Optimal Design of SIDs/STARs in Terminal Maneuvering Area

Jun Zhou

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *28/04/2017* **par :**

Jun ZHOU

Optimisation des procédures de départ et d'arrivée dans une zone terminale

Optimal Design of SIDs/STARs in Terminal Maneuvering Area

JURY

HAMSA BALAKRISHNAN	Professeur	Rapporteur
ADNAN YASSINE	Professeur	Rapporteur
PIERRE MARECHAL	Professeur	Examinateur
VALENTIN POLISHCHUK	Professeur	Examinateur
SONIA CAFIERI	Professeur	Directeur de thèse
MOHAMMED SBIHI	Enseignant-chercheur	CoDirecteur de thèse

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Laboratoire ENAC
Directeur(s) de Thèse :
Sonia CAFIERI et *Mohammed SBIHI*
Rapporteurs :
Hamsa BALAKRISHNAN et *Adnan YASSINE*

Résumé

L'objectif de cette thèse est de proposer une méthodologie d'optimisation des routes de départ et d'arrivée dans une zone terminale autour d'un aéroport, dite Terminal Maneuvering Area (TMA). En effet, les départs d'un aéroport se font suivant des routes, dites Standard Instrument Departure (SID), qui relient les pistes aux points de sortie de la TMA. Les arrivées sur un aéroport se font suivant des routes, dites Standard Terminal Arrival Routes (STAR), qui relient les points d'entrée de la TMA aux pistes. La conception de ces SID/STAR s'inscrit dans le cadre de la conception de l'espace aérien, qui est un problème très critique, à cause notamment des prévisions de forte croissance du trafic aérien et de la congestion qui en résulte.

Selon différentes études, le trafic aérien mondial va subir une croissance de 4 à 5% par an au cours des 20 prochaines années. Cette forte augmentation aura comme conséquence la saturation de l'espace aérien, particulièrement celui entourant les aéroports. Pour faire face à cette augmentation du trafic, de nouveaux aéroports et/ou pistes peuvent être construits. Cependant, ce type de solutions conduit habituellement à des coûts et à des délais de construction élevés. La conception efficace des routes SID/STAR constitue alors un autre levier pour augmenter, à moindre coût, la capacité des TMA, et ainsi réduire la congestion autour des aéroports. Dans cette optique, cette thèse s'intéresse au problème de conception optimale des SID/STAR, en prenant en compte la configuration et l'environnement autour des aéroports, et les différentes contraintes opérationnelles sous-jacentes. L'évitement des obstacles et la séparation des routes en constituent les contraintes principales.

Nous proposons une formulation mathématique conduisant à un problème d'optimisation combinatoire, ainsi que des méthodes de résolution *ad hoc* efficaces pour le problème. Dans notre approche, une route est modélisée en 3D par deux éléments: une courbe dans le plan horizontal, formée par une succession d'arcs de cercles et de segments, et un cône associé dans le plan vertical, contenant tous les profils de montée (ou de descente) des vols sur cette route. Quant aux obstacles, ils sont modélisés par des cylindres.

Pour la résolution du problème, nous procédons en deux étapes. Nous considérons dans un premier temps la conception d'une route de longueur minimale évitant les obstacles. Nous proposons une approche d'optimisation globale déterministe, basée sur une méthode de type Branch and Bound (B&B). Dans ce B&B, les stratégies de branchement sont liées à la forme des routes et aux manières d'éviter des obstacles (contournement dans le sens horaire ou anti-horaire, ou imposition d'un palier sous un obstacle dans le plan vertical).

Dans un deuxième temps, nous nous intéressons à la conception de plusieurs routes. Dans ce cas, la difficulté principale est d'assurer la séparation des routes. Nous proposons deux approches pour y faire face. Dans la première approche, basée sur la méthode B&B développée pour une route, nous construisons les routes séquentiellement suivant un ordre fixé à l'avance (par exemple, suivant la charge du trafic). Plus précisément, nous construisons d'abord les routes individuellement à l'aide de la méthode de B&B, définie précédemment. Ensuite, les routes sont déviées localement autour des zones de conflit (zones où les routes ne se sont pas séparées). Pour effectuer la déviation des

routes, des obstacles fictifs cylindriques enveloppant les zones de conflits sont introduits et ensuite évités à l'aide de la méthode B&B défini précédemment. Comme les routes obtenues par cette approche dépendent fortement de l'ordre de construction des routes, nous proposons une seconde approche utilisant le recuit simulé pour construire les routes simultanément. Des obstacles fictifs cylindriques sont aussi introduits, pour envelopper les zones de conflit, et ensuite évités suivant des stratégies sélectionnées aléatoirement dans la méthode de recuit simulé.

Notre approche est validée sur un ensemble de problèmes tests générés artificiellement et sur des problèmes correspondants à des TMA existantes (Paris CDG et Zurich). Dans le premier ensemble de problèmes tests, diverses configurations de TMA (nombre et disposition des obstacles, pistes, positions des points d'entrée et de sortie de TMA) sont considérées. Les routes obtenues sont continues et lisses, adaptées aux opérations de montée continue (CCO) et aux opérations de descente continue (CDO). Dans les tests effectués sur des TMAs réelles, le choix de Paris CDG et de Zurich a permis de confronter notre approche, d'une part, à une TMA comportant de nombreux points d'entrée/sortie, et d'autre part, à une TMA comportant de nombreux obstacles. Les résultats montrent une réduction de la longueur totale des routes par rapport aux SID/STAR publiées, ce qui est intéressant du point de vue de la réduction de la consommation de carburant. Par ailleurs, les tests sur la TMA de Zurich montrent que notre approche peut être appliquée efficacement en présence de nombreux obstacles, comme les montagnes entourant l'aéroport de Zurich.

Abstract

The objective of this thesis is to propose a methodology for the optimization of departure and arrival routes in the airspace surrounding airports, named Terminal Maneuvering Area (TMA). The air traffic departing from and arriving to airports follows pre-designed routes named Standard Instrument Departure (SID) routes, that connect the runways to the TMA exit points, and Standard Terminal Arrival Routes (STAR), that connect the TMA entry points to the runways. The design of SIDs/STARs falls into the area of airspace design problems, that are very critical, mainly due to the predictions of air traffic growth and the consequent traffic congestion.

In fact, according to several studies, the world-wide air traffic is projected to grow 4 to 5 percent annually in the next 20 years. This sharp increase leads directly to the capacity insufficiency of the airspace surrounding airports. In order to adapt the capacity of TMAs to the increased traffic demand, new airports and runways can be constructed. However, this kind of solution usually leads to high construction costs and long construction times. Designing SIDs/STARs more efficiently is another possible way to increase the capacity of TMA airspace, and so to reduce the congestion around airports. In this thesis we propose an optimal design of SIDs/STARs, taking into account the configuration and environment around airports, and the related operational constraints, in particular the avoidance of obstacles and the separation between routes.

We propose a mathematical formulation leading to a combinatorial optimization problem, as well as efficient *ad hoc* resolution methods for the problem. More precisely, each route is modeled in 3D, consisting of two components: a curve in the horizontal plane which is composed by a succession of arcs of circles and segments, associated with a cone in the vertical plane that contains all ascent (or descent) profiles of the aircraft flying on this route. Moreover, the obstacles as well as their protection area are modeled in cylinder shape.

This route design problem is solved in two steps. In the first step, we deal with the design of one optimal route avoiding obstacles with respect to minimum route length. We propose a deterministic global optimization approach based on a Branch and Bound (B&B) method. In this B&B, the branching strategies are related to the form of a route, and are tailored to the ways the obstacles are avoided (bypassing clockwise or counter-clockwise, or imposing a level flight below an obstacle in the vertical plane).

In the second step, the problem of designing multiple routes is considered. The main difficulty in this case is to ensure the pairwise separation between routes. We propose two approaches to deal with the design of multiple routes. The first is a B&B-based approach, where routes are generated sequentially in a given order (for example, in a decreasing order of the traffic load). We first build routes individually using the B&B method defined previously. Then, the routes are deviated locally around the conflicting zones (zones where routes lose the minimum separation norm). In order to carry out the route deviation, fictitious obstacles in cylinder shape enveloping the conflicting zones are introduced, and then avoided by using the B&B. The quality of a solution provided by the B&B-based approach depends on the routes generation order. Thus another method building routes simultaneously is proposed, that is the Simulated Annealing (SA) method. Fictitious obstacles in

cylinder shape are also introduced to envelop conflicting zones, and their avoidance strategies are randomly selected in the SA method.

Our approach is validated on a set of artificially generated problems as well as on two problems corresponding to existing TMAs (Paris CDG and Zurich). Concerning the first set of test problems, several configurations of TMA (number and layout of obstacles, runways, positions of the TMA entry/exit points) are considered. We show that we obtain continuous and smooth routes which are suitable for Continuous Climb Operation (CCO) and Continuous Descent Operation (CDO). Concerning the tests performed on real data, the choice of the TMAs of Paris CDG and Zurich allows us to test our approach, on the one hand, on a TMA with numerous TMA entry/exit points, and on the other hand, on a TMA with many obstacles. The simulation results show a gain in the total route length compared with the published standard charts, that can be promising in terms of reducing jet fuel consumption. Furthermore, tests on the TMA of Zurich show that our approach can be applied effectively in a TMA in the presence of several obstacles, such as the mountains surrounding the Zurich airport.

Acknowledgements

I would like to take this chance to express my sincere thanks to the people who have guided and helped me to accomplish my dissertation, as well as to the people who have supported me and shared the pleasant time during my PhD study. Because of you, I could have learned a lot of things in the professional area and got so many shining memories.

First of all, I wish to give my deepest gratitude to my supervisors, Sonia Cafieri and Mohammed Sbihi, for their excellent guidance, caring and encouragements. They not only taught me vast mathematical knowledge, but also showed me the correct way of working, always be rigorous, be active and be patient. Their advice on both research and on my career have been invaluable. I would also like to thank Daniel Delahaye, for accepting me into the optimization group and being my advisor during my internship. He always provided me guidelines and inspired me with new ideas whenever I got lost in the research. I was fortunate to have the chance to work with them and I hope that we can continue to collaborate in the future.

Many thanks also to my committee members for their interest in my work. Thanks to the reporters, Hamsa Balakrishnan and Adnan Yassine, for their precious advice and feedback that were helpful to improve my manuscript and presentation. Thanks to the examiners, Pierre Maréchal and Valentin Polishchuk, for all the interesting discussions during and after my defense, as well as for their precise remarks that inspired me to ameliorate my current work. Thank you all for letting my defense be an enjoyable moment.

My deep appreciation goes out to my university, the Civil Aviation University of China (CAUC), especially to my institute, the Sino-European Institute of Aviation Engineering (SIAE), for selecting me to study in France, and providing me the scholarship through the collaboration with China Scholarship Council (CSC) during the 5 years and a half (including 2 years of engineering study). Thanks for giving me such an excellent opportunity to expand my view and experience a totally different and attractive culture.

I greatly acknowledge Serge Roux for all the helps and supports that he provided. Thanks for resolving the infinite problems of my computer, teaching me to use Xtraj and Gnuplot, helping me to generate nice videos to illustrate the results, as well as developing the interface to visualize the route design. There are so many things that I could not have accomplished without him.

My grateful thanks also goes out to Ji Ma. Thanks for accompanying me during my depression days and bringing me the joy with her sense of humor. She could not have realized how important that meant to me. Also thank her for sharing so many unforgettable memories with me during our trips to Portugal, Spain and Switzerland. I am also very thankful to Tambet Treimuth, for helping me dealing with the problems of Java and explaining me the algorithms. I also thank him for taking charge of the massive labor that I distributed to him in order to prepare the party after my defense.

I wish equally to thank other members in the Z-building. Thanks to Stéphane Pueshmorel for his help to efficiently provide the documents for my PhD inscription every year. Thanks to Marcel Mongeau for giving us good advice in the selection of journal. Thanks to Alana Moore for interesting discussions and feedback on the manuscript for publication. Thanks to Gilles Baroin for

his help in formatting the figures for publication. Thanks to Andrija Vidosavljevic for his useful and professional suggestions after the repetition of my defense which helped a lot to improve my presentation.

I would also like to say a heartfelt thank you to other PhD students, for bringing me so many happy and wonderful moments. Thanks to Romaric Breil for inviting us constantly for a coffee break, and organizing the PhD students for a dinner together from time to time. Thanks to Man Liang for the interesting discussions that provide me some guidelines for my future work. Thanks to Arianit Islami for bringing the delicious chocolates, providing me the priceless slides in the ATM domain, and sharing me his working experience. Thanks to Ruixin Wang for inviting me to jogging and helping me sending my packages back to China. Thanks to Paolo Scala for his sense of humor and sharing the picnic at Pech David. Thanks to Florian Mitjana for organizing the seminars among PhD students. Also thanks to him for scaring me from time to time when I was sleepy, this was a good way to wake me up. Thanks to Ning Wang for sharing with me the experience to come to France at the first time. Thanks to Isabelle Santos for teaching me piano and taking me for a pleasant flight. Thanks to Imen Dhief for sharing the delicious dessert and her culture. Thanks to Vincent Courjault-Radé for practicing chinese and enjoying the chinses snacks. Thanks to Marina Sergeeva and Clément Bouttier for our splendid visit to Marseille during the conference ROADEF2015. Thanks to Karim Legrand for the long discussion on the route design problem and giving me many advice from the professional point of view. Thanks to Jérémie Chevalier for the interesting discussion and the precise remarks on my manuscript.

I greatly appreciate some graduated PhD students for helping me to adapt to the environment and making good examples for me. Thanks to Supatcha Chaimatanan for all the interesting chats when we were office-mate and the patient discussion to explain me her method, also for bring me so many tasty Thai food. Thanks to Olga Rodionova for making delicious salad for our picnic and visiting Montreal together during the conference Cors/Informs2015. Thanks to Brunilde Girardet for explaining me the Fast Marching Method during my internship and providing me her Java code. Thanks to Laureline Guys for preparing the delicious pancakes. A special thank you goes out to Daichi Toratani for his positive altitude to everything and for his excellent guidance to visit Tokyo.

Finally, I would like to specially acknowledge my boyfriend and my family. I am grateful for their love, sacrifice and tolerance. Thanks for always believing in my abilities and encouraging me during the challenging period. Words can not express how grateful I am to them.

To anyone that I may have forgotten. I apologize and thank you as well.

I wish a bright future to you all.

Contents

List of Figures

List of Tables

Abbreviations

ACC	**A**rea **C**ontrol **C**enter
ACO	**A**nt **C**olony **O**ptimization
ADS-B	**A**utomatic **D**ependent **S**urveillance-**B**roadcast
AIS	**A**eronautical **I**nformation **S**ervice
ARTCC	**A**ir **R**oute **T**raffic **C**ontrol **C**enter
ATC	**A**ir **T**raffic **C**ontrol
ATFM	**A**ir **T**raffic **F**low **M**anagement
ATM	**A**ir **T**raffic **M**anagement
ATS	**A**ir **T**raffic **S**ervice
BADA	**B**ase of **A**ircraft **DA**ta
B&B	Branch and Bound
B-spline	Basis spline
BST	Best route length
CCO	**C**ontinuous **C**limb **O**peration
CDO	**C**ontinuous **D**escent **O**peration
CTR	Control zone
CVH-filter	Convex hull filter
DataComm	**Data Comm**unications
FAF	**F**inal **A**pproach **F**ix
FI	**F**irst **I**ntersected
FIR	**F**light **I**nformation **R**egion
FMM	**F**ast **M**arching **M**ethod
FMS	**F**light **M**anagement **S**ystem
GA	**G**enetic **A**lgorithm
GDP	**G**ross **D**omestic **P**roduct
GI	**G**reatest **I**ncursion
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem
IAF	**I**nitial **A**pproach **F**ix
ICAO	**I**nternational **C**ivil **A**viation **O**rganization

IF	**I**ntermediate **F**ix
IFR	**I**nstrument **F**light **R**ule
ILS	**I**nstrument **L**anding **S**ystem
IP	**I**nteger **P**rogramming
LCO	**L**east **C**onsidered **O**bstacles
LI	**L**east **I**ncursion
LPA	**L**ight **P**ropagation **A**lgorithm
MCO	**M**ost **C**onsidered **O**bstacles
NextGen	**Next Gen**eration Air Transportation System
PBN	**P**erformance **B**ased **N**avigation
PRM	**P**robabilistic **R**oadmap
PSO	**P**article **S**warm **O**ptimization
RF	**R**adius-to-**F**ix
RNAV	Area navigation
RNP	**R**equired **N**avigation **P**erformance
RRT	**R**apidly-exploring **R**andom **T**ree
SA	**S**imulated **A**nnealing
SESAR	**S**ingle **E**uropean **S**ky **A**TM **R**esearch
SID	**S**tandard **I**nstrument **D**eparture
STAR	**S**tandard **T**erminal **A**rrival **R**oute
TCA	**T**erminal **C**ontrol **A**rea
TMA	**T**erminal **M**aneuvering **A**rea
TOC	**T**op **o**f **C**limb
TOD	**T**op **o**f **D**escent
TRACON	**T**erminal **R**adar **A**pproach **CON**trol
UAV	**U**nmanned **A**erial **V**ehicle
UIR	**U**pper **I**nformation **R**egion
VFR	**V**isual **F**light **R**ule

Introduction

Air transportation is one of the most modern way of transport, and its major advantage lies in its quickness. It provides not only vital economic benefits but also significant social benefits. From the economic point of view, air transportation builds up a worldwide network (Fig. 1) that promotes and encourages global business and tourism. According to [11], in the year 2015, approximately 3.6 billion passengers globally are transported by air, which is 6.8 percent higher than the previous year, and about 54 percent of international tourists travel by air. In addition, about 62.7 million new jobs are generated through direct, indirect or induced impacts of air transportation industry in the world [11]. From the social welfare point of view, air transportation improves the quality of life by facilitating traveling, especially over long-distance. This also provides possibilities to enrich people's leisure and cultural experiences. Besides, air transportation enables the prompt delivery of emergency aids anywhere on earth.

Playing such an important role in the transport industry, air transportation has a huge potential of development in the future. The Gross Domestic Product (GDP) is one of the key indicators within the aviation market, and the increase of world GDP leads to a continuous growth of the air traffic. According to [12], airline passenger traffic and air cargo traffic are expected to grow at annual rates of 4.8 and 4.2 percent respectively in the next 20 years. The Asia-related markets as well as the Latin America market have the most important increase. The developed market such as the markets inside Europe and North America have a relatively steady increment around 3 percent in the same

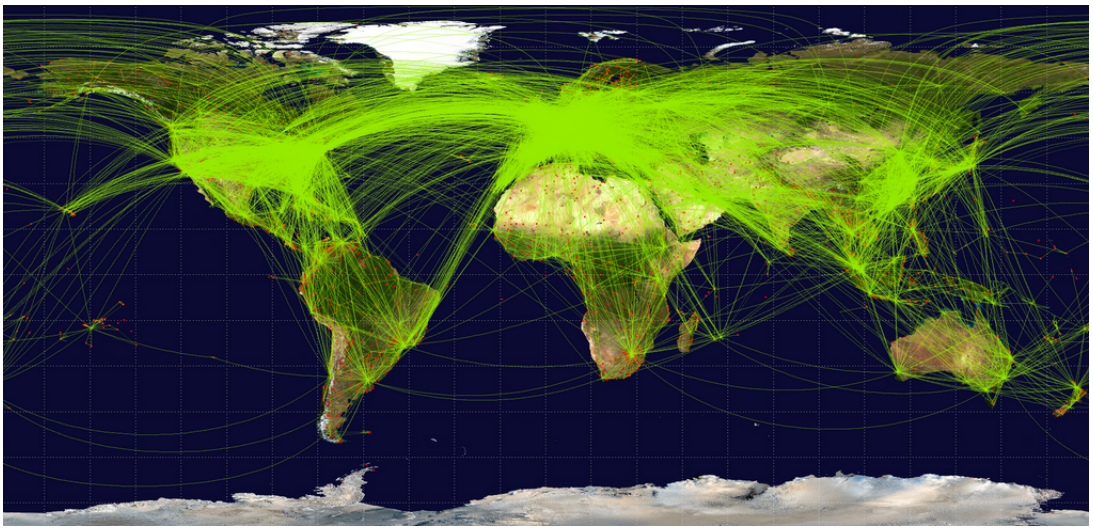


Figure 1: Global air routes network[6].

[6]source: `http://openflights.org/data.html`

period. In the year 2015, about 48 percent of people travel with airlines based in Europe and North America, while this market share is anticipated to shrink to 37 percent in the year 2035.

The promising market forecast of the aviation industry brings not only opportunities but also challenges. The main challenge is that the increase in air traffic demand is limited from the supply side by the capacity of the network. The extra traffic demand, that is not able to be absorbed by the network, will result in congestions. This is especially true in the airspaces surrounding airports, which serve as both the starting and ending points of the air traffic. The traffic amount is very huge in such areas, while the airspace is relatively limited. When the capacity of airports and the surrounding areas is insufficient, congestion surrounding airports increases quite rapidly, which leads to extra pressure on the network and causes more delays. According to [1], around 1.9 million flights are predicted to be unaccommodated in the year 2035 in Europe, accounting for 12 percent of the demand. More detailed information on the mismatch between capacity and demand in European countries is presented in Fig. 2, the shortfall is especially important in the countries such as Turkey, Bulgaria, Hungary and Romania. It can be seen that future air traffic operations will be limited by airports capacity. To face this challenge, new runways and airports are built in order to absorb the pressure on the network. However, this kind of solution leads to very high costs and relatively long construction period. In the future, new projects aiming at developing advanced technologies and improving airport infrastructures may be launched. It is also necessary to further ameliorate the operating environment and to allow more automation and more efficient utilization of the airspace.

Another challenge that aviation industry has to face lies in the environmental sustainability. Aviation environmental impacts include noise pollution as well as CO_2 emissions and air quality impacts from the burning fuel. This challenge is especially serious in the airspaces surrounding airports, since these areas are very close to the city residents, and people's awareness of the impact brought by aviation industry becomes more and more sensitive and critical. In fact, notable results, aiming at reducing the environmental impacts, have been achieved nowadays, as a result of long-standing investigations. According to [1], current average noise level is reduced by 20 dB compared with 40 years ago, corresponding to a reduction in noise annoyance of 75 percent. Moreover, the

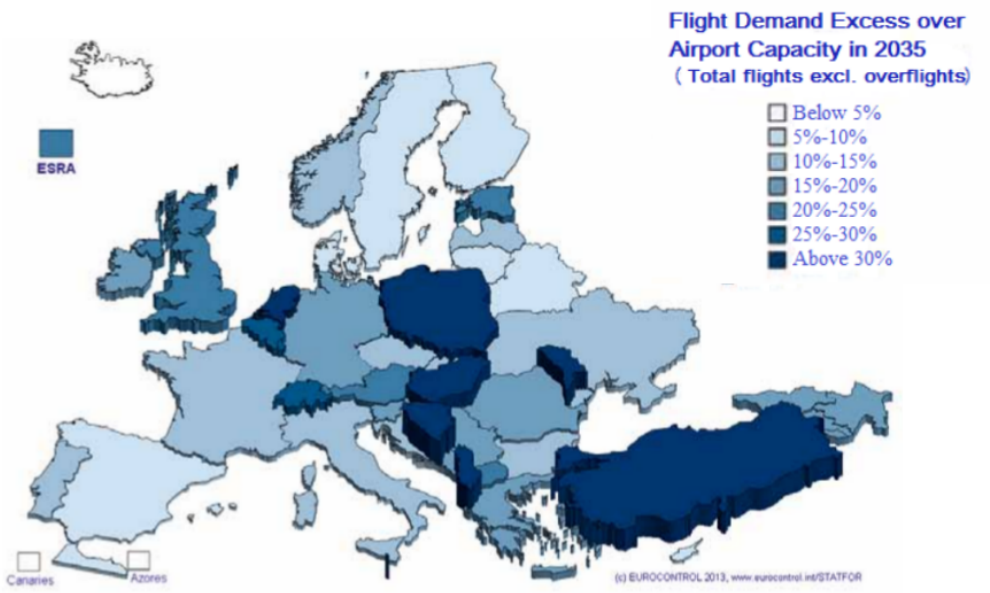


Figure 2: Flight demand excess over airport capacity in Europe in 2035 (source [1]).

CO_2 emission decreased from 160 grams per passenger km in 1995 to 120 grams per passenger km in 2010. Even though, these improvements are unlikely to compensate the continuous increase of the air traffic demand. Moreover, it is hard to make trade-offs between the mentioned environmental impacts. For example, re-routing aircraft in order to avoid noise-sensitive areas may lead to longer routes, which will result in more fuel consumption, thus more CO_2 emissions. While operating more fuel-efficient and noise-reduced procedures, such as Continuous Decent Operation (CDO) and Continuous Climb Operation (CCO), will limit the capacity of the airspaces surrounding airports. Thus, implementing these procedures is not an easy task, and more investigations and researches are needed.

Airports play an important role in the future development of the aviation industry. The airspace surrounding airports is named Terminal Maneuvering Area (TMA), which is designed to handle aircraft arriving to and departing from airports. It is crucial to increase the capacity of TMAs, and thus to deal with the congestion around airports caused by the worldwide air traffic growth. Most of the airports have pre-designed procedures indicating how aircraft depart from or arrive to airports. These procedures are called Standard Instrument Departure (SID) route and Standard Terminal Arrival Route (STAR). More precisely, a SID is a flight route which is followed by aircraft from its take-off phase until it starts the en-route phase, and a STAR is a route which connects the last en-route way-point to the Initial Approach Fix (IAF). In this thesis, the problem of designing SIDs/STARs is addressed in an optimal way, taking into account the configuration of the airport and nearby environment, as well as related operational constraints, including the avoidance of obstacles and the separation between routes. The designed routes use the airspace more efficiently, thus more traffic can be accommodated.

This thesis is organized in the following way. Chapter 1 introduces the context of the considered problem. An overview of the current Air Traffic Management (ATM) system and its future development trends is given. Then, different types of controlled airspaces, especially the Terminal Maneuvering Area (TMA), are introduced. We give a short introduction to mathematical optimization problems and resolution methods, with an emphasis on the optimization methods used in this thesis. Finally, the objectives and contributions of this thesis are given. Chapter 2 first reviews some existing methods in literature related to path planning problems. These methods are classified into two groups: roadmap based methods and path shape based methods. Then researches specifically dealing with the route design in ATM domain are presented, and the optimization methods applied in these works are discussed. Finally, we give a brief overview of the works related to trajectory planning. In Chapter 3, a mathematical framework to deal with the routes design problem is presented. First, the 3D routes and obstacles modeling are given, then the routes design problem is formulated as a combinatorial optimization problem. In Chapter 4, the way one single optimal route is designed using a Branch and Bound (B&B) method is proposed. Besides, a technique to reduce efficiently the state space is also explained. Numerical results of several artificially generated problems are given. The preliminary results of this method have been presented at EIWAC[7] conference in 2015 [13], and have been published in the Lecture Notes [14]. In Chapter 5, we develop two different approaches to solve the design of multiple routes. The first one is a B&B-based approach, where routes are generated sequentially. This approach is characterized by the B&B to generate a single route, and a strategy to deviate a conflicting route. This approach as well as some simulation results will be published in [15]. The second one is a typical heuristic approach, named Simulated Annealing (SA), where routes are generated simultaneously. The preliminary results of the SA method have been presented at DASC[8] conference in 2016 [16]. In Chapter 6, the two proposed

[7]EIWAC2015, the 4th International Workshop on ATM/CNS, Tokyo, Japan

[8]DASC2016, the 35th Digital Avionics Systems Conference, Sacramento, CA, United States

approaches for designing multiple routes are validated first on several artificially generated problems, then applied to two problems corresponding to existing TMAs (Paris and Zurich). The results obtained from both approaches are compared and discussed. Finally we draw a conclusion for this thesis and discuss about the future research directions.

Chapter 1

Problem Context

In this chapter, we first introduce the current Air Traffic Management (ATM) system, including its definition and main activities. We also introduce some advanced technologies in a modernized ATM system to improve safety and increase efficiency in air traffic operation. In Section 1.2, different kinds of controlled airspaces, as well as the corresponding flight phases managed by them are presented. In Section 1.3, the controlled airspace named Terminal Maneuvering Area (TMA) is presented in more detail, including its general layout as well as related operational and environmental requirements. In Section 1.4, we give a short overview of optimization problems and resolution methods, with an emphasis on the methods applied in this thesis. Finally, we give the objectives and contributions of this thesis.

1.1 Current Air Traffic Management (ATM) system and its future trends

Air Traffic Management (ATM) system generally refers to all the activities involved in ensuring the safe and orderly flow of air traffic flying between pairwise airports. Being such a huge and complex system, the functionality of ATM can be affected by many aspects. Effective planning is indispensable to guarantee that ATM operates safely and efficiently. Depending on the time horizon, the planning in ATM can be classified into three levels: strategic level, pre-tactical level and tactical level.

- A *strategic level planning* takes place about one year to several weeks before the real-time operations. Some macroscopic indicators, such as the traffic load, the congestions and complexity are estimated at this level. The design of route structure is also realized at this level, in order to optimize the traffic flow from scratch.

- A *pre-tactical level planning* is executed several weeks to one day before the real-time operations. The information of traffic demand, weather forecast and other operational conditions are provided with higher accuracy. The planning performed at strategic level can be adjusted accordingly to decrease the complexity and congestions.

- A *tactical level planning* is conducted on the day of operations. It aims primarily at aircraft conflict resolution through the departure slot allocation, speed control, re-routing and other approaches. It attempts to maximize flights efficiency and to make full use of the available airspace.

In order to ensure the safe and efficient operation of air traffic, the ATM system provides three main services:

- *Air Traffic Flow Management* (ATFM), its primary objective is to regulate air traffic efficiently, so as to avoid or decrease the congestion. To fulfill this objective, it is very important to ensure the best possible match between supply and demand. One possible solution is to stagger the demand over time and space, another one is through better planning of the control capacities to meet the demand. According to the planning time horizon, the ATFM is at strategic level.

- *Air Traffic Control* (ATC), its main purpose is to maintain sufficient separation between aircraft, and between aircraft and obstructions on the ground. However, this safety purpose must not impede the flow of traffic. According to the planning time horizon, the ATC is at tactical level.

- *Aeronautical Information Services* (AISs), its principal responsibility is to compile and distribute all aeronautical information necessary for the safety, regularity and efficiency of air navigation to airspace users.

The current ATM system is being modernized in order to handle air traffic growth and to provide greater safety at a lower cost. Two major projects, the Next Generation Air Transportation System (NextGen) [17] in the USA and the Single European Sky ATM Research (SESAR) [18][19] in Europe, are carried out to contribute to ATM modernization. The research area include: network and strategic traffic flow optimization, air-ground integrated concepts, trajectory and queue management, airport improvement programs, etc. Through these projects, new technologies of communication, navigation and surveillance are developed and implemented to the current ATM systems.

- The *communication system* allows the interaction between pilots and air traffic controllers. In the Data Communications (DataComm) project of future ATM system, air-to-ground and ground-to-ground *data communications* are added, which enables not only automated message generation and receipt in the ground side, but also message routing and transmission to aircraft avionics. Compared with the current ways of communication, which perform largely by voice, the DataComm is more efficient and accurate. Therefore, by using DataComm, the productivity of controllers is improved which could contribute to the growth of airspace capacity.

- The *navigation system* guarantees the aircraft to follow pre-defined routes and to access airports safely. The *Performance Based Navigation* (PBN) [20], including *Area navigation* (RNAV) and *Required Navigation Performance* (RNP), is one of the key factors to enable more precise navigation in the modernized ATM system. A PBN route follows a succession of *waypoints*, which are specified geographical locations, simply defined by latitude and longitude coordinates. Compared with conventional navigation, which relies on ground-based infrastructures, the PBN routes are more flexible and have a higher level of navigation precision. Thus the airspace can be more fully used by implementing PBN routes.

- The *surveillance system* is used to monitor the traffic situation and to prevent collisions. In the modernized surveillance system, the *Automatic Dependent Surveillance-Broadcast* (ADS-B) is implemented. ADS-B uses Global Positioning System (GPS) satellite signals to determine aircraft's information, including speed and precise position, and also broadcasts these data

to other aircraft and air traffic controllers. Compared with the currently used Radar system, the ADS-B has a more accurate and reliable performance. With the complete implementation and application of ADS-B, both the pilots and the controllers can share the same real-time information of air traffic.

Through the developments in communication, navigation and surveillance technologies, an operating environment with increased safety, efficiency and capacity, and with reduced delay and cost can be achieved. The route design problem considered in this thesis takes into account the navigation mode PBN, more precisely, the RNP under PBN. Thus more details about these navigation systems are presented in the following.

The navigation mode PBN not only defines the performance requirements for routes and procedures designing, but also offers operational benefits such as enhanced safety and increased efficiency. As mentioned, it consists of RNAV and RNP. The mode RNAV allows aircraft to fly on any desired path within the coverage of ground- or space-based navigation aids [21]. The mode RNP is fundamentally similar to RNAV, but it requires additionally on-board performance monitoring and alerting [20]. Furthermore, RNP enables the Radius-to-Fix (RF) functionality, defined as an arc with specified center and radius between two defined waypoints [2]. An illustration of RF is shown in Fig. 1.1.

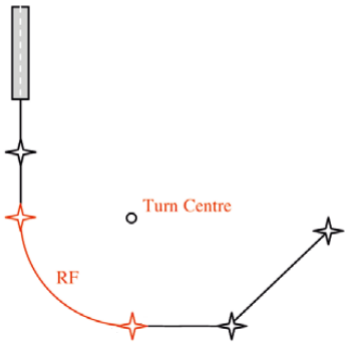


Figure 1.1: Radius-to-Fix illustration (source: [2]).

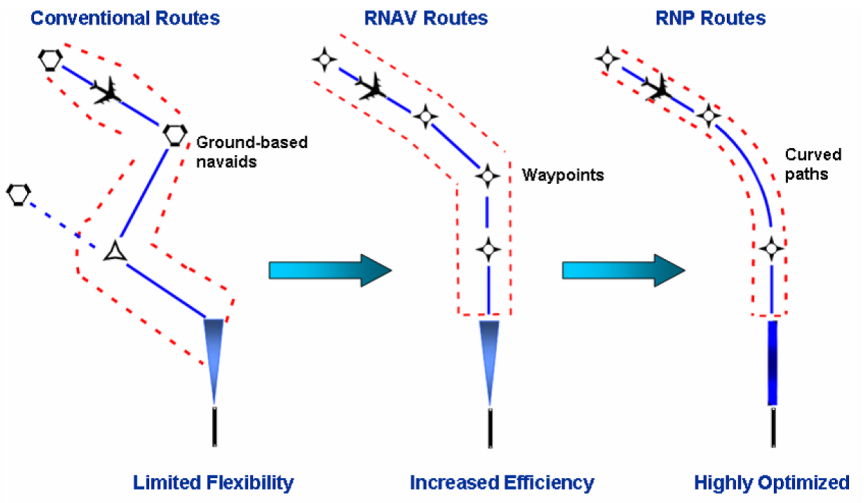


Figure 1.2: A comparison of conventional, RNAV and RNP routes.

A comparison among the conventional navigation, RNAV and RNP routes is illustrated in Fig. 1.2, where the conventional navigation routes follow ground-based navigation aids instruments, and the RNAV and RNP routes follow a succession of waypoints. Since the RNAV and RNP routes' structure is no more limited by the ground-based navigation aids, the available airspace is highly increased. From the operational point of view, flying a RNAV or RNP route is easy to operate for pilots. To define a flight path, pilots can choose which waypoints to follow in the navigation database implemented in the Flight Management System (FMS). During the flight, a Global Navigation Satellite System (GNSS) provides the aircraft position information to the FMS, and the FMS guides the aircraft along the flight plan.

Both RNAV and RNP have different performance levels, denoted by RNAV-x or RNP-x, where x represents the corresponding navigation accuracy. More precisely, RNAV-x represents that the horizontal distance between the aircraft and the standard path centerline is no more than x nautical mile (Nm) within 95% of the whole flight time. RNP-x is defined similarly, with the additional condition that the horizontal distance between the aircraft and the standard path centerline is no more than 2x nautical mile (Nm) within 99.999% of the whole flight time. Figure. 1.3 represents the navigation levels used in different flight phases, where the unit of the numbers is NM. A more detailed description of different flight phases is provided in Section 1.2. In general, higher navigation accuracy is provided to the flight phases with more intense traffic. Moreover, it can be seen that RNP generally provides higher navigation precision than the RNAV, thus RNP is more useful in complex airspaces with dense traffic. The route design problem considered in this thesis takes into account the RNP-1, that covers arrival, approach and departure flight phases.

NAVIGATION SPECIFICATION	FLIGHT PHASE							
	En-Route Oceanic Remote	En-Route Continent'l	APR	Approach				DEP
				Initial	Intermed	Final	Missed	
RNAV 10 (RNP 10)	10							
RNAV 5		5	5					
RNAV 2		2	2					2
RNAV 1		1	1	1	1		1	1
RNP 4	4							
RNP 2	2	2						
RNP 1			1	1	1		1	1
Advanced RNP	2	2or1	1	1	1	0.3	1	1
RNP APCH				1	1	0.3	1	
RNP AR APCH				1-0.1	1-0.1	0.3-0.1	1-0.1	
RNP 0.3		0.3	0.3	0.3	0.3	-	0.3	0.3

Figure 1.3: PBN applied in different flight phases (unit in NM)[1].

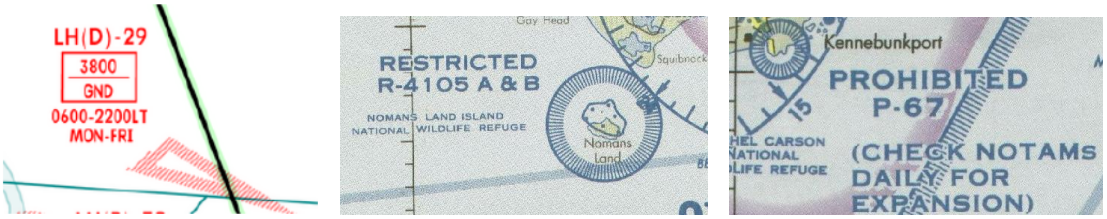
[1] source:http://www.avbuyer.com/articles/business-aviation-safety/are-you-ready-for-performance-based-navigation-part-2/

1.2 Controlled airspace and Air Traffic Service (ATS) route

The global airspace is divided into Flight Information Regions (FIRs) by the International Civil Aviation Organization (ICAO). It is currently the largest regular division of airspace in use. Flight information service and alerting service are provided by FIR. There is no standard size of a FIR, a large country can be divided into several FIRs, while several small countries may become components of a single FIR. Moreover, in some cases, a FIR is further divided vertically, where the lower portion remains the name as FIR, and the upper portion is named Upper Information Region (UIR). Airspaces are classified to different categories, from class A to class G (7 classes in total), according to specified service standards. More detailed information on the airspace classification can be found in [22].

Some parts of airspace may have specific restrictions and reservations for operating a flight. This kind of regions includes danger area, restricted area and prohibited area.

- A *danger area* (Fig. 1.4(a)), defined with the letter "D" in aeronautical charts, is a limited zone used for dangerous activities, such as parachuting and rocket-launching. It is the least restrictive one among these three kinds of areas, and pilots can decide whether or not to cross such an area.
- A *restricted area* (Fig. 1.4(b)), defined with the letter "R" in aeronautical charts, is a limited zone for developing activities like training flights and military training. The pilots can only cross such an area under the permission of controllers.
- A *prohibited area* (Fig. 1.4(c)), defined with the letter "P" in aeronautical charts, is a limited zone where flights are totally prohibited, except for some authorized military and government use. It is the most restrictive one among these three kinds of areas.



(a) (b) (c)

Figure 1.4: Airspace with specific restriction. (a) Danger area. (b) Restricted area. (c). Prohibited area.

Airspace can be distinguished as controlled airspace or uncontrolled airspace in a general way. A *controlled airspace* is established to provide ATC services to flights under Instrument Flight Rules (IFR) and Visual Flight Rules (VFR), so as to promote a safe, orderly and expeditious flow of air traffic. On the contrary, in an *uncontrolled airspace*, ATC does not exert any executive authority. The VFR is not very commonly used for commercial flights, thus it is not included in the scope of this thesis. There exist three mains types of controlled airspaces, taking charge of different phases of a flight.

The principal phases of an IFR flight are take-off, departure, en-route, arrival, approach and landing, as illustrated in Fig. 1.5. More precisely, *take-off* is the flight phase where aircraft go through a transition from moving along the ground (taxi-out) to flying in the air, usually starting on a runway and finishing after reaching about 400 feet above the runway. Then aircraft continue to

depart and leave the airport. When the Top of Climb (TOC) is reached, the *en-route* phase starts, aircraft may change flight levels in this phase. Afterwards, in the *arrival* phase, aircraft start to descend at the Top of Descent (TOD). The *approach* begins at the Initial Approach Fix (IAF) and terminates at the Final Approach Fix (FAF). A FAF is usually about 3000 feet above the runway. Finally, aircraft *land* to airport (finishing with taxi-in) and the flight is complete.

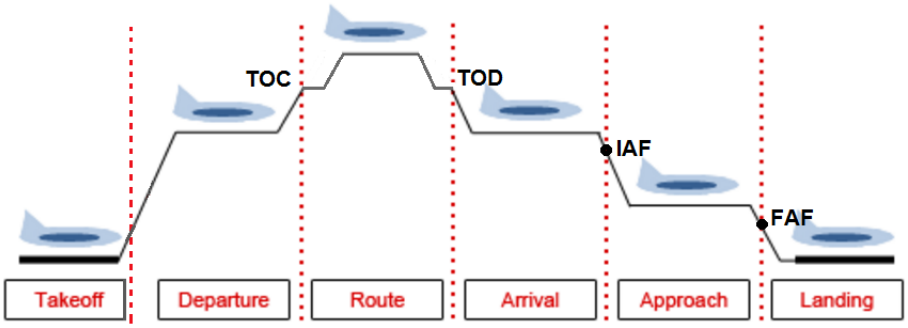


Figure 1.5: Principal phases of a flight.

The three mains types of controlled airspaces encountered by a flight are: Control zone (CTR), Terminal Maneuvering Area (TMA), Area Control Center (ACC). Figure. 1.6 illustrates the mentioned controlled airspace structure in the vertical plane.

- A *Control zone* (CTR) is the controlled airspace nearby an airport, which extends from the surface to a specified upper limit. It is established to control aircraft operating to and from the airport. Thus the taxi, take-off and landing flight phases are regulated by CTR.
- A *Terminal Maneuvering Area* (TMA), also called Terminal Control Area (TCA) or Terminal Radar Approach Control (TRACON) in the U.S., is an area surrounding one or more neighboring airports, designed to handle the departure, arrival and approach flight phases of aircraft.
- An *Area Control Center* (ACC), also referred to as an Air Route Traffic Control Center (ARTCC) in the U.S., is the controlled airspace extended from TMA to upper altitude, where en-route control service is provided.

Before an IFR flight takes place, a flight plan must be submitted to its corresponding ATFM unit. A flight plan contains especially the following information:

- aircraft identification number, type and navigation equipment;
- departure airport, proposed departure time, climb profiles;
- requested flight routes;
- cruising airspeed and requested cruising altitude;
- descent profiles and destination airport.

The corresponding ATFM unit analyzes the flight plan, and may propose some alternative routes in the case when the flight plan is not in compliance with the structure and capacity of the overflown airspace. One of the essential information in a flight plan is the requested flight routes, which are selected from the routes provided by *Air Traffic Service* (ATS). An *ATS route* is designed for

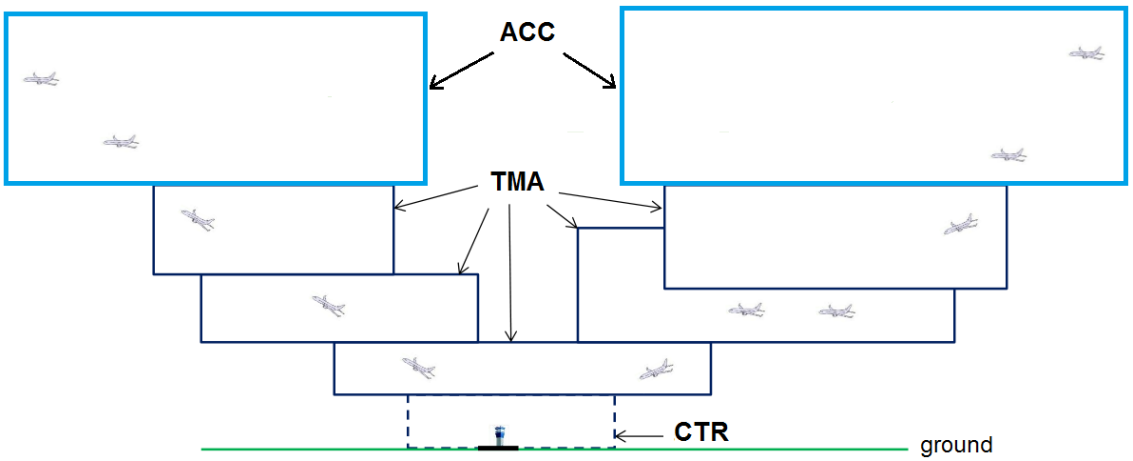


Figure 1.6: Controlled airspace structure.

channeling the flow of traffic as necessary for the management of air traffic. It may refer to various type of routes, such as low altitude airways (based on VOR[2] stations below FL180[3]), jet routes (based on VOR stations from FL180 to FL450 feet), RNAV routes, etc.. An example of ATS routes in aeronautical chart is presented in Fig. 1.7, the routes with letter "V" (respectively, "T") are low altitude airways (respectively, RNAV routes below FL180).

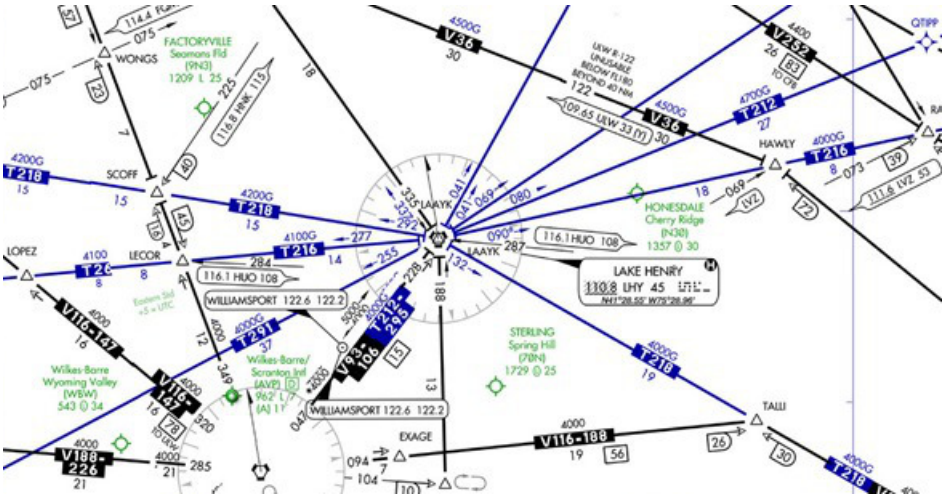


Figure 1.7: An example of ATS routes.

[2]VOR: VHF omni-directional range, a short-range radio navigation system for aircraft

[3]In aviation, 1 FL is equal to 100 feet, thus FL180 is equal to 18000 feet.

1.3 Departure and arrival routes in Terminal Maneuvering Area (TMA)

In this thesis, we specifically focus on the controlled airspace surrounding airports, that is the TMA, as well as the ATS routes for departure and arrival traffic. The scale of a TMA usually depends on the traffic amount to be handled, for example, the range of a TMA surrounding one or several major airports may extend to more than 80Nm from the corresponding TMA center, while the one of a TMA surround a small airport can be about 30Nm. Besides, different TMAs may have various shapes.

An example of the TMA of the Paris region in the horizontal plane is presented in Fig. 1.8, where the red arrows represent the departure traffic and the black arrows represent the arrival traffic. The case when more than one airports present in the same TMA is called metroplex. For example, in Fig. 1.8, both Roissy and Orly airports are in the same TMA. The metroplex TMA usually surrounds big cities with high traffic density. Traffic converging to different airports produces massive conflicts between aircraft. In order to decrease the interaction between traffic from or to different directions, the departure traffic (red arrows) and arrival traffic (black arrows) are usually located alternately, as shown in Fig. 1.8.

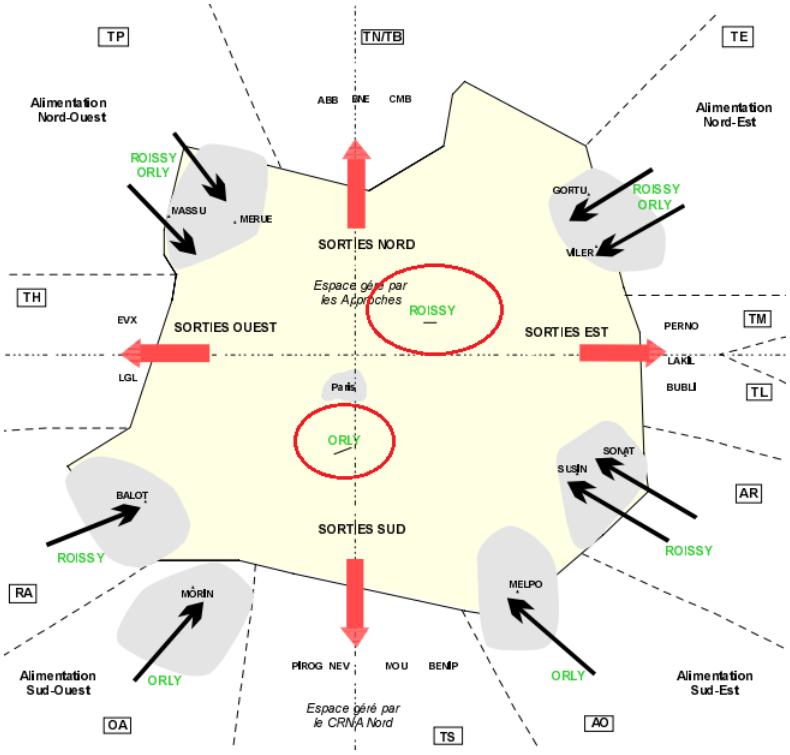


Figure 1.8: An example of the TMA of the Paris region.

Aircraft exit (respectively, enter) a TMA at some pre-defined points, named TMA exit (respectively, entry) points. These points are usually located at the boundary of the TMA. An example of the layout of TMA entry/exit points is presented in Fig. 1.9, where the TMA is in a circular shape. Moreover, the ATS route connecting the runway to a TMA exit point is called *Standard Instrument Departure* route (SID), as illustrated by the blue curve. Meanwhile, the route connecting a TMA entry point to the runway is composed by several parts. The route section connecting the TMA entry point to the IAF is called *Standard Terminal Arrival Route* (STAR). There is usually a holding pattern located at the IAF, used for holding or descending aircraft to a desired altitude. Afterwards,

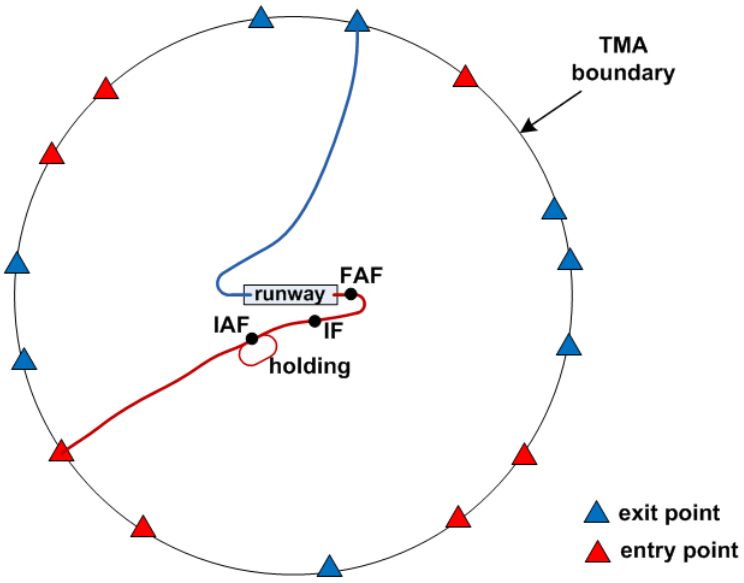


Figure 1.9: TMA entry/exit points.

the route section connecting IAF to FAF is called *arrival* route. Sometimes, an Intermediate Fix (IF) may exist between IAF and FAF. At the FAF, aircraft must be aligned to the runway, so as to intercept the ground-based infrastructure for the guidance of landing.

TMA is one of the most complex types of airspace, not only because of the highly dense traffic, but also because of the numerous constraints to be satisfied. The constraints generally fall into two main categories: operational constraints related to air traffic operations, and environmental constraints related to environmental sustainability. The operational constraints include:

- *obstacle avoidance*, the obstacle refers to mountains, restricted and military areas;
- *aircraft separation*, the standard separation norm between aircraft in a TMA is 3Nm in the horizontal plane or 1000ft in the vertical plane as shown in Fig. 1.10;
- *flyable routes*, the route must be smooth and with few turns;
- *runway alignment*, aircraft must head straightly to the runway before landing;
- *runway capacity limitation*, this may lead to aircraft holding in the air;
- *severe weather avoidance*.

The environmental constraints include:

- *noise abatement*, aircraft must fly over cities and neighborhoods above a vertical minimum;
- *CO_2 emission reduction*, continuous descent/climb operations consume less jet fuel, thus result in less CO_2 emission. However, the airspace capacity is reduced by using these operations.

The constraints on runway capacity limitation and severe weather avoidance are usually handled at tactical or pre-tactical levels, while the other constraints can be managed at strategic level. All these constraints make the operation of aircraft in TMA very hard to be regulated at the same time safely

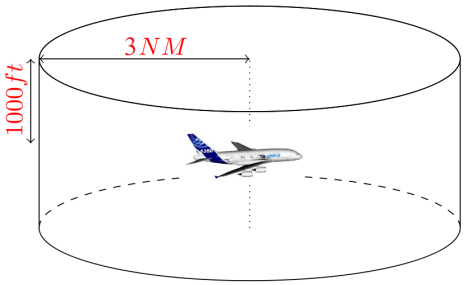


Figure 1.10: Standard separation norm in a TMA.

and efficiently, not only because the available airspace is restricted, but also because the controllers' workload to manage traffic is increased.

The ATS routes in TMA must be designed with respect to these constraints, especially the ones at strategic level. Airports usually have published aeronautical charts, showing the ATS routes for departing from and arriving to it. An aeronautical chart specifies the information such as the name of this chart, the navigation mode (conventional, RNAV, RNP), the corresponding runway in use, the names and coordinates of the waypoints to follow, the constraints on altitude and speed of aircraft, etc. An example of a STAR chart of New Zealand Queenstown airport is presented in Fig. 1.11. The environment of the TMA surrounding Queenstown airport is very complex, due to the presence of mountains (area in brown color) and river (area in blue color). Thus RNP is used in such context, because of its high precision and flexibility, and the route is composed by segments and arcs of circles. In the real-world operation, before a flight takes place, pilot must choose the SID and STAR to follow, and enter the related waypoints to the Flight Management System (FMS) in the cockpit, so that the aircraft will be guided during the flight. Moreover, the use of SID/STAR charts simplifies the cooperation between pilots and controllers, by indicating directly the names of waypoints or routes.

The difficulty of designing SIDs/STARs lies especially in dealing with the numerous operational and environmental constraints mentioned above. Currently, SIDs/STARs are designed manually according to the requirements defined by the International Civil Aviation Organization (ICAO) [23], taking into account airport layout and nearby environment. This kind of design is generally not very efficient and is not expected to optimize the use of available airspace. The software GéoTITAN, developed by Ecole Nationale de l'Aviation Civile (ENAC) PANS-OPS unit can be applied for procedure design in compliance with [23]. Even though this tool is efficient and accurate, it is still not expected to optimize any specific criterion.

1.4 Optimization problems and methods

Many complex problems in the real world are solved through the mathematical optimization, which is an important tool in making decisions and analyzing complex systems. An optimization problem is built up through the process of *modeling*, and three basic ingredients are identified and expressed in mathematical terms in this process. The basic ingredients of an optimization problem are:

- the *objective function* is used to measure quantitatively the performance of the system, and its value is minimized or maximized;

[4]source:`http://www.aip.net.nz/pdf/NZQN_45.1_45.2.pdf`

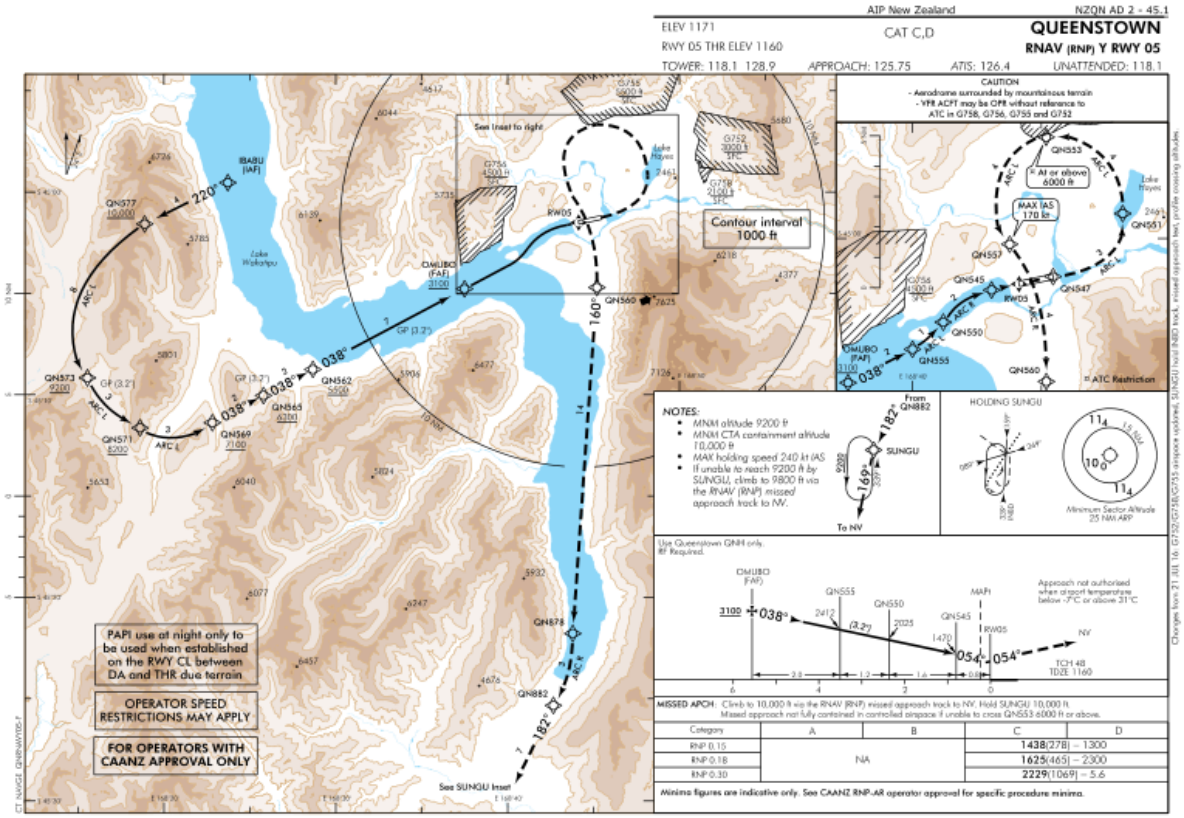


Figure 1.11: A STAR chart of New Zealand Queenstown airport[4].

- the *decision variables* are the components of the system, and serve as levers to act on the system. Their values are to be determined in the resolution process. The objective function is expressed as a function of the variables;
- the *constraints* give the relationship among the variables, and allow the variables to take certain values while exclude others.

One standard form of an optimization problem is:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \cdots, m \\ & h_i(\boldsymbol{x}) = 0, \quad i = 1, \cdots, p \end{aligned} \tag{1.1}$$

where

- $\boldsymbol{x}$ represents the decision variables;
- f is the objective function;
- $g_i, i = 1, \cdots, m$ are the inequality constraints, and
- $h_i, i = 1, \cdots, p$ are the equality constraints.

In the case of a maximization problem, the objective function can be negated in order to be in compliance with this standard form.

It is important to categorize the optimization model, since algorithms for solving optimization problems are tailored particularly to a type of problem. According to whether the variables are continuous or discrete, an optimization model can be classified into three categories:

- *Continuous Optimization*, where each variable can take value from a set of continuous real values;
- *Combinatorial Optimization*, where each variable can only take value from a discrete set, usually a subset of integers;
- *Mixed-Integer Programming*, where some of the variables are constrained to be integer values and the others are continuous.

Optimization methods are used to provide the values of the decision variables, which lead to the highest level of system performance. The optimization methods can be roughly divided into two categories: exact approaches and heuristics approaches.

- An *exact approach* guarantees to provide an optimum solution of the problem, if such a solution exists, but it may be time-consuming to find the optimal solution of a difficult problem. For this reason, exact approaches are usually difficult to be applied in some highly combinatorial NP-hard problems. In the path or trajectory planning problems, the exact approaches are generally applied in the design of one path or trajectory in 2D.
- Compared with an exact approach, an *heuristic approach* seeks to produce good-quality but not necessarily optimal solutions in reasonable computation times. Thus this kind of approaches are more commonly used for complex and highly combinatorial problems. In the case of designing multiple 3D paths or trajectories, especially when taking into account numerous constraints, the heuristic methods seem to be more appropriate.

Exact approaches

Among the exact methods used in the context of path and trajectory planning, we find naturally the shortest path algorithm (*Dijkstra's algorithm* [24], *Bellman-Ford algorithm* [25]). Besides, a widely known exact approach for solving combinatorial optimization problems is the *Branch and Bound* (B&B) method. This method is applied in the route design problem considered in this thesis, thus we present it in more detail in the following.

The B&B method is proposed by A. H. Land and A. G. Doig in 1960 [26]. Readers may refer to [27] and [28] for a survey and some examples of this method. The B&B algorithm checks the complete search space to get the global best solution. The number of potential solutions increases exponentially during the search, therefore explicit enumeration is hardly to be conducted. Instead, the implicit enumeration consisting of branching and bounding is applied to deal with the large number of potential solutions. The *branching* refers to the strategies of dividing the problem into subsets by partitioning the search space. The *bounding* refers to computing a lower bound (in a minimization problem) of a relaxed form of a subset, which is used to decide whether the corresponding subset needs to be further branched on. More precisely, in the case when the lower bound of a subproblem is greater than the cost value of the current best solution found so far, it is sure that there exists no better solution than the current best one with further branching on this branch.

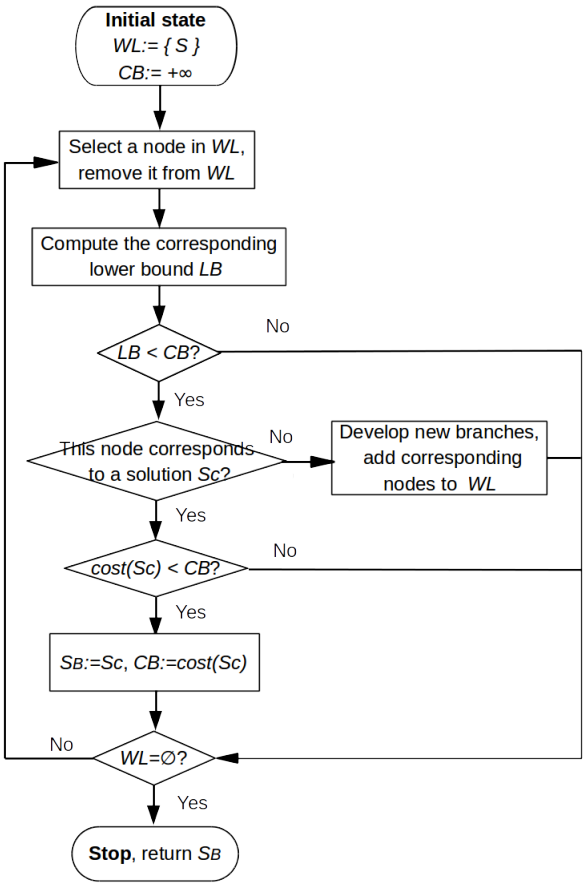


Figure 1.12: B&B flow chart.

In a B&B method, the subsets generated are represented as nodes in a search tree. A flow chart of a general B&B method is presented in Fig. 1.12, where the related notations are given in Table 1.1.

WL: the waiting list storing unconsidered nodes in the search tree
S: the complete search space
CB: the cost value of the current best solution
LB: the lower bound associated with the considered node
S_C: the current solution
$cost(S_C)$: the cost of the S_C
S_B: the current best solution

Table 1.1: Notations related to the B&B method.

At the initial state, the complete search space S is denoted as the root node of the search tree and added to WL, while the value of CB is set as infinity. Afterwards, the B&B algorithm deals with the nodes in the search tree iteratively. For each selected node, the corresponding LB is computed. If LB is greater than CB, which means the subset does not contain a better solution, then the whole subset is discarded. Otherwise, the algorithm continues to check whether the node corresponds

to a feasible solution S_C. If it is the case, the cost of S_C is computed and is compared with the current best value CB. This solution is accepted as S_B when it has a lower cost value. If this node does not correspond to a feasible solution, two or more new branches are developed based on this node and added to WL, by subdividing the corresponding search space into two or more subsets. The algorithm terminates when all nodes are considered, and the solution is the best solution found through the whole process.

The computing time for obtaining the global optimal solution in the B&B method depends not only on the complexity of the problem but also on the branching strategy, which defines the order of tree exploration. Different branching strategies have been investigated, such as the depth-first search [29], which seeks to process the most recent nodes in the search tree, and the best-bound search [30], which chooses to branch on the subset with the smallest lower bound. A good branching strategy enumerates fewer tree branches before obtaining the global optimum.

Heuristic approaches

A commonly used heuristic method searching for the shortest path is the *A* algorithm* [31]. It is applied in several works that we will cite in Chapter 2. For a better understanding of its applications in literature, we present in more detail this method in the following. This method is formulated in terms of weighted graphs. The principle of the A* algorithm is to search for the best path (with least travel distance, shortest time, etc.) among a tree of possible partial paths, and each time it restarts from the one that may lead most quickly to the best path. The root of the search tree is the starting point of the path to design. At each step, the current best partial path is extended one step by connecting its end node to the neighboring nodes. Several new partial paths are then obtained and are added to the search tree. The algorithm terminates until one path reaches the ending point of the path to design. The cost function to evaluate a partial path with ending point x is represented by $f(x) = g(x) + h(x)$, where $g(x)$ is the cost of getting from the starting point to node x, and $h(x)$ is a heuristic estimation of the cost to get from node x to the ending point. In order to find the actual shortest path, the heuristic cost $h(x)$ must be a lower bound on the actual cost to get to the ending node. Some applications of A* algorithm are presented in Chapter 2.

To illustrate how the A* algorithm works, an example is given in Fig. 1.13. In the weighted graph, a node represents a city, an edge shows the connectivity between two cities, and the weight associated with an edge denotes the distance between two connected cities. The problem of finding the shortest path from city S to city E is considered. To apply the A* algorithm, the cost $g(x)$ at node x is computed by summing up the lengths of the passed edges, and the heuristic cost $h(x)$ is the straight line distance between node x and ending node E. In Fig. 1.13, the heuristic cost at each node is shown in the brackets. At initial state, the only partial path in the search tree is the starting node S. In the first iteration, 3 partial paths are generated and added to the search tree by connecting S to A, B, C respectively. The partial path $(S-B)$ has the minimum cost, thus is extended in the second iteration. Two partial paths are generated based on partial path $(S-B)$ by connecting B to D and G respectively, and they are added to the search tree. To this step, there are four possible partial paths in the search tree, and the one with the minimum cost is $(S-B-G)$, which will be extended in the next iteration. The third and fourth iterations are shown in Fig. 1.13. The optimal path connecting S to E is $(S-B-G-E)$ with cost 28.

Within the scope of heuristics methods, some of them are high-level problem-independent algorithmic frameworks, which do not take advantage of any specificity of the problem. These methods provide guidelines or strategies to solve the complex optimization problems. This kind of approaches are called *meta-heuristic*[32], including for example Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO) [33], Ant Colony Optimization (ACO) [34]

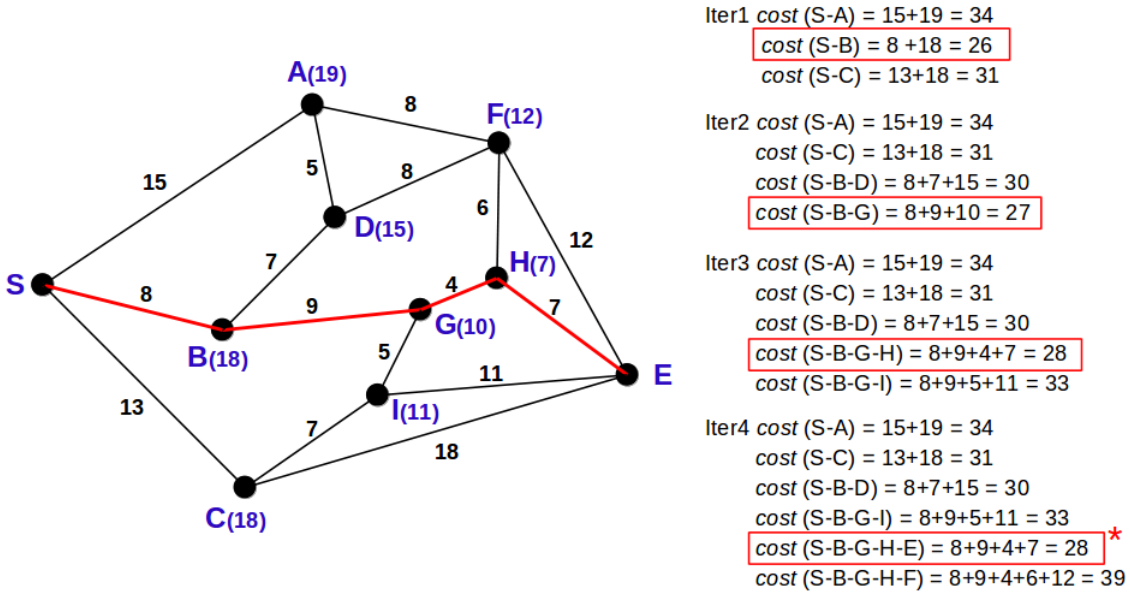


Figure 1.13: An example of the A* algorithm. The value in the brackets after each node denotes the heuristic cost of that node.

etc..

Genetic Algorithm (GA) is one of the most famous meta-heuristic approaches, based on the natural evolution of genetic selection. For an optimization problem solved by GA, each candidate solution is called an *individual*, and each individual consists of a set of properties, named *chromosomes*. At the beginning of a GA process, a *generation* of individuals is randomly generated. Then, iteratively at each generation, the individuals are evaluated according to a *fitness* function. The individuals with best fitness values are selected, and evolved through *crossover* and *mutation* operators. The crossover operator aims at producing a better child individual by mixing two good parent individuals of the previous generation, and the mutation operator is used to maintain genetic diversity from the previous generation to the next. Afterwards, a new generation is created by selecting the best evolved individuals according to the fitness function. Readers may refer to [35], [36] and [37] for more details of the GA. Some applications of GA in the context of path and trajectory planning are presented in Chapter 2.

In this thesis, we apply another well-known meta-heuristic method, that is the *Simulated Annealing* (SA), proposed by S. Kirkpatrick et al. in 1983 [38] and by V. Cerny in 1985 [39]. The principle of SA is to emulate the physical process whereby a solid is slowly cooled so that when eventually its structure is "frozen", a minimum energy configuration is obtained. A flow chart of SA algorithm is presented in Fig. 1.14, where the parameters and related notations are presented in Table 1.2.

In the SA algorithm, the parameter T controls the temperature schedule, and it decreases as the total number of iterations increases. There exist different temperature decreasing laws in real-world applications. In the flow chart Fig. 1.14, the temperature decreases by following a geometrical law ($T := \beta T$). Another parameter N_I controls the number of iterations to be carried out at each temperature stage. At each iteration, a neighboring solution is generated based on the current state of the system. The way of generating a neighboring solution is usually problem-dependent. In addition, this step is fundamental to the SA algorithm, since the final solution will come after a succession of neighboring solutions. Afterwards, the neighboring solution is evaluated through a

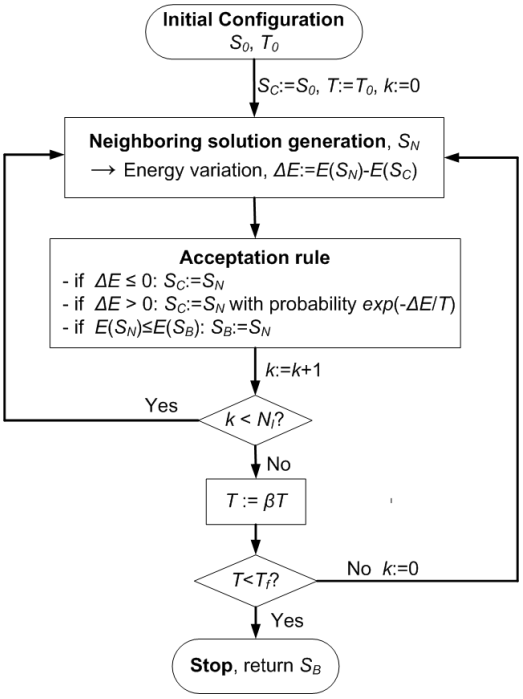


Figure 1.14: Simulated Annealing flow chart.

T_0, T, T_f: the initial, current, final temperatures respectively
S_0, S_C, S_N, S_B: the initial, current, neighboring, best solutions respectively
β: the temperature cool-down factor ($\beta < 1$)
E: the SA fitness function
ΔE: the degradation of the SA fitness function value
k: the counter of iterations
N_I: the criterion for change of the temperature stage

Table 1.2: Parameters of the SA method.

fitness function. According to the acceptation rule shown in Fig. 1.14, at high temperature, the probability $\exp(\frac{-\Delta E}{T})$ is close to 1, thus the system probably accepts a worse solution; while at low temperature, $\exp(\frac{-\Delta E}{T})$ is close to 0, thus the system tends to accept only better solution. This mechanism is an advantage of SA method, since it helps the system to escape from a local minimum. The convergence speed of the SA method depends not only on the configuration space but also on the following user-defined control parameters: T_0, β, N_I and T_f, whose values are usually set by empirical adjustments. Readers can refer to [32] for practical suggestions for choosing these parameters.

1.5 Objectives and contributions of this thesis

The objective of this thesis is to propose a decision support methodology of designing automatically and optimally SIDs/STARs in a TMA surrounding only one airport. In fact, by designing SIDs/STARs in a reasonable and effective way, the airspace can be sufficiently used, and the work-

load of controllers to supervise and manage aircraft can be decreased. As a result, the capacity of a TMA is increased and more traffic can be absorbed. Currently, SID/STAR are designed manually according to operational requirements [23], taking into account airport layout and nearby constraints. However, this kind of design is generally not very efficient and not expected to optimize any specific criterion. The methodology proposed in this thesis deals with the design of SIDs/STARs in an optimal way, with respect to the total length of the designed routes, while taking into account some strategic level constraints including obstacle avoidance and separation between routes as main constraints. The severe weather, normally managed at pre-tactical or tactical level, is not included in the scope of this thesis. Our designed routes are compatible with the RNP concept, taking advantage of its high accuracy and flexibility.

Designing SIDs/STARs is a hard problem in the real-world application that has to take into account many factors, including:

- airport configuration: runways in use, runway directions, ground-based infrastructures, etc;
- surrounding environment: mountains, cities, restricted airspace, etc;
- aircraft performance: types, mass, take-off/landing slopes, curvature of a turn, etc;
- operational requirements: obstacle avoidance, routes separation, etc;
- environmental requirements: noise abatement, CO_2 emission, air pollution, etc.

Optimization can be an effective means to deal with such a complex problem. However, representing this problem with mathematical terms is not evident, and it is especially difficult to describe explicitly the impacts of the mentioned aspects on the design problem.

The first main contribution of this thesis is to propose a modeling of the concerned problem in the mathematical framework. An obstacle, as well as its protection area, is modeled as a cylinder in 3D, whose projection to the horizontal plane is a disk. Each route to design is modeled in 3D, consisting of two components: a curve in the horizontal plane associated with a cone in the vertical plane.

- In the horizontal plane, the curve is composed by a succession of straight line segments and arcs of circles. The form of the horizontal curve is flyable under the RNP concept, since each segment corresponds to a standard point-to-point leg in RNP, and each arc corresponds to a RF leg in RNP. Moreover, modeling the horizontal curve as described make use of the high flexibility and precision of RNP, so that the airspace can be better used.
- In the vertical plane, instead of simply taking a straight line segment representing the vertical profile of aircraft, we model the vertical profile in shape of a cone. The cone consists of two straight lines determined by the maximum and minimum take-off (in a SID case) or landing (in a STAR case) slopes of the aircraft following this route. The reason of modeling in such a way is that, aircraft following the same route may have various performance, thus their take-off and landing slopes are not uniform. This fact makes their vertical profile diverge from the runway. Modeling the vertical profile as we described is more conform with the reality.

The form of a horizontal curve (a succession of straight line segments and arcs of circles) is in compliance with the shape of obstacles, since an arc can be used to contour an obstacle and a segment can be applied to connect two arcs. To deal with the obstacle avoidance, taking into account the shape of routes and obstacles, three maneuvers of route deviation are proposed: bypassing an

obstacle clockwise, bypassing an obstacle counter-clockwise and imposing a level flight below the obstacle. The two bypassing maneuvers work on the deviation of the horizontal curve, using an arc that contours the considered obstacle. While the maneuver of imposing level flight acts on the vertical cone by maintaining the altitude at a certain level for a certain length. Besides their adaptation to the form of routes and obstacles, these obstacle avoidance strategies also correspond to real-world operations. Moreover, two groups of decision variables associated with the obstacle avoidance strategies are proposed. The first group defines whether an obstacle is active or not with respect to a route. The second group defines in which way the route avoids the active obstacle (bypassing clockwise, bypassing counter-clockwise or imposing level flight). Then the constraints and the objective function are expressed in terms of the decision variables either explicitly or implicitly. The routes design problem is consequently formulated as a combinatorial optimization problem.

The second contribution of this thesis is that a two-step solution approach tailored to this problem is developed. The problem of designing multiple 3D routes is hard, especially in consideration of many constraints. Thus, we first deal with the design of one single route with the minimum length, while taking the obstacle avoidance as the main constraints. The Branch and Bound (B&B) method is applied to solve this combinatorial optimization problem. A branching strategy specifically based on the obstacle avoidance maneuvers (bypassing counter-clockwise, bypassing clockwise, imposing level flight) is proposed. In the second step, the design of multiple routes is addressed, and the constraint of route separation is considered additionally. We define a *conflict* as a loss of minimum separation between routes. Two different solution approaches are developed. The first one is a B&B-based method, which builds routes sequentially according to a given order (*e.g.* the decreasing order of their traffic loads). This approach is characterized by the B&B to generate a single route, and a strategy to deviate a conflicting route. The route deviation is again performed by contouring or imposing level flights below the conflicting areas. However, the quality of the solution obtained by this B&B-based approach depends a lot on the order of routes generation, due to the fact that the route generated at first has more free space, thus tends to be straight, while the routes built at last are usually composed by more deviations and level flights, so as to avoid obstacles and existing routes. Therefore, another method, the Simulated Annealing (SA), is applied to design multiple routes, where routes are generated simultaneously. The length of route sections involved in conflicts is integrated in the cost function. In the SA algorithm, the initial solution is obtained by applying the B&B on each individual route. Afterwards, a neighboring solution is generated iteratively, based mainly on the choice of conflicting area avoidance strategies. In contrast to the B&B-based method, the SA chooses the obstacle and conflict avoidance strategies independently for each route, without knowing the form of other routes.

Finally, our proposed approach is tested on a set of artificially generated problems as well as on two existing TMAs (Paris CDG and Zurich). In the artificially generated problems, we test various configurations of TMA (number and layout of obstacles, runways, positions of the TMA entry/exit points), both B&B-based and SA methods are efficient and provide good quality results. The choice of the TMAs of Paris CDG airport and Zurich airport allows us to confront our approach, on the one hand, on a TMA with numerous TMA entry/exit points, and on the other hand, on a TMA with many obstacles. The simulation results of both methods show a gain in the total routes length compared with the published standard charts. This can be promising in terms of reducing jet fuel consumption. Furthermore, tests on the TMA of Zurich show that both approaches can be applied effectively in a TMA in the presence of many obstacles.

Chapter 2

Literature Review

The path and trajectory planning problems have been studied since long time. In this chapter, we present a literature review on the existing works related to path and trajectory planning problems. We first explain the definitions and differences between path planning and trajectory planning:

- A *path planning* problem aims at searching for a continuous curve that connects given starting and ending points while avoiding static obstacles. A path planning problem does not concern any individual mobile following the route, therefore the speed as well as the control laws of the mobile are not considered. As a result, the notion of time is not associated. The "path planning" is also referred to as "route planning" or "route design", while the latter ones are more commonly used in the aeronautical domain.

- A *trajectory planning* problem is not only to seek the curve that connects the starting and ending points, but also to associate a control law of the mobile, so that the speed with which the mobile follows the curve is determined. Thus the notion of time is associated and dynamic obstacles have to be considered additionally.

The problem considered in this thesis is in the scope of path planning. In this chapter, we first introduce some typical path planning methods, especially in the robotic domain, as well as their applications. Next, the route design problems in the aeronautical domain are presented; modeling and resolution approaches are discussed. Finally, the related trajectory planning problems, particularly in the ATM domain, are visited.

2.1 Path planning methods

Some typical path planning methods for a mobile robot are introduced in [40] [41] and [42]. A majority of path planning methods in the robotic domain concentrate only on 2D design, where the mobiles are supposed to move in a horizontal plane. The problem of designing one optimal path is usually considered in two steps. The first step is to model the environment or the shape of the path. In the second step, the optimal path with respect to a certain criterion (for example, the minimum path length) is generated by applying an optimization method which is adapted to the modeling. In this section, some typical methods for modeling the environment and the shape of the path are presented.

2.1.1 Roadmap based methods

The first group of methods are applied to build up a *roadmap* in the search space, representing a set of paths between given staring and ending points that the robot can travel without collision. In other words, the roadmap methods aim at capturing the free-space connectivity with a graph. These methods are usually combined with a graph-based algorithm to search for the shortest path, such as the *Dijkstra's algorithm*, the *Bellman-Ford algorithm* and the *A* algorithm*. In the case when k shortest paths connecting a pair of starting and ending points are to be designed, some approaches are proposed in [43][44].

Visibility Graph

The *Visibility Graph* approach is introduced by Lozano-Pérez and Wesley in 1979 [45]. The Visibility Graph is composed of the nodes which are the vertices of the polygonal obstacle, and the edges which are the line segments connecting pairs of nodes lying in the collision-free space. Each edge is associated with a weight, usually equal to the length of this edge. It is proven that the 2D shortest path avoiding polygonal obstacles is the shortest path on the corresponding Visibility Graph [45]. Different methods searching for the shortest path, based on the Visibility Graph, are explained in [45] [46] [47]. An example of Visibility Graph with four triangle shape obstacles is presented in Fig. 2.1, the starting point P_{start}, ending point P_{goal}, the vertices and edges, as well as the shortest path are illustrated. Some works contribute to the extension of the 2D visibility graph to 3D with polyhedral obstacles [48] [49] [50]. The problem becomes NP-hard in 3D, since the shortest path does not necessarily pass through the vertices of polyhedrons [51].

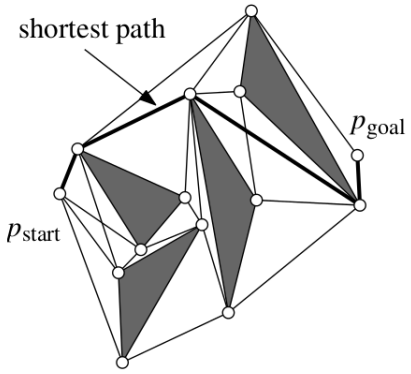


Figure 2.1: An example of the Visibility Graph in 2D.

In the case when both polygonal and curved obstacles are present, the notion of *Tangent Graph* was introduced by Liu and Arimoto [52] in 1992. For a Tangent Graph, a node is a tangent point lying on the boundary of an obstacle. An edge corresponds either to a collision-free line segment which is tangent to a pair of obstacles or to a convex boundary section between two nodes of an obstacle. A Tangent Graph allows one to take into account both polygonal and curved obstacles [52], as well as obstacle in the form of circular discs, such as in [53] [3]. In these graph-based approaches, the edges are built in collision-free area, thus the constraint on obstacle avoidance is automatically satisfied.

The computing time for finding the shortest path in a graph depends strongly on the number of edges, which is then linked to the number of obstacles. In [3], a problem for computing the shortest path among a set of mutually disjoint disks is studied. The authors propose some techniques in

order to eliminate the circles which definitely have no impact on the form of the shortest path. They first prove that the shortest path lies in the ellipse with the starting and end points as the foci. Then they prove additionally that the shortest path lies in a convex hull of a few circular obstacles around the line segment connecting the starting and ending points. With the ellipse and convex hull filters, the size of the state space is reduced significantly.

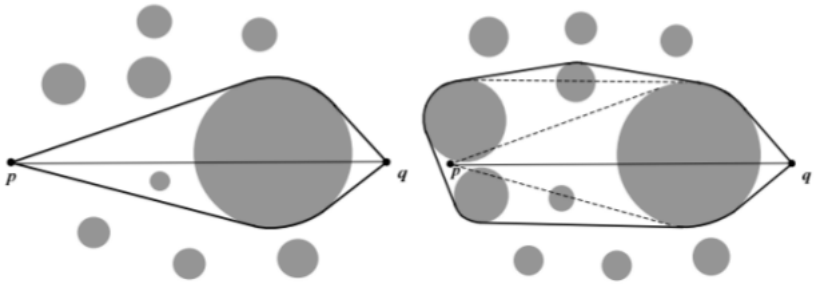


Figure 2.2: An illustration of the convex hull filter in [3].

Voronoi Diagram

In the case when obstacles are in polygonal shape, as shown in Fig. 2.3, the *Voronoi Diagram* is built by drawing line segments and curves that have equal distance to the edges of the polygonal obstacles. These line segments and curves become edges in the Voronoi Diagram and their intersections become the nodes. This approach is usually used to reduce the chance of collision, since the robot is kept as far away from the obstacles as possible. Thus the obtained path is not necessarily to be the shortest one. A bibliography of this method is given in [54].

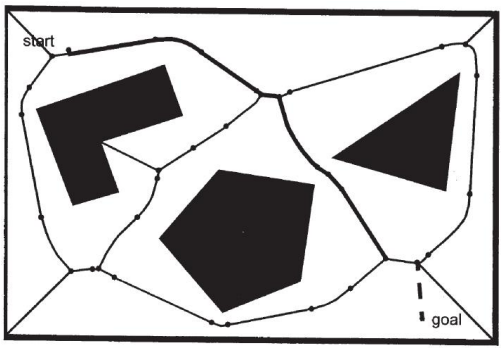


Figure 2.3: Voronoi diagram in a 2D polygonal environment[1].

Cell Decomposition

The principle of *Cell Decomposition* is to first subdivide the free space of the given environment into small regions, called cells. An example of environment subdivision is shown in Fig. 2.4(a), where parallel line segments are drawn as the boundaries of the cells. Each line segment passes through a vertex of a polygonal obstacle, or uses such a vertex as an ending point. Other techniques

[1] source:`https://ayorho.wordpress.com/2011/01/03/ltc2-ans/`

of environment subdivision can be applied [55] [56]. Afterwards, a connectivity graph is built up based on this decomposition, where the adjacency relationships between the cells are identified, as shown in Fig. 2.4(b). The nodes in the connectivity graph denote the cells in the free space, and the links show the adjacency between nodes. A channel in the connectivity graph connecting the two nodes that contain the starting and ending points can be determined by simply linking adjacent nodes. Finally, this channel corresponds to a conflict-free path in the environment with polygonal obstacles, as shown in Fig. 2.4(c). More details for the application of Cell Decomposition in path planning problems are provided in [57] [58].

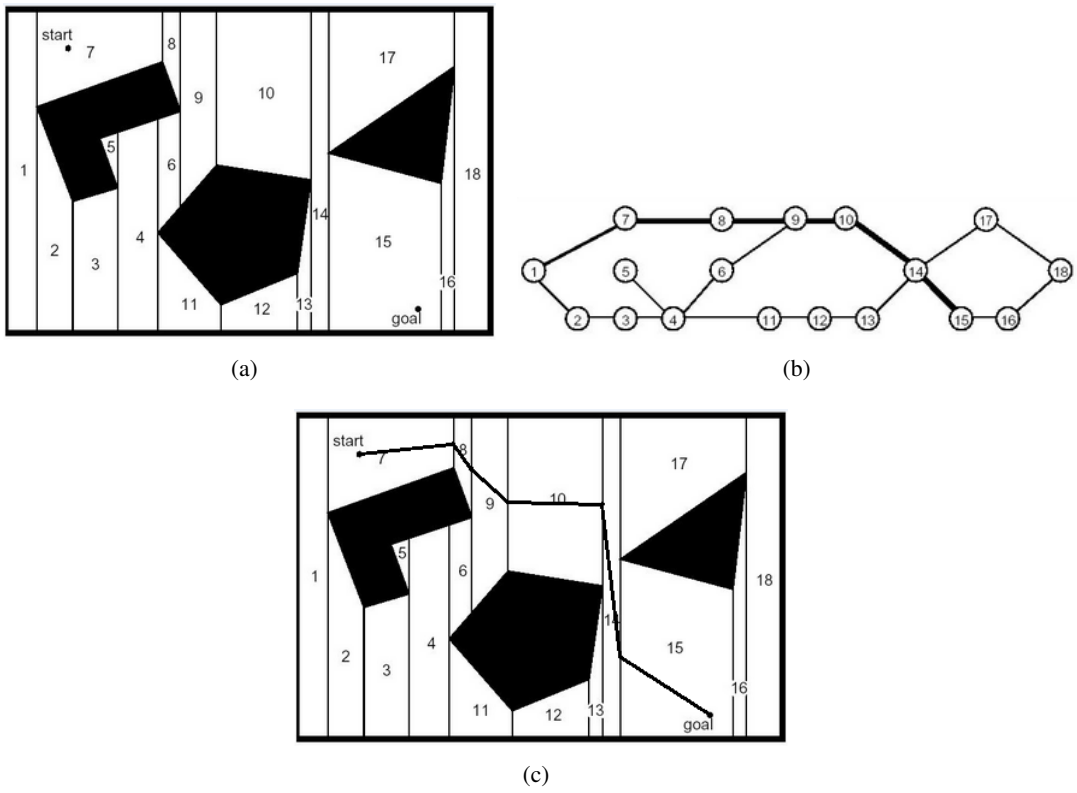


Figure 2.4: Cell Decomposition in a 2D polygonal environment. (a) Environment subdivision into small regions. (b) The corresponding connectivity graph. (c) A conflict-free path.

Probabilistic Roadmap

The idea of *Probabilistic Roadmap* (PRM) [59] is to take random samples in the environment where the mobile moves. The samplings located in the free space are kept and serve as nodes in the graph built by PRM, while the others are discarded. Afterwards, a remaining sampling is connected to its neighboring sampling in a certain way defined by a local planner. The collision-free links between pairwise samplings are retained as edges in the graph built by PRM. The starting and ending nodes are also included in the graph. Finally, an algorithm searching for the shortest path can be applied.

Rapidly-exploring Random Tree

The *Rapidly-exploring Random Tree* (RRT) method [60] is inspired by the PRM, and it is designed for efficiently searching non-convex high-dimensional spaces. The specific point of RRT is that it does not require any upstream environment modeling. Given the starting and ending points, a RRT tree grows progressively from the starting point towards the ending point. Iteratively in the RRT process, a sampling is randomly taken in the search space, then a connection is drawn between this sampling and the its nearest node in the tree. If this connection lies in the free space and satisfies other motion constraints, then it is added to in the RRT, the end points of this connection are the nodes, and the link between the end points becomes an edge. The RRT terminates when the destination point is reached by a connection.

2.1.2 Path shape based methods

The second group of methods represents the path to be designed as a combination of some basis analytical functions, and the shape of the path is controlled by a vector of parameters. In such a way, the dimension of the state space is reduced. Through an optimization process to search for the optimal parameters, the optimal path avoiding obstacles is obtained correspondingly. Some basis analytical functions to model the shape of a path are summarized in [40] and [61].

Lagrange interpolating polynomial

The formula of *Lagrange interpolating polynomials* [62] was first published by Waring in 1779, and then rediscovered by Euler in 1783. It is named after Lagrange, who published it again in 1795. For a given set of $n+1$ points $\{(x_i, y_i) |\ i = 0, \cdots, n\}$, where the x-coordinates are pairwise distinct, the Lagrange interpolating polynomial is the polynomial $L(x)$ of degree n that passes through the $n+1$ given points. The polynomial $L(x)$ is presented by

$$L(x) = \sum_{i=0}^{n} y_i l_i(x) \tag{2.1}$$

where each basis polynomial $l_i(x)$ is presented by

$$l_i(x) = \prod_{\substack{0 \le j \le n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \tag{2.2}$$

Each basis polynomial $l_i(x)$ passes through its respective control point (x_i, y_i) and is 0 when x-coordinate corresponds to the other control points. Figure 2.5 illustrates an example of Lagrange interpolation, where 4 control points are given, and the polynomial $L(x)$ (black dashed line) passes through them. The drawback of Lagrange interpolation is that the derivatives at the control points are not interpolated.

Piecewise polynomial interpolation

Another scheme to obtain an interpolation of a given set of points $\{(x_i, y_i) |\ i = 0, \cdots, n\}$, where $x_0 < x_1 < \cdots < x_n$, is by using *piecewise polynomial interpolation* on each subinterval $[x_i, x_{i+1}], i =$

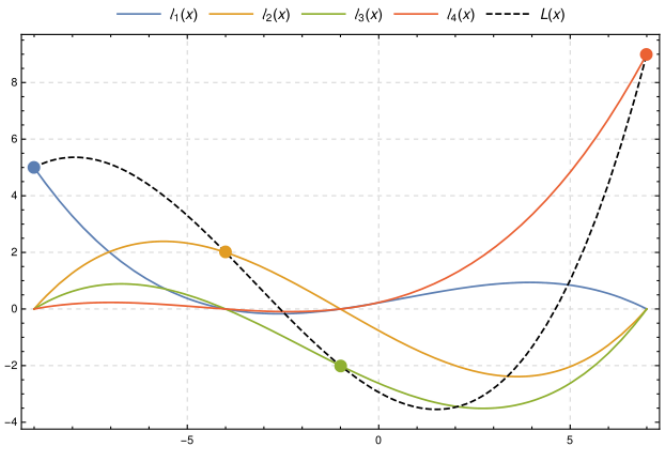


Figure 2.5: An example of Lagrange interpolating polynomial.

$0, \cdots, n-1$. The interpolating function can be presented by

$$S(x) = \begin{cases} S_0(x), & x \in [x_0, x_1] \\ S_1(x), & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x), & x \in [x_{n-1}, x_n] \end{cases} \tag{2.3}$$

Piecewise linear interpolation

The simplest case of piecewise polynomial interpolation is the *piecewise linear interpolation*, where each basis polynomial $S_i(x), i = 0, \cdots, n-1$ is a linear polynomial, defined as a straight line segment connecting points x_i and x_{i+1}. It can be expressed by

$$S_i(x) = a_i + b_i(x - x_i), \; x \in [x_i, x_{i+1}] \tag{2.4}$$

with the coefficients $a_i = y_i$ and $b_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$. An example of piecewise linear interpolation is presented in Fig. 2.6, where the black points are the given points and the red lines are the piecewise linear polynomials. The drawback of this method is that the derivative of the obtained path is not continuous.

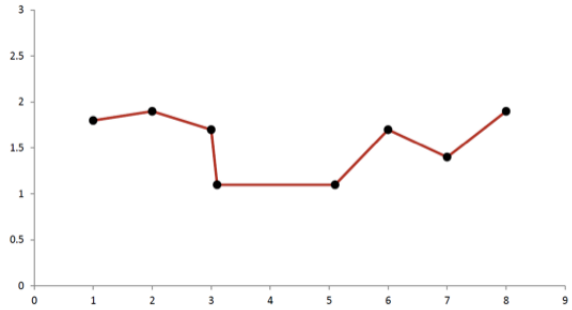


Figure 2.6: An example of piecewise linear interpolation.

Piecewise cubic Hermite interpolation

In the case when the first derivatives of the given points are also interpolated, the *piecewise cubic Hermite interpolation* can be applied. For a subinterval $[x_i, x_{i+1}], i = 0, \cdots, n-1$, the corresponding basis polynomial $S_i(x)$ is a cubic polynomial, expressed by

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \tag{2.5}$$

where the coefficients a_i, b_i, c_i and d_i can be computed through the following equations, representing the boundary conditions of $S_i(x)$

$$\begin{cases} y_i & = & S_i(x_i) \\ y_{i+1} & = & S_i(x_{i+1}) \\ y'_i & = & S'_i(x_i) \\ y'_{i+1} & = & S'_i(x_{i+1}) \end{cases} \tag{2.6}$$

Bézier curves

In the case when the shape of the designed path is controlled by a set of given points, but not obliged to pass through the given points, then the *Bézier curve* [63] can be applied. Let $\mathbf{P_i}, i = 0, \cdots, n$ be $n+1$ control points, the corresponding Bézier curve is of degree n, its analytical function is presented by

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P_i}, \ 0 \leq t \leq 1 \tag{2.7}$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$. The first control point $\mathbf{P_0}$ and the last control point $\mathbf{P_n}$ are always the starting and ending points of the Bézier curve, while the intermediate points only provide directional information and generally lie out of the curve. An example of Bézier curve with 4 control points is presented in Fig. 2.7. One of the drawbacks of the Bézier curves is that the degree of the polynomials increase with the number of control points. In the case with many control points, it is hard to manipulate such a curve. Moreover, any change to an individual control point leads to changes in the curve along its full length.

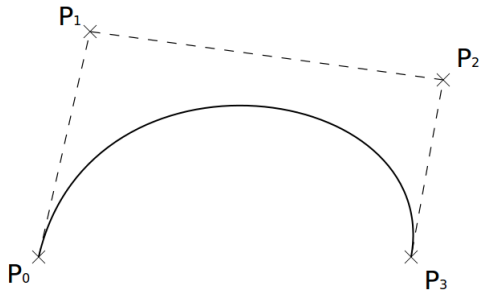


Figure 2.7: An example of Bézier curve.

Basis spline curves

To overcome the drawbacks of Bézier curve, the *Basis spline* (B-spline) [64] can be applied. It is a more general mathematical model based on Bézier curve. The degree of a B-spline curve is no more dependent on the number of control points, and any change made to a control point would only affect some neighboring curve segments. Given $n+1$ control points $\mathbf{P_i}, i = 0, \cdots, n$, and $m+1$

knots $0 = u_0 < u_1 < \cdots < u_m = 1$, the B-spline curve of degree p defined by these control points and knots is expressed by

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P_i} \tag{2.8}$$

where $N_{i,p}$ is the i^{th} B-spline basis function of degree p, it is computed recursively from two B-spline basis functions of degree $p-1$:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{2.9}$$

Note that n, m and p must satisfy $m = n + p + 1$. For example, if a B-spline curve of degree p with $n+1$ control points is to be defined, then the number of knots to be supplied is $n+p+2$. An illustration of $N_{i,p}$ of degree 0, 1, 2, and 3 is presented in Fig. 2.8. More details of B-spline curves can be found in [65]. An example of generating path for Unmanned Aerial Vehicle (UAV) using B-spline is presented in [66]. Another example using B-spline to model aircraft trajectories is presented in [67], and the problem of air-traffic conflict resolution is addressed through optimizing the spline control points.

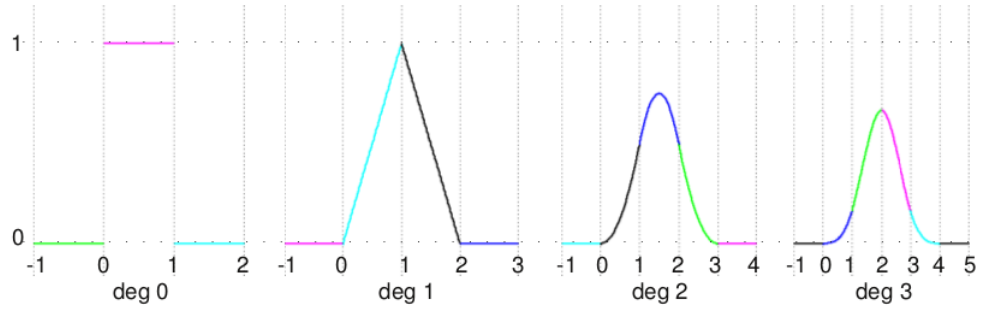


Figure 2.8: An illustration of $N_{i,p}$ of degree 0, 1, 2, and 3.

Dubins path

In a more specific problem context, given two points in the 2D plane with prescribed initial and terminal tangents, and also given a constraint on the curvature of a path, the *Dubins path* [68] is a strategy to find the shortest path joining the given oriented points while satisfying the curvature constraint. A Dubins path is composed by three portions, based on straight line segments (corresponding to moving forwards) or arcs of circles (corresponding to turning left or right) of a given radius. An example of Dubins path is illustrated in Fig. 2.9, where the shortest path is composed by a right turn R_α with central angle α, a straight line segment S_d with length d, and a left turn L_γ with central angle γ. In the case when obstacles are present in the environment, the problem becomes more difficult, some works contribute to path planning in such context using Dubins path, such as in [69] and [70]. The application of Dubins path is also extended to airplane path planning problems, such as in [71] and [72]. However, this method is not very adapted to the path planning in a TMA, where the tangent at the TMA entry/exit points are usually not fixed; besides, by fixing the radius of arcs of circles on a Dubins path, the search space for admissible paths is limited.

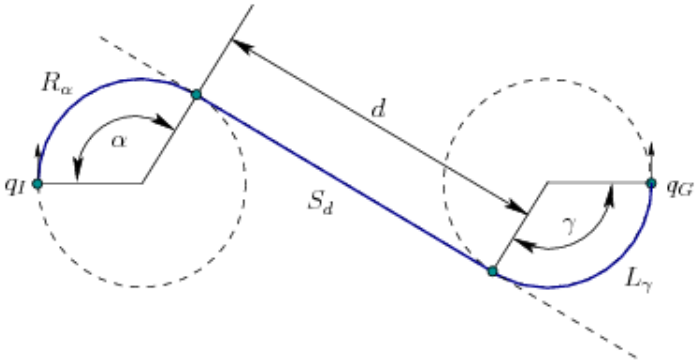


Figure 2.9: An example of Dubins path.

2.2 Route design in ATM

Nowadays optimal paths planning for air traffic is attracting a growing attention, some approaches of air traffic modeling and optimization have been presented in book [73]. Designing routes for the en-route (cruise) phase is generally addressed in 2D, since the altitude of an aircraft is a constant value or piecewise constant values at most of the time. On the contrary, the design of routes in a TMA is naturally to be considered in 3D, as aircraft take off and land with varied slopes in the vertical plane, thus the route can not stay in the horizontal plane. However, some papers simplify the problem to 2D by assuming that aircraft are associated with optimal profiles in the vertical plane. This kind of simplification does not take into account the possibility of evaluating the altitude of the designed route and the height of the obstacle. Hence, the situation when the route pass directly above or below the obstacle in the vertical plane, which may lead to a solution with shorter route length, is disregarded. In addition, modifying the vertical profile of a route, for example maintaining the flight level on a certain route section, is also a possible way to avoid obstacles.

The route design in 3D is more difficult than the one in 2D. Even though some environment modeling methods, such as some roadmap based methods, still hold in 3D, the corresponding resolution methods are less effective. Moreover, the design of 3D routes in TMAs includes an additional challenge, that is to take into account the aircraft vertical profiles. In fact, the take-off and landing slopes of aircraft depend on many factors, such as the aircraft weight and performance, the effect of wind, etc. Their values are generally bounded by the maximum/minimum take-off/landing slopes.

Concerning the design of multiple routes, the main issue is to take into account the separation between routes, which is especially important from the ATM point of view. Since the route design is not related to the notion of time, the routes must be separated spatially. Two routes are in *conflict* when a loss of separation occurs in both horizontal and vertical planes simultaneously. More precisely, for each pair of routes (γ_1, γ_2), let $p_1(x_{p_1}, y_{p_1}, z_{p_1}) \in \gamma_1$ and $p_2(x_{p_2}, y_{p_2}, z_{p_2}) \in \gamma_2$ be the closest points on the two routes. Denote N_h and N_v as the minimum separation norm in horizontal plane and vertical plane respectively. These two routes are in conflict if the following two conditions are verified at the same time.

$$(x_{p_1} - x_{p_2})^2 + (y_{p_1} - y_{p_2})^2 < N_h^2 \tag{2.10}$$

$$|z_{p_1} - z_{p_2}| < N_v \tag{2.11}$$

The design of multiple routes generally consists of conflict detection and resolution. Two strategies are proposed and applied in the literature. The first is a sequential *1-against-n strategy* where

the routes are generated one after the other according to a user-defined priority order (for example according to the decreasing traffic load on each route). The previously built routes become obstacles for the route that will be considered later. Therefore, the quality of solution depends strongly on the order of routes generation. For this reason, some papers apply a *global strategy*, where all routes are generated simultaneously, and the objective function is usually a global cost associated with the set of routes. In the following, we present the literatures of routes design in ATM domain by distinguishing the works in 2D or 3D.

2.2.1 Route design in 2D

In [4], the problem of designing one route that avoids hazardous weather is considered. The environment is modeled as a grid map, which is defined over a 2D rectangular region, and it is divided into very small squares. Each square is associated with a weight, indicating the cost for passing through this square. The more severe the weather condition is, the higher the weight is. Figure 2.10 illustrates the modeling of the hazardous weather, higher grid weight corresponds to darker color. The route is in a piecewise-linear form, composed of way points and links. In order to design the optimal route connecting the starting and ending points, while minimizing the total cost passing through the environment, a dynamic programming based on the Bellman-Ford shortest path algorithm is applied.

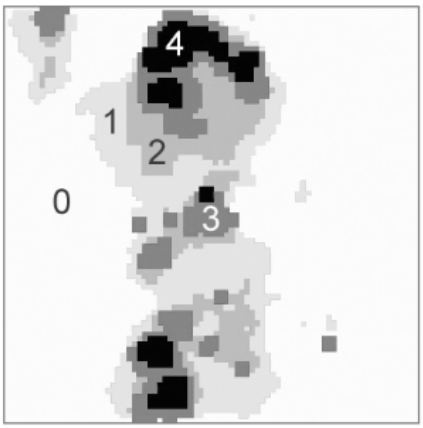


Figure 2.10: Hazardous weather modeling in [4].

In [74], a route design problem avoiding circular obstacles is studied in 2D. Each obstacle can be avoided by a turn in clockwise or counter-clockwise. The form of the optimal route is in fact determined by the ways of obstacles avoidance. To find the shortest route, a B&B method is applied, whose branching strategy is based on the decision of obstacle bypassing direction. The number of obstacles to be avoided is M. Each subproblem in the B&B search tree consists of a set of active obstacles, each active obstacle is associated with a bypassing direction. Let $A \subseteq \{1, \cdots, M\}$ denotes a subset of the indices of the active obstacles. The lower bound of a subproblem is equal to the length of a route built according to the active obstacles in subset A and their bypassing orientations, while ignoring the non-active obstacles. The branching of a subproblem is conducted based on the choice of counter-clockwise or clockwise bypassing around an inactive obstacle in the subset $\{1, \cdots, M\} \setminus A$. Different branching strategies for choosing the next subproblem to consider and for choosing the inactive obstacle to branch on are also developed and tested. The resolution approach applied in this thesis is inspired by this work. We extend its branching strategy to take into account

the specificity of our problem, where obstacles can be avoided also by imposing a level flight below the obstacle.

In [5], a route design problem in a TMA is considered. The design is in 2D, since the authors suppose that the aircraft perform the optimal altitude on the obtained 2D curve. An obstacle is contoured by a set of points, which form a convex region. Then the authors propose an approach based on building convex hulls around obstacles. Each obstacle can be bypassed clockwise or counter-clockwise. A Genetic Algorithm (GA) is used to deal with the choices of the bypassing direction on each obstacle. More precisely, An individual in the GA process consists of only one chromosome, which refers to a path. The chromosome stores the data of obstacle avoidance strategies, and the path is built based on these strategies. The fitness function is composed of two terms, the first one is linked to the path length, and the second one is associated with the path length lying inside obstacles. Figure 2.11 illustrates an example of a path bypassing two obstacles. Both obstacles are modeled by a set of points. The convex region formed by the red points is bypassed in counter-clockwise, and the other one (in green color) is bypassed in clockwise.

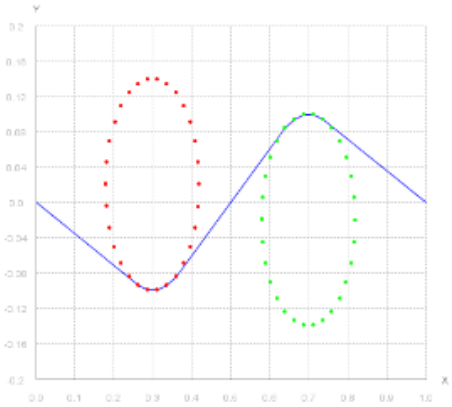


Figure 2.11: An example of a path bypassing two obstacles in [5].

In [75], the design of multiple arrival routes is considered. The RF of the RNP concept is taken into account, thus the routes are modeled as a succession of segments and arcs. The routes are generated sequentially taking into account some constraints such as the minimum and maximum radius of a turn and the minimum separation distance between two RF legs. The obtained arrival routes merge gradually before reaching the runway.

The paper [76] also deals with the problem of designing multiple arrival routes in 2D that merge gradually in a TMA. Some operational constraints in TMA are considered in this work, such as the obstacle avoidance and the separation between STAR merge points (in order to provide desirable route for control). Moreover, to exclude conflicts between departure and arrival traffic, SID-STAR crossing may only occur far from the runway. Since aircraft have different descent and climb slopes, the arriving and departing traffic are sufficiently separated in the vertical plane. The locations of TMA entry points and the runway, as well as the layout of SIDs are given as input data of the problem. In order to find the optimal STAR merge tree, a square grid map covering the design area is built up. The length of an edge of a grid pixel is equal to the minimum merge point separation distance, so that the merge point separation separation constraint is satisfied by any path in the grid. Besides, in order to satisfy the constraint on obstacle avoidance, any edge located in such an area is eliminated before designing the route. The objective function to be minimized is composed by two terms: the total length of the routes (each edge is counted as many times as it appears in the

designed routes) and the total length of the edges (each edge is counted only once). The problem is then formulated as an Integer Programming (IP) problem, and is solved by using Gurobi [77].

2.2.2 Route design in 3D

In [6], authors propose an optimization approach to generate departure and arrival routes in TMA while avoiding obstacles. In order to model the environment, a uniform 3D grid map is built up. Each grid point is associated with a coefficient, whose value varies in the range 0 to 1. If the grid point is located in a forbidden area, the coefficient is equal to 1; if it is located in free space, the coefficient is 0; if it is not located in a forbidden area but is preferred to be avoided (for example, grid points close to a forbidden area), the coefficient is between 0 and 1. In order to compute one optimal route, a Fast Marching Method (FMM) [78] is applied. It is a wavefront propagation method, designed to track the evolution of interfaces. To design multiple routes, a sequential strategy is applied. Each previously considered route is surrounded by a protection area. Afterwards, when generating the remaining routes, the previous routes as well as their protection area are treated as obstacles. The new built route can only pass through the free space, and the route separation requirement is satisfied in such a way. The form of obtained routes depends on the routes generation order. In fact, the free space for passing a route decreases with the increased number of existing routes. As a result, the previously generated routes are more possible to contain direct route sections, thus have less distance, and there may be no solution for the routes considered at last. Therefore, the authors apply a Simulated Annealing (SA) method to alternate the order of routes generation. Figure. 2.12 illustrates an example of the design of 8 routes, the obstacles are represented by the colored area.

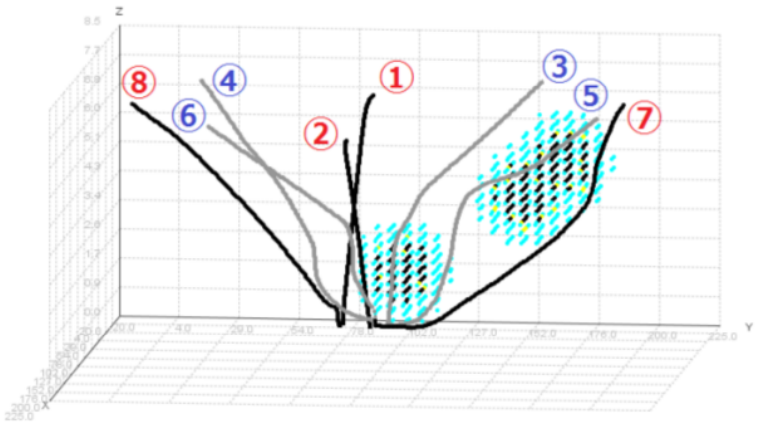


Figure 2.12: An example of multi-routes in [6].

The thesis [7] considers the design of terminal routes getting around obstacles, meanwhile taking into account the constraints such as the need for an aircraft to perform smooth turns and the vertical minima for noise restrictions. The vertical profile is estimated empirically using the real traffic data, and then modeled in form of a cone bounded by the minimum and maximum aircraft slopes. Then at each point of a route, a corresponding altitude interval in the vertical plane is computed based on the maximum and minimum slopes, and the flown distance from the starting point. The author applies the A* algorithm to search for the shortest path. Nodes are expanded as needed at each iteration, from the end node of the best possible partial route in the previous search. In order to deal with the constraints, some adjustments on the A* algorithm are made, on the way of expend-

ing the nodes and on the way of evaluating whether a node is eliminated. Let $u = (x_u, y_u, z_u, \theta_u, \beta_u)$ be the end node of the best partial path encountered during the previous search. The pair (x_u, y_u) is the coordinates of the node in the horizontal plane, $z_u = [z_{u,min}, z_{u,max}]$ represents the corresponding altitude interval governed by the minimum slope α_{min} and maximum slope α_{max}, and θ_u and β_u are the heading and total heading at this node respectively. In a general case, there are 5 nodes expanded from node u, denoted as $v_i = (x_{v_i}, y_{v_i}, z_{v_i}, \theta_{v_i}, \beta_{v_i}), i = 1, \cdots, 5$, as shown in Fig. 2.13. These 5 nodes are determined by a given distance Δd and a given incremental heading change $\Delta\theta$. More precisely, for each $v_i, i = 1, \cdots, 5$ we have:

$$\begin{aligned} \|(x_u, y_u) - (x_{v_i}, y_{v_i})\| &= \Delta d \\ \theta_{v_i} \in \{\theta_u - 2\Delta\theta, \theta_u - \Delta\theta, \theta_u, \theta_u + \Delta\theta, \theta_u + 2\Delta\theta\} & \end{aligned} \tag{2.12}$$

The altitude interval at v_i is then computed by $z_{v_i,min} = z_{u,min} + \alpha_{min}\Delta d$ and $z_{v_i,max} = z_{u,max} + \alpha_{max}\Delta d$. The total heading is computed by $\beta_{v_i} = \beta_u + |\theta_{v_i} - \theta_u|$. The requirement of performing a smooth path without large turns is accomplished by taking small values for Δd and $\Delta\theta$. To prohibit the spiral turns, an upper bound constraint on the total heading change is added. Moreover, the minimum altitude for the noise restriction, obstacle avoidance and route separation can be realized by eliminating the node located in such area. In Fig. 2.13, the node v_3 and v_4 are predicted to be in the obstacle region, thus they are not expanded in the future search. The optimal route is found when the ending point is reached. To design multiple routes, the routes are generated sequentially by regarding the existing routes are obstacle. In [79] [80], the weather avoidance routing problem is also solved by A* algorithm.

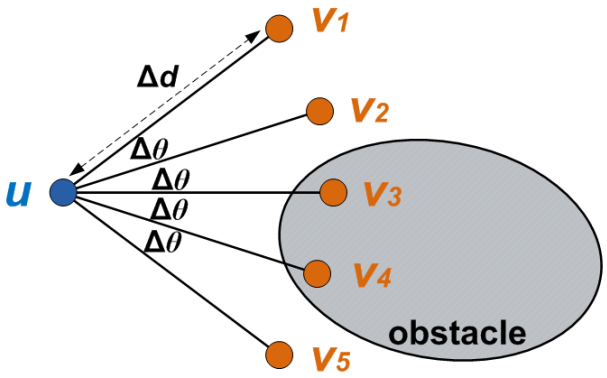
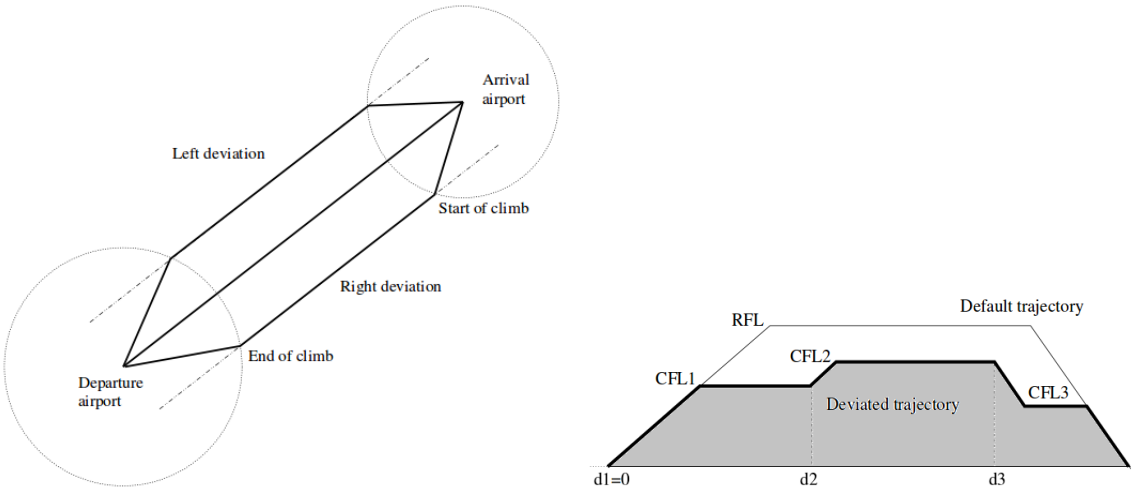


Figure 2.13: Node expanding in [7].

In [8] and its following works [81] [82], the problem of designing routes connecting pairs of airports is addressed. The starting point and ending point are both airports, thus the designed route covers all the flight phases: take-off, en-route and landing. In order to ensure the route separation constraint, a route can be both horizontal and vertical planes.

- In the horizontal plane, a route is in polygonal shape , as shown in Fig. 2.14(a). More precisely, in the TMA region, the route structure may have three possibilities, a route section directing to the destination airport, a route section deviated to the left and a route section deviated to the right. Outside the TMA region, three parallel segments are connected to the deviations in the TMA region. The radius around departure and arrival airports, as well as the angle of deviation are input data.



(a) (b)

Figure 2.14: Route deviations in [8]. (a) Deviation in the horizontal plane. (b) Deviation in the vertical plane.

- In the vertical plane, a succession of different flight levels are allowed, as shown in Fig. 2.14(b). The altitude RFL denotes the request flight level, which is maintained from the exit of the departure TMA until the entry of the arrival TMA. The choice of each level flight is denoted by (d_j, CFL_j), meaning that a vertical evolution towards flight level CFL_j is started at distance d_j from the starting airport. The climb or descent follow the given standard slopes. A natural constraint on the vertical profile is that it must end at the altitude of the destination airport.

The deviation cost of each route is measured in both horizontal and vertical planes. The horizontal deviation cost is computed by:

$$cost_H = K \frac{l - l_{ref}}{l_{ref}}, \tag{2.13}$$

where l is the actual route length in the horizontal plane, l_{ref} is the length of the direct route and K is a weight coefficient. The vertical deviation cost depends on the difference between the surface of the vertical profile with the request flight level and the surface of the vertical profile with the chosen flight levels. The latter surface is denoted as $surface$ and is illustrated by the gray area in Fig. 2.14(b). The vertical deviation cost is computed by:

$$cost_V = RFL - \frac{surface}{l}, \tag{2.14}$$

The total deviation cost for each route is the sum of $cost_H$ and $cost_V$.

The authors defined that an *interference* between two routes (γ_1, γ_2) occurs when the closest points p_1 and p_2 on them satisfy the following condition:

$$\frac{(x_{p_1} - x_{p_2})^2 + (y_{p_1} - y_{p_2})^2}{N_h^2} + \frac{(z_{p_1} - z_{p_2})^2}{N_v^2} < \sqrt{2} \tag{2.15}$$

This criterion brings an additional margin in the separation, in fact (2.10) and (2.11) imply (2.15). An A* algorithms is applied to generate sequentially multiple arrival routes without interference. For each route, nodes are generated iteratively in the A* search. The starting node refers to the

starting airport of the considered route, three son nodes can be developed from it, corresponding to the three possible deviations in the horizontal plane. The other nodes can have at most two sons, corresponding to maintaining or changing the flight level in the vertical plane. A node is discarded if it is in conflict with the existing routes. To estimate the heuristic cost, authors compute a vertical profile starting at the altitude corresponding to the current node x, and joining the default flight level (RFL) with a given linear climb or descent slope. The value of $h(x)$ is equal to the surface delimited by this computed vertical profile and the default profile defined by RFL, divided by the route length. Besides the 1-against-n strategy, the authors also try a global strategy through the use of Genetic Algorithm (GA), where routes are generated simultaneously. In their GA, each individual consists of n chromosomes, each chromosome represent a route, where n is the number of routes to be designed. Moreover, the route separation is no more considered as a constraint, but it is integrated in the cost function with a weight coefficient. The cost function formulated as a combination of two terms in the GA. The first term is a cumulated cost of routes deviations,and the second term is related to the total length of routes section involved in conflict. By minimizing the cost function associated with all routes, the conflict between routes is expected to be solved, or at least to be decreased.

2.3 Related problem of trajectory planning

As already mentioned, the main difference from the route design problem is that trajectory design is related to an individual mobile, and the speed of the mobile on the trajectory need to be determined, thus the notion of time is also taken into account. In the case of designing only one trajectory, similar methods as in route design problems can be applied, while considering additionally the dynamics of a mobile. However, in the case of designing multiple conflict-free trajectories, where a conflict refers to a loss of minimum separation between at least two mobiles, the conflict detection and resolution methods are different from the ones applied in multi-routes design problems. A review of conflict detection and resolution modeling methods is presented in [83]. Since this kind of research is not included in the scope of this work, we only present briefly some typical approaches for conflict detection and resolution between aircraft, as well as several recent works focus on multiple trajectories design in the ATM domain.

In order to detect a conflict, the future position of aircraft need to be predicted. One commonly used approach is to compute the mobile future position based on its flight plan (where the nominal trajectory and departure time are given), its current state and dynamic model, and the wind conditions. The uncertainty of conflict detection can be taken into account by enlarging the minimum separation criterion or introducing a safety buffer. Some applications of the conflict detection methods are presented in [84] [85] [86].

Considering conflict resolution, the *1-against-n* and *global* strategies, similarly as in multi-routes design problems, can be applied. Moreover, the conflict resolution is generally realized through the following approaches: horizontal deviation by heading change (as in [84] [87] [88]), vertical deviation by flight level change (as in [87] [85]), departure time allocation by ground holding (as in [86] [84]) and speed variation (as in [89] [90]). In the case where a great amount of trajectories need to be dealt with, several resolution approaches can be combined together.

In [84], a strategic trajectory planning methodology to minimize interaction between aircraft within European-continent scale is considered. The interaction between trajectories is predicted according to the flight plans. In order to minimize the interaction, horizontal deviations and departure time shifting can be applied to modify trajectories. More precisely, for one nominal trajectory to be modified, a set of virtual waypoints near the original en-route segment is created. The alter-

native trajectory is built by choosing at least one virtual waypoint, and connecting the successive waypoints with straight-line segments. Moreover, the departure time may be delayed by ground holding or shift in advance based on the nominal departure time. The problem is modeled as a highly combinatorial optimization problem and is solved by a SA method.

Some works in literature are based on nature-inspired approaches, for instance, force field method and light propagation algorithm. In *force field* method [91], the environment where the mobiles propagate is modeled as a force field. The destinations have an attractive force on mobiles, while obstacles generate repulsive forces on mobiles. Besides, mobiles act repulsive forces on each other. By following this force field, mobiles join the destinations while avoiding conflicts and obstacles. Some applications based on force field methods are presented in [92] [93]. In *light propagation algorithm* (LPA), the process of building a trajectory imitates the propagation of light between two points. The conflicting areas and obstacles are modeled as area with high refractive-index, so that they are avoided through the light wavefront propagation. In [94], LPA is applied to solve conflicts between 4D trajectories.

Another type of methods to solve trajectory planning problems is the *optimal control*, which aims at finding a control law for a given system, so that an optimality is obtained. A survey of optimal control applied in trajectory optimization problems is provided in [95]. In the ATM domain, it usually consists of a set of differential equations that describe the dynamics of aircraft, for instance, the position, speed, heading angle, bank angle, etc. In [96], the conflict resolution methods for the terminal area is considered, where the base of aircraft data (BADA) model is introduced to the optimal control theory. In the following work [97] [98] of [95], the merging optimization method for the terminal area is developed, it is able to optimize the trajectories of aircraft and their sequencing at merging point simultaneously. An optimal control problem can be also formalized as Hamilton-Jacobi-Bellman equations, whose solution gives the optimal cost of the corresponding dynamic system. The Ordered Upwind algorithm, first introduced in [99], provides approximate solution of Hamilton-Jacobi-Bellman equations. An application of Ordered Upwind algorithm is presented in [100], where trajectories minimizing congestion and travel time of each aircraft are generated.

2.4 Conclusion

In this chapter, we first reviewed some typical methods for path planning problems in a general context. These methods were categorized into two groups: roadmap based methods and path shape based methods. The roadmap based methods aim at capturing the free-space connectivity with a graph, and the path shape based methods represent the path to be designed as a combination of some basis analytical functions, so that the dimension of the state space is reduced. These methods are usually applied in 2D. Afterwards, the studies related to routes design in the Air Traffic Management (ATM) domain were presented. We first presented several works of designing one or multiple routes in 2D, by supposing that aircraft follow optimal profile in the vertical plane. This kind of simplification ignores the possibility of evaluating the altitude of the designed route and the height of the obstacle. Some exact approaches are applied to solve the problems. Then, some other works designing multiple routes in 3D were presented. As the route vertical profile and the pairwise separation between routes are considered additionally, the complexity of the problem increases. The exact approaches become less effective in such context, and the heuristic approaches are applied in the mentioned works. In order to deal with the pairwise separation between routes, two different strategies can be applied: the first is a sequential strategy, where routes are generated sequentially by considering the existing routes as constraints; the second is a global strategy where routes are

generated simultaneously. In order to separate two routes in a Terminal Maneuvering Area (TMA) in 3D, the most commonly used method in literature is the horizontal route deviation. The possibility of modifying the vertical profile of a route (for example, maintaining the flight level on a route section) is not very much explored. In this thesis, we consider the route deviation in the vertical plane as an additional and effective way for obstacle avoidance and routes separation, since it enriches the space of possible maneuvers and corresponds to what is done in practice in a TMA. As we will see in the next chapter, our 3D obstacles and routes modeling allows the possibilities of deviating a route in both horizontal and vertical planes.

Chapter 3

Problem Modeling

In this chapter, we present a mathematical framework to deal with the routes design problem in a TMA. We first present the input data of this problem, related to airport configuration and the routes to design. In Section 3.2, we give the ways the obstacles and routes are modeled. In Section 3.3, we formulate the problem as a mathematical optimization problem.

3.1 Input data

The input data of this problem falls into two groups. The first one concerns the airport configuration, as presented in Table 3.1. The second group is related to the characteristics of the routes to design, as presented in Table 3.2.

the total number of obstacles to be avoided, $M \in \mathbb{N}$
the total number of the runways in the considered airport
the usage of each runway (take-off or landing)
the direction of take-off or landing on each runway
the coordinates and altitudes of thresholds and center for each runway
the coordinate and altitude of the corresponding FAF for a runway used for landing

Table 3.1: Input data related to airport configuration. The related coordinates are given in a Cartesian Coordinate System.

3.2 Obstacle and route modeling

There exist different ways to model routes and obstacles, as presented in Chapter 2. The modeling usually has an impact on the resolution method to be used. Most of the previously mentioned works consider the modeling in 2D. In most of the works related to the design of routes in 3D, the vertical profiles of the aircraft are not specifically dealt with. In this section, we present how we model the 3D obstacles and routes, where the form of a route is compatible with the advanced navigation mode on the one hand, and takes into account the vertical profiles of aircraft on the other hand.

the number of routes to be built, $N \in \mathbb{N}$
the starting point A_i (x_{A_i}, y_{A_i}) with its altitude H_{A_i} for route i, $i = 1, \cdots, N$
the ending point B_i (x_{B_i}, y_{B_i}) for route i, $i = 1, \cdots, N$
traffic load on each route
minimum (respectively, maximum) take-off slope $\alpha_{min,TO}$ (respectively, $\alpha_{max,TO}$)
minimum (respectively, maximum) landing slope $\alpha_{min,LD}$ (respectively, $\alpha_{max,LD}$)
minimum (respectively, maximum) radius on a RF leg R_{min} (respectively, R_{max})
maximum number of level flights on each route N_{max}
minimum altitude of each level flight H_{min}
minimum length of each level flight L_{min}
minimum distance between two successive level flights D_{min}

Table 3.2: Input data related to routes to design.

3.2.1 Obstacle modeling

The obstacles in a TMA could be mountains, cities, military area, etc. The way of modeling the environment usually affects the form of designed routes. In this work, the obstacles (together with their protection areas), in number of $M \in \mathbb{N}$, are modeled as cylinders in 3D as presented in Fig. 3.1. Each cylinder $\Omega_j, j = 1, \cdots, M$ is defined by $(C_j(x_j, y_j), r_j, z_{j_{inf}}, z_{j_{sup}})$, where $C_j(x_j, y_j)$ and r_j are the center and the radius of the two bases respectively; $z_{j_{inf}}$ and $z_{j_{sup}}$ are the altitude of the lower and upper bases respectively.

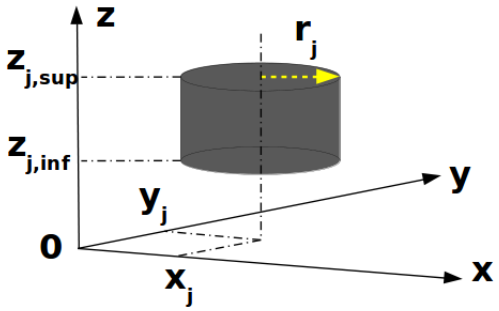


Figure 3.1: Obstacle modeling.

In our design, we need to pre-process the obstacles according to the relative positions of their projections in the horizontal plane. Let $\Omega_j(C_j(x_j, y_j), r_j, z_{j_{inf}}, z_{j_{sup}})$ and $\Omega_k(C_k(x_k, y_k), r_k, z_{k_{inf}}, z_{k_{sup}})$ be two distinct obstacles. Their projections in the horizontal plane are two disks centered at (x_j, y_j) and (x_k, y_k) respectively, and with radius r_j and r_k respectively. These two disks can be:

- *disjoint*, if and only if
$$(x_j - x_k)^2 + (y_j - y_k)^2 > (r_j + r_k)^2. \tag{3.1}$$
In this case, they have 2 external common tangents and 2 internal common tangents (Fig. 3.2(a)).

- *externally tangent*, if and only if
$$(x_j - x_k)^2 + (y_j - y_k)^2 = (r_j + r_k)^2. \tag{3.2}$$

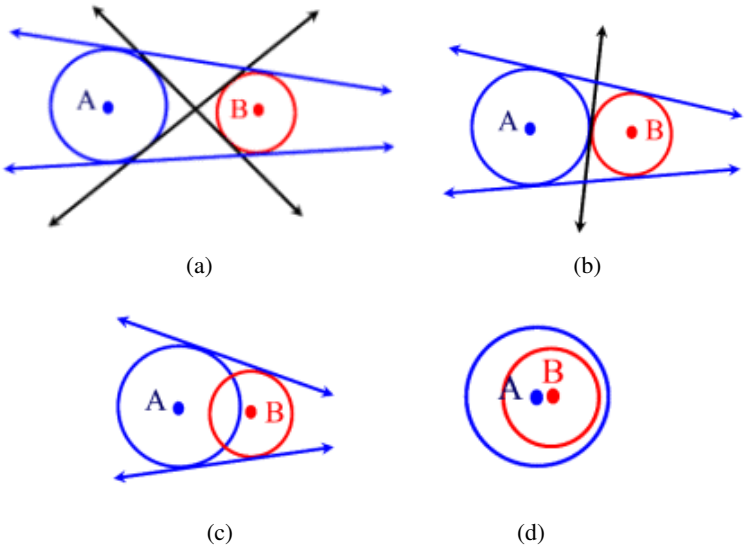


Figure 3.2: Relative position between two disks. (a) Disjoint disks. (b) Externally tangent disks. (c) Overlapped disks. (d) One disk is completely included in another. The external (respectively, internal) common tangents are in blue (respectively, black) color.

In this case, they have 2 external common tangents and 1 internal common tangent (Fig. 3.2(b)).

- *overlapped*, if and only if

$$(r_j - r_k)^2 < (x_j - x_k)^2 + (y_j - y_k)^2 < (r_j + r_k)^2. \tag{3.3}$$

In this case, they only have 2 external common tangents (Fig. 3.2(c)).

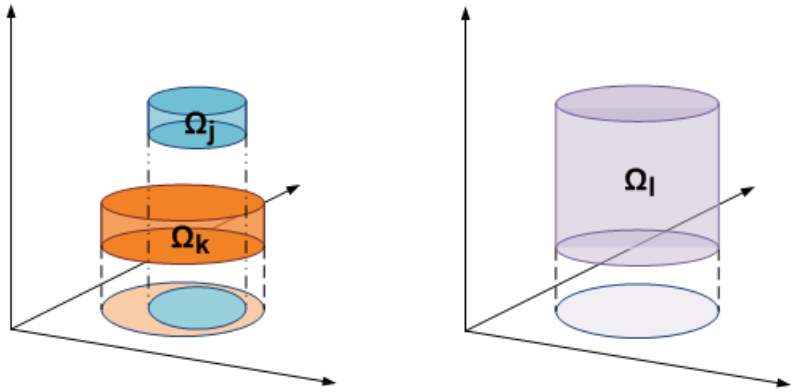
- one disk is *completely included* in another, if and only if

$$(x_j - x_k)^2 + (y_j - y_k)^2 \leq (r_j - r_k)^2. \tag{3.4}$$

In this case, they have one or none common tangent line (Fig. 3.2(d)).

In this thesis, we allow only three relative positions of the obstacles' projections in the horizontal plane: disjoint (Fig. 3.2(a)), externally tangent (Fig. 3.2(b)) and overlapped (Fig. 3.2(c)). Note that, in the case when the projection of one cylinder is completely included in the projection of another (Fig. 3.2(d)), for the sake of simplification, these two cylinders are re-grouped as a new larger cylinder enveloping them (as shown in Fig. 3.3(b)). The new cylinder denoted as $\Omega_l(C_l(x_l, y_l), r_l, z_{l_{inf}}, z_{l_{sup}})$ is characterized by:

$$\begin{aligned} (x_l, y_l, r_l) &= \begin{cases} (x_j, y_j, r_j), & \text{if } r_j > r_k \\ (x_k, y_k, r_k), & \text{otherwise} \end{cases} \\ z_{l_{inf}} &= \min(z_{j_{inf}}, z_{k_{inf}}) \\ z_{l_{sup}} &= \max(z_{j_{sup}}, z_{k_{sup}}) \end{aligned} \tag{3.5}$$



(a) (b)

Figure 3.3: Two cylinders separated in 3D, while the projection of one cylinder is included in the projection of the other. (a) Illustration of two obstacles Ω_j and Ω_k. (b) Illustration of obstacle Ω_l enveloping Ω_j and Ω_k.

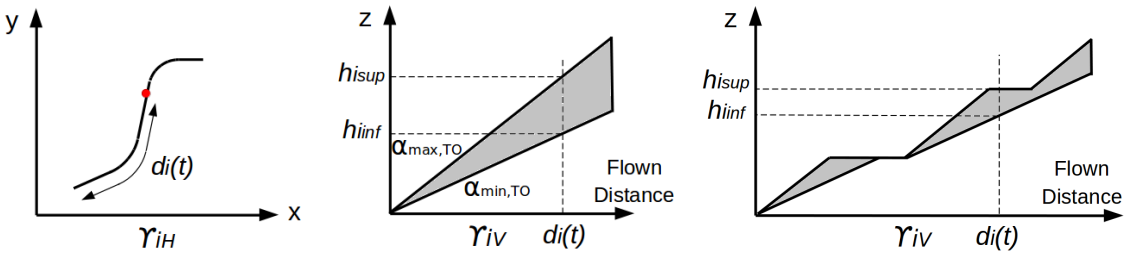
3.2.2 Route modeling

We define a 3D route $\gamma_i, i = 1, \cdots, N$ by two elements: a curve γ_{i_H} in the horizontal plane, associated with a cone γ_{i_V} in the vertical plane.

- The horizontal curve γ_{i_H} is designed in the form of a succession of segments and arcs of circles, motivated by the fact that the shortest path among circular obstacles in the horizontal plane consists of segments (connecting tangentially two obstacles) and arcs of circles (lying on the border of obstacles) [52]. An example is shown in Fig. 3.4(a). The form of the horizontal curve, on the one hand simplifies the route modeling, on the other hand corresponds to the shape of SIDs/STARs under RNP. More precisely, a segment defined by two tangent points corresponds to a standard point-to-point leg in ATM, and an arc corresponds to a RF leg in ATM, defined by starting and ending tangent points, turn center and radius.

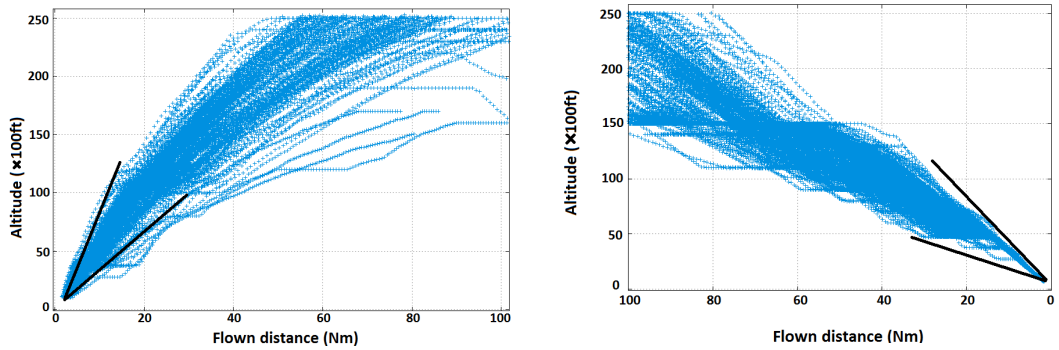
- The vertical cone γ_{i_V} contains all ascent (or descent) profiles of the aircraft flying on this route. The vertical profile is a band enclosed by two piecewise linear continuous functions. Two examples are shown in Figs. 3.4(b), 3.4(c)), where the bands are represented by the shaded areas. More precisely, the slope of each segment on the function corresponding to the lower bound is either α_{min} (on a climbing or descending route section) or 0 (on a route section where the flight level is maintained). Similarly, the one on the function corresponding to the upper bound is either α_{max} or 0. By abuse of language, this band is called by *cone*. The idea of taking a cone that contains all vertical profiles is inspired by the behavior illustrated in Fig. 3.5, which shows some real take-off and landing data of Paris CDG airport. From the figure we can see clearly that the vertical profiles are contained in a cone defined by two straight lines in either case. This behavior is mainly due to the different aircraft masses and performances and to the effect of the wind.

In the horizontal plane, γ_{i_H} connects a starting point $A_i\ (x_{A_i}, y_{A_i})$ to an ending point $B_i\ (x_{B_i}, y_{B_i})$. In a SID case, the starting point is at the midpoint of a runway threshold and the ending point is an exit point of a TMA. In a STAR case, for the sake of simplicity of implementation, we build the route considering a FAF as starting point and an entry point of TMA as ending point. Figure 3.6 is



(a) (b) (c)

Figure 3.4: Examples of $\boldsymbol{\gamma}_{i_H}$ and $\boldsymbol{\gamma}_{i_V}$.



(a) (b)

Figure 3.5: Take-off and landing profiles in CDG airport. (a) Take-off profiles. (b) Landing profiles.

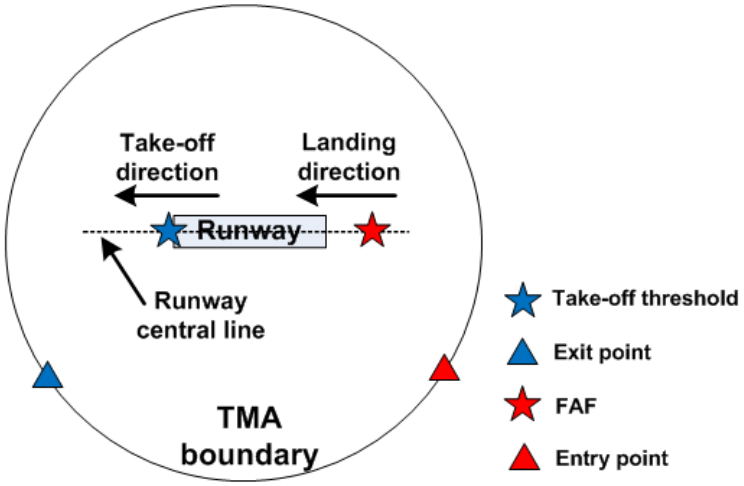


Figure 3.6: An example of a TMA in a circular shape.

an example of a TMA in a circular shape where the starting and ending points are illustrated. The FAF is usually aligned to the central line of the corresponding runway (Fig. 3.6), about 10Nm away from the runway threshold. To guarantee a standard final approach with a 3° descent slope, the FAF is about 3000ft above the runway threshold. Aircraft follow specific guidance (for example, the

Instrument Landing System (ILS)) from the FAF to the runway threshold, thus the route section between FAF and the runway threshold is not considered in this thesis. Mathematically, the horizontal route γ_{i_H} is a smooth mapping defined as:

$$\gamma_{i_H} : [0,1] \to \mathbb{R}^2 \tag{3.6}$$

where $\gamma_{i_H}(0) = (x_{A_i}, y_{A_i})$ and $\gamma_{i_H}(1) = (x_{B_i}, y_{B_i})$.

In the vertical plane, the starting point A_i (x_{A_i}, y_{A_i}) is associated with an altitude H_{A_i}, corresponding to the airport elevation in a SID case, and to the altitude of the FAF in a STAR case. The vertical profile γ_{i_V} is defined as:

$$\gamma_{i_V} : \begin{array}{ccc} [0,1] & \to & I^{\mathbb{R}} \\ t & \to & [h_{i_{inf}}(d_i(t)), h_{i_{sup}}(d_i(t))] \end{array} \tag{3.7}$$

where $I^{\mathbb{R}}$ defines the set of intervals of $\mathbb{R}$, and $d_i(t) = \int_0^t \|\gamma'_{i_H}(s)\|_2 \, \mathrm{d}s$ is the flown distance until t in the horizontal plane, $[h_{i_{inf}}(d_i), h_{i_{sup}}(d_i)]$ is the interval defined by the cross section of the cone at d_i, and $\gamma_{i_V}(0) = [H_{A_i}, H_{A_i}]$. Figure 3.4 illustrates how γ_{i_V} is associated with γ_{i_H} in the case of a SID, where $H_{A_i} = 0$ and $\alpha_{min,TO}$ (respectively, $\alpha_{max,TO}$) is the minimum (respectively, maximum) take-off rate of aircraft on this route.

3.3 Optimization problem formulation

In this section, we first introduce the proposed decision variables related to the obstacles avoidance strategies, which will have an impact on the form of a route. Next, the constraints to be taken into account in the design are explained. Finally, we give the objective function to be minimized in the design.

3.3.1 Decision variables

Considering the form of a route in the horizontal plane which consists of segments (connecting tangentially two obstacles) and arcs of circles (lying on the border of obstacles), as well as the possibility of imposing a level flight under an obstacle in the vertical plane, we define an obstacle as *active* when it is touched by a route and it has to be avoided according to one of the following maneuvers: turn counter-clockwise, turn clockwise or impose a level flight. For a route γ_i, each cylinder Ω_j is associated with two decision variables s_{ij} and t_{ij}, s_{ij} defines whether Ω_j is active or not with respect to the route γ_i:

$$s_{ij} = \begin{cases} 0, & \text{if } \Omega_j \text{ not active} \\ 1, & \text{if } \Omega_j \text{ active} \end{cases} \qquad \forall i = 1, \cdots, N;\ \forall j = 1, \cdots, M \tag{3.8}$$

while t_{ij} defines the ways an active obstacle Ω_j is avoided on the route γ_i:

$$t_{ij} = \begin{cases} 0, & \text{if turn counter-clockwise} \\ 1, & \text{if turn clockwise} \\ 2, & \text{if impose a level flight below obstacle } \Omega_j \end{cases} \qquad \forall i = 1, \cdots, N;\ \forall j = 1, \cdots, M \tag{3.9}$$

Note that when an obstacle is not active ($s_{ij} = 0$), the value of t_{ij} is not applicable in the design.

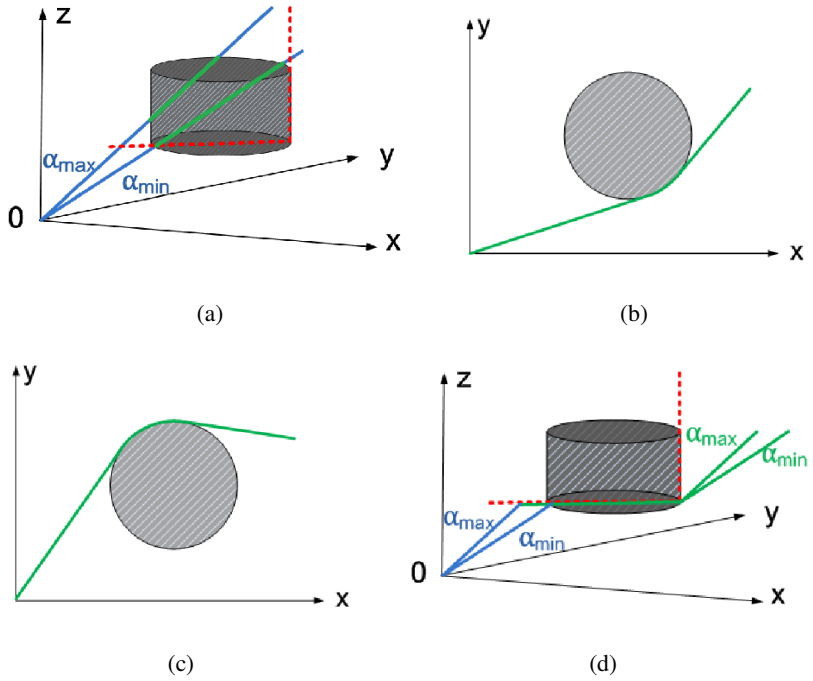


Figure 3.7: The routes associated with different values of the decision variables. (a) $s_{ij} = 0$, 3D View. (b) $(s_{ij}, t_{ij}) = (1, 0)$, 2D View. (c) $(s_{ij}, t_{ij}) = (1, 1)$, 2D View. (d) $(s_{ij}, t_{ij}) = (1, 2)$, 3D View.

An illustration of different values of the decision variables for an example of a SID γ_i with one obstacle Ω_j is presented in Fig. 3.7. In Fig. 3.7(a), the obstacle is not active, so $s_{ij} = 0$. The horizontal route is a straight line segment connecting A_i and B_i. It is associated with a cone in the vertical plane. This route in the considered example is not a feasible one, because it intersects the obstacle. Then when the obstacle is active ($s_{ij} = 1$), 3 possibilities are considered to avoid it: turn counter-clockwise (Fig. 3.7(b)), turn clockwise (Fig. 3.7(c)) and impose a level flight under the obstacle at altitude $z_{j_{inf}}$ (Fig. 3.7(d)), corresponding to $t_{ij} = 0, 1, 2$ respectively. It can be seen that the decision variables not only decide the obstacle avoidance strategies, but also control the form of the designed route. However, at this stage, the values of these variables do not define a unique route. We will explain the way a unique route is defined in Section 4.1.

3.3.2 Constraints

The numerous constraints to be considered in this thesis are listed here:

- obstacle avoidance;
- route separation;
- RF related constraints;
- monotonicity for the vertical profile;
- level flights related constraints;
- runway alignment.

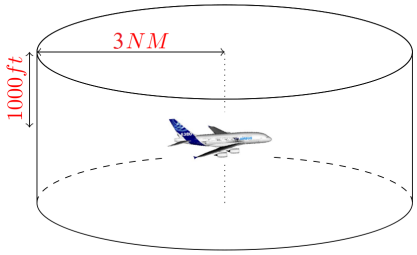


Figure 3.8: Standard separation norm in a TMA.

Obstacle avoidance

For the safety reason, the obstacles such as the restricted area, mountains, cities must be taken into account in the route design. This constraint is considered and handled through the choice of the decision variables.

Route separation

The standard separation norm between pairwise aircraft in a TMA is 3Nm in the horizontal plane or 1000ft in the vertical plane, as shown in Fig. 3.8. We design routes with respect to the same separation criteria, *i.e.*, two routes are separated when their closest points are separated by 3Nm in the horizontal plane or 1000ft in the vertical plane. As a consequence, aircraft following two different routes are automatically separated.

RF related constraints

The first constraint related to RF is the minimum turn radius. The ability for an aircraft to remain on track during an RF leg is limited by bank angle and ground speed [101] [102], linked to each other by the following relation:

$$R = \frac{V_{GS}^2}{g \cdot tan(\Phi)} \tag{3.10}$$

where R is the required radius on the RF leg, V_{GS} is the ground speed, Φ is the bank angle and g is the acceleration of gravity. According to (3.10), for a fixed value of V_{GS}, the lowest radius of a flyable RF leg is given by $R_{min}(V_{GS}) = \frac{V_{GS}^2}{g \cdot tan(\Phi_{max})}$, where Φ_{max} is the maximum value of the bank angle. Consequently, in order to ensure that a RF leg is flyable for all aircraft following it, the radius of the RF leg has to be greater than $R_{min} = \frac{V_{GS,max}^2}{g \cdot tan(\Phi_{max})}$, where $V_{GS,max}$ is the maximum ground speed in a TMA. According to [103], we take $V_{GS,max} = 400$kt, $\Phi_{max} = 25°$, the corresponding minimum radius R_{min} is equal to 5NM. To satisfy the constraint on minimum turn radius, a pre-processing is applied on the obstacles: for an obstacle with radius lower than R_{min}, its radius is increased to R_{min} and its center and altitudes of lower and upper bases are not changed. In order not to perform a very big turn, we also impose an upper bound on the RF leg radius. Let R_{max} be the largest radius of a RF leg. According to [104], which is in support of FAA memorandum of agreement no.DTFAWA-11-A-80009, the value of R_{max} is set to be 13Nm. In the pre-processing step, all input obstacles are modeled as cylinders whose radius of lower and upper bases is lower than R_{max}. The third constraint related to RF is that RF legs must terminate at least 2NM prior to the FAF [101]. This constraint is treated when considering the runway alignment constraint.

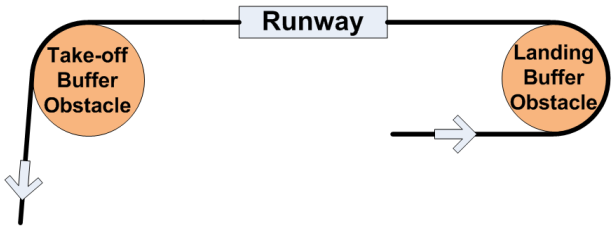


Figure 3.9: Buffer obstacle illustration.

Monotonicity for the vertical profile

The vertical profile for a SID should be monotonically ascending and for a STAR should be monotonously descending. Thus, for $u, v \in [0,1]$, if $u \leq v$, we must have

$$h_{i_{inf}}(u) \leq h_{i_{inf}}(v),\ i = 1, \cdots, N \tag{3.11}$$

$$h_{i_{sup}}(u) \leq h_{i_{sup}}(v),\ i = 1, \cdots, N \tag{3.12}$$

This constraint is satisfied by the route construction.

Level flights related constraints

Since the changes in flight levels have an influence on the fuel consumption and CO_2 emission, the number of level flights on each route is bounded by a maximum number N_{max}, usually fixed to 2, for route γ_i:

$$\sum_{j=1}^{M} \max(t_{ij} - 1, 0) \leq N_{max} \tag{3.13}$$

Moreover, as the altitudes of imposed level flights have a direct impact on the noise pollution, a minimum altitude H_{min} for each level flight is defined. In practice, we impose the following constraints: for an obstacle Ω_j, if $z_{j_{inf}} < H_{min}$, then no level flight is imposed below it, therefore $\forall i, t_{ij} \in \{0,1\}$. Next, as to take into account the passengers comfort, the length of each level flight should not be too short, a minimum length L_{min} for each level flight is then imposed. Finally, in order not to have two successive level flights too close to each other, we define D_{min} as the minimum distance between two successive level flights.

Runway alignment

A SID must join smoothly the following route section from a take-off leg and a STAR must head straightly to the corresponding runway before the FAF. To deal with the runway alignment constraint, we propose the notion of *buffer* obstacles as shown in Fig. 3.9. Each route γ_i is associated with one buffer obstacle Ω_{b_i} in a cylinder shape. A buffer obstacle Ω_{b_i} is defined by $(C_{b_i}(x_{b_i}, y_{b_i}), r_{b_i}, z_{b_{i_{inf}}}, z_{b_{i_{sup}}}, t_{\Omega_{b_i}})$, where $t_{\Omega_{b_i}}$ is the turn orientation on Ω_{b_i} and the other parameters are defined in the same way as in Subsection 3.2.1. The values of these parameters are user-defined, in such a way that the segment connecting tangentially the buffer obstacle and the starting point is parallel to the corresponding runway, so that the designed route joins straightly the runway. As a buffer obstacle is always active, the choice of its avoiding strategy is given between counter-clockwise turn or clockwise turn. Note that, as aircraft ground speed is very low near the runway, the radius of a runway buffer obstacle can be lower than R_{min}.

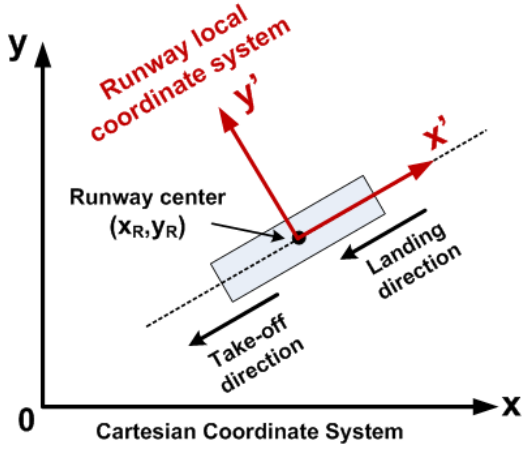


Figure 3.10: A runway in the Cartesian Coordinate System.

To present the configuration of the runway buffer obstacles more precisely, we introduce a local coordinate system related to a runway. An example of the runway corresponding to γ_i is presented in Fig. 3.10 in the proposed Cartesian Coordinate System. The runway center is (x_R, y_R). The local coordinate system related to this runway is defined in the following way: the origin centralizes at (x_R, y_R), the direction of the x'-axis is opposite to the landing direction and the direction of the y'-axis is obtained by a 90° turn in counter-clockwise with respect to x'-axis. The coordinates of the starting point (x_{A_i}, y_{A_i}) and the center of a buffer obstacle (x_{bi}, y_{bi}) in the Cartesian Coordinate System are converted to $((x'_{A_i}, 0))$ and (x'_{bi}, y'_{bi}) respectively in the runway local coordinate system.

The configuration of a buffer obstacle is related to whether the route to design is a SID ($x'_{A_i} < 0$) or STAR ($x'_{A_i} > 0$), and to the given buffer obstacle bypassing orientation $t_{\Omega_{bi}}$. There are in total four possibilities, as shown in Fig. 3.11. More precisely,

- case 1: a SID case ($x'_{A_i} < 0$), turn clockwise on buffer obstacle ($t_{\Omega_{bi}} = 1$);
- case 2: a SID case ($x'_{A_i} < 0$), turn counter-clockwise on buffer obstacle ($t_{\Omega_{bi}} = 0$);
- case 3: a STAR case ($x'_{A_i} > 0$), turn counter-clockwise on buffer obstacle ($t_{\Omega_{bi}} = 0$);
- case 4: a STAR case ($x'_{A_i} > 0$), turn clockwise on buffer obstacle ($t_{\Omega_{bi}} = 1$).

In order to ensure the runway alignment in both SID and STAR cases, and to guarantee that RF legs must terminate at least 2NM prior to the FAF in a STAR case, the given parameters of the buffer obstacle must satisfy:

- in case 1: $y'_{bi} = r_{bi}$, $x'_{bi} < x'_{A_i}$;
- in case 2: $y'_{bi} = -r_{bi}$, $x'_{bi} < x'_{A_i}$;
- in case 3: $y'_{bi} = r_{bi}$, $x'_{bi} - x'_{A_i} > 2\text{Nm}$;
- in case 4: $y'_{bi} = -r_{bi}$, $x'_{bi} - x'_{A_i} > 2\text{Nm}$.

Moreover, note that the construction of buffer obstacles is determined by data like runway configuration and SID/STAR case, and is done in a pre-processing step. The only degrees of freedom that have to be handled are the radius of the buffer obstacle and the distance between the corresponding runway threshold and the abscissa of the buffer obstacle centered in the runway local coordinate system.

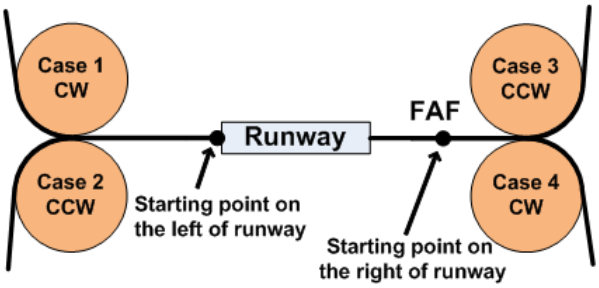


Figure 3.11: Different buffer obstacle configurations. CW is clockwise for short, CCW is counter-clockwise for short.

3.3.3 Objective function

To define our objective function, let us first define, for each route γ_i, a weighted sum L_{γ_i} of two terms: $L_{\gamma_{i_H}}$, the length of horizontal curve γ_{i_H}, and $\ell_{i_{LF}}$, the sum of the lengths of route sections corresponding to level flights projected on the horizontal plane. More precisely:

$$L_{\gamma_i} = c_1 L_{\gamma_{i_H}} + c_2 \ell_{i_{LF}} \tag{3.14}$$

where the coefficients c_1 and c_2 are two penalty parameters, whose values are user-defined parameters depending on the weight of the corresponding term. The values of $L_{\gamma_{i_H}}$ and $\ell_{i_{LF}}$ depend on the form of γ_{i_H} and γ_{i_V}, which are determined by the choice of decision variables $\{(s_{ij}, t_{ij}) |\ j = 1, \cdots, M\}$. However, it is difficult to find explicit mathematical formulations to express $L_{\gamma_{i_H}}$ and $\ell_{i_{LF}}$ by decision variables. The way these two terms are computed is associated with the way a route is built. This is presented later in Section 4.1 (Algorithms 1 and 2). Finally, we minimize the sum of $L_{\gamma_i}, i = 1, \cdots, N$:

$$L = \sum_{i=1}^{N} L_{\gamma_i} \tag{3.15}$$

To summarize, the route design problem can be defined by the following combinatorial optimization problem $(\mathscr{P})$:

$$(\mathscr{P}) \begin{cases} \min & L = \sum_{i=1}^{N} L_{\gamma_i} \\ \text{s.t.} & \text{obstacle avoidance} \\ & \text{route separation} \\ & \text{runway alignment} \\ & \text{RF related constraints} \\ & \text{monotonicity for the vertical profile} \\ & \text{level flights related constraints} \end{cases} \tag{3.16}$$

3.4 Conclusion

In this chapter, we proposed a mathematical framework to deal with the routes design problem in a Terminal Maneuvering Area (TMA), and we formulated the problem as a combinatorial optimization problem. The design of 3D routes is a very complex problem, because of its numerous mentioned constraints. To deal with this difficulty, we will solve this problem in two steps. We

will first consider, in Chapter 4, a simpler subproblem of designing one optimal route. Then, the design of multiple routes will be addressed in Chapter 5, where the challenging constraint of routes separation is considered additionally.

Chapter 4

Designing One Route Using Branch and Bound

The numerous constraints to be considered make the SIDs/STARs design a very complex problem. Therefore in this chapter we consider the simpler subproblem of designing only one optimal route, taking into account all the constraints presented in Subsection 3.3.2, except the route separation. This is indeed the basis of the solution approach to build multiple routes, where the constraint of route separation is considered additionally. As already mentioned, the problem is a combinatorial optimization problem, where the objective function and constraints cannot be explicitly represented in terms of decision variables. The Branch and Bound (B&B) method is applied to solve the problem. In Section 4.1, we present the way one route is built when some values of the decision variables are given. In Section 4.2, we explain the B&B method tailored to our problem, and we give a step-by-step illustration to show how it works. In Section 4.3, a method to reduce the number of obstacles, and consequently the size of the state space is introduced. Finally, in Section 4.4 some simulation results of designing one optimal route are presented.

4.1 Building one route

We explain in this section how a unique route connecting points A_i and B_i, is associated with the given values of some decision variables, denoted as $\{(s_{ij},t_{ij})\}_{j\in J}, J \subset \{1,\cdots,M\}$, which in fact correspond to a subproblem in the B&B method. The form of this unique route depends on the subset J, but for the sake of simplification, we omit J and we denote the route by γ_i. Similarly, the corresponding horizontal curve (respectively, vertical cone) is denoted by γ_{i_H} (respectively, γ_{i_V}), instead of $\gamma_{i_H}^J$ (respectively, $\gamma_{i_V}^J$). For the obstacles whose values of decision variables are not given, they are regarded as non-active when building the route, that is $\{(s_{ij},t_{ij}) = (0,0) | j \in \{1,\cdots,M\} \setminus J\}$.

4.1.1 Building a horizontal curve γ_{i_H}

The horizontal curve γ_{i_H} is built as presented in Algorithm 1. First, the active obstacles to be avoided by a turn ($t_{ij} = 0$ or 1), except the buffer obstacle, are numbered in an increasing order of $length(A_i, Proj_{(A_iB_i)}C_j)$, where $Proj_{(A_iB_i)}C_j$ is the projection of the center C_j onto the line (A_iB_i) in the horizontal plane, as shown in Fig. 4.1. Note that building a horizontal curve with respect to the obstacle numbered in such a way simplifies the computation but it does not necessarily lead to the shortest horizontal curve between A_i and B_i. Afterwards, we compute the tangent points first

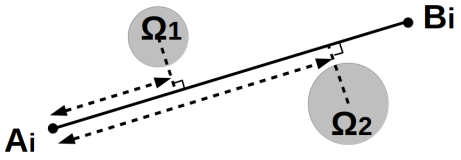


Figure 4.1: Obstacle numbering.

on the buffer obstacle then on the successive active ($s_{ij} = 1$) obstacles which are associated with counter-clockwise ($t_{ij} = 0$) or clockwise ($t_{ij} = 1$) turns in the increasing order of their numbering.

The way of computing the tangent points depends on the bypassing orientations on the successive obstacles. Different cases are presented in Fig. 4.2. Afterwards, the horizontal curve is built by connecting the successive tangent points, using tangent segments or arcs of circles. More precisely, a tangent segment is used to connect the starting point to the buffer obstacle, the obstacle to the ending point, or two successive obstacles. An arc is used to avoid an obstacle (including the buffer obstacle). Note that in the case of two successive externally tangent obstacles that are avoided by opposite bypassing orientations, the extremities of the segment connecting the two obstacles are the same tangent point (points T_k^3 and T_{k+1}^3 in Fig. 4.2(c)). However, we still regard this kind of segment as a tangent segment. The horizontal curve is hence built piecewise, composed alternatively by segments and arcs. The length of the horizontal curve, denoted as $L_{\gamma_{i_H}}$ in Subsection 3.3.3, is computed by summing up the length of segments and arcs that compose the curve.

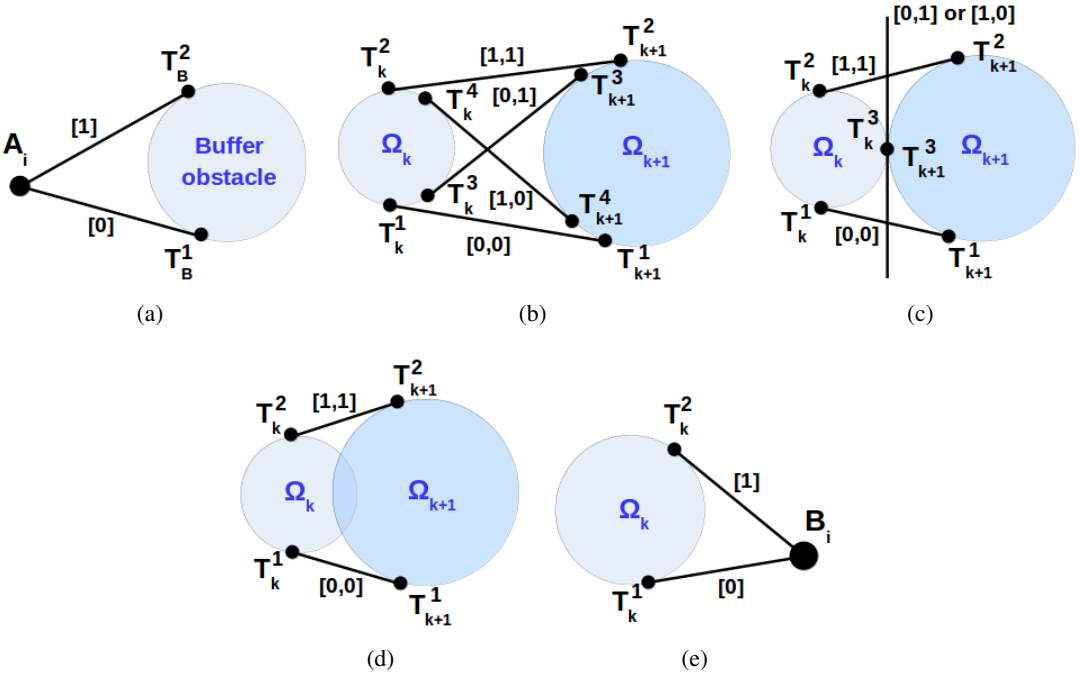


Figure 4.2: Different cases of computing the tangent points. (a) Case of the buffer obstacle. (b) Case of two disjoint obstacles. (c) Case of two externally tangent obstacles. (d) Case of two overlapped obstacles. (e) Case of the last active obstacle avoided by a turn. In the notation $[t_{i,k}]$ (and in $[t_{i,k}, t_{i,k+1}]$), $t_{i,k}$ indicates the bypassing direction of obstacle Ω_k on γ_i.

Algorithm 1 Build γ_{i_H}

Require: starting point A_i, ending point B_i, buffer obstacle Ω_{bi}, obstacles $\{\Omega_j\}_{j=1,\cdots,M}$ and the values of decision variables $\{(s_{ij},t_{ij})\}_{j\in J}, J \subset \{1,\cdots,M\}$

1: select the active obstacles to be avoided by a turn ($s_{ij}=1, t_{ij}=0$ or 1), except the buffer obstacle
2: compute the tangent point on the buffer obstacle
3: compute the segment connecting A_i to the tangent point on the buffer obstacle
4: **for** each pair of two successive active obstacles Ω_k and Ω_{k+1} which are avoided by turns **do**
5: compute the tangent points on Ω_k and Ω_{k+1}
6: compute the arc on Ω_k connecting the two tangent points on it ▷ the first (respectively, second) tangent point is computed when considering Ω_{k-1} and Ω_k (respectively, Ω_k and Ω_{k+1})
7: compute the segment connecting the two tangent points when considering Ω_k and Ω_{k+1}
8: **end for**
9: compute the tangent point on the last active obstacle avoided by a turn
10: compute the segment connecting the tangent point on the last active obstacle to B_i
11: **return** γ_{i_H}

4.1.2 Building a vertical cone γ_{i_V}

After obtaining γ_{i_H}, the associated vertical profile γ_{i_V} is built according to Algorithm 2. It is in form of a cone bounded by two straight line segments whose slopes are $\alpha_{min,TO}$, $\alpha_{max,TO}$ in a SID case (respectively, $\alpha_{min,LD}$, $\alpha_{max,LD}$ in a STAR case), see Fig. 3.4. For the sake of simplification, in the following, we denote the minimum and maximum slopes as α_{min}, α_{max} respectively.

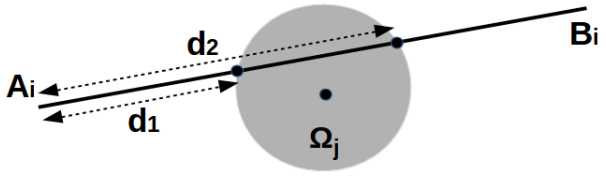


Figure 4.3: Checking intersection between γ_{i_H} and Ω_j in the horizontal plane.

For an active obstacle Ω_j with $t_{ij}=2$, before imposing the level flight, we check whether the cone intersects Ω_j both in the horizontal and vertical plane. We first check the intersection in the horizontal plane, as shown in Fig. 4.3, the length of route sections from A_i to the two intersection points are denoted as d_1 and d_2. When an intersection exists in the horizontal plane, we continue to check in the vertical plane whether the cone intersects Ω_j, as shown in Fig. 4.4. If neither of the following conditions

$$h_{i_{inf}}(d_1) \geq z_{j_{sup}} \text{ (Fig. 4.4(a))} \tag{4.1}$$

$$h_{i_{sup}}(d_2) \leq z_{j_{inf}} \text{ (Fig. 4.4(b))} \tag{4.2}$$

is satisfied, then the cone intersects the obstacle.

If an intersection occurs both in the horizontal and the vertical planes, then a level flight is imposed under the obstacle. Moreover, the lower and upper bounds of a cone are computed separately. For each bound, the level flight starts at the point where the altitude reaches $z_{j_{inf}}$, and ends at the point where the corresponding length in the horizontal plane is d_2. Afterwards, the upper bound (respectively, the lower bound) continues to ascend with the slope α_{max} (respectively, α_{min}). Figure. 4.5 illustrates different cases of imposing a level flight. Note that if some active obstacle with $t_{ij}=2$ is not intersected by the cone associated with the horizontal route, then the corresponding

Algorithm 2 Build γ_{i_V}

Require: the corresponding horizontal curve γ_{i_H}, altitude of the starting point H_{A_i}, obstacles $\{\Omega_j\}_{j=1,\cdots,M}$ and the values of decision variables$\{(s_{ij},t_{ij})\}_{j\in J}, J \subset \{1,\cdots,M\}$

```
for each active obstacle avoided by imposing a level flight do
    check intersection in the horizontal plane
    if horizontal intersection exists then
        compute the length of the route section from A_i to the two intersection points in the horizontal plane, denoted as d_1 and d_2
        check intersection in the vertical plane at d_1 and d_2
        if vertical intersection exists then
            if this is the first level flight then
                compute the vertical profiles of the upper and lower bounds from the starting point A_i to the beginning of the level flight, taking into account α_max and α_min respectively
                impose level flight under the obstacle
            else
                compute the vertical profiles of the upper and lower bounds from the ending point of the previous level flight to the beginning of the current level flight, taking into account α_max and α_min respectively
                impose level flight under the obstacle
                if this is the last level flight then
                    compute the vertical profiles of the upper and lower bounds from the ending point of the current level flight to the ending point B_i, taking into account α_max and α_min respectively
                end if
            end if
        else
            break
        end if
    else
        break
    end if
end for
return γ_{i_V}
```

level flight cannot be imposed. In such a case, we define that the route and the level flight are *infeasible*, regarding to our definition of "active obstacle". After building the vertical cone γ_{i_V}, the value of $\ell_{i_{LF}}$, defined in Subsection 3.3.3, is computed by summing up the lengths of route sections corresponding to level flights projected on the horizontal plane.

Examples of building a route

To illustrate the route computation, let us consider an example with two obstacles and no buffer obstacle. Since there is only one route to build, for the sake of simplification, we omit the index i in s_{ij} and t_{ij}. We denote (s_1,t_1) (respectively, (s_2,t_2)) the variables associated with obstacle Ω_1 (respectively, Ω_2). In the first case (Figs. 4.6(a), 4.6(b)), $(s_1,t_1) = (1,0), (s_2,t_2) = (1,1)$, the horizontal curve is composed by five parts: three segments and two arcs of circles. The three segments are used to connect tangentially the starting point and Ω_1, to connect Ω_1 and Ω_2, and to connect Ω_2

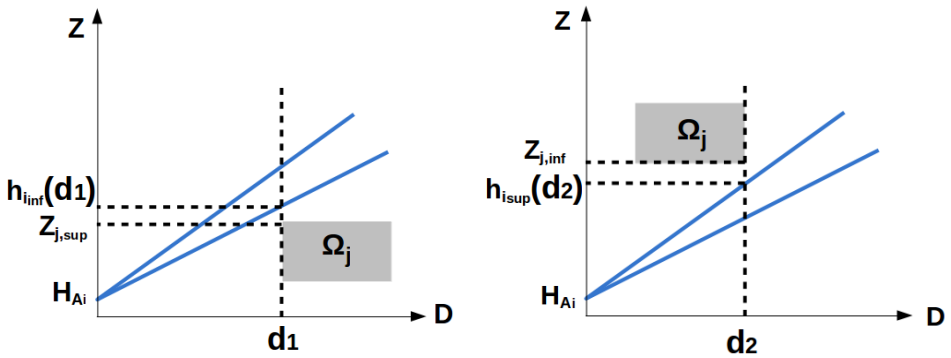


Figure 4.4: Checking intersection between γ_{iv} and Ω_j in the vertical plane. (a) Checking intersection at d_1. (b) Checking intersection at d_2.

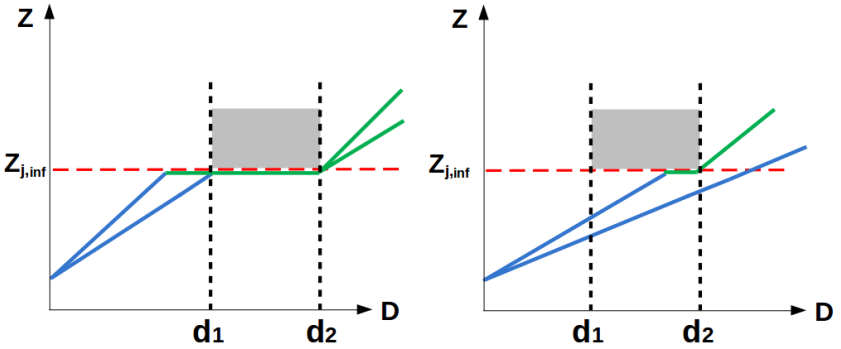


Figure 4.5: Imposing level flight under obstacle. (a) Imposing level flight to both the lower and the upper bounds of the cone. (b) Imposing level flight only to the upper bound of the cone.

and the ending point respectively. The two arcs are used to bypass Ω_1 counter-clockwise and Ω_2 clockwise respectively. In the second case (Figs. 4.6(c), 4.6(d)), $(s_1,t_1) = (1,2), (s_2,t_2) = (1,1)$, the horizontal route is constructed by only bypassing Ω_2, thus it is composed by two segments and one arc of circle. In the vertical plane, when the route reaches the altitude of the lower basis of Ω_1, a level flight is imposed. The level flight ends at the flown distance where the horizontal route passes the border of Ω_1. Afterwards, the lower and upper bound of the vertical cone continue to ascend.

4.2 Designing one optimal route using Branch and Bound (B&B)

To design one optimal route, we apply a Branch and Bound (B&B) method. It is a well known exact method to solve discrete and combinatorial optimization problems [27] [28]. The application of the B&B method is inspired by the approach proposed in [74], where a path planning problem avoiding circular obstacles is studied in 2D. In [74], they propose a branching strategy, where for each obstacle two branches are created depending on the clockwise or counter-clockwise obstacle bypassing. We extend this branching strategy in order to take into account the specificity of our problem, where obstacles can be avoided also by imposing a level flight below the obstacle. The pseudo-code of the B&B method applied in this thesis for generating one optimal route γ_i is pre-

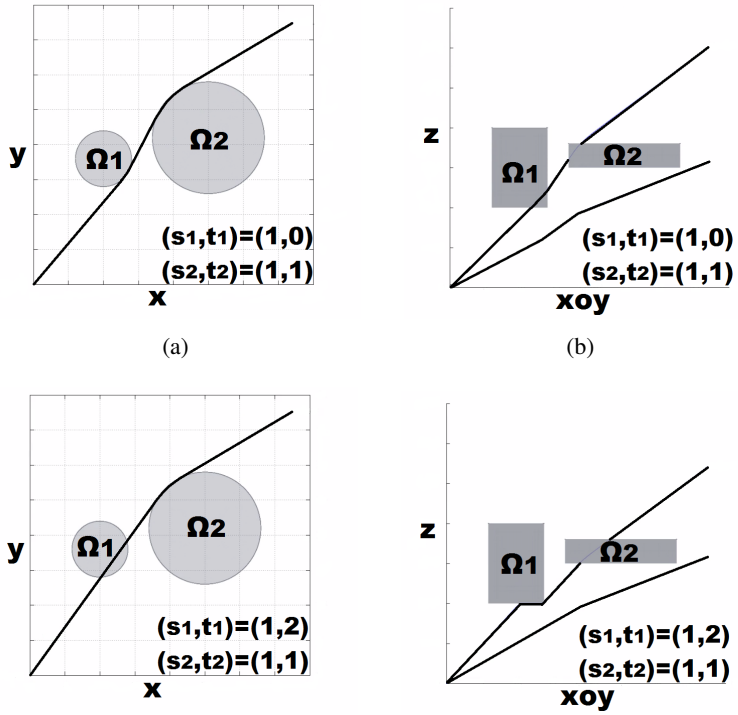


Figure 4.6: Routes construction. (a) Case 1, horizontal plane. (b) Case 1, vertical plane. (c) Case 2, horizontal Plan. (d) Case 2, vertical Plan. Note that in (b) and (d), slopes appear discontinuous as an effect of a projection of a 3D image on a plane.

sented in Algorithm 3. In the following, we explain in more detail our branching strategy and the way the lower bound of a subproblem is computed in the B&B tree.

4.2.1 Branching strategy

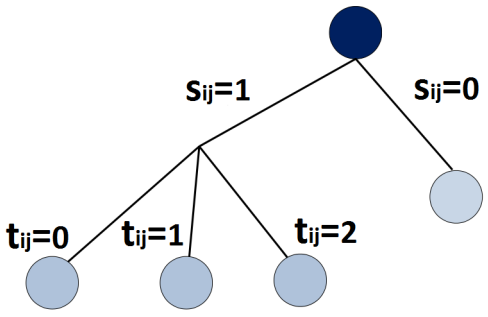


Figure 4.7: Branch and Bound branching strategy in our method.

Our proposed branching strategy is illustrated in Fig. 4.7. When designing route γ_i, for an obstacle Ω_j, we start by setting it as active ($s_{ij} = 1$) or not ($s_{ij} = 0$); when it is active, we develop three branches in order to account for the 3 possibilities to avoid it: counter-clockwise ($t_{ij} = 0$),

Algorithm 3 Generation of one route: Branch and Bound

Require: starting and ending points (A_i, B_i), the altitude of the starting point H_{A_i}, buffer obstacle Ω_{bi}, obstacles $\Omega_j, j = 1, \cdots, M$

1: Initialize: subproblems list $SPL = \emptyset$, current best length $L_{CB} = +\infty$
2: Build the direct route connecting A_i and B_i, denoted as γ_i^0
3: **if** γ_i^0 satisfies all constraints **then**
4: γ_i^0 is the optimal route, $\gamma_i := \gamma_i^0$, $L_{CB} := L_{\gamma_i^0}$
5: **else**
6: Develop new branches according to Fig. 4.7, add them to *SPL*
7: **end if**
8: **while** $SPL \neq \emptyset$ **do**
9: Select 1 subproblem SP ($\{(s_{ij}, t_{ij})\}_{j\in J}, J \subset \{1, \cdots, M\}$) in *SPL* and remove it from *SPL*
10: Build the corresponding horizontal curves γ_{SP_H} and $\tilde{\gamma}_{SP_H}$ ▷ call Algorithm 1(A_i, B_i, Ω_{bi}, $\{\Omega_j\}_{j=1,\cdots,M}$, $\{(s_{ij}, t_{ij})\}_{j\in J}$)
11: Build the corresponding vertical cones γ_{SP_V} ▷ call Algorithm 2(γ_{SP_H}, H_{A_i}, $\{\Omega_j\}_{j=1,\cdots,M}$, $\{(s_{ij}, t_{ij})\}_{j\in J}$)
12: Compute the lower bound LB_{SP} according to Equation (4.4)
13: **if** $LB_{SP} < L_{CB}$ **then**
14: **if** γ_{SP} satisfies all constraints and all level flights of *SP* are feasible **then**
15: Compute $L_{\gamma_{SP}}$, the value of the objective function corresponding to γ_{SP}, according to Equation (3.14)
16: **if** $L_{\gamma_{SP}} < L_{CB}$ **then**
17: $\gamma_i := \gamma_{SP}$, $L_{CB} := L_{\gamma_{SP}}$
18: **else if** *SP* has remaining obstacle to be branched on **then**
19: Develop new branches according to Fig. 4.7, add them to *SPL*
20: **end if**
21: **else if** None of the imposed level of *SP* is definitely infeasible and *SP* has remaining obstacle to be branched on **then**
22: Develop new branches according to Fig. 4.7, add them to *SPL*
23: **end if**
24: **end if**
25: **end while**
26: **return** γ_i and L_{CB}

clockwise ($t_{ij} = 1$) or imposing a level flight ($t_{ij} = 2$). In the exploration of the B&B tree, there are several exceptions and remarks related to the branches corresponding to level flights that need to be explained.

There exist two exceptions where the branch corresponding to a level flight is not developed. The first case is when the altitude of the lower basis of an obstacle to be branched on is lower than H_{min}, since the corresponding level flight does not satisfy the constraint on the minimum altitude of a level flight, and therefore the corresponding route will not be accepted as a solution. The second case is when the number of the imposed level flights in the father branch is already equal to N_{max}. Imposing an additional level flight in a son branch leads to the number of level flights exceeding N_{max}, thus the corresponding route will not be accepted as a solution.

Note that in step 14 of Algorithm 3, a level flight is not always feasible, as mentioned in Section 4.1. However, with further branching, the level flight may become feasible since the form of the horizontal curve changes, and the altitude of the vertical cone when reaching this obstacle changes

accordingly. An example is illustrated in Fig. 4.8. In the father branch, Ω_1 is active and avoided by a clockwise bypassing, and Ω_2 is active associated with a level flight. However, the level flight is not feasible since the corresponding route has no intersection with Ω_2. While with further branching by setting Ω_3 as active avoided by a counter-clockwise turn, the corresponding horizontal curve intersects Ω_2 in the horizontal plane. If Ω_2 is also intersected in the vertical plane, then the level flight under Ω_2 becomes feasible.

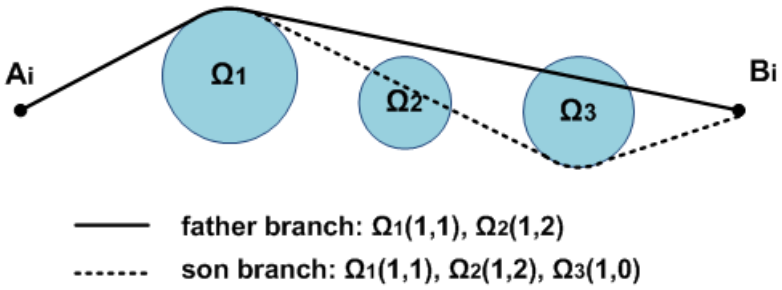


Figure 4.8: A level flight becomes feasible in further branching.

We also remark that there exists one case where an imposed level flight is definitely unfeasible, that is when an obstacle supposed to be avoided by level flight is never intersected by a horizontal curve. It is the case when the form of a portion of the horizontal curve is unchangeable in further branching, as expressed for example in Property. 4.2.1.

Property 4.2.1. *Let SP* $(\{(s_{ij},t_{ij})\}_{j\in J}, J \subset \{1,\cdots,M\})$ *be a subproblem when designing* γ_i*. Assume that there exist* $l,j,k \in \{1,\cdots,M\}$*, such that*

$$\begin{aligned}
&j < l < k \\
&\{j,\cdots,k\} \subset J \\
&s_{ij} = 1 \text{ and } t_{ij} \neq 2 \\
&s_{ik} = 1 \text{ and } t_{ik} \neq 2 \\
&\forall p \in \{j+1,\cdots,k-1\} \setminus l, s_{ip} = 0 \text{ or } (s_{ip},t_{ip}) = (1,2) \\
&(s_{il},t_{il}) = (1,2)
\end{aligned} \tag{4.3}$$

if the segment connecting Ω_j *and* Ω_k *does not intersect obstacle* Ω_l*, then the level flight below* Ω_l *is definitely unfeasible.*

An example illustrating Property. 4.2.1 is presented in Fig. 4.9. Obstacles $\{\Omega_i | i \in \{j,\cdots,k\}\}$ are all active, where Ω_j and Ω_k are avoided by clockwise and counter-clockwise turns respectively. Since the obstacles whose indices are between j and k are supposed to be avoided by level flights, the form of the curve section between Ω_j and Ω_k is fixed and unchangeable in further branching. The level flight below Ω_l is definitely unfeasible, because Ω_l is never intersected by the curve section between Ω_j and Ω_k.

Once a level flight in the subproblem is definitely unfeasible, no further branching is needed. Otherwise, if there is still remaining non-considered obstacle, further branches are developed (step 21 of Algorithm 3).

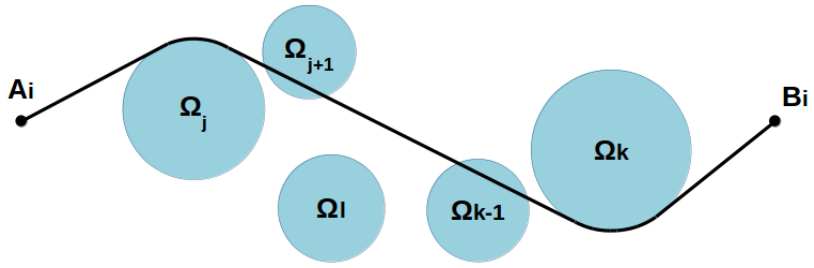


Figure 4.9: An example of a definitely unfeasible level flight.

4.2.2 Lower bound computation

Let SP be a subproblem in the Branch and Bound process for the generation of γ_i, where only some of the values of the decision variables are instantiated ($\{(s_{ij},t_{ij})\}_{j\in J}, J \subset \{1,\cdots,M\}$). The values of the decision variables that are not yet determined are set as non-active ($\{(s_{ij},t_{ij}) = (0,0)\}_{j\in\{1,\cdots,M\}\setminus J}$). The corresponding route γ_{SP}, composed by a horizontal curve γ_{SP_H} and a vertical cone γ_{SP_V}, is first computed according to the previously explained Algorithm 1 and Algorithm 2 respectively. Then we check if γ_{SP} bypasses an active obstacle by an arc with a central angle larger than 180°. If this is the case, we recompute a route $\tilde{\gamma}_{SP}$, by setting the concerned obstacles as non-active. The lower bound corresponding to SP, denoted as LB_{SP}, is computed by:

$$LB_{SP} = c_1 L_{\tilde{\gamma}_{SP_H}} + c_2 L_{min} \sum_{j=1}^{M} \max(t_{ij} - 1, 0) \tag{4.4}$$

where $\tilde{\gamma}_{SP_H}$ is the horizontal profile of $\tilde{\gamma}_{SP}$. The obtained lower bound is then used to identify whether a branch requires further subdivisions. Once the lower bound of a subproblem is greater than the current best value, then no more branch is developed based on this subproblem.

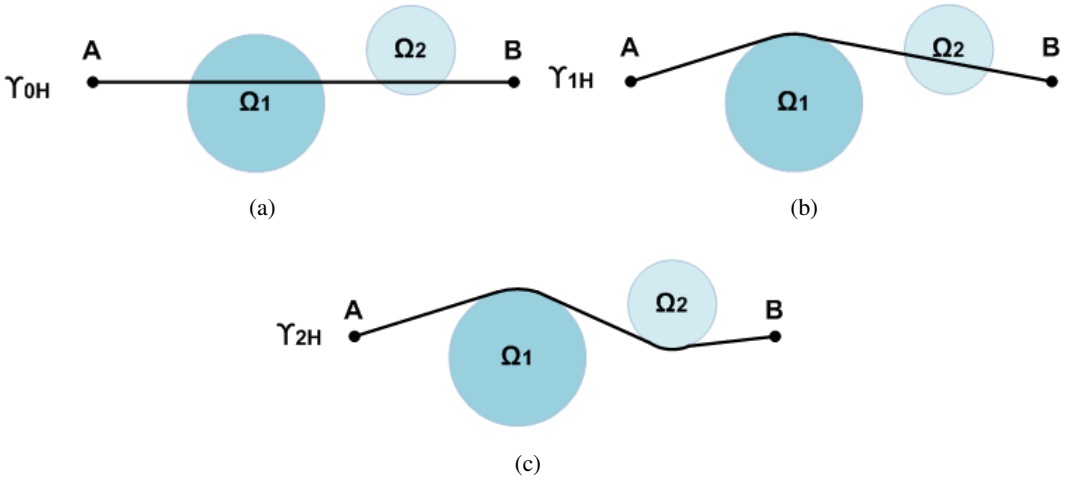
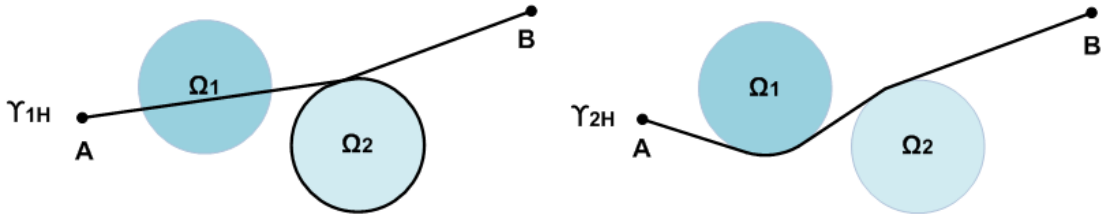


Figure 4.10: General case of computing the lower bound in the horizontal plane. (a) Illustration of γ_{0_H}. (b) Illustration of γ_{1_H}. (c) Illustration of γ_{2_H}.

In the horizontal plane, it can be proven that, in a general case, the length of a horizontal curve increases when taking into account an additional active obstacle avoided by a turn. An example is illustrated in Fig. 4.10, when setting consecutively obstacles Ω_1 and Ω_2 as active obstacles avoided



(a) (b)

Figure 4.11: An exception of computing the lower bound in the horizontal plane. (a) In the father branch, the arc on Ω_2 corresponds to a central angle greater than 180°. (b) In the son branch, the arc on Ω_2 corresponds to a central angle lower than 180°.

by turns, the length of the horizontal curves increase, that is $L_{\gamma_{0_H}} < L_{\gamma_{1_H}} < L_{\gamma_{2_H}}$. However, there is an exception, when the arc on an active obstacle corresponds to a central angle greater than 180°. An example is shown in Fig. 4.11. Initially, in the father branch, only Ω_2 is active and avoided by a clockwise turn, the arc on Ω_1 corresponds to a central angle greater than 180°, as shown in Fig. 4.11(a). Afterwards, in the son branch, when Ω_1 is set as active and avoided by a counter-clockwise turn, the form of arc on Ω_2 is changed whose length is shorter than in the previous case, as shown in Fig. 4.11(b). This kind of situation is due to the way a horizontal curve is built with respect to the increasing order of the obstacles numbering. In order to take into account this situation, we compute the lower bound by taking the active obstacle bypassed by an arc with a central angle greater than 180° as non-active.

In the vertical plane, the length of a level flight route section depends on the form of the corresponding horizontal curve. However, the length of each level flight is always greater than L_{min}. Thus, $L_{min}\sum_{j=1}^{M}\max(t_{ij}-1,0)$ is a lower bound of the length of level flight route sections.

4.2.3 B&B tree exploration strategies

The computing time for obtaining the global optimal solution in the B&B method depends not only on the complexity of the problem but also on the way the B&B tree is explored [29] [30]. A good B&B tree exploration strategy enumerates fewer tree branches before obtaining the global optimum. In the following, the B&B tree exploration strategies applied in this thesis are presented.

In step 9 of Algorithm 3, we can select the next subproblem with respect to different criteria, including:

- *best route length (BST)*: select the subproblem with the shortest route length;
- *most considered obstacles (MCO)*: select the subproblem with the most considered obstacles;
- *least considered obstacles (LCO)*: select the subproblem with the least considered obstacles.

In the case when an obstacle Ω_j is intersected by a route γ_i, we define the *incursion* length between Ω_j and γ_i as the length of the route section included in the projection of Ω_j in the horizontal plane. In order to select the next obstacle to branch on in steps 6, 19, 22 of Algorithm 3, we also have several possibilities, including:

- *first intersected (FI)*: select the first non-considered intersected obstacle;
- *least incursion (LI)*: select the non-considered intersected obstacle with the smallest incursion length;

- *greatest incursion (GI)*: select the non-considered intersected obstacle with the greatest incursion length.

If there is no obstacle intersected by the route, we choose the non-considered obstacle with the lowest index to branch on. In Section 4.4, these B&B tree exploration strategies are implemented and tested.

4.2.4 Step-by-step illustration

We present a step-by-step illustration (Fig. 4.12) to show how the Branch and Bound method works. The starting and ending points as well as two obstacles Ω_1, Ω_2 are presented in Fig. 4.12(a). Since there is only one route to build, we simply denote s_{ij} as s_j, and denote t_{ij} as t_j in this example. Besides, the buffer obstacle is not introduced. We take $c_1 = 1, c_2 = 1$, that is we penalize the length of level flights in the objective function.

Step 1: We develop 4 branches on Ω_1. We start by deviating the route counter-clockwise, and we obtain a route that does not intersect Ω_2. The lower bound in this case is 54.17 Nm (100320 m). Besides, the value of the objective function associated with solution $(s_1, t_1) = (1,0),\ s_2 = 0$ is equal to the lower bound. Therefore, no further exploration is needed.

Step 2: Another branch on Ω_1 is developed by deviating the direct route clockwise around Ω_1. The length of this horizontal route is the lower bound of this subproblem, the value is greater than the current best value. There is no possibility to get a better solution by further branching on Ω_2, therefore we cut this branch.

Step 3: The third branch on Ω_1 is obtained by imposing a level flight, as shown in Figs. 4.12(d), 4.12(e). The lower bound is greater than the current best value, so the branch is cut.

Step 4: The last branch on Ω_1 is with $s_1 = 0$. The route intersects Ω_2 and its lower bound, which corresponds to the length of the direct route, is less than the current best length, thus 4 branches on Ω_2 are developed. In the case $s_2 = 0$, the corresponding route intersects both Ω_1 and Ω_2, thus is not accepted as a solution.

Step 5: By branching counter-clockwise around Ω_2, the obtained route still intersects Ω_1, thus it is not accepted.

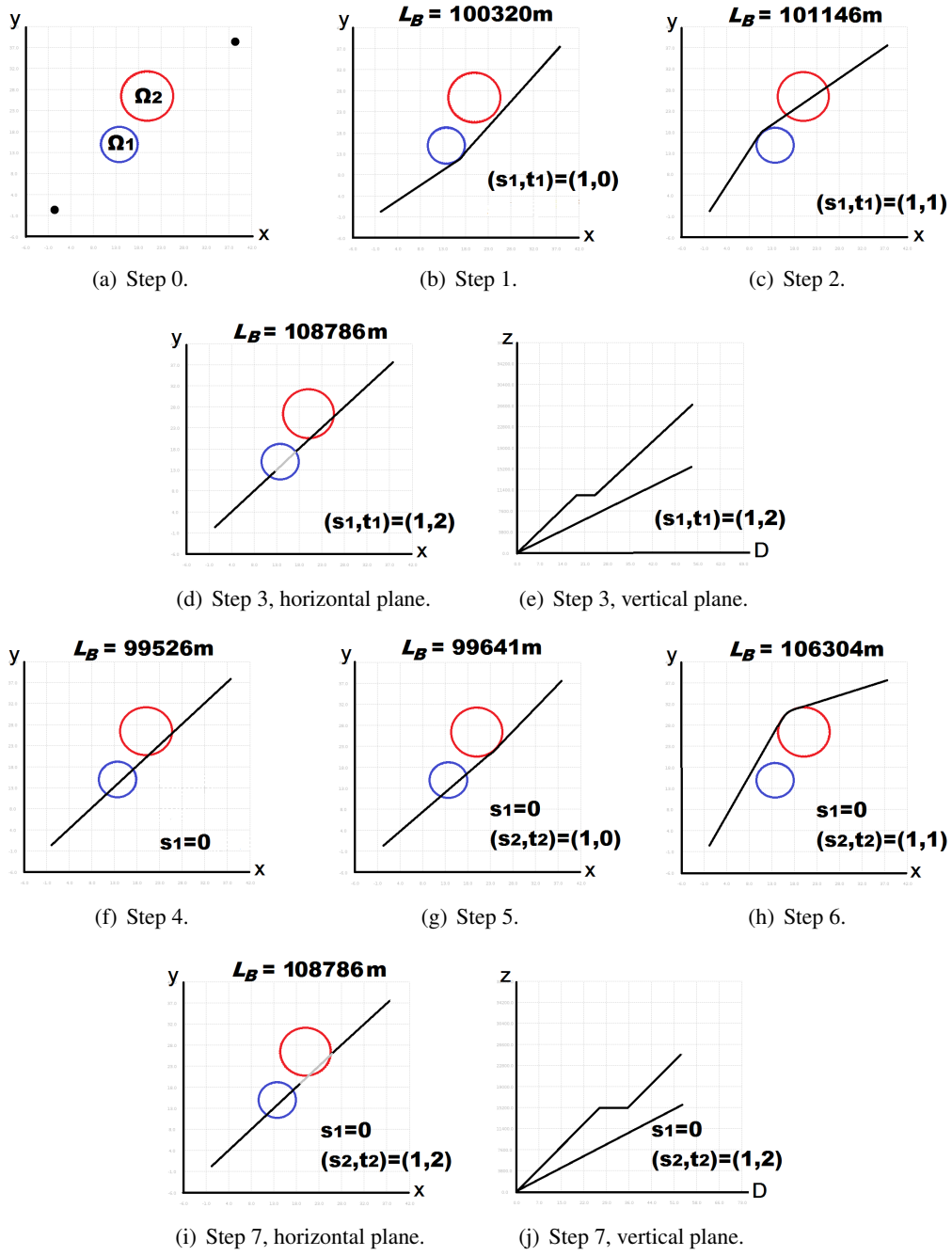
Step 6: By branching clockwise around Ω_2, we obtain a feasible route with length greater than the current best value, so it is not accepted.

Step 7: The last branch is obtained by imposing a level flight under Ω_2 as shown in Figs. 4.12(i), 4.12(j). This route is still encountered by Ω_1, so it is not accepted.

All the possible branches have been considered. The shortest distance is 54.17 Nm (100320 m) and is obtained by taking $(s_1, t_1) = (1,0), s_2 = 0$.

4.3 State space reduction by using convex hull filter

We have 4 possibilities to deal with each obstacle ($(s_{ij}, t_{ij}) = (0,0)$, $(1,0)$, $(1,1)$, or $(1,2)$), thus the maximum size of the state space of our problem is 4^M, where M is the number of obstacles. In order to reduce the size of the state space, we apply a pre-processing technique proposed in [3], namely the *convex hull filter* (CVH-filter). In [3], the authors prove that in an environment where 2D disjoint circular obstacles are present, the shortest path lies in a convex hull of a few circular obstacles around the line segment connecting the starting and ending points. By using this pre-processing technique, the number of the potential obstacles that may have an impact on the form of the shortest path is reduced significantly.



(a) Step 0. (b) Step 1. (c) Step 2.

(d) Step 3, horizontal plane. (e) Step 3, vertical plane.

(f) Step 4. (g) Step 5. (h) Step 6.

(i) Step 7, horizontal plane. (j) Step 7, vertical plane.

Figure 4.12: Branch and Bound illustration.

In order to generate such a convex hull, we start by building the segment connecting the starting and ending points. If this segment has no intersection with any obstacle, the shortest path corresponds to the segment connecting the starting and ending points. Otherwise, the convex hull is defined as the tight envelop including this obstacle and both starting and ending points. Afterwards, the intersections between the boundary of the convex hull and obstacles are checked iteratively. Once an intersected obstacle is found, it is added to the convex hull, and the boundary of the convex hull is recomputed. The algorithm terminates when no more intersection occurs between obstacles

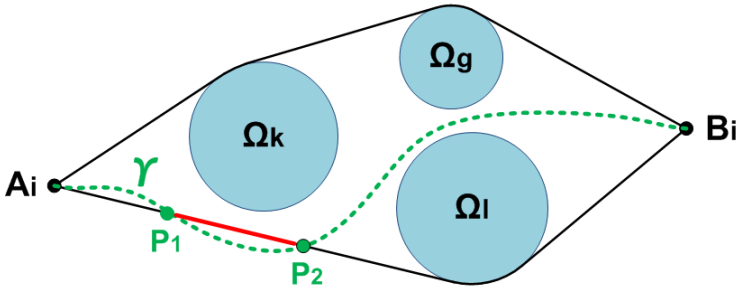


Figure 4.13: An example illustrating that the shortest path lies in the convex hull.

and the boundary of the convex hull. Finally, the obstacles enveloped in the convex hull are kept for the route design, while the others are ignored since they have no impact on the optimal route form. In order to prove that the shortest path lies in the convex hull, we give an example in Fig. 4.13. The convex hull in this example is the one enveloping obstacles Ω_k, Ω_g and Ω_l. For the sake of simplification, the other obstacles which are not included in the convex hull are not shown. Suppose that a shortest path γ contains a section lying out of the convex hull. This leads to two intersections with the convex hull, denoted as P_1 and P_2. The convex hull boundary between P_1 and P_2 (represented in red color in Fig. 4.13) is obviously shorter the curve section on γ between P_1 and P_2. Thus γ can not be the shortest path.

In this thesis, we apply the convex hull filter to pre-process obstacles. The application of this filter is extended by taking into account the specificities of our problem.

- First of all, the shape of an obstacle is modeled as a 3D cylinder, instead of a 2D disk as in [3]. In fact the filter still holds in 3D, since a 3-dimensional feasible route can be indeed built based on the 2D shortest path avoiding the obstacle projections on the plane. Thus in the filter that we propose, we deal with the projections of cylinders in the horizontal plane, which are in form of disks.

- Secondly, the relative position of the projections of two obstacles in the horizontal plane could be externally tangent or overlapped, instead of only disjoint to each other as in [3].

- Thirdly, as we associate a buffer obstacle with given size and bypassing orientation to each route, the route section connecting the starting point to the buffer obstacle is fixed in the design. Therefore, the buffer obstacle is always included in the convex hull computed by the filter.

- Finally, it is possible that the projection of the ending point in the horizontal plane lies in the projection of an obstacle in the horizontal plane, since they may be separated in 3D. We also take this case into account.

4.4 Simulation results

In this section we present some numerical results on the design of one route. The proposed B&B method is tested on several artificially generated problems, where the number and layout of obstacles are different. The number of explored nodes is compared with respect to different B&B tree exploration strategies, and is compared in the cases with or without applying the convex hull filter (denoted as CVH-filter in the following sections). In addition, we also show how the shape of

a route is influenced by the choices of different weight coefficients in the objective function. Tests were run on a Linux platform with a 2.4 GHz processor and 8 GB RAM.

Each test in this section consists of one single route to design. For the sake of simplification, we omit the index i and denote the starting point A_i as A, the ending point B_i as B, and the buffer obstacle $\Omega_{bi}(C_{bi}(x_{bi}, y_{bi}), r_{bi}, z_{bi_{inf}}, z_{bi_{sup}}, t_{\Omega_{bi}})$ as $\Omega_b(C_b(x_b, y_b), r_b, z_{binf}, z_{bsup}, t_{\Omega_b})$. Moreover, in the following sections, the unit of coordinates in x-, y-axis, radius r, and lengths related to routes is in Nm, and the unit of altitudes in z-axis is in ft. The input data related to the routes to design are given in Table 4.1.

minimum radius of a RF leg, R_{min}	5Nm
maximum radius of a RF leg, R_{max}	13Nm
maximum number of level flights on each route, N_{max}	2
minimum altitude of each level flight, H_{min}	3500ft
minimum length of each level flight, L_{min}	5Nm
minimum distance between two successive level flights, D_{min}	5Nm
minimum take-off slopes $\alpha_{min,TO}$	7% ($\sim 4^\circ$)
maximum take-off slopes $\alpha_{max,TO}$	11% ($\sim 6.3^\circ$)

Table 4.1: Input data related to routes to design.

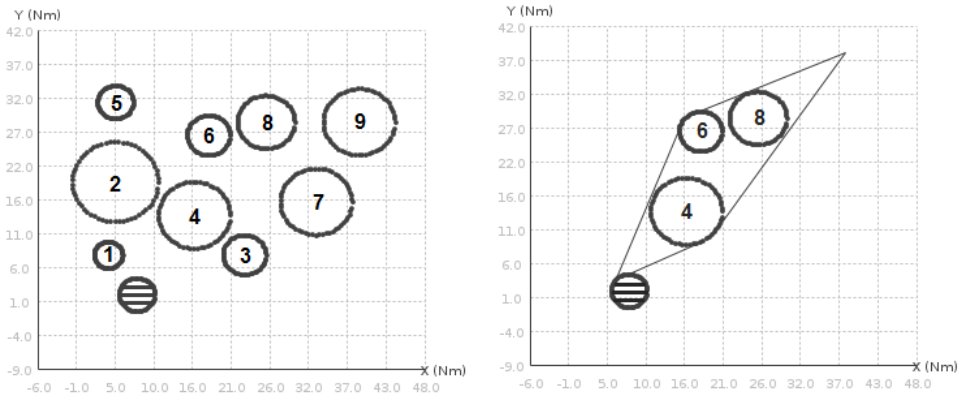
Test 1, generation of 1 SID route with 9 disjoint obstacles ($N = 1$, $M = 9$)

The characteristics of the route to design are given in Table 4.2. The obstacles as well as their indices ordered according to the projection length on line (AB) are illustrated in Fig. 4.14(a), where the striped obstacle is the buffer obstacle. The characteristics related to the obstacles are presented in Table 4.3. After applying the convex hull filter, only 3 obstacles are taken into account for the route design, the other obstacles are discarded. The convex hull formed by these three obstacles are presented in Fig. 4.14(b).

starting point, (x_A, y_A)	(9Nm, -3Nm)
altitude of starting point, H_A	0
ending point, (x_B, y_B)	$(38\text{Nm}, 38\text{Nm})$
buffer obstacle, $(x_b, y_b, r_b, z_{binf}, z_{bsup})$	$(8\text{Nm}, 2\text{Nm}, 2.5\text{Nm}, 0, 50000\text{ft})$
buffer obstacle bypassing direction, t_{Ω_b}	1

Table 4.2: Test 1: characteristics of the route to design.

We first present the simulation results obtained by applying B&B tree exploration strategy "BST" to select the next subproblem, and strategy "FI" to select the obstacle to branch on, as presented in Section 4.2.3. Three cases with different weight coefficients are tested. In the first case, $c_1 = 1$, $c_2 = 0$, the optimal route is illustrated in Figs. 4.15(a) 4.15(b). Without penalizing the length of level flights in the objective function, the optimal route avoid obstacles 4 and 8 by imposing level flights under them. While in the second case, $c_1 = 1$, $c_2 = 0.05$, the length of level flight is penalized with a small weight coefficient. The result, presented in Figs. 4.15(c) 4.15(d), shows that instead of imposing a level flight under obstacle 8, a bypassing in counter-clockwise around it leads



(a) (b)

Figure 4.14: Test 1, obstacles illustration. (a) Buffer obstacle (striped) and 9 obstacles. (b) 3 remaining obstacles after using CVH-filter.

to the best solution. In the third case, $c_1 = 1$, $c_2 = 1$, the length of level flights is further penalized, thus no level flight is imposed on the optimal route, as shown in Figs. 4.15(e) 4.15(f). The optimal route avoids obstacle 4 in counter-clockwise bypassing. The simulation time as well as the number of nodes explored in the B&B tree in each case are compared, depending on whether the CVH-filter is used for pre-processing. The numerical results are presented in Table 4.4. It can be seen that, in the cases when $c_2 \neq 0$, the computation time and the number of iterations are reduced significantly when the CVH-filter is applied.

Index	(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
1	(4, 8)	2	(2300, 7800)
2	(5, 19)	6	(2900, 5600)
3	(23, 8)	3	(7500, 14400)
4	(16, 14)	5	(8800, 10200)
5	(5, 31)	2.5	(7500, 14400)
6	(18, 26)	3	(9300, 17800)
7	(33, 16)	5	(10000, 22000)
8	(26, 28)	4	(12600, 22400)
9	(39, 28)	5	(13630, 26320)

Table 4.3: Test 1: characteristics of the obstacles.

We test also the proposed method with different B&B tree exploration strategies presented in Section 4.2.3, by fixing the weight coefficients of the objective function $c_1 = 1$, $c_2 = 1$, and without using the CVH-filter. The number of nodes explored in the B&B method is presented in Table 4.5, the strategies "LCO" combined with "LI" generate the least number of nodes. However, the strategies "LI" and "GI" require further computation and comparison on the incursion length of intersected obstacles, thus these two strategies usually drive to longer computing time than using the strategy "FI".

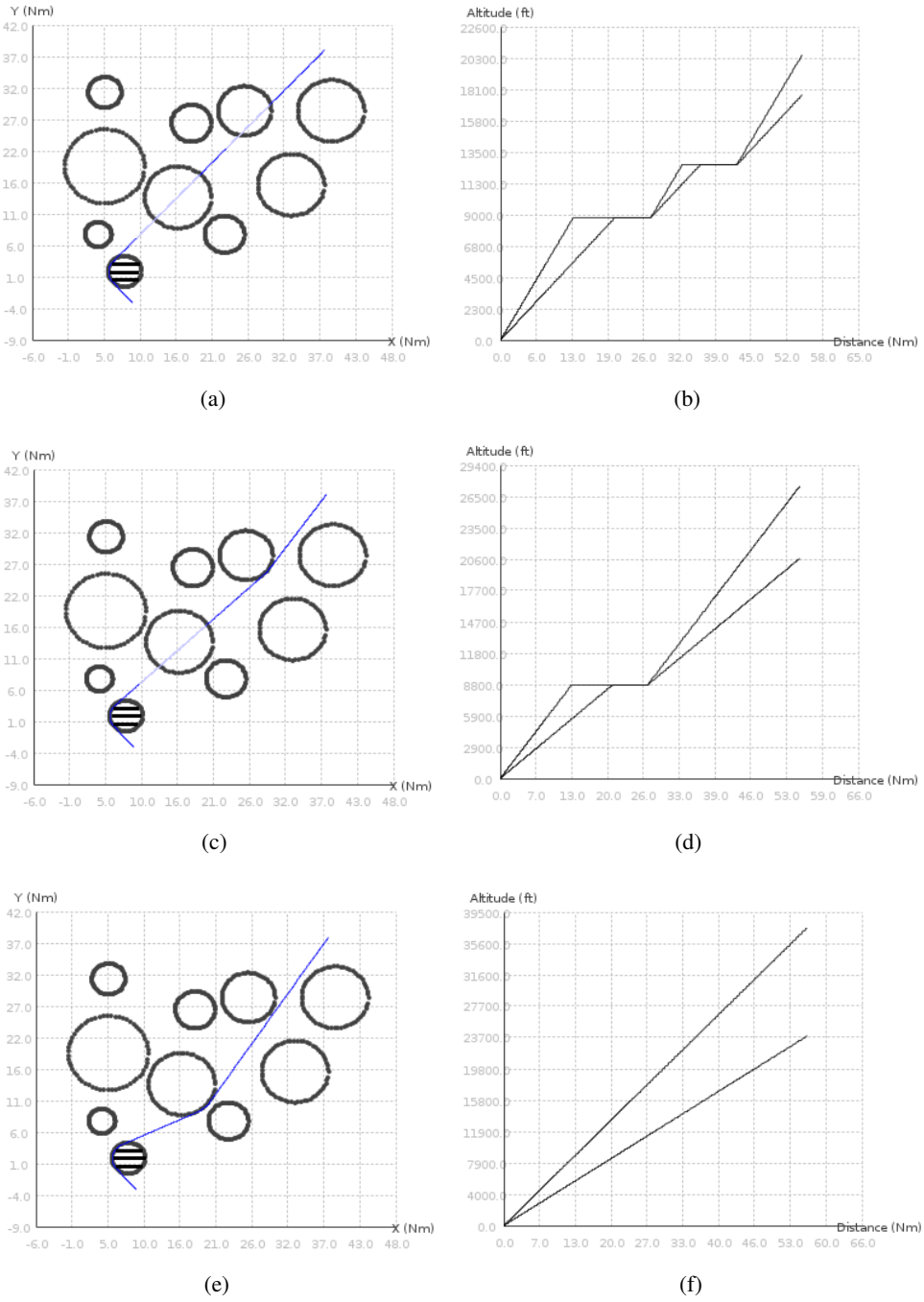


Figure 4.15: Test 1, simulation results. (a) γ_H when $c_1 = 1$, $c_2 = 0$. (b) γ_V when $c_1 = 1$, $c_2 = 0$. (c) γ_H when $c_1 = 1$, $c_2 = 0.05$. (d) when $c_1 = 1$, $c_2 = 0.05$. (e) γ_H when $c_1 = 1$, $c_2 = 1$. (f) γ_V when $c_1 = 1$, $c_2 = 1$.

Test 2, generation of 1 SID route with 40 disjoint obstacles ($N = 1$, $M = 40$)

The characteristics of the route to design are given in Table 4.6. Figure 4.16(a) illustrates the buffer obstacle (striped) and the other obstacles. The CVH-filter reduces the number of potential obstacles from 40 to 8, the corresponding convex hull is shown in Fig. 4.16(b). Different values of

Test 1		$c_1 = 1, c_2 = 0$	$c_1 = 1, c_2 = 0.05$	$c_1 = 1, c_2 = 1$
no CVH-filter	time (s)	0.068	7.22	2.991
	explored nodes	17	4178	1701
with CVH-filter	time (s)	0.082	0.105	0.084
	explored nodes	24	33	28
$\int_0^1 \|\|\gamma'_H(t)\|\|_2 \, dt$		54.88	55.08	56.28
l_{LF}		23.86	13.9	0
L_γ		54.88	55.77	56.28

Table 4.4: Test 1: numerical results.

Test 1		Subproblem selection		
		BST	MCO	LCO
Obstacle selection	FI	1701	1898	1493
	LI	1681	1487	1479
	GI	1701	1898	1493

Table 4.5: Test 1: number of nodes generated in the B&B method under different tree exploration strategies.

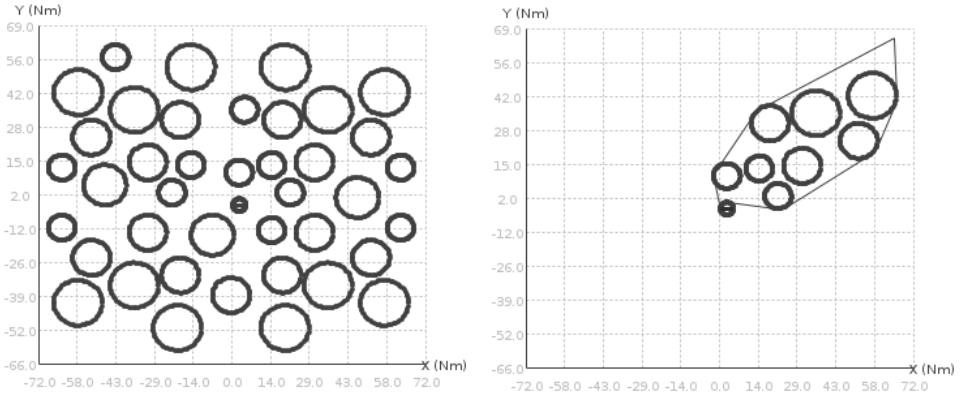
weight coefficients are given: in the first case, $c_1 = 1$, $c_2 = 0$; in the second case, $c_1 = 1$, $c_2 = 1$. The optimal route in each case is presented in Figs. 4.17. Similarly as in test 1, when the length of level flights is not penalized, the algorithm provides a direct route in the horizontal plane, where obstacles are avoided by imposing level flights in the vertical plane. On the contrary, when level flight is not preferred, several turns may be applied for obstacle avoidance. The numerical results using B&B tree exploration strategies "BST" and "FI", also with the application of CVH-filter are presented in Table 4.7. The simulation time, as well as the number of explored nodes, in the first case when $c_2 = 0$, are much less than in the second case when $c_2 = 1$. The reason is that by imposing level flights without any penalty, the B&B algorithm tends to select the subproblems with level flights at first, since they have smaller lower bounds. Thus the optimal solution is found very quickly, and other subproblems with higher lower bounds are discarded. As a result, only a few number of nodes are explored in the B&B tree. On the contrary, when the level flight is penalized, the subproblems avoiding obstacles by counter-clockwise and clockwise bypassing are considered first. Their corresponding routes may intersect other non-considered obstacles. Thus much more nodes are generated in such a way.

starting point, (x_A, y_A)	(4Nm, -8Nm)
altitude of starting point, H_A	0
ending point, (x_B, y_B)	(65Nm, 65Nm)
buffer obstacle, $(x_b, y_b, r_b, z_{binf}, z_{bsup})$	$(3\text{Nm}, -3\text{Nm}, 2.5\text{Nm}, 0, 50000\text{ft})$
buffer obstacle bypassing direction, t_{Ω_b}	1

Table 4.6: Test 2: characteristics of the route to design.

We also provide the result on the number of explored nodes with respect to different tree ex-

ploration strategies, as presented in Table 4.8, where the CVH-filter is applied for pre-processing, and the values of the weight coefficients in the objective function is fixed as $c_1 = 1$, $c_2 = 1$. the strategies "LCO" combined with "LI" generate the least number of nodes. Moreover, the strategy "LCO" performs better compared with strategies "BST" or "MCO" in this example.



(a) (b)

Figure 4.16: Test 2, obstacles illustration. (a) Buffer obstacle (striped) and 40 input obstacles. (b) 8 remaining obstacles after using CVH-filter.

Test 2		$c_1 = 1, c_2 = 0$	$c_1 = 1, c_2 = 1$
with CVH-filter	time (s)	0.093	4.7
	explored nodes	18	1746
$\int_0^1 \|\gamma_H'(t)\|_2 \, dt$		100.12	103.28
l_{LF}		44.43	0
L_γ		100.12	103.28

Table 4.7: Test 2: numerical results.

Test 2		Subproblem selection		
		BST	MCO	LCO
Obstacle selection	FI	1746	3880	1441
	LI	1725	4137	1423
	GI	1897	3115	1534

Table 4.8: Test 2: number of nodes generated in the B&B method under different tree exploration strategies.

Test 3, generation of 1 SID route with 18 overlapped obstacles ($N = 1$, $M = 18$)

The starting and ending points, as well as the buffer obstacle, are the same as in test 2 (Table 4.6). The characteristics of the 18 obstacles are given in Table 4.9. An illustration of the layout of the obstacles is presented in Fig. 4.18(a). By using the CVH-filter, the number of potential obstacles is reduced to 10, as shown in Fig. 4.18(b).

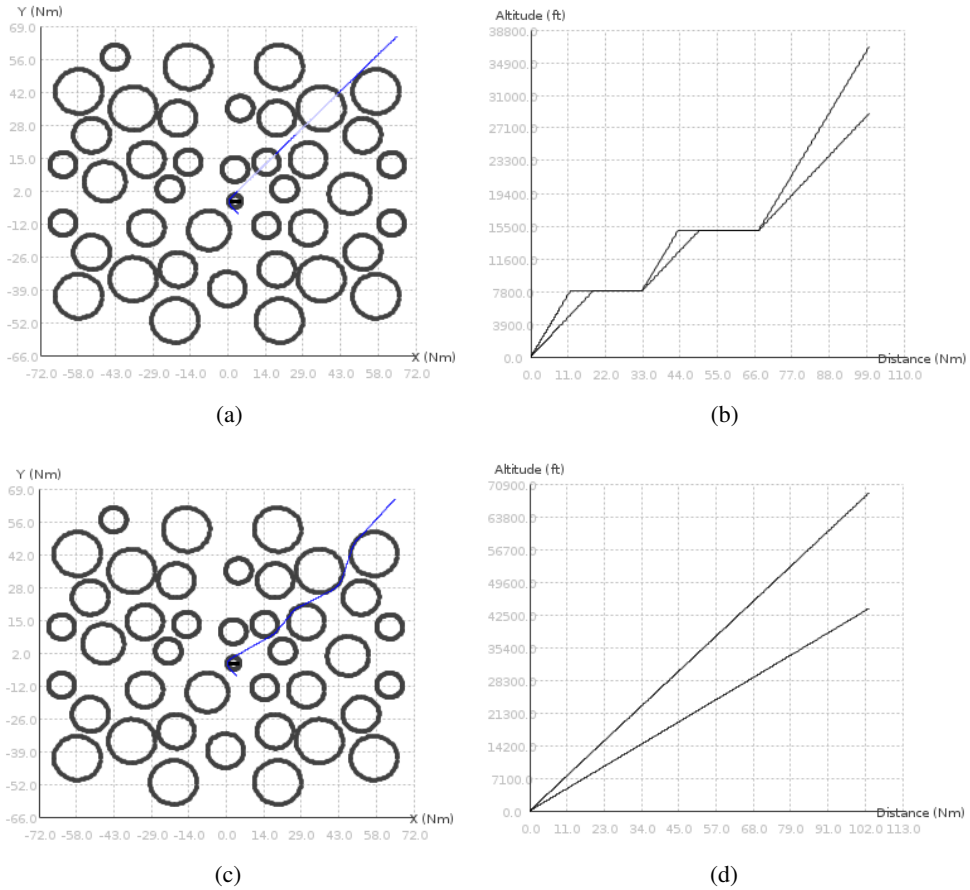


Figure 4.17: Test 2, simulation results. (a) γ_H when $c_1 = 1, c_2 = 0$. (b) γ_V when $c_1 = 1, c_2 = 0$. (c) γ_H when $c_1 = 1, c_2 = 1$. (d) γ_V when $c_1 = 1, c_2 = 1$.

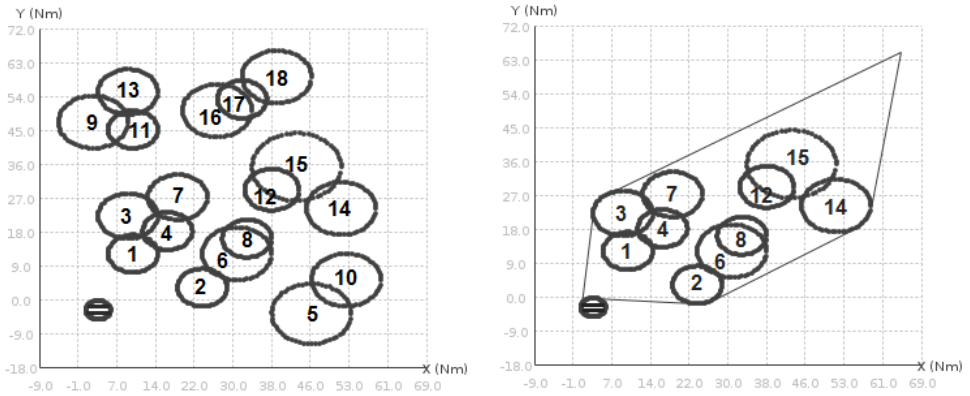


Figure 4.18: Test 3, obstacles illustration. (a) Buffer obstacle (striped) and 18 input obstacles. (b) 10 remaining obstacles after using CVH-filter.

We test two different cases, under the B&B tree exploration strategies "BST" combined with "FI", by varying the value of c_2. When $c_1 = 1$, $c_2 = 0$, the length of level flights is not penalized.

Index	(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
1	(10, 12)	5	(2300, 17800)
2	(24, 3)	5	(7500, 14400)
3	(9, 22)	6	(2900, 5600)
4	(17, 18)	5	(8800, 30000)
5	(46, -4)	8	(0, 50000)
6	(31, 12)	7	(11000, 50000)
7	(19, 27)	6	(9300, 17800)
8	(33, 16)	5	(10000, 22000)
9	(2, 47)	7	(0, 50000)
10	(53, 5)	7	(0, 50000)
11	(10, 45)	5	(0, 50000)
12	(38, 29)	5.5	(0, 36320)
13	(9, 55)	6	(0, 50000)
14	(52, 24)	7	(23000, 50000)
15	(43, 35)	9	(17400, 42400)
16	(27, 50)	7	(0, 50000)
17	(32, 53)	5	(7500, 34400)
18	(39, 59)	7	(0, 50000)

Table 4.9: Test 3: characteristics of the obstacles.

However, by setting $N_{max} = 2$, the number of level flight is limited to 2. Two level flights are imposed under obstacles 4 and 15. To avoid obstacles 1 and 12, counter-clockwise and clockwise bypassing are selected respectively. The simulation result is shown in Figs. 4.19(a) 4.19(b). While when $c_1 = 1$, $c_2 = 1$, only bypassing strategies are applied to avoid obstacles, as illustrated in Figs. 4.19(c) 4.19(d). The computing time, number of explored nodes and the route length are presented in Table 4.10. The simulation time and the number of explored nodes, in the first case when $c_2 = 0$, are larger than in the second case when $c_2 = 1$. The reason is that, in the first case, even though the subproblems using level flights to avoid obstacles are considered first, the constraint on $N_{max} = 2$ drives the algorithm to search for a solution where obstacles are avoided not only by level flights but also by counter-clockwise and clockwise bypassing. Thus more B&B nodes are generated in such a process.

The tests are run under different B&B tree exploration strategies, by fixing $c_1 = 1$, $c_2 = 1$ and by using the CVH-filter for pre-processing. The number of explored nodes in each case is presented in Table 4.11, and the strategy "BST" combined with strategies "LI"and "GI" offer the least nodes. The performance of strategy "BST" is better than "MCO" and "LCO" in this example.

Test 3		$c_1 = 1, c_2 = 0$	$c_1 = 1, c_2 = 1$
with CVH-filter	time (s)	11.07	5.17
	explored nodes	5522	2587
$\int_0^1 \|\gamma'_H(t)\|_2 \, dt$		100.34	101.38
l_{LF}		44.65	0
L_γ		100.34	101.38

Table 4.10: Test 3: numerical results.

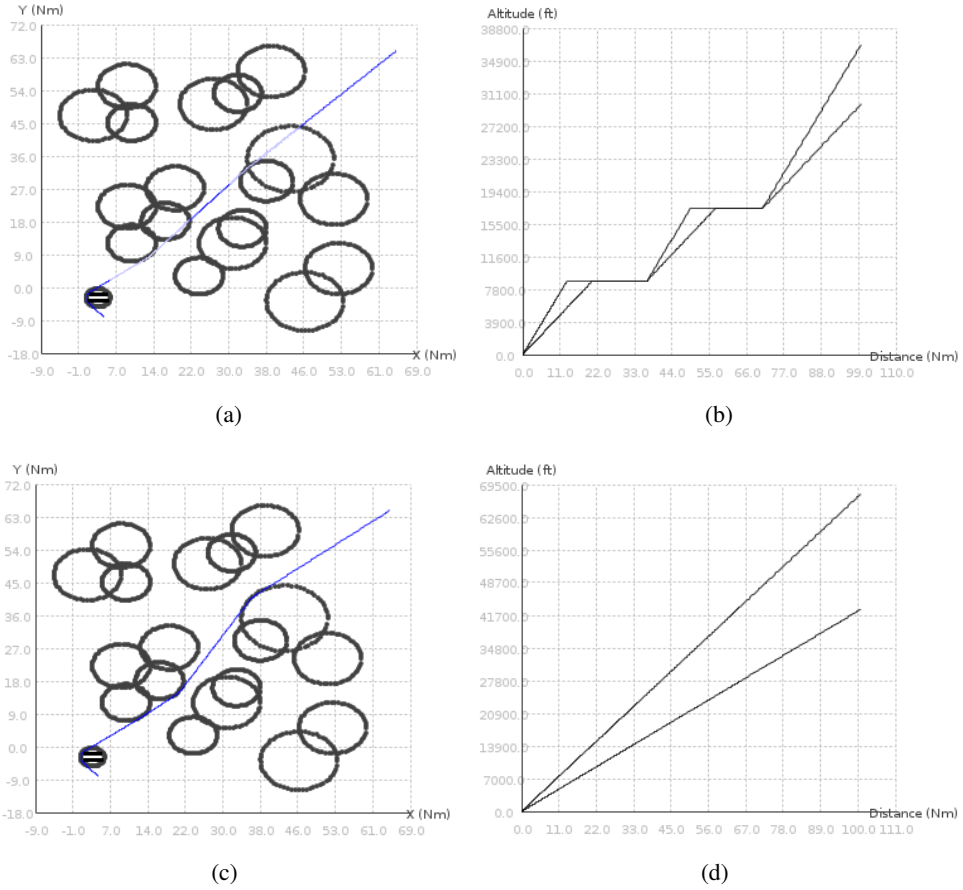


Figure 4.19: Test 3, simulation results. (a) γ_H when $c_1 = 1, c_2 = 0$. (b) γ_V when $c_1 = 1, c_2 = 0$. (c) γ_H when $c_1 = 1, c_2 = 1$. (d) γ_V when $c_1 = 1, c_2 = 1$.

Test 3		Subproblem selection		
		BST	MCO	LCO
Obstacle selection	FI	2587	10385	4042
	LI	2516	9925	3928
	GI	2516	4721	2352

Table 4.11: Test 3: number of nodes generated in the B&B method under different tree exploration strategies.

4.5 Conclusion

In this chapter, the problem of building one single route was addressed. We first presented the way a unique route is built corresponding to the values of some given decision variables. Then the Branch and Bound (B&B) method to design one optimal route was presented. The proposed method was validated on several artificially generated tests, where various numbers and layouts of obstacles were given. The obtained routes can be continuous and smooth, which are available for Continuous Climb Operation (CCO) [105] and Continuous Descent Operation (CDO) [106]. The convex hull

filter shows its effectiveness to reduce the dimension of the state space. Moreover, the combination of the B&B tree exploration strategies “BST” and “FI” has a relatively good and stable performance in terms of the number of explored nodes and the computing time in the tested problems. Thus we will use them in the rest of our simulation tests.

Chapter 5

Designing Multiple Routes

In this chapter, the problem of designing multiple routes is addressed, where the constraint on routes separation needs to be considered additionally. We recall that two routes are in *conflict* if they violate the minimum separation norm (3Nm in the horizontal plane, or 1000ft in the vertical plane). We present two different approaches to solve this problem. The first one is a B&B-based approach, where routes are generated sequentially according to a fixed order (*e.g.*, the decreasing order of the traffic load on each route). Since the quality of a solution depends on the order of routes generation, we develop another resolution approach, the Simulated Annealing (SA) method, where routes are generated simultaneously using a global strategy. One important task in both methods is the conflict detection, so we start this chapter by presenting an appropriate and efficient method to detect conflicts between pairwise 3D routes.

5.1 Conflict detection method

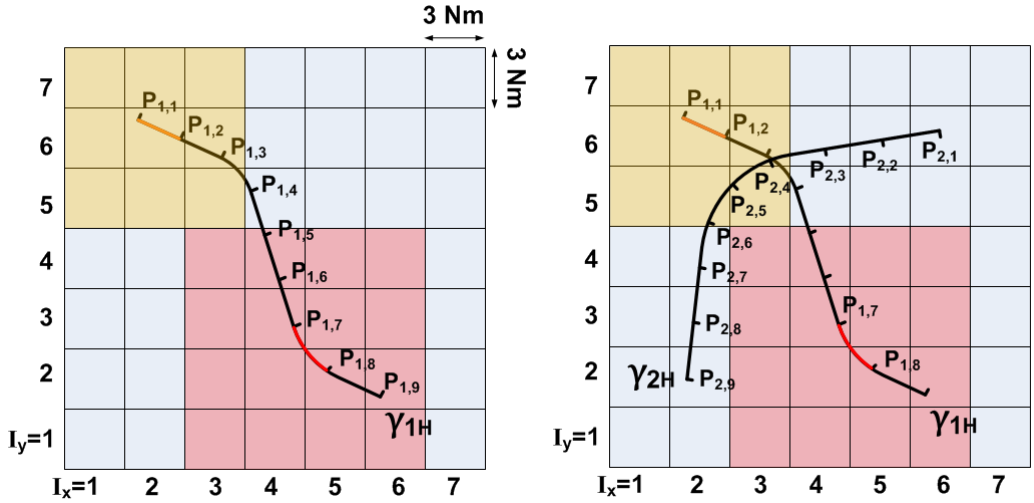
Detecting conflicts between pairwise 3D routes, especially when the routes vertical profiles are cones instead of curves, is not an easy problem. We propose a two-steps scheme to deal with this problem. First we detect in the horizontal plane whether pairs of routes lose the 3Nm separation. Then, for the detected route sections in a loss of horizontal separation, we evaluate whether they are separated in the vertical plane. If a route section loses both horizontal and vertical separations, then it is in conflict.

5.1.1 Horizontal detection

Route projections in the horizontal plane $\gamma_{iH}, i = 1, \cdots, N$ are in the form of 2D-curves. We propose a 2D-grid which covers all the horizontal curves. Such idea of using a 2D-grid for conflict detection has been applied in [84]. The dimension of a cell on the 2D-grid is 3Nm$\times$3Nm, as defined by the horizontal separation norm. Each cell is identified by two indices (I_x, I_y), where I_x is the index on the x-axis and I_y is the index on the y-axis, where $I_x, I_y \in \mathbb{N}$. We also define the following terms:

- $x_{inf}(I_x, I_y)$, the x-coordinate of the left edge of cell (I_x, I_y);
- $x_{sup}(I_x, I_y)$, the x-coordinate of the right edge of cell (I_x, I_y);
- $y_{inf}(I_x, I_y)$, the y-coordinate of the bottom edge of cell (I_x, I_y);
- $y_{sup}(I_x, I_y)$, the y-coordinate of the top edge of cell (I_x, I_y).

Each horizontal curve is discretized with a discretization step δt. A post-processing to add discretization points is applied, so that each time the curve passes through a cell, there is at least one discretization point in the occupied cell. Then, each discretization point is associated with one cell on a 2D-grid. Let $0 = t_0^i < t_1^i < t_2^i < \cdots < t_{N_i}^i = 1$ be a subdivision of $[0,1]$, for each t_k^i, the corresponding discretization point on curve γ_{i_H} is denoted as $P_{i,k}$, where $P_{i,k} = \gamma_{i_H}(t_k^i)$. We define a *curve section* as the section of a curve between two successive discretization points. Let $P_{i,k}$ and $P_{i,k+1}$ be two successive discretization points on curve γ_{i_H}, then the curve section between $P_{i,k}$ and $P_{i,k+1}$ is denoted by $\gamma_{i_{k,k+1}}$. An illustration of the curve discretization on a 2D-grid is presented in Fig. 5.1(a).



(a) (b)

Figure 5.1: An example of horizontal conflict detection. (a) Illustration of horizontal curve discretization. (b) Illustration of horizontal conflict detection.

We define a *horizontal violation*, which occurs when the minimum distance between two curve sections (in the form of arc or segment) on different curves is less than 3Nm. For γ_{i_H}, the horizontal violation is detected by considering successively its curve sections. Each curve section occupies at least one cell in the 2D-grid. For each curve section on γ_{i_H}, we only check the horizontal violation with other curve sections (on different curves) located in the same or neighboring cells, instead of checking the whole grid. The reason is that the horizontal violation with other curves only occurs in the same or neighboring cells, due to the size of each grid (3Nm×3Nm). In such a way, the efficiency of conflict detection is improved. Suppose that $P_{i,k}$ (respectively, $P_{i,k+1}$) is located in cell $({I_x}^{i,k}, {I_y}^{i,k})$ (respectively, $({I_x}^{i,k+1}, {I_y}^{i,k+1})$). Then the neighboring cells where potential horizontal violation with $\gamma_{i_{k,k+1}}$ may occur are $\{(I_x,I_y)|\min({I_x}^{i,k}, {I_x}^{i,k+1}) - 1 \leq I_x \leq \max({I_x}^{i,k}, {I_x}^{i,k+1}) + 1, \min({I_y}^{i,k}, {I_y}^{i,k+1}) - 1 \leq I_y \leq \max({I_y}^{i,k}, {I_y}^{i,k+1}) + 1\}$.

An example of neighboring cells is illustrated in Fig. 5.1(a). The discretization points on route γ_{1_H} are $P_{1,k}, k = 1, \cdots, 9$. The first curve section on γ_{1_H} is $\gamma_{1_{1,2}}$ (as shown in orange color). Both points are in cell $(2,6)$. Thus the neighboring cells are $\{(I_x,I_y)|1 \leq I_x \leq 3, 5 \leq I_y \leq 7\}$ (area covered by orange color). While considering the curve section between points $P_{1,7}$ (in cell $(4,3)$) and $P_{1,8}$ (in cell $(5,2)$), the neighboring cells are $\{(I_x,I_y)|3 \leq I_x \leq 6, 1 \leq I_y \leq 4\}$ (area covered by red color). Figure 5.1(b) presents an example of horizontal violation detection. Curve γ_{2_H} is discretized by points $P_{2,k}, k = 1, \cdots, 9$. Considering the curve section $\gamma_{1_{1,2}}$, we check all the neighboring cells (area in orange color) and three discretization points ($P_{2,4}$, $P_{2,5}$, $P_{2,6}$) on γ_{2_H} are found. Thus we measure

pairwise the minimum distance between $\gamma_{1_{1,2}}$ and $\gamma_{2_{3,4}}$ (respectively, $\gamma_{2_{4,5}}$, $\gamma_{2_{5,6}}$, $\gamma_{2_{6,7}}$) to determine whether horizontal violation exists. By repeating this operation along γ_{1_H}, all the violated curve sections on γ_{1_H} are found.

5.1.2 Vertical detection

Once a horizontal violation is detected, a further check in the vertical plane is needed. We denote $H_{inf}^{i,k}$ (respectively, $H_{sup}^{i,k}$) the lower (respectively, upper) bound of the cross section in the vertical plane at discretization point $P_{i,k}$ on route γ_i, where $H_{inf}^{i,k} = h_{i_{inf}}(t_k^i)$ and $H_{sup}^{i,k} = h_{i_{sup}}(t_k^i)$. Suppose that two curve sections $\gamma_{i_{k,k+1}}$ and $\gamma_{j_{l,l+1}}$ on curve γ_{i_H} and γ_{j_H} respectively ($i \neq j$) has a horizontal violation. We define these two curve sections in *vertical violation* when neither of the following conditions is satisfied:

$$\min\left(H_{inf}^{i,k}, H_{inf}^{i,k+1}\right) - \max\left(H_{sup}^{j,l}, H_{sup}^{j,l+1}\right) > 1000\text{ft} \tag{5.1}$$

$$\min\left(H_{inf}^{j,l}, H_{inf}^{j,l+1}\right) - \max\left(H_{sup}^{i,k}, H_{sup}^{i,k+1}\right) > 1000\text{ft} \tag{5.2}$$

If one or both conditions is satisfied, which implies that two curve sections are separated vertically at their extremities, then they are separated along the sections. The reason is that route vertical profiles are monotonically increasing. An example of vertical violation detection is illustrated in Fig. 5.2. Even though this detection method brings an additional margin in the separation, it has the advantage of being simple to implement. Finally, once two curve sections are violated in both horizontal and vertical plane, a conflict is identified.

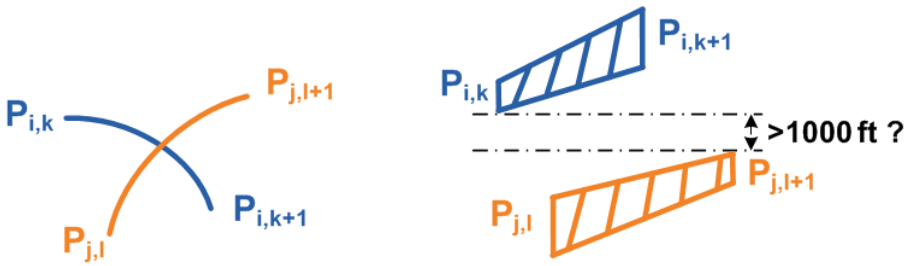
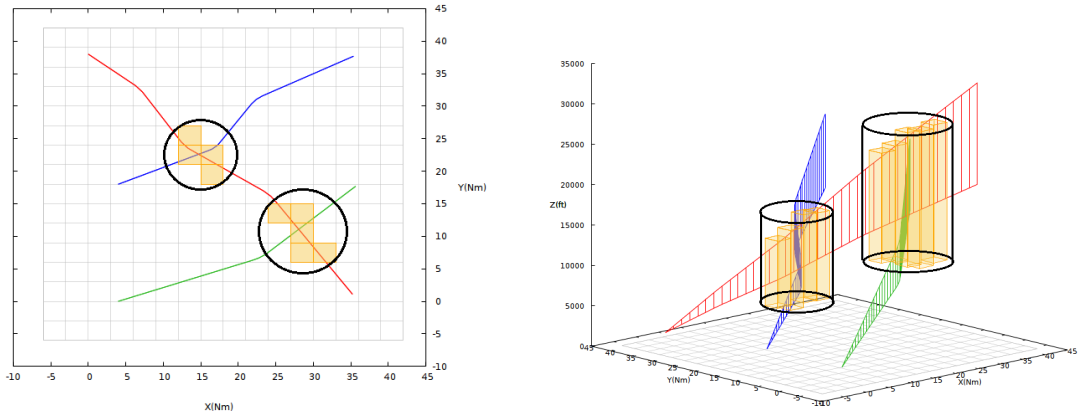


Figure 5.2: An example of vertical conflict detection. (a) A horizontal violation occurs between curve sections $\gamma_{i_{k,k+1}}$ and $\gamma_{j_{l,l+1}}$. (b) Detection of the vertical violation.

5.2 B&B-based approach

We propose a sequential *1-against-n* approach, based on the B&B method described in Chapter 4. The previously built routes become obstacles for the route that is considered later. Thus the quality of the solution depends on the routes generation order. Some route deviation strategies are applied in the B&B-based approach, in order to deviate a route locally around the conflicting area, while keeping the deviated route close to its optimal form computed by the B&B method.



(a) (b)

Figure 5.3: An example of clustering violated cells and creating fictitious obstacles. (a) Illustration in the horizontal plane. (b) Illustration in 3D.

5.2.1 Overview of the method

Multiples routes are generated sequentially according to a given order (*i.e.*, the decreasing order of their traffic loads). Each new route is generated progressively piecewise based on the B&B method, and on strategies to deviate the route around conflicting zones when conflicts with other routes occur. More precisely, each new route is firstly generated by B&B. Then we detect pairwise conflicts between it and the previously generated routes, as explained in Section. 5.1. The detected conflicts are clustered into groups. Each group of conflicts is then associated with a *fictitious* obstacle, modeled as a cylinder (see Section 3.2). In the case when more than one group of conflicts exist, we order the corresponding fictitious obstacles according to their projection lengths onto the line connecting the starting and ending points of the route, as explained in Section 4.1, and we consider the first fictitious obstacle $\Omega_f(C_f(x_f, y_f), r_f, z_{f_{inf}}, z_{f_{sup}})$. In order to solve the conflicts corresponding to the first fictitious obstacle, the natural idea is to deviate the considered route locally around this conflicting zone. Three possible ways of deviation are proposed, inspired by the three strategies to avoid an obstacle proposed in Section 3.3.1: bypassing counter-clockwise, bypassing clockwise, imposing a level flight.

After the deviation, the remaining route section is built by applying again the B&B, then the conflict on the new built route section is detected and the route deviation methods are applied once another conflict occurs. In such a way, the new route is built progressively, and each of its sections is deviated based on the optimal route structure given by the B&B. We note that a conflict is not guaranteed to be solved by one of the considered route deviation strategies. It may happen, indeed, that, building routes one after the other, not enough degrees of freedom are available to be able to solve the conflict. In this case, the proposed algorithm provides the information on the unsolved conflict and on the length of involved route sections. A pseudo-code of the proposed B&B-based sequential approach is presented in Algorithm 4. The following sections describe the key intermediate steps.

5.2.2 Clustering conflicting cells and generating fictitious obstacles

After detecting pairwise conflicts between the current considered route and the previously generated route, each conflict zone is associated with a fictitious obstacle (step 12 in Algorithm 4). The fictitious obstacle is in cylinder-shape, modeled in the same way as presented in Section 3.2.1.

Algorithm 4 Generation of multiple routes: Branch and Bound-based sequential approach

Require: routes starting and ending points, altitudes of the corresponding starting points, and the associated buffer obstacles $\{(A_i,B_i),H_{A_i},\Omega_{bi}|i=1,\cdots,N\}$ (classified in an increasing order of traffic load), obstacles $\{\Omega_j|j=1,\cdots,M\}$

1: Initialize: list of optimal routes $list=\emptyset$
2: Apply the B&B to generate the optimal route γ_1 connecting (A_1,B_1) ▷ call Algorithm 3($(A_1$,$B_1)$, H_{A_1}, Ω_{b1}, $\{\Omega_j|j=1,\cdots,M\}$)
3: Add γ_1 to $list$
4: **for** i=2 to N **do**
5: Apply the B&B to generate an initial route γ_i^0 connecting (A_i,B_i) ▷ call Algorithm 3(A_i,B_i), H_{A_i} Ω_{bi}, $\{\Omega_j|j=1,\cdots,M\}$)
6: Detect pairwise conflicts between γ_i^0 and routes in $list$
7: **if** no conflict detected **then**
8: Set $\gamma_i:=\gamma_i^0$, add γ_i to $list$
9: **else**
10: Set the current route $\gamma_i^{cur}:=\gamma_i^0$
11: **while** conflict detected **do**
12: Cluster conflicting cells and associate a fictitious obstacle with each cluster
13: Order the fictitious obstacles with respect to (A_i,B_i) in the same way as in Section 4.1
14: Deviate γ_i^{cur} locally around the conflict zone corresponding to the first fictitious obstacle Ω_f ▷ call Algorithm 5(γ_i^{cur},Ω_f, $\{\Omega_j|j=1,\cdots,M\}$, $list$)
15: Detect pairwise conflicts between the remaining route section after the last deviation on γ_i^{cur} and routes in $list$
16: **end while**
17: Apply the post-processing technique on γ_i^{cur}
18: Set $\gamma_i:=\gamma_i^{cur}$, add γ_i to $list$
19: **end if**
20: **end for**
21: **return** $list$

In order to generate fictitious obstacles, we define a *violated cell* as a cell containing at least one conflicting curve section on the current route. The minimum and maximum altitudes associated with a violated cell (I_x,I_y) are denoted as $z_{inf}(I_x,I_y)$ and $z_{sup}(I_x,I_y)$ respectively. Let $I\subset\{1,\cdots,N\}$ be the subset denoting the indices of the previously generated routes, and let $\{\gamma_{i_{k,k+1}}|k\in K_i,K_i\subset\{0,\cdots,N_i\},\ i\in I\}$ be the set of curve sections in conflict with the current route in cell (I_x,I_y). Then $z_{inf}(I_x,I_y)$ and $z_{sup}(I_x,I_y)$ are computed by:

$$z_{inf}(I_x,I_y)=\min\{H_{inf}^{i,k}|k\in K_i,\ i\in I\} \tag{5.3}$$

$$z_{sup}(I_x,I_y)=\max\{H_{sup}^{i,k}|k\in K_i,\ i\in I\} \tag{5.4}$$

The violated cells are then clustered into groups in such a way that two *adjacent* violated cells are in the same group. Two cells (I_x^k,I_y^k) and (I_x^l,I_y^l) are adjacent when $|I_x^k-I_x^l|\leq 1$ and $|I_y^k-I_y^l|\leq 1$. Each group of violated cells is associated with a fictitious obstacle. This fictitious obstacle is a cylinder enveloping the route sections corresponding to the violated cells. The altitude of its lower (respectively, upper) basis is the minimum (respectively, maximum) altitude associated with the violated cells in the corresponding group, as defined in (5.3) and (5.4).

An example of clustering violated cells and creating fictitious obstacles is illustrated in Fig. 5.3. The route in red color is built at last, and the conflict is detected between routes in blue and red colors, and between routes in green and red colors. Two conflict zones are detected, corresponding to two fictitious obstacles.

To explain more precisely the way of computing the characteristics of a fictitious obstacle corresponding to a group of violated cells, let G be a set of indices of violated cells clustered in the same group, and let $(x_{inf,G}, x_{sup,G})$ (respectively, $(y_{inf,G}, y_{sup,G})$, $(z_{inf,G}, z_{sup,G})$) be the lower and upper bounds of the x-coordinates (respectively, y-, z-) of the violated cells whose indices belong to group G, which are computed by:

$$\begin{cases} x_{inf,G} = \min\left(x_{inf}(I_x, I_y) | (I_x, I_y) \in G\right) \\ x_{sup,G} = \max\left(x_{sup}(I_x, I_y) | (I_x, I_y) \in G\right) \\ y_{inf,G} = \min\left(y_{inf}(I_x, I_y) | (I_x, I_y) \in G\right) \\ y_{sup,G} = \max\left(y_{sup}(I_x, I_y) | (I_x, I_y) \in G\right) \\ z_{inf,G} = \min\left(z_{inf}(I_x, I_y) | (I_x, I_y) \in G\right) \\ z_{sup,G} = \max\left(z_{sup}(I_x, I_y) | (I_x, I_y) \in G\right) \end{cases} \tag{5.5}$$

Then the associated fictitious obstacle is $\Omega_f(C_f(x_f, y_f), r_f, z_{f_{inf}}, z_{f_{sup}})$, where:

$$\begin{cases} x_f = \frac{1}{2}(x_{inf,G} + x_{sup,G}) \\ y_f = \frac{1}{2}(y_{inf,G} + y_{sup,G}) \\ r_f = \frac{1}{2}(x_f - x_{inf,G} + y_f - y_{inf,G}) \\ z_{f_{inf}} = z_{inf,G} \\ z_{f_{sup}} = z_{sup,G} \end{cases} \tag{5.6}$$

5.2.3 Route deviation strategies

If one or more fictitious obstacles are generated in Step 12 in Algorithm 4, we deviate the current considered route locally around the conflict zone corresponding to the first fictitious obstacle $\Omega_f(C_f(x_f, y_f), r_f, z_{f_{inf}}, z_{f_{sup}})$ (step 14 in Algorithm 4). A fictitious obstacle is modeled as a cylinder (see Subsection 3.2.1), thus it is natural to associate two decision variables s_{ij} and t_{ij} with each fictitious obstacle. When a fictitious obstacle is active ($s_{ij} = 1$), it can be avoided by three bypassing strategies: turn counter-clockwise ($t_{ij} = 0$), turn clockwise ($t_{ij} = 1$) and impose a level flight ($t_{ij} = 2$). Since the Continuous Climb Operation (CCO) and Continuous Descent Operation (CDO) are preferred in TMAs, we consider in our route deviation strategies that the horizontal deviation has a priority over a vertical deviation. If the horizontal deviation is able to solve the considered conflict, then there is no need to try the vertical deviation. A pseudo-code of the route deviation is presented in Algorithm 5. In the following, we present in more detail the different strategies of route deviation.

Deviating route by counter-clockwise and clockwise turn

In the case of avoiding a fictitious obstacle by a turn ($t_{ij} = 0$ or 1), the deviated route must be on the one hand smooth and flyable, on the other hand close to the current route γ_i^{cur}. Indeed, in the route deviation process, we aim not only at reducing conflicts, but also at keeping the deviated route close to the optimal route generated by the B&B. To reach these objectives, for each fictitious obstacle, four *smoothing* obstacles tangent to both the considered fictitious obstacle and γ_i^{cur} are generated. A smoothing obstacle is in a cylinder shape, modeled in the same way as presented

Algorithm 5 Route deviation

Require: the current route γ_i^{cur}, the first fictitious obstacle in the ordered list Ω_f, obstacles $\{\Omega_j | j = 1, \cdots, M\}$, the list of previously generated routes *list*

Re-generate the portion of route γ_i^{cur} involved in the conflict by deviating it counter-clockwise $\Rightarrow$ obtain γ_i^{ccw} ▷ call Algorithm 6(γ_i^{cur}, Ω_f, $\{\Omega_j | j = 1, \cdots, M\}$, *list*)
Re-generate the portion of route γ_i^{cur} involved in the conflict by deviating it clockwise $\Rightarrow$ obtain γ_i^{cw} ▷ call Algorithm 6(γ_i^{cur}, Ω_f, $\{\Omega_j | j = 1, \cdots, M\}$, *list*)
if both γ_i^{ccw} and γ_i^{cw} solve the corresponding conflict **then**
    Set γ_i^{cur} equal to the one with the shorter length
else if only one of γ_i^{ccw} and γ_i^{cw} solves the corresponding conflict **then**
    Set γ_i^{cur} equal to the one that solves the corresponding conflict
else
    Re-generate the portion of route γ_i^{cur} involved in the conflict by imposing a level flight $\Rightarrow$ obtain γ_i^{lf} ▷ call Algorithm 7(γ_i^{cur}, Ω_f, $\{\Omega_j | j = 1, \cdots, M\}$, *list*)
    if γ_i^{lf} solves the corresponding conflict **then**
        Set $\gamma_i^{cur} := \gamma_i^{lf}$
    else
        Set γ_i^{cur} equal to the one among γ_i^{ccw}, γ_i^{cw} and γ_i^{lf} which has the minimum length involved in conflict on the deviated portion of route
    end if
end if
return γ_i^{cur}

in Section 3.2.1. Its radius r_s ($r_s \geq R_{min}$) is a user-defined parameter (the same for all smoothing obstacles), and the altitudes of its lower and upper basis are the same as the corresponding fictitious obstacle. Figure 5.4 illustrates the four smoothing obstacles given γ_i^{cur} and a fictitious obstacle.

Let γ_i^{ccw} and γ_i^{cw} denote the new routes obtained by deviating the portion of route γ_i^{cur} involved in a conflict counter-clockwise and clockwise respectively. When the counter-clockwise bypassing strategy is chosen, smoothing obstacles 1 and 3 are taken to build γ_i^{ccw} and the turn direction on both smoothing obstacles is clockwise, as shown in Fig. 5.5(a). The point B_i' is the tangent point of γ_i^{cur} and smoothing obstacle 3. The partial deviated route $\gamma_i^{ccw,par}$ between the starting point A_i and point B_i' is built in the following way:

- Firstly, set A_i and B_i' as the starting and ending points of $\gamma_i^{ccw,par}$ respectively.
- Secondly, create a list of obstacles, containing all the active obstacles when building γ_i^{cur}.
- Thirdly, add the fictitious obstacle Ω_f to the list of obstacles and order the obstacles in the list according to their projection length to line (A_iB_i').
- Afterwards, for the obstacles whose order index is lower than that of Ω_f, keep them as active and their avoiding strategies; while for the other obstacles whose order index is higher than that of Ω_f, set them as non-considered.
- If an active real obstacle is avoided by a turn and intersects the fictitious obstacle, also set it as non-considered.
- Moreover, add smoothing obstacles 1 and 3 before and after the fictitious obstacle respectively.

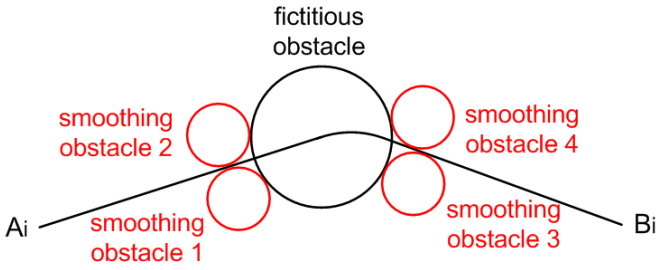


Figure 5.4: Configuration of the smoothing obstacles.

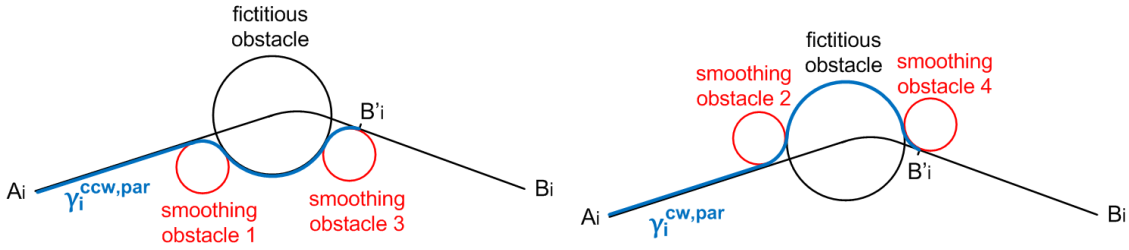


Figure 5.5: Route deviation in the horizontal Plan. (a) Turn counter-clockwise around the fictitious obstacle. (b) Turn clockwise around the fictitious obstacle.

- Finally, build the route in the same way as presented in Section 4.1.

Afterwards, we detect pairwise conflicts between $\gamma_i^{ccw,par}$ and the previously generated routes. If any residual conflict is detected, we iteratively increase the radius of the fictitious obstacle by a user-defined value δ_R, until the conflict is solved or the radius reaches the maximum radius R_{max}. In the case when the radius of the fictitious obstacle reaches R_{max} and a residual conflict still exists, $\gamma_i^{ccw,par}$ is set to be the deviated route portion satisfying all constraints except route separation and with the smallest length involved in conflicts obtained in the iterations.

To complete the deviated route γ_i^{ccw} by building the route section between B_i' and B_i, we propose a *modified-B&B* method. The principle is that, first, we create a list of obstacles including all the real obstacles as well as the fictitious and smoothing obstacles created when deviating the route. Then, in order to keep the deviated route section between A_i and B_i' unchanged in the B&B method, the index as well as the values of the decision variables for the obstacles considered when building $\gamma_i^{ccw,par}$ are not changed. Afterwards, the other obstacles non-considered when building $\gamma_i^{ccw,par}$ are ordered according to their projection length on line (A_iB_i). Moreover, the values of the decision variables for the non-active obstacles are determined in the B&B method.

In the case when the clockwise bypassing strategy is chosen, smoothing obstacles 2 and 4 are taken to build $\gamma_i^{cw,par}$ and the turn direction on them is counter-clockwise, as presented in Fig. 5.5(b). The approach for building γ_i^{cw} is similar to that proposed for building γ_i^{ccw}. A pseudo-code to summarize how to generate γ_i^{ccw} (respectively, γ_i^{cw}) is presented in Algorithm 6.

Deviating route by imposing level flights

In the case of avoiding a fictitious obstacle by imposing a level flight ($t_{ij} = 2$), there are two different cases of route deviation to be handled, depending on the relative position between the fictitious obstacle and γ_i^{cur}. Each fictitious obstacle has two intersections with γ_i^{cur} in the horizontal

Algorithm 6 Generation of γ_i^{ccw} (respectively, γ_i^{cw})

Require: the current route γ_i^{cur}, the first fictitious obstacle Ω_f, obstacles $\{\Omega_j | j = 1, \cdots, M\}$, the list of previously generated routes *list*

Initialize: radius $r'_f := r_f$
Set Ω_f as active and avoided by a counter-clockwise turn (respectively, clockwise turn)
while $r'_f < R_{max}$ **do**
    Generate the partial deviated route section $\gamma_i^{ccw,par}$ (respectively, $\gamma_i^{cw,par}$) between A_i and B'_i
    Detect pairwise conflict between $\gamma_i^{ccw,par}$ (respectively,$\gamma_i^{cw,par}$) and routes in *list*
    if No residual conflict and $\gamma_i^{ccw,par}$ (respectively,$\gamma_i^{cw,par}$) satisfies all constraints except route separation **then**
        Complete the remaining route section by the modified-B&B $\Rightarrow$ obtain γ_i^{ccw} (respectively,γ_i^{cw})
        Break
    else
        $r'_f := r'_f + \delta_R$, continue
    end if
end while
if γ_i^{ccw} (respectively,γ_i^{cw}) is not obtained **then**
    Set $\gamma_i^{ccw,par}$ (respectively,$\gamma_i^{cw,par}$) the deviated route portion satisfying all constraints except route separation and with the smallest length involved in conflicts obtained in the iterations
    Complete the remaining route section by the modified-B&B $\Rightarrow$ obtain γ_i^{ccw} (respectively,γ_i^{cw})
end if
return γ_i^{ccw} (respectively,γ_i^{cw})

plane, since the corresponding conflict zone is located around γ_i^{cur}. Let us denote the intersection closer to B_i as B'_i.

The first case occurs when B'_i is located on an arc of γ_i^{cur}, as shown in Fig. 5.6(a). Then the partial deviated route $\gamma_i^{lf,par}$ connecting A_i and B'_i is built in the following way:

- Firstly, set A_i and B'_i as the starting and ending points of $\gamma_i^{lf,par}$ respectively.
- Secondly, create a list of obstacles, consisting all the active obstacles when building γ_i^{cur}.
- Then, add the fictitious obstacle to the list of obstacles and order them according to their projection length to line $(A_iB'_i)$.
- Moreover, for the obstacles in the list whose order index is lower than that of the obstacle where B'_i lies on (including the obstacle itself), keep their avoiding strategies; while for the other obstacles, set them as non-considered.
- Finally, build the route in the same way as presented in Section 4.1.

The second case occurs when B'_i is located on an segment of γ_i^{cur}. In order to keep the horizontal form of the deviated route, we add two smoothing obstacles tangent to both the fictitious obstacle and the line where the segment is lying on, as shown in Fig. 5.6(b). Then the partial deviated route $\gamma_i^{lf,par}$ connecting A_i and B'_i is built in the following way:

- Firstly, set A_i and B'_i as the starting and ending points of $\gamma_i^{lf,par}$.

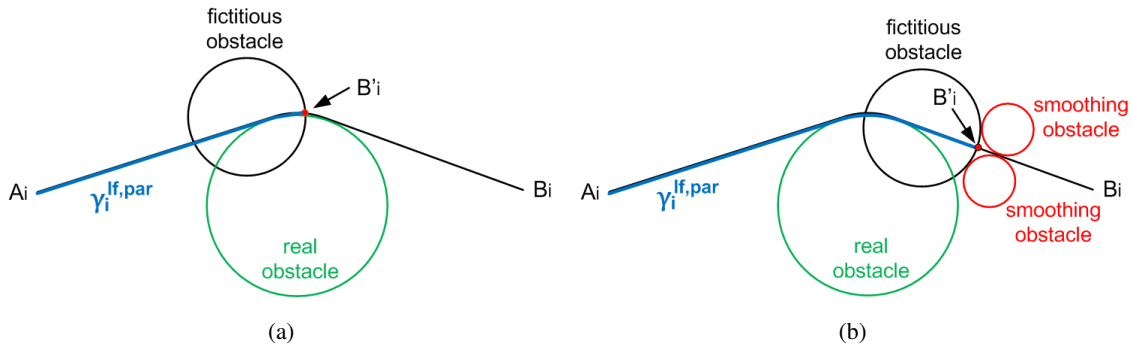


Figure 5.6: Route deviation in the vertical plane. (a) Case 1, when the second intersection point is located on an arc. (b) Case 2, when the second intersection point is located on a segment.

- Secondly, create a list of obstacles, consisting all the active obstacles when building γ_i^{cur}.
- Then, add the fictitious obstacle to the list of obstacles and order them according to their projection length to line (A_iB_i').
- Moreover, for the obstacles whose order index is lower than that of the fictitious obstacle, keep their avoiding strategies; while for the other obstacles, set them as non-considered.
- Finally, build the route in the same way as presented in Section 4.1.

After building $\gamma_i^{lf,par}$, we detect pairwise conflicts between $\gamma_i^{lf,par}$ and the previous routes. If any residual conflict is detected, we iteratively decrease the altitude of the lower basis of the fictitious obstacle by a user-defined value δ_H, until the conflict is solved or the altitude of the lower basis reaches H_{min}. In the case when the altitude of the lower basis of the fictitious obstacle reaches H_{min} and residual conflict still exist, $\gamma_i^{lf,par}$ is set to be the deviated route portion satisfying all constraints except route separation and with the smallest length involved in conflicts obtained in the iterations.

To complete the route section between B_i' and B_i, a *modified-B&B* method is applied. In the case of imposing level flight, we create a list of obstacles including all the real obstacles as well as the fictitious and smoothing obstacles created when deviating the route. These obstacles are then ordered according to their projection length on line (A_iB_i). In order to keep the form of $\gamma_i^{lf,par}$ unchanged in the modified-B&B method, the values of the decision variables for the obstacles whose order index is lower than that of the fictitious obstacle (including the fictitious obstacle itself) are kept. Afterwards, the methods for completing the remaining route section in the first case (when B_i' is located on an arc) and in the second case (when B_i' is located on a segment) are slightly different.

- In the first case, the values of the decision variables of the obstacle where the second intersection point is located is kept. The values of the decision variables for the other obstacles are determined in the modified-B&B method.
- In the second case, the values of the decision variables for the other obstacles (including the two smoothing obstacles) are determined in the modified-B&B method. Meanwhile, only counter-clockwise bypassing is allowed on smoothing obstacle 1 when it is active and only clockwise bypassing is allowed on smoothing obstacle 2 when it is active.

A pseudo-code to summarize the way γ_i^{lf} is generated is presented in Algorithm 7.

Algorithm 7 Generation of γ_i^{lf}

Require: the current route γ_i^{cur}, the first fictitious obstacle Ω_f, obstacles $\{\Omega_j | j = 1, \cdots, M\}$, the list of previously generated routes *list*

1: Initialize: altitude $z'_{f_{inf}} = z_{f_{inf}}$
2: Set Ω_f as active and avoided by imposing a level flight
3: **while** $z'_{f_{inf}} > H_{min}$ **do**
4: Generate the partial deviated route $\gamma_i^{lf,par}$
5: Detect pairwise conflict between $\gamma_i^{lf,par}$ and the previous routes
6: **if** No residual conflict and $\gamma_i^{lf,par}$ satisfies all constraints except route separation **then**
7: Complete the remaining route section by the modified-B&B $\Rightarrow$ obtain γ_i^{lf}
8: Break
9: **else**
10: $z'_{f_{inf}} := z'_{f_{inf}} - \delta_H$, continue
11: **end if**
12: **end while**
13: **if** γ_i^{lf} is not obtained **then**
14: Set $\gamma_i^{lf,par}$ as the deviated route portion satisfying all constraints except route separation and with the smallest length involved in conflicts obtained in the iterations.
15: Complete the remaining route section by the modified-B&B $\Rightarrow$ obtain γ_i^{lf}
16: **end if**
17: **return** γ_i^{lf}

Post-processing technique

In the route deviation process, each horizontal deviation is realized by bypassing sequentially a smoothing obstacle, a fictitious obstacle and a second smoothing obstacle, as shown in Fig. 5.5. Thus the deviated route section is composed by three successive arcs lying on the three obstacles. This kind of curve is flyable in reality, since a RF leg can join another RF leg with an opposite turn direction and a different radius [107]. However, in order to simplify the operation for pilots, we propose a post-processing technique on γ_i^{cur} (step 17 in Algorithm 4). For a real SID/STAR, a horizontal deviation occurs to avoid an obstacle or to avoid another route, afterwards the route usually connects directly to the next route section. Thus, in a SID (respectively, STAR) case, for each horizontal route deviation, we set the second (respectively, first) smoothing obstacle as non-active and the values of the decision variables for the other obstacles are preserved and we re-build the route γ_i^{cur}. If no more conflict with the previously generated routes is induced by the post-processing, then the new γ_i^{cur} is accepted.

5.2.4 Step-by-step illustration

To illustrate the algorithm, let us give an example of building a SID when two routes are already generated, as illustrated in Fig. 5.7. For the sake of simplification, there is neither an obstacle, nor a buffer obstacle.

Step 0 (Fig. 5.7(a)): Routes γ_1 and γ_2 are previously generated routes. An initial route γ_3^0 connecting A_3 and B_3 is generated by the B&B method (Algorithm 3). The conflict detection technique is applied pairwise between γ_3^0 and γ_1, and between γ_3^0 and γ_2. The conflicts are clustered into two groups, each of them is associated with a fictitious obstacle. By ordering the fictitious obstacles

according to their projection lengths to line (A_3B_3), they are numbered as Ω_{f1} and Ω_{f2}.

Step 1 (Fig. 5.7(b)): In order to deviate γ_3^0 around the conflict zone corresponding to Ω_{f1}, four smoothing obstacles Ω_{s1}, Ω_{s2}, Ω_{s3} and Ω_{s4} are generated.

Step 2 (Fig. 5.7(c)): The partial deviated route $\gamma_3^{ccw,par}$ bypassing counter-clockwise around Ω_{f1} is generated.

Step 3 (Fig. 5.7(d)): The corresponding conflict is not totally solved, thus the radius of the fictitious obstacle is increased, and the partial deviated route $\gamma_3^{ccw,par}$ is again generated.

Step 4 (Fig. 5.7(e)): The corresponding conflict is solved, the remaining route section between B'_3 and B_3 is completed by B&B.

Step 5 (Fig. 5.7(f)): The partial deviated route $\gamma_3^{cw,par}$ bypassing clockwise Ω_{f1} is generated.

Step 6 (Fig. 5.7(g)): The corresponding conflict is solved, the remaining route section between B'_3 and B_3 is completed by B&B.

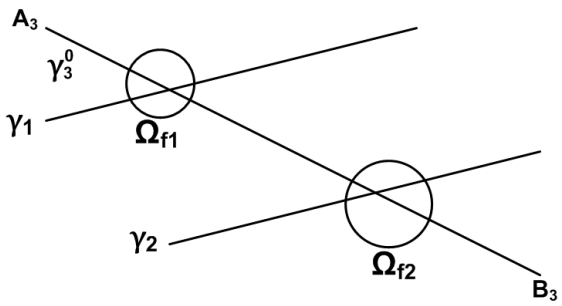
Step 7 (Fig. 5.7(h)): The current route γ_3^{cur} is set to be the route deviated clockwise, since its length is shorter. Afterwards, the conflict detection technique is again applied to the route section between B'_3 and B_3 and the previously generated routes. A new fictitious obstacle is created.

Step 8 (Fig. 5.7(i)): We generate a partial deviated route to avoid the conflict zone corresponding to the first fictitious obstacle.

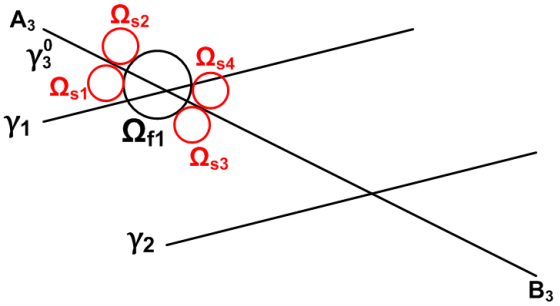
Step 9 (Fig. 5.7(j)): After completing the remaining route section, the current route becomes γ_3^{cur}.

Step 10 (Fig. 5.7(k)): We apply the post-processing on γ_3^{cur}, by setting the second smoothing obstacles for all the horizontal deviations as non-active, while keeping the values of decision variables for the other obstacles.

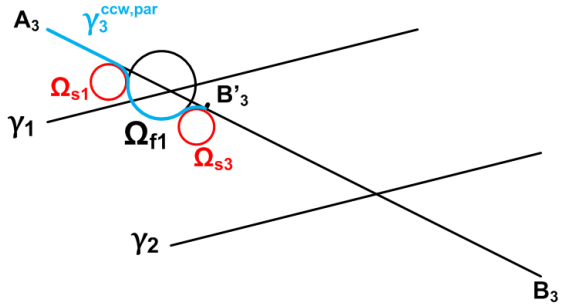
Step 11 (Fig. 5.7(l)): The obtained route after post-processing does not introduce any conflict, thus it is accepted as the optimal route γ_3 connecting A_3 and B_3.



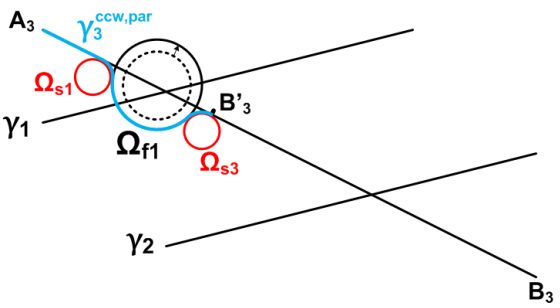
(a) Step 0.



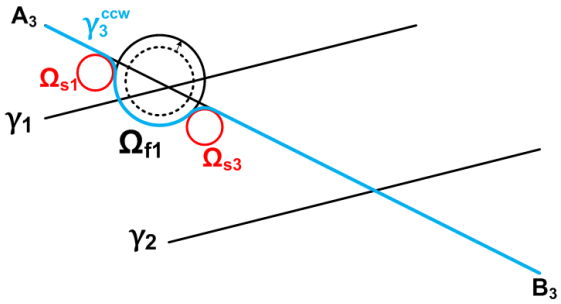
(b) Step 1.



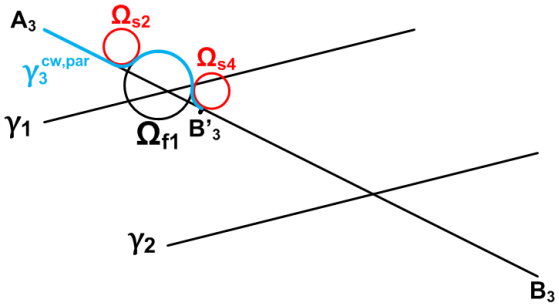
(c) Step 2.



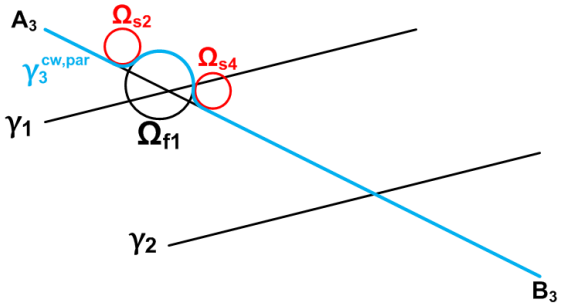
(d) Step 3.



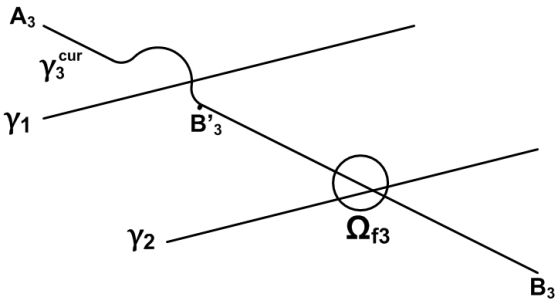
(e) Step 4.



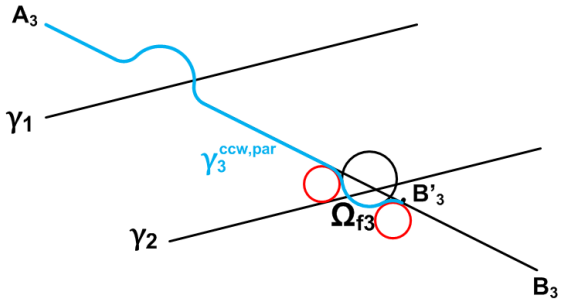
(f) Step 5.



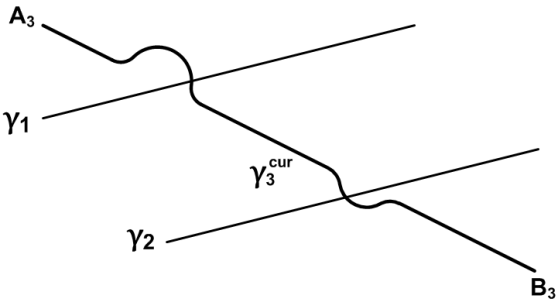
(g) Step 6.



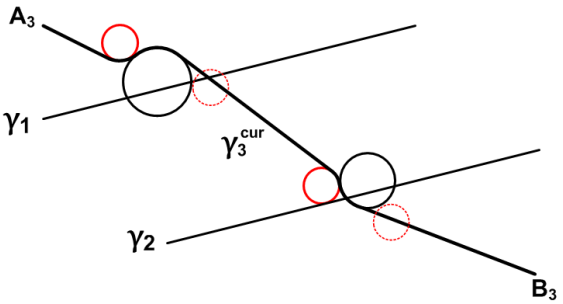
(h) Step 7.



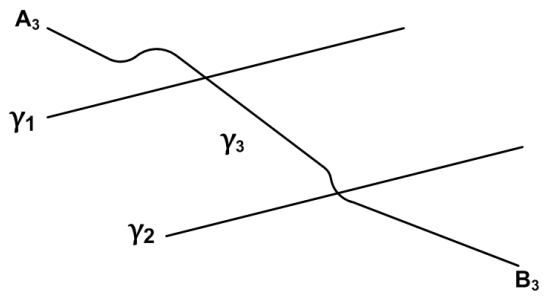
(i) Step 8.



(j) Step 9.



(k) Step 10.



(l) Step 11.

Figure 5.7: Step-by-step illustration.

5.3 Simulated Annealing (SA) approach

The quality of a solution provided by the B&B-based method depends on the routes generation order. In this section, we propose another method using a global strategy, where routes are generated simultaneously. The method that we apply is the Simulated Annealing (SA) method, which is usually applied to highly combinatorial optimization problems.

5.3.1 Description of the method

A general description of the SA method has been provided in Section 1.4. The SA method starts from an initial configuration S_0. In our case, it refers to the set of routes built individually by the B&B. If these routes are conflict-free, then S_0 is already an optimal solution. Otherwise, at each step in the SA algorithm, a neighboring solution S_N is generated based on the current solution S_C. The way of generating S_N is specifically tailored to our problem, and it is presented in Algorithm 8. Some intermediate steps are described in the following.

Algorithm 8 Generation of a neighboring solution S_N

Require: total number of routes to build N, the current solution S_C

Initialize: the neighboring solution $S_N := \emptyset$, a list of buffer obstacle, obstacles and fictitious obstacles $list_{obst} := \emptyset$, a boolean $accept := false$
for Each curent route γ_i^{cur} in S_C **do**
    Set $list_{obst} := \emptyset$, $accept := false$
    Detect pairwise conflicts between γ_i^{cur} and the other routes in S_C (as explained in Section 5.1)
    if γ_i^{cur} is involved in conflicts **then**
        Create at least one *fictitious* obstacle in cylinder shape corresponding to conflicting zones (as explained in Section 5.2.2)
    end if
    Create a list $list_{obst}$ associated with the new route γ_i^{new} to build, consisting of the buffer obstacle, obstacles, fictitious obstacles associated with S_C, and the new created fictitious obstacles
    Set the values of the decision variables related to the buffer obstacle and obstacles in $list_{obst}$ the same as the ones when building γ_i^{cur}
    Apply a clustering technique on the fictitious obstacles in $list_{obst}$ whose projections in the horizontal plane are overlapped
    while $accept = false$ **do**
        for Each fictitious obstacle in $list_{obst}$ **do**
            Select the decision variables related to this fictitious obstacle
        end for
        Build route γ_i^{new} according to $list_{obst}$ (as explained in Section 4.1)
        Post-process γ_i^{new}
        if γ_i^{new} satisfies all constraints (as mentioned in Section 3.3.2) except the route separation **then**
            Add γ_i^{new} to S_N
            Set $accept := true$
        end if
    end while
end for
return S_N

The fictitious obstacles in $list_{obst}$, whose projections in the horizontal plane are overlapped, are clustered and replaced by a new fictitious obstacle (step 10 of Algorithm 8). This new fictitious obstacle is the smallest cylinder enveloping them. There are two main reasons for doing so. First, as new fictitious obstacles are generated iteratively in the SA algorithm, their amount continues to increase, clustering them is a possible way to reduce the size of the state space. Second, the situation when fictitious obstacles are overlapped may indicate that the potential conflict zone is in a larger scale than the currently detected one, represented by the fictitious obstacles. Enveloping the existing overlapped fictitious obstacles is a good way to enlarge the area to be avoided.

We develop different schemes on selecting the decision variables related to the fictitious obstacles (step 13 of Algorithm 8), and we present two representative ones among them.

- In the first scheme, a fictitious obstacle that is not active in S_C or is new created (steps 6 or 10 of Algorithm 8), is set randomly to be active in S_N with a probability ρ_1; meanwhile, a fictitious obstacle that is active in S_C, is set randomly to be active in S_N with a probability ρ_2. In addition, a fictitious obstacle which is active in both S_C and S_N, has a probability ρ_3 to keep the same avoidance strategy in S_N as in S_C. The values of ρ_1, ρ_2 and ρ_3 are user-defined, between 0 and 1. By adjusting their values, users may choose implicitly to which extent the new solution S_N is modified based on S_C. For example, by giving a low value to ρ_1 and high values to ρ_2 and ρ_3, the fictitious obstacles that are active in S_C tend to be active in S_N, and their avoidance strategies will probably be maintained. But the other fictitious obstacles are less likely to be active. In such a way, the form of new generated routes in S_N could be similar to the ones in S_C.

- In the second scheme, the traffic load on each route is compared with a parameter ρ_4, which is also a user-defined value between 0 and 1. Only if the route has a traffic load lower than ρ_4, the related fictitious obstacles may have a chance to be avoided by a level flight. By using this scheme, it is possible to ensure that the route with important traffic load has a continuous ascent or descent vertical profile.

The post-processing (step 16 of Algorithm 8) deals with the new built route γ_i^{new} from three aspects:

- First of all, it checks whether a route intersects any real obstacle, due to the route deviation brought by fictitious obstacles. If it is the case, the real obstacle is set as active on this route and its avoidance strategy is randomly selected.

- Afterwards, it checks whether any arc on a route corresponds to a central angle greater than 180°. If it is the case, the corresponding obstacle is set as non-active on that route.

- Finally, a non-active fictitious obstacle on each route is deleted with a probability ρ_5 (user-defined value between 0 and 1), so as to reduce the size of the state space.

The cost function to evaluate the fitness of a solution in our SA method is different from the objective function (3.15) proposed in Section 3.3.3. In the cost function of SA, a term representing the length of route sections involved in conflicts is added:

$$cost = L + c_3 \sum_{i=1}^{N} \ell_{i_{cflt}} \tag{5.7}$$

where L is defined by the objective function (3.15), $\ell_{i_{cflt}}$ is the total length of route sections on route γ_i that are involved in conflicts, and c_3 is a weight coefficient.

In fact, the SA applies a global strategy where routes are generated at the same time, thus the constraint on route separation can not be dealt with in the same way as in a sequential strategy (where the previously generated routes are regarded as obstacles for the routes to be generated later). Adding a term measuring the conflicts in the cost function is a common way to handle conflicts in the methods using a global strategy. In order to solve conflicts between routes, the value of weight coefficient c_3 is generally much higher than the values of c_1 and c_2, so that the SA method gives a priority for conflict resolution.

5.3.2 Step-by-step illustration

To illustrate the way a neighboring solution is generated in the SA algorithm, we give a step-by-step illustration in Fig. 5.8. For the sake of simplification, there is neither an obstacle, nor a buffer obstacle.

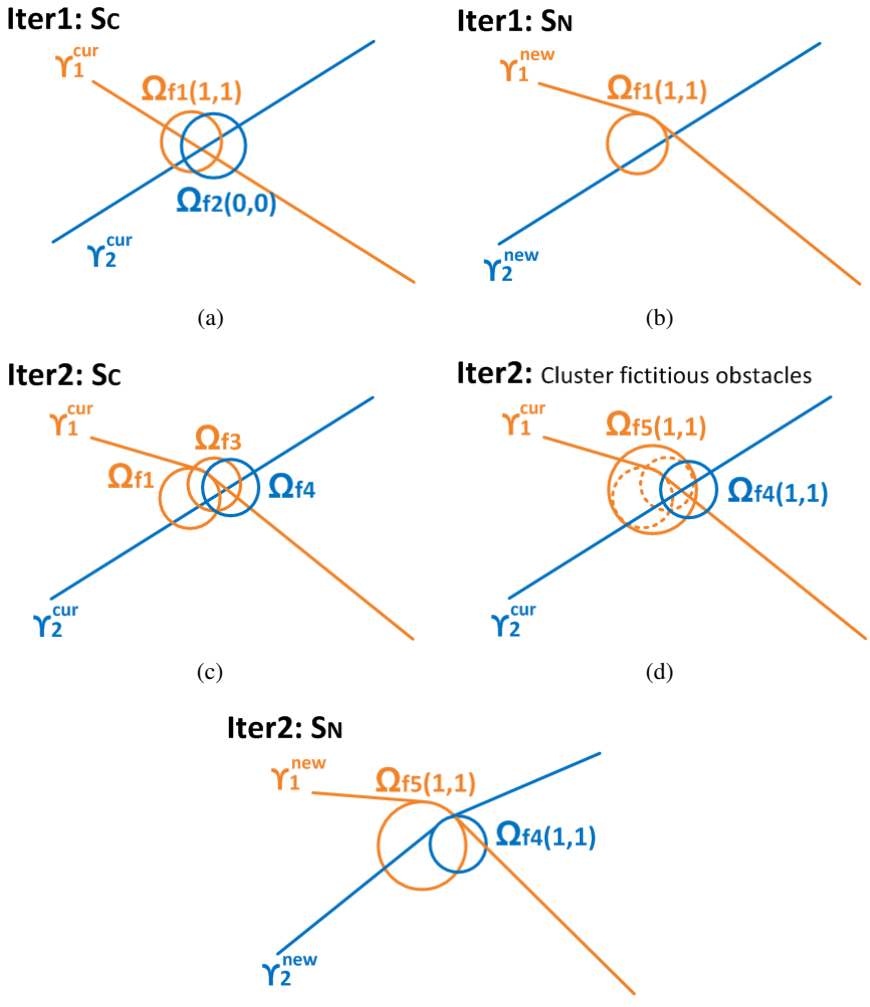


Figure 5.8: An example of generating neighboring solutions in the SA algorithm. (a) Iteration 1, current solution S_C and the generated fictitious obstacles. (b) Iteration 1, neighboring solution S_N. (c) Iteration 2, current solution S_C and the generated fictitious obstacles. (d) Iteration 2, clustering fictitious obstacles associated with γ_1^{new}. (e)Iteration 2, neighboring solution S_N.

The current solution S_C in the first iteration is shown in Fig. 5.8(a). After detecting conflicts between routes γ_1^{cur} and γ_2^{cur}, the fictitious obstacle Ω_{f1} (respectively, Ω_{f2}) is generated corresponding to the conflicting zone on γ_1^{cur} (respectively, γ_2^{cur}). The avoidance strategies of these two fictitious obstacles are randomly selected, and the one of Ω_{f1} (respectively, Ω_{f2}) is $(1,1)$ (respectively, $(0,0)$).

The neighboring solution S_N in the first iteration is generated correspondingly, as shown in Fig. 5.8(b). Since the fictitious obstacle Ω_{f2} is not active, it is eliminated in the post-processing.

The previously obtained neighboring solution is accepted as the current solution for the second iteration, as shown in Fig. 5.8(c). After the conflict detection, the fictitious obstacles Ω_{f3} (respectively, Ω_{f4}) is generated corresponding to the conflicting zone on γ_1^{cur} (respectively, γ_2^{cur}).

Fictitious obstacles Ω_{f1} and Ω_{f3} are both associated with the route γ_1^{new} to be built in the neighboring solution S_N, and their projections in the horizontal plane are overlapped, as shown in Fig. 5.8(d). Thus they are clustered and replaced by a larger fictitious obstacle Ω_{f5} enveloping them. The avoidance strategy of Ω_{f4} (respectively, Ω_{f5}) is selected as $(1,1)$ (respectively, $(1,1)$).

The neighboring solution S_N in the second iteration is generated correspondingly, as shown in Fig. 5.8(e).

5.4 Conclusion

In this chapter, the design of multiple routes was considered. We first presented an efficient method to detect pairwise conflict between routes. To design multiple conflict-free routes, two different approaches were proposed. The first one is a Branch and Bound (B&B)-based method, where routes are built sequentially and the existing routes become obstacles for the routes to be generated later. The second method is the Simulated Annealing (SA) where routes are generated simultaneously. Another direction to deal with the impact of routes generation order in the B&B-based method is, to combine it with a SA method which handles the order of routes generation.

Chapter 6

Simulation Results

In this chapter some tests on the design of multiple routes are carried out. We first test the proposed methodology on two artificially generated problems with various numbers of routes to design, as well as various numbers and layouts of obstacles. Then we present the tests performed on real TMAs: the TMA of Paris Charles de Gaulle (CDG) airport and the TMA of Zurich airport. In the case of CDG airport, we design 15 routes and compare the obtained routes with the standard routes published in Jeppesen charts. In the case of Zurich airport, the design of 9 routes in mountainous environment is considered, a comparison between the obtained routes and the standard routes is also provided. For each presented test, both B&B-based method and SA method are applied. The SA method is run 10 times for each test, and the statistics of the 10 simulations as well as some simulation results including the best solution are given. Tests were run on a Linux platform with a 2.4 GHz processor and 8 GB RAM. In the following, some parameters in preparation for the tests performed in this chapter are given.

6.1 Description of the simulation context (parameters, color legend and symbols)

The input data related to the routes to design are given in Table 6.1. The values of the user-defined parameters related to the B&B-based method are given in Table 6.2. By taking the weight coefficients $c_1 = 1$, $c_2 = 0$, the value to be minimized in the objective function (3.15) is in fact the length of a route in the horizontal plane. The user-defined parameters related to the SA method are set after several empirical tests, their values are given for each test in the following sections.

minimum radius of a RF leg, R_{min}	5Nm
maximum radius of a RF leg, R_{max}	13Nm
maximum number of level flights on each route, N_{max}	2
minimum altitude of each level flight, H_{min}	3500ft
minimum length of each level flight, L_{min}	5Nm
minimum distance between two successive level flights, D_{min}	5Nm

Table 6.1: Input data related to routes to design.

parameter	value
radius of smoothing obstacles, r_s	5Nm
fictitious obstacle radius increment, δ_R	1Nm
fictitious obstacle lower basis altitude decrement, δ_H	1000ft
weight coefficient in (3.15), c_1	1
weight coefficient in (3.15), c_2	0

Table 6.2: Values of user-defined parameters related to the B&B-based method.

Colors legend in the figures

In the figures presented in the following sections, we use different colors to represent different types of routes, obstacles or areas. More precisely,

- SIDs in the horizontal plane or in 3D: *blue* curves;
- STARs in the horizontal plane or in 3D: *red* curves;
- buffer obstacle: *black* circle filled by *black stripes*;
- real obstacle: *black* circle;
- fictitious obstacle: *pink* circle;
- smoothing obstacle: *purple* circle;
- conflicting area: *orange* circle.

Symbols and units used in the tables

In the tables presented in the following sections, we apply the following symbols to simplify the notations related to route length, where $i = 1, \cdots, N$ denotes the route index:

- γ_i^0, the route computed individually by the B&B method proposed in Chapter 4, without considering the impact of other routes.
- $L_{\gamma_i^0}$, the length of route γ_i^0 in the horizontal plane.
- $\ell_{i_{LF}}^0$, the total length of level flight on route γ_i^0.
- $\ell_{i_{cflt}}^0$, the total length of conflicting route sections on γ_i^0. In the B&B-based method, this length is computed by only taking into account the route sections in conflict with the previously generated routes. Note that, a previously generated route may be deviated from the form generated individually by the B&B. In the SA method, the route sections on γ_i^0 in conflict with all the other routes $\{\gamma_j^0 | j \in \{1, \cdots, N\} \setminus i\}$ are taken into account.
- γ_i^*, the route obtained by the B&B-based method or the SA method.
- $L_{\gamma_i^*}$, the length of route γ_i^* computed according to (3.15).
- $\ell_{i_{LF}}^*$, the total length of level flight on route γ_i^*.

- $\ell^*_{i_{cflt}}$, the total length of conflicting route sections on γ^*_i.

Moreover, in the following sections, the unit of coordinates in x-, y-axis, radius r, and lengths related to routes is in Nm, and the unit of altitudes in z-axis is in ft. In the tests of Paris CDG airport and Zurich airport, the coordinates are given in a Cartesian Coordinate System where the center is 47°N in latitude and 0° in longitude.

6.2 Artificially generated problems

We first present the results of tests carried out on two artificially generated problems. The take-off and landing slopes for the following tests are:

- taking-off slopes $\alpha_{min,TO} = 7\%$ ($\sim 4°$), $\alpha_{max,TO} = 11\%$ ($\sim 6.3°$);
- landing slopes $\alpha_{min,LD} = 1.6\%$ ($\sim 0.92°$), $\alpha_{max,LD} = 4.2\%$ ($\sim 2.4°$).

6.2.1 Test 1, generation of 1 SID and 1 STAR, with 6 obstacles (N=2, M=6)

The input data related to the coordinates of the starting and ending points of the routes to design, and the characteristics of the corresponding buffer obstacles are presented in Table 6.3. The characteristics of the obstacles are presented in Table 6.4. The routes generated individually (as the other routes do not exist) by the B&B, as well as the conflicting area, are illustrated in Fig. 6.1.

route	starting point		ending point	buffer obstacle		
	(x_{A_i}, y_{A_i})	H_{A_i}	(x_{B_i}, y_{B_i})	(x_{bi}, y_{bi}, r_{bi})	$(z_{bi_{inf}}, z_{bi_{sup}})$	$t_{\Omega_{bi}}$
SID	(4, -8)	0	(65, 65)	(3, -3, 2.5)	(0, 50000)	1
STAR	(54, -15)	0	(-20, 60)	(58, -10, 2.5)	(0, 50000)	0

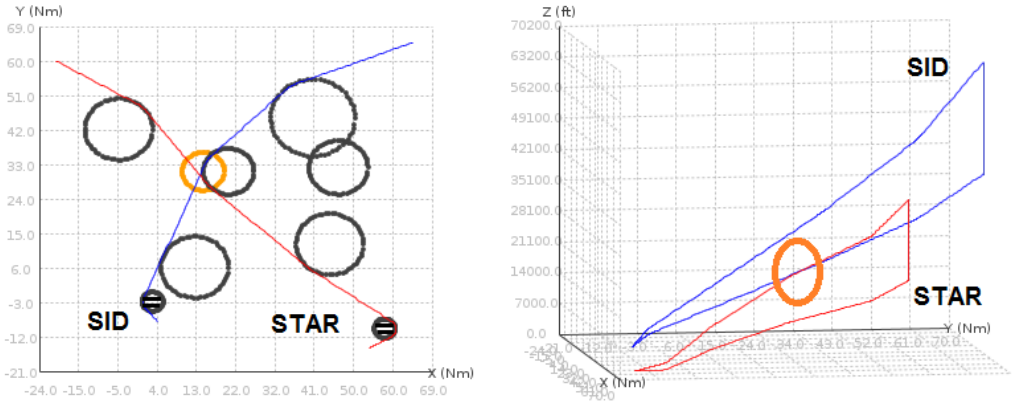
Table 6.3: Test 1: starting and ending points, and corresponding buffer obstacles.

(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
(-5, 42)	8	(0, 50000)
(13, 6)	8	(0, 50000)
(21, 31)	6	(0, 50000)
(41, 45)	10	(0, 50000)
(45, 12)	8	(0, 50000)
(47, 32)	7	(0, 50000)

Table 6.4: Test 1: characteristics of 6 obstacles.

Simulation results of B&B-based method

We design routes according to the following two orders: generating SID before STAR or generating STAR before SID. The simulation results are presented in Fig. 6.2, and the numerical results are given in Table 6.7. All conflicts are removed in both cases, and the computing time is about 0.5s for each. It can be seen in both cases that the second generated route is deviated around a fictitious obstacle whose scale is much larger than the size of the initial conflicting area. The reason for such



(a) (b)

Figure 6.1: Test 1: routes generated independently by B&B and initial conflicting area (orange circle) (a) Illustration in 2D. (b) Illustration in 3D.

a deviation is that the deviated route should avoid not only the conflicting area, but also the obstacle (centered at $(21,31)$(Nm)) close to the initial conflicting area. In both cases, the STAR passes below the SID due to the route deviations. Moreover, the results show that the routes generation order has an impact on the form of optimal routes. In fact, generating STAR before SID introduces less route deviation and provides better solution in terms of total routes length.

Simulation results of SA method

The values of the user-defined parameters in the SA method are empirically determined after several tests. The chosen values for these parameters are given in Table 6.5.

parameter	value
initial temperature, T_0	50
final temperature, T_f	45
temperature cool-down factor, β	0.95
number of iterations at each temperature stage, N_I	15
probability for selecting fictitious obstacle avoidance strategy, ρ_1	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_2	0.8
probability for selecting fictitious obstacle avoidance strategy, ρ_3	0.9
probability for selecting fictitious obstacle avoidance strategy, ρ_4	1
probability for eliminating a non-active fictitious obstacle, ρ_5	0.7
weight coefficient in (5.7), c_3	100

Table 6.5: Test 1: empirically-determined SA parameters.

We run the proposed SA method 10 times with the same input parameters. Table 6.6 shows the minimum/maximum/average value of total length of all obtained routes ($\sum_{i=1}^{2} L_{\gamma_i^*}$) and total length of conflicting route sections ($\sum_{i=1}^{2} \ell^*_{i_{cflt}}$) within the 10 simulations. All conflicts are removed in the 10 simulations, and the average computing time for each simulation is about 1.9s. Figure 6.3 illustrates

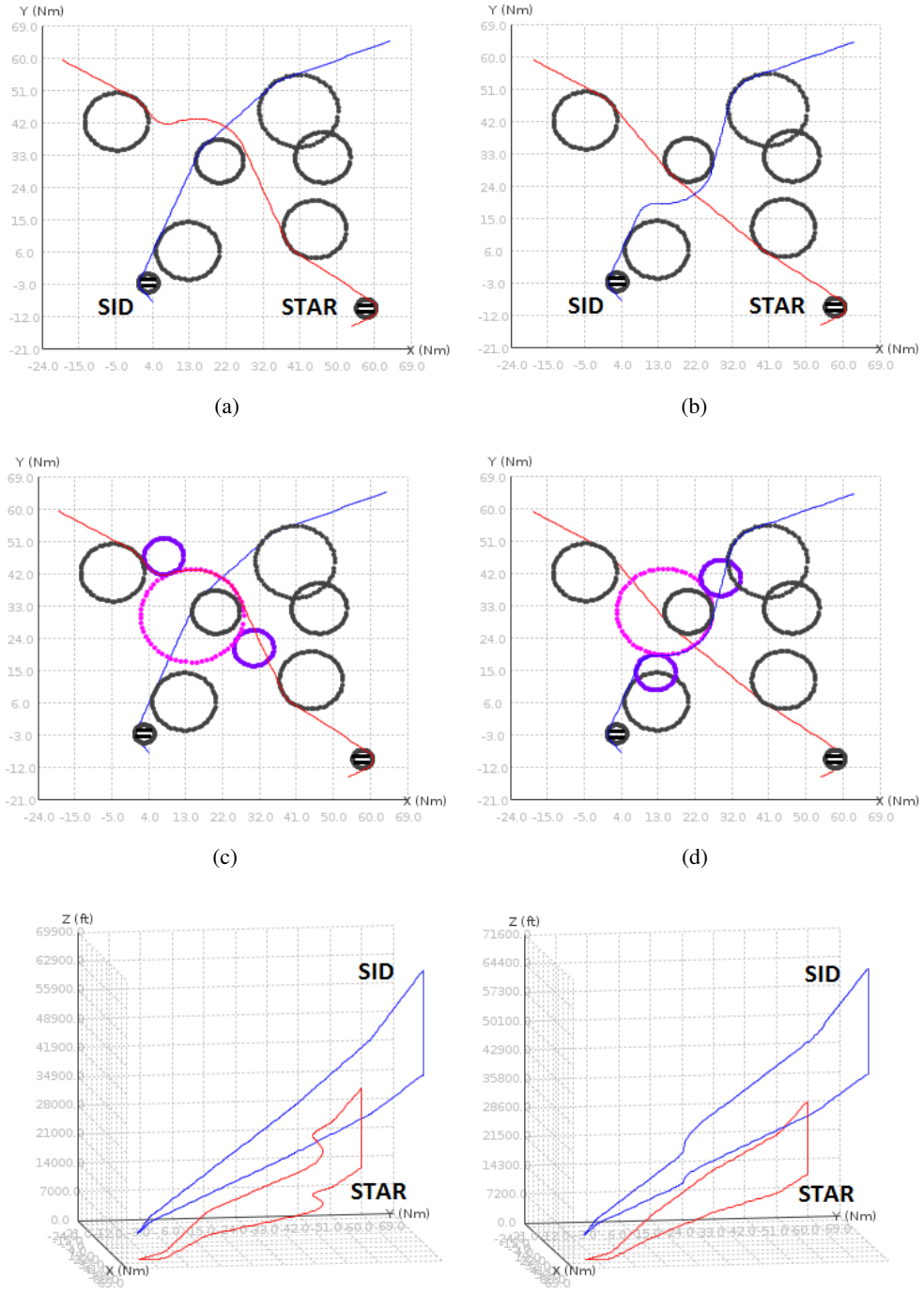


Figure 6.2: Test 1: B&B-based method simulation results. (a) SID generated before STAR, final result. (b) STAR generated before SID, final result. (c) SID generated before STAR, illustration of fictitious and smoothing obstacles. (d) STAR generated before SID, illustration of fictitious and smoothing obstacles. (e) SID generated before STAR, illustration in 3D.(f) STAR generated before SID, illustration in 3D.

the simulation result corresponding to the solution with the least total routes length ($\sum_{i=1}^{2} L_{\gamma_i^*}$). The STAR is imposed by a level flight, so that the conflict is solved. The detailed numerical results related to this solution is given in Table 6.7. The SA method provides a better solution than the B&B-based method in terms of total routes length.

Another solution provided by the SA method is illustrated in Fig. 6.4.In this solution, no fictitious obstacle is used for route deviation. However, the form of the STAR is deviated by modifying the bypassing direction on obstacle centered at $(21,31)$(Nm) from clockwise (in the initial solution, as shown in Fig. 6.1(a)) to counter-clockwise (as shown in Fig. 6.4(a)). By doing so, the altitude of the SID at the intersection with the STAR is higher than the one in the initial solution, so that the STAR passes below the SID without any conflict (as shown in Fig. 6.4(b)). The deviation on the STAR is similar to the one provided by the B&B-based method when the SID is generated before the STAR (Fig. 6.2(a)), but the total routes length of this solution is 225.15Nm, which is shorter than the B&B-based solutions.

	$\sum_{i=1}^{2} L_{\gamma_i^*}$	$\sum_{i=1}^{2} \ell^*_{i_{cflt}}$
Minimum value	222.22	0
Maximum value	225.15	0
Average value	223.39	0
Standard deviation	1.435	0

Table 6.6: Test 1: statistics of SA method.

Test 1	$L_{\gamma_i^0}$	B&B-based method				SA method		
		build SID first		build STAR first				
		$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$
SID	105.66	0	105.66	7.5	109.8	7.5	105.66	0
STAR	116.56	7.09	123.48	0	116.56	7.09	116.56	15.65
Total	222.22	7.09	229.14	7.5	226.36	14.59	222.22	15.65

Table 6.7: Test 1: numerical results of B&B-based method and SA method.

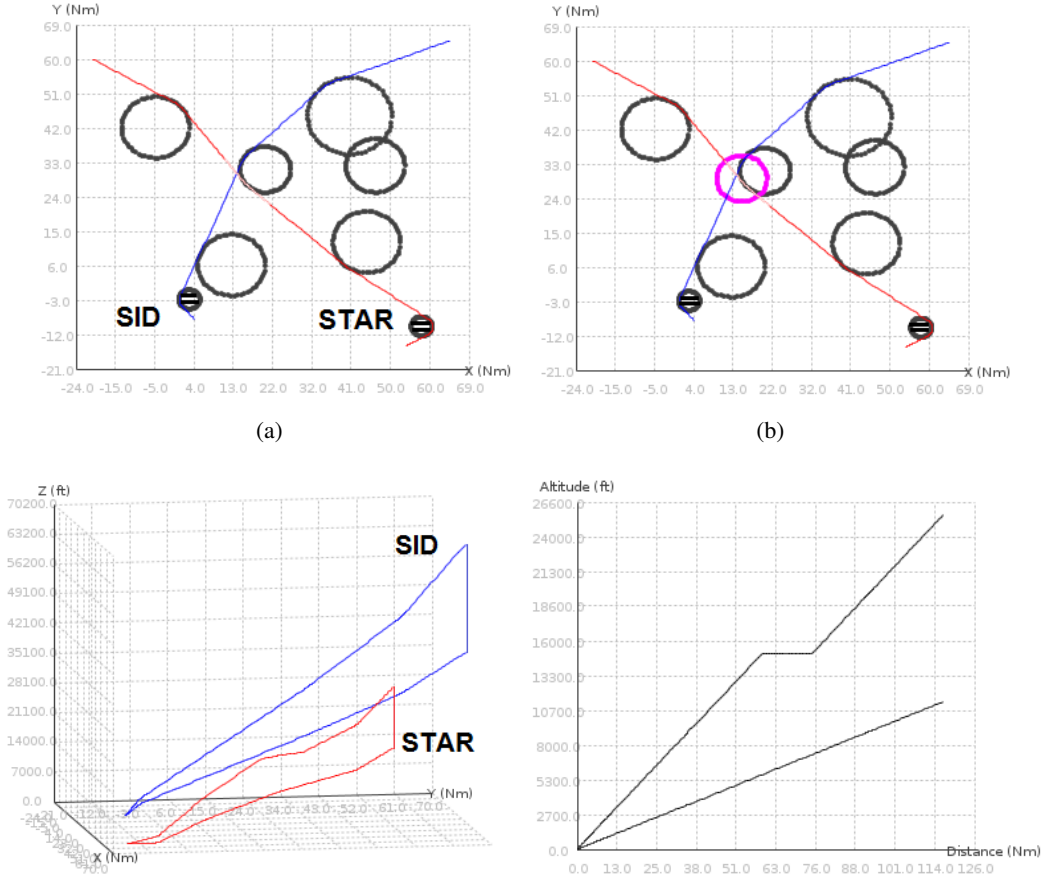


Figure 6.3: Test 1: SA method simulation result, the routes with the least total routes length. (a) Illustration in 2D. (b) Fictitious obstacle for imposing level flight. (c) Illustration in 3D. (d) Vertical profile of the STAR.

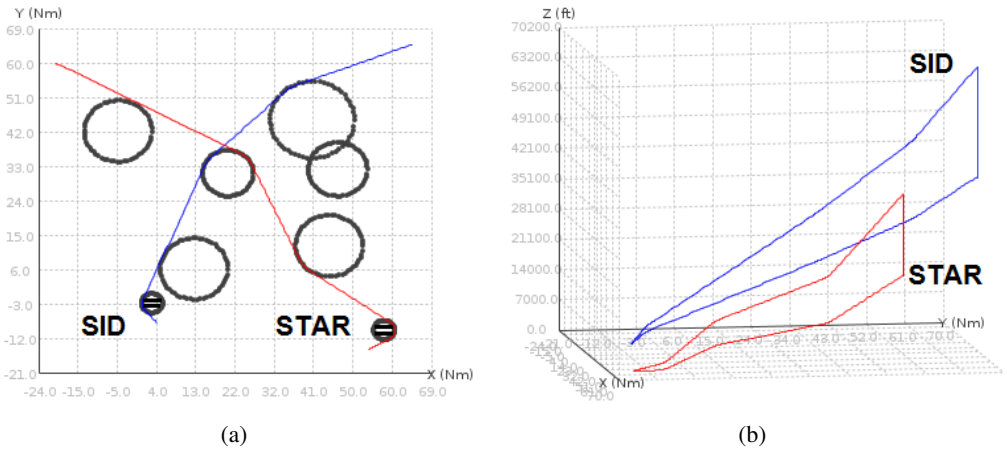


Figure 6.4: Test 1: another solution of the SA method. (a) Illustration in 2D. (b) Illustration in 3D.

6.2.2 Test 2, generation of 2 SIDs and 3 STARs, with 9 obstacles (N=5, M=9)

In order to test our methods in a more complex situation, the number of routes to design as well as the number of obstacles are increased. The input data related to the traffic load (in percentage) on each route to design, the coordinates of the starting and ending points, and the characteristics of the corresponding buffer obstacles are presented in Table 6.8. The characteristics of obstacles are presented in Table 6.9. The routes generated individually by the B&B, as well as the conflicting areas, are illustrated in Fig. 6.5.

γ_i–SID/STAR–traffic load	starting point		ending point	buffer obstacle		
	(x_{A_i}, y_{A_i})	H_{A_i}	(x_{B_i}, y_{B_i})	(x_{bi}, y_{bi}, r_{bi})	$(z_{bi_{inf}}, z_{bi_{sup}})$	$t_{\Omega_{bi}}$
γ_1 – SID – 30%	(21, 89)	0	(75, -18)	(20, 84, 2.5)	(0, 50000)	0
γ_2 – SID – 25%	(4, -8)	0	(55, 93)	(3, -3, 2.5)	(0, 50000)	1
γ_3 – STAR – 20%	(30, -20)	0	(42, 99)	(35, -18, 2.5)	(0, 50000)	0
γ_4 – STAR – 15%	(65, 66)	0	(-12, 66)	(62, 72, 2.5)	(0, 50000)	0
γ_5 – STAR – 10%	(-15, 50)	0	(51, -11)	(-12, 43, 2.5)	(0, 50000)	0

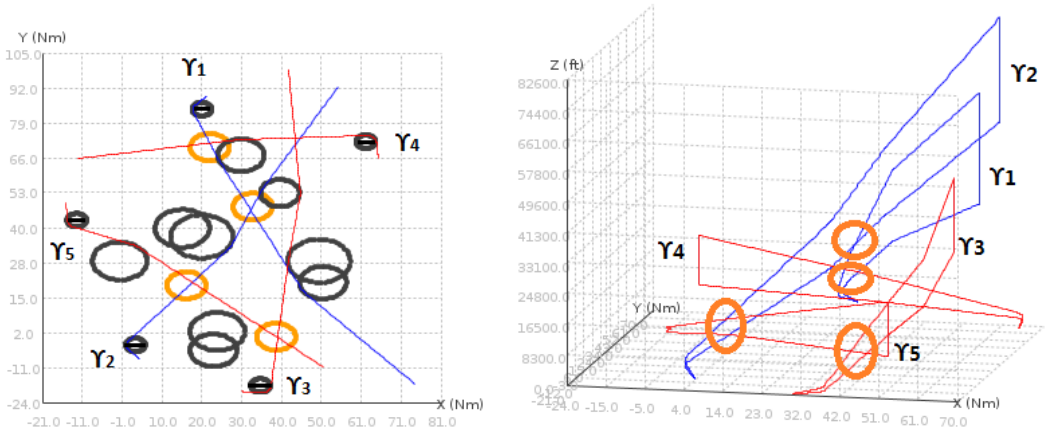
Table 6.8: Test 2: starting and ending points, and corresponding buffer obstacles of the routes.

(x_i, y_i)	r_i	$(z_{i_{inf}}, z_{i_{sup}})$
(-1, 28)	7	(0, 50000)
(15, 40)	7	(0, 50000)
(20, 37)	8	(0, 50000)
(23, -5)	6	(0, 50000)
(24, 2)	7	(0, 50000)
(30, 67)	6	(0, 50000)
(40, 53)	5	(0, 50000)
(50, 28)	8	(0, 50000)
(51, 20)	6	(0, 50000)

Table 6.9: Test 2: characteristics of nine obstacles.

Simulation results of B&B-based method

The routes are generated according to the decreasing order of their traffic load. An illustration of the simulation results in the horizontal plane is presented in Figs. 6.6, where the fictitious and smoothing obstacles on each route are also shown. Figure 6.7 shows the simulation results in 3D. The numerical results are presented in Table 6.12. All conflicts are solved in about 1.1s. Route γ_2 is initially in conflict with γ_1, thus it is deviated clockwise, in order to pass above γ_1 (Fig. 6.6(b)). Route γ_3 is not in conflict with any previously generated route, thus it is accepted directly (Fig. 6.6(c)). Route γ_4 is imposed a level flight, so that it passes below γ_3 and γ_1 (Fig. 6.6(d)). The vertical profile of γ_4 is illustrated in Fig. 6.6(e). Route γ_5 consists of two successive horizontal deviations in order to pass below γ_2 and above γ_3 (Fig. 6.6(f)).



(a) (b)

Figure 6.5: Test 2: routes generated independently by B&B and initial conflicting areas (orange circles). (a) Illustration in 2D. (b) Illustration in 3D.

Simulation results of SA method

The empirically determined values for the SA parameters are given in Table 6.10.

parameter	value
initial temperature, T_0	50
final temperature, T_f	5
temperature cool-down factor, β	0.95
number of iterations at each temperature stage, N_I	20
probability for selecting fictitious obstacle avoidance strategy, ρ_1	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_2	0.8
probability for selecting fictitious obstacle avoidance strategy, ρ_3	0.9
probability for selecting fictitious obstacle avoidance strategy, ρ_4	1
probability for eliminating a non-active fictitious obstacle, ρ_5	0.7
weight coefficient in (5.7), c_3	100

Table 6.10: Test 2: empirically-determined SA parameters.

We run the proposed SA method 10 times with the same input parameters. The statistics of the obtained solutions are given in Table 6.11. All conflicts are removed in the 10 simulations, and the average computing time for each simulation is about 81s. Figure 6.8 illustrates the simulation result corresponding to the solution with the least total routes length. The detailed numerical results related to this solution is given in Table 6.7. To explain the routes deviation more clearly, the fictitious obstacles shown in Fig. 6.8(b) are numbered from 1 to 4. Route γ_2 is not deviated compared to its initial form. A level flight is imposed to γ_1 below fictitious obstacle 1, so that the conflict between γ_1 and γ_2 is solved. Route γ_5 is deviated counter-clockwise around fictitious obstacle 2, so the conflict between γ_2 and γ_5 is solved. A level flight below fictitious obstacle 3 is imposed to γ_3, so that γ_3 passes below γ_5 without conflict. The vertical profile of this level flight is shown in Fig. 6.8(e). On γ_4, a level flight is imposed under fictitious obstacle 4, so that γ_4 has no conflict with the other routes. The total length of all optimal routes obtained by the SA method is shorter than the one of

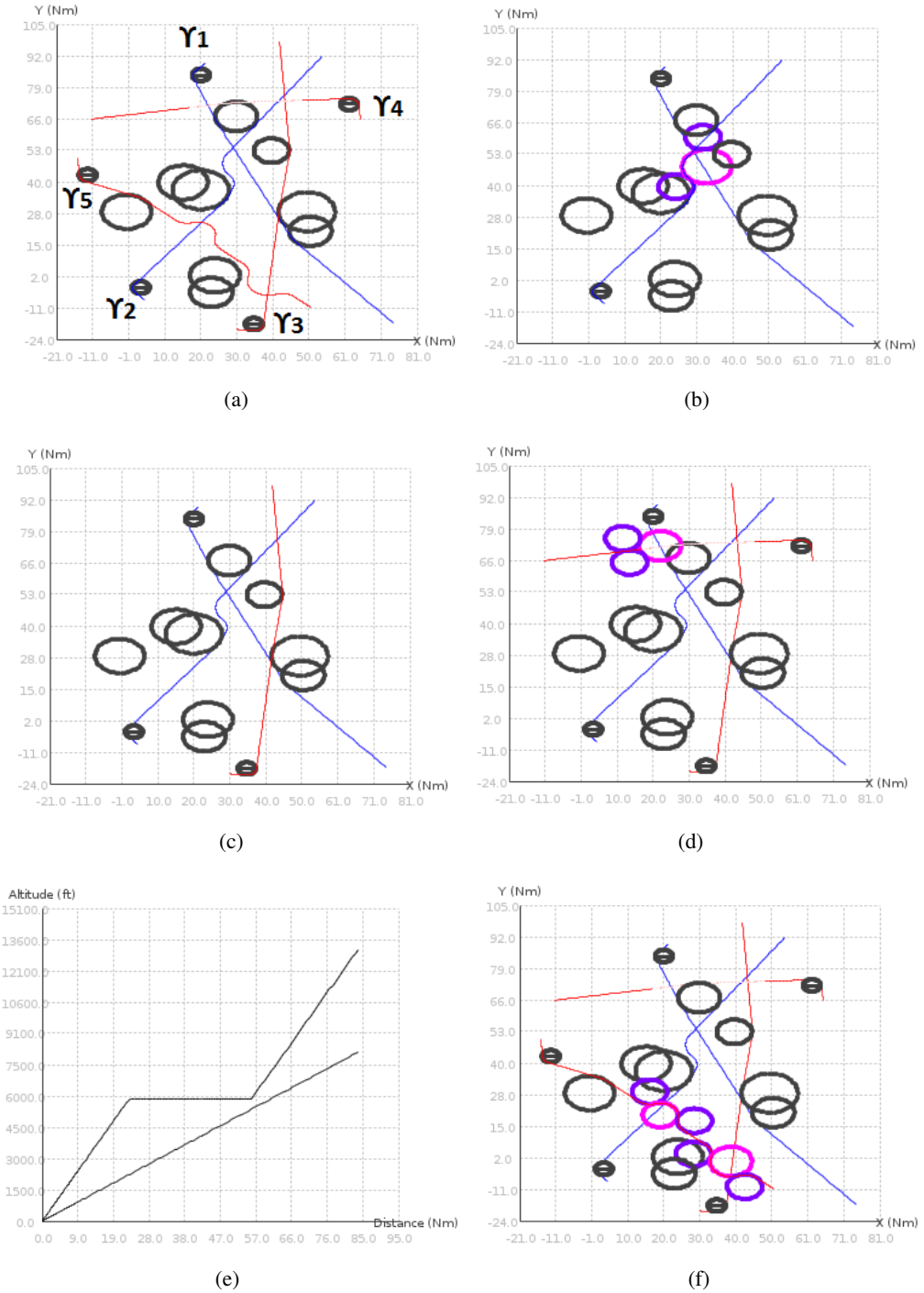


Figure 6.6: Test 2: B&B-based method simulation results. (a) Optimal routes illustration in 2D. (b) Deviation of γ_2. (c) Preservation of route γ_3. (d) Deviation of γ_4. (e) Deviation of γ_4, illustration in the vertical plane. (f) Deviation of γ_5.

the B&B-based method, but the total length of level flights is longer in the solution of SA method. The B&B-method is faster than the SA method in terms of computing time.

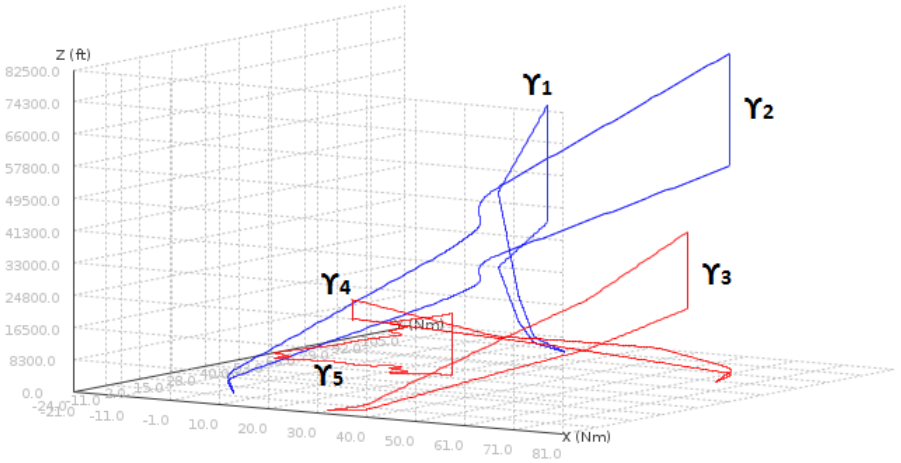


Figure 6.7: Test 2: B&B-based method simulation results, illustration in 3D.

	$\sum_{i=1}^{2} L_{\gamma_i^*}$	$\sum_{i=1}^{2} \ell^*_{i_{cflt}}$
Minimum value	547.98	0
Maximum value	552.1	0
Average value	549.13	0
Standard deviation	1.56	0

Table 6.11: Test 2: statistics of SA method.

Test 2	$L_{\gamma_i^0}$	B&B-based method			SA method		
		$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$	$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$
γ_1 – SID	124.59	0	124.59	0	16.24	126.3	10.49
γ_2 – SID	117.32	9.06	121.02	0	10.21	117.32	0
γ_3 – STAR	126.45	0	126.45	0	7.76	126.45	15.54
γ_4 – STAR	84.56	6.72	84.56	32.77	7.32	84.57	33.11
γ_5 – STAR	92.99	9.4	101.48	0	9.79	93.34	0
Total	545.91	25.18	558.1	32.77	51.32	547.98	59.14

Table 6.12: Test 2: numerical results of B&B-based method and SA method.

6.3 Design of multiple routes in the TMA of Paris CDG airport

We now test the proposed methods on the design of multiple routes in a real TMA, namely the TMA of Paris CDG airport. We use the radar data of real traffic in this TMA during one day. The simulation results are then compared with the standard published SIDs and STARs.

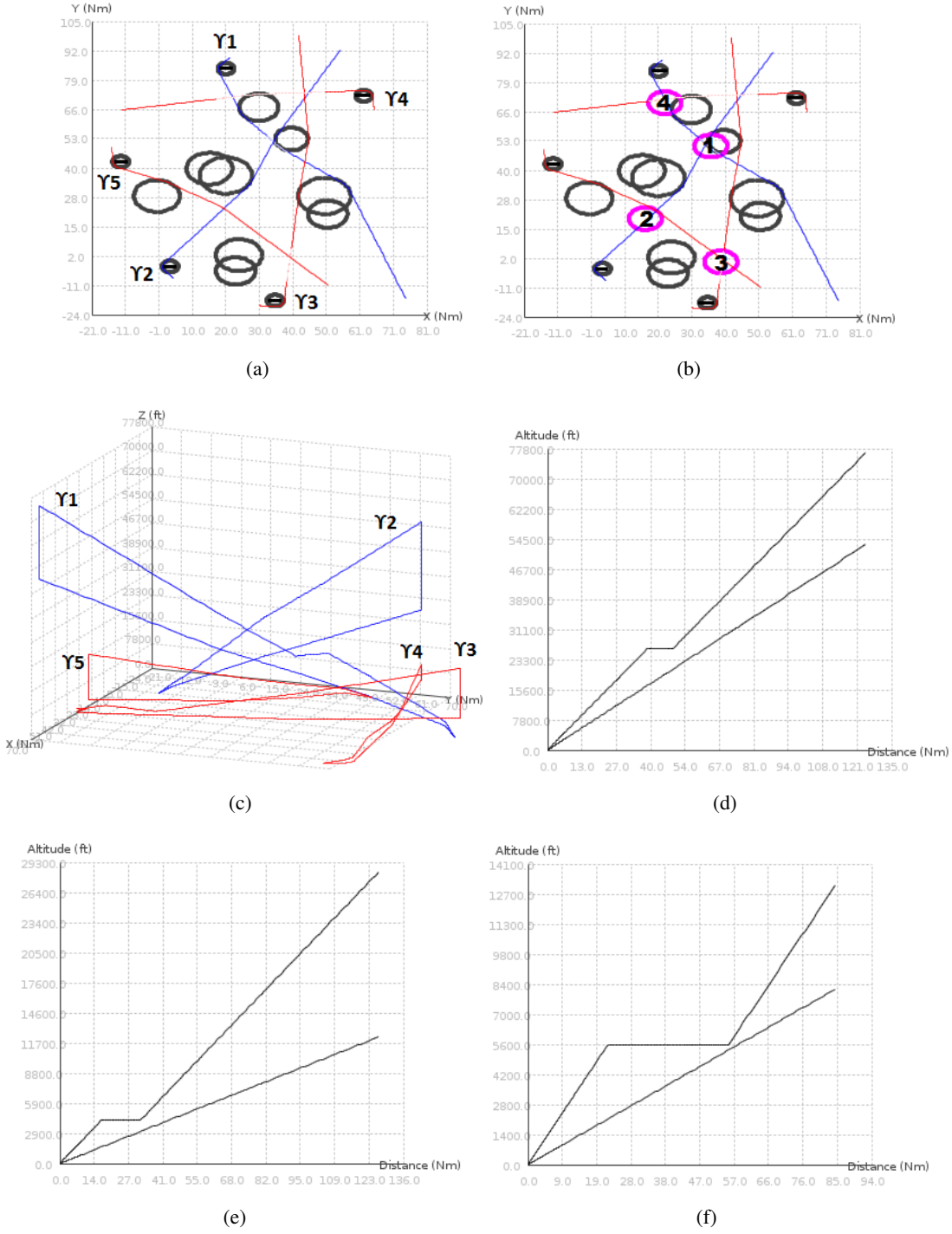
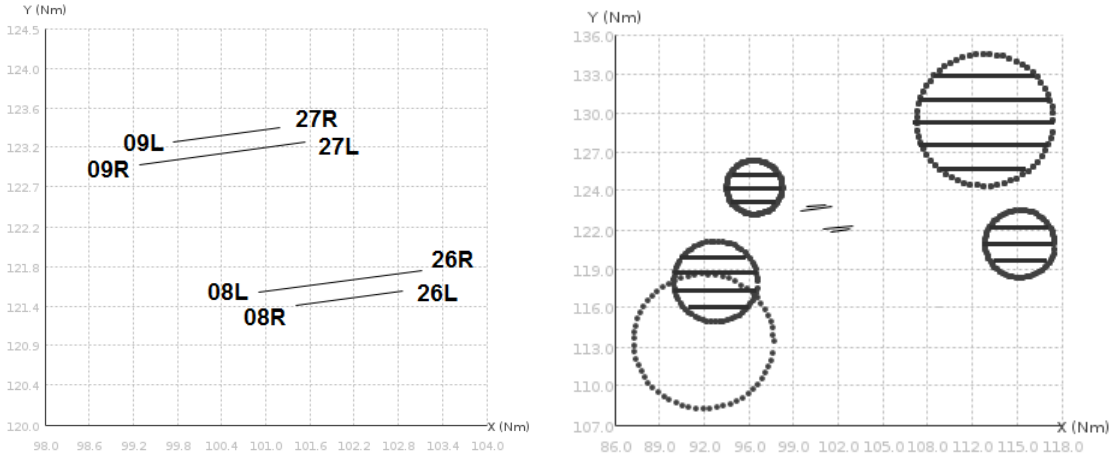


Figure 6.8: Test 2: SA method simulation result. (a) Optimal routes, illustration in 2D. (b) Fictitious obstacles for route deviation. (c) Optimal routes, illustration in 3D. (d) Vertical profile of γ_1. (e) Vertical profile of γ_3. (f) Vertical profile of γ_4.

Runway configuration

CDG airport has four parallel runways, as shown in Fig. 6.9(a), where the numbering of the 8 thresholds (two thresholds per runway) are also illustrated. In a general case, two runways are used

for take-off, the other two are used for landing. The radar data that we use for testing correspond to a day when the runway thresholds used for take-off are 09R and 08L, and the runway thresholds used for landing are 27R and 26L. Thus we take the same thresholds for take-off and landing in our simulation. The detailed information about the thresholds used for take-off and landing in the simulation, together with their coordinates and altitudes are presented in Table 6.13. In the case of landing threshold, the coordinates and altitudes of the associated FAFs are given additionally.



(a) (b)

Figure 6.9: Paris CDG airport configuration. (a) Four parallel runways. (b) Buffer obstacles and a real obstacle.

threshold	take-off/landing	threshold		FAF	
		coordinate	altitude	coordinate	altitude
09R	take-off	(99.28, 122.95)	370		
08L	take-off	(100.9, 121.5)	338		
27R	landing	(101.19, 123.38)	392	(111.12, 124.51)	3392
26L	landing	(102.86, 121.52)	316	(112.8, 122.66)	3316

Table 6.13: Test Paris CDG airport: characteristics of the thresholds, and the FAFs associated with STARs.

One day radar data

The used radar data are illustrated in Fig. 6.10, where the real traffic arriving to and departing from Paris CDG airport during one day is presented. The traffic in light blue (respectively, red) color represents the departure (respectively, arrival) flights. It can be seen that the departure and arrival routes are located alternately in order to decrease the interaction between them. Moreover, from the radar data, we notice that there are mainly two conflicting areas. One occurs between the arrival flows from the north-west and the departure flows to the north, the other one occurs between the arrival flows from the south-west and the departure flows to the south. In order to avoid conflicts in real operation, the arrival flights from the north-west maintain their flight level at about 12000ft

on the route section between MERUE and CREIL, so that the departure flights to the north can pass below; similarly, the arrival flights from the south-west maintain their flight level at about 13000ft on the route section between DOMUS and PG515, and the departure flights to the south pass below them.

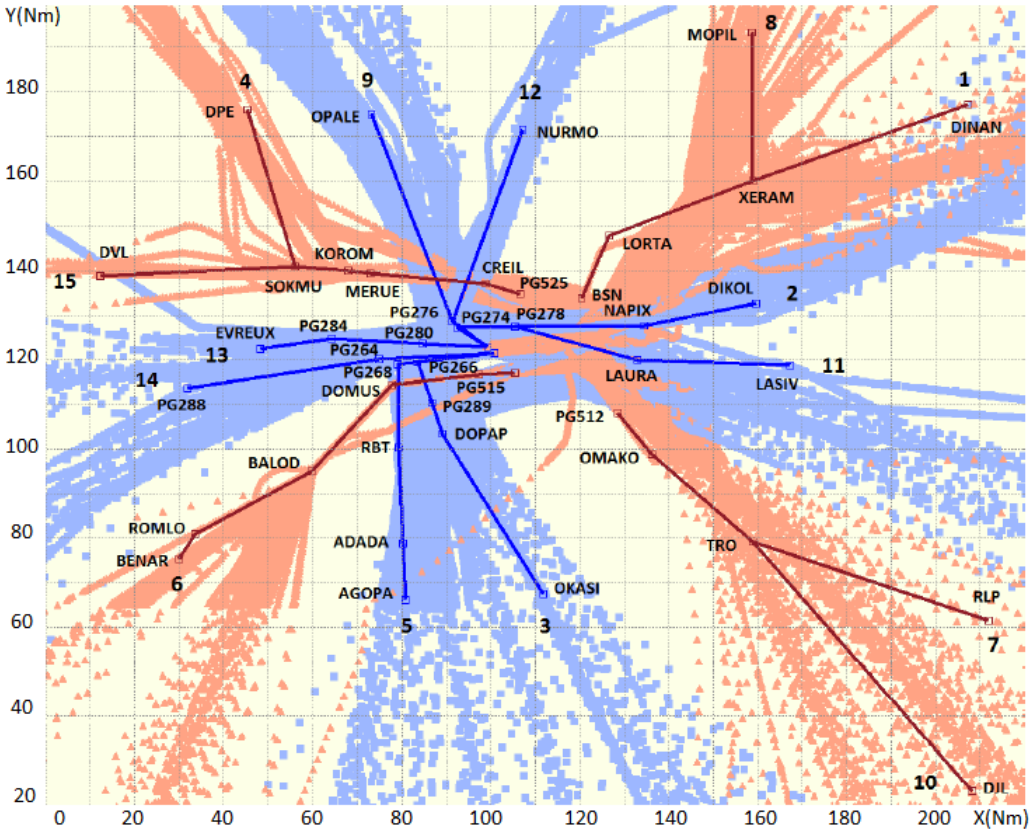


Figure 6.10: One day radar data of Paris CDG airport, and principal SIDs and STARs.

Selected standard routes

Based on the radar data, we select 15 principal routes for jet aircraft (including 8 SIDs and 7 STARs) from the published Jeppesen charts [10]. These standard routes, as well as the related waypoints, are also presented in Fig. 6.10, where the dark blue color represents the 8 SIDs and dark red color represents the 7 STARs. In order to compute the traffic load on each principal route, we define an exit (respectively, entry) TMA window for each exit (respectively, entry) point on a SID (respectively, STAR). The traffic load on each principal route is counted by adding up the number of flights passing through the corresponding exit or entry window. The information on the selected standard routes, including whether it is a SID or a STAR, the threshold in use, the related traffic load (in percentage), and the bypassing waypoints are presented in Table 6.14. Moreover, in Table 6.14 and Fig. 6.10, the selected routes are numbered according to the decreasing order of their traffic loads.

route	SID/STAR	threshold	traffic load	Waypoints
1	STAR	27R	12.77%	DINAN, XERAM, LORTA, BSN
2	SID	09R	10.82%	PG274, PG278, NAPIX, DIKOL
3	SID	08L	10.72%	PG266, PG289, DOPAP, OKASI
4	STAR	27R	8.7%	DPE, SOKMU, KOROM, MERUE, CREIL, PG525
5	SID	08L	7.44%	PG268, RBT, ADADA, AGOPA
6	STAR	26L	7.33%	BENAR, ROMLO, BALOD, DOMUS, PG515, PG516
7	STAR	26L	7.16%	RLP, TROY, OMAKO, PG512
8	STAR	27R	6.97%	MOPIL, XERAM, LORTA, BSN
9	SID	09R	5.63%	PG276, OPALE
10	STAR	26L	5.61%	DJL, TRO, OMAKO, PG512
11	SID	09R	4.9%	PG274, PG278, LAURA, LASIV
12	SID	09R	4.76%	PG276, NURMO
13	SID	09R	3.38%	PG280, PG284, EVREUX
14	SID	08L	2.33%	PG264, PG288
15	STAR	27R	1.48%	DVL, SOKMU, KOROM, MERUE, CREIL, PG525

Table 6.14: Test Paris CDG airport: traffic load and waypoints of the selected standard routes.

Specific input data of the test

In this test, we consider the design of 15 routes (denoted as $\gamma_i, i = 1, \cdots, 15$), whose starting and ending points, as well as the traffic load, correspond to the ones of the 15 standard principal routes that we select. More precisely, in a SID case, the starting and ending points of γ_i are the same as the ones of the standard route i, while in a STAR case, the ending point of γ_i corresponds to the TMA entry point on the standard route i, and the starting point is the FAF of the corresponding runway threshold. Moreover, each route to design is associated with a buffer obstacle, as described in Section 3.3.2. In this test, we use the same buffer obstacle for the routes using the same runway. The TMA of CDG airport is located in a relatively simple geographical environment, the only area to be avoided is the Paris city. This area is modeled as an obstacle. Thus $M = 1$ in our simulation, and the characteristics of this obstacle are $(x_1, y_1, r_1, z1_{inf}, z1_{sup}) = (92.43\text{Nm}, 113.16\text{Nm}, 5\text{Nm}, 0, 50000\text{ft})$. The buffer obstacles as well as the obstacle representing Paris city are illustrated in Fig. 6.9(b). The starting and ending points, and characteristics of the buffer obstacles corresponding to each route to design are presented in Table 6.15.

In this test, the values of the take-off and landing slopes are:

- taking-off slopes $\alpha_{min,TO} = 7\%$ $(\sim 4^\circ)$, $\alpha_{max,TO} = 11\%$ $(\sim 6.3^\circ)$;
- landing slopes $\alpha_{min,LD} = 1.6\%$ $(\sim 0.92^\circ)$, $\alpha_{max,LD} = 4.2\%$ $(\sim 2.4^\circ)$.

In fact, the slopes for take-off and landing may vary according to different routes and days, as they can be affected by several factors, such as airport surrounding environment (mountainous area or free area), wind condition, performances of aircraft flying on one route, etc. In general, aircraft engine is at full thrust in take-off and climb phases, which leads to a relatively steep slope. On

the contrary, in the approaching and landing phases, the engine power is reduced, aircraft speed decreases gradually, so that the corresponding slopes is relatively steady.

We note that, according to the way one route is built, routes using the same buffer obstacle merge consecutively at the border of the corresponding buffer obstacle. In this case, these routes are not separated in the area surrounding the buffer obstacle. This is also what happens in the real-world operations, and it is the air traffic controllers who take charge of aircraft sequencing at real time. In our test, we allow the possibility of routes merging at the border of buffer obstacles. Consequently, we propose a circular-shaped area surrounding each buffer obstacle, where conflicts are not detected between the routes merging at that buffer obstacle. This area has the same center as the corresponding buffer obstacle, and its radius is a user-defined value, depending on the scale of the considered airport and the number of the merging routes. After obtaining the optimal routes, if any two routes using the same buffer obstacle intersect each other, they can be corrected manually to merge together. In this test, the radius of the circular area surrounding a buffer obstacle is set to 15Nm plus the radius of the corresponding buffer obstacle.

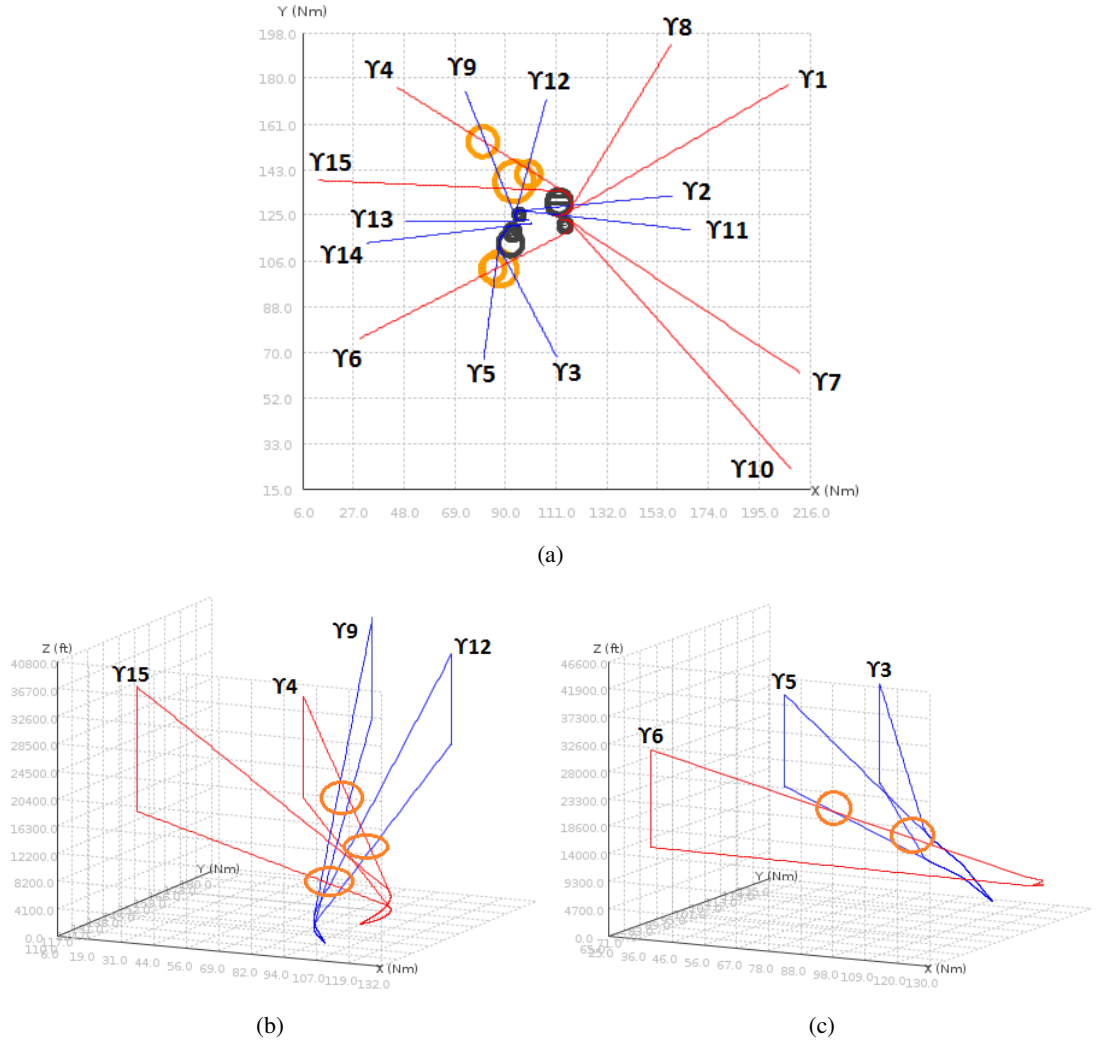
γ_i	threshold	starting point		ending point	buffer obstacle	
		(x_{A_i}, y_{A_i})	H_{A_i}	(x_{B_i}, y_{B_i})	$(x_{bi}, y_{bi}, r_{bi}, z_{bi_{inf}}, z_{bi_{sup}})$	$t_{\Omega_{bi}}$
γ_2	09R	(99.28, 122.95)	370	(159.87, 132.61)	(96.07, 124.59, 2, 0, 50000)	1
γ_9				(73.18, 175.07)		
γ_{11}				(167.34, 118.76)		
γ_{12}				(107.11, 171.69)		
γ_{13}				(48.26, 122.38)		
γ_3	08L	(100.9, 121.5)	338	(111.77, 67.03)	(93.29, 117.61, 3, 0, 50000)	0
γ_5				(80.87, 66.08)		
γ_{14}				(31.89, 113.55)		
γ_1	27R	(111.12, 124.51)	3392	(207.18, 177.27)	(112.55, 129.61, 4.9, 0, 50000)	0
γ_4				(45.4, 176.02)		
γ_8				(158.66, 193.36)		
γ_{15}				(12.31, 138.75)		
γ_6	26L	(112.8, 122.66)	3316	(29.88, 75.39)	(115.07, 120.4, 2.5, 0, 50000)	1
γ_7				(211.86, 61.57)		
γ_{10}				(208.23, 23.08)		

Table 6.15: Starting and ending points, and corresponding buffer obstacles of the routes to design.

Routes generated independently by the B&B

The routes generated independently by the B&B method are shown in Fig. 6.11, the conflicting areas are in orange color. The length on each route in the conflicting areas is given in Table 6.18. Since only the obstacle corresponding to Paris city is to be avoided, the obtained routes are basically in form of straight line segments. Moreover, The presented results show that two main conflicting zones appear in the simulation (Fig. 6.11(a)). One conflicting area is between departure routes γ_9, γ_{12} and arrival routes γ_4, γ_{15} (Fig. 6.11(b)), the other is between departure routes γ_3, γ_5 and arrival

route γ_6 (Fig. 6.11(c)). This observation on the conflicting areas is coherent with our analysis on the real radar data.



(a)

(b)

(c)

Figure 6.11: Test Paris CDG airport: routes generated independently by B&B and initial conflicting areas (orange circles). (a) Illustration in the horizontal plane. (b) Illustration of γ_4, γ_9, γ_{12} and γ_{15} in 3D. (c) Illustration of γ_3, γ_5 and γ_6 in 3D.

Simulation results of B&B-based method

Routes are built sequentially according to the decreasing order of the traffic load. The simulation result is presented in Fig. 6.12, and the numerical results are given in Table 6.18. All conflicts are removed and the computing time is about 9.8s. Concerning the conflicts among departure routes γ_9, γ_{12} and arrival routes γ_4, γ_{15} in our simulation, route γ_4 is the first one to be generated among these four routes, since it has a higher traffic load. Routes γ_9 and γ_{12} are generated afterwards. In order to avoid the conflict with γ_4, route γ_9 (respectively, γ_{12}) is deviated clockwise (respectively, counter-clockwise). Then, route γ_{15} is in conflict with both γ_{12} and γ_9. To avoid γ_{12}, a level flight is imposed, and to avoid γ_9, a counter-clockwise bypassing is applied. These four routes are illustrated in 3D in Fig. 6.12(c), and the vertical profile of γ_{15} is illustrated in Fig. 6.12(d). Furthermore, concerning

the departure routes γ_3, γ_5 and arrival route γ_6, route γ_6 is generated after γ_3, γ_5, and a level flight is imposed to γ_6, so as to avoid conflict with previously generated routes. These three routes are illustrated in 3D in Fig. 6.12(e), and the vertical profile of γ_6 is illustrated in Fig. 6.12(f).

Figure. 6.12(b) represents the comparison between the selected standard routes and our simulation result in the horizontal plane. The curves in black color represent the selected standard SIDs and STARs. The routes in blue (respectively, red) color are the optimal SIDs (respectively, STARs) obtained by using the proposed methodology, where the dashed sections represent the level flights. The detailed numerical results are given in Table 6.18. The total length of the 15 selected standard routes is 1358.26Nm, and it is reduced by about 45Nm using the proposed B&B-based approach. Note that, except routes γ_9, γ_{12} and γ_{15}, which are deviated to avoid conflicts, the other routes obtained by the B&B-based approach are all shorter than the standard routes. This is especially true for the routes with relatively heavy traffic load, since they have a priority order in the design and therefore perform straight line segments. The average amount of take-off and landing traffic in CDG airport is about 1300 flights per day. By using the optimal routes obtained in our simulation, taking into account their traffic load, the total flown distance is reduced by 5889Nm per day.

Simulation results of SA method

The empirically-set values of the user-defined parameters in the SA method are given in Table 6.16.

parameter	value
initial temperature, T_0	60
final temperature, T_f	5
temperature cool-down factor, β	0.95
number of iterations at each temperature stage, N_I	20
probability for selecting fictitious obstacle avoidance strategy, ρ_1	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_2	0.8
probability for selecting fictitious obstacle avoidance strategy, ρ_3	0.9
probability for selecting fictitious obstacle avoidance strategy, ρ_4	0.1
probability for eliminating a non-active fictitious obstacle, ρ_5	0.7
weight coefficient in (5.7), c_3	100

Table 6.16: Test Paris CDG airport: empirically-determined SA parameters.

We run the proposed SA method 10 times with the same input parameters. The statistics of the obtained solutions are given in Table 6.17. All conflicts are removed in the 10 simulations, and the average computing time for each simulation is about 1200s. Figures 6.13, 6.14 illustrate the simulation result corresponding to the solution with the least total routes length ($\sum_{i=1}^{2} L_{\gamma_i^*}$). The detailed numerical results related to this solution is given in Table 6.18. All conflicts are solved by imposing level flights below routes γ_4, γ_6 and γ_{15}. The altitude of the level flight on γ_4 (Fig. 6.14(c)) is higher than the one below γ_{15} (Fig. 6.14(d)), since γ_4 intersects γ_9 and γ_{15} at a higher altitude. According to Table 6.18, the total length of all standard routes is reduced by about 63Nm using the SA method, which is more than the length reduction provided by the B&B-based method. However, the total length of level flights ($\sum_{i=1}^{15} \ell^*_{i_{LF}}$) in the SA method is about 24Nm longer than the one in the B&B-based method. The reason is that in the B&B-based method, if a conflict can be solved

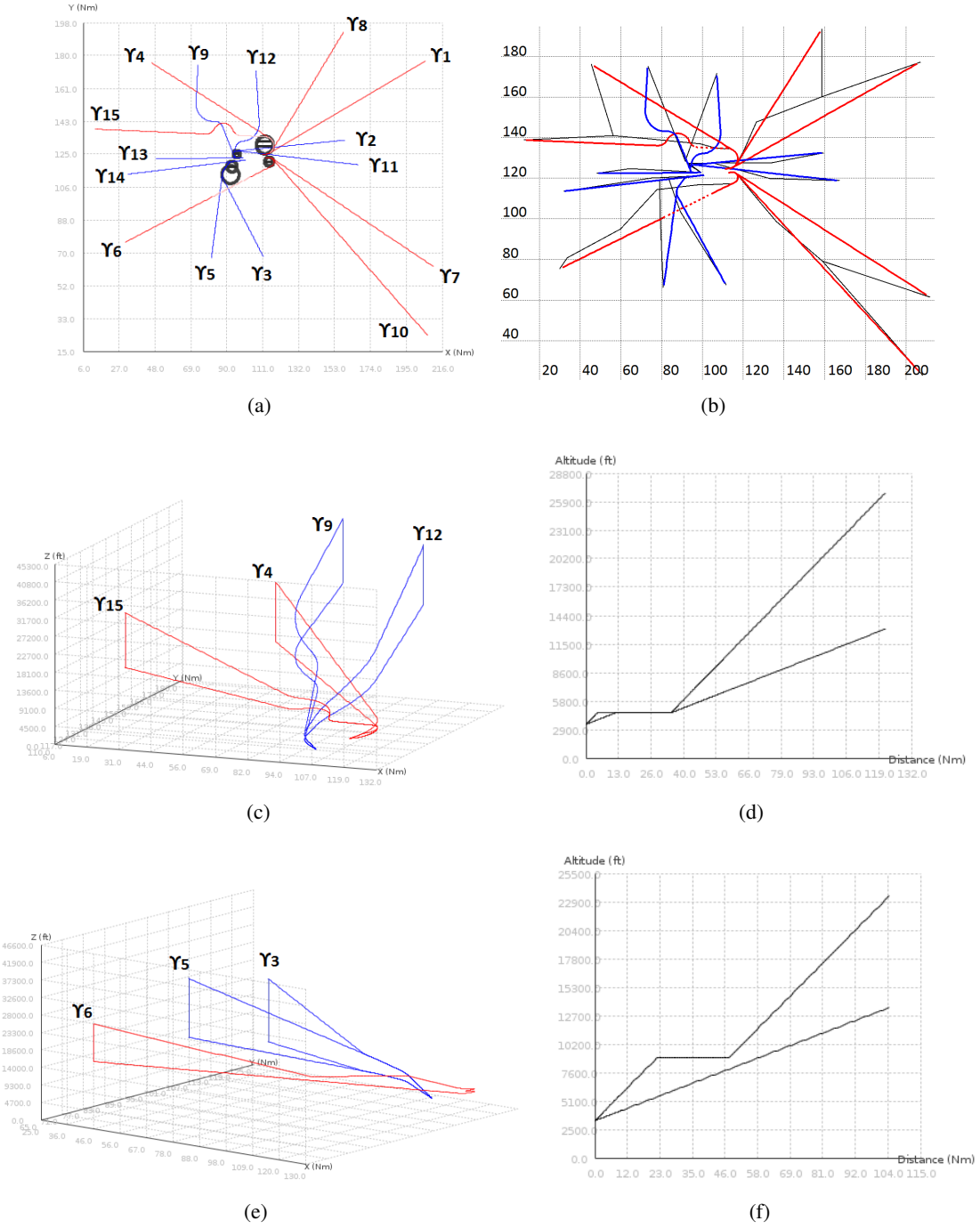


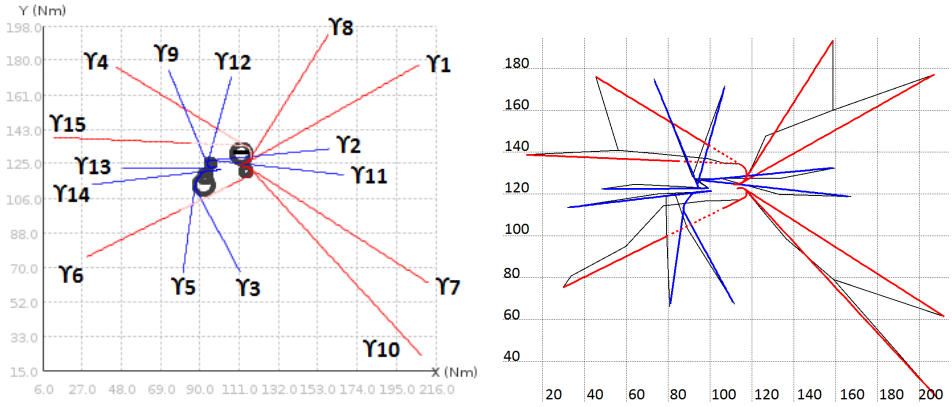
Figure 6.12: Test Paris CDG airport: B&B-based method simulation results. (a) Illustration in the horizontal plane. (b) Comparison with standard routes. (c) Illustration of γ_4, γ_9, γ_{12} and γ_{15}. (d) Illustration of γ_{15} in the vertical plane. (e) Illustration of γ_3, γ_5 and γ_6. (f) Illustration of γ_6 in the vertical plane.

by route deviation, then no more level flight is considered. While in the SA method, for the routes where a level flight is allowed, the level flight has the same chance to be applied as the deviations in the horizontal plane. By using the optimal routes obtained in this solution, taking into account the

average amount of traffic in CDG airport (1300 flights per day) and the traffic load on each route, the total flown distance is reduced by about 6858Nm per day.

	$\sum_{i=1}^{2} L_{\gamma_i^*}$	$\sum_{i=1}^{2} \ell^*_{i_{cflt}}$
Minimum value	1295.51	0
Maximum value	1310.77	0
Average value	1301.696	0
Standard deviation	4.817	0

Table 6.17: Test Paris CDG airport: statistics of SA method.



(a) (b)

Figure 6.13: Test Paris CDG airport: SA method simulation result, the routes with the least total routes length. (a) Optimal routes illustration in the horizontal plane. (b) Comparison with standard routes.

Within the 10 simulations run with the SA method, we also obtain solutions where conflicts are solved by both level flights and horizontal deviations. An example is illustrated in Figs. 6.15, 6.16. In this solution, to avoid the conflicts between routes γ_4, γ_9, γ_{12} and γ_{15}, the two SIDs (γ_9, γ_{12}) are deviated counter-clockwise, and the two STARs (γ_4, γ_{15}) are imposed level flights. Concerning the conflicts between routes γ_3, γ_5 and γ_6, route γ_6 is slightly deviated clockwise and imposed a level flight, as shown in Figs. 6.16(e), 6.16(f). The total routes length of this solution is 1306.29Nm and the total length of level flights is 66.37Nm. Comparing the level fights imposed to γ_6 in the previous solution (Fig. 6.14(f)) and in this solution (Fig. 6.16(f)), the length of the former (respectively, latter) one is 26.87Nm (respectively, 19.54Nm). The reason for a shorter level flight in the latter case is that route γ_6 is deviated clockwise around a fictitious obstacle, thus it intersects route γ_3 at a higher altitude, which implies a level flight with higher altitude and shorter length. By using the optimal routes obtained in this solution, taking into account the average number of traffic in CDG airport (1300 flights per day) and the traffic load on each route, the total flown distance is reduced by about 6091Nm per day. Both solutions provided by the SA method have less total routes length than the one provided by the B&B-based method. However, the B&B-based method has much shorter computing time.

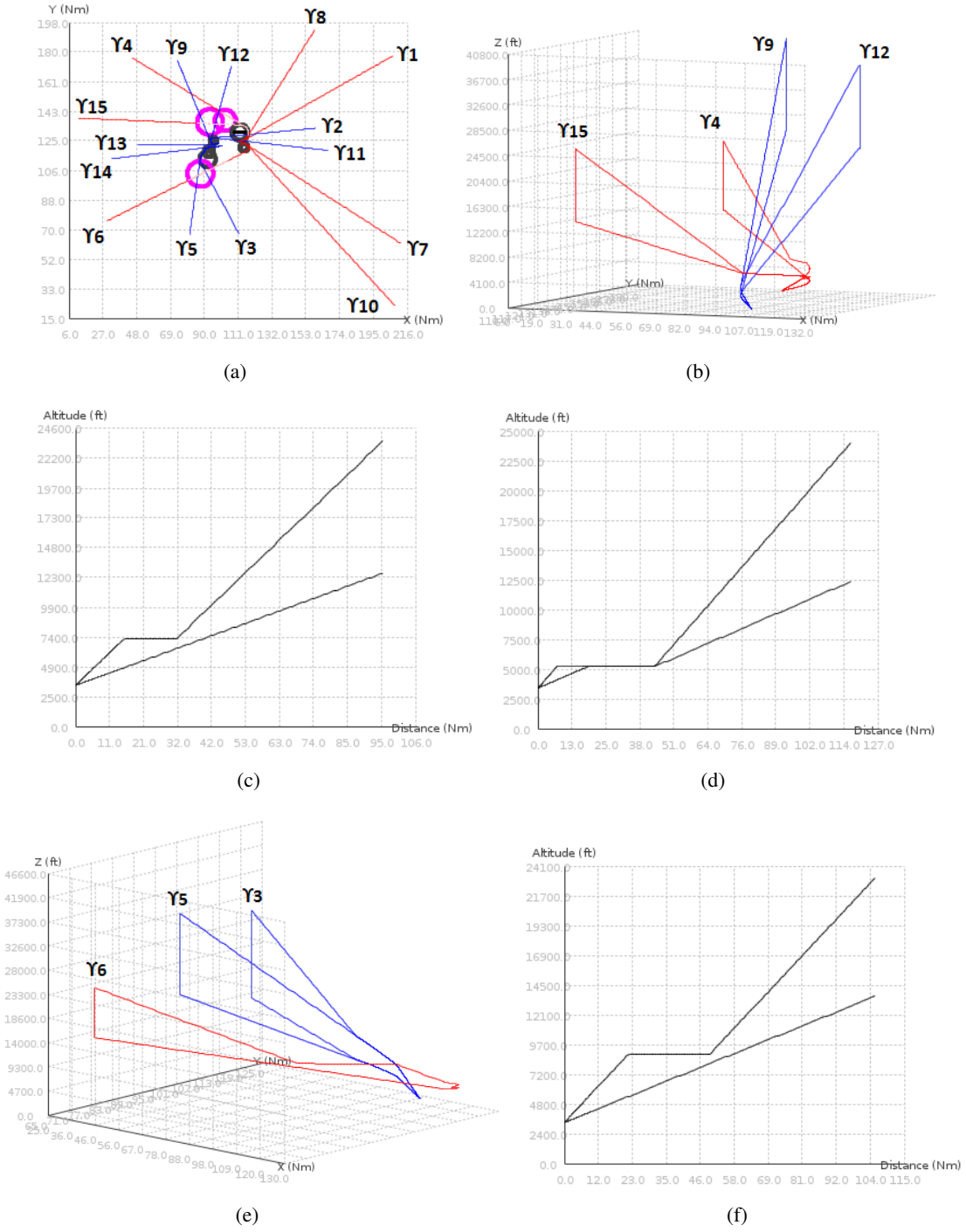
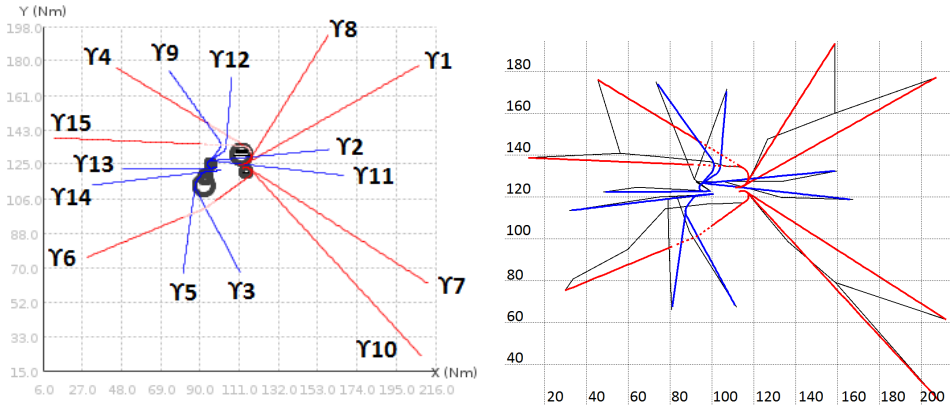


Figure 6.14: Test Paris CDG airport: SA method simulation result, the routes with the least total routes length. (a) Fictitious obstacles for route deviation. (b) Illustration of γ_4, γ_9, γ_{12} and γ_{15}. (c) Illustration of γ_4 in the vertical plane. (d) Illustration of γ_{15} in the vertical plane. (e) Illustration of γ_3, γ_5 and γ_6. (f) Illustration of γ_6 in the vertical plane.

i	L_i^{std}	$L_{\gamma_i^0}$	B&B-based method				SA method			
			$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$	δL_i	$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$	δL_i
1	115.19	109.79	0	109.79	0	5.4	0	109.79	0	5.4
2	75.85	73.58	0	73.58	0	2.27	0	73.58	0	2.27
3	77	69.98	0	69.98	0	7.02	6.19	69.98	0	7.02
4	110.38	95.65	0	95.65	0	14.73	17.39	95.65	16.13	14.73
5	75	65.26	0	65.26	0	9.74	7.71	65.26	0	9.74
6	110.4	105.3	12.44	105.3	25.71	5.1	13.36	105.3	26.87	5.1
7	120.38	116.94	0	116.94	0	3.44	0	116.94	0	3.44
8	97.09	84.89	0	84.89	0	12.2	0	84.89	0	12.2
9	59.62	60.98	10.73	67.69	0	-8.07	17.73	60.98	0	-1.36
10	139.28	139.04	0	139.04	0	0.24	0	139.04	0	0.24
11	84.16	81.2	0	81.2	0	2.96	0	81.2	0	2.96
12	55.51	55.24	6.7	61.36	0	-5.85	13.4	55.24	0	0.27
13	51.25	51.04	0	51.04	0	0.21	0	51.04	0	0.21
14	69.57	69.46	0	69.46	0	0.11	0	69.46	0	0.11
15	117.58	117.16	18.23	122.31	29.91	-4.73	13.94	117.16	36.45	0.42
Total	1358.26	1295.51	48.1	1313.49	55.62	44.77	89.72	1295.51	79.45	62.75

Table 6.18: Test Paris CDG airport: numerical results of B&B-based method and SA method (L_i^{std} is the length of the i^{th} selected standard route, and $\delta L_i = L_i^{std} - L_{\gamma_i^*}$).



(a) (b)

Figure 6.15: Test Paris CDG airport: SA method simulation results. (a) Optimal routes illustration in the horizontal plane. (b) Comparison with standard routes.

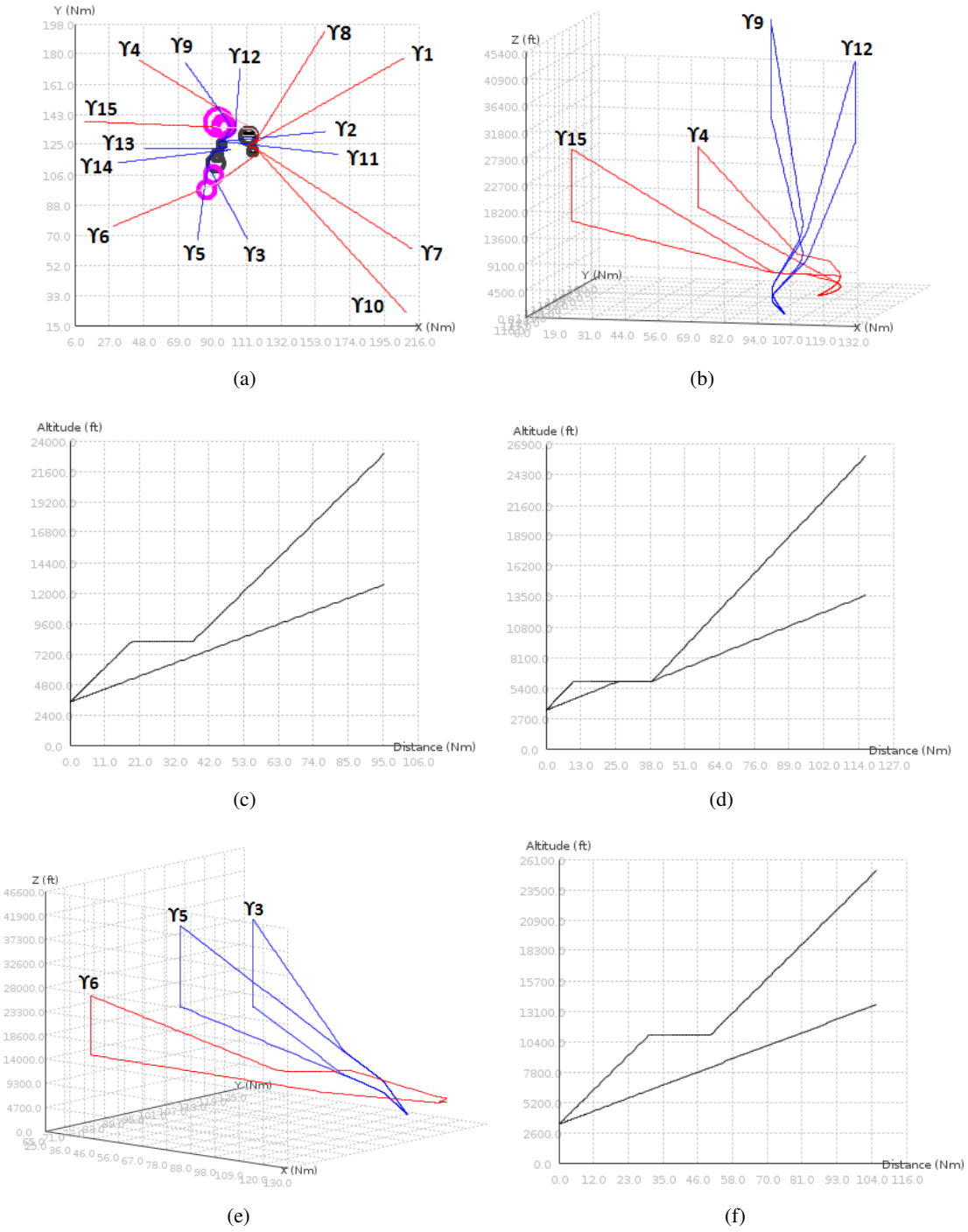


Figure 6.16: Test Paris CDG airport: SA method simulation results. (a) Fictitious obstacles for route deviation. (b) Illustration of γ_4, γ_9, γ_{12} and γ_{15}. (c) Illustration of γ_4 in the vertical plane. (d) Illustration of γ_{15} in the vertical plane. (e) Illustration of γ_3, γ_5 and γ_6. (f) Illustration of γ_6 in the vertical plane.

6.4 Design of multiple routes in the TMA of Zurich airport

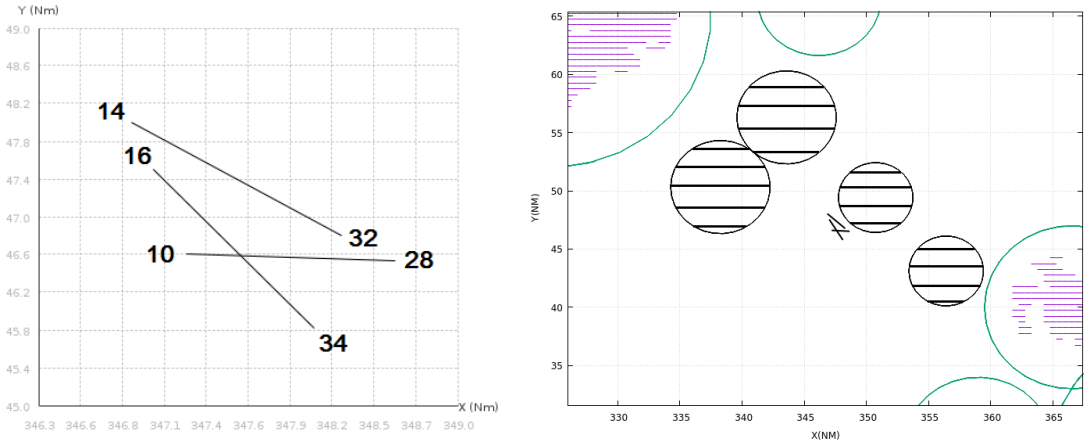
In the previous test of Paris CDG airport, our proposed methodology is validated for the design of 15 routes. However, the environment in the considered TMA is relatively simple, since only the Paris city is considered as an obstacle for the design. In order to test our algorithm in a more complex environment, the Zurich airport is selected as it is surrounded by mountains.

Runway configuration

The airport of Zurich has three runways, as illustrated in Fig. 6.17(a), and the numbering of each threshold is also presented. A runway threshold can be used for take-off or landing according to different wind conditions, thus there exist different SIDs/STARs for the same runway threshold in the published Jeppesen charts [9]. In our design, we select the threshold 28 for take-off, and the thresholds 14 and 16 for landing. The detailed information about the selected thresholds, together with their coordinates and altitudes are presented in Table 6.19. In the case of landing threshold, the coordinates and altitudes of the associated FAFs are given additionally.

threshold	take-off/landing	threshold		FAF	
		coordinate	altitude	coordinate	altitude
28	take-off	(348.60, 46.53)	1400		
14	landing	(346.9, 48)	1400	(343.9, 50.66)	2615
16	landing	(101.19, 123.38)	1400	(343.9, 50.66)	2615

Table 6.19: Test Zurich airport: characteristics of the thresholds, and the FAFs associated with STARs.



(a) (b)

Figure 6.17: Zurich airport configuration. (a) Three runways. (b) Buffer obstacles.

Airport surrounding environment

The radar minimum altitudes in the TMA of Zurich airport is presented in Fig. 6.18, where the areas in orange color correspond to the surrounding mountains. In the pre-processing step,

we model the mountainous areas as cylinder obstacles, whose radii are bounded by R_{min} and R_{max} (as mentioned in Subsection 3.3.2). For a mountain which can not be completely enveloped by a cylinder of this size, it is enveloped as a union of several overlapped cylinders. Moreover, the altitude of the upper basis of each cylinder is equal to the maximum altitude of the part of mountains enveloped in this cylinder plus 2000ft, so that the operational requirement for obstacle clearance in the mountainous areas [23] can be satisfied. An illustration of the cylinder obstacles modeled in the pre-processing step is presented in Figs. 6.19 (in the horizontal plane) and 6.20 (in 3D).

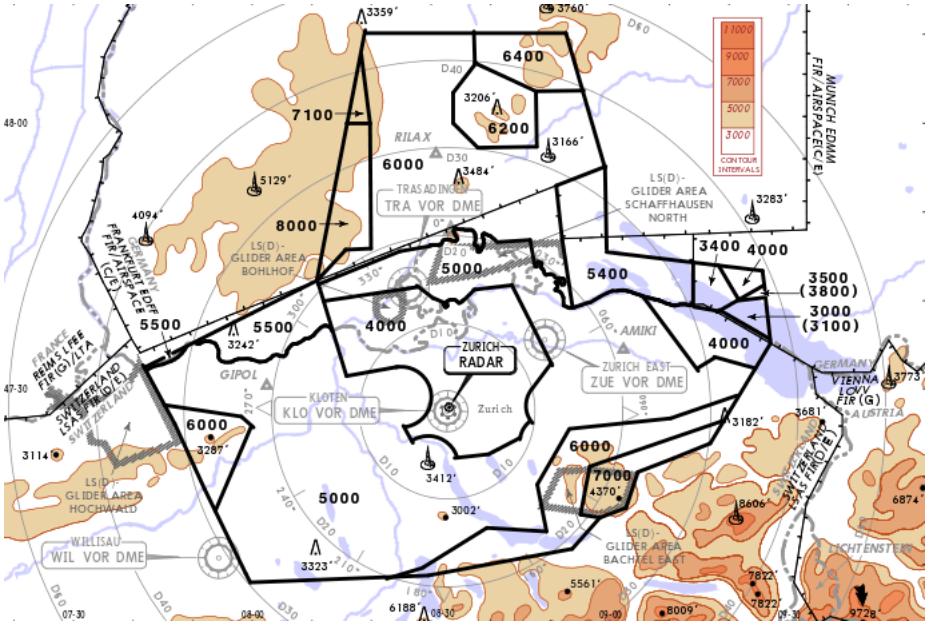


Figure 6.18: Radar minimum altitudes in the TMA of Zurich airport (source:[9]).

Selected standard routes

Based on the runway thresholds in use for take-off and landing, we select 8 standard routes, including 3 SIDs and 5 STARs, published in Jeppesen charts [9]. The selected routes, as well as the related waypoints, are shown in Fig. 6.21, where the routes in blue (respectively, red) color are SIDs (respectively, STARs). The final approach phase of a STAR is not included. Table 6.20 defines the runway threshold in use as well as the bypassing waypoints for each standard route. Since runway thresholds 14 and 16 are almost parallel, the arrival aircraft may use both routes for landing, and the corresponding routes are the same.

Input data of the test

In this test, we consider the design of 9 routes, denoted as $\gamma_i, i = 1, \cdots, 9$. The starting and ending points of γ_1 to γ_3 correspond to the ones of SID1 to SID3 respectively, as given in Table 6.20, and the starting and ending points of γ_5 to γ_9 correspond to the ones of STAR1 to STAR5 respectively, also given in Table 6.20. In addition, the starting poing of route γ_4 is the runway threshold 28, and the ending point is a TMA exit point named "SONGI" (presented by the blue solid square in Fig. 6.21). In fact, there exist two standard SIDs routes connecting the runway thresholds 14/16 to "SONGI", however the runway thresholds in use are not coherent with the one (threshold 28) that we select for departure. Therefore, these standard routes are not drawn in Fig. 6.21. The input data related to the

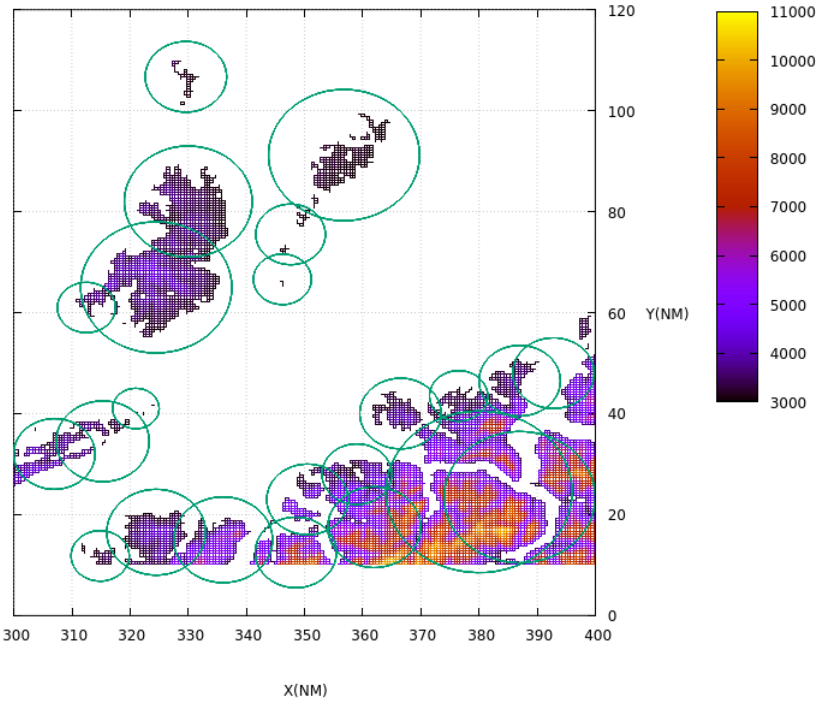


Figure 6.19: Obstacles modeling in the TMA of Zurich airport, illustration in 2D.

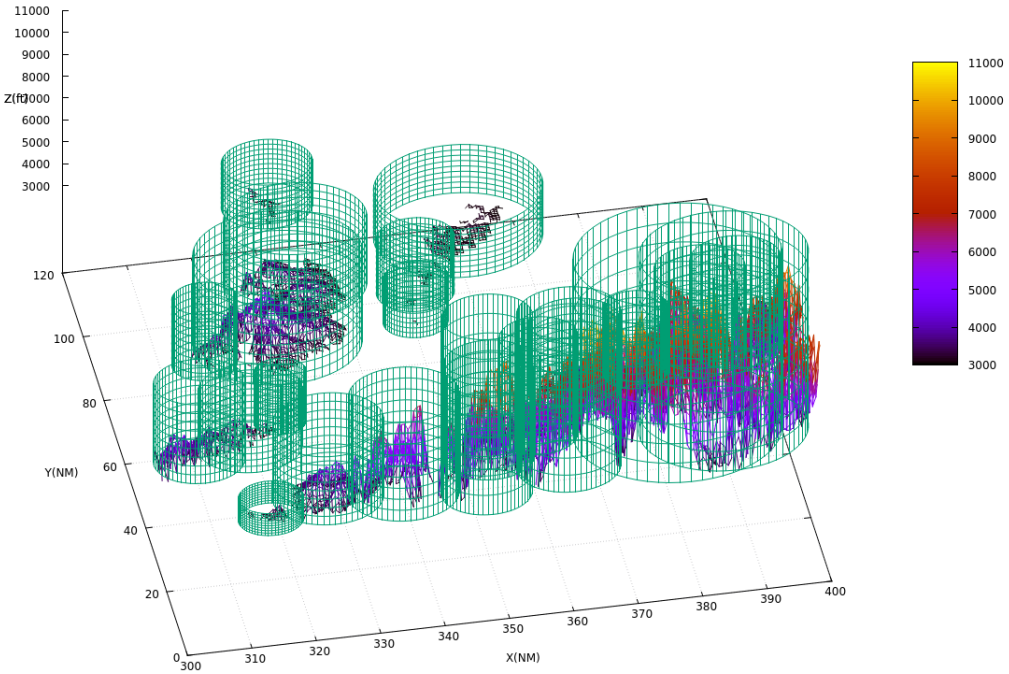


Figure 6.20: Obstacles modeling in the TMA of Zurich airport, illustration in 3D.

9 routes to design are presented in Table 6.21, including the coordinates of the starting and ending points, and the related buffer obstacle, whose characteristics are taken based on the standard routes.

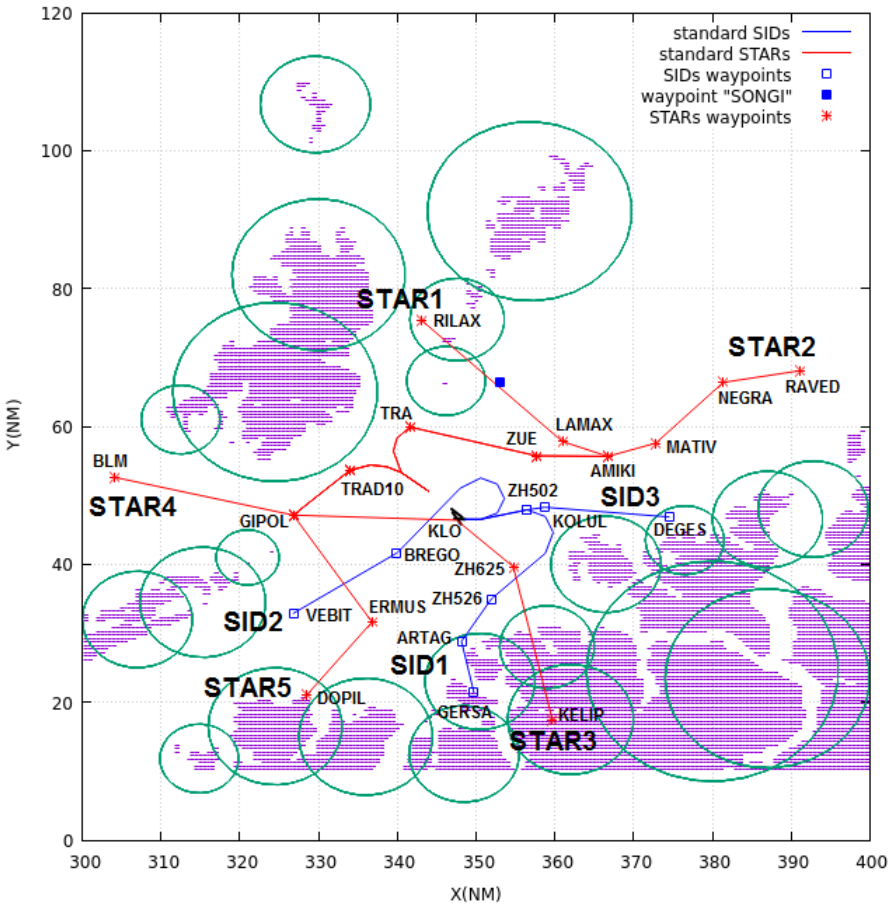


Figure 6.21: Selected standard routes, SIDs in blue color, STARs in red color.

SID/STAR	threshold	Waypoints
SID1	28	ZH502, ZH526, ARTAG, GERSA
SID2	28	BREGO, VEBIT
SID3	28	ZH502, KOLUL, DEGES
STAR1	14/16	RILAX, LAMAX, AMIKI, ZUE, TRA
STAR2	14/16	RAVED, NEGRA, MATIV, AMIKI, ZUE, TRA
STAR3	14/16	KELIP, ZH625, KLO, GIPOL, TRAD10
STAR4	14/16	BLM, GIPOL, TRAD10
STAR5	14/16	DOPIL, ERMUS, GIPOL, TRAD10

Table 6.20: Test Zurich airport: waypoints of the selected standard routes.

Figure 6.17(b) illustrates the layout of 4 buffer obstacles near the runways.

In this test, the values of the take-off and landing slopes are:

- taking-off slopes $\alpha_{min,TO} = 7\%$ $(\sim 4^\circ)$, $\alpha_{max,TO} = 11\%$ $(\sim 6.3^\circ)$;
- landing slopes $\alpha_{min,LD} = 2\%$ $(\sim 1.15^\circ)$, $\alpha_{max,LD} = 5\%$ $(\sim 2.86^\circ)$.

The landing slopes are a little bit higher than the values taken in the test of Paris CDG airport. The reason is that the environment surrounding Zurich airport is more constrained than the one in Paris CDG airport, thus aircraft need larger slopes to fly above mountains. Moreover, similarly as in the test of Paris CDG airport, we propose a circular area surrounding each buffer obstacle, where the conflict between routes using this buffer obstacle is not detected. The radius of each circular area is equal to 10Nm plus the radius of the corresponding buffer obstacle, this value is smaller than the one in the test of Paris CDG airport, since there are less routes using the same buffer obstacle in this test.

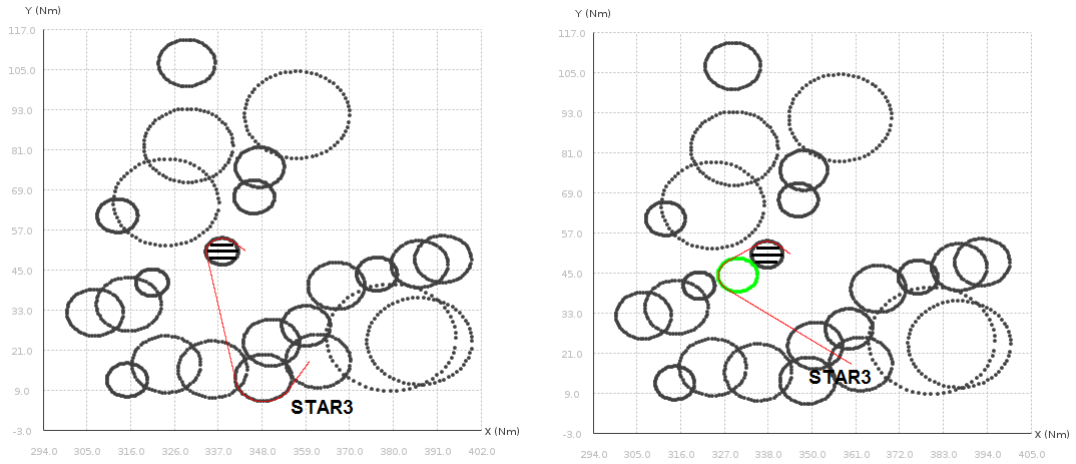
γ_i	threshold	starting point		ending point	buffer obstacle	
		(x_{A_i}, y_{A_i})	H_{A_i}	(x_{B_i}, y_{B_i})	$(x_{bi}, y_{bi}, r_{bi}, z_{bi_{inf}}, z_{bi_{sup}})$	$t_{\Omega_{bi}}$
γ_1	28	(348.60, 46.53)	1400	(349.72, 21.42)	(356.42, 43.12, 3, 0, 50000)	1
γ_2				(326.84, 32.85)	(350.75, 49.42, 3, 0, 50000)	0
γ_3				(374.48, 46.86)		
γ_4				(352.95, 66.5)		
γ_5	14/16	(343.9, 50.66)	2615	(343.07, 75.39)	(343.58, 56.3, 4, 0, 50000)	1
γ_6				(391.14, 68.09)		
γ_7				(359.69, 17.48)	(338.26, 50.32, 4, 0, 50000)	0
γ_8				(304.10, 52.63)		
γ_9				(328.47, 20.99)		

Table 6.21: Starting and ending points, and corresponding buffer obstacles of the selected routes.

Routes generated independently by the B&B

The route γ_7, generated individually by the B&B method is shown in Fig. 6.22(a), in order to increase the flown distance, the route performs a turn before arriving to the waypoint "KELIP", so that when it finally reaches "KELIP", the altitude is high enough to pass above the mountains. Even though this route corresponds to a solution of the B&B, it is not preferred from the operational point of view, since pilots need to perform a detour to proceed a waypoint after passing it. To deal with this, we propose to impose a fictitious obstacle with a given bypassing direction, corresponding to the Initial Approach Fix (IAF) "GIPOL" in the Jeppesen chart. The center and radius of this fictitious obstacle are $(330.81, 44.13)$ and 5 respectively, and it is bypassed counter-clockwise. The simulation result after adding the fictitious obstacle corresponding to the IAF is presented in Fig. 6.22(b). It can be seen that the flown distance of γ_7 is increased before reaching the ending point, and the optimal route performs a straight line segment. This manually added fictitious obstacle is kept in the generation of multiple routes.

The optimal routes generated by the B&B method is shown in Fig. 6.23(a). Three main conflicting areas occurs (areas in orange color). The first one is between routes γ_4 and γ_6 (Fig. 6.23(b)). The second one is between routes γ_1 and γ_7 (Fig. 6.23(c)). The third one occurs at the intersection among routes γ_2, γ_7 and γ_9, and after analyzing their profiles in 3D (Fig. 6.23(d)) we find that the conflict is actually between routes γ_7 and γ_9.



(a) (b)

Figure 6.22: Building route γ_7 (a) Simulation result of γ_7 by using the B&B independently. (b) Simulation result of γ_7 after adding the fictitious obstacle corresponding to the IAF (in green color).

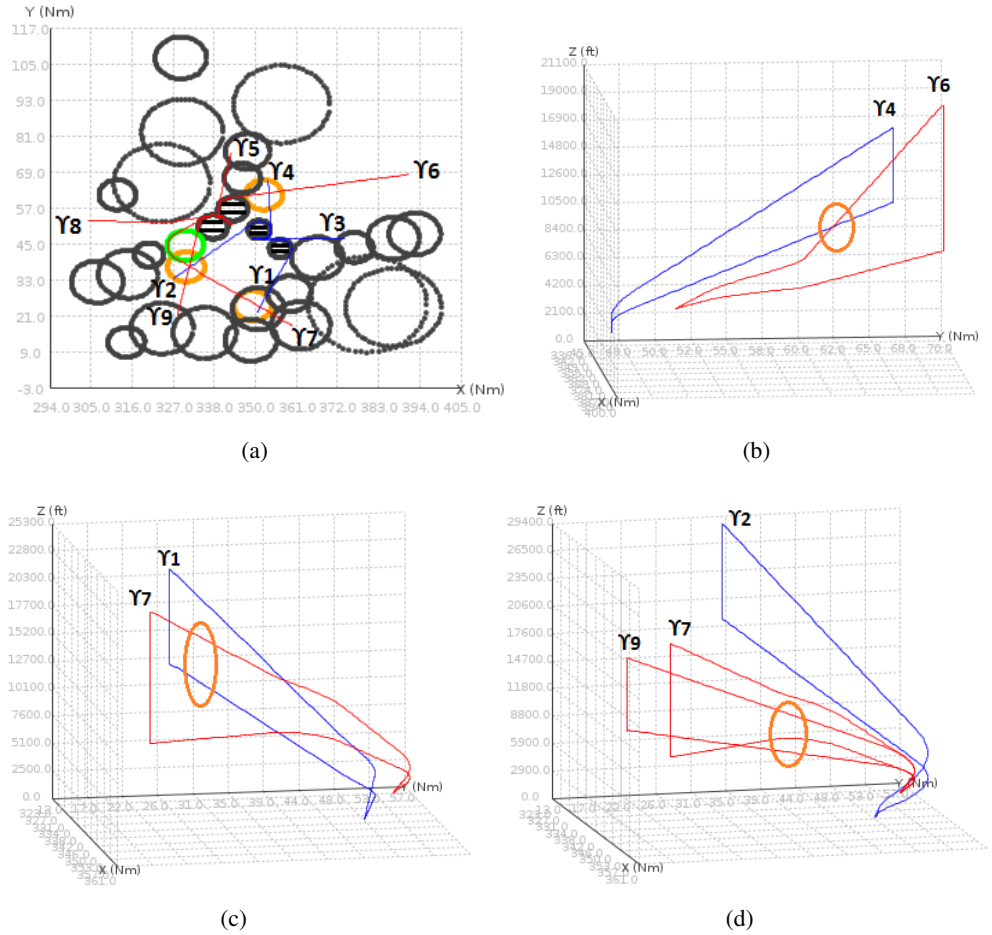
Simulation results of B&B-based method

In this test, the SIDs are built before the STARs, and the sequence for route generation is based on the increasing order of route index defined in Table 6.21. The simulation result of 9 routes is presented in Fig. 6.24 and the numerical results are given in Table 6.24. All conflicts are removed and the computing time is about 2.7s. Routes γ_6 and γ_7 are imposed level flights (Figs. 6.24(b) to 6.24(e)), so as to avoid the SIDs that are generated previously. Route γ_9 is deviated counter-clockwise in order to avoid the conflict with route γ_7. A comparison between the 9 optimal routes provided by our methodology and the 8 selected standard routes in the horizontal plane is presented in Fig. 6.25. By applying the proposed B&B-based approach, the total flown distance is decreased by 59Nm.

Simulation results of SA method

The empirically-set values of the SA parameters are given in Table 6.22.

We run the proposed SA method 10 times with the same input parameters. Table 6.23 shows the statistics of the 10 simulations. The average computing time for each simulation is about 770s. The solution with the least total routes length obtained by the SA method are illustrated in Fig. 6.26 and a comparison between the routes obtained in this solution and the standard routes is shown in Fig. 6.27. The numerical results are given in Table 6.24. In this solution, the conflict between routes γ_4 and γ_6 is solved in a similar way as in the result provided by the B&B-based method, by imposing a level flight to route γ_6. Route γ_7 is deviated counter-clockwise around a fictitious obstacle (Fig. 6.26(b)), so that the conflicts with routes γ_9 and γ_1 are removed. In this test, the B&B-based method provides a better result in terms of total length of all obtained routes ($\sum_{i=1}^{9} L_{\gamma_i^*}$), as well as the computing time.



(a) (b) (c) (d)

Figure 6.23: Test Zurich airport: routes generated independently by B&B and initial conflicting areas (orange circles). (a) Illustration in the horizontal plane. (b) Illustration of γ_4 and γ_6 in 3D. (c) Illustration of γ_1 and γ_7 in 3D. (d) Illustration of γ_2, γ_7 and γ_9 in 3D.

parameter	value
initial temperature, T_0	60
final temperature, T_f	5
temperature cool-down factor, β	0.95
number of iterations at each temperature stage, N_I	20
probability for selecting fictitious obstacle avoidance strategy, ρ_1	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_2	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_3	0.6
probability for selecting fictitious obstacle avoidance strategy, ρ_4	1
probability for eliminating a non-active fictitious obstacle, ρ_5	0.7
weight coefficient in (5.7), c_3	100

Table 6.22: Test Zurich airport: empirically-determined SA parameters.

	$\sum_{i=1}^{2} L_{\gamma_i^*}$	$\sum_{i=1}^{2} \ell^*_{i_{cflt}}$
Minimum value	387.69	0
Maximum value	418.81	4.48
Average value	397.955	0.68
Standard deviation	7.998	1.452

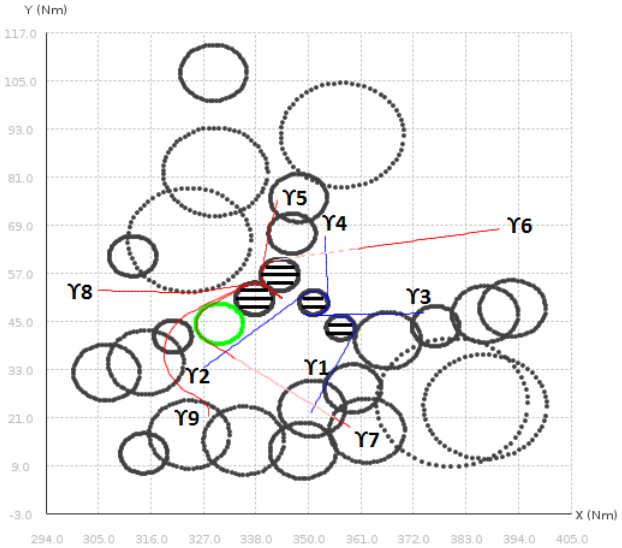
Table 6.23: Test Zurich airport: statistics of SA method.

i	L_i^{std}	$L_{\gamma_i^0}$	B&B-based method				SA method			
			$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$	δL_i	$\ell^0_{i_{cflt}}$	$L_{\gamma_i^*}$	$\ell^*_{i_{LF}}$	δL_i
1	40.58	36.35	0	36.35	0	4.23	5.14	36.35	0	4.23
2	42.99	42.64	0	42.64	0	0.35	0	42.64	0	0.35
3	26.13	25.88	0	25.88	0	0.25	0	25.88	0	0.25
4	23.96	23.96	0	23.96	0	0	6.34	23.96	0	0
5	68.72	26.78	0	26.78	0	41.94	0	26.78	0	41.94
6	66.03	61.83	8.11	61.83	18.69	4.2	8.11	61.83	15.67	4.2
7	74.15	67.53	16.21	67.53	26.89	6.62	16.21	85.91	0	-11.76
8	44.19	41.27	0	41.27	0	2.92	0	41.27	0	2.92
9	52.77	43.07	7.65	54.2	0	-1.43	7.65	43.07	0	9.7
Total	439.52	369.31	31.97	380.44	45.58	59.08	43.45	387.69	15.67	51.83

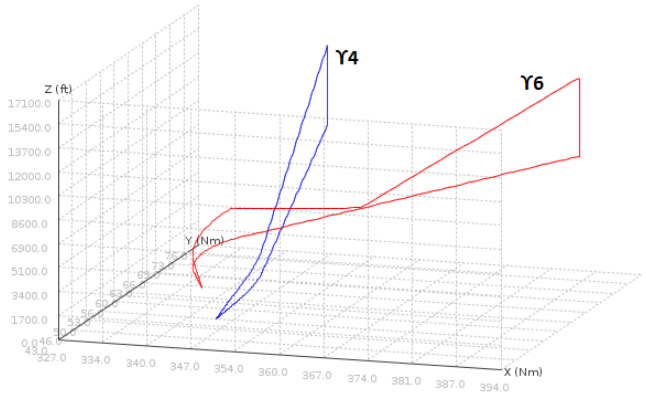
Table 6.24: Test Zurich airport: numerical results of B&B-based method and SA method (L_i^{std} is the length of the i^{th} selected standard route, and $\delta L_i = L_i^{std} - L_{\gamma_i^*}$).

6.5 Conclusion

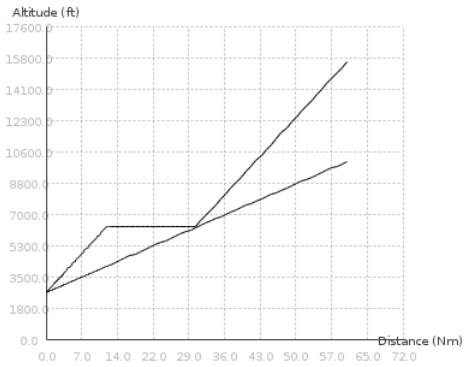
In this chapter, the Branch and Bound (B&B)-based method and the Simulated Annealing (SA) method were validated first on two artificially generated problems with various Terminal Maneuvering Area (TMA) configurations (number and layout of obstacles, runways, positions of the TMA entry/exit points). Then, the routes design was carried out on two real TMAs, the TMA of Paris CDG airport and the TMA of Zurich airport. The choice of these two TMAs allowed us to test our approach, on the one hand, on a TMA with numerous TMA entry/exit points, and on the other hand, on a TMA with many obstacles. Our proposed approaches can efficiently design conflict-free routes in all the tested problems. Comparisons between the designed routes and standard routes were given in the cases of Paris CDG and Zurich. The simulation results shows a gain in the total routes length, which may lead to a reduction in jet fuel consumption. The B&B-based method run faster than the SA method, but the SA method provided better solutions in most of the cases. A Human Computer Interaction (HCI) interface has been developed, which can be used as a visualization tool. A brief introduction of the interface is given in Appendix A.



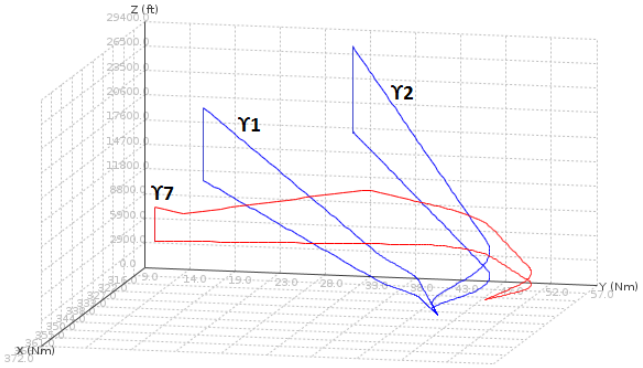
(a)



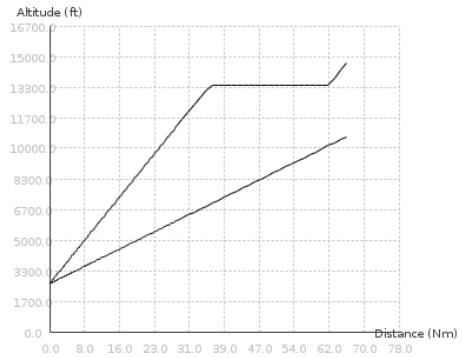
(b)



(c)



(d)



(e)

Figure 6.24: Test Zurich airport: B&B-based method simulation results. (a) Illustration in the horizontal plane. (b) Illustration of γ_4 and γ_6 in 3D. (c) Illustration of γ_6 in the vertical plane. (d) Illustration of γ_1, γ_2 and γ_7 in 3D. (e) Illustration of γ_7 in the vertical plane.

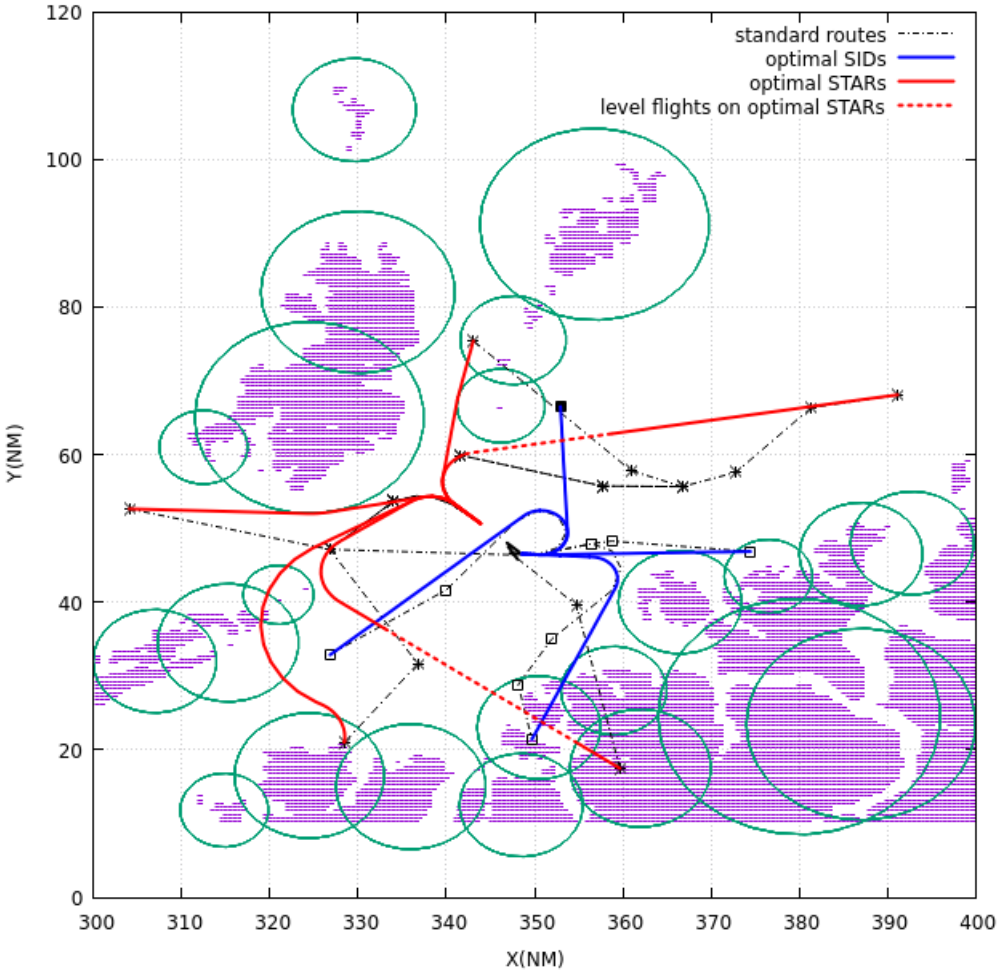
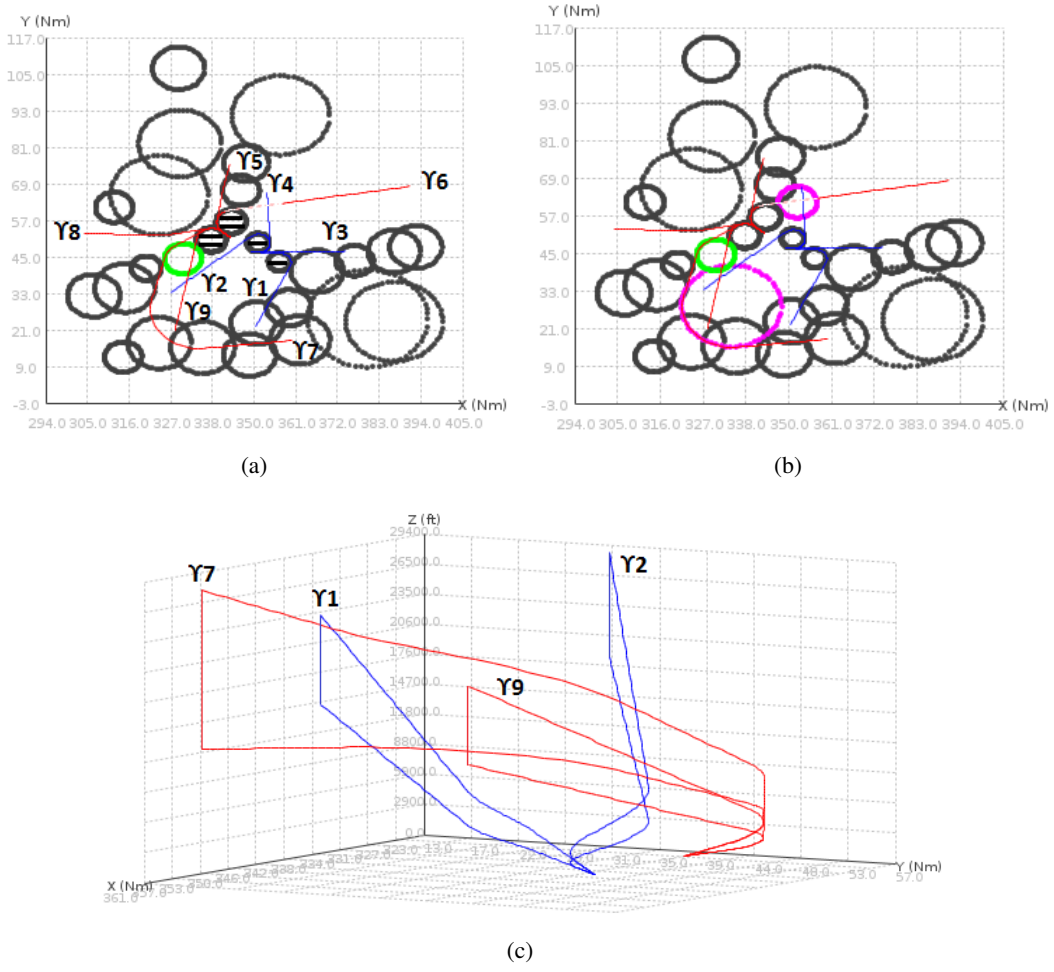


Figure 6.25: Comparison between simulation result and selected routes in the horizontal plane.



(a) (b)

(c)

Figure 6.26: Test Zurich airport: SA method simulation results. (a) Illustration in the horizontal plane. (b) Fictitious obstacles for route deviation. (c) Illustration of γ_1, γ_2, γ_7 and γ_9 in 3D.

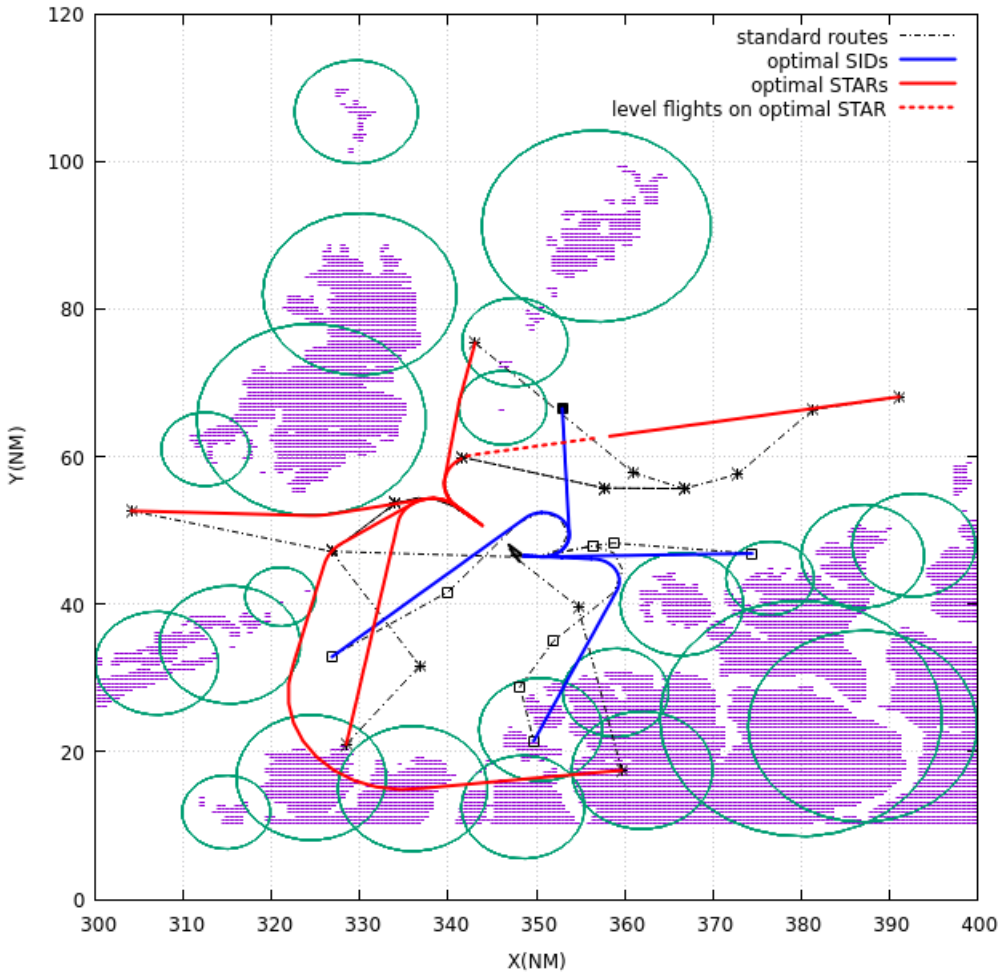


Figure 6.27: Comparison between simulation result and selected routes in the horizontal plane.

Conclusions and Perspectives

The focus of this thesis has been on the optimal design of Standard Instrument Departure routes (SIDs) and the Standard Terminal Arrival Routes (STARs) in Terminal Maneuvering Areas (TMAs). We conclude with a summary of this thesis and its main contributions. In addition, we discuss about the way the proposed methodology is implemented in practice. Finally, some future directions that can be followed in the following research are proposed.

Review of the thesis

In this thesis, we presented a decision support methodology for the design of multiple Standard Instrument Departure routes (SIDs) and Standard Terminal Arrival Routes (STARs) in a Terminal Maneuvering Area (TMA) at a strategic level. The SIDs and STARs were designed in an optimal way, with respect to the total routes length. The design on the one hand takes into account numerous operational and environmental constraints, especially the obstacle avoidance and routes separation as main constraints, on the other hand takes advantage of the highly efficient and flexible navigation mode, the Required navigation performance (RNP).

Each route to design was modeled in 3D, consisting of two components: a curve in the horizontal plane associated with a cone in the vertical plane. In the horizontal plane, the form of a curve is a succession of segments and arcs of circles. This modeling is compatible with the RNP, thus the flyability of such a curve is guaranteed in the real operation. More precisely, each segment curve section corresponds to a point-to-point leg and each arc curve section corresponds to a Radius-to-Fix (RF) leg. In the vertical plane, the cone is designed to envelop all vertical profiles of flights following this route. This corresponds to what is done in the manual design of SIDs/STARs. An obstacle to be avoided was modeled as a cylinder, whose projection in the horizontal plane is a disk. We proposed three different ways to avoid an obstacle: bypassing clockwise, bypassing counter-clockwise, or imposing a level flight below the cylinder. These three maneuvers are feasible from the operational point of view, and conform to what pilots execute to avoid obstacles in reality. In addition, a route bypasses an cylinder clockwise or counter-clockwise by using an arc lying on the border of the cylinder.

The design of multiple routes was then modeled as a combinatorial optimization problem. One of the main difficulty is to deal with the separation between routes. To get rid of the complexity of the problem, first, a simpler subproblem of designing one optimal route was addressed. Since the objective function and some constraints are not expressed explicitly by the decision variables, we applied a Branch and Bound (B&B) method to solve the problem. The branching strategies in the B&B method correspond to the ways of obstacle avoidance. Some artificially generated tests were carried out, where different layouts of obstacles were given. The B&B method shows its efficiency in these tests. The obtained routes are continuous and smoothing which are compatible with for Continuous Climb Operation (CCO) [105] and Continuous Descent Operation (CDO) [106]. These

continuous operations may lead to reductions in noise, fuel burn and emissions.

The design of multiple routes was addressed by using two different approaches, where the constraint on routes separation was considered additionally. The first resolution approach is a B&B-based method, where routes are built sequentially according to some fixed order (for example, in decreasing order of traffic load on each route). Each route is generated by first applying the B&B method independently. If it is in conflict with existing routes, fictitious obstacles enveloping the conflicting areas are introduced. The route is then deviated by contouring the fictitious obstacles or imposing level flights. This allows us to keep the non-conflicting route portions and only re-generate route portions involved in conflicts. In this first approach, the quality of the obtained routes depends on the order of routes generation. For this reason, we developed another approach where routes are generated simultaneously by applying the Simulated Annealing (SA) method. Fictitious obstacles are also introduced to envelop conflicting zones, and the strategies of avoiding them are randomly selected.

These two approaches were first tested on two artificially generated problems with various TMA configurations (number and layout of obstacles, runways, positions of the TMA entry/exit points). Then tests of routes design in real TMAs (the TMA of Paris CDG airport and the TMA of Zurich airport) were also carried out. The choice of the TMAs of Paris CDG and Zurich allows us to test our approach, on the one hand, on a TMA with numerous TMA entry/exit points, and on the other hand, on a TMA with many obstacles. Both methods can effectively generate conflict-free routes in all tests. In the case of Paris CDG and Zurich, comparisons between the obtained routes and the selected standard routes were given. The simulation results shows a gain in the total routes length, that can be promising in terms of jet fuel expense reduction. The SA method provides better solutions in terms of total routes length in most of the cases, however, the B&B-based method is faster.

Discussion on the implementation of the method in real-world operations

The real-world procedure design often considers a variety of constraints that are not completely included in this work. Thus the routes obtained by using the proposed approach can be regarded as preliminary guidelines for the manual design. This methodology can be easily applied in reality. Users send the input data related to the routes to design, including:

- airport configuration (runway thresholds for take-off and landing, together with the associated buffer obstacles),
- obstacles in the surrounding environment (mountains cities, restricted airspaces),
- desired TMA entry/exit points,
- desired maximum/minimum take-off/landing slopes,

as well as other user-defined parameters (for example, the minimum altitude of level flights, the maximum number of level flights allowed on each route, etc.) to the algorithm. A set of optimal routes in accordance with users' demands will be provided after an efficient computation. Note that, it is possible that conflicts still exist in the provided routes. In this case, users may correct the form of a certain route by adding fictitious obstacles manually and by selecting their positions and bypassing directions, so as to remove the conflicts.

A RNP route follows a succession of waypoints which are defined by latitude and longitude coordinates, as mentioned in Section 1.1. In order to implement a designed route to real-world application, we need to select some waypoints defining this route. These waypoints can be selected by considering two aspects: the ones defining the form of the projection of a designed route in the horizontal plane, and the ones defining the route sections corresponding to level flights in the vertical plane. In the horizontal plane, recall that the projection of a designed route is a succession of segments (corresponding to point-to-point legs) and arcs (corresponding to RF legs). Thus the end points of RF legs on a route can be selected as waypoints. In the vertical plane, the end points of a route section corresponding to a level flight can be selected as waypoints.

In order to represent the selected waypoints on a designed route, let us first see two published charts as examples. The first example is a RNP arrival route of Queenstown airport (Fig. 6.28), each RF leg is defined as an arc between two waypoints, besides, the length of the arc as well as the bypassing direction on the arc are also marked. The second example is an arrival route of Paris CDG airport (Fig. 6.29), where a route section maintains its flight level at or above FL120 (in red circle). We can represent the selected waypoints on our designed routes in similar ways. We use the route γ_9 (respectively, γ_6), obtained in the simulation of the Paris CDG airport by using the B&B-based method (see Fig. 6.12(a)), to illustrate how the selected waypoints defining the projection of a designed route in the horizontal plane (respectively, defining a route section corresponding to a level flight) are represented, an illustration is given in Fig. 6.30(a) (respectively, Fig. 6.30(b)).

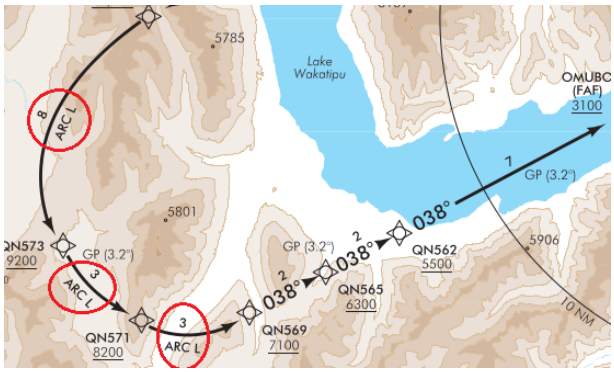


Figure 6.28: A RNP arrival route of Queenstown airport[1].

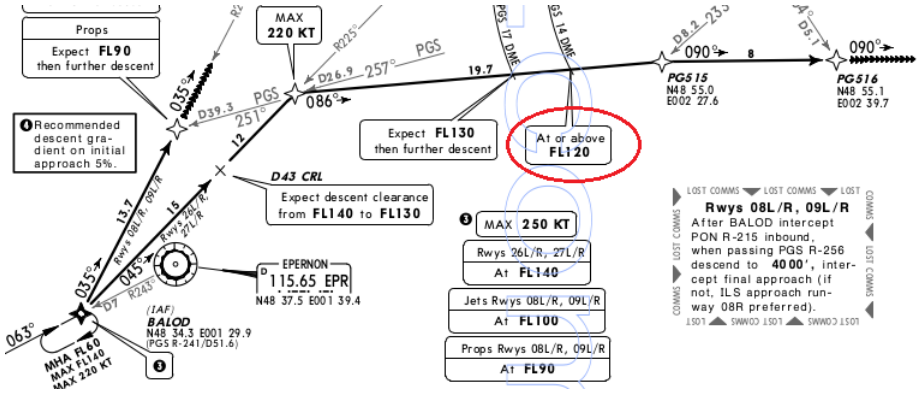
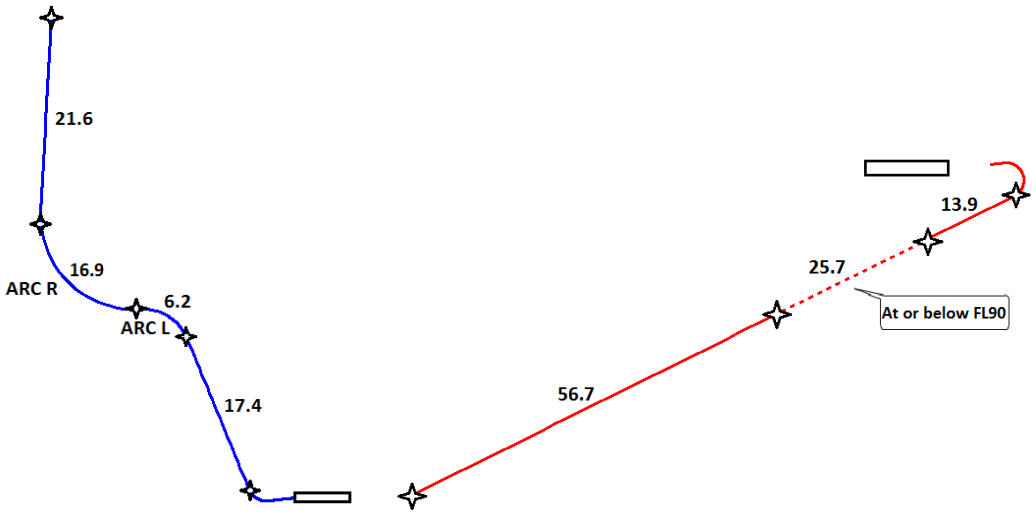


Figure 6.29: An arrival route of Paris CDG airport (source: [10]).

[1]source:`http://www.aip.net.nz/pdf/NZQN_45.1_45.2.pdf`



(a) (b)

Figure 6.30: Representing the selected waypoints on designed routes. (a) Waypoints defining the projection of a designed route in the horizontal plane. (b) Waypoints defining a route section corresponding to a level flight.

Perspectives

Several research directions can be followed in the future work.

Routes merging

In the TMA surrounding a major airport, where a huge amount of traffic is present, the arrival routes usually merge gradually before reaching the runways. Indeed, it is easier for the controllers to supervise and sequence flights under this kind of routes structure. In our current design, all arrival routes merge in an area near the runways. In the future work, the routes merging problem can be considered. Since routes merge progressively, more free space for the design is offered.

Metroplex TMA

A TMA considered in this thesis surrounds one single airport. In reality, there are many TMAs surrounding more than one airports, named *metroplex* TMAs. It is the case as in many big cities. An example of departure and arrival routes in the metroplex TMA of New York is illustrated in Fig. 6.31. To design routes in such a context, the interactions between the routes converging to different airports need to be dealt with. As the complexity of such a problem is higher compared to what we considered currently, extra decision variables and constraints need to be added additionally.

Route re-designing at tactical level with respect to hazardous weather

In the case when severe weather condition (for example, storms) occurs near an airport, the problem of re-designing a route at tactical level need to be considered. Currently, to avoid hazardous weather, pilots communicate with controllers to allocate a new route. This kind of operation can hardly be optimized in terms of flown distance. In the future work, it is possible to extend our current method by taking into account moving obstacles (referring to areas with severe weather).

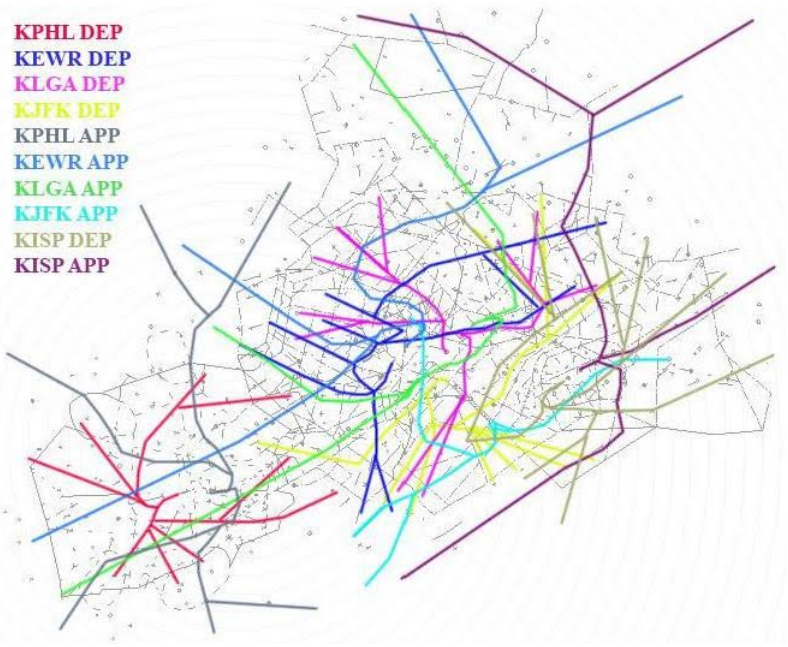


Figure 6.31: Departure and arrival routes in the metroplex TMA of New York.

The efficiency of our current methodology has been proven through different types of tests, thus it can be expected to be efficient after taking into account the weather condition. In the future Air Traffic Management (ATM) system, the improved methodology can by implemented to the Flight Management System (FMS), or to the interface of controllers, so as to provide safe and optimal routes avoiding hazardous weather at pre-tactical or tactical level.

Appendix A

Visualization tool: SID/STAR design interface

The overview of the interface is shown in Fig. A.1. The users create a scenario in the interface, including the data related to routes to be designed, as well as the values of user-defined parameters. These informations are exported and sent to the solution algorithms. The routes obtained are then exported to the interface for illustration.

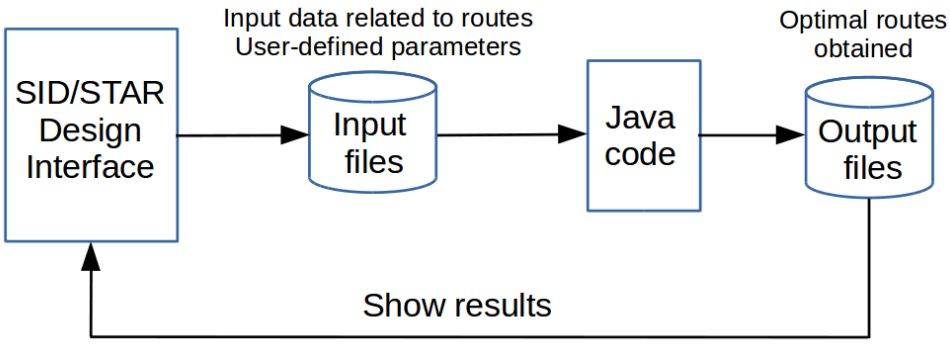


Figure A.1: Overview of the interface

An illustration of the interface is shown in Fig. A.2. The following actions can be done through the interface in order to create a scenario:

- create/delete/move runways, TMA entry or exit points, obstacles, and buffer obstacles;
- increase/decrease the radius of obstacles and buffer obstacles;
- select pairwise points as the starting and ending points of a route to be designed;
- input the values of user-defined parameters;
- call the java code to solve the corresponding problem;
- show results in 2D and 3D.

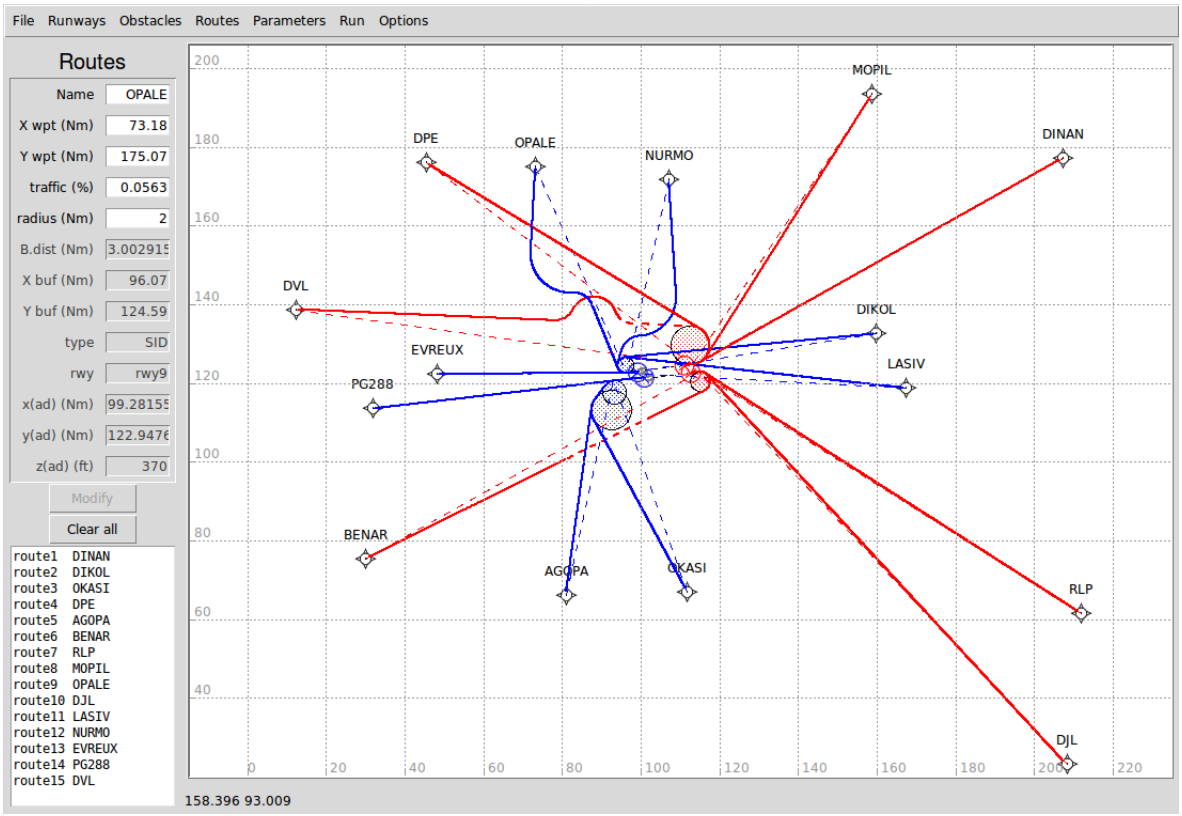


Figure A.2: SID/STAR design interface

Bibliography

[1] Eurocontrol, "Challenges of growth 2013, task 4: European air traffic in 2035," 2013.

[2] Eurocontrol, "European airspace concept handbook for PBN implementation," 2013.

[3] D. S. Kim, K. Yu, Y. Cho, D. Kim, and C. Yap, "Shortest paths for disc obstacles," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA2004)*, (Berlin, Heidelberg), pp. 62–70, Springer Berlin Heidelberg, 2004.

[4] K. Jimmy, L. Changkil, and M. Joseph S. B., "Turn-constrained route planning for avoiding hazardous weather," *Air Traffic Control Quarterly*, vol. 14, no. 2, pp. 159–182, 2006.

[5] S. Pierre, D. Delahaye, and S. Cafieri, "Aircraft trajectory planning with dynamical obstacles by artificial evolution and convex hull generations," in *the 4th ENRI International Workshop on ATM/CNS (EIWAC2015)*, (Tokyo, Japan), Nov. 2015.

[6] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimization of arrival and departure routes in terminal maneuvering area," in *the 6th International Conference on Research in Air Transportation (ICRAT2014)*, (Istanbul, Turkey), May 2014.

[7] D. M. Pfeil, *Optimization of airport terminal-area air traffic operations under uncertain weather conditions*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2011.

[8] D. Gianazza, N. Durand, and N. Archambault, "Allocating 3D-trajectories to air traffic flows using A* and genetic algorithms," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control, and Automation (CIMCA04)*, (Gold Coast, Australia), July 2004.

[9] Jeppesen, "LSZH (Zurich)," 2007.

[10] Jeppesen, "LFPG (Charles-De-Gaulle)," 2011.

[11] International Air Transport Association (IATA), "Economic & Social Benefits of Air Transport," 2016.

[12] Boeing, "Current market outlook 2016-2035," 2016.

[13] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimizing the design of a route in Terminal Maneuvering Area using Branch and Bound," in *the 4th ENRI International Workshop on ATM/CNS (EIWAC2015)*, (Tokyo, Japan), Nov. 2015.

[14] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimizing the design of a route in Terminal Maneuvering Area using Branch and Bound," *Air Traffic Management and Systems – II, Lecture Notes in Electrical Engineering*, vol. 420, pp. 171–184, 2017.

[15] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimization-Based Design of Departure and Arrival Routes in Terminal Maneuvering Area," *Journal of Guidance Control and Dynamics*. (in press).

[16] J. Zhou, S. Cafieri, D. Delahaye, and M. Sbihi, "Optimal design of SIDs/STARs in TMA using Simulated Annealing," in *the 35th IEEE/AIAA Digital Avionics Systems Conference (DASC2016)*, (Sacramento, CA, United States), Sept. 2016.

[17] Federal Aviation Administration (FAA), "NextGen Implementation Plan 2016," 2016.

[18] European Commission, "SESAR: The future of flying," 2010.

[19] European Cockpit Association, "The future of flying in a single european sky, a crew perspective," 2015.

[20] International Civil Aviation Organization (ICAO), "Performance-based Navigation (PBN) Manual (Doc 9613)," 2008.

[21] Civil Aviation Safety Authority (CASA) Australia, "Performance based navigation operational approval handbook," Aug. 2010.

[22] International Civil Aviation Organization (ICAO), "Air traffic services (annex 11)," 2001.

[23] International Civil Aviation Organization (ICAO), "Aircraft Operations (Doc 8168)," 2006.

[24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[25] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[26] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.

[27] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

[28] J. Clausen, "Branch and Bound algorithms – principles and examples." Department of Computer Science, University of Copenhagen, 1999. `http://www.imada.sdu.dk/~jbj/heuristikker/TSPtext.pdf`.

[29] A. M. Geoffrion, "Integer programming by implicit enumeration and balas' method," *SIAM Review*, vol. 9, no. 2, pp. 178–190, 1967.

[30] E. Balas, "Discrete programming by the filter method," *Operations Research*, vol. 15, no. 5, pp. 915–957, 1967.

[31] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, July 1968.

[32] J. Dreo, P. Siarry, A. Petrowski, and E. Taillard, *Metaheuristics for Hard Optimization*. Springer, 2006.

[33] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Nov. 1995.

[34] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.

[35] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.

[36] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[37] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, vol. 1, no. 1, pp. 31–39, 1995.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[39] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[40] P. Gallina and A. Gasparetto, "A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot," *Journal of Intelligent and Robotic Systems*, vol. 27, no. 3, pp. 237–262, 2000.

[41] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM2013)*, pp. 1–8, Oct. 2013.

[42] Y. Liang, Q. Juntong, S. Dalei, X. Jizhong, H. Jianda, and X. Yong, "Survey of robot 3d path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[43] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Computing*, vol. 28, pp. 652–673, Feb. 1999.

[44] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[45] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, pp. 560–570, Oct. 1979.

[46] H. Rohnert, "Shortest paths in the plane with convex polygonal obstacles," *Information Processing Letters*, vol. 23, no. 2, pp. 71 – 76, 1986.

[47] J. A. Storer and J. H. Reif, "Shortest paths in the plane with polygonal obstacles," *Journal of the ACM*, vol. 41, pp. 982–1012, Sept. 1994.

[48] M. Sharir and A. Schorr, "On shortest paths in polyhedral spaces," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, (New York, NY, USA), pp. 144–153, ACM, 1984.

[49] K. Jiang, L. Seneviratne, and S. Earles, "Finding the 3d shortest path with visibility graph and minimum potential energy," in *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 679–684, July 1993.

[50] M. N. Bygi and M. Ghodsi, "3D Visibility Graph," in *the 12th CSI Computer Conference (CSICC2006)*, (Shahid Beheshti University, Tehran), 2006.

[51] J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *the 28th Annual Symposium on Foundations of Computer Science*, pp. 49–60, Oct. 1987.

[52] Y.-H. Liu and S. Arimoto, "Path planning using a tangent graph for mobile robots among polygonal and curved obstacles," *International Journal of Robotics Research*, vol. 11, pp. 376–382, Aug. 1992.

[53] M. Pocchiola and G. Vegter, "Minimal tangent visibility graphs," *Computational Geometry*, vol. 6, no. 5, pp. 303 – 314, 1996.

[54] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, pp. 345–405, Sept. 1991.

[55] H. Choset, *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, Pasadena, CA, USA, 1996.

[56] B. Chazelle and L. J. Guibas, "Visibility and intersection problems in plane geometry," *Discrete & Computational Geometry*, vol. 4, no. 6, pp. 551–581, 1989.

[57] N. H. Sleumer and N. Tschichold-Gürman, "Exact cell decomposition of arrangements used for path planning in robotics," tech. rep., Institute of Theoretical Computer Science Zurich, 1999.

[58] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Proceedings of 2004 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 467–472, Apr. 2004.

[59] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, Aug. 1996.

[60] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," tech. rep., Computer Science Department, Iowa State University.

[61] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron, "Mathematical models for aircraft trajectory design: A survey," in *Air Traffic Management and Systems: Selected Papers of the 3rd ENRI International Workshop on ATM/CNS (EIWAC2013)*, (Tokyo), pp. 205–247, Springer Japan, 2014.

[62] H. Jeffreys and B. S. Jeffreys, *Methods of mathematical physics*. Cambridge University Press, 1988.

[63] G. E. Farin and D. Hansford, *The Essentials of CAGD*. Natick, MA, USA: A. K. Peters, Ltd., 1st ed., 2000.

[64] C. de Boor, *A Practical Guide to Splines*. New York: Springer, 1978.

[65] P. Hartmut, B. Wolfgang, and P. Marco, *Bézier and B-Spline Techniques*. New York: Springer, 2002.

[66] D. Jung and P. Tsiotras, "On-line path generation for small unmanned aerial vehicles using B-Spline path templates," in *AIAA Guidance, Navigation, and Control Conference, Honolulu, HI*, AIAA Paper 2008-7135, Aug. 2008.

[67] C. Peyronne, D. Delahaye, M. Mongeau, and L. Lapasset, "Optimizing B-splines using Genetic Algorithms Applied to Air-traffic Conflict Resolution," in *ICEC 2010, International Conference on Evolutionary Computation*, (Valencia, Spain), pp. 213–218, Oct. 2010.

[68] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[69] J.-D. Boissonnat and S. Lazard, "A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles (extended abstract)," in *Proceedings of the 12th Annual Symposium on Computational Geometry*, SCG '96, (New York, NY, USA), pp. 242–251, ACM, 1996.

[70] S. Bereg and D. Kirkpatrick, "Curvature-bounded traversals of narrow corridors," in *Proceedings of the 21st Annual Symposium on Computational Geometry*, SCG '05, (New York, NY, USA), pp. 278–287, ACM, 2005.

[71] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *2007 46th IEEE Conference on Decision and Control*, pp. 2379–2384, Dec. 2007.

[72] I. Lugo-Cárdenas, G. Flores, S. Salazar, and R. Lozano, "Dubins path generation for a fixed wing UAV," in *International Conference on Unmanned Aircraft Systems (ICUAS 2014)*, (Orlando, FL , United States), pp. 339–346, May 2014.

[73] D. Delahaye and S. Puechmorel, *Modeling and Optimization of Air Traffic*. Wiley, June 2013.

[74] A. Eele and A. Richards, "Path planning with avoidance using nonlinear branch and bound optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, pp. 384–394, 2009.

[75] V. Polishchuk, "Generating Arrival Routes with Radius-to-Fix Functionalities," in *the 7th International Conference on Research in Air Transportation (ICRAT2016)*, (Philadelphia, United States), June 2016.

[76] T. A. Granberg, T. Polishchuk, V. Polishchuk, and C. Schmidt, "Automatic Design of Aircraft Arrival Routes with Limited Turning Angle," in *the 16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*, vol. 54, (Dagstuhl, Germany), pp. 1–13, 2016.

[77] Gurobi Optimization, Inc., "Gurobi Optimizer Quick Start Guide," 2017. `https://www.gurobi.com/documentation/7.0/quickstart_windows.pdf`.

[78] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, pp. 199–235, 1998.

[79] J. T. Chen, A. Yousefi, S. Krishna, B. Sliney, and P. Smith, "Weather avoidance optimal routing for extended terminal airspace in support of dynamic airspace configuration," in *the 31st IEEE/AIAA Digital Avionics Systems Conference (DASC2012)*, pp. 3A1–1–3A1–16, Oct. 2012.

[80] J. T. Chen, A. Yousefi, S. Krishna, D. Wesely, B. Sliney, and P. Smith, "Integrated arrival and departure weather avoidance routing within extended terminal airspace," in *the 32nd IEEE/AIAA Digital Avionics Systems Conference (DASC2013)*, pp. 1A4–1–1A4–17, Oct. 2013.

[81] D. Gianazza and N. Durand, "Separating air traffic flows by allocating 3d-trajectories," in *the 23rd Digital Avionics Systems Conference (DASC2004)*, vol. 1, pp. 2.D.4–21–13 Vol.1, Oct. 2004.

[82] D. Gianazza and N. Durand, "Assessment of the 3D-separation of Air Traffic Flows," in *the 6th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2005*, (Baltimore, United States), June 2005.

[83] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 179–189, Dec. 2000.

[84] S. Chaimatanan, D. Delahaye, and M. Mongeau, "A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale," *IEEE Computational Intelligence Magazine*, vol. 9, pp. 46–61, Nov. 2014.

[85] C. Allignol, N. Barnier, and A. Gondran, "Optimized vertical separation in europe," in *the 31st IEEE/AIAA Digital Avionics Systems Conference (DASC2012)*, pp. 4B3–1–4B3–10, Oct. 2012.

[86] N. Barnier and C. Allignol, "4D-trajectory deconfliction through departure time adjustment," in *the 8th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2009)*, (Napa, United States), June 2009.

[87] N. Durand and J.-B. Gotteland, "Genetic Algorithms Applied to Air Traffic Management," in *Metaheuristics for Hard OptimizationMethods and Case Studies*, pp. 277–306, Springer, Jan. 2006.

[88] O. Rodionova, M. Sbihi, D. Delahaye, and M. Mongeau, "North Atlantic Aircraft Trajectory Optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 2202–2212, Oct. 2014.

[89] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: new models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, 2014.

[90] R. Breil, D. Delahaye, L. Lapasset, and E. Féron, "Multi-agent Systems for Air Traffic Conflicts Resolution by Local Speed Regulation," in *the 7th International Conference on Research in Air Transportation (ICRAT 2016)*, (Philadelphie, PA, United States), June 2016. Best Paper Award for Automation track.

[91] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, pp. 90–98, Apr. 1986.

[92] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[93] L. Guys, *Planification de Trajectoires d'Avions sans Conflit : Fonctions Biharmoniques et Fonction de Navigation Harmonique*. PhD thesis, Université Paul Sabatier, Toulouse, Oct. 2014.

[94] N. E. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *Journal of Global Optimization*, vol. 56, pp. 873–895, July 2013.

[95] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[96] D. Toratani and S. Ueno, "A study on trajectory optimization for the terminal area," in *the 6th International Conference on Research in Air Transportation (ICRAT2014)*, (Istanbul, Turkey), May 2014.

[97] D. Toratani, S. Ueno, and T. Higuchi, "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, 2015.

[98] D. Toratani, D. Delahaye, S. Ueno, and T. Higuchi, "Merging Optimization Method with Multiple Entry Points for Extended Terminal Maneuvering Area," in *the 4th ENRI International Workshop on ATM/CNS (EIWAC2015)*, (Tokyo, Japan), Nov. 2015.

[99] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static Hamilton–Jacobi equations," in *Proceedings of the National Academy of Sciences*, vol. 98, pp. 11069–11074, 2001.

[100] B. Girardet, L. Lapasset, D. Delahaye, C. Rabut, and Y. Brenier, "Generating optimal aircraft trajectories with respect to weather conditions," in *the 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management (ISIATM2013)*, (Toulouse, France), July 2013.

[101] Federal Aviation Administration (FAA), "The United States Standard for Area Navigation (RNAV) (Orders 8260.54A)," 2007.

[102] Federal Aviation Administration (FAA), "United States Standard for Performance Based Navigation (PBN) Instrument Procedure Design (Orders 8260.58A)," 2016.

[103] Eurocontrol, "User Manual for the Base of Aircraft Data (BADA) Revision 3.9," April 2011.

[104] Garmin, "Garmin radius to fix leg project report," 2013. `http://www.phenom.aero/resources/library/garmin_radius_fix_legs_jan15_2013.pdf`.

[105] International Civil Aviation Organization (ICAO), "Continuous Climb Operations (CCO) Manual (Doc 9993)."

[106] International Civil Aviation Organization (ICAO), "Continuous Descent Operations (CDO) Manual (Doc 9931)."

[107] International Civil Aviation Organization (ICAO), "Performance Based Navigation (PBN) Operational Approval Handbook,"