment problem. Expert Systems with Applications, Vol, 39, pp. 316-327.

Goetz, A.R. (1992). Air passenger transportation and growth in the U.S. urban system, 1950-1987. Growth and Change, Spring 1992 pp. 217-238.

Gotteland, J-B., Deau, R., Durand, N. (2009). Airport surface management and runways scheduling. 8th USA/Europe Air Traffic Management R&D Seminar, Napa, CA.

Green, R.K. (2007). Airports and economic development. Real Estate Economics. Vol. 35, pp. 91-112.

Guepet, J., Briant, O., Gayon, J.P, Acuna-Agost, R. (2017). Integration of Aircraft Ground Movements and Runway Operations. Transportation Research Part E, Vol. 104, pp. 131-149.

Gupta, G., Malik, W., Jung, Y.C. (2009). A mixed integer linear program for airport departure scheduling. In the Proceedings of the 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), September 21-23 2009, Hilton Head, South Carolina, USA.

Hah, S., Willems, B., Philips, R. (2006). The effect of air traffic increase on controller workload. In the Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting, pp. 50-54.

Hao, S., Zhang, Y., Cheng, S., Liu, R., Xing, Z. (2018). Probabilistic multi-aircaft conflict detection approach for trajectory-based operation. Transportation Research Part C, Vol. 95, pp. 698-712.

Heidt, A., Helmke, H., Kapolke, M., Liers, F., Martin, A. (2016). Robust runway scheduling under uncertain conditions. Journal od Air Transport Management, Vol. 56, pp. 28-37.

Henderson, D., Jacobson, S.H., Johnson, A.W. (2003). The Theory and Practice of Simulated Annealing. Handbook of Metaheuristics, pp.2 87- 320.

Hoekstra, J., Ellerbroek, L. (2016). BlueSky ATC simulator project: an open data and open source approach. In the proceedings of the 7th International Conference on Research in Air Transportation, 20-24 June 2016, Philadelphia, PA, USA.

Hong, Y., Choi, B., Lee, K., Kim, Y. (2016). Conflict Management Considering a Smooth Transition of Aircraft Into Adjacent Airspace. IEEE Transactions on Intelligent Transportation Systems, Vol. 17, No. 9, pp. 2490-2501.

Hong, Y., Choi, B., Lee, K., Kim, Y. (2017). Dynamic Robust Sequencing and Scheduling Under Uncertainty for the Point Merge System in Terminal Airspace. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, No. 9, pp. 2933-2943.

Hu, X., Chen, W. (2005). Receding horizon control for aircraft arrival sequencing and scheduling. IEEE Transactions on Intelligent Transportation Systems, Vol. 6, no. 2, pp. 198-197.

IATA. (2017). Fact Sheet Aviation Charges, Fuel fees and Taxes. December 2017.

IATA. (2019). Airport Development Reference Manual. Ed. 11. March 2019.

ICAO. (2005). Annex 2 – Rules of the Air. 10 ed., July 2005.

ICAO. (2016). Doc 4444 - Procedures for Air Navigation Services (PANS) Air Traffic Management, 16th Edition, 2016.

ICAO. (2018). Annex 14 – Aerodromes – Volume 1 – Aerodromes Design and Operations. 8 Ed., July 2018.

ICAO. (2019). Doc 8643 – Aircraft Type Designators.

Idris, H. R., Anagnostakis, B., Delcaire, R. J., Hansman, J., Clarke, J. P., Feron, E., Odoni, A. R. (1999). Observations of departure process at Logan Airport to support the development of departure planning tools. Air Traffic Control Quarterly, Vol. 7, No. 4, pp. 229-257.

Innovation for Sustainable Aviation. (2018). https://www.isa-software.com/ramsplus/ accessed online the 23rd of July 2018.

Jeppesen. (2018). http://ww1.jeppesen.com/aviation/products/taam/total-airspace-airport-modeler.jsp accessed online the 23rd of July 2018.

Jones, D., Chartrand, R., Wilson, S., Commo, S., Otero, S., Barker, G. (2012). Airprot Traffic conflict detection and resolution algorithm evaluation. In the Procedings of the 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), 14-18 October 2012, Williamsburg, VA, USA.

Kang., L., Hanse, M. (2018). Assessing the impact of tactical airport surface operations on airline schedule block time setting. Transportation Research Part C, Vol. 89, pp. 133-147.

Kapolke, M., Furstenau, N., Heidt, A., Liers, F., Mittendorf, M., Weiss, C. (2016). Pre-tactical optimization of runway utilization under uncertainty. Journal of Air Transport Management, Vol. 56, pp. 48-56.

Kern, S., Schultz, M. (2016). Evaluation of a standardized single runway airport model with respect to runway capacity. In the Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference, pp. 1-13, 13-17 June 2016, Washington, DC, USA.

Khadaroo, J., Boopen S. (2008). The role of transport infrastructure in international tourism development: a gravity model approach. Tourism Management, 29 (2008) pp. 831-840.

Khadilkar, H., Balakrishnan, H. (2014). Network Congestion Control of Airport Surface Operations. Journal of Guidance, Control, and Dynamics, Vol. 37 No. 3, pp. 933-940.

Khoury, H.M., Kamat, V.R., Ioannou, P.G. (2007). Evaluation of general-purpose construction simulation and visualization tools for modeling and animating airside airport operations. Simulation, Vol. 83, No. 9, pp. 663-679.

Kim, S.H., Feron, E. (2011). Robust Gate Assignment. In AIAA Guidance, Navigation, and Control Conference, pp. 1161–1166. Portland, USA.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by simulated annealing. Science, Vol. 220, Issue 4598, pp. 671-680.

Kjenstad, D., Mannino, C., Schittekat, P., Smedsrud, M. (2013). Integrated surface and departure management at airports by optimization. 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO), 28-30 April 2013, Hammamet, Tunisia.

Klein, A. (2017). Using fast-time simulation to assess weather forecast accuracy requirements for air traffic flow management. In the Proceedings of the 2017 Winter Simulation Conference, pp. 2638-2649, Las Vegas, NV, USA.

Kleinman, N., Hill, S.D, Llends, V.A. (1998). Simulation optimization of air traffic delay cost. In the Proceedings of the 1998 Winter Simulation Conference, pp. 1177-1181. 13-16 December 1998, Washington, DC, USA.

Kuchar, J.K., Yang, L.C. (2000). A review of conflict detection and resolution modeling methods. IEEE Transactions on Intelligent Transportation System, Vol. 1, No. 4, pp. 179-189.

Lee, H. (2014). Airport surface traffic optimization and simulation in the presence of uncertainties. PhD thesis at the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, February 2014.

Lehouillier, T., NAsri, M.I., Desaulniers, G., Omer, J. (2017). Solving the Air Conflict resolution Problem Under Uncertainty Using an Iterative Biobjective Mixed Integer Programing Approach. Transportation Science, Vol. 51, No. 4.

Liang, M., Delahaye, D., Xu, X. (2015). A novel approach to automated merge 4D arrival trajectories for multi-parallel runways. In Proc. ENRI International workshop on ATM/CNS, Tokyo, Japan, 17-19 Nov. 2015.

Liang, M., Delahaye, D., Marechal, P. (2017). Integrated Sequencing and Merging Aircraft to Parallel Runways with Automated Conflict Resolution and Advanced Avionics. *Transportation research Part C* 85:268-291.

Lieder, A., Briskorn, D., Stolletz, R. (2015). A dynamic programming approach for the aircraft landing problem with aircraft classes. European Journal of Operational research, Vol. 243, No. 1, pp. 61-69.

Lijesen, M. G., Nijkamp, P., Pels, E., Rietveld, P. (2006). The Home carrier Advantage in Civil Aviation. In D. Lee (Ed.), Competition Policy and Antitrust (pp. 215-249). (Advances in Airline Economics; No. 1). Oxford: Elsevier.

Ma, J., Delahaye, D., Sbihi, M., Mongeau, M. (2016). Integrated optimization of terminal maneuvering area and airport. In Proc. SESAR Innovation Days, Delft, The Netherlands, 8-10 Nov. 2016.

Majumdar, A., Ochieng, W. (2002). Factors Affecting Air Traffic Controller Workload: Multivariate Analysis Based on Simulation Modeling of Controller Workload. Transportation Research Record, Vol. 1788 No. 1, pp. 58–69.

Makhloof, M. A. A., Waheed, M. E., Badawi, U. E. (2014). Real-time aircraft turnaround operations manager. Production Planning & Control, Vol. 25, No. 1, pp. 2-25.

Martinez, J.C., Trani, A.A., Ioannou, P.G. (2014). Modeling airside airport operations using general-purpose, activity-based, discrete-event simulation tools. Transportation Research Record: Journal of the Transportation Research Board. Vol. 1744, pp. 65-71.

Marzuoli, A., Gariel, M., Vela, A., Feron, E. (2014). Data-Based Modeling and Optimization of En Route Traffic. Journal of Guidance, Control, and Dynamics. Vol. 37, No. 6, pp. 1930-1945.

Michelin, A., Idan, M., Speyer, J. L. (2011). Merging of air traffic flows. Journal of Guidance, Control, and Dynamics, Vol. 34, No. 1, pp. 13-28.

MILT. (2010). CARATS Collaborative Actions for Renovation of Air Traffic Systems. http://www.mlit.go.jp/common/000128185.pdf accessed online the 29th of July 2019.

Mirkovic, B. (2011). Airport Apron Capacity Estimation – Model Enhancement. Procedia Social and Behavioral Sciences, Vol. 20, pp. 1108-1117.

Mogford, R., Guttman, J.A., Morrow, S.L., Kopardekar, P. (1995). The Complexity Construct in Air Traffic Control: a Review and Synthesis of the Literature. (No.DOT/FAA/CT-TN95/22). Atlantic City, NJ: Federal Aviation Administration, William Hughes Technical Center.

Montoya, J., Woord, Z., Rathinam, S., Malik, W. (2010). A Mixed Integer Linear Program for Solving a Multiple Route Taxi Scheduling Problem. In the Proceedings of the AIAA Guidance, Navigation, and Control Conference, pp. 1-15, August 2-5 2010, Toronto, Ontario, Canada.

Mujica, M. (2015). Check-in allocation improvements through the use of a simulation-optimization approach. Transport Research Part A, Vol. 77, pp. 320-335.

Mujica, M., Flores, I. (2017). Simulation and Optimization 2. 2nd ed, Mujica and Flores eds., Springer.

Mujica, M., Scala, P., Delahaye, D. (2017). Improving airport performance through a model-based and optimization approach. Simulation and Optimization 2, 2nd ed, Mujica and Flores eds., Springer, pp. 109-129.

Negahban, A., Smith, J.S. (2014). Simulation for Manufacturing System Design and Operation: Literature Review and Analysis. Journal of Manufacturing Systems, 33:241-261.

Norin, A., Granberg, T.A., Yuan, D., Varbrand, P. (2012). Airport logistics – A case study of the turn-around process. Journal of Air Transport Management, Vol, 20 (2012), pp. 31-34.

OAG. (2018). OAG punctuality league. OAG, www.oag.com. Accessed online on January 2018.

Odoni, A.R., Bowman, J., Delahaye, D., Deyst, J.J., Feron, E., Hansman, R.J., Khan, K., Kuchan, J.K., Pujet, N., Simpson, R.W. (1997). Existing and required modeling capabilities for evaluating ATM systems and concepts. International Center for Air Transportation Massachusetts Institute of Technology, Cambridge, MA, USA, March 1997.

Odoni, A.R. (1994). The importance of probability theory in the airport and air traffic control sectors. Int. J. Continuing Engineering Education, Vol. 4, pp. 105-113.

OptTek. (2018). http://www.opttek.com/products/optquest/ accessed online the 23rd of July 2018.

Potgieter, M., Saayman, M., Du Plessis, L. (2014). Key Success Factors in managing a visitors' experience at a South African international airport. Journal of Contemporary Management, Vol. 11, No. 1, pp. 510-533.

Productivity Apex. (2018). http://www.productivityapex.com/services/modeling-and-simulation.aspx accessed online the 23rd of July 2018.

Pujet, N., Declaire, B., Feron, E. (1999). Input-output modeling and control of the departure process of congested airports. In Conference proceedings of the Guidance, Navigation, and Control Conference and Exhibit, August 9-11 1999, Portland, Oregon, USA.

Ran, A., XiaoKang, Y. (2013). Application of simulation technology on airport system. Applied Mechanics and Materials, Vol. 392, pp. 936-940.

Rockwell Automation. (2018). https://www.arenasimulation.com/ accessed on line the 23rd of July 2018.

Roling, P., Visser, H.G. (2008). Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming. Int. J of Aerospace Engineering, Vol. 2008, pp.1-11.

Ruiz, S., Piera, M.A., Del Pozo, I. (2013). A Medium Term Conflict Detection and Resolution system for Terminal Maneuvering Area based on Spatial Data Structures and 4D Trajectories. Transportation Research Part C, Vol. 26, pp. 396-417.

Sama, M., D'Ariano, A., Pacciarelli, D. (2013). Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. Transportation Research Part E, Vol. 60, pp. 140-155.

Sama, M., D'Ariano, A., Corman, F., Pacciarelli, D. (2017). Coordination of scheduling decisions in the management of airport airspace and taxiway operations. Transportation Research Procedia, Vol. 23, pp. 246-262.

Scala, P., Mujica, M., de Bock, N. (2017). A divide and conquer approach for simulating an airport system. Int. Journal of Simulation and Process Modelling, Vol. 12, No. 6, pp. 470-484.

Schiphol Group. (2017). Traffic review 2017. http://trafficreview2017.schiphol.tangelo.nl/movements#per-type accessed online the 29th of July 2019.

Serhan, D., Lee, H., Yoon, S.W. (2018). Minimizing airline and passenger delay cost in airport surface and terminal airspace operations. Journal of Air Transport Management, Vol. 73, pp. 120-133.

Simaiakis, I., Balakrishnan, H. (2015). A Queuing Model of the Airport Departure Process. Transportation Science, Vol. 50 No. 1, pp. 94-109.

Simaiakis, I., Khadilkar, H., Balakrishnan, H., Reynolds, T.G., Hansman, R.J. (2014). Demonstration of reduced airport congestion through pushback rate control. Transportation Research Part A: Policy and Practice, Vol. 66, pp. 251-267.

Simio LLC. (2018). https://www.simio.com/index.php accessed online the 23rd of July 2018.

Smeltink, J.W., Soomer, M.J., de Waal, P., van der Mei, R. (2005). An Optimization Model for Airport Taxi Scheduling. In the Proceedings of the 13th Conference on the Mathematics of Operations Research, January 2005, Lunteren, The Netherlands.

Taurino, D., (2017). Automated Airport: problem definition. TaCo Take control – Automated solutions for the management of ground airport movements. http://www.tacoproject.eu/wp-content/uploads/2017/05/D2.1_automated-airport-prob-def_v01.00.00.issued.pdf accessed online on the 29th of July 2019.

Toratani, D., Ueno, S., Higuchi, T. (2015). Simultaneous optimization method for trajectory and sequence for receding horizon guidance in terminal area. SICE Journal of Control, Measurement, and System Integration, Vol.8, no. 2, pp. 144-153.

Transoft Solutions. (2018). http://airtopsoft.com/ accessed online the 23rd of July 2018.

Van de Vijver, E., Derudder, B., Wiltox, F. (2015). Air passenger transport and regional development: cause and effect in Europe. Transport and Sustainable Development. Vol. 28, No. 2, pp. 143-154.

Van Schaijk, O.R.P., Visser, H.G. (2017). Robust flight-to-gate assignment using flight presence probabilities. Transportation Planning and Teechnology, Vol. 40, No. 8, pp. 928-945.

Wei, G., Siyuan, J. (2010). Simulation study on closed spaced parallel runway analysis using SIMMOD plus. In the Proceedings of the International Conference on Intelligent Computation Technology and Automation, pp. 344-347. 11-12 May 2010, Changsha, China.

Wu, C.L., Caves, R.E. (2004). Modelling and simulation of aircraft turnaround operations at airports. Transportation Planning and Technology, Vol. 27, No. 1, pp. 25-46.

Zhan, Z., Zhang, J., Li, Y., Liu, O., Kwok, S.K., Ip, W.H., Kaynak, O. (2010). An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem. IEEE Transactions on Intelligent Transportation Systems, Vol. 11, no. 2, pp. 299-312.

Zhang, H.H., Xue, Q.W., Jiang, Y. (2017). Multi-objective gate assignment based on robustness in hub airports. Advances in Mechanical Engineering, Vol. 9, No.3, pp.1-13.

Zheng, Y., Sui, D., Lin, Y., Zhang, J. (2015). Multi-agent based air traffic simulation system. In the Proceedings of the AIAA Modeling and Simulation Technologies Conference, 22-26 June 2015, Dallas, TX, USA.

Zou, X., Cheng, P., Liu, W.D., Cheng, N., Zhang, J.P. (2018). A two-stage taxi scheduling strategy at airports with multiple independent runways. Transportation Research Part C, Vol. 95, pp. 165-184.

Zuniga, C., Delahaye, D., Piera, M.A. (2011). Integrating and Sequencing Flows in Terminal Maneuvering Area by Evolutionary Algorithms. In the proceedings of the 30th Digital Avionics Systems Conference, October 16-20, 2011, Seattle, WA, USA.

Zuniga, C., Piera, M.A., Ruiz, S., Del Pozo, I. (2013). A CD&CR causal model based on path shortening/path stretching techniques. Transportation Research Part C, Vol 33 (2013), pp. 238-256.

Appendix A: Simulation model of the Opt-sim method 1 (SIMIO)

Model structure. In the Opt-sim method 1, the simulation model of the airport TMA airspace and ground was developed by using the general purpose simulation software SIMIO. In Figure A.1 a schema of the simulation model is given, by including the main objects used for building the model.

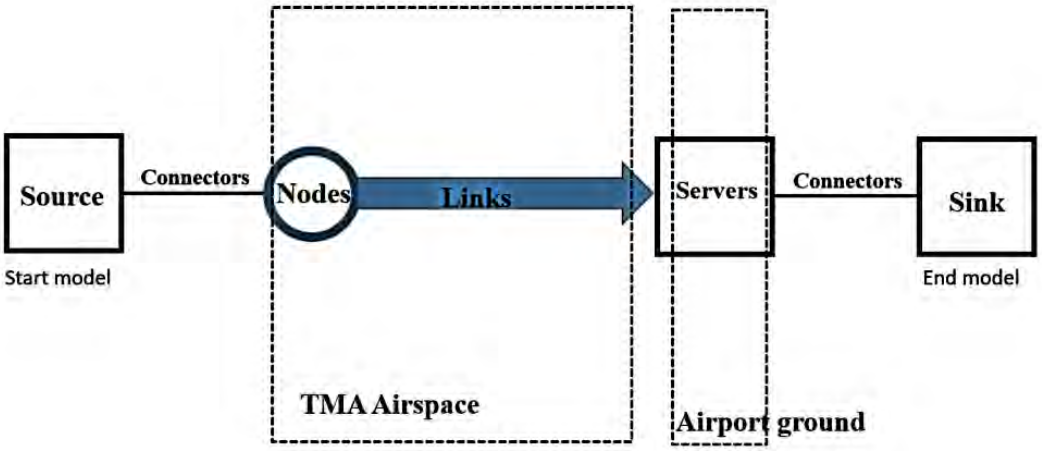


Figure A.1: Schema of the simulation model of the Opt-sim Method 1

The airport TMA airspace was modelled as a network of node objects and path objects, representing the waypoints and segments of the landing routes. In each node, logic such as speed-check and separation minima-check were implemented. Segments were modelled by using paths, where properties such as length, maximum capacity, and maximum speed could be set. Figure A.2 shows the animation of the TMA airspace modelled. In the following, each model object is described with its main characteristics and functionalities.

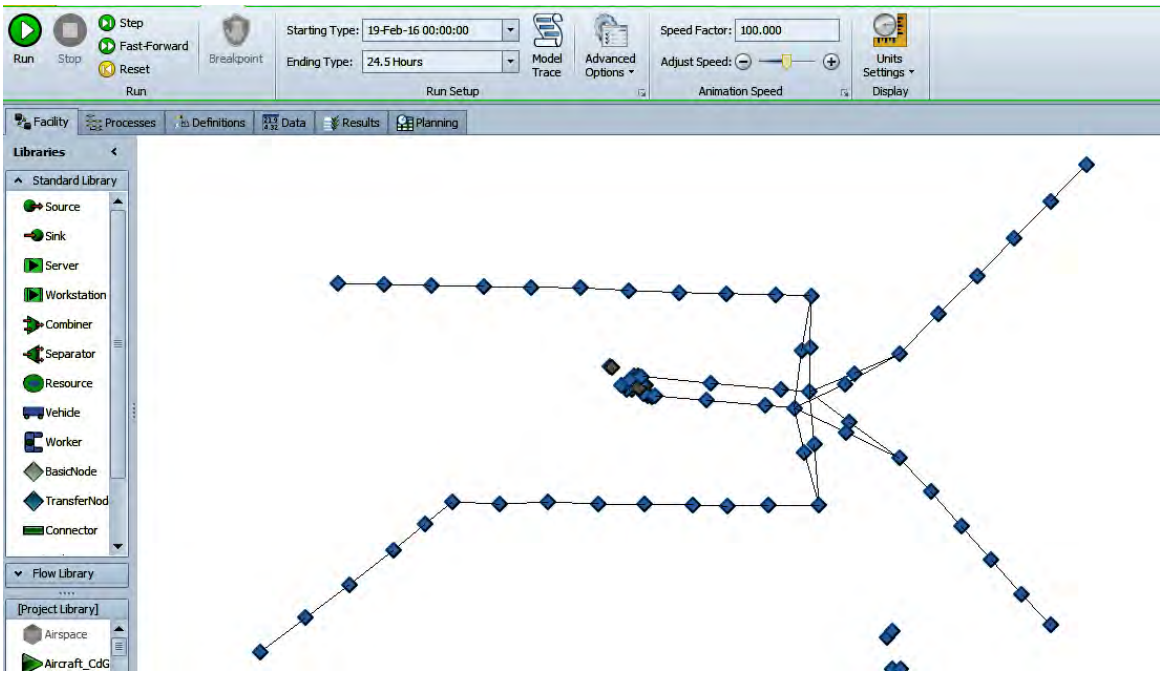


Figure A.2: Airspace model structure (SIMIO)

- Model paths. Paths represent the segment of the airspace landing routes. Figure A.3 shows the property view of the path object. The main properties of this object are: the length of the segment; the path *type*, which can be unidirectional or bidirectional; and the path *speed limit*, which determines the maximum speed that can be reached on the path.

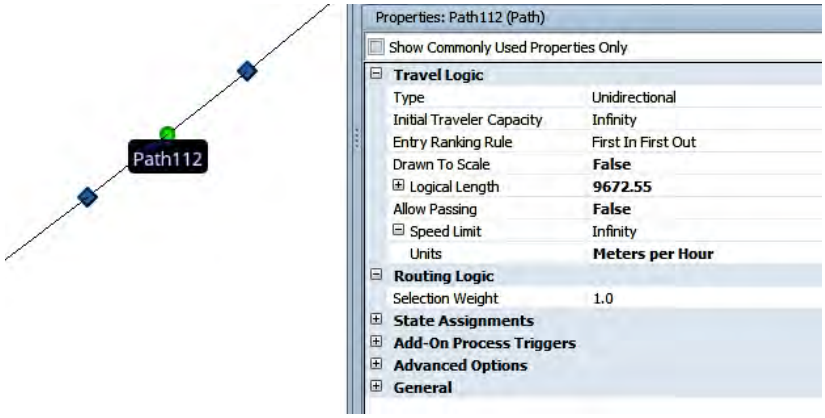


Figure A.3: Path object properties (SIMIO)

- Model nodes. Nodes represent the waypoints of the airspace landing routes. In Figure A.4, there is the property view for the object. In the section "add-on Process Triggers" there are two processes used for implementing logic such as separation minima between aircraft and speed updates.

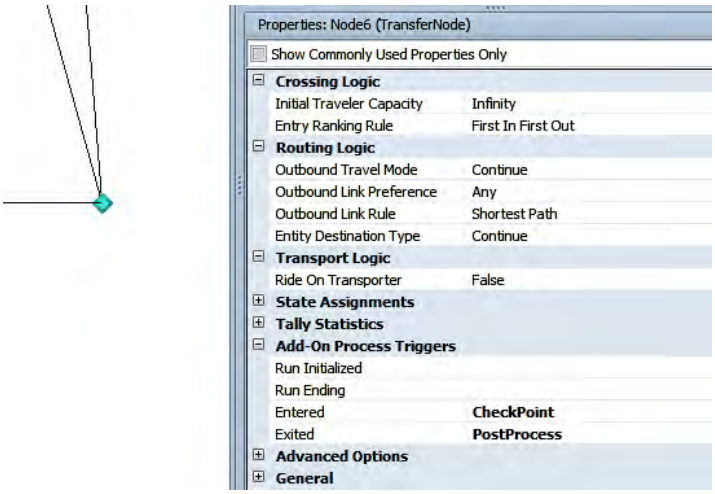


Figure A.4: Node object properties (SIMIO)

Regarding the airport ground components, runways, taxiway network and terminals were modelled by using server objects. Server objects were linked to each other by using connectors (a specific time-zero object). In Figure A.5 it can be seen how these objects were utilized in the model. Flights were created by the object source, and they were destroyed by the object sink at the end of the model. In the following, each model object is described with its main characteristics and functionalities.



Figure A.5: Airport ground side model structure (SIMIO)

- Models servers. Server objects were used for modeling ground components such as runways, taxiway network and terminals. Figure A.6 shows the property view of the server object. The main properties of this object are: the capacity, which sets the maximum number of entities that can be processes by this object simultaneously; and the processing time, which sets the amount of time that each entity will be processed by this object.

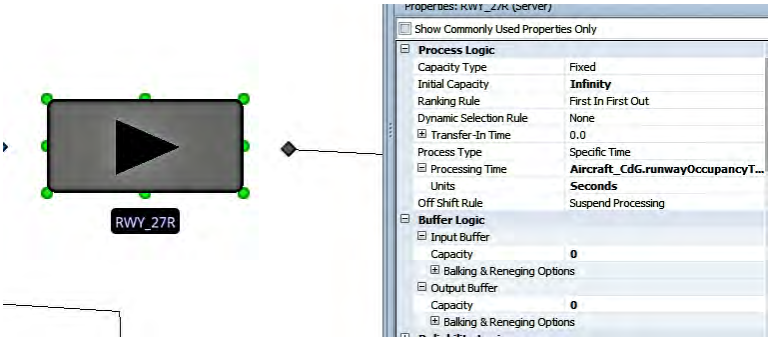


Figure A.6: Server object properties (SIMIO)

- Model source. The source object is used for creating the entities of the model. The source generates entities according to a flight schedule. Figure A.7 shows the property window of this object. The main properties of this object are: the entity type, which define the entity to be created; and the arrival mode, which is the way the entities are created, following an inter-arrival time, a specific schedule and so on.

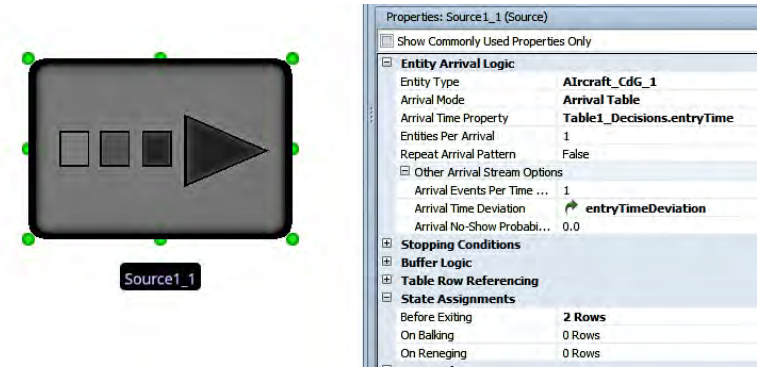


Figure A.7: Source object properties (SIMIO)

- Model sink. The sink object is placed at the end of the model and it is used for destroying the entities. Figure A.8, shows the property view of the sink object. Usually, in the sink object, there are not many properties to be set, however, in this object, it is possible to insert some logic for modeling specific behaviors.

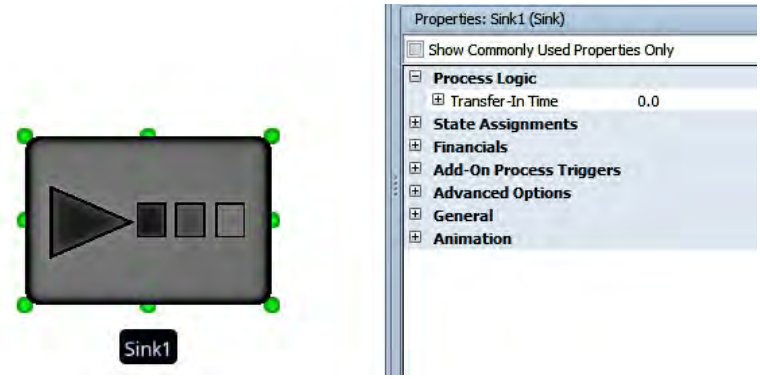


Figure A.8: Sink object properties (SIMIO)

Model Logic. Implementing the logic of the model in SIMIO is possible with the use of the Processes. A process is a logical sequence of steps (see Figure A.9). Once a process is created it can be called by different objects within the model. Each step is a feature that allows to do some specific actions. The main steps used in the model are: Assign, Decide, Delay, Wait, Fire, Execute and SetNode:

- Assign: assigns a value to a variable
- Decide: based on a condition or a probability, the entity can exit a true or a false exit, in this way if-then-else statement can be implemented.
- Delay: delays the occurrence of an event.
- Wait: allows an entity to wait until a specific event is triggered ("Fired").
- Fire: triggers an "Event".
- Execute: executes a "Process" within another "Process".
- SetNode: sets the node destination to an entity.

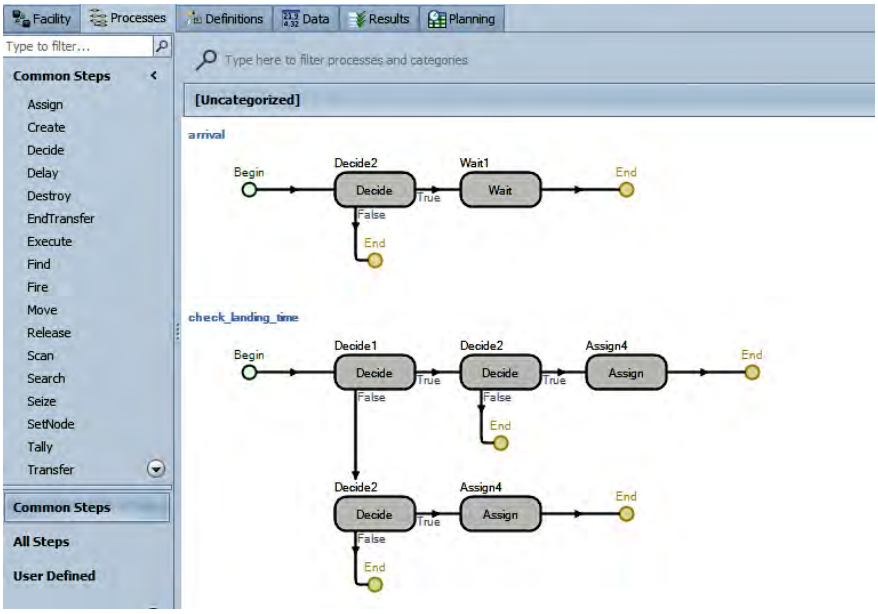


Figure A.9: Processes in SIMIO

In Table A.1, the main processes implemented in the simulation model are listed, providing with their descriptions and also with their locations within the model.

Table A.1. List of processes implemented in the model

Process name	Process logic	Location in the model
Output_Source1_Entered	Assigns airspace route to each aircraft. Assigns data input to each aircraft. Calculates the acceleration for each aircraft.	Source object (entered)
Separation_Minima_Check	Defines separation between aircraft	Node object (entered)
Process1	Detects airspace conflicts	Separation_Minima_Check (process)
Airspace_conflict_percentage	Calculates the average airspace conflict size as a percentage of the total separation minima	Process1 (process)
Airspace_conflict_size	Calculates the number of airspace conflicts falling into one of the conflict size interval	Process1 (process)
Update_Leading_Aircraft	Update leading aircraft data (speed, time, aircraft type)	Node object (exited)
Calculate_Landing_Separation_Minima	Calculates the separation minima to be respected given two specific consecutive aircraft	Server object input node (entered) (Landing runway)
Rwy_26L/27R_Landing	Detects landing runway conflicts	Server object (processing) (Landing runway)
Input_Taxiway1_entered	Calculates the taxiway utilization. Detects taxiway overload. Assigns taxi time.	Server object input node (Entered) (Taxiway)
Output_taxiway1_entered	Updates taxiway utilization.	Server object (after processing) (taxiway)
Terminal_utilization	Calculates the terminals utilization.	Server object (processing) (terminals)
Terminal_conflict_detection	Detects terminals overload.	Terminal_utilization (process)
Rwy_26R/27L_Departure	Calculates the separation minima to be respected give two specific consecutive aircraft. Detects departing runway conflicts.	Server object (Processing) (Departing runway)
Acceleration2	Updates aircraft speed.	Called every time an aircraft entity move from one node to another node.

Appendix B: Simulation model of the Opt-sim method 2 (Anylogic)

Model structure. In the Opt-sim method 2, the simulation model of the airport TMA airspace and ground was developed by using the general purpose simulation software Anylogic. In Figure B.1 a schema of the simulation model is given, while Figures B.2 shows the structure of the model with all the different objects utilized for building it.

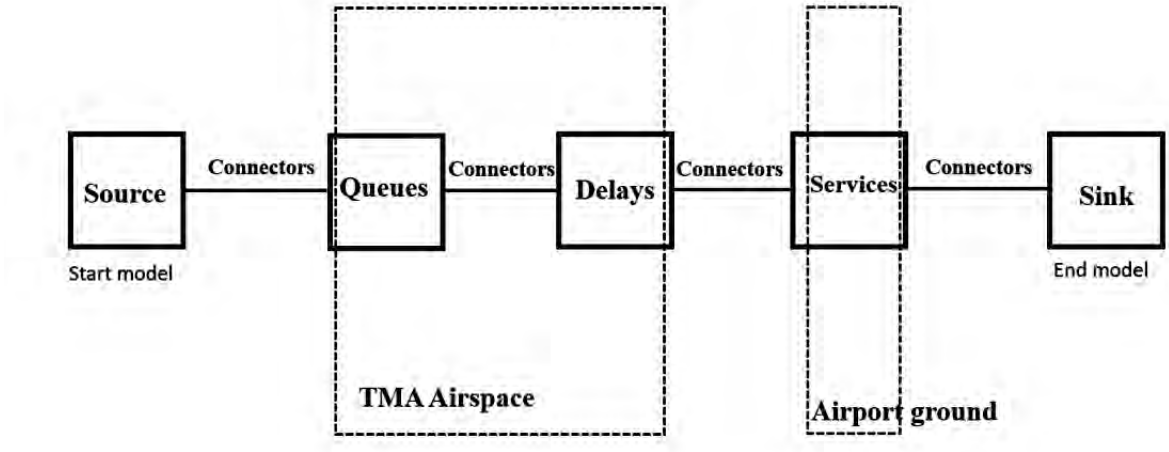


Figure B.1: Schema of the simulation model of the Opt-sim Method 2

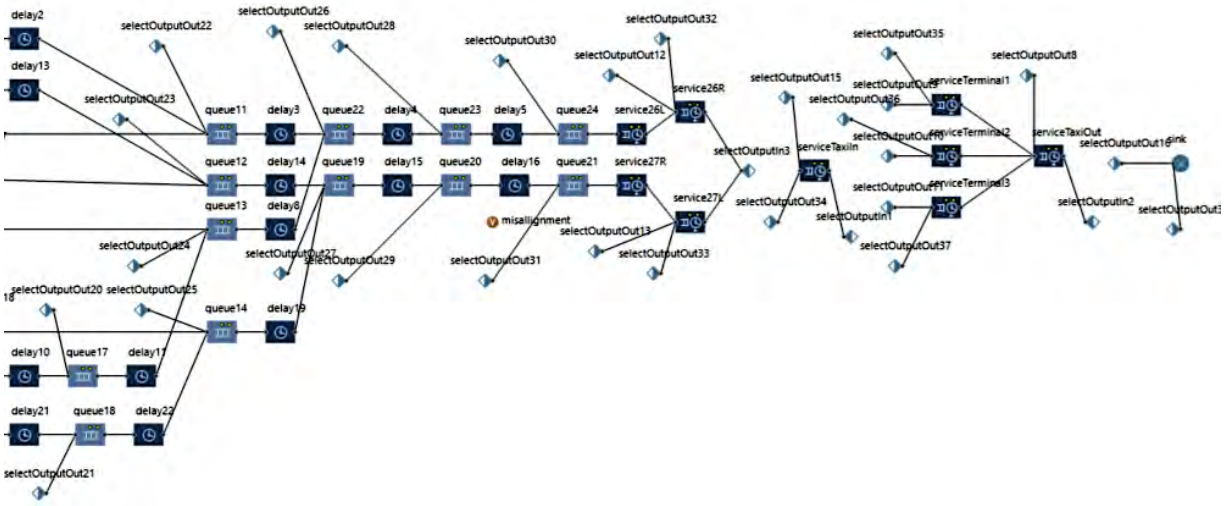


Figure B.2: Airport airspace and ground model structure (Anylogic)

Regarding the modeling of airport TMA airspace, the segments and waypoints of the airspace landing routes were modelled by using specific objects called “delay” and “queue”. Figure B.3 shows the animation of the TMA airspace. In the following, the objects used for modeling the TMA airspace are described.

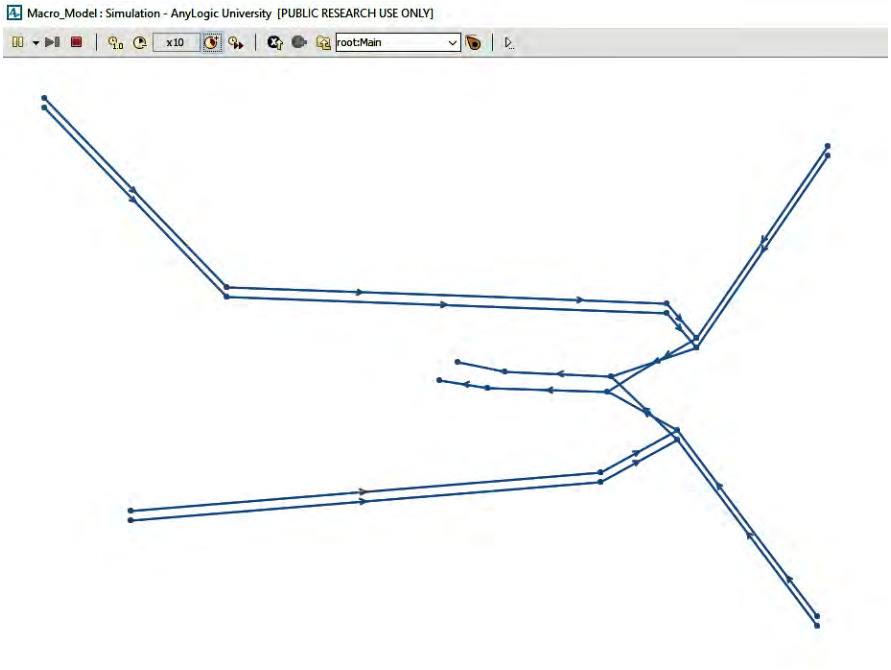


Figure B.3: Airspace visualization (Anylogic)

- Model delays. Delay objects represent the segments of the airspace landing routes. The object delay is characterized by a time value, delay time, in this way it could have been modeled the aircraft travel time on the segment. Figure B.4 shows the property window of the delay object. In the property window of this object, it can be seen the property *delay time* and *capacity*, moreover, in the section "Action", advanced logic can be added by calling functions or by directly coding.

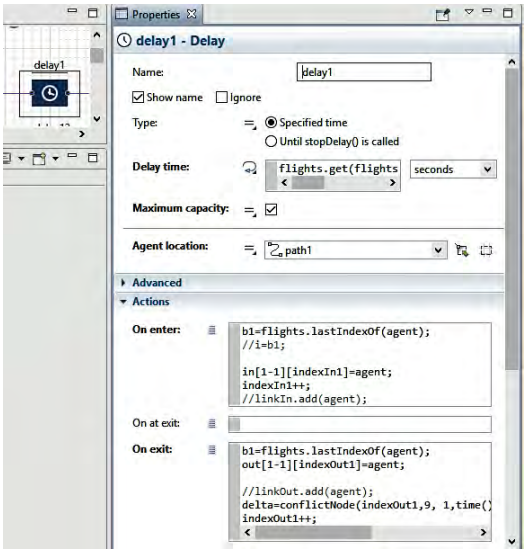


Figure B.4: Delay object properties (Anylogic)

- Model queues. Queue objects represents the waypoints of the airspace landing routes. Figure B.5 shows the property window of the queue object. The property window is similar to the one for the delay object, the only difference is that the queue object does

not have a property to set the time. Moreover, similarly as the delay object, in the "Action" section it is possible to implement logic by calling functions or directly coding.

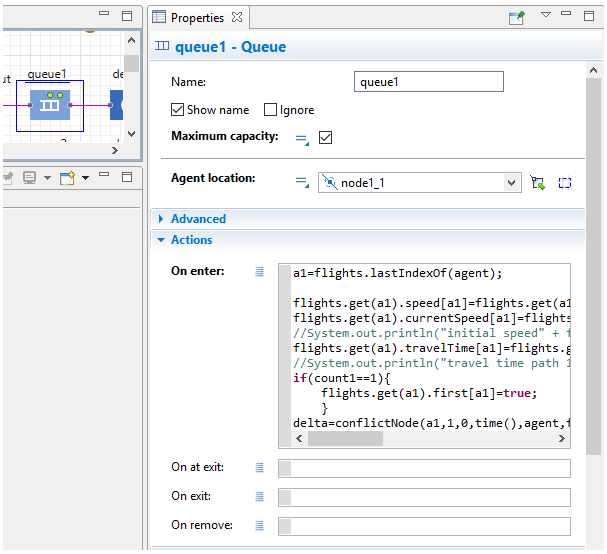


Figure B.5: Queue object properties (Anylogic)

Regarding the ground, the main components such as runways, taxiway network and terminals were modelled by service objects. Moreover, the aircraft entities were created by the object source, and subsequently destroyed by the object sink at the end of the model. Figure B.6 shows the animation of the airport ground side components. In the following the objects used for modelling the airport ground side components are described.

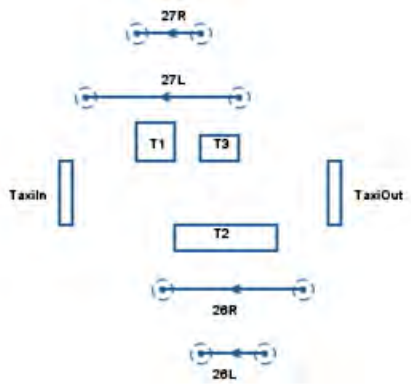


Figure B.6: Ground side visualization (Anylogic)

- Models services. Service objects were used for modeling ground components such as runways, taxiway network and terminals. In Figure B.7 is shown the property window of the service object. This object can be viewed as the union between the queue object and delay object, therefore, the main properties are the same as these two objects: delay time and queue capacity. Similarly as the queue and delay objects, in the service object there is the "Action" section where extra and advanced logic can be implemented by calling functions and directly coding.

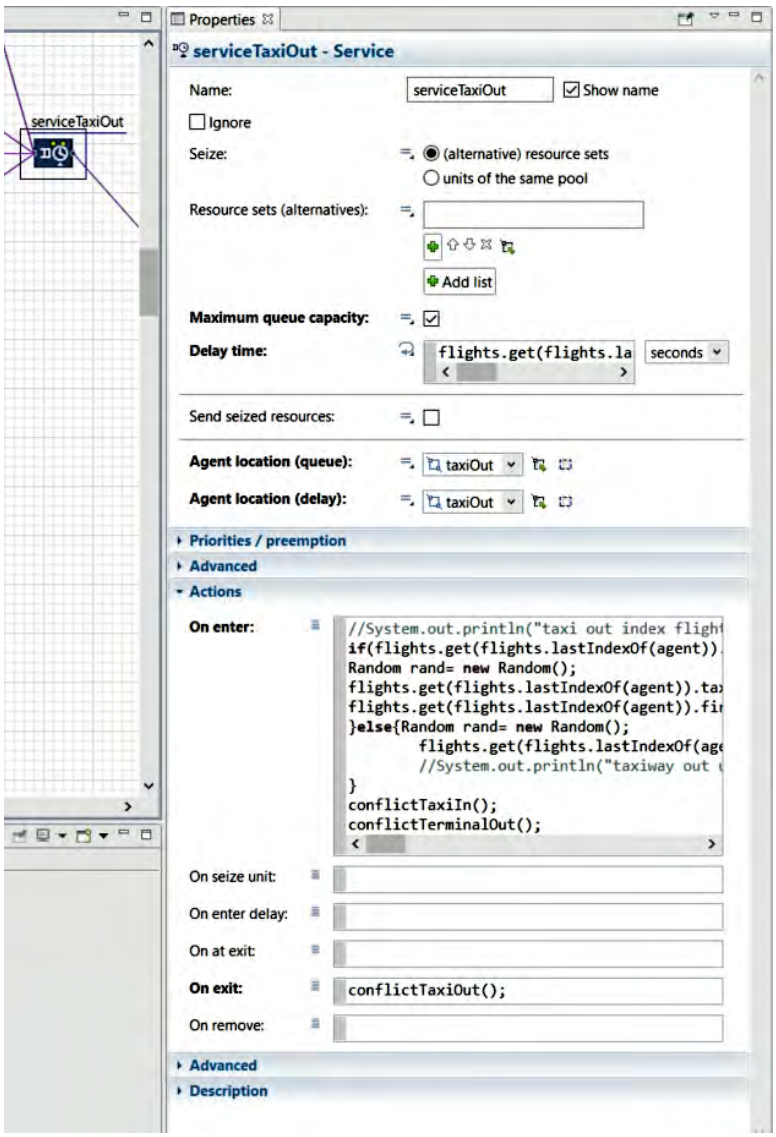


Figure B.7: Service object properties (Anylogic)

- Model source. The source object is used for creating entities of type aircraft. The source generates entities according to a flight schedule. Figure B.8 shows the property window of this object. In this context, the flight schedule needs to be stored in a database so that the source object can find it. In the source object there is also the “Action” section for implementing extra advanced logic.
- Model sink. The sink object is used for destroying the entities and terminating the model. As it can be seen in Figure B.9 the property window is composed mainly by the “Action” section, where it is possible to implement extra and advanced logic.

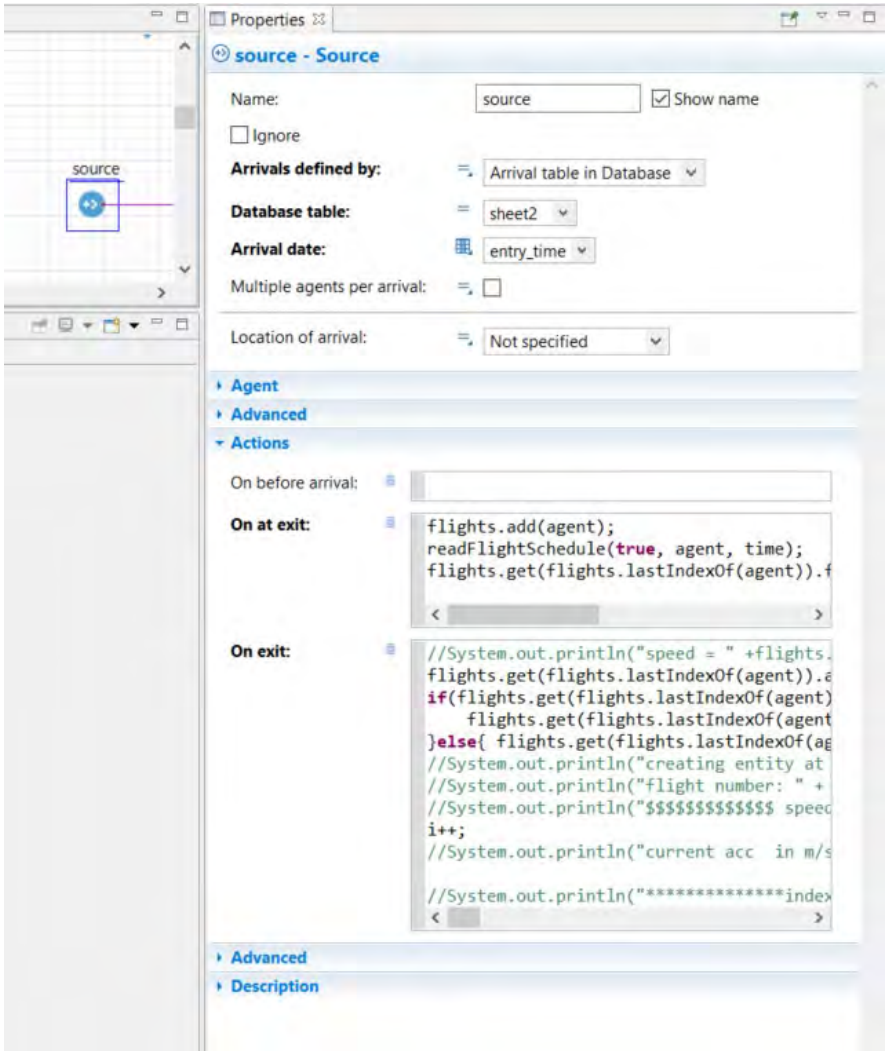


Figure B.8: Source object properties (Anylogic)

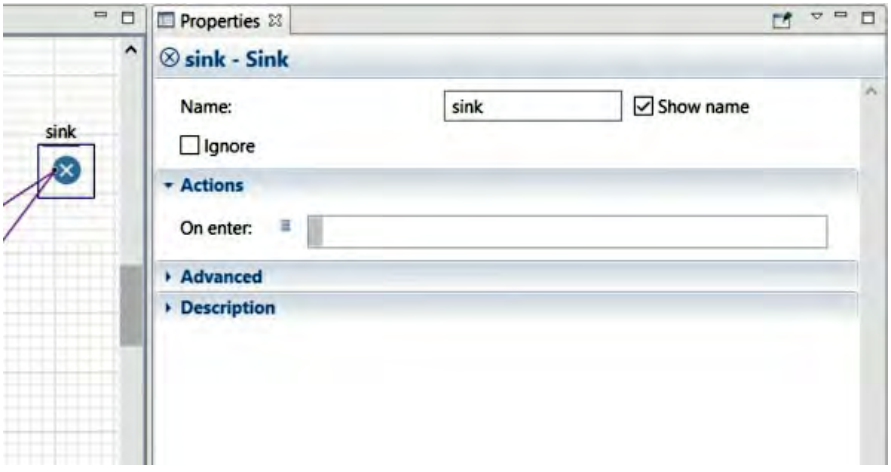


Figure B.9: Sink object properties (Anylogic)

Model logic. Implementing the logic of the model in Anylogic is possible with the use of the functions or by directly coding in some specific area of the objects. Anylogic simulation software is based on the Java programming language, therefore, any function can be created by implementing the code in the Java syntax (see Figure B.10). Once a function is created it can be called by different objects within the model.

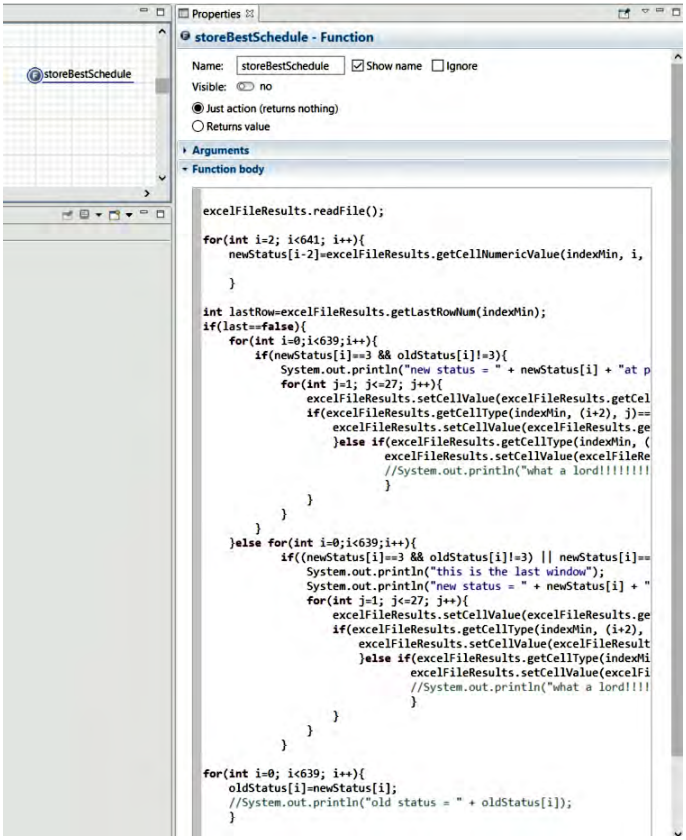


Figure B.10: Function in Anylogic

In Table B.1, the main functions implemented in the simulation model are listed, providing with their descriptions and also with their locations within the model.

Table B.1. List of functions implemented in the model

Function	Function logic	Location in the model
readFlightSchedule	Get the data from the input flight schedule.	Source object (enter)
calculateAcceleration	Calculates the acceleration for each aircraft.	Source object (exit)
selectNode	Assigns the airspace route to each aircraft.	SelectOutputIn object
calculateSpeed	Calculates aircraft speed.	Queue object (enter)
calculateTravelTime	Calculates the aircraft travel time for each airspace segment.	Queue object (enter)
conflictNode	Detects airspace conflicts.	Delay object (exit)
selectRwyInNode	Assigns the landing runway to each aircraft.	SelectOutputIn3 object
conflictRunway	Detects landing and/or departing runway conflicts.	Service object (enter) (Landing/departing runway)
conflictTaxiwayIn/out	Calculates taxiway in/out utilization. Detects taxiway in/out overload.	Service object (enter) (taxiway in/out)
selectTerminalNode	Assigns the terminal to each aircraft.	SelectOutputIn1 object
conflictTerminal	Calculates terminals utilization. Detects terminals overload.	Service object (enter)
selectRwyOutNode	Assigns the departing runway to each aircraft.	SelectOutputIn2 object
storeSchedule	Stores the schedule coming from the optimization model solution.	Custom Experiment
storeBestSchedule	Stores the best schedule coming from the optimization model solution based on the simulation results.	Custom Experiment

Publications

- Journal Papers
  - Tackling uncertainty for the development of efficient decision support system in air traffic management, IEEE Transactions on Intelligent Transportation Systems (accepted). Scala, P., Mujica, M., Ma, J., Delahaye, D
  - Integrated optimization of terminal maneuvering area and airport at the macroscopic level, Transportation Research Part C, Vol. 98 (2019), pp. 338-357. Ma, J., Delahaye, D., Sbihi, M., Scala, P., Mujica, M.
- Book Chapters
  - Improving Airport Performance through a Model-based analysis and optimization approach, Applied Simulation and Optimization 2, New applications in Logistics, Industrial and Aeronautical Practice, Mujica Mota and De La Mota Eds.,Springer. Mujica, M., Scala, P., Delahaye, D.
- Conference papers
  - A generic framework for modeling airport operations at a macroscopic level, Winter Simulation Congress, Dec., 2019, Maryland (US). Scala, P., Mujica, M., Delahaye, D., Ma, J.
  - Sim-opt in the loop: algorithmic framework for solving airport capacity problems, Winter Simulation Congress, Dec, 2018, Gothenburg, Sweden. Scala, P., Mujica, M., Wu, C.L., Delahaye, D.
  - Integrated Optimization of Arrival, Departure, and Surface Operations, International Conference for Research in Air Transportation, June 2018, Barcelona, Spain. Ma, J., Delahaye, D., Sbihi, M., Scala, P.
  - A Down To Earth Solution: Applying a Robust Simulation Optimization Approach to Resolve Aviation Problems, Winter Simulation Congress, Dec. 2017, Las Vegas, Nevada (US). Scala, P., Mujica, M., Delahaye, D.
  - A study of Tradeoffs in Airport Coordinated Surface Operations, EIWAC 2017, Nov. 2017, Tokyo, Japan. Ma, J., Delahaye, D., Sbihi, M., Scala, P., Mujica, M
  - No more conflicts: the development of a generic airport model in a sequence optimization framework, EMSS 2016, Sep. 2016, Larnaca, Cyprus. Scala, P., Mujica, M., Delahaye, D.
  - Implementation of an optimization and simulation based approach for detecting and resolving conflicts at airports, EUROSIM 2016, Sep. 2016, Oulu, Finland. Scala, P., Mujica, M., Delahaye, D.
  - Methodology for assessing and optimizing operation performance in airport systems, EIWAC 2015, Nov. 2015, Tokyo, Japan. Mujica, M., Scala, P., Delahaye, D.