Application of Metaheuristic and Deterministic Algorithms for Aircraft Reference Trajectory Optimization

by

Alejandro MURRIETA MENDOZA

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULLFILMENT FOR THE DEGREE OF DOCTOR OF AEROSPACE ENGINEERING
Ph. D.

MONTREAL, 6 SEPTEMBER 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mrs. Ruxandra Mihaela Botez, Thesis Supervisor
Department of Automated Production Engineering, École de technologie supérieure

Mrs. Lyne Woodward, Chair Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mr. Marc Paquet, Member of the jury
Department of Automated Production Engineering, École de technologie supérieure

Mr. Adrian Hiliuta, External Evaluator
CMC Electronics - Esterline

THIS THESIS WAS PRENSENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

17 AUGUST 2017

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENT

First of all, I would like to express my gratitude to my professor Ruxandra Botez her guidance, encouragement, support, and advice during my doctoral studies. Also would like to mention her patience and her thrust during my PhD studies. It was an amazing opportunity for me to be part of her laboratory and her team.

I would also like to thank all the members of LARCASE, but especially to all students I had the pleasure to work with, share ideas, and with whom I had interesting discussions: Bruce Beuze, Laurane Ternisien, Audric Bunel, Sonya Kessaci, Joahnn Bichara, and *les beaux gosses*: Antoine Hamy, Paul Mugnier, Charles Romain, and Hugo Ruiz, AKA "*Stiglitz"*. I would not have been able to accomplish as much as I did without their efforts and implication in our project. Merci beaucoup!

I would also like to acknowledge the financial support provided by CONACYT in Mexico, the Fonds de recherché Nature et Technologie (FRQNT) in Quebec, the Green Aviation Research & Development Network (GARDN) in Canada, and CMC Electronics – Esterline.

I would like to thank Margaux, Marine, and Polo for their patience and for all those lunch hours we passed talking about anything and everything. I would also thank Georges Ghazi for his friendship, all the conversations and discussions we had, all his comments and corrections, all the fun moments, and all our trips over the world.

I would like Roberto Felix with whom I began this adventure of coming to Canada, for his friendship, his advices, his help in both professional and personal ways, and for all projects that we had, we have, and that we will have.

I would like to thank Vanessa, the other one in this *canadian expedition* for being there in both good and bad moments.

I would also like to thank Dr. Sinhue Benitez, for all his advices, for kicking me out my comfort zone, for always believing in me, and not letting me slip out of my dream of pursuing graduated studies.

Finally, I would like my family, Yolanda (*yola*), Alejandro (*dotor*), and Patty for their support in every single moment during all these years, for understanding my desire of moving away to do my master and PhD. studies, for all your help and advices.

APPLICATION D'ALGORITHMES METAHEURISTIQUES ET DETERMINISTES POUR L'OPTIMISATION DE TRAJECTOIRE DE REFERENCE D'AERONÉFS

Alejandro MURRIETA MENDOZA

RÉSUMÉ

L'optimisation de la trajectoire de référence des aéronefs est une méthode alternative pour réduire la consommation de carburant, ainsi que pour réduire la pollution rejetée dans l'atmosphère. La réduction de la consommation de carburant revêt une importance particulière pour deux raisons: premièrement, l'industrie aéronautique est responsable de 2 % du CO_2 rejeté dans l'atmosphère et deuxièmement, la réduction du coût du vol est important.

Le modèle de carburant de l'avion a été obtenu à partir d'un modèle de performance numérique qui a été créé et validé par notre partenaire industriel à partir des données expérimentales de vol. Une nouvelle méthodologie utilisant ce modèle a été proposée dans cette thèse pour calculer la consommation du combustible pour une trajectoire donnée.

Des paramètres météorologiques tels que le vent et la température ont été pris en compte car ils ont un effet important dans la combustion du carburant. Nous avons choisi comme les prévisions météorologiques fournies par Météo Canada comme modèle de l'atmosphère. Une combinaison des interpolations linéaires et bilinéaires dans la base des données fournies par Météo Canada a permis de trouver les données météorologiques requises.

L'espace de recherche a été modélisé en utilisant des graphiques différents : un mappage des différentes phases de vol et un mappage de l'espace physique où l'avion effectuait son vol.

La trajectoire a été optimisée dans le plan de référence verticale à l'aide d'une combinaison de l'algorithme de recherche en faisceau avec une technique de réduction de l'espace de recherche.

La trajectoire a également été optimisée simultanément dans le plan vertical et latéral de référence tout en respectant une contrainte requise d'arrivée en utilisant trois algorithmes métaheuristiques différents: la colonie d'abeilles artificielles, et l'optimisation des colonies de fourmis.

Les résultats ont été validés en utilisant le logiciel FlightSIM ®, un système de gestion de vol commercial, un algorithme de recherche exhaustif et des vols effectués à partir de flightaware®. L'utilisation de tous les algorithmes a conduit à la réduction des coûts de vol.

Mots-Clés : Trajectoire, Optimisation, Metaheuristic, Algorithmes, Flight, Avionique

APPLICATION OF METAHEURISTIC AND DETERMINISTIC ALGORITHMS FOR AICRAFT REFERENCE TRAJECTORY OPTIMIZATION

Alejandro MURRIETA MENDOZA

ABSTRACT

Aircraft reference trajectory is an alternative method to reduce fuel consumption, thus the pollution released to the atmosphere. Fuel consumption reduction is of special importance for two reasons: first, because the aeronautical industry is responsible of 2% of the CO_2 released to the atmosphere, and second, because it will reduce the flight cost.

The aircraft fuel model was obtained from a numerical performance database which was created and validated by our industrial partner from flight experimental test data. A new methodology using the numerical database was proposed in this thesis to compute the fuel burn for a given trajectory.

Weather parameters such as wind and temperature were taken into account as they have an important effect in fuel burn. The open source model used to obtain the weather forecast was provided by Weather Canada. A combination of linear and bi-linear interpolations allowed finding the required weather data.

The search space was modelled using different graphs: one graph was used for mapping the different flight phases such as climb, cruise and descent, and another graph was used for mapping the physical space in which the aircraft would perform its flight.

The trajectory was optimized in its vertical reference trajectory using the Beam Search algorithm, and a combination of the Beam Search algorithm with a search space reduction technique.

The trajectory was optimized simultaneously for the vertical and lateral reference navigation plans while fulfilling a Required Time of Arrival constraint using three different metaheuristic algorithms: the artificial bee's colony, and the ant colony optimization.

Results were validated using the software FlightSIM®, a commercial Flight Management System, an exhaustive search algorithm, and as flown flights obtained from flightaware®. All algorithms were able to reduce the fuel burn, and the flight costs.

Keywords: Trajectory, Optimization, Metaheuristic, Algorithms, Flight, Avionics.

TABLE OF CONTENTS

Page

INTRODUCTION 1
0.1 Fuel Burn Reduction Efforts Before Airborne 1
0.2 Satement of the problem 2
0.3 Research Objetives 5
0.4 Methodology 6
0.4.1 Aircraft Numerical Performance Model 6
0.4.2 Weather Forecast Model 7
0.4.3 Beam Search 8
0.4.4 Search Space Reduction 8
0.4.5 Artificial Bee's Colony 9
0.4.6 Golden Section Search 10
0.4.7 Ant Colony Optimization Algorithm 10
0.5 Literature Contribution 11

CHAPTER 1 LITERATURE REVIEW 11
1.1 Fuel Burn Reduction Efforts Before Airborne 11
1.2 Fuel Burn Reduction Efforts During Airborne 12

CHAPTER 2 RESEARCH APPROACH AND THESIS ORGANIZATION 19
2.1 Thesis Organization 20
2.1.1 First Journal Paper 21
2.1.2 Second Journal Paper 21
2.1.3 Third Journal Paper 22
2.1.4 Fourth Journal Paper 23
2.1.5 Fifth Journal Paper 23

CHAPTER 3 NEW METHODOLOGY FOR NAV FLIGHT TRAJECTORY COST CALCULATION USING A FMS PERFROMANCE DATABASE 25
3.1 Introduction 26
3.2 Methodology 33
3.2.1 The Conventional Flight 33
3.2.2 The Performance Database (PDB) 33
3.2.3 Flight Cost 35
3.2.4 Trajectory Calculation Method 37
3.3 Results 50
3.3.1 Flight Comparison 50
3.3.2 Cost Index Effect 56
3.3.3 Computation Time For Different Trajectories 57

Page

3.3.4 Cruise Aircraft Distance Between Weight Update Points 58
3.4 Conclusion 59

CHAPTER 4 NEW REFERNCE TRAJECTORY OPTIMIZATION ALGORITHM FOR A FLIGHT MANAGEMENT SYSTEM INSIPIRED IN BEAM SEARCH 61
4.1 Introduction 62
4.2 Methodology 68
4.2.1 Numerical Performance Model 68
4.2.2 Flight Cost Computation 70
4.3 The Optimization Algorithm 72
4.3.1 The Search Space: A Decision Graph 72
4.3.2 Problem Definition 74
4.3.3 The Beam Search Algorithm 77
4.4 Exhaustive Search Algorithm 87
4.5 Results 87
4.5.1 The Optimism Coefficient Effect 88
4.5.2 The Beam Search Algorithm and the Exhaustive Search 92
4.5.3 The Beam Search Algorithm and Results Obtained from the FMS/PTT 95
4.5.4 The Beam Search Algorithm Considering Wind Influence 96
4.6 Conclusion 98

CHAPTER 5 AIRCRAFT VERTICAL ROUTE OPTIMIZATION BY BEAM SEARCH AND INITIAL SEARCH SPACE REDUCTION 101
5.1 Introduction 102
5.2 Methodology 109
5.2.1 The Studied Flight 109
5.2.2 The Numerical Performance Model 109
5.2.3 ..... Flight Cost: Fuel burn and Flight Time Computations Using the Numerical Performance Model 111
5.2.4 Interpolations: Computing the Required Value from the Numerical Performance Model 112
5.2.5 The Flight Cost Computation: Fuel Burn and Considerations 114
5.2.6 Problem Definition: The Vertical Reference Trajectory Optimization 116
5.3 The Optimization Algorithm 118
5.3.1 Algorithm's Input 119
5.3.2 Search Space Reduction Module (SSRM) 120
5.3.3 The Vertical Reference Trajectory Search Space: the Graph Construction 122
5.3.4 The Vertical Reference Trajectory Search Space: the Graph Construction 123

Page

5.3.5 The Bounding Function: Heuristics....124
5.3.6 Weather presence in the bounding function....126
5.4 Results....127
5.4.1 Standalone Algorithms Results Comparison....127
5.4.2 The reference trajectory flight cost between the trajectory provided by a FMS and the trajectory of the developed algorithm....131
5.4.3 The reference trajectory flight of the developed algorithm compared to a real lateral trajectory....132
5.5 Conclusion....134

CHAPTER 6 4D AIRCRAFT EN-ROUTE OPTIMIZATION ALGORITHM USING THE ARTIFICIAL BEE COLONY....137
6.1 Introduction....138
6.2 Methodology....144
6.2.1 Flight Cost....144
6.2.2 The Studied Flight: The Search Space....148
6.2.3 The Optimization Algorithm....156
6.2.4 Algorithm Summary....173
6.3 Results....176
6.3.1 Number of Iterations' Influence on the Resulting Trajectory....176
6.3.2 The ABC´s Robustness....182
6.3.3 Real Flights Study....184
6.3.4 Multiple Flights Fuel Reduction....188
6.4 Conclusion....190

CHAPTER 7 3D AND 4D AIRCRAFT REFERENCE TRAJECTORY OPTIMIZATION USING THE ANT COLONY OPTIMIZATION....193
7.1 Introduction....194
7.2 Numerical Models and the Search Space....200
7.2.1 Fuel Consumption: The Numerical Performance Model....200
7.2.2 Fuel Burn Computation....203
7.2.3 Flight Cost....205
7.2.4 Weather Information....205
7.2.5 The Search Space....208
7.3 Introduction to the Ant Colony Optimization Algorithm....212
7.3.1 Bio Mimicry and Metaheuristic algorithms....212
7.3.2 The Ant Colony In Nature....212
7.3.3 The ACO algorithm implementation for trajectory optimization....214
7.4 3D Reference Trajectory Optimization....215
7.4.1 ACO First Stage: 3D – Module 1 (M1)....215
7.4.2 ACO First Stage: 3D – Module 2 (M2)....217

Page

7.4.3 ACO First Stage: 3D – Module 3 (M3) 219
7.4.4 Functioning 220
7.5 4D Reference Trajectory Optimization: RTA Fulfillment 221
7.5.1 RTA (4D) First Module – M1 223
7.5.2 RTA Second Module – M2: 224
7.5.3 RTA Third Module – P3: 225
7.5.4 RTA Functioning 226
7.6 Results 227
7.6.1 The ACO algorithm trajectory comparison with the geodesic trajectory 227
7.6.2 The ACO algorithm versus a real as flown flight plan results 233
7.6.3 The ACO algorithm versus different as flown flights 235
7.6.4 The Required Time of Arrival 238
7.7 Conclusion 243

CHAPTER 8 DISCUSSION OF RESULTS 245

CONCLUSION AND RECOMMENDATIONS 247

BIBLIOGRAPHY 250

LIST OF TABLES

Page

Table 3.1 PDB Description 34

Table 3.2 IAS/Mach Crossover Altitude Approximation 43

Table 3.3 Aircraft Characteristics 51

Table 3.4 Platforms Comparisons for Validation 51

Table 3.5 Algorithm execution time 57

Table 4.1 Numerical Performance Model Sub-Databases 69

Table 4.2 Aircraft General Characteristics 87

Table 4.3 Flight Distances for Different Trajectories 88

Table 4.4 Optimism Coefficient Results of a Winnipeg to 89

Table 4.5 Optimism Coefficient Results of a Winnipeg to 91

Table 4.6 Algorithm Performance for Aircraft A 93

Table 4.7 Algorithm Performance for Aircraft B 94

Table 4.8 Algorithm Performance for Aircraft C 95

Table 4.9 Optimal Flight Provided by the Exhaustive Search Algorithm 98

Table 4.10 Results Provided by the “Beam Search” 98

Table 5.1 Polluting emissions generated by fuel burn and their effects 103

Table 5.2 Numerical Performance Model Sub-databases 110

Table 6.1 Solution Evolution when Varying the Number of Iterations 177

Table 6.2 Fuel consumption for different optimized flights 189

Table 7.1 Average fuel burn and flight cost optimization for three 232

Table 7.2 Average savings percentages of fuel burn and flight 237

Table 7.3 Absolute flight time differences from the RTA for the optimized trajectories in 3D 239

LIST OF FIGURES

Page

Figure 0.1 Comparison between the vertical and the lateral reference trajectories 4

Figure 0.2 Weather Canada forecast grid 6

Figure 1.1 CDA versus conventional descent trajectories 12

Figure 3.1 Conventional Flight 33

Figure 3.2 Typical Interpolations flowchart 38

Figure 3.3 Acceleration Calculations Flowchart 40

Figure 3.4 Acceleration Example 41

Figure 3.5 Climb Calculations Procedure 45

Figure 3.6 Cruise Calculations Procedure 47

Figure 3.7 Descent Calculation Procedure 49

Figure 3.8 Aircraft A Calculated Fuel Burned 52

Figure 3.9 Aircraft A Calculated Flight Time 52

Figure 3.10 Aircraft B Calculated Fuel Burned 54

Figure 3.11 Aircraft B Calculated Flight Time 54

Figure 3.12 Aircraft C Calculated Fuel Burned 55

Figure 3.13 Aircraft C Calculated Flight Time 56

Figure 3.14 Cost Variation by Cost Index 57

Figure 3.15 Cruise Segment Size Influence 58

Figure 4.1 Typical reference vertical flight profile 72

Figure 4.2 Vertical reference trajectory optimization graph-tree 74

Figure 4.3 Decision tree example for given solution 76

Figure 4.4 Beam search for reference trajectory optimization algorithm's flowchart 85

Figure 4.5 FMS/PTT and algorithm cost comparison 96

Figure 4.6 Flight Cost Solution Difference between the Exhaustive.....97

Figure 5.1 General Interpolation Graphic Representation.....112

Figure 5.2 Tree-Graph for the available combinations.....117

Figure 5.3 The SSRM and the BSOAM effects reducing the combinations. 119

Figure 5.4 Algorithm Modules Execution Order.....119

Figure 5.5 Space Reduction Parameters Definition.....121

Figure 5.6 Algorithm Execution Time as a Function of Flight Distance.....128

Figure 5.7 Fuel Burn Difference Between the Optimal Trajectory 129

Figure 5.8 Execution time ratio comparison between three different optimization 130

Figure 5.9 Flight Cost Between the FMS and the Developed 132

Figure 5.10 Lateral reference trajectory for a Vancouver to Cancun flight.....133

Figure 5.11 Optimal Vertical Reference Trajectory Provided by the Exhaustive 134

Figure 6.1 Weather Interpolations 148

Figure 6.2 Vertical Dimension Search Space 149

Figure 6.3 Lateral Dimension Boundaries.....150

Figure 6.4 Five Generic Trajectories Inside the Time Boundaries and its Mach Variation .152

Figure 6.5 Function fc used in the Golden Section Search Algorithm 154

Figure 6.6 Creation of the lateral trajectories with respect to the reference trajectory.....160

Figure 6.7 Lateral trajectories representation as placed on the Earth 161

Figure 6.8 Fixed grid and dynamic grid comparison.....162

Figure 6.9 Flight Time Limits.....163

Figure 6.10 Left: Trajectories in the time dimension. 165

Figure 6.11 Trajectory Mutation on the Vertical Dimension 168

Figure 6.12 Last Waypoint Mach Number Selection 172

Figure 6.13 ABC Algorithm Flowchart 174

Figure 6.14 Last candidate trajectories (green), and the 178

Figure 6.15 Lateral Trajectories: Optimized trajectory (red), the geodesic trajectory (blue),178

Figure 6.16 Altitude versus Distance (a). 179

Figure 6.17 Lateral offset depending on the travelled 180

Figure 6.18 Optimized lateral trajectory (red) nearby the reference (blue), ...181

Figure 6.19 Optimized altitude and speed (red).....182

Figure 6.20 Saved amount of fuel for 800 trajectories 183

Figure 6.21 Flight Plan From Edmonton to Punta Caña.....184

Figure 6.22 Lateral offset of the optimized trajectory (red), around the reference 185

Figure 6.23 Lateral optimized trajectory (red), around the reference.....186

Figure 6.24 Optimized vertical trajectory (red), and 187

Figure 6.25 Speed versus distance variation for the optimized speed (red), ...188

Figure 7.1 Inputs required to obtain fuel flow during cruise from 201

Figure 7.2 Inputs required to obtain fuel burn during climb.....201

Figure 7.3 Fuel burn and horizontal travelled distance.....202

Figure 7.4 Interpolation schema for the fuel flow interpolation.....204

Figure 7.5 Weather information interpolation around a plane.....206

Figure 7.6 Wind effect on the aircraft ground speed 208

Figure 7.7 3D graph for trajectory optimization, every circle is a node (or a waypoint) 209

Figure 7.8 Consecutive nodes available to create a trajectory.....210

Figure 7.9 Graph of available Mach numbers in the search space 211

Figure 7.10 Only the three closest Mach numbers can be selected. 211

Figure 7.11 The ant colony organization in nature 214

Figure 7.12 Algorithm stages: First, the ACO is used to find the optimal 3D trajectory, then the combination of Mach numbers that fulfill the RTA constraint 215

Figure 7.13 3D random trajectory creation and its parameters per trajectory 216

Figure 7.14 Module 3 – Route building on a 3D simplified model, following 220

Figure 7.15 ACO functioning 221

Figure 7.16 Mach number selection 223

Figure 7.17 Mach number selection with a random trajectory 224

Figure 7.18 Mach number ant selection 226

Figure 7.19 Fuel burn saving percentage for different flights 228

Figure 7.20 Flight cost saving percentage for several flights 228

Figure 7.21 ACO trajectory and geodesic lateral reference trajectory. 230

Figure 7.22 ACO trajectory step climb 231

Figure 7.23 Mach number selection 232

Figure 7.24 Lateral reference trajectory for the ACO algorithm and for a real flight plan 233

Figure 7.25 ACO vertical trajectory step climb and the as flown vertical trajectory 234

Figure 7.26 Mach number selection 235

Figure 7.27 Fuel burn savings percentages for different flights 236

Figure 7.28 Flight cost savings percentage for several flights 236

Figure 7.29 Flight time comparison between the 3D flight time 241

Figure 7.30 Flight time comparison between the 3D flight time 242

Figure 7.31 Flight time comparison between the 3D flight time 242

LIST OF ABREVIATIONS

ABC	Artificial Bees Colony
ACO	Ant Colony Optimization
ADS-B	Automatic Dependent Surveillance + Broadcast
ASI	Airspeed Indicator
ATC	Air Traffic Control
ATM	Air Traffic Management
BSOA	Beam Search Optimization Algorithm
B&B	Branch and Bound
CARATS	Collaborative Actions for Renovation of Air Traffic Systems
CDA	Continous Descent Approach
CDO	Continous Descent Operations
CO2	Carbon Dyoxide
EOM	Equations of Motion
FMS	Flight Management System
GSS	Golden Section Search
IAS	Indicated Air Speed
ISA	International Standard Atmosphere

LNAV	Lateral Navigation
NextGen	Next Generation Air Transport System
NOx	Nitrogen Oxides
PDB	Performance Database
PSO	Particle Swarm Optimization
PTT	Part Task Trainer
RNP	Required Navigation Performance
RTA	Required Time of Arrival
SESAR	Single European Sky ATM Research
SSRM	Search Space Reduction Module
STAR	Standard Terminal Approach
SID	Standard Instruments Departure
TAS	True Airspeed
TBO	Trajectory Based Operations
TOC	Top of Climb
TOD	Top of descent
UAV	Unmanned Aerial Vehicles
UTC	Universal Time Coordinate

VNAV Vertical Navigation

LIST OF SYMBOLS

α	Used to give priority to a given parameter
C_{opt}	Optimism Coefficient
β	Used to give priority to a given parameter
φ	Maximum Turning Angle
ϕ	Aircraft Heading
γ	Specific heat of air

INTRODUCTION

0.1 Fuel Burn Reduction Efforts Before Airborne

In our day to day lives, there are many different things that are taken for granted. This is the case of exotic fruits, fish, meat, exclusive drinks in the supermarket, flowers. In addition, traveling is made fast for various purposes, such as to cross the globe in order to meet colleagues in conferences, visit loved ones, take vacations, or knowing that an organ can help our lives in less than a day of travel in case of an emergency.

There are goods that require extreme care in order to arrive quickly and safely to our cities. These goods are diverse, such as specialized machinery, high sensitive microchips, chemicals oils, among many others. These goods allow economic growth and development (ATAG, 2016) (Boeing, 2016b). All their transportation is possible due to aviation.

In the early 1900s, the traveling from Europe to America would take around 5 days. Nowadays cruises slowed it down a little bit to allow passengers to enjoy their trip for 7 days (New York, USA to Southampton, UK). However, for a business trip, for someone that is time constrained, or for perishing/urgent goods, this is a long time considering that a flight from New York to London takes roughly 7 hours. The same time consideration is true for a continental flight within the US and Canada, driving a car from Montreal to Vancouver would take a non-stop trip of 46 hours, while taking a flight would take around 5h30m. Even in the presence and acceptance of strict regulations, aircraft remain the safest way of traveling.

As aviation is the most convenient way of traveling, it has allowed connecting the world in ways that no one in history could imagine. This way has brought as a consequence the exchange of ideas, better understanding of other cultures, human development, commercial exchange, and economy growth. Lives have been saved due to rescue missions, it sets the basis of space conquest, and it is fundamental for national defense systems (ATAG, 2005).

However, as all in life, it comes with a price: huge quantities of fossil fuel are required to power a flight, thus it pollutes.

0.2 Statement of the problem

As it will be shown in the main part of this thesis, a quantity of 30 Tons of fuel is required to power engines of a long haul aircraft in regards to generate enough thrust to perform a flight is within normal fuel quantities to fly a long haul flight. Evidently, burning fuel brings as a consequence the emissions of polluting particles and gas molecules to the atmosphere. One of these gas molecules is carbon dioxide (CO_2) which is known to be one of the main drivers of global warming. It has been reported that in 2015, around 740 million of CO_2, were released to the atmosphere by airlines (ATAG, 2016). This number corresponds to 2% of the global CO_2 emissions. The aeronautical industry has set itself the goal of reducing CO_2 generation to 50% of the levels recorded in 2005 by 2050 (ATAG, 2016; ICAO, 2010).

For this reason, different products, services, and organizations have been developed or created to address and thus, solve the pollution emissions problem. Such an organization is the Green Aviation Research & Development Network (GARDN) in Canada. This organization aims to encourage the development of environment friendly aircraft technology. The research methodology and results shown in this thesis were developed under this organisation program frame in the projects: “CMC-1 Optimized Descents and Cruise”, “Flight Management Performance Optimization II”, and “CMC- 21 (Project Extension): Flight Management Performance Optimization III”. This program is being developed by the Laboratory of Applied Research in Active Controls, Avionics and Servoelasticity (LARCASE) based at the ETS in partnership with the internationally known CMC Electronics – Esterline, specialist in avionics systems.

Many different factors affect the aircraft trajectory efficiency. Some of the most important ones taken into account in this thesis are the weight, altitude, weather, aircraft direction, and speed. As a general rule, it is desirable to have a light aircraft (less weight needs to be lifted,

so less energy) flying at high altitudes (air density is lower, thus drag is lower), at low speeds (less fuel pumped into the engines), at favourable winds (tailwind increases the ground speed). However, aircraft are normally heavy due to passengers, cargo, fuel and the aircraft itself. Aircraft speed is coupled with altitudes and the weight itself (heavy aircraft normally require a faster speed). Besides, an additional important factor such as *time cost* is associated to speed. As time cost items, the crew's salary, arriving in time to passenger's connections, maintenance factors are included, among others. Altitude is associated with weight as heavy aircraft cannot climb to the highest altitudes, as it is mechanically impossible. This is because of the fact that not enough thrust can be provided to generate the required lift to climb. As aircraft loses weight due to fuel burn, extra lift is liberated allowing the aircraft to change altitudes to more efficient ones (Lovegren, 2011). This change of altitude during cruise is called step climb. Finding the combination of altitudes and speeds during the flight that reduces the fuel consumption while shortening the flight time as much as possible, is already a difficult task, yet there is another stochastic parameter that strongly influences flights, thus complicates this task: Weather.

Winds and temperatures influence aircraft speed, and temperature influences as well engine's efficiency. Headwinds slow down the aircraft's speed relative to the ground by requiring more thrust to keep the same ground speed (similar as when we are biking against the wind), high temperatures increase the sound of speed (thus the speed for a given Mach number), but these high temperatures at the same time make the engines to be less efficient. It is then desirable to guide the aircraft to zones where weather is favourable to the flight optimization objectives (reduce fuel burn, flight time, and flight cost).

The aircraft reference trajectory optimization problem consists in finding the combination of speeds, altitudes, geographical zones and the right time when the aircraft's trajectory is able to satisfy the desired objective function; normally, a compromise between fuel burn and flight time should be done.

Trajectories are somewhat complex. Luckily, there is an airborne device able to manage these trajectories this: the Flight Management System (FMS). Among many different tasks, this devices is able to manage a trajectory, and in some cases even optimize it (Collinson, 2011). However, current FMS only optimizes the vertical profile (speeds and altitudes following a pre-defined lateral reference trajectory).

For this thesis, there are three types of reference trajectories taken into account: lateral trajectories, vertical trajectories, 3D, and 4D.

Lateral reference trajectories consist of the combinations of waypoints followed by an aircraft from the initial to the destination point; this is known as *ground track*. Vertical references trajectories consist of the combination of altitudes and speeds followed by an aircraft while flying over a pre-defined lateral reference trajectory. 3D reference trajectories consist of combining the lateral and the vertical reference trajectories. Finally, 4D reference trajectories consist of finding the optimal 3D reference trajectory and fulfilling a Required Time of Arrival (RTA) constraint. Figure 0.1 shows the lateral and the vertical reference trajectories for a flight from New York to San Francisco.

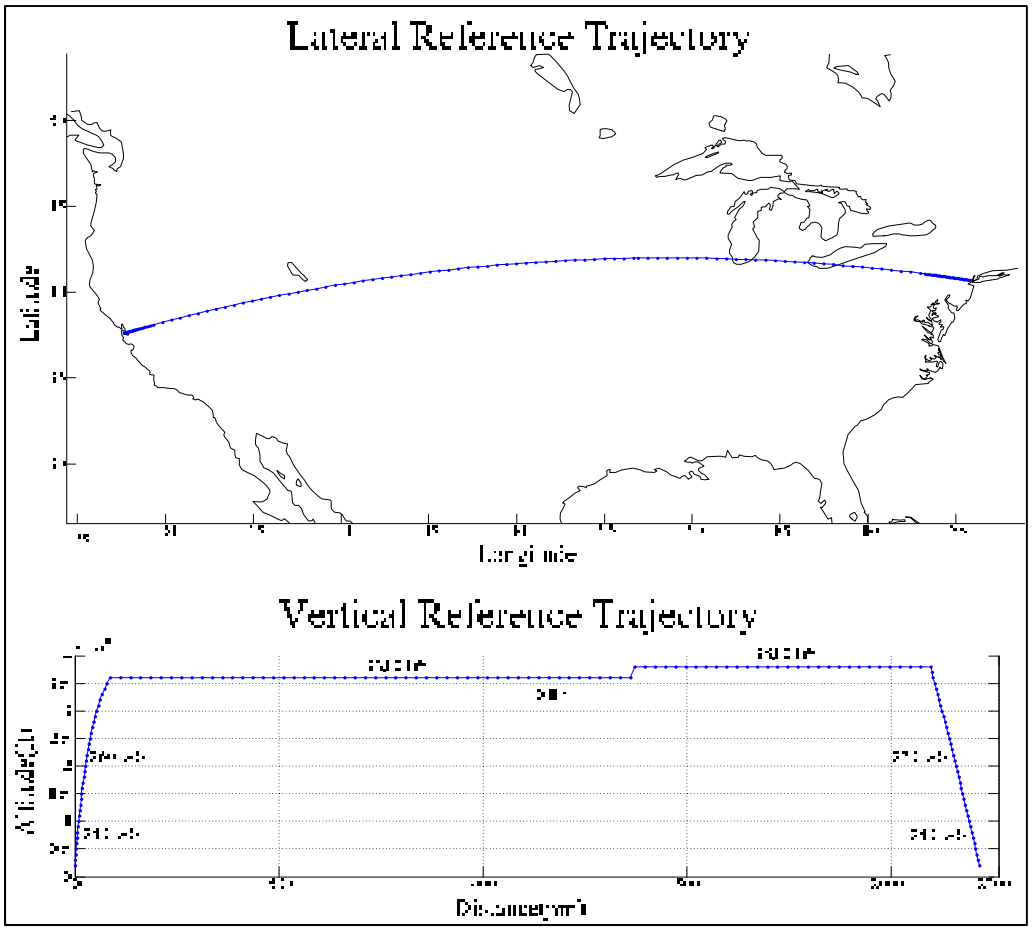

Figure 0.1 Comparison between the vertical and the lateral reference trajectories

0.3 Research Objectives

The goal of this research is to implement different algorithms in order to find the combination of altitudes, speeds, waypoints that minimizes the flight cost in a 3D and a 4D search space. The 3D search space contains aircraft geographical position, altitudes, and speeds. The 4D search space contains geographical positions, altitudes, speeds, and a time constraint (RTA) as it will be explained later in details in this thesis.

To reach this ultimate goal, the main research objective can be divided in different sub-objectives:

1) Develop and validate a methodology to compute the fuel burn using a numerical performance model.

2) Optimize the flight reference trajectory in 3D following a pre-defined ground track and compare the obtained results with a commercial FMS results using deterministic algorithms.

3) Optimize the flight reference trajectory in 4D using different metaheuristic algorithms and evaluate their capabilities of performing this optimization task.

4) Compare algorithms results against the FMS of reference results and all algorithms results and against *as flown* flights to validate fuel burn reduction potential.

5) All flights should include the dynamic wind and temperature forecast.

Different databases and algorithms were implemented during this research. A list f each one of the databases and methodology to compute the flight cost and optimize the flight trajectory will be briefly introduced as they are explained in details in the publication during the rest of this thesis.

0.4 Methodology

0.4.1 Aircraft Numerical Performance Model

As the main goal of this research is reducing fuel consumption in order to reduce pollution, a numerical model able to provide the fuel burn for a given flight condition is required. This model was obtained from experimental flight data, and it was provided by our industrial partner. Thousands of txt lines were converted at the LARCASE to Matlab® files. As it will be described in Chapter 3, a series of linear Lagrange interpolations will provide the required fuel burn and the flight time at a specific position of the aircraft in flight. These values will ultimately define the flight cost.

This model is composed from a database for various in flight phases: Climb, Acceleration, Cruise, and Descent. Inputs change depending on the flight phase, but altitude, speed, and

weight are always the required inputs independent of the flight plan. The outputs are fuel flow or fuel burn and travelled distance.

0.4.2 Weather Forecast Model

Weather forecast was obtained from the public data available from Weather Canada. Among the different models developed by Weather Canada, the one called Global Deterministic Prediction System (GDPS) was selected as it contains information for the entire world. This information is provided in the form of a grid with a resolution of 0.6° x 0.6°, or 601 latitude x 301 longitude points –see Figure 0.2. 28 different pressure altitudes were taken into account to cover the aircraft flying altitudes. This model is also provided in 3 hours blocks beginning at 00h Universal Time Coordinate (UTC) to 144h. This is at 00h, 03h, 06h, 09h… and so on. However, for this research data up to 36 hours are taken into account for overnight flights. This is for example, flights taking off late at night in the west of the US to Europe. For this research, it is considered that data longer than 36 hours is non-reliable, as weather can changed at any time.

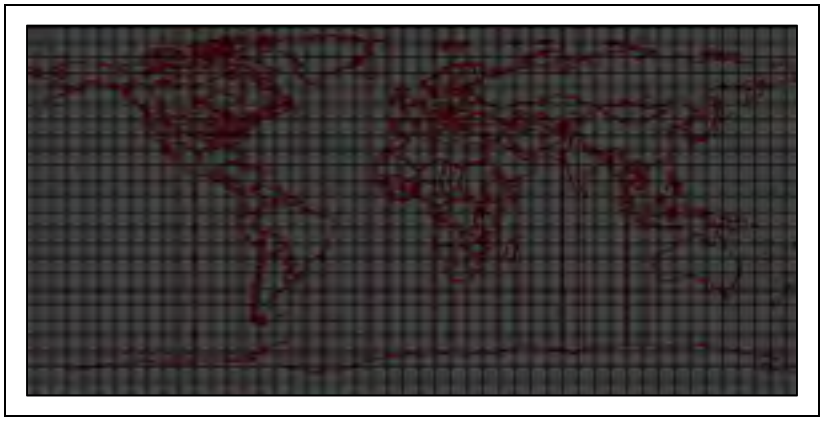

Figure 0.2 Weather Canada forecast grid
(Source: Weather Canada)

As weather data is provided in the form of a grid, the aircraft is normally located between four different points, for two different pressure altitudes, and two different time block. For this reason, a combination of lineal (pressure altitudes, and time), and bilinear (geographical

position between four points) interpolation is required in order to obtain the weather information for the exact aircraft's location in terms of altitude, time, and geographical position. This bilinear interpolation is explained in detail in (Murrieta-Mendoza, 2013) and (Jocelyn Gagné, 2013).

The most important information obtained from this model is temperature, wind speed, wind direction (relative to north), and sea level pressure.

0.4.3 Beam Search

Beam search is a variation of the most well-known Branch and Bound algorithm. Both algorithms explore a given graph (interconnection of nodes with different values connecting them), and decides if it is worth it to expand a node, or if it is a better strategy to remove that node from the graph. Removing a node and all its descendants is desired as it reduces the number of available combinations, thus the computation time.

As discussed in more detail in Chapter 4 and Chapter 5, there are two main differences between the beam search and the branch and bound algorithm. The first one is that the beam search can limit the number of nodes to be evaluated per level, while Branch and Bound might evaluate all nodes. The second main difference between both algorithms is that Branch and Bound rejects a node only if it can be proved that the optimal solution is not under that node. On the contrary, Beam Search is less restrictive at cutting nodes as any node can be rejected even if it has not been proved that the optimal solution is under that node.

0.4.4 Search Space Reduction

The search space reduction algorithm was developed at the LARCASE and it has been presented in (Murrieta-Mendoza, 2013) and (Murrieta-Mendoza & Botez, 2014b). This algorithm aims to reduce the number of combinations of Mach numbers and altitudes a given aircraft can select from in order to determine the most economical flight, the optimal solution

for this research purposes. As a consequence, the algorithm computation time is reduced and the optimal solution was always found.
This algorithm was implemented before the beam search algorithm in this thesis in order to reduce the computation time in Chapter 5

0.4.5 Artificial Bee's Colony

The Artificial Bee's Colony (ABC) mimics the behaviour of honeybees swarm in the search space exploration in their search of food sources. Bees gather as much food as possible using the minimum energy possible.

After flying a given trajectory from the hive to the food source, the *employer* bee modifies a little bit their trajectory trying to improve it. This is called *mutation*. Once the bee is back in the hive, it dances trying to impress the *on-looker* bees. If bees are impressed, they follow the dancing bee to its trajectory. In that way, more bees can improve the trajectory, as the number of trajectories increment. When a given trajectory does not improve anymore, *scout* bees separate from the group looking for new food sources.

This methodology is presented in Chapter 6, and has the aim to implement aircraft reference trajectory.

0.4.6 Golden Section Search

The Golden Section Search is a technique developed to find the minimum or maximum of a unimodal function. It differs from other techniques such as the bisection method as the number of iterations is reduced in order to find a solution. This technique was used in Chapter 6 in order to find the Mach number candidate that fulfills an arrival time constraint.

0.4.7 Ant Colony Optimization Algorithm

This algorithm reproduces the behaviour of ants in the exploration of space in the search for food sources. One ant by itself is not capable of finding and transporting enough food to nourish the hive. While exploring paths for a food source, ants release a chemical component called *pheromone*. In this way, ants can find their ways back to the hive, and to the food source. Other ants can find this pheromone trail and might follow it. It is more probable that an ant will follow a path if this contains a high pheromone concentration. Ants following the shortest path do a higher number of round trips, thus more pheromone is deposited in the trail. After some time, the shortest path contains the highest amount of pheromone. Hence, most of the ants follow this path. This algorithm was used in Chapter 7.

0.5 Literature Contribution

This research brought as a result the publication of three journal articles, and up to date, the submission of two more journal articles (all five composing this thesis), the publication of two additional journal papers (related, but out of the scope of this thesis), the writing and submission of three more journal articles, a book chapter on metaheuristic algorithms for flight optimization.

A total of 10 journal papers have been written and 26 articles were presented at international conferences in Canada, the United States, Mexico, the Netherlands, France, and Austria. 4 posters were presented in the students aerospace forum 2012 and 2013, the American Romanian Academy of Arts and Science, and CASI 2017, 1 was received the poster award in Navigation Modeling in the American Romanian Academy of Arts and Science Conference. In addition, 4 more media non- reviewed articles were published in the substance blog at the ETS.

CHAPTER 1

LITERATURE REVIEW

The aeronautical industry is aware of the environmental consequences of flying, and it has been committed to reduce pollution, especially CO_2. For this reason, after analysis, there are three main action pillars that were proposed in order to attain the industry environmental goals. The first pillar can be defined as known technology, operations and infrastructure measures, the second pillar is economic measures (taxation to pollution), and the third pillar concerns the use of biofuels, and additional new-generation technologies (ATAG, 2009).

The literature review found in this Chapter is intended to be an introduction to the aircraft reference trajectory problem. A more extensive literature review will be presented in the beginning of each of the subsequent chapters, as each one of them is an article by itself containing its personalized literature review.

1.1 Fuel Burn Reduction Efforts Before Airborne

There is research focusing in the aircraft itself in order to reduce fuel burn such as reducing the components weight, a single engine taxi, the use of electrical motors to control the landing gear, the use of ground power unit instead of the auxiliary power unit, and washing engines prior departure to reduce the exhaust gas temperature. All these alternatives were discussed in (McConnachie, Wollersheim, & Hansman, 2013).

There are also studies aiming to improve the traffic flow through the taxiways. It is important to minimize the queuing time before using the in-service runaway as it was estimated that 740 million gallons of fuel were burned due to delays and congestion in taxiways; different ways to improve these delays were explained in (Balakrishnan, 2016). The Branch and Bound technique was also proposed to develop a tool to optimize the aircraft routing and optimization (P. J. Godbole, A. G. Ranade, & R. S. Pant, 2014).

1.2 Fuel Burn Reduction Efforts During Airborne

When the aircraft takes off there is not enough optimization opportunity, as the motor setting is already defined. Following take-off, rigid procedures also known as Standard Instruments Departures (SID) restraining altitudes and ground track trajectories are followed. Once the SID is executed, it is preferable to climb to the cruise altitude and free the airspace.

Descent, on the other hand, has presented important fuel burn savings. Although a restraining procedure called Standard Terminal Arrival Route (STAR) is also followed to control the traffic flow of arrival aircraft into airport; a technique which enables fuel burn saving was developed and further implemented in different airports for various aircraft. This technique is called Continuous Descent Approach (CDA) or Continuous Descent Operations (CDO). With this technique, aircraft has to descend following a constant slope in the IDLE setting, the lowest fuel consuming setting. This constant slope is followed until the beginning of the STAR procedure. Conventional descent consists of a *step descent*. The aircraft follows a constant cruise segment, then a descent to the next altitude. The aircraft once again follows a constant cruise segment at the new altitude; it then descends to the next altitude, and so on until the STAR procedure begins. Important quantities of fuel are burned while the cruise segment is flown, especially at low altitude where engines are not as efficient. The descent comparison of a conventional descent and the CDA/CDO can be seen in Figure 1.1.

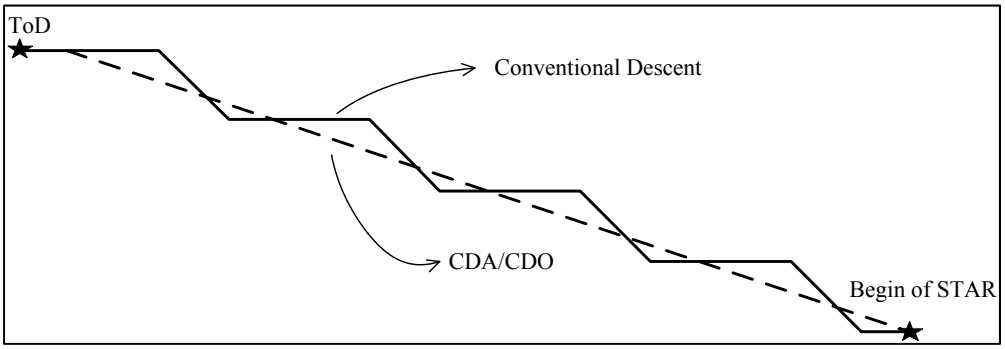

Figure 1.1 CDA versus conventional descent trajectories

This technique has reported savings of around 40% of the typical descent (ATAG, 2010). This procedure has been quantified at different airports such as in the SFO where up to 1860

kg of fuel were saved for a Boeing 747-200 (Jin, Cao, & Sun, 2013). The average gallons of fuel savings for the Atlanta airport was of 37 gallons of fuel (around 120 kg) per flight, and 49 gallons of fuel (around 136 kg) per flight in the Miami International Airport (Sprong, Klein, Shiotsuki, Arrighi, & Liu, 2008). Another study in Los Angeles International Airport showed an average fuel burn reduction of 64 kg per flight (Clarke et al., 2013).

Another trajectory related improvement is the Required Navigation Performance (RNP). By using satellite technology such as GPS instead of typical Navaids, such as Instrument Landing System (ILS) or VHF Omnidirectional Range (VOR), more accurate position measurements are provided. For this reason, aircraft can fly in more *hostile* environments, such as between two mountains, close to lower terrain, and etcetera. The aircraft is thus allowed to perform sharper curve turns. As a consequence, the flown distance for arrival procedures is reduced due to a 3D flight planning[1]. A shorter distance evidently reduces the required fuel burn. Boeing reported fuel savings ranging from 400 lb (182 kg) to 800 lb (362 kg) for a 737 at Oslo Airport, 600 lb (272 kg) for the RNP procedure implemented in San Francisco, and 200 liters per flight for the Calgary International Airport (Boeing, 2016a). Another Boeing study reported savings up to 746 lbs (338 kg) of fuel for the Seattle-Tacoma International Airport for a Boeing 737-800W (Boeing, 2015). NAV Canada reported fuel savings ranging from 265,000 to 285,000 liters of fuel annually for the Kelowna and Abbotsford airports in Canada (Marasa, 2010). Evidently, all this fuel reduction brings as a consequence a reduction of polluting emissions released to the atmosphere. Not only Boeing, but also Airbus has tested this technology using the A321 in the Goteborg Landvetter Airport in Stockholm (Airbus, 2011).

The aircraft cruise phase presents great optimization potential. Different studies have demonstrated that aircraft do not fly at their optimal speeds and altitudes (Luke Jensen, Hansman, Venuti, & Reynolds, 2014; Luke Jensen, Hansman, Venuti, & Reynolds, 2013; Luke Jensen, Tran, & Hansman, 2015; Turgut et al., 2014). This inefficiency could be due to

[1] http://www.airbus.com/innovation/proven-concepts/in-operations/required-navigation-performance/

airlines politics, weather constraints, traffic constraint, or even ATM inefficiency. Traffic has been increasing in the last years, and is expected to keep growing worldwide. This increase may further give new traffic management problems. For this reason, developed countries are aiming to improve the airspace by adding better procedures, and new infrastructure by funding traffic management programs. Among these programs, the two leading ones are the Next Generation Air Transportation System[2] (NextGEN) in North America and the Single European Sky ATM Research[3] (SESAR) in Europe. Both programs use the concept of free flight, or Trajectory Based Operations (TBO).

Under current regulations, aircraft require Air Traffic Control Services to provide surveillance, control, and separation (Hoekstra, 2001). For this reason, an aircraft should submit their flight intentions (or flight plan) to Air Traffic Management (ATM) before take-off. In this way, different air traffic control zones will manage the flight of the aircraft from the departure gate to the arrival gate. ATM may authorize the flight plan as it is, or it might require modifications (Field, 1985). Anyhow, this flight plan is later loaded into the Flight Management System (FMS). This flight plan is not written in stone as it can be changed during flight due to crew request or ATC request (Nolan, 2010). Among the reasons to modify the original flight plan, ATC might request this change in the flight plan due to storms, aggressive wind patterns, too much congestion in a given zone, by enabling a no fly zone, an aircraft emergency, and etcetera. The crew itself can request a change in the flight plan due to finding a more economical route (i.e. change of altitude or change of speed), or in declaring an emergency. As all these changes are strictly controlled by ATC, aircraft are obliged to follow the provided flight plan. This manual workload is difficult for ATC, and normally, it does not allow a given aircraft to fly its optimal route.

The concept of *free flight*, where aircraft could fly its preferred route is an ongoing project. Under this concept, the separation responsibility is given to the pilot. The first stage for the

---

[2] https://www.faa.gov/nextgen/
[3] http://www.sesarju.eu/

free flight concept is requiring all aircraft to be equipped with an Automatic Dependent Surveillance + Broadcast (ADS-B) [4]. This device is able to send information about the aircraft position, altitude, speed, and direction.

If all available aircraft are able to broadcast this information, it is possible to estimate their position and intentions. In this way, negotiation between these aircraft could begin and each one of them might be able to fly its optimal trajectory. For this *free flight* concept, a negotiation protocol and an automatically control would be required as proposed by (Gardi, Sabatini, Ramsamy, Marino, & Kistan, 2015), while another strategy would be the management of conflicts for different trajectories (Ruiz, Piera, Nosedal, & Ranieri, 2014). The Particle Swarm Optimization (PSO) mixed with different optimization techniques was implemented by (Cobano, Alejo, Heredia, & Ollero, 2013) to negotiate conflict resolution by finding economical solutions.

However, the fully free flight concept will not see the light any time soon as avionics systems, procedures, standards, infrastructure, among others will be developed and tested. The closest new operations technology that is being implemented is the *4D trajectories*. Using this concept, aircraft report their current position while a Required Time of Arrival (RTA) is required by ATC. It is the basis of future free flight as aircraft will report future position at a precise time at a given waypoint. Aircraft will use this information to grant themselves clearance and to guarantee their separation.

Different algorithms based on this 4D concept have been proposed. One of the first authors to tackle this problem published interesting results and analysis was documented by (Chakravarty, 1985) where trajectories were computed, and were further analyzed to evaluate the delays costs from the end of cruise to a point during cruise. Later, (Sam Liden, 1992a) studied achieving the RTA time by minimizing the flight cost and by taking into account winds, and by removing discontinuities caused by flying at pre-defined altitudes. Many

[4] http://www.navcanada.ca/en/products-and-services/pages/on-board-operational-initiatives-ads-b.aspx

different authors have addressed this problem using the aircraft's Equations of Motion and other techniques. Among these techniques, a Multiphase mixed integer optimal control algorithm was developed by (Soler-Arnedo, Hansen, & Zou, 2013) where contrail formation zones where avoided. Contrails are vaporized water exhausted by engines which form clouds during certain meteorological conditions. Dynamic programming with neural networks was implemented by (Hagelauer & Mora-Camino, 1998). Dynamic programming including a new technique which reduces the search space as the algorithm is executed was proposed by (Miyazawa, Wickramasinghe, Harada, & Miyamoto, 2013)

3D reference trajectory optimization algorithms are also developed, and they can be considered to be the actual algorithms. Genetic Algorithms (GA) have been implemented for the lateral and vertical reference trajectories (Roberto Salvador Félix-Patrón & Botez, 2015; Roberto S. Félix-Patrón, Kessaci, & Botez, 2014). The vertical reference trajectory was optimized by using the Golden Section Search (R.S. Felix Patron, Botez, & Labour, 2013).

The lateral reference trajectory avoiding contrails was as well optimized by treating the problem as a two boundary value problem in (Sridhar, Ng, & Chen, 2013). In another research, the vertical reference trajectory was optimized over the computed optimal lateral reference trajectory (Ng, Sridhar, & Grabbe, 2014). Following current ATC restrictions and by treating the aircraft vertical reference trajectory problem as trajectory patterns, an algorithm was developed to optimize the reference trajectory (Valenzuela & Rivas, 2014). An hybrid optimal control technique was evaluated for a complete flight consisting in climb, cruise, one step climb and descent phases for four different wind profiles (Franco & Rivas, 2015). Graph Search techniques using the Dijstrkra's and A*algorithms were also implemented to optimize the reference flight trajectory (Dicheva & Bestaoui, 2014; Rippel, Bar-Gill, & Shimkin, 2005).

As it can be seen in this literature review, and as it will be shown in detail in the next Chapters, there is a gap in algorithms needed to evaluate the 4D aircraft reference trajectory. Metaheuristic algorithms, besides the 3D Genetic Algorithms work by Felix-Patron at

LARCASE, are practically not implemented for the solving of the 3D or 4D reference trajectory problem, which constitutes one of this research originality.

Different Graph search based algorithms using deterministic and metaheuristic techniques for the 3D and the 4D aircraft reference trajectories are proposed, and their optimization capabilities are studied in this thesis. All algorithms were developed with the ultimate goal of reducing fuel burn and when possible, the total flight cost.

CHAPTER 2

METHODOLOGY

The research project performed on the commercial aircraft reference trajectory optimization presented in this thesis was divided in the following phases:

- Statement of the problem;
- Development of a methodology to compute the flight cost using a numerical performance model and its validation;
- Development of an algorithm to compute the vertical reference trajectory optimization;
- Reduction of the vertical reference trajectory optimization computation time;
- Implementation and evaluation of the fuel saving potential for the 4D reference trajectory optimization using and adapting the three following algorithms: the Artificial Bee's Colony, and the Ant Colony Optimization.

Each of these phases was implemented in the given order to achieve the research objective. All of these phases, except the statement of the problem, were developed in Matlab®. Flights were evaluated using an electronic commercial Flight Management System software called Past Task Trainer (PTT), FlightSIM® developed by Presagis®, Exhaustive search algorithms, and *as flown* flight information provided by FlightAware®.

The first phase consisted in performing an extensive literature review; identification of the main problems to overcome, and creation of the working plan. The results of this analysis were published in five journal articles (and some others) composing this thesis.

The second phase consisted in developing a methodology able to compute the flight cost using a numerical performance model by taking into account all the flight phases, such as initial climb, acceleration, climb in Mach, cruise, changes of altitude during cruise, descent in Mach, deceleration, and descent in IAS was developed.

The third phase consisted in developing an algorithm able to find the combination of speeds and altitudes that provide the most economical vertical reference trajectory. The developed algorithm brings the “novelty of modeling the set of speeds and altitudes as a graph”. The Beam Search algorithm was implemented to obtain the most economical combination within the created graph.

After obtaining interesting results at the end of the second phase, in the fourth phase, it was of interest to develop an algorithm with a faster convergence than the algorithm in the 3rh phase to the optimal vertical reference trajectory. It is desirable to implement this algorithm in a low computation power device such as the Flight Management System (FMS), as requested by CMC Electrnonics - Esterline.

The fifth phase consisted in developing three different metaheuristic algorithms to study the state of the art trajectory optimization: the *4D reference* trajectories. These algorithms couple the vertical reference trajectory, the lateral reference trajectory and respect a Required Time of Arrival (RTA) constraint. These algorithms focused in optimizing the cruise phase of long haul aircraft.

Metaheuristic algorithms were selected as they have proved to solve problems in a dense search space with great success. In this phase three journal papers were published that have shown the optimization potential of three different metaheuristic algorithms.

2.1 Thesis Organization

As main author, five peer-review journal papers are contained in this manuscript-based thesis. Three journal papers are already published and the other two journal papers are under review for publication. These papers are presented from Chapter 3 to Chapter 7.

Dr. Ruxandra Mihaela Botez, as co-author for all journal and conference papers, supervised the progress of this research through regular meetings in collaboration witn CMC Electronics

– Esterline team. In the second and third journal papers, Bacherlor's students Miss Lauranne Ternisien and Mr. Bruce Beuce were included as co-authors as they contributed in the development and testing of the Beam Search algorithm. In the fourth paper, Bachelor's student Mr. Audric Bunel was included as co-author as he was implicated in the development and testing of the Artificial Bee's Colony algorithm. In the fifth paper and last paper, Mr. Antoine Hamy was included as co-author due to his implications in the development and validation of the Ants Colony Optimization Algorithm

2.1.1 First Journal Paper

This paper is entitled "New Methodology For Nav Flight Traectory Cost Calculation Using a FMS Performance Database" was published in the Journal of Aerospace Information Systems in September 2015.

This paper details the methodology followed to systematically obtain data from the numerical performance model and, to execute the set of Lagrange interpolations to compute the flight cost which is composed of fuel burn and flight time. The technique developed in this paper focuses on the vertical reference trajectory cost determination, but it can also be applied for the lateral reference trajectory flight cost as it consists in the cruise phase.

Different flight costs obtained from the model were compared against the results obtained from the PTT and FlightSIM®. Results showed that the developed methodology is able to compute accurate flight costs in a short time.

2.1.2 Second Journal Paper

This paper entitled "New Reference Trajectory Optimization Algorithm For a Management System Inspired in Beam Search" was accepted for publication in the Chinese Journal of Aeronautics and it will be published in August 2017.

In this paper, the combination of speeds, and altitudes was modeled as a graph. The Beam Search algorithm was implemented to search the most economical combination of nodes in order to find the most economical flight trajectory.

This paper also proposed a heuristic which estimates the flight cost with limited flight information. This heuristic took into account step climbs and weather. This heuristic allows rejecting nodes and thus reduces the computation time. A parameter called *optimism coefficient* was introduced to control the effectiveness of this heuristic in estimating the flight cost. Hypothetic and *as-flown* flights were studied to determine the trajectory optimization potential. This algorithm was able to provide the optimal solution, or extremely good sub-optimal solutions.

2.1.3 Third Journal Paper

This paper entitled “Aircraft Vertical Route Optimization By Beam Search And Initial Search Space Reduction” was submitted to the AIAA Journal of Aerospace Information Systems in May 2017, and it is currently under review.

After the development of an algorithm able to find the optimal reference trajectory, it was of interest to reduce its computation time. The graph search composing the candidate solution was reduced using a search space reduction technique developed for reference trajectory optimization problems. This algorithm was also able to find the optimal trajectory or really good sub-optimal in just a fraction of the time required by the Beam Search algorithm in the second journal paper. This time reduction made this algorithm a good candidate to implement it in the FMS.

2.1.4 Fourth Journal Paper

This paper is entitled “4D Aircraft En-Route Optimization Algorithm Using the Artificial Bee Colony” was submitted to the AIAA Journal of Aerospace Information Systems in December 2016 and it is currently under review.

This paper is the first one of the series of papers evaluating a metaheuristic algorithm and its application for an optimized trajectory. Besides the series of Genetic Algorithms developed by Felix-Patron at LARCASE, this is one of the first metaheuristic algorithms developed to solve the reference trajectory optimization problem.

The Artificial Bees’ Colony (ABC) was implemented for the 4D reference trajectory optimization. For this algorithm, the reference trajectory was simultaneously optimized in the vertical and lateral reference trajectories while complying with the Required Time of Arrival (RTA) constraint.

This paper also set the main framework for other algorithms to be developed (to be presented in Section 2.1.5 and Section 2.1.6). The search space was modeled under the form of a graph, the solution is given as a combination of altitudes, speeds, and geographical position and the optimization of the cruise phase only for long-haul flights.

2.1.5 Fifth Journal Paper

This paper entitled “3D and 4D Aircraft Reference Trajectory Optimization Using the Ant Colony Optimization” was submitted to the AIAA Journal of Aerospace Information Systems in February 2017 and it is currently under review.

The exploration of metaheuristic algorithms is continued in this paper. An algorithm inspired from the Ant Colony Optimization (ACO) was implemented to find the most economical

combination of waypoints (altitudes, and geographical coordinates) while fulfilling the RTA constraint.

Besides finding the optimal trajectory, this algorithm finds a 3D trajectory which has a low cost with a high potential to fulfill the RTA constraint. The Mach number for this 3D trajectory is then optimized to provide a 4D reference trajectory. This allows exploring the fuel savings that are lost as a consequence of fulfilling the RTA constraint.

CHAPTER 3

METHODOLOGY FOR VERTICAL-NAVIGATION FLIGHT TRAJECTORY COST CALCULATION USING A PERFROMANCE DATABASE

Alejandro Murrieta-Mendoza and Ruxandra Mihaela Botez

École de Technologie Supérieure / Université du Québec, Montreal, Canada
Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticty

This article was published in the AIAA Journal of Aerospace Information Systems, Vol. 12, No. 8, August 2015, pp. 519-532
doi: http://dx.doi.org/10.2514/1.I010347

Résumé

L'optimisation de trajectoire de vol permet de réduire considérablement les coûts de vol et les émissions polluantes. La puissance de calcul d'équipement embarqué tel que le système de gestion de vol est limité. Conséquemment, une méthode rapide pour calculer le coût d'un vol a été développée. De nombreux systèmes de gestion de vol utilisent des tables avec des données expérimentales pour chaque phase de vol expérimental. Cette base de données est appelée base de données de performance. Dans cet article, le coût de la trajectoire de vol est calculé en utilisant une base de données de performance au lieu d'utiliser les équations de mouvement. La trajectoire à calculer se compose de la montée, de l'accélération, de la croisière, de la descente, et de la décélération. L'influence de l'altitude de croisement pendant la montée et la descente, ainsi que les montées en escalade en croisière ont été considérées. Les interpolations linéaires de Lagrange ont été appliquées à la base de données de performance pour calculer les valeurs requises. En fournissant un poids au décollage, les coordonnées initiales et finales et le plan de vol souhaité, le modèle de trajectoire fournit les coordonnées du sommet de la montée et du début de la descente, le carburant brûlé et le temps de vol nécessaire pour suivre le plan de vol donné. La précision des coûts de trajectoire calculés avec la méthode proposée a été validée avec les résultats du modèle aérodynamique donné par FlightSIM®, développé par Presagis®, et avec le coût de

trajectoire donné par le système de gestion de vol de référence. Les résultats ont montré que pour les mêmes trajectoires de référence et pour les mêmes entrées, le coût calculé par la méthode proposée dans cet article est proche ce ceux fournis par FlightSIM® et par le système de gestion de vol de référence.

Abstract

Trajectory optimization has been identified as an important way to reduce flight cost and polluting emissions. Due to the power capacity limitations in airborne devices such as the Flight Management System (FMS), a fast method should be implemented to calculate the full trajectory cost. Many FMSs use a set of look up tables with experimental data for each flight phase, and are called Performance Database (PDB). In this paper, the trajectory flight cost is calculated using a PDB, instead of using classical equations of motion. The trajectory to be calculated is composed by climb, acceleration, cruise, descent and deceleration. The influence of the crossover altitude during climb and descent, as well as step climbs in cruise was considered. Lagrange linear interpolations were performed within the PDB discrete values to calculate the required values. By providing a takeoff weight, the initial and final coordinates and the desired flight plan, the trajectory model provides the Top of Climb coordinates, the Top of Descent coordinates, the fuel burned and the flight time needed to follow the given flight plan. The accuracy of the trajectory costs calculated with the proposed method was validated with an aerodynamic model in FlightSIM®, software developed by Presagis® and with the trajectory cost given by the FMS benchmark of reference. Results showed that for the same reference trajectories, and for the same inputs, the cost computed by the method proposed in this paper is close to the costs provided by FlightSIM, and by the FMS benchmark or reference.

3.1 Introduction

Air transportation has become an important pillar to economical interchange transporting around 35% of the world's trade value according with a recent study by the Air Transport Action Group (ATAG) (ATAG, 2014). Air transportation is also one of the preferred ways to

travel with occupancy of 79% in 2013, higher than other means of transportation. This activity has motivated air transportation providers to increment their fleets. The ATAG, in the same study, estimated that for the year 2032, there will be around 41,000 airplanes in operation, doubling the airplanes available in 2010. This could potentially increment the fuel consumption, thus the polluting emissions.

Aircraft fly with conventional fuel, which has the disadvantage of releasing contaminant emissions such as carbon dioxide (CO_2), nitrogen oxides, (NOx), hydrocarbons, water, among others (Murrieta-Mendoza, 2013). From these emissions, CO_2 is of special interest due to its contribution to global warming. The aeronautical industry is responsible for approximately 2% of the total CO_2 released to the atmosphere. The aeronautical industry has set as goal to reduce their CO_2 emissions in 2050 to the 50% of the emissions recorded in 2005 (IATA, 2011). Hydrocarbons and NOx are released to the atmosphere at high altitudes which causes damage to the ozone layer (Crutzen, 1970; Nojoumi, Dincer, & Naterer, 2009; Robinson, 1978).

Different technologies are being developed and implemented to reduce emissions and noise such as engines improvements, air-to-air refuelling (Nangia, 2006), the use of biofuels (ATAG, 2009), weight reduction by the use of advanced avionics, replace heavy materials for composites and aircraft improvements such as winglets (Freitag, 2009), morphing wings (Koreanschi, Sugar Gabor, & Botez, 2014), among others. These improvements focused on fuel reductions.

Optimal flight planning and optimal flight trajectory have been identified as an area of opportunity to reduce fuel consumption, thus reducing polluting emissions. Air authorities in North America and Europe have considered this challenge and are redefining their airspace with the Next Generation Air Transportation System (NextGen) (Theunissen, Rademaker, & Lambregts, 2011) and the Single European Sky (SESAR).

There are two reference trajectory components that can be optimized, using the Boeing terminology, the reference trajectory for the Vertical Navigation (VNAV), and the reference trajectory for the Lateral Navigation (LNAV). There also exists the possibility of coupling of both. The VNAV reference trajectory is composed by the speeds and altitudes to follow during flight and the LNAV reference trajectory is the geographical coordinates set the aircraft has to follow.

Trajectory optimization is of interest, not only for conventional aircraft, but also for unmanned aerial vehicles (UAV). Wilburn in (Wilburn, Perhinschi, & Wilburn, 2013b) developed a methodology to implement a clothoid planner for UAVs, and in (Wilburn, Perhinschi, & Wilburn, 2013a) 3D trajectories were planned using the concept of the Dubins's particle.

Due to the proximity of airports to cities, the descent phase has been of great concern regarding trajectory optimization. Murrieta *et al* in (Murrieta-Mendoza, Botez, & Ford, 2014) and Dancila *et al.* in (R. Dancila, Botez, & Ford, 2013) using information from the European Monitoring and Evacuation Programme and the European Environment Agency (EMEP/EEA) emission inventory guidebook developed methods to calculate the fuel needed and the pollution generated by the execution of the missed approach (go-around) procedure.

The Continuous Descent Approach, descent with idle motors at a constant angle, has provided excellent results. Stell in (Stell, 2010) developed a way to estimate the location of the Top of Descent in order to execute a Continuous Descent Approach (CDA) at a 3 degrees slope. Kwok-On *et al.* in (Kwok-On, Anthony, & John, 2003) measured the fuel burned reduction effectiveness of the CDA in the airport of Louisville, Kentucky in the USA. Clarke et al in (Clarke et al., 2013) proposed a method to implement the CDA at Los Angeles International Airport. Bronsvoort *et al.* in (Bronsvoort, McDonald, Boucquey, Garcia-Avello, & Besada, 2013) performed a study using an Aircraft Intent Description Language, equations of motion (EOM), and existing communication protocols to point out the importance of communication to accurately predict the CDA trajectory. Wind estimation has also been of

importance for descent trajectory generation. Oliveira in (Oliveira, Quachio, & Cugnasca, 2014) modeled the wind as a multivariate Gaussian stochastic process. This algorithm was improved by using the idea of periodically discarding part of the history and using conditional distributions.

Different studies from the MIT have shown the potential of savings in trajectory optimization. Jensen in (Luke Jensen et al., 2014; Luke Jensen et al., 2013) studied speed and altitude in over 200,000 flights within the continental United States using Enhanced Traffic Management System data. These flights were then compared against optimal speeds and optimal altitudes from models developed with information from Piano-X. These studies showed the optimization opportunities available since many aircraft do not fly at their optimal speed or/and altitude. Bonnefoy in (Bonnefoy & Hansman, 2010) using data from the Bureau of Transport Statistics (BTS) studied the fuel burn benefits and the airlines scheduling consequences of reducing cruise speeds and discussed how to mitigate these consequences.

In the decade of 1970 a device called Flight Management System (FMS) was conceived and widely introduced to aircraft in the decade of 1980 (Sam Liden, 1994). This device, among other tasks, is responsible of managing the flight plan helping reducing the crew workload and is responsible of computing the optimal trajectory (Collinson, 2011). There exist ground teams that calculate the flight trajectory before airborne. However, it is of interest to provide the FMS with the autonomy to calculate the optimal trajectory because pre-calculated routes can change during or before airborne. The available processing power in the FMS to perform all of its tasks is limited and only a fraction of this processing power is dedicated to finding the optimal trajectory. Thus, fast methods to perform trajectory calculations are required. Many different researches worked in trajectory optimization; some of them could have the potential to be implemented in a FMS.

Lidén was one of the first researchers to study the flight trajectory for FMSs. In (Sam Liden, 1985) Lidén studied the speed control to minimize flight costs, the effects of arriving too late

at a given destination were considered, and a way to estimate the Cost Index (CI) was discussed. Franco and Rivas in (Franco & Rivas, 2011) used the Equations of Motion (EOM) of an aircraft at constant altitude and constant heading to calculate its optimal profile in the presence of winds. For the flight cost, arriving early or late to a given destination was modeled as a penalty. This paper suggested the use of singular optimal control to find the optimal flight profile. Sridhar *et al.* in (Sridhar et al., 2013) utilized EOM to develop an algorithm to calculate an optimal trajectory considering wind and avoiding regions favorable to contrail formation. Fays *et al.* in (Fays & Botez, 2013) by using information from Boeing, implemented metaheuristic algorithms to find the optimal trajectory while avoiding obstacle along the routes. These obstacles can be interpreted as no fly zones, traffic or potential weather affected routes. Dicheva in (Dicheva & Bestaoui, 2014) using the EOM as the mathematical model of an airborne launch vehicle and solved the shortest path problem using A*. In this paper fixed and mobile obstacles were considered and the capability of changing the final destination en-route was implemented.

Miyazawa *et al.* in (Miyazawa et al., 2013) used the EOM to propose a technique called "Moving Search Space Dynamic Programming" to calculate the optimal trajectory of a flight considering weather and required arrival time (RTA). Hok et al. in (Ng et al., 2014) used the Base of Aircraft (BADA) parameters to model the aircraft using the EOM, and used dynamic programming to find the optimal trajectory considering winds. For altitude optimization, weight was tracked with a pre-defined optimal weight-altitude relation; at the defined threshold the aircraft would change altitude. Hagelauer and Mora-Camino in (Hagelauer & Mora-Camino, 1998) also considered EOM to implement dynamic programming with neuronal networks in order to optimize the trajectory. This paper points out the fact that the FMS uses databases instead of EOM. To emulate these databases from this approach, the EOM were discretized.

Thus, for optimization algorithms, the literature shows that researches normally use historical data, which is not convenient to calculate real time optimization algorithms, and models based on the EOM. Solving a set of differential equations (EOM) to find the optimal

trajectory in a device with limited processing power such as the FMS might be time consuming. Normally, the FMSs do not use these equations, but a set of look up tables with experimental data called Performance Database (PDB). Different authors have considered a PDB as the aircraft model to calculate the optimal trajectory.

Felix-Patron *et al* in (R. S. Felix-Patron, Botez, & Labour, 2012; R.S. Felix Patron et al., 2013; Roberto Salvador Felix Patron, Oyono Owono, Botez, & Labour, 2013) used a PDB, implemented step climbs, the golden section search and genetic algorithms to optimize the VNAV profile of an aircraft considering all flight phases of flight for three different commercial aircraft. Dancila *et al.* in (Bogdan Dancila, Botez, & Labour, 2012; B Dancila, Botez, & Labour, 2013) used a PDB, developed a method to estimate the fuel burn for cruise segments at constant altitudes.

Gagné *et al.* in (Jocelyn Gagné, Murrieta, Botez, & Labour, 2013) used the PDB to develop an optimization algorithm by semi exhaustive search for a commercial aircraft considering weather information along the trajectory. All stages of flight were analyzed and step climbs opportunities were identified and performed during cruise to optimize the trajectory cost. Murrieta *et al.* in (Murrieta Mendoza & Botez, 2014) developed an algorithm to reduce the search space and calculate the optimal VNAV profile. Sidibé in (Sidibe & Botez, 2013) implemented a conventional dynamic programming algorithm to find the optimal VNAV by using the numerical model provided by the PDB.

Authors have also tried to couple LNAV and VNAV using a PDB as it was the case of Murrieta in (Murrieta-Mendoza, 2013) where five lateral parallel trajectories were evaluated taking advantage of weather parameters after determining the VNAV. Felix-Patron *et al.* in (R. Felix-Patron, Kessaci, & Botez, 2013) used genetic algorithms and information from the PDB to develop an algorithm to find the optimal LNAV trajectory at constant altitude taking advantage of tailwinds. Later, Felix-Patron *et al.* in (Roberto S. Félix-Patrón & Botez, 2014; Félix Patrón, Berrou, & Botez, 2014) used genetic algorithms to calculate the optimal VNAV(with and without step climbs) profile, then coupled the optimal LNAV trajectory.

Murrieta in (Murrieta-Mendoza & Botez, 2014a) used the Dijkstra's Algorithm to find the optimal route taking winds and temperature into account.

In spite of the number of optimization algorithms using a PDB present in the literature, there are just a few papers that provided a brief overview of the method to calculate the trajectory cost, they mostly focus in the optimization method. Felix-Patron *et al.* in (R. Felix-Patron, Botez, & Labour, 2013) described a way to reduce interpolations. Nevertheless, only a brief part of climb calculations was explained.

The objective of this paper is to develop a novel method to calculate the cost of the reference vertical navigation reference trajectory using a PDB. The costs for different trajectories were computed using the same inputs, and then compared using the novel method developed in Matlab®, a real time commercial flight simulator such as FlightSIM®, and by a commercial FMS benchmark known as Part Task Trainer (PTT). The PTT is a FMS simulator which gives as an output a given trajectory cost. Results showed that the novel algorithm provides good results.

The method developed in this paper is useful to compute flight costs for flight optimization algorithms. The optimization algorithms using a PDB did not fully explain the methodology to compute the trajectory cost. The PDBs, which were created from aircraft experimental data including fuel burn, were provided by our industrial partner and were used for commercial FMSs. The trajectory method is composed by a climb, acceleration, cruise, descent and deceleration. Concepts such as crossover altitude effects during climb, during descent, and step climbs in cruise were considered. Since the PDB is a set of discrete data, Lagrange linear interpolations are performed within the PDB tables to calculate the trajectory cost. Considerations to be taken into account for this application are discussed.

This paper is organized as follows: First the conventional flight to be calculated is defined. Secondly the PDB with all its sub-databases is described. Thirdly the Flight Cost and its

variables are discussed. Fourthly, the trajectory calculation method is described. Finally, simulation using the methodology results are presented and discussed.

3.2 Methodology

3.2.1 The Conventional Flight

The flights analyzed in this paper are standard commercial flights composed of constant climb at 250/240 kts Indicated Airspeed (IAS), an acceleration phase to the desired constant climb speed. At a given altitude the Airspeed Indicator (ASI) reference is changed from IAS to Mach while as continue climbing in Mach until the Top of Climb (TOC). During cruise step climbs (change of altitudes) may be executed. At the Top of Descent, the descent is calculated in the inverse order as the climb. Due to regulation, aircraft should not fly faster than 250 IAS below 10,000 ft. Figure 3.1 shows the order of these flight phases.

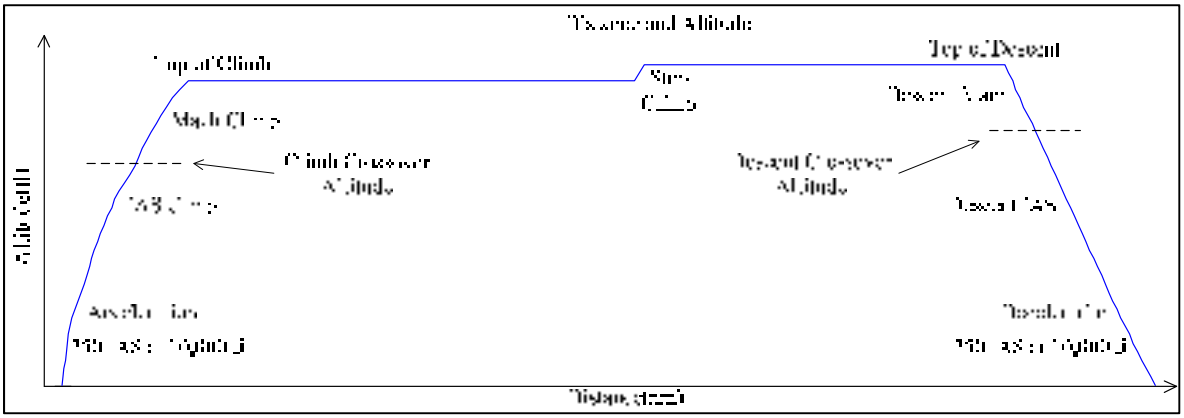

Figure 3.1 Conventional Flight

3.2.2 The Performance Database (PDB)

The PDB is a numerical experimental model of the aircraft and is divided in 7 sub-databases, one for every phase of flight. In order to obtain data from the PDBs, all the input parameters

have to be provided. Table 3.1 describes the inputs and outputs of the different sub-databases.

Due to the fact that the sub-databases are defined in a discrete form, only certain parameters are available in the PDB inputs. For example, the International Standard Atmosphere (ISA) deviation temperature may be given in steps of 10 degrees, gross weight in steps of 15,000 kg, and so on. For the cases when the values required to be introduced in the inputs are not exactly the ones available in the PDB, interpolations between the outputs of the available data are performed as it will be shown below. Altitudes are normally given at a 1,000 ft step, for this reason, to obtain data from the PDB, the aircraft should be placed at 1,000 ft step.

Table 3.1 PDB Description

Sub-database	**Inputs**	**Outputs**
Climb IAS	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burned (kg) Horizontal traveled distance (nm)
Climb acceleration	Gross weight Initial IAS (knots) Acceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burned (kg) Horizontal traveled distance (nm) Altitude needed (ft)
Climb Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burned (kg) Horizontal traveled distance (nm)

Table 3.1 PDB Description (Continue)

Sub-database	Inputs	Outputs
Cruise Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel flow (kg/hr)
Descent Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burned (kg) Horizontal traveled distance (nm)
Descent deceleration	Gross weight Initial IAS (knots) Deceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burned (kg) Horizontal traveled distance (nm) Altitude needed (ft)
Descent IAS	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burned (kg) Horizontal traveled distance (nm)

3.2.3 Flight Cost

Flight costs are not only measured with the fuel required to complete a given flight, but also with flight time. The typical equation used in the literature, to define flight cost is given in Eq. (3.1).

$$Cost = Fuel_{Burned} + CI * Flight_{Time} * 60 \tag{3.1}$$

Where $Fuel_{Burned}$ (kg) is the total fuel burned during flight, $Flight_{Time}$ (hr) is the total duration of flight and *CI* (kg/hr) is the Cost Index. *CI* is the variable that relates the cost of time to cost of fuel.

It is of interest to develop a method to calculate the trajectory cost in terms of the variables present in Eq. (3.1): fuel burned and flight time. Fuel burned can be obtained directly from the PDB. However, to calculate Flight Time, the Aircraft Ground Speed (GS) has to be known. In order to calculate the GS, the atmosphere is considered to be the ISA and winds to be null. When speed is given in *IAS*, neglecting the instrument error, *TAS* is calculated with Eq (3.2).

$$TAS_{IAS} = \sqrt{\frac{2a_h^2}{\gamma - 1}\left[\left(\frac{P_0}{P_1}\right)^{(\gamma-1)/\gamma} - 1\right]} \tag{3.2}$$

where a_h is the speed of sound (kts) at the given altitude, γ is the specific heat of air, typically 1.4, P_1 is the static pressure at the given altitude and P_0 is the stagnation pressure in the Pitot tube calculated as in Eq (3.3) where P_s is the sea level pressure.

$$P_0 = P_s\left[\left(\frac{IAS^2(\gamma - 1)}{2a_h^2}\right) + 1\right]^{\gamma/(\gamma-1)} + P_1 \tag{3.3}$$

When the speed is given in Mach, the TAS is obtained with Eq (3.4). Commercial flights are normally in subsonic Mach.

$$TAS_{mach} = mach \cdot a_h \tag{3.4}$$

If no wind effects are considered, TAS equals GS. Knowing TAS and neglecting the path angle, the Flight_Time can be calculated from the *Horizontal_traveled_distance* distance obtained from the PDB (Table 3.1) as shown in Eq (3.5). During cruise the *Horizontal_traveled_distance* is pre-defined by the user as explained below in the cruise Section.

$$Flight\ Time = \frac{Horizontal\ Traveled\ Distance}{TAS} \tag{3.5}$$

The rate of climb and the path angle for climb and descent are not provided in the PDB. However, they can be computed using the data in the PDB as follows in equations (3.6) and (3.7) where *Delta Horizontal Traveled Dist* is the horizontal distance traveled to climb from the *Lower Altitude* to the *Higher Altitude*.

$$Path\ Angle = arctan\left(\frac{Higher\ Altitude\ (ft) - Lower\ Altitude\ (ft)}{Delta\ Horizontal\ Traeled\ Dist\ (ft)}\right) \tag{3.6}$$

$$RoC = \frac{1000\ ft}{Climb\ Time\ (min)} \tag{3.7}$$

3.2.4 Trajectory Calculation Method

Variables such as weight and ISA deviation temperature are rarely the exact discrete values in the PDB. For this reason, as mentioned above, interpolations in the outputs of the PDB are required. The trajectory procedure uses the Lagrange linear interpolations from the PDB outputs at the discrete input containing the desired value. Eq (3.8) describes the Lagrange linear interpolation where f_0 and f_1 are respectively the lowest and highest value to interpolate, either fuel burn, distance or altitude.

$$p_1(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) f_0 + \left(\frac{x - x_0}{x_1 - x_0}\right) f_1 \tag{3.8}$$

The required interpolations path for the majority of the PDB tables, with exception of acceleration and deceleration, is graphically shown in Figure 3.2. The word limit in this figure refers to the interpolation discrete values taken from the PDB that contain the desired value to interpolate. The desired outputs are fuel consumption and horizontal traveled distance. For cruise, only fuel flow is calculated. For acceleration and deceleration, the needed altitude to reach the desired speed is also given as an output.

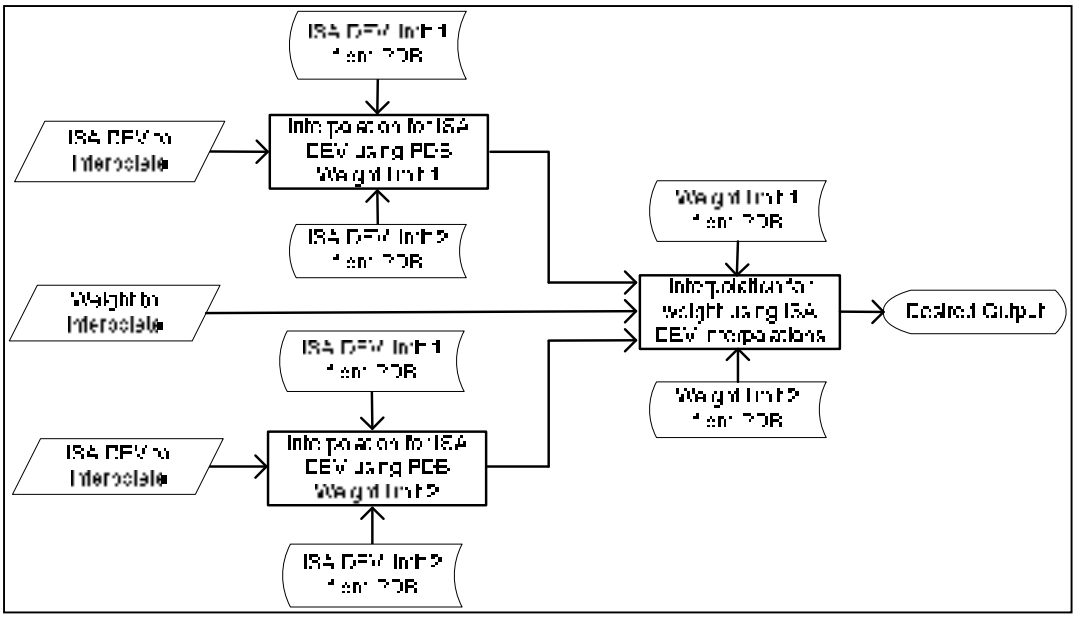

Figure 3.2 Typical Interpolations flowchart

The inputs required for the application of this method are: the aircraft total weight, the LNAV reference trajectory (the waypoints to follow), the altitudes and speeds for the VNAV reference trajectory, as well as the ISA deviation standard temperature at each waypoint. If flying in ISA conditions, the ISA Deviation Standard is considered to be 0 along the flight. Wind information is not required, but if desired, TAS in Eq. (3.5) needs to be changed for GS. GS considers the effects of the winds as fully explained in (Roberto S. Félix-Patrón & Botez, 2014; Murrieta-Mendoza & Botez, 2014a).

Each flight stage has some particularities which will be described next. At the end of each phase, *Fuel burned* and the required *Flight Time* are accumulated to their respective variables in Eq (3.1).

3.2.4.1 Climb IAS

In the method proposed in this paper, for this flight stage, the Sub-database "Climb IAS" from Table 3.1 is used. Because the Take Off procedure is not considered, the flight always initiates at 2,000 ft. At every 1,000 ft, an interpolation is executed to calculate the horizontal traveled distance and the fuel burned to climb at a higher flight level. The fuel calculated is subtracted from the aircraft weight every 1,000 ft to improve accuracy. Using TAS from Eq (3.2) and the horizontal traveled distance obtained, the segment flight time is calculated with Eq. (3.5).

The default speed used to perform the climb from 2,000 ft to 10,000 ft is normally at or below 250 IAS. This is due to the International Civil Aviation Organization (ICAO) regulation, which can also be found in local regulations such as the Code of Federal Regulations 91.117 in the United States and the Canadian Aviation Regulation 602.32 limits speed below 10,000 ft to 250 IAS.

3.2.4.2 Acceleration

Climb speed after 10,000 ft is normally higher than 250 IAS, thus an acceleration is required. The acceleration phase has two different stages: The first stage computes the required fuel, horizontal distance, and altitude to perform an acceleration using the "acceleration" PDB. This stage normally gives a high fuel burned due to the power required to accelerate. Because at the end of the acceleration, the required altitude is rarely a multiple of 1,000 ft, the second is a small climb at the new constant IAS using the "climb IAS" sub database to reach the next multiple of 1,000 ft altitude.

Figure 3.3 describes the interpolations needed to calculate the acceleration outputs. The required speed incremental (*delta speed*) that the aircraft has to accelerate is determined. Interpolations for the required increment speed are performed. If the PDB initial speeds differ from 250 IAS, interpolations in speed may be required. This first set of interpolations is performed for the lower step weight. The same set of interpolations is performed for the highest weight step. Finally, an interpolation between the results for the sets of both weights is performed using the weight of the aircraft at the beginning of the acceleration phase. For flight time calculation, speed is the average of the initial and the final speed. Fuel burned is reduced from total gross weight. The required interpolations for accelerations are provided in Figure 3.3.

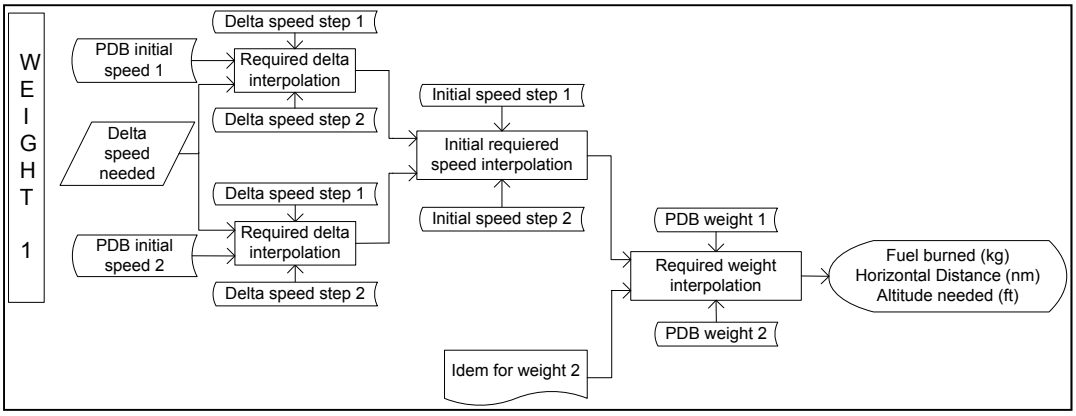

Figure 3.3 Acceleration Calculations Flowchart

The second part of the acceleration phase computations begins after calculating the needed acceleration altitude. As stated before, this altitude is rarely a multiple of 1,000 ft as required by the PDB. Thus a small climb is calculated using the methodology described in Climb IAS. An example is shown in Figure 3.4. In this figure, an acceleration from 250 IAS to 300 IAS is performed at 10,000 ft and finishes at an altitude of 12,520 ft. With the gross weight after acceleration, a climb at the constant new altitude from 12,000 to 13,000 is calculated. The

percentage of the remaining climb (480 ft or 48% of climb) is added to the previously calculated acceleration fuel burned and/or horizontal traveled distance.

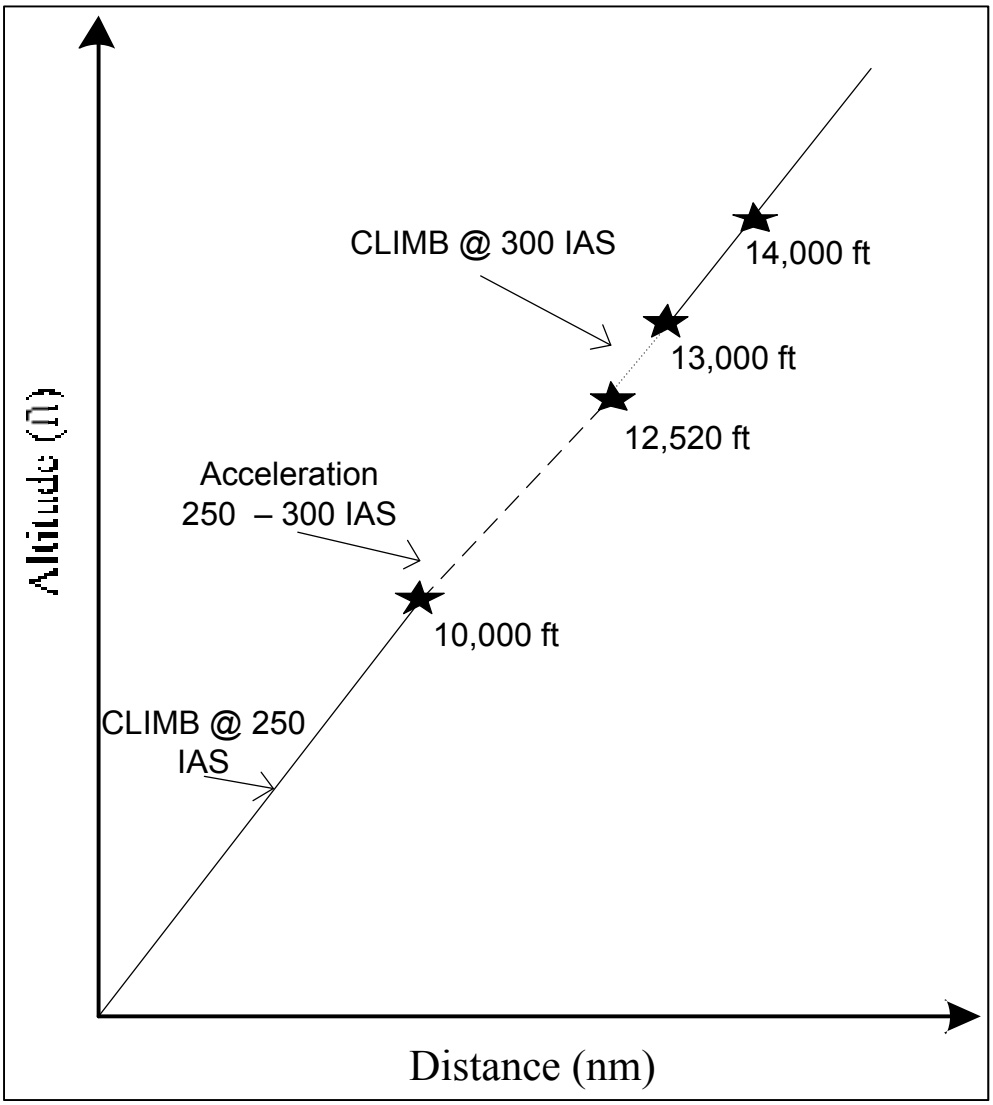

Figure 3.4 Acceleration Example

Eq. (3.9) expresses the total fuel cost added to the acceleration fuel burned.

$$Total\ fuel_{acc} = fuel_{acc} + \left(\frac{A_1 - A_0}{1{,}000}\right)(fuel_{IAS}) \tag{3.9}$$

A_0 is the altitude after the acceleration, A_1 is the next multiple of 1,000 ft altitude after A_0, $fuel_{acc}$ (kg) is the fuel obtained after the acceleration interpolations, and $fuel_{IAS}$ is the fuel needed to climb from altitude A_0 to A_1. A similar equation is used to obtain the total horizontal traveled distance.

3.2.4.3 Climb Mach

When the crossover altitude is reached, the PDB table is changed from "Climb IAS" to "Climb Mach". Keeping the IAS reference may lead the aircraft to fly at higher speeds than the desired Mach speed, sometimes near sonic speeds. Commercial aircraft are not normally designed to fly at these speeds. The crossover altitude can be defined as the altitude where the TAS of IAS equals the TAS of the scheduled mach. An approximation of the crossover altitudes for some Mach/IAS calculated under ISA conditions is shown in Table 3.2. If the crossover altitudes are not multiples of 1,000 ft, a part of the climb has to be calculated using "Climb IAS" sub-database and the other part with the "Climb Mach" sub-database. Similar considerations as the ones in acceleration are performed to define the cost percentage of Climb in IAS and the percentage of climb in Mach.

Using TAS from Eq. (3.4), flight time for this phase is calculated with Eq. (3.5). Climb finishes when the Top of Climb or the maximal altitude is reached. Figure 3.5 provides a flow chart of the steps followed to calculate the climb of the trajectory.

Table 3.2 IAS/Mach Crossover Altitude Approximation

IAS / Mach	**260**	**270**	**280**	**290**	**300**	**310**
0.50	14000	12000	10000	10000	10000	10000
0.56	20000	18000	16000	14000	12000	11000
0.59	22000	21000	19000	17000	15000	13000
0.62	25000	23000	21000	20000	18000	16000
0.65	27000	26000	24000	22000	20000	19000
0.68	30000	28000	26000	24000	23000	21000
0.71	32000	30000	28000	27000	25000	23000
0.74	34000	32000	30000	29000	27000	26000

3.2.4.4 Descent Estimation

Before cruise, an approximate descent horizontal traveled distance is required in order to define an estimated Top of Descent (TOD). In other words, define where cruise ends. The estimation is performed by fetching the horizontal traveled distance for the given altitude and the given Mach directly from the “Mach climb” PDB. A more precise descent is calculated at the end of cruise.

3.2.4.5 Cruise

Cruise is normally the longest phase of flight; it begins at the TOC and ends at the TOD. During this phase, the waypoints to be followed have to be proposed or obtained from a proposed route. For this paper, equidistant great circle points are considered as waypoints. The great circle, or geodesic, is the shortest distance between two points on a sphere.

At every waypoint the fuel burned to attain the present waypoint is reduced from the total aircraft weight. This has an influence on the calculations accuracy. The more waypoints and

shorter distance from one to another, the better the accuracy of the calculations will be since fuel reduction will be reflected in the gross weight more often. However, a high resolution will increment computation time since more interpolations and data fetching will be required.

Interpolations in weight and ISA Temperature Deviation such as the ones described in Figure 3.2 are performed in this stage.

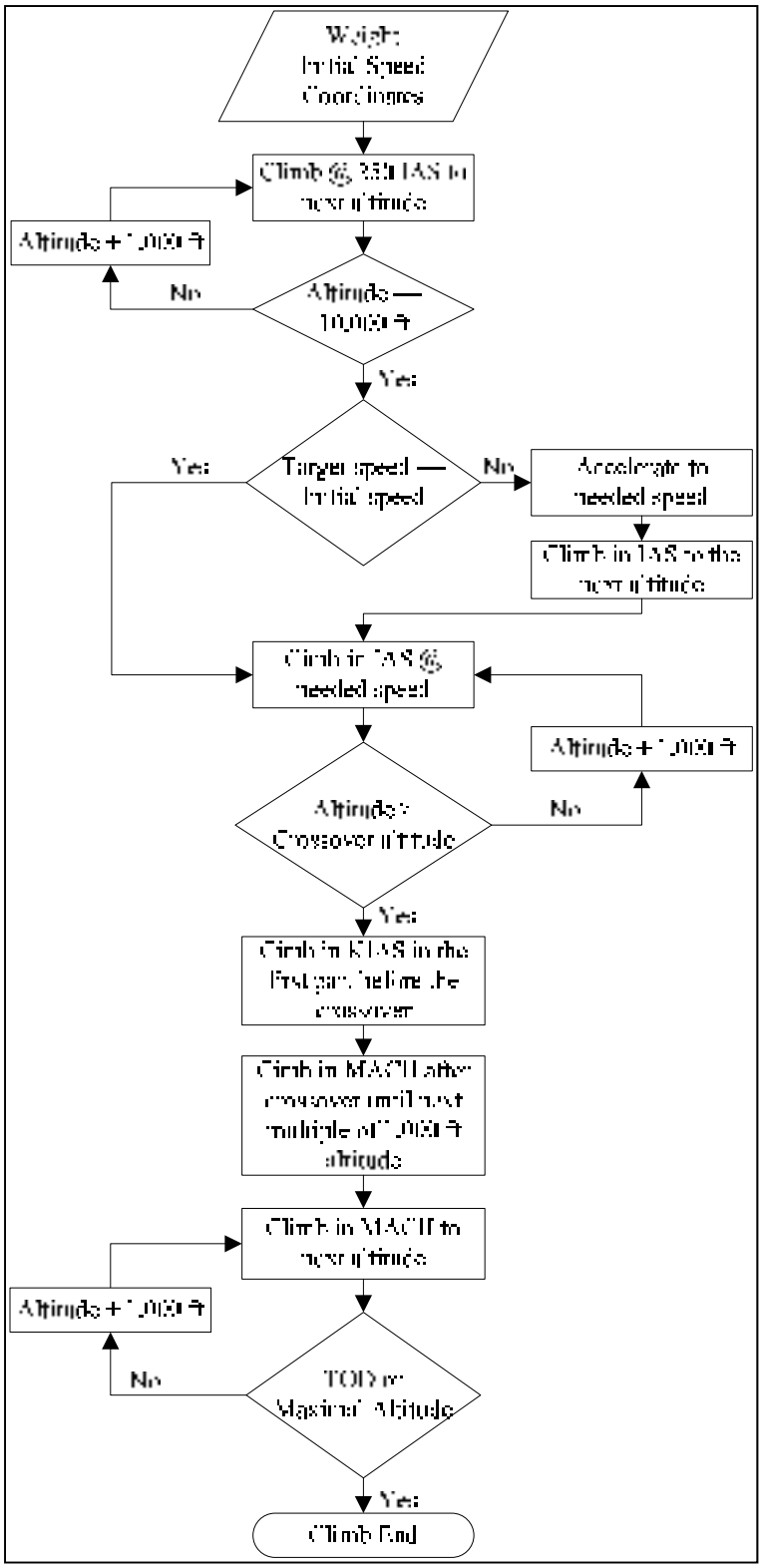

Figure 3.5 Climb Calculations Procedure

3.2.4.5.1 Step Climb

"Step climb" is the change of cruise altitude to a higher one to try to emulate an ideal constant climb cruise (Ojha, 1995), which is desirable due to the weight reduction due to the constant burn of fuel. This change in altitude was studied by Lovegren (Lovegren, 2011) and proved to save fuel specially in long haul flights. A study of the importance of the step-climb and the improvement opportunities in the United States was performed by Jensen *et al.* in (Luke Jensen et al., 2014). The climb step is normally executed in 2,000 ft steps. This is to respect the normal 1,000 ft separation between airways where aircraft flight at opposing heading. However, depending on Air Traffic Management (ATM), steps of different heights could be executed.

This change in altitude requires calculating the climb cost. Thus, the step climb cost is calculated using the "Climb Mach" database at the desired geographical point. The first waypoint at the new altitude is calculated with the climb horizontal distance obtained from the PDB. Once the desired altitude has been reached, the computations are continued using the "Cruise Mach" sub-database. Step climb procedure can be executed as many times as required or until the maximal altitude is reached. In this method, steps can be executed to the altitudes available in the PDB, which normally are 1,000 ft multiples. Figure 3.6 is a flowchart to follow to perform the required Cruise calculations with "step climbs".

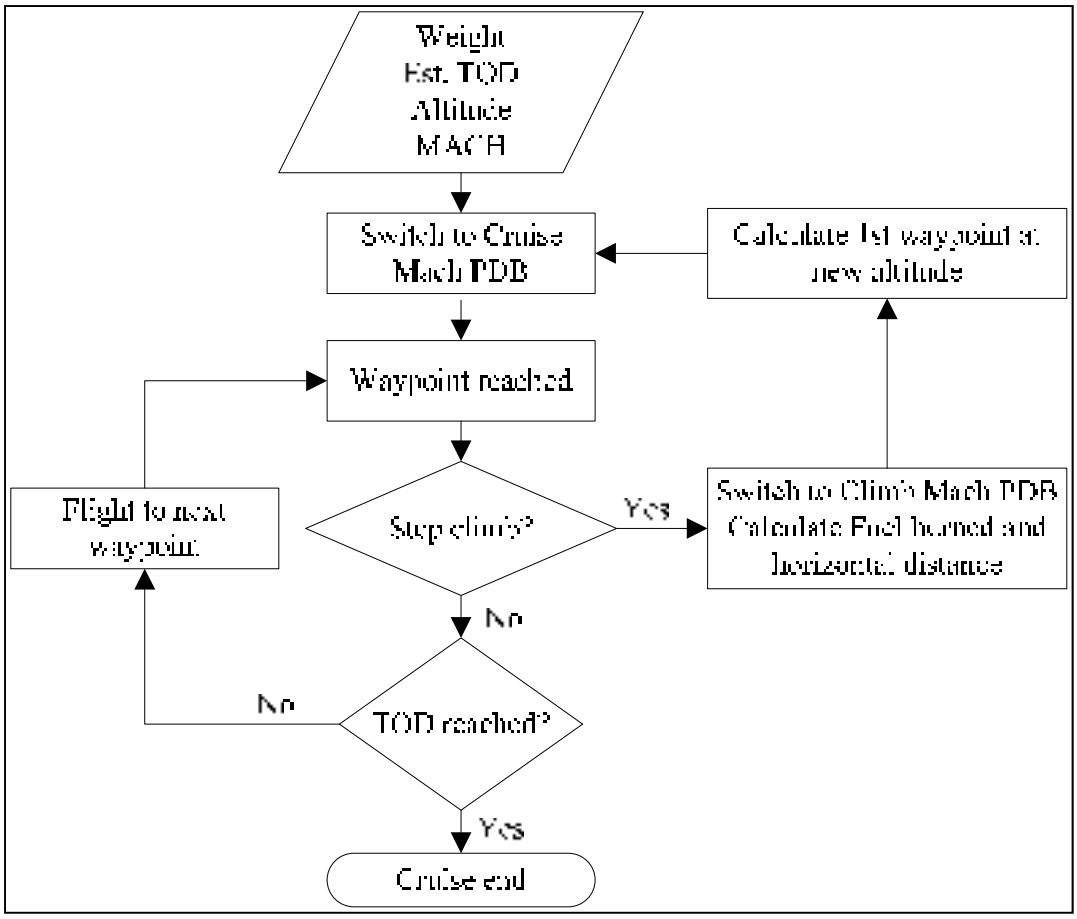

Figure 3.6 Cruise Calculations Procedure

3.2.4.6 Final Descent

Final Descent is assumed to be CDA. When the TOD has been reached, the final descent begins. This phase is highly dependent on cruise since the weight to be used is the one at the end of cruise and TOD location errors might exist. The last waypoint of cruise is not equidistant. It is the distance that is left from the (TOD - 1) waypoint to the estimated TOD.

Descent is similar to climb, but the order of the databases to be used and the order of calculations are performed backwards. First, the Mach descent is calculated until the crossover altitude is reached. Second, a descent in IAS is calculated followed by a deceleration. At 10,000 ft, per legislation, the final speed must to be 250 IAS. Finally, the

descent is executed at this constant speed of 250 IAS and ends at an altitude of 2,000 ft. The same considerations taken in climb are taken during descent phases. The final location of the aircraft is then compared to the final point of the trajectory. If the final position of the aircraft is located after the destination point, or if it is not located within a given distance before the destination point, the missing or surpassed distance is added or removed from the cruise phase, the TOD is redefined and the descent is recalculated. This process is repeated until the aircraft ends within the imposed limits. For this paper the imposed limits are of +/- 0.3 nautical miles.

Figure 3.7 is a description of the final descent procedure and the coupling with the cruise phase. The flight trajectory procedure ends here. At this point, fuel burned and flight time were calculated for each section and accumulated in the fuel and the time variable in Eq. (3.1).

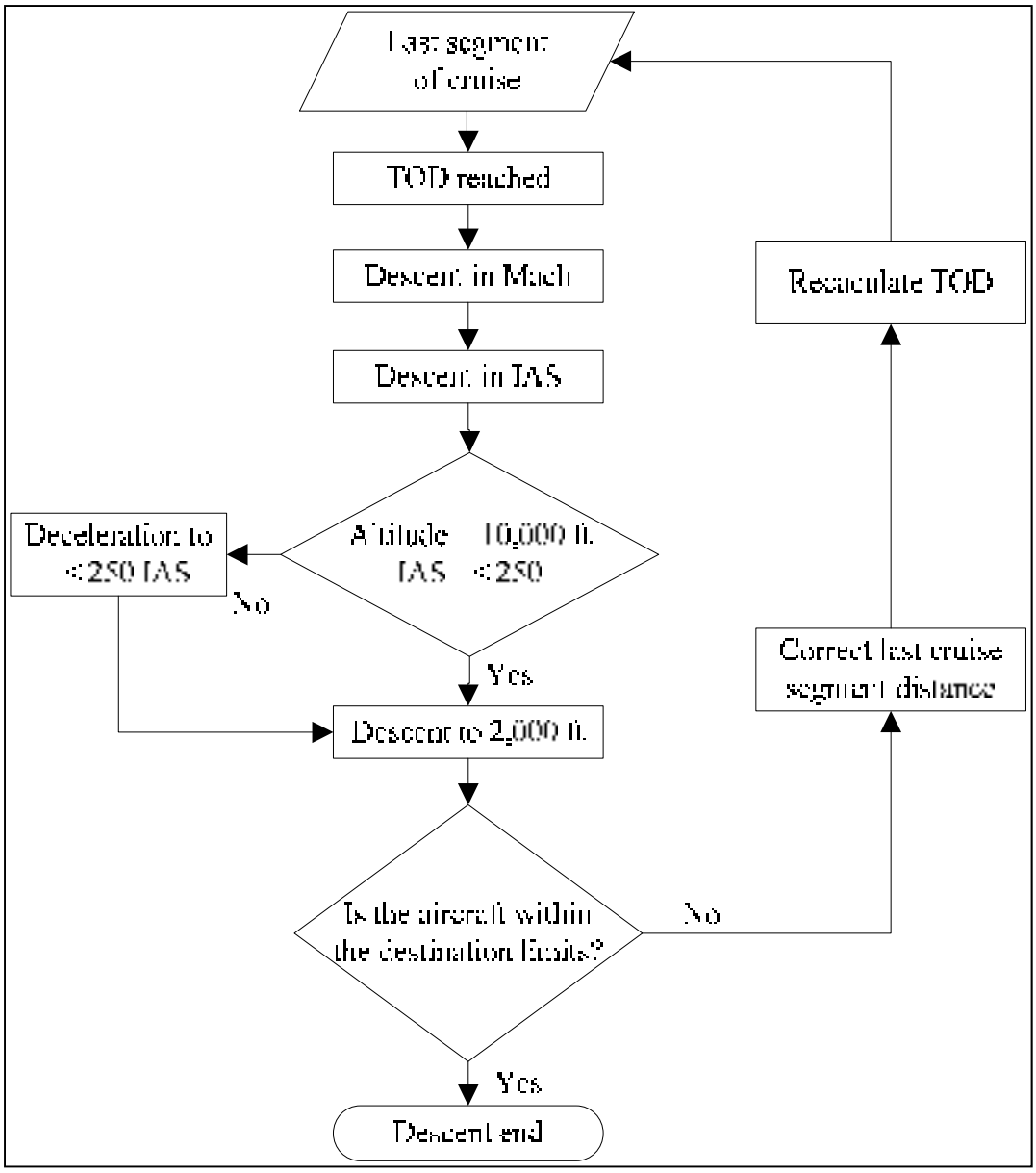

Figure 3.7 Descent Calculation Procedure

3.3 Results

3.3.1 Flight Comparison

The parameters from flight trajectory cost to be evaluated are flight time and fuel consumption because they are the variables that define Eq. (3.1). The validation consisted in calculating the cost of a giving set of trajectories with our method, then flying the same trajectories with either: a commercial flight simulator or the benchmark of a commercial FMS. The method results and the testing platforms results were then compared.

Three different aircraft PDBs were implemented in using the method described in this paper. These aircraft are referred as aircraft A, B and C. Aircraft A is a 3 engine commercial long haul aircraft. Aircraft B is a 2 engine commercial long haul aircraft. Aircraft C is a 2 engine regional jet. Tests for Aircraft A were performed comparing the results of the method developed here against a full dynamical model simulated with the software FlightSIM® from Presagis®. The FlightSIM® model was provided by our industrial partner who constructed and validated the model based on actual flight data, including fuel burn. Aircraft B, and Aircraft C results were validated with the Part Task Trainer (PTT), a simulator of the commercial FMS of reference provided as well by our industrial partner. The PTTs use the same PDBs as our algorithm, which was created from flight test data, to calculate fuel burned with a commercial algorithm.

General characteristics of the evaluated aircraft are shown in Table 3.3. TO refers to Take-off. Comparison between platforms is shown in
Table 3.4.

Table 3.3 Aircraft Characteristics

Aircraft	Min TO Weight (kg)	Max TO Weight (kg)	Min Speed (IAS/Mach)	Max Speed (IAS/Mach)
A	125,000	245,000	250/0.78	365/0.84
B	80,000	160,000	240/0.6	335/0.82
C	27,000	47,000	240/0.5	310/0.8

Table 3.4 Platforms Comparisons for Validation

Platform/ Aircraft	Algorithm	FlightSIM®	PTT/FMS
A	•	•	
B	•		•
C	•		•

3.3.1.1 Aircraft A: FlightSIM® and the Algorithm

Figure 3.8 and Figure 3.9 summarize the results for Aircraft A. The first eight flights were Montreal (YUL) to Toronto (YYZ) with a distance of 272 nm. Flights nine and ten were from Los Angeles (LAX) to Minneapolis (MSP) with a distance of 1324 nm, and the last flight was YUL to Vancouver (YVR) with a distance of 1993 nm. Different gross weight and speed were considered among flights. Tests with FlightSIM® are simulations in real time of the aircraft flights therefore, from simulation time point of view, each FlightSIM® test takes the same exact time as a real flight would take.

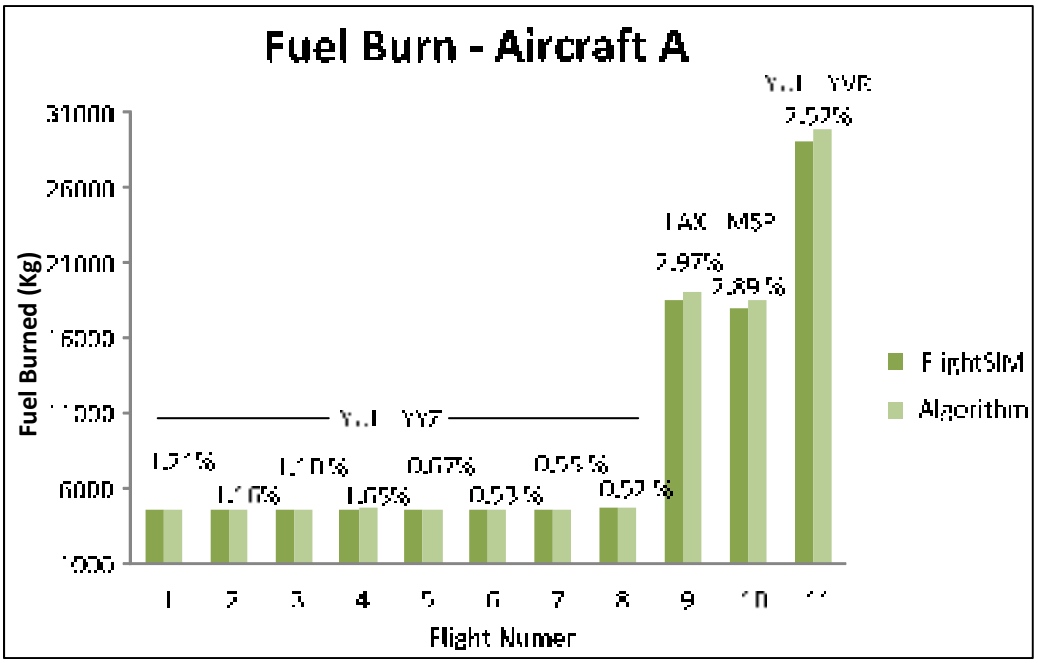

Figure 3.8 Aircraft A Calculated Fuel Burned

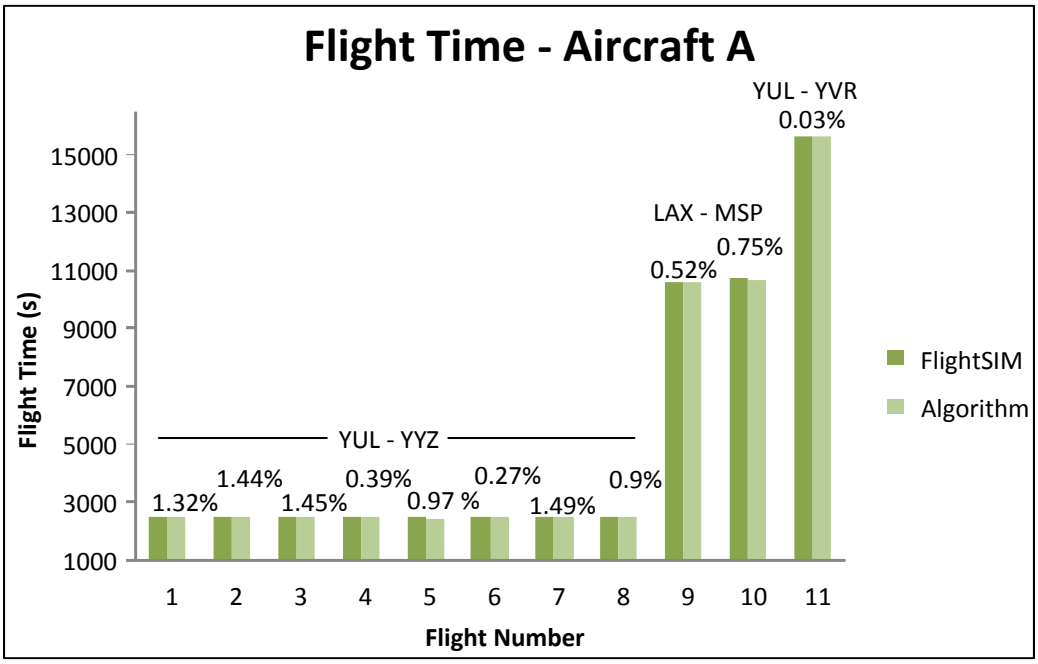

Figure 3.9 Aircraft A Calculated Flight Time

Figure 3.8 and Figure 3.9 for Aircraft A serve to validate the results obtained with the method explained in this paper. Figure 8 shows that, for all cases, the results of the method

calculated more fuel than FlightSIM®. The differences are attributed mainly to the fact that the algorithm does not instantaneously update fuel burned from the gross weight, and for the differences between the PDB and the aerodynamic model in FlightSIM®. This can be seen in flights 9-11, which present higher percentage difference than the first eight flights. These flights have the similarity that they have longer cruise phases. For these tests, the total weight was upgraded every 25 nm. Reducing these distances would reduce the fuel burned differences in Figure 3.8.

Figure 3.9 shows the differences in flight time are somewhat constant for all flights. This is because flight speed is kept constant regardless of weight. Thus no error is induced by the executed interpolations. These results indicate that the speed equations and considerations taken along the method allow accurate time calculations.

3.3.1.2 Aircraft B and C: PTT and the Algorithm

This method is intended to be used for any aircraft with a PDB. To verify this, the algorithm was implemented for two other aircraft. Figure 3.10 and Figure 3.11 are the results for Aircraft B. The first six flights are from YUL to YYZ and the last two are from YUL to YVR. The difference was obtained letting the results of the PTT as reference.

In Figure 3.10, it can be seen that the fuel burned has a similar difference percentage in all flights. This difference average is of only 0.72%, where normally the algorithm results show less fuel consumption than the PTT. As explained before, improving the distance of the waypoints would improve the fuel consumption estimation. Flight Time in Figure 3.11 shows a very little difference which can be neglected.

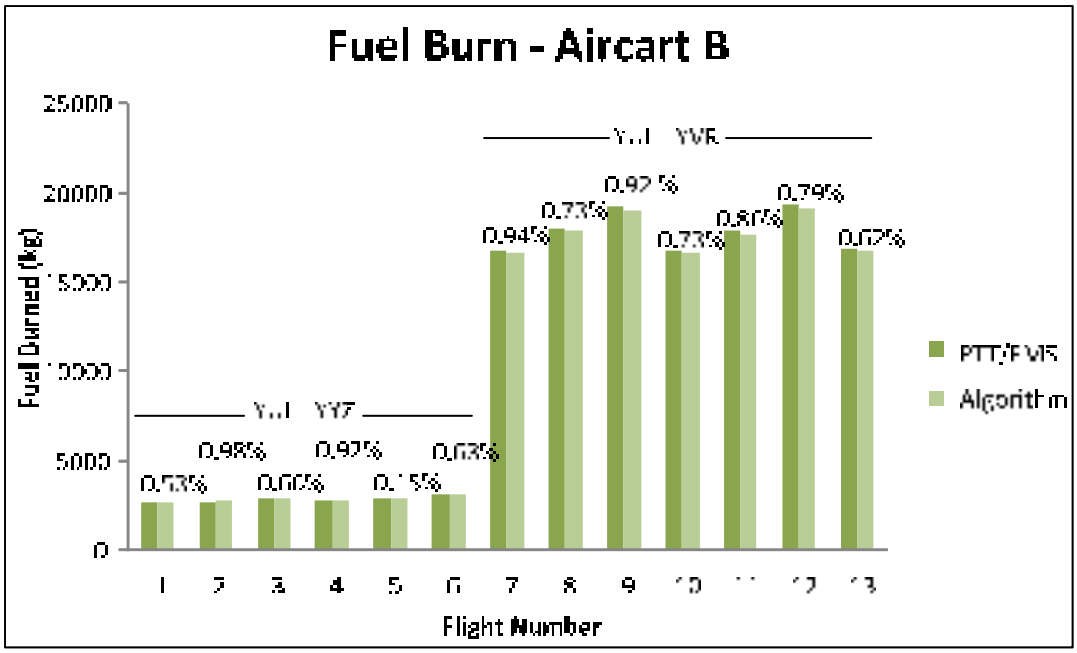

Figure 3.10 Aircraft B Calculated Fuel Burned

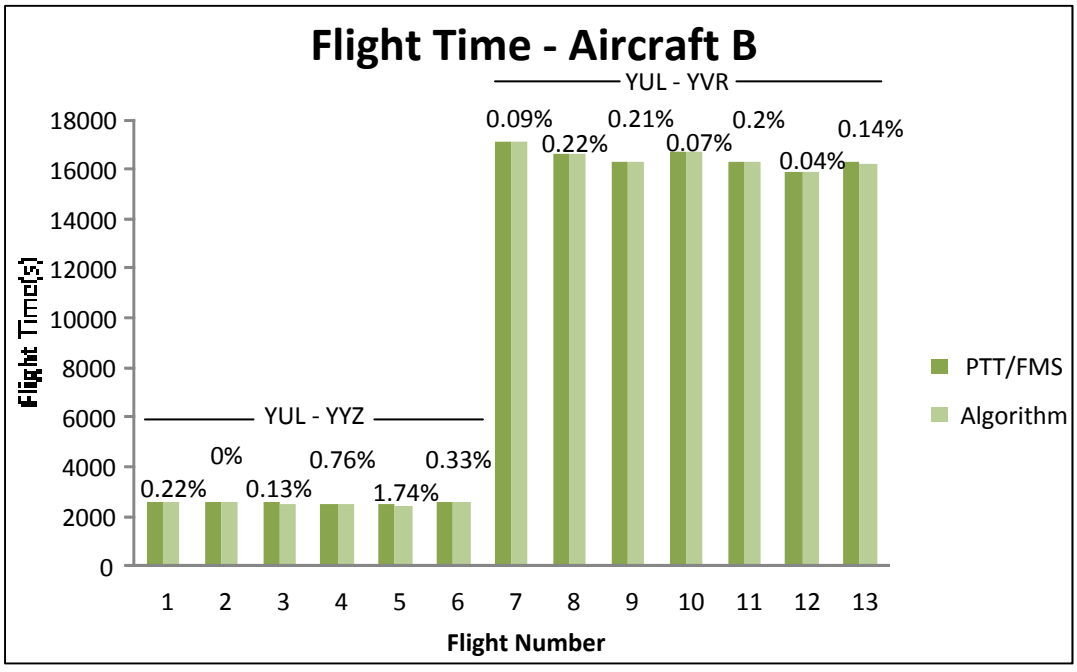

Figure 3.11 Aircraft B Calculated Flight Time

Figure 3.12 and Figure 3.13 are the results for Aircraft C. The first two flights are from YUL to Chicago (ORD) with a distance of 651 nm, flights three to five were from LAX to MSP,

flights six and seven were from YUL to Dallas Forth – Worth (DFW) with a distance of 1317 nm, flight 8 is from YUL to La Havana (HAV) with a distance of 1416 nm, flight 10 was from YUL to Cancun (CUN) with a distance of 1605 nm, and the last two were from YUL to YVR. As with aircraft B, the difference was obtained letting the results of the PTT as reference.

The results in Figure 3.12, are similar as those of Aircraft B, again with the algorithm calculating less fuel than the PTT. This difference average for Aircraft C was of 1.62%. Flight Time in Figure 3.13 suggests that the difference in flight time can be neglected.

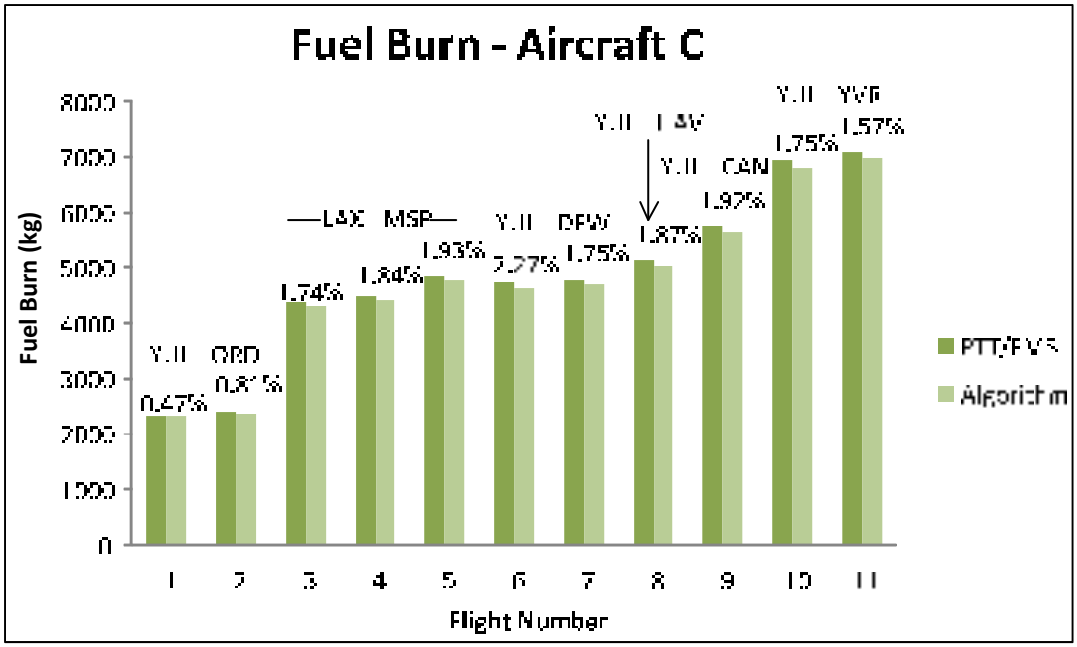

Figure 3.12 Aircraft C Calculated Fuel Burned

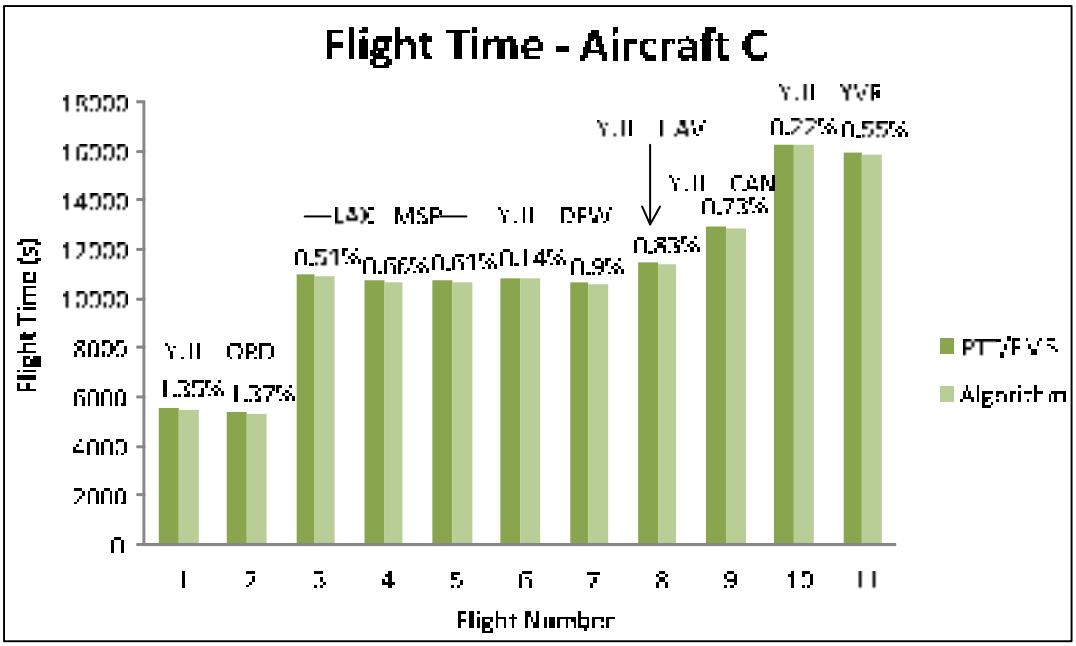

Figure 3.13 Aircraft C Calculated Flight Time

3.3.2 Cost Index Effect

For the flights evaluated above, the CI was always equal to 0. Varying the CI in the same flight would assign a cost to flight time as indicated in Eq (3.1) without modifying fuel burn. From Figure 3.9, flight 11 from Aircraft A was taken and the flight time was multiplied by different cost index values. The higher the cost index, the higher the time cost. This is of interest for optimization algorithms where time represents an important constraint and a compromise between fuel burned and flight time needs to be achieved to optimize the global cost. It can also be observed that the percentage of difference between the algorithm and FlightSIM® diminishes as the CI is higher. This is expected as according to Figure 3.9, flight time calculations difference is lower, and since time becomes a higher part of the total cost, the difference diminishes.

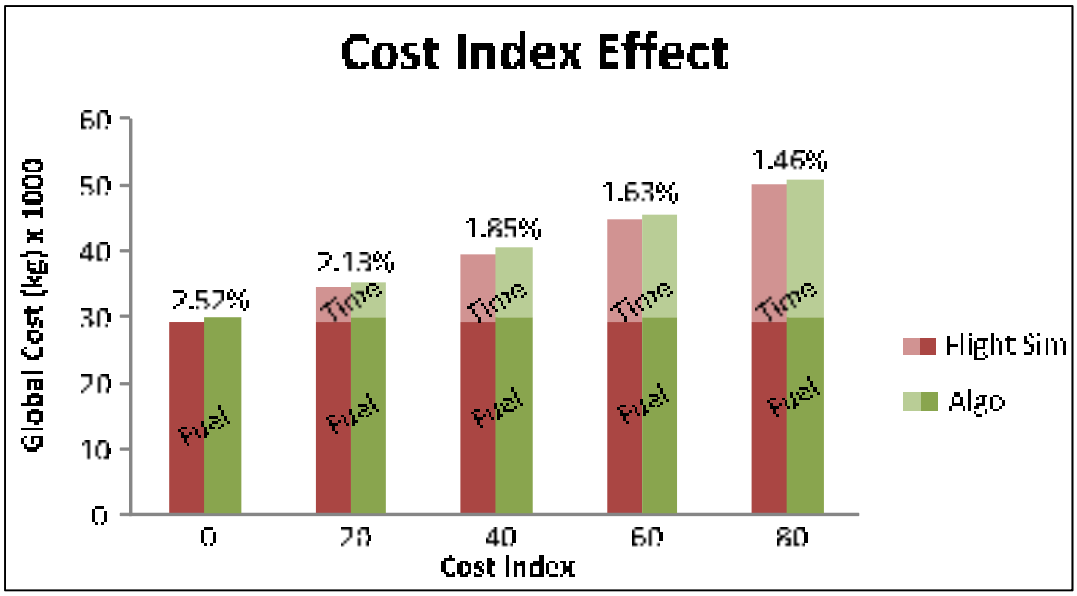

Figure 3.14 Cost Variation by Cost Index

3.3.3 Computation Time For Different Trajectories

In order to observe the algorithm execution time, different reference trajectories were flown and their execution times were saved and are shown in Figure 3.5.

Table 3.5 Algorithm execution time

	YUL YYZ	YUL DET	YUL WPG	YUL REG	YUL EDM	YUL VAN	YUL LAX	YUL PAR
Execution Time (s)	1.29	1.32	1.35	1.38	1.42	1.52	1.6	2.07
Distance (nm)	273	460	978	1272	1607	2013	2149	2993

The computation time increments as the flight distance is longer. This is expected since as the cruise is longer, more interpolations in the cruise phase are required, thus more computations. The execution time provides reliable computation at a lower execution time than FlightSIM®, which require the real flight time. These tests were performed using Matlab® in an AMD Phenom® B93 processor at 2.8Ghz.

3.3.4 Cruise Aircraft Distance Between Weight Update Points

The length of segments in cruise or, in other words the distance between the points where the fuel burned is reduced from the aircraft total weight, has a strong influence on the method computation accuracy. Figure 3.15 shows the cost variance and the calculation time for a flight from YUL to YVR. The length of the cruise segments was changed from 5 nm to 500 nm, the fuel burned and the execution time were analyzed.

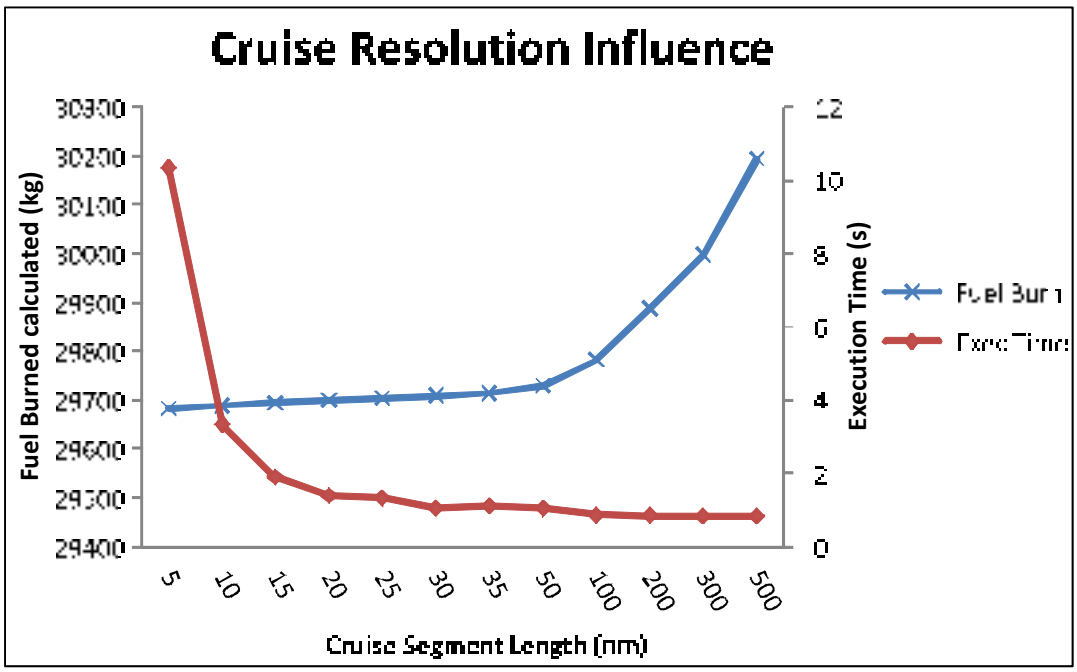

Figure 3.15 Cruise Segment Size Influence

From Figure 3.15, it can be observed that when reducing the fuel burned from the total gross weight by small cruise segments, the fuel burned is at its lowest and it becomes higher for the same flight as the cruise segment when the fuel burned is reduced from the gross weight increments. This is because at 5 nm the total weight of the aircraft is updated more often, and the interpolations performed to calculate the flight cost have a more accurate weight than those of 500 nm. However, having so small cruise segments will require more interpolations making the execution time higher. This can represent a problem for devices with small computation power such as the FMS, since for flight trajectory optimization many different

trajectories need to be calculated. From this same figure, it can be seen that separation between 5 nm and 50 nm maintains the fuel burned difference practically negligible. However, taking computation time into account, any value between 20 nm and 50 nm can be considered a good compromise between fuel burn accuracy and execution time.

3.4 Conclusion

A method to calculate the VNAV cost considering all stages of a commercial flight using a Performance Database and typical aircraft speed equations were implemented and validated. Lagrange linear interpolations were used to calculate values not available in the PDBs. Particularities of each stage of flight such as the crossover altitude; acceleration and "step climbs" were considered. The values of interest to calculate were flight time and fuel consumption since those are the ones that define a typical VNAV flight cost function.

By comparing the reference trajectories costs results of the novel algorithm developed in Matlab®, against the costs for the same trajectories provided by two different commercial platforms of reference: FlightSIM®, and the PTT, have shown that the method provides an accurate flight trajectories costs. The method results for Aircraft A were compared against the results of a dynamic model developed in FlightSIM® to shown that the accuracy of fuel burned and flight time calculations are reliable. This was also confirmed by comparing results against the calculation generated by Aircraft B and Aircraft C by a commercial PTT/FMS. This demonstrates that this method can be implemented in any experimental data based model such as the PDB.

This method may be directly implemented in a FMS (or used in ground simulations) as base of VNAV trajectory calculation for optimization algorithms. Using a PDB to calculate needed parameters has the advantage of not having to be concerned about aerodynamic parameters such as lift, drag, etc. It was observed that the CI changes the global costs by giving time a cost without affecting fuel consumption. This is of use when calculating optimal trajectories and time represents a constraint.

A study was performed to analyze the influence of the length of the cruise segments where the total weight is updated in order to observe its effect in the method calculation's accuracy. It was observed that the shorter the length of these segments the more accurate the method becomes, but the longer its execution time.

As future work it would be of interest to adapt this method to consider weather information, such as wind and temperature in the calculations. Wind affects aircraft speed, and temperature affects engines performance. This implementation would make this method more adequate for the computation of optimal trajectories. It would be as well of interest to compare this method against the methods commercially used to improve its execution time and its accuracy.

CHAPTER 4

NEW REFERENCE TRAJECTORY OPTIMIZATION ALGORITHM FOR A FLIGHT MANAGEMENT SYSTEM INSIPIRED FROM BEAM SEARCH

Alejandro Murrieta-Mendoza, Bruce Beuze, Laurane Ternisien, Ruxandra Mihaela Botez

École de Technologie Supérieure / Université du Québec, Montreal, Canada
Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticty

This article was accepted for publication in the Chinese Journal of Aeronautics and it will be published in August 2017

Résumé

Dans le but de réduire le coût du vol et la quantité d'émissions polluantes rejetées dans l'atmosphère, un nouvel algorithme d'optimisation est présenté pour la trajectoire verticale de vol prenant en compte les phases de montée, de croisière et de descente. La sélection des vitesses verticales de référence et des altitudes a été effectuée avec la résolution d'un problème combinatoire discret au moyen d'un arbre graphique traversant des nœuds en utilisant la technique d'optimisation de la recherche du faisceau. Pour obtenir un compromis entre le temps d'exécution et la capacité de l'algorithme à trouver la solution optimale, une méthode heuristique introduisant un paramètre appelé "coefficient d'optimisme" a été utilisé pour estimer le coût du vol de la trajectoire à chaque nœud. Le coût optimal de la trajectoire obtenu avec l'algorithme développé a été comparé au coût de la trajectoire optimale fournie par un système de gestion de vol commercial. La solution globale optimale a été validée en comparant les résultats avec un algorithme de recherche exhaustive, autre que l'algorithme proposé. L'algorithme développé prend en compte les effets météorologiques, la montée par paliers pendant la croisière, et les contraintes de gestion du trafic aérien tels que des segments d'altitude constante, la croisière constante Mach et une voie de navigation latérale de référence prédéfinie. La consommation de carburant de l'avion a été calculée en utilisant un modèle de performance numérique qui a été créé et validé en utilisant des données expérimentales d'essais en vol.

Abstract

With the objective of reducing the flight cost and the amount of polluting emissions released in the atmosphere, a new optimization algorithm considering the climb, cruise and descent phases is presented for the reference vertical flight trajectory. The selection of the reference vertical navigation speeds and altitudes was solved as a discrete combinatory problem by means of a graph-tree passing through nodes using the beam search optimization technique. To achieve a compromise between the execution time and the algorithm's ability to find the global optimal solution, a heuristic methodology introducing a parameter called "optimism coefficient was used in order to estimate the trajectory's flight cost at every node. The optimal trajectory cost obtained with the developed algorithm was compared with the cost of the optimal trajectory provided by a commercial Flight Management System. The global optimal solution was validated against an Exhaustive Search Algorithm, other than the proposed algorithm. The developed algorithm takes into account weather effects, step climbs during cruise and air traffic management constraints such as constant altitude segments, constant cruise Mach, and a pre-defined reference lateral navigation route. The aircraft fuel burn was computed using a numerical performance model which was created and validated using flight test experimental data.

4.1 Introduction

Several studies by aeronautical authorities and associations have estimated that the number of aircraft in service will increase in the forthcoming years (IATA, 2011). Having more aircraft airborne will bring new challenges to air traffic management to guarantee safe and efficient flights.

Worldwide, several programs have been initiated for the research, development and implementation of systems, regulations and procedures for future air traffic management. These programs include the Next Generation Air Transport System (NextGen) in North America, led by the United States, the Single European Sky ATM Research (SESAR) in Europe, and the Collaborative Actions for Renovation of Air Traffic Systems (CARATS) in

Japan. These programs improve Air Traffic Management to guarantee safety, expand air space capacity, reduce polluting emissions, and to provide flight trajectories as efficient as possible.

Efficient flight reference trajectories result in both flight time and fuel burn reductions. Flying these trajectories is extremely desirable, since fuel burn reduction directly diminishes polluting emissions, as well as saved on costs. As pointed out by the Air Transport Action Group (ATAG), around 2% of the carbon dioxide (CO2) emissions released to the atmosphere are caused by aeronautical activities (ATAG, 2014). The industry has set itself the goal of reducing its CO2 emissions to 50% of those generated in 2005 by 2050 (ICAO, 2010). Many organizations have been created to address and solve the pollution problems. One such example is the Business-led Network of Centers of Excellence (GARDN) in Canada, which encourages services and products to reduce polluting emissions.

Several new technologies and aircraft modifications have already been considered with the aim to reduce polluting emissions in which winglets have been proven to improve aircraft efficiency (Freitag, 2009). What's more, changes in materials and avionics equipment have reduced aircraft weight, as well as the introduction of biofuels (ATAG, 2009), and engine improvements have also reduced fuel burn. Airlines can achieve fuel savings with their current fleets by implementing different operational measures, such as the single engine taxiing, ground power units, engine washing, and flight reference trajectory optimization as presented in (McConnachie et al., 2013). Therefore, implementing these improvements in current fleets is becoming a necessity, since the newer aircraft generations alone will not be enough to fulfill the industry's polluting reduction goals (Randt P., Jessberger, & Ploetner, 2015).

The Continuous Descent Approach (CDA), an operational procedure related to flight reference trajectory optimization, has been proven to reduce fuel consumption. With this procedure, the aircraft descends with the engines in the IDLE mode (least fuel consuming setting) instead of descending in the conventional step descent pattern. Several airports have

implemented this technique because it contributes to fuel savings and noise reduction (Cao, Kotegawa, & Post, 2011; Clarke et al., 2013; Kwok-On et al., 2003; Kwok-On, Daniel, & Anthony, 2006). The CDA must be implemented correctly to avoid the need to execute a missed approach procedure, which is expensive in terms of fuel burn and pollution emission release(R. Dancila, Botez, & Ford, 2014; Murrieta-Mendoza et al., 2014). Other approaches, such as air to air refueling was studied in (Nangia, 2006) and aircraft ground movement optimization was suggested in (P. Godbole, J., A. G. Ranade, & R. S. Pant, 2014).

For "en-route" operations, Jensen et al. in (Luke Jensen et al., 2014; Luke Jensen et al., 2013) stated that aircrafts in the United States do not fly at their optimal speed or altitude. Other discussions have been conducted regarding the savings that flying at low cruise speeds may bring, as well as at how lower cruise speeds would affect other aircraft flying near the low-speed cruise aircraft (Bonnefoy & Hansman, 2010; Filippone, 2007). Turgut et al. (Turgut et al., 2014) developed equations to obtain the fuel flow for different aircrafts and found out that flight trajectories for national flight within Turkey can be improved.

Current reference trajectory optimization was done by ground teams or by airborne avionics such as the commercial Flight Management System (FMS). The former can use algorithms that require many computational resources to find the optimal reference trajectory; the latter require fast algorithms to find an optimal or a sub-optimal reference trajectory.

The reference trajectory could be studied in two different dimensions: the vertical dimension and the lateral dimension. The first consists in the speeds and altitudes that provide the most economical flights, and the latter consists in the geographical waypoints where no obstacles are present, and where the aircraft can take advantage of weather, such as tail winds, to reduce the flight cost.

A number of optimization algorithms have been implemented to solve the reference trajectory optimization problems. Different optimal control techniques considered various constraints and optimization objectives (Franco & Rivas, 2011; Franco & Rivas, 2015; Guo

& Zhu, 2013; Houacine & Khardi, 2010; McEnteggart & Whidborne, 2012; Pargett & Ardema, 2007; Valenzuela & Rivas, 2014). Equations of Motion (EoM) were used in these optimal control based algorithms in order to find the optimal solution. Other authors applied the Dynamic Programming; such as Miyazawa et al. (Miyazawa et al., 2013) who developed a moving search space algorithm by taking into account both weather and the Required Time of Arrival (RTA) constraints. Ng et al. also used Dynamic Programming, by taking into account winds for a fixed Mach number and by allowing changes in the flight level during the cruise regime (Ng et al., 2014). Hagelauer and Mora-Camino utilized the EoM to implement dynamic programming with neural networks to optimize trajectories, underscoring the fact that FMSs use numerical performance databases instead of EoM (Hagelauer & Mora-Camino, 1998).

Reference trajectory optimization focusing on avoidance (obstacles or weather constraints) has also been investigated. Cobano et al. applied the Particle Swarm Optimization for a 4D (RTA constraint considered) trajectory (Cobano et al., 2013). Ripper et al. used graph search algorithms to optimize the route of a general aviation aircraft while avoiding obstacles (Rippel et al., 2005). The dynamic physical principles of a stream avoiding obstacles were adapted to the path trajectory optimization. This methodology allowed the avoidance of objects of different shapes (Wang, Lyu, Yao, Liang, & Liu, 2015). An algorithm able to optimize the flight trajectory of cooperative Unmanned Aerial Vehicles (UAV) able to avoid obstacles was developed using a combination of the central force optimization algorithm and genetic algorithms in (Chen et al., 2016).

All these optimization algorithms were able to optimize the flight reference trajectory by fulfilling their imposed constraints. However, solving the EoM to find the optimal trajectory in a system with limited processing power, such as an FMS, can be very much time consuming, leading its implementation to be impractical. Therefore, the FMS do not normally use these equations, and instead uses a set of look up tables with experimental data called a numerical performance model.

Numerical performance models provide fuel consumption information, and therefore researchers have developed fuel burn estimators that work with this type of models to compute the cost of a cruise segment (B. Dancila et al., 2013). These estimators were later used to optimize flights in cruise (B Dancila, Botez, & D, 2012). Félix *et al.* used genetic algorithms to find the optimal trajectory by taking advantage of winds (Roberto S. Félix-Patrón et al., 2014). Murrieta and Botez used the Dijkstra's algorithm to find the optimal trajectory by taking advantage of winds and temperatures (Murrieta-Mendoza & Botez, 2014a).

The vertical reference trajectory optimization problem applied on a numerical performance model for all flight phases can be treated as a combinatorial problem. The number of possible solutions (feasible or not) are defined by combinations of Indicated Air Speeds (IAS), Mach numbers and altitudes available in the numerical performance model. Although the number of combinations for evaluation available in the numerical performance model might be considered low compared against the number of combinations available in classical optimization problems (e.g. the Traveling Salesman Problem, where a salesmen at a given origin city should visit every available city at least once and he must return to the origin city, is a hard Nondeterministic Polynomial time (NP) problem. For a high number of cities, there is no practical optimization solution able to guarantee optimality), it should be taken into account that the FMS possesses low power computation capabilities. Just a small part of these computation capabilities are dedicated to the reference trajectory optimization, as other critical operations have higher priority (Collinson, 2011) such as to provide the current trajectory guidance, to monitor the aircraft flight envelope so as to guarantee a safe flight, to control automatically the engine thrust, and etcetera. In a real time environment, systems such as the FMS, should be able to provide solutions (either "optimal" or good "sub-optimal") in a short period.

To solve this optimization problem for an FMS, Gagné et al. developed a space reduction search algorithm (Jocelyn Gagné, Murrieta-Mendoza, Botez, & Labour, 2013). Murrieta and Botez developed an optimization algorithm by reducing the space search even further and

thus the execution time (Murrieta-Mendoza & Botez, 2014b). Félix et al. used a numerical performance database and the golden section search algorithm to find the climb speed schedule and the Top of Climb (ToC) altitude in order to optimize a short flight, and to implement step climbs in long haul flights (R.S. Felix Patron et al., 2013).

Coupling and optimizing the lateral and the vertical reference profiles has also been a promising research area. Murrieta proposed five different reference lateral trajectories to be evaluated using a previously computed optimal vertical reference trajectory (Murrieta-Mendoza, 2013). Félix *et al*. implemented genetic algorithms to optimize the vertical reference trajectory (Roberto Salvador Felix Patron et al., 2013). Felix *et al*. used genetic algorithms to simultaneously optimize the lateral and vertical reference trajectories for the cruise phase (Roberto S Félix-Patrón, Berrou, & Botez, 2014; Roberto S. Félix-Patrón & Botez, 2014). Genetic algorithms provided good solutions, however their metaheuristic nature made them difficult to certify, and thus implemented in current airborne avionics.

In this chapter, the vertical reference trajectory optimization problem is solved with the Beam Search Algorithm. The implementation of this algorithm to the reference trajectory problem has not yet been studied in the literature. Beam search is a deterministic algorithm which requires low memory (the decision tree is reduced). This algorithm can quickly provide good solutions, even if they are not the most economical (optimal for this paper) ones. All these characteristics are desirable for the FMS.

The algorithm proposed in this paper, compared with the above mentioned algorithms used for the vertical reference trajectory optimization by using numerical performance models, brings the novelty of treating this problem in a “decision tree”. The speeds and altitudes in the numerical performance model are discrete values. The Beam Search Algorithm, unexplored for this problem, has been adapted to the vertical trajectory problem to explore its optimization capabilities. This paper also introduces a new heuristic (technique used to provide an approximate solution to a problem in a reasonable time) used by the algorithm to estimate the flight cost at any node within the decision tree. This heuristic takes into account

all flight phases, weather conditions, step climbs, and is designed to be used in a numerical performance model.

The algorithm developed in this paper takes into account winds, temperature and current traffic control constraints such as the constant speed and the constant flight level. Changes in the flight level (step climbs) during cruise are evaluated at a pre-defined time frame to mimic the cruise-climb regime. Since this algorithm was developed with the goal of being implemented in current and future FMS generations, a numerical performance model was used instead of the EoM.

The paper is organized as follows: Section 4.2.1 defines the performance database used to compute flight costs; Section 4.2.2 offers a brief explanation of how a flight cost is computed using a performance numerical model; The Beam Search Algorithm is briefly explained, and its implementation is fully explained in detail in Section 4.3.2, and Section 4.3.3. Section 4.5 presents and discusses the results; and the conclusions are provided in the last section.

4.2 Methodology

4.2.1 Numerical Performance Model

The numerical performance model is a set of tables containing an aircraft's performance in different flight phases. The available flight phases are provided in Table 4.1. This numerical performance model is developed and validated using experimental flight test data. It replaces the conventional Equations of Motion (EoM) widely used in the literature. This model is fairly expensive to be created with experimental flight data, and an alternative is to use a Level D research flight simulator (Murrieta-Mendoza, Demange, George, & Botez, 2015).
The inputs in the numerical performance model are provided as discrete values. For example, for the Indicated Air Speed (IAS) climb sub-database, the IAS input is provided in 10-knots separations, as shown in Eq. (4.1), where n is the last available IAS. The abbreviation ISA refers to International Standard Atmosphere.

$$IAS \in [190 \quad 200 \quad 210 \quad 220 \quad 230 \quad \ldots n] \in IAS_{PDB} \tag{4.1}$$

Table 4.1 Numerical Performance Model Sub-Databases

Sub-database	Inputs	Outputs
IAS Climb	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)
Climb Acceleration	Gross weight (kg) Initial IAS (knots) Acceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burn (kg) Horizontal traveled distance (nm) Altitude needed (ft)
Climb Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)
Cruise Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel flow (kg/hr)
Descent Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)

Table 4.1 Numerical Performance Model Sub-Databases (continue)

Sub-database	Inputs	Outputs
Deceleration	Gross weight (kg) Initial IAS (knots) Deceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burn (kg) Horizontal traveled distance (nm) Altitude needed (ft)
Descent IAS	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)

4.2.2 Flight Cost Computation

The flight cost computations for the vertical reference trajectory were performed using the method described in (Alejandro Murrieta-Mendoza & Ruxandra M. Botez, 2015). In this method, climb in IAS, climb in Mach, acceleration, cruise, descent in Mach, deceleration and descent in IAS are taken into account by considering the particularities of each phase, such as "crossover altitudes", "step-climbs", etc. A brief explanation of this method is given here for completeness.

By providing the input vector given by Eq. (4.2), along with a takeoff weight, a cost index and a reference lateral route, the method computes the fuel burn, the flight time and the flight cost. These parameters are accumulated in two different variables called *Fuel Burn* (kg) and *Flight Time* (h), respectively. Eq. (4.3) is used for computing the trajectory reference flight cost.

$$Vertical\ Profile = [IAS_{Climb} \quad Mach \quad Cruise_{Altitude} \quad IAS_{Descent}] \qquad (4.2)$$

$$Global\ Cost\ (kg) = \ Fuel\ Burned\ + Cost\ Index * Flight\ Time \quad (4.3)$$

where *Cost Index* (kg/hr) is a parameter that relates time to cost in fuel terms. The *Cost Index* is normally defined by airlines before take-off and considers crew, maintenance and other time-related costs. A high *Cost Index* value makes the aircraft fly at higher speeds in order to reduce *Flight Time* even if these speeds are not efficient in terms of fuel burn. This is because *Flight Time* has a strong influence on Eq. (4.3). A low cost index value diminishes the importance of *Flight Time* in *Global Cost*, thus the algorithm focuses mostly on reducing the fuel burn. The aircraft can fly at lower speeds if they are more efficient in terms of fuel burn.

The altitudes and speeds considered to compute the flight cost are only those available in the numerical performance model. However, the aircraft`s gross weight constantly changes as fuel is burned, and the ISA deviation temperature changes while the aircraft is in flight. Weight and ISA deviation temperature deviations can take values that are not exactly those available in the discrete numerical performance model input values.

For this reason, a series of linear Lagrange interpolations were executed using the numerical performance models limits to compute the fuel burned and the horizontal traveled distance. During climb and descent, interpolations were performed at altitudes multiples of 1000 ft. During the cruise regime, interpolations were executed every 25 nautical miles for the current aircraft weight and the local ISA deviation temperature. This method takes into account the step climb costs as well as weather conditions. The possibility of executing step climbs is evaluated every hour of flight. Weather information is obtained from forecasts provided by Environment Canada. A typical vertical flight profile is shown in Figure 4.1.

Throughout the paper, when this interpolations method is used, it is referred to as “real cost computation”. This method has the particularity that is time consuming, so it should be executed as few times as possible.

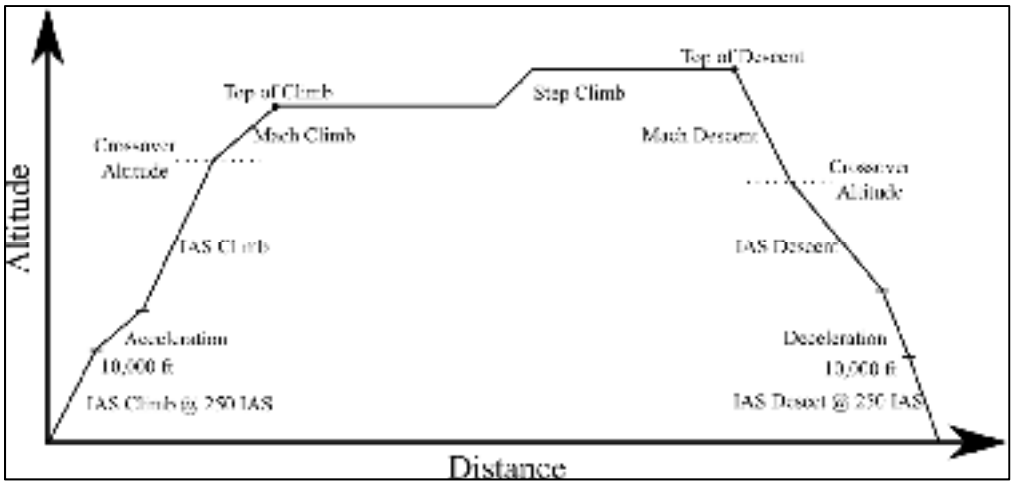

Figure 4.1 Typical reference vertical flight profile

4.3 The Optimization Algorithm

4.3.1 The Search Space: A Decision Graph

The decision variables for the vertical reference trajectory optimization problem are the IAS climb, the IAS descent, the Mach number and the cruise altitude. Since the decision variables can take only the pre-defined values in the numerical performance model, the problem becomes a "discrete combinatory" one. The combinations available to solve the vertical reference trajectory problem that fulfill Eq. (4.2) can be modeled by means of a "decision graph". By the International Civil Aviation Organization (ICAO) recommendation, aircraft cannot fly faster than 250 IAS below 10,000 ft. The speed restrictions along with the following conditions are imposed in order to build the "tree-graph:

- Flight begins at 2000 ft at an initial IAS of 240/250 kts;
- Acceleration can only be executed once at 10000 ft.;
- The Mach number for the cruise, climb and descent phases is the same, and is kept constant along the flight;
- The initial cruise altitude is the one at the ToC;

- Step climbs are not taken into account to build the decision graph, only the altitude at the ToC is considered. However, Step Climbs are taken into account for the optimal trajectory;
- Deceleration can only be executed once during descent;
- While descending, IAS must be at or below 250 kts at 10000 ft.;
- The flight trajectory always ends at 2000 ft at or below an IAS of 250 kts.

With these conditions, the decision tree-graph can be constructed, starting from defining the root node as 250 kts or 240 kts IAS at 2000 ft. The first tree level is composed of $Nodes_{ALT}$, the second tree level is composed of $Nodes_{Mach}$, the third tree level, of $Nodes_{IASCLB}$, and the last tree level is composed of $Nodes_{IASDES}$. These last nodes (the $Nodes_{IASDES}$) are called "leaves". The abbreviation "ALT" defines the Altitude, "CLB" defines the Climb Regime, and "DES" defines the Descent Regime.

The tree-graph order level was determined based on its influence on the flight cost. "Altitude" and "Mach" parameters are important for both short and long flights as explained next.

For short flights, "Altitude" determines how much the aircraft has to climb (a higher climb is normally more expensive). "Mach number" determines the crossover altitude (change from IAS to Mach), which has an influence on the climb cost.

For long flights, the cruise is the most expensive part of a flight. The parameters of the cruise phase are the Altitude and Mach number. For these reasons, Altitude and Mach number are the first two levels of the vertical reference tree. As it will be explained in Section 4.3, this order of the levels is important to reduce the total number of computations. A reduced graph-tree is shown in Figure 4.2 where *l* is the first available value and *n* is the maximal available value for that set.

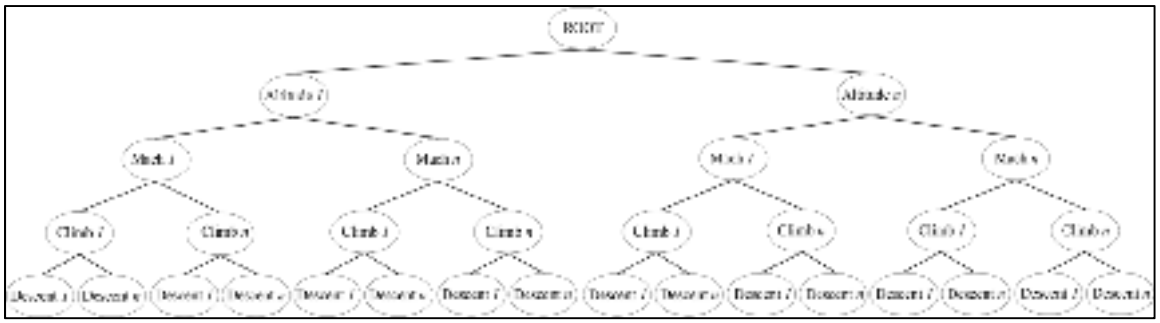

Figure 4.2 Vertical reference trajectory optimization graph-tree

4.3.2 Problem Definition

The vertical reference trajectory discrete optimization problem regards the selection of speeds and altitudes that minimize the flight cost as an integer linear programming problem:

$$Minimize: Global\ Cost\ (IAS_{climb}, Mach_{PDB}, Cruise_{Altitude}, IAS_{Descent}) \quad (4.4)$$

$$IAS_{climb} \in IAS_{CLIMB\ PDB} \quad (4.5)$$

$$Mach_{PDB} \in Mach_{PDB} \quad (4.6)$$

$$Cruise_{Altitude} \in Cruise_{Altitude\ PDB} \quad (4.7)$$

$$IAS_{Descent} \in IAS_{DESCENT\ PDB} \quad (4.8)$$

where variables given by Eqs. (4.5) to (4.8) have discrete finite values available in the numerical performance model (see Eq. (4.1) - (4.4)) and compose the vector expressed by Eq (4.2). The solution of this system is given in the Vertical$_{OPTIMAL}$ vector shown in Eq (4.9), where "optimal" defines the combination given in Eq (4.4) that minimizes Eq (4.3).

$$Vertical_{OPTIMAL} = \left[Climb_{OptIAS} \quad Mach_{Opt} \quad Altitude_{Opt} \quad Descend_{OptIAS}\right] \tag{4.9}$$

The "design discrete variables" given in Eqs. (4.5) to (4.8) are modeled, and form a decision tree as shown in where each "node" is a "decision variable". Depending on the decision, the "decision variable" (e.g. a given "IAS node") is taken (or not) into account for the flight cost computation.

Using the "decision tree", the problem can conceptually be formulated as constrained programming as follows:

$$Minimize: \sum x_i \cdot Branch\ cost_i \tag{4.10}$$

$$i \in Total\ Nodes\ in\ the\ tree \tag{4.11}$$

$$x_i \in \{0, 1\} \tag{4.12}$$

$$\sum x_i(per\ tree\ level) = 1 \tag{4.13}$$

In Eq (4.10), *x* refers to the selection of nodes with a binary value of '1', *Branch cost* refers to the cost of branches connecting those nodes. Eq. (4.12) shows the binary value chosen for each node. Eq. (4.13) is a node constraint stating that for every level, only one node can have the binary value of '1'.

In other words, if a given node is selected as part of the optimal solution, it takes the binary value of '1'. If a node is not selected as part of the optimal solution, it takes the binary value of '0'. The cost between two levels is provided by the branches connecting the nodes with a

binary value of ‘1’. Only one node per tree level can take the value of ‘1’ to form the final solution. All the other nodes in a given level take the value of ‘0’, for example, for an optimal vertical reference trajectory solution such as the one given by Eq. (4.14), the tree takes the form shown in Figure 4.3.

$$VNAV_{OPTIMAL} = [260\ kts \quad 0.82\ Mach \quad 36{,}000\ ft \quad 280\ kts] \tag{4.14}$$

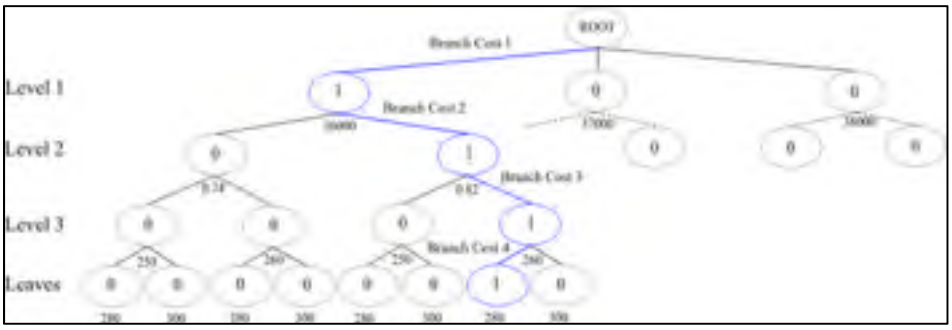

Figure 4.3 Decision tree example for given solution

Note that at Level 1, the altitude of 36,000 ft was selected as the optimal flight altitude, and thus it took a binary value of ‘1’; all the other altitudes take a value of ‘0’ since they did not compose the optimal solution. At Level 2, the Mach number 0.82 was selected as part of the optimal solution, and thus it took the value of ‘1’; all the other Mach nodes took a binary decision value of ‘0’ since they were not part of the optimal solution. Level 3 and Level 4 (leaves) behave in a similar way. The binary values defined the decision taken if a given node was or not used as part of the solution. It is the Beam Search Algorithm explained in the next section that makes these decisions.

The Beam Search Algorithm is an approximation of Branch and Bound (B&B) optimization technique. In both algorithms, some promising nodes are further explored, while the unpromising nodes and all their descendants are permanently pruned. The main difference between these two algorithms is that B&B does not allow pruning unpromising nodes in cases when it cannot guarantee that the node to be pruned do not contain the optimal

solution. Guaranteeing the optimality before pruning a node might require extensive computing time in the B&B algorithm. The Beam Search is more aggressive than the B&B to prune the nodes; a node can be pruned without guaranteeing the optimality by reducing the computing time (Sabuncuoglu & Bayiz, 1999).

The Beam Search Algorithm expands a pre-defined number of the most promising nodes per tree level, that number of nodes is usually smaller than the total number of nodes. The number of promising nodes per level to be developed is called *beam width* (*b*). Every tree generated (node expansion) from a promising node is called a *beam*. To determine if a node is promising or not, either a *local* evaluation (node's cost is just a projection) or a *global* evaluation (all nodes composing the solution are known, so that cost can be more exact) are executed. Local evaluations are quick, but normally discard good solutions. While, on the contrary, global evaluations are accurate, but normally require high computational time.

At the beginning, after the expansion and the local evaluation of the root node, the Beam Search Algorithm selects the *b* most efficient nodes at the first tree level. Neither promising nodes nor nodes outside b are pruned. For example, if there are ten nodes defining the first level, and $b = 6$, only the six most promising nodes are expanded, and the remaining nodes are pruned. This operation is repeated for the descendent nodes. Expanding a descendent node generates a new *beam*; thus the search is performed through different parallel beams. There are many different ways of exploring these parallel beams, either independently, or through a priority list as in the algorithm developed in this paper. New beams are generated and nodes are explored until all nodes have been either explored or pruned.

4.3.3 The Beam Search Algorithm

After defining the tree representation of the vertical reference trajectory optimization problem in Section 4.3.1, and the general theory of the Beam Search technique, the optimization methodology used for the algorithm described in this paper is defined in the following paragraphs.

Every node evaluated in the tree is associated with a flight cost estimation that is calculated using a heuristic. This heuristic is a *local evaluation of nodes*, which takes into account the whole vertical reference trajectory. This flight cost estimation is used to evaluate if a node is promising or not. Since this is a minimization problem, the lower the local evaluation cost is, the more promising the node becomes. If a node is considered as promising (the local evaluation cost is higher than the current best cost solution) and within the beam width (*b*), it is expanded. However, if a node is not promising, then it is permanently pruned from the tree together with all its descendants. In other words, it is pruned because it is not economical enough to be included in *b*, and/or because the most optimistic cost prediction from the combination of a node's descendants will not give a more economical solution than the current best solution. This particularity makes this algorithm desirable, as discarding nodes reduces the number of available combinations to compute at the trade-off of not guaranteeing optimality. This is because the optimal solution might be pruned due to the size of *b* or as the *local evaluation* fails to provide a good cost estimation. The objective is to find a good trajectory while pruning the highest number of nodes with the minimal number of leaves to visit/evaluate; thereby, the algorithm execution time and the required memory are reduced. However, care must be given to the not prune nodes that could lead to a potential optimal solution, which is the reason why attention should be given to the evaluation function (heuristic). This function is explained in details in Section 4.3.3.1.

In the beginning of the algorithm, the best current solution is defined as "infinite". The first real best solution is calculated the first time when a leaf (last level) is visited. At the last node level (the leaves), all the parameters (nodes) shown in Eq. (4.2) are known. When all the parameters in the solution are known, a complete solution has been found. At this point, a *global evaluation*, which takes the form of the "real cost computation" described in Section 4.2.2, is performed to provide an accurate flight cost. As stated before, this real cost computation method is time consuming; it is thus desirable to minimize the number of times that the method is executed. Therefore, once at the last level (leaf), instead of executing the real cost computation directly, a last *local evaluation* is computed. If the local evaluation

suggests that a more economical cost should be found in that last leaf, then the real flight cost is computed.

Reaching a leaf quickly is important because computing a "first solution" presents the opportunity of pruning nodes, which reduces the search space. However, it is desirable to give priority to the expansion of the most promising nodes in order to find the optimal solution faster. The faster the optimal solution is found, the more branches can be pruned. Within the developed algorithm, lists that rank the nodes by priority are used to keep track of the existing beams.

The algorithm procedure is as follows. In the first level, after the root node expansion, all the nodes' costs estimations are computed using the *local evaluation* function (heuristic). A first queue list ($List_{ALT}$) containing beam width (b_1) elements is sorted using the most promising nodes (most economical nodes first), the nodes that are not included in $List_{ALT}$ due to b_1 are permanently pruned. The first node from $List_{ALT}$ is selected. If this node is more economical than the current best cost, it is further expanded, otherwise it is pruned. For the second level, all descendant nodes' costs are estimated with the local evaluation function. A queue list ($List_{Mach}$) containing b_2 elements is created and sorted with the most economical node first. Nodes not included in $List_{Mach}$ due to b_2 are permanently pruned. The best node is selected. If this node is more economical than the current best cost, it is expanded, otherwise it is pruned. At the third level, after the node expansion, all nodes' costs are estimated with the local evaluation function. A third queue list ($List_{IASCLB}$) containing b_3 elements is created and sorted with the most economical node first. Once again, the first best node is selected. If the selected node is more economical than the current best cost, it is expanded to the last level: the leaves, otherwise, it is pruned. At the last level the first encountered node's cost is estimated with the *local evaluation* function. If the cost estimation delivered by the *local evaluation* function is higher than the current best solution's cost, the node is pruned. Otherwise, a *global evaluation* (time consuming method) is executed. When the *global evaluation* cost is more economical than the current best solution's cost, the combination of nodes that form this trajectory becomes the new best solution. Otherwise, this node is

discarded. The next node at the last level is evaluated following the same procedure as the preceding node. Once all descendant nodes are evaluated at this level, the algorithm comes back to the preceding level. The expanded node in $List_{IASCLB}$ is removed from the list. The next node in the list is selected. The *local evaluation* is executed. If this node's cost estimation is more economical than the current best cost, the node is expanded, otherwise it is pruned. This is repeated until the $List_{IASCLB}$ is empty. Then, the algorithm returns to the preceding level and the expanded node is removed from the $List_{Mach}$, where the next node in the list is selected. If the node's estimation with the *local evaluation* is more economical than the current best cost, the node is expanded, otherwise it is pruned. This is repeated until the $List_{IASCLB}$ is empty. Then, the algorithm comes back to the preceding level and the selected node is removed from $List_{ALT}$. The next node in the list is selected. If the node's cost estimation is more economical than the current best cost, the node is expanded, otherwise it is pruned. Once the $List_{ALT}$ is empty, the algorithm stops and delivers the optimal trajectory. So the next section will describe the local evaluation function (heuristics) and a pseudo code describing the algorithm will be shown in Section 4.3.3.6.

4.3.3.1 The Local Evaluation Function: Heuristics

The heuristic in the local evaluation is the key element of the Beam Search Algorithm. This function gives the flight cost estimation taking into account all the flight phases. The data in the numerical performance model allows performing this flight cost estimation. The key is to provide a good estimation that is optimistic enough to prevent the algorithm from pruning high-potential branches. "Optimistic" in this application means that the estimated cost is lower than the real flight cost.

Depending on a node's position within the tree (Figure 4.2), some parameters of Eq. (4.2) are known. Two cases exist, and they can be divided into many sub-cases. The first case is identified when only some parameters in Eq. (4.2) are known and the rest of the parameters are unknown. In this case, the local evaluation function identifies the known parameters and provides a cost projection by considering the unknown parameters for the whole flight. The

greater the numbers of parameters from Eq. (4.2) that are available for the local evaluation function to produce the estimation, the more accurate the estimation is. In other words, the estimation is more accurate the deeper it is executed in the tree. The second case is identified when all the input parameters in Eq. (4.2) are known, which allows the computation of the real flight cost (global evaluation/real computation module). In the first case (in which only some parameters are known), different sub-cases are studied as the function of these known parameters. The second case (in which all parameters are known) happens only at the last tree level (the "leaves").

The new estimation coefficient, called the "optimism coefficient", is used to control the optimism of the local evaluation, or how aggressively nodes can be pruned.

The local evaluation function becomes less optimistic as it gets closer to the leaves, where more solution values become known. Since the function is less optimistic, it tends to prune more nodes/branches, thereby reducing the calculation time. The local evaluation function's behavior at each level is explained next.

4.3.3.2 Local Evaluation at the First Node Level: Altitude

The altitude node value is known at the first node level. However, the Mach number, IAS climb, and IAS descent parameters in Eq. (4.2) are still unknown. Giving random values to the unknown parameters would likely result in an unrealistic estimation for the different flight trajectories that an aircraft could actually perform.

The optimism coefficient (C_{opt}) is introduced in the local evaluating function in order to influence the optimism level while calculating the projected cost. This coefficient can have any value between zero (0) to one (1). A $C_{opt} = 0$ corresponds to a "completely pessimistic" local evaluation function, and $C_{opt} = 1$ corresponds to a very "optimistic" local evaluation function. By taking this coefficient into account, the function takes the form in Eq. (4.15).

$$Cost_{EST} = Average - C_{opt} * (Average - NPM_{min}) \quad (4.15)$$

where $Cost_{EST}$ is the flight cost estimation (or bound), NPM_{min} is the minimal value obtained from the numerical performance model for the current flight phase, C_{opt} is the optimism coefficient, and *Average* is the mean between the maximal and minimal value found in the numerical performance model for the given flight condition by taking into account the known vertical flight profile parameters.

Flight time is computed using the ground speed and the flight distance. Since the Mach number at this level is unknown, the maximal Mach number available is used to calculate the flight time, as a high Mach number reduces flight time.

4.3.3.3 Local Evaluation at the Second and Third Levels: Mach & Climb IAS

Once at the second level, the Altitude and Mach parameters are known. Since the climb and descent Mach numbers are considered to be the same as the cruise Mach number, as formulated in the hypothesis in Section 4.3.1, a short climb and descent estimation can be considered. This short climb/descent cost should be "optimistic", since climb/descent in Mach comprises just a short phase of a flight; therefore only a 1,000 ft. descent/climb is considered. The flight cost estimation is computed using Eq. (4.15).

The evaluating function is able to consider the effect of step-climbs. To do so without enlarging the tree or increasing the algorithm's computation time, the maximum number of altitude changes the aircraft can perform for a given cruise altitude is calculated, allowing 2,000-ft step climbs as suggested in the literature (Ojha, 1995). For example, at a cruise altitude of 36000 ft (and a ceiling altitude of 40,000 ft), only two step climbs can be performed. The fuel flow for the first and the last altitude are taken from the numerical performance model. The fuel flow for the given Mach Cruise Altitude is then the mean of the fuel flow, as shown in Eq. (4.16).

$$Fuel\ Flow_{Cruise} = \frac{Fuel\ Flow\ (Mach, Initial\ Cruise\ Alt) + Fuel\ Flow\ (Mach, Last\ Cruise\ Alt)}{2} \quad (4.16)$$

where *Fuel Flow*Cruise is the fuel flow to be considered in the cruise regime, *Fuel Flow* (*Mach*, *Initial Cruise Alt*) is the fuel flow at the beginning of the cruise, and *Fuel Flow* (*Mach*, *Last Cruise Alt*) is the fuel flow at the highest altitude at which the aircraft can fly for the given cruise.

At the third level, the parameters Climb IAS, Mach and Altitude are known, and therefore the "crossover altitude" (altitude where the IAS reference speed is changed to Mach number) is known. Consequently, the climb phase can be estimated by making the local evaluation function more accurate.

4.3.3.4 Local Evaluation at the Fourth Level: Descent IAS

At the fourth and last level (the leaves), Eq. (4.15) takes a different C_{opt} value than in the other tree levels. This is because the descent phase has a small effect on the fuel computations and thus a lower C_{opt} is used to correctly estimate the bound. This coefficient is called C_{opt2}. To limit the need to execute the real cost computation method as much as possible, it is executed only if the resulting leaf bound is lower than the current best cost. This delimiting factor is a practical one and is considered because the evaluating function execution time is faster than the real cost computation method execution time.

4.3.3.5 Local Evaluation Function and Weather

When weather effects are considered, the real flight cost method considers weather changes at every point along the flight. The execution time therefore increases since the method fetches the weather at every point within the trajectory. Considering the weather implementation in the evaluating function in the same way it is considered in the real flight cost method would indeed make the evaluation function very time consuming.

We base our solution on the fact that the wind speed is normally higher (in the jet stream) at high altitudes over 30,000 ft. For the evaluating functions, starting at an altitude of 30,000 ft, a tailwind speed starting from 1.5 knots is incremented linearly to 6.5 knots at a maximal altitude of 40,000 ft. The temperature is considered to be the standard atmosphere (*ISA DEV* = 0) throughout. The ground speed increment due to the tailwind does not change the speed value in the input to the numerical performance model; the speed in the database is the IAS, rather than the ground speed used to compute the flight time.

This pseudo code is summarized under the form of a flowchart in Figure 4.4.

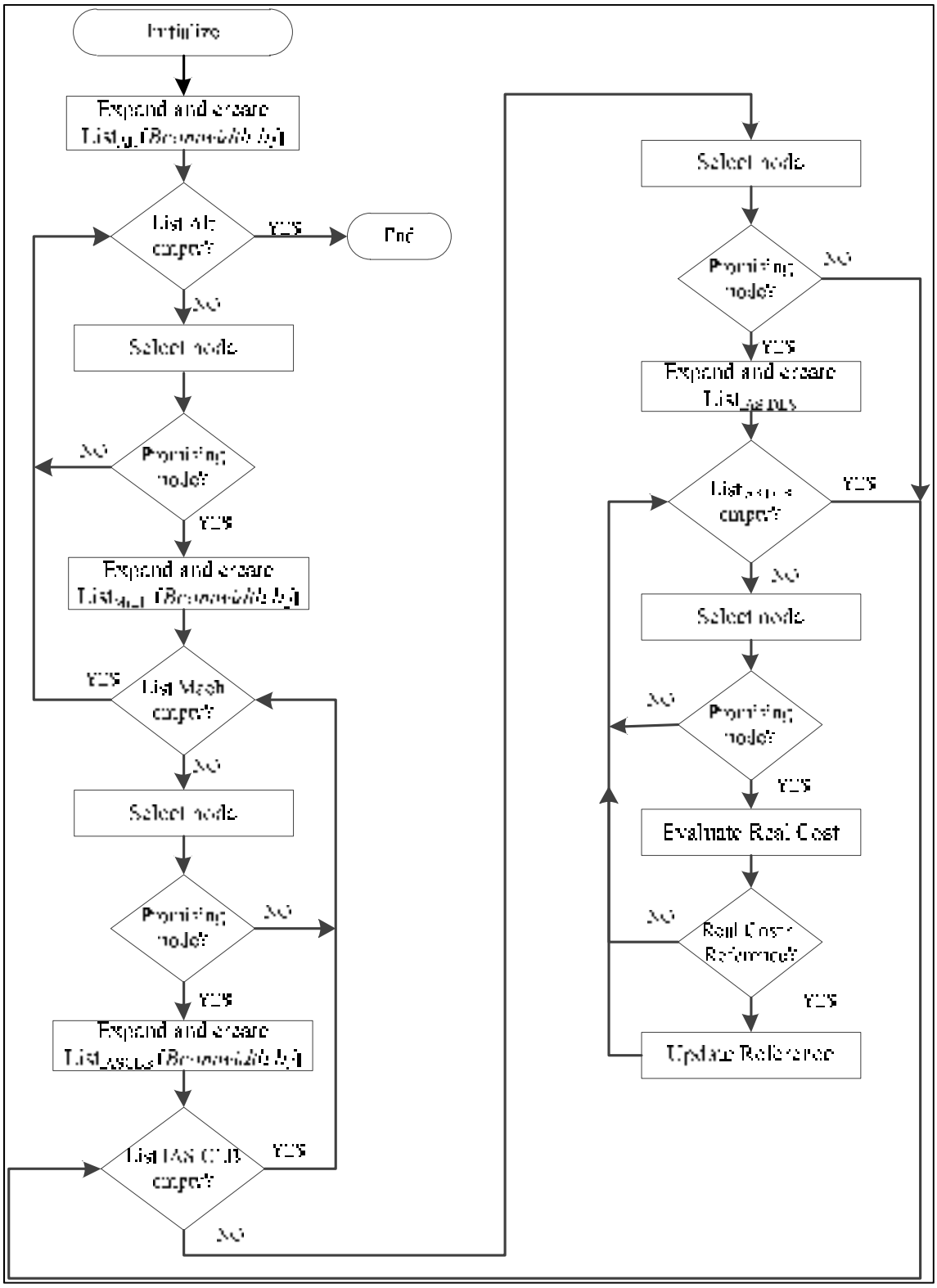

Figure 4.4 Beam search for reference trajectory optimization algorithm's flowchart

4.3.3.6 Pseudo-Code

Algorithm 4.1 Beam Search pseudocode

```
1.  Inputs = coordinates, takeoff weight, C_opt, Reference_Cost = Inf, VNAV_OPTIMAL = []
                    Step 1: first tree level
2.  For each Node Altitude do
3.     Local_Evaluation
4.     Insert Node Altitude into List_ALT
5.  End For
6.  Rank List_ALT
7.  If number of elements in List_ALT< beamwidth (b1)
8.     Prune Node Altitude in List_ALT to fit beamwidth (b1)
9.  End If
                    Step 2: All other nodes
10.  While List_ALT is not empty
11.     Take the first Node Altitude in List_ALTS
12.     Expand Node
13.     For each Node Mach
14.        Local_Evaluation
15.        If Local_Evaluation < Reference_Cost then
16.           Insert Node Mach into queue list_mach
17.        Else
18.           Prune
19.        End If
20.     End For
21.     Rank List_Mach
22.     If number of elements in List_mach< beamwidth (b2)
23.        Prune Node Mach in List_mach to fit beamwidth (b2)
24.     End If
25.     While list_mach is not empty
26.        Take the first Node Mach in list_mach
27.        Expand Node
28.        For each Node IAS do
29.            Local_Evaluation
30.            If Local_Evaluation < Reference_Cost
31.               Insert Node IAS into queue list_IASCLB
32.            Else
33.               Prune
34.            End If
35.        End For
36.        Rank List_IASCLB
37.        If number of elements in List_IASCLB< beamwidth (b3)
38.           Prune Node Mach in List_IASCLB to fit beamwidth (b3)
39.        End If
40.        While List_IASCLB is not empty
41.           Take the first Node Climb IAS in LIST_IASCLB
42.           Expand Node
43.           For each Node Descent IAS (or List IAS Des)
44.               Local_Evaluation
45.               If Local_Evaluation < Reference_Cost then
46.                  Compute Real cost
47.                  If Real cost < Reference_Cost then
48.                     Update Reference_Cost
49.                     Save Solution in VNAV_OPTIMAL
50.                  End If
51.               End If
52.           End For
53.           Remove tested Climb IAS Node from list_IASCLB
54.        End while
55.        Remove tested Mach Node from list_Mach
56.     End while
57.     Remove tested Altitude Node from list_ALT
58. End while
59. Display best cost and VNAV_OPTIMAL
```

4.4 Exhaustive Search Algorithm

An algorithm able to compute the entire vertical reference trajectory combinations defined in Eqs (4.4) – (4.8) is developed. This Exhaustive Search Algorithm is impractical to be implemented in a low processing power device such as the FMS. However, this algorithm is interesting from the optimization perspective as it always provides the optimal solution. This allows validating the solution provided by the Beam Search Algorithm developed in this paper. This Exhaustive Search Algorithm uses the same numerical performance model as the developed algorithm.

4.5 Results

The optimal trajectory was evaluated for three different aircraft. The general characteristics of the aircrafts used for the validation are shown in Table 4.2.

Table 4.2 Aircraft General Characteristics

Aircraft	Min take-off weight (kg)	Min speed (IAS/Mach)	Max speed (IAS/Mach)	Number of turbofans
A	125000	245000	250/0.78	365/0.84
B	80000	160000	240/0.6	335/0.82
C	27000	47000	240/0.5	310/0.8

In the first part of the results, the effectiveness of the optimism coefficient (C_{opt}) is shown along with the values of the pruned nodes and branches at different levels. The optimal vertical navigation profile obtained by the algorithm was compared with the vertical navigation profile solution given by an Exhaustive Search Algorithm to validate the optimal solution. The costs of trajectories obtained using the Beam Search Algorithm developed in this paper were compared with the outputs from a Flight Management System (FMS) simulator called the Part Task Trainer (PTT), obtained from our industrial partner. The PTT used the same numerical performance model as the Beam Search Algorithm and the

Exhaustive Search Algorithm to perform all the required computations. Finally, to validate that introducing weather information does not affect the algorithm's ability to find the optimal (or near optimal) trajectory, these trajectories were evaluated considering the Beam Search Algorithm and the weather parameters, and then these trajectories were compared with the trajectories found by the Exhaustive Search Algorithm using the same weather parameters.

Different city pairs were used to perform the required tests for different aircraft. Table 4.3 shows the different trajectories used for the tests described next with their ICAO codes.

Table 4.3 Flight Distances for Different Trajectories

Airport 1	Airport 2	Distance (nm)
Martinique (TFFF)	Paris Orly (LFPO)	3713
Montreal (YUL)	Vancouver (YVR)	1992
Los Angeles (LAX)	Minneapolis (MSP)	1334
Vancouver (YVR)	Winnipeg (YWG)	1007
Montreal (YUL)	Winnipeg (YWG)	983
Montreal (YUL)	Minneapolis (MSP)	824
Toronto (YYZ)	Winnipeg (YWG)	813

4.5.1 The Optimism Coefficient Effect

The "optimism coefficient" has an important influence on the optimal trajectory found. To illustrate it, a Winnipeg to Montreal flight with a total cost of 12,587 kg was analyzed under the standard atmosphere.

4.5.1.1 The Optimism Coefficient and the Computation Time

The optimism coefficient (C_{opt1}) value varied between 0 and 1 to observe its influence in branch pruning at the different tree levels (between Level 1 and Level 4). The second optimism coefficient was fixed to be C_{opt2} =0.65. As mentioned above in Section 4.3.3.4, C_{opt2} was used only for the last tree level. As previously discussed in Section 4.3.4.4, there are two coefficients, as the descent phase does not have much of an influence as the rest of the other phases, thus it is desired to cut as many nodes at the last level as possible without influencing the other levels.

Table 4.4 shows the flight cost, the number of nodes (branches) pruned at levels 1, 2, and 3-4, the total number of nodes (leaves) not evaluated as a consequence of the pruned branches, and the required computation time for seven different C_{opt1} values, and for a C_{opt2} = 0.65 for Aircraft A.

Table 4.4 Optimism Coefficient Results of a Winnipeg to Montreal Flight for a C_{opt2} = 0.65

C_{opt1}	0	0.2	0.4	0.6	0.8	0.9	1
Flight cost (kg)	12616	12616	12616	12587	12587	12587	12587
Branches (nodes) pruned at the 1st level	5	5	5	3	2	2	2
Branches (nodes) pruned at the 2nd level	77	77	77	45	36	36	36
Branches (node) pruned at the 3rd and 4th level	3	2	2	80	128	129	132
Total of leaves not evaluated	9270	9260	9260	6200	5600	5610	5640
Computation Time (s)	8.1	8.7	8.7	231.56	280.21	279.7	275.03

For a C_{opt1} close to zero, a pessimist coefficient value, more branches are pruned at the highest level (level 1), thus reducing the algorithm execution time. This fact can be seen for $C_{opt1} < 0.6$. However, this fast computation brings with it the consequence that the global optimal cost was not found, because the branch that contained the optimal solution was pruned at a higher level due to the C_{opt1} pessimism value. According to the results in Table 4.4, for a $C_{opt1} >= 0.6$, the optimal trajectory was always found; however, the computation time was considerable higher. For these same tests, for a $C_{opt1} < 0.6$, the algorithm was not able to find the optimal result. However, for a different flight, the solution might be found for a $C_{opt1} < 0.6$.

The computation time is directly related to the number of leaves evaluated, as illustrated in Table 4.4, where for a $C_{opt1} = 0$, a total of 9,270 possible trajectories (leaves) were not evaluated (more branches were pruned) using the real cost, which lead to a computation time of 8.1 s. A total of 5610 possible trajectories were not evaluated (less branches were pruned) with a $C_{opt1} = 0.8$, which lead to a computation time of 280.2 seconds. Obviously, C_{opt1} needs to be adjusted to the required value in order to obtain very good results.

C_{opt2} also has an important effect on the pruning process, as seen in Table 4.5, where the same flight as the one shown in Table 4.4 was executed, but with a less optimistic $C_{opt2} = 0.2$.

Table 4.5 Optimism Coefficient Results of a Winnipeg to Montreal Flight for a $C_{opt2} = 0.2$

C_{opt1}	0	0.2	0.4	0.6	0.8	0.9	1
Flight Cost (kg)	12616	12616	12616	12587	12587	12587	12587
Branches pruned at the 1st level	5	5	5	3	2	2	2
Branches pruned at the 2nd level	77	77	77	45	36	36	36
Branches not evaluated at the 3rd and 4th level	3	2	2	101	209	209	209
Total of leaves not evaluated	9270	9260	9260	6410	6410	6410	6410
Computation time (s)	8.04	8.61	8.62	160.32	160.9	159.43	159.3

It is clear that the "Flight Cost" value is the same in Table 4.4 and Table 4.5, which shows the importance of C_{opt1} in finding the optimal solution. C_{opt2} contributes to rejecting the last node to be evaluated by increasing the number of leaves not evaluated. For a $C_{opt1} > 0.6$, the computation time was reduced for around 100 s in the results in Table 4.5 ($C_{opt2} = 0.2$) compared against the results in Table 4.4 ($C_{opt2} = 0.65$). This computation time reduction is due to the number of branches/leaves not evaluated at the 3rd and 4th levels due to the C_{opt2} pessimist in Table 4.5.

4.5.1.2 The Optimism Coefficient and its Effects in the Convergence of the Algorithm

The developed algorithm tends to converge to the optimal solution as it was seen in the results provided by Table 4.4 and Table 4.5. However, the convergence was affected by the selection of parameters C_{opt1} and C_{opt2}.

By selecting high values of C_{opt}, the algorithm converges to the optimal solution (12,587 kg) at the expenses of computation time. By selecting low values of C_{opt}, the algorithm also converges to the optimal solution, but fails to deliver the optimal solution staying in a "close" local optimal (12,616 kg). Selecting the right combination of C_{opt} values makes the algorithm to systematically converge to the optimal solution.

4.5.2 The Beam Search Algorithm and the Exhaustive Search

The Beam Search Algorithm was compared with the Exhaustive Search Algorithm (which guarantees the global optimal) for different flights to measure the quality of the solution. For these sets of tests, the standard atmosphere was used, and the *Cost Index* = 0, since the goal of these tests was to observe if the optimal flight profile was found. This test was performed for three different aircraft

4.5.2.1 Aircraft A

The airports considered in the tests were Montreal (YUL), Vancouver (YVR), Toronto (YYZ), Winnipeg (YWG), Minneapolis (MSP), Los Angeles (LAX), Martinique (TFFF) and Paris Orly (LFPO). For aircraft A, the number of possible leaves (flight combinations) is close to 15,730. Table 4.6 shows the optimal cost for each of the flights by means of an Exhaustive Search Algorithm and by the Beam Search Algorithm developed in this paper.

For the evaluation of Aircraft A, the coefficients values were $C_{opt1} = 0.6$ and $C_{opt2} = 0.2$. The flight cost was compared in Table 4.6 as calculated by the Exhaustive (first row) and the Beam Search (second row) algorithms. This comparison showed that 6 out of 7 flights provided the same cost, while for one trajectory, the flight cost difference was of only 44 kg. In order to find the optimal flight cost, the C_{opt1} should be higher to make the evaluating function more "optimistic", this way, fewer nodes (branches) would be pruned and the optimal flight might be found. However, incrementing the C_{opt1} as seen in Table 4.4 and Table 4.5 would increment the computation time.

As shown in Table 4.6, the algorithm was capable of not evaluating up to 9350 leaves due to pruned nodes (branches) for the flight YUL-YYZ in Table 4.6. This means that only around 30% of the space search was evaluated (4690 leaves).

Table 4.6 Algorithm Performance for Aircraft A

Trajectory	**YUL-YVR**	**YUL-YYZ**	**YVR-YWG**	**YYZ-YWG**	**YUL-MSP**	**LAX-MSP**	**TFFF-LFPO**
Exhaustive Optimal Flight Cost (kg)	27793	4901	14790	12587	12692	19403	53091
Beam Search Flight cost (kg)	27793	4945	14790	12587	12692	19403	53091
# of leaves not evaluated	5140	9350	6370	6410	6370	6030	6540
Step Climbs	3	0	1	0	0	2	1
Nodes evaluated (%)	63	33	55	54	55	57	53

4.5.2.2 Aircraft B

For this Aircraft, the total number of leaves (flight combinations) is around 4320. As with Aircraft A, Table 4.7 shows the optimal flight cost trajectories found by using the Exhaustive Search Algorithm versus the trajectories found with the developed Beam Search Algorithm.

Table 4.7 Algorithm Performance for Aircraft B

Trajectory	YUL-YVR	YUL-YYZ	YUL-YWG	YVR-YWG	YYZ-YWG	YUL-MSP	LAX-MSP
Exhaustive Optimal Flight Cost (kg)	17685	2650	78794	12220	7188	15014	13075
Beam Search Flight cost (kg)	17685	2650	78794	12220	7188	15014	13075
# of leaves not evaluated	2712	1924	1850	1997	1939	1784	3250
Step Climbs	1	0	0	0	0	0	0
Nodes evaluated (%)	37	55	57	54	55	59	28

The algorithm was able to find the optimal solution for all flight cases. The optimism coefficients used for aircraft B were: C_{opt1} = 0.6 and C_{opt2} = 0.1. The numbers of nodes (branches) note evaluated are shown in Table 4.7. Cases such as the Los Angles (LAX – Minneapolis (MSP) can be found, where the search space was explored only by 28% (1070 leaves evaluated).

4.5.2.3 Aircraft C

For this aircraft, the total number of leaves (flight combinations) is close to 6,468. As with Aircraft A, and Aircraft B, Table 4.8 gives the results found for Aircraft C. In the same way, the Exhaustive Search Algorithm flight is compared against the developed Beam Search Algorithm. Since this is a medium-range aircraft, the destinations considered are different from the destinations for the other aircraft (A & B). The new airport added is Chicago (ORD).

Table 4.8 Algorithm Performance for Aircraft C

Trajectory	YUL-YYZ	YYZ-ORD	ORD-MSP	YUL-MSP	LAX-MSP
Exhaustive Optimal Flight Cost (kg)	1072	1387	1122	1060	1521
Beam Search Flight cost (kg)	1072	1387	1122	1060	1521
# of leaves not evaluated	3990	5844	4320	3990	4992
Nodes evaluated (%)	38	10	33	38	23

For Aircraft C, the algorithm is able to find the optimal trajectory for all flights. For these tests, the optimism coefficients are C_{opt1} = 0.6 and C_{opt2} = 0.1. The number of nodes (branches) and leaves (flight combinations) not evaluated is shown in Table 4.8. For this aircraft, it can be seen that for the YYZ-ORD flight, only around the 10% (624 flight combinations) of the search space was evaluated.

4.5.3 The Beam Search Algorithm and Results Obtained from the FMS/PTT

It will be useful to determine if the Beam Search Algorithm is able to improve the trajectories suggested by the FMS/PTT. A flight cost comparison between the optimal trajectories generated by the FMS/PTT and the optimal trajectory generated by the algorithm developed in this paper was executed. The FMS/PTT is considered as a black box since the methodology used in the FMS/PTT is confidential.

To make this comparison, inputs were introduced first in the FMS/PTT, obtaining the optimal $\text{Vertical}_{\text{FMS/PTT}}$ reference profile. The same inputs were then introduced to the Beam Search Algorithm to obtain the optimal $\text{Vertica}_{\text{ALGO}}$ reference profile. The costs of both Vertical reference profiles were calculated using the "real cost computation" method (see Section 4.2.2), and these costs were compared to observe the flight cost differences between the commercial FMS and the developed algorithm.

The “input parameters” were: the initial and final destination points, the reference Lateral Navigation trajectory, the Cost Index and the initial aircraft weight.

For Aircraft B, a flight from Montreal to Vancouver was simulated varying the Cost Index up to 90. The results are visualized in Figure 4.5, and show the variation of the flight cost with the Cost Index.

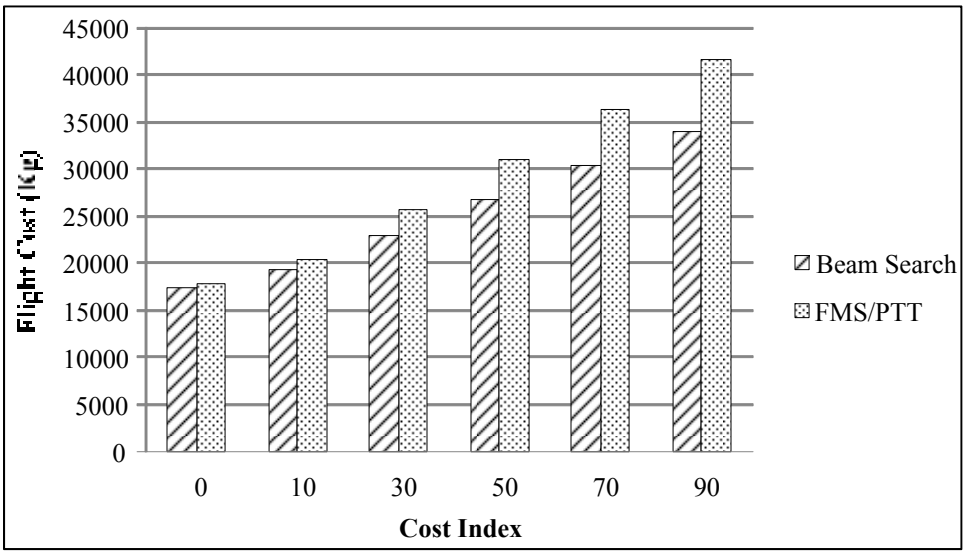

Figure 4.5 FMS/PTT and algorithm cost comparison

As illustrated in Figure 4.5, the algorithm developed in this paper reduced more the flight cost with the generated vertical reference trajectories than the reference FMS/PTT.

4.5.4 The Beam Search Algorithm Considering Wind Influence

Weather information was obtained from the Environment Canada website. The parameters of interest were the wind speed, the wind direction, the temperature and the pressure. The tests performed in this section validated if the developed algorithm was able to find the optimal (or near-optimal) solution by considering the stochastic wind influence.

Firstly, an Exhaustive Search was performed for different flights by considering the available weather information. The same flights were executed using the algorithm developed in this paper. The flight costs obtained using both algorithms (*Beam Search* and *Exhaustive Search*) were compared and the results difference between both algorithms is shown in Figure 4.6.

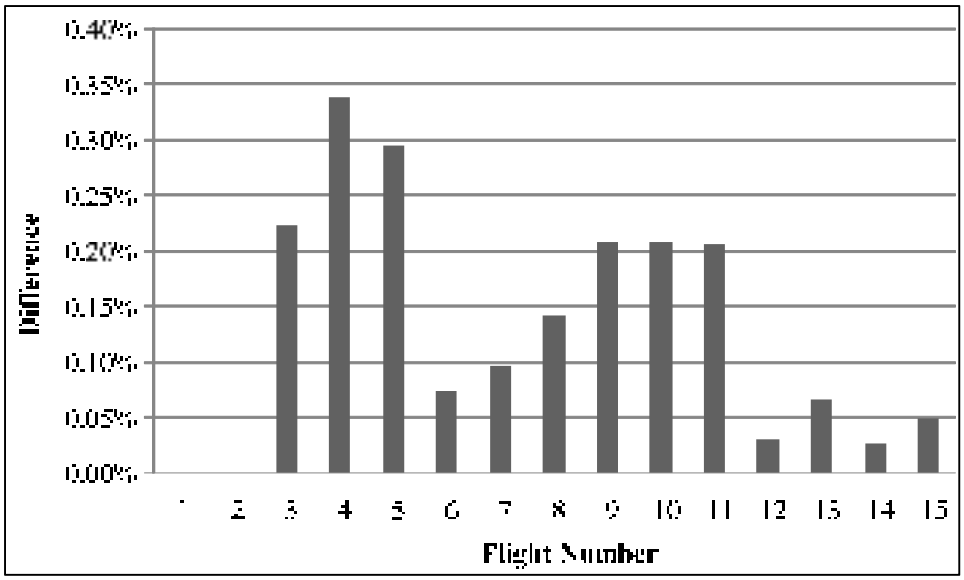

Figure 4.6 Flight Cost Solution Difference between the Exhaustive Search and the Beam Search Algorithm

As expected, when the weather influence was introduced, the algorithm was not always able to converge to the optimal solution, and for the flights tests set, only two flights, as shown in Figure 4.6 (flight 1 and flight 2) provided the optimal solution. This lack of convergence was obtained because the heuristics fail to capture the wind influence; thus, nodes corresponding to the optimal solution were pruned. Very good suboptimal solutions were found because the "worst" solution (flight 4 in Figure 4.6) was only 0.34% more expensive than the optimal solution.

The optimal solutions provided by the Exhaustive Search and by the Beam Search Algorithm for Flight 2 in Figure 4.6 are shown in Table 4.8. The developed algorithm was executed for that flight with the same weather information for different values of C_{opt1} and an arbitrary

constant C_{opt2} = 0.3. The initial gross weight was of 120000 kg. Results are found in Table 4.9.

Table 4.9 Optimal Flight Provided by the Exhaustive Search Algorithm

Flight	Climb IAS	Mach	Altitude	Descent IAS	# Step Climbs	Flight Cost (kg)
YUL – YVR	300	0.78	37000	335	1	26863

Table 4.10 Results Provided by the "Beam Search" Algorithm Considering Weather

C_{opt1}	**0.4**	**0.6**	**0.8**	**1**
Flight Cost (kg)	26892	26863	26863	26863
# of leaves not evaluated	4274	4200	4164	3984

The results in Table 4.10 show that, as expected, the number of pruned branches decreases as the coefficient is less optimistic.

4.6 Conclusion

This paper described an algorithm to optimize the vertical reference flight trajectory by using a numerical performance model. The problem was solved using a decision tree in the implementation of the deterministic "Beam Search" algorithm. The "Optimism Coefficient" has influenced the algorithm's aggressiveness. The algorithm made a compromise between flight time and fuel burn optimization. Fuel burn reduction has a direct effect in the reduction of polluting emissions, which are known to cause global warming and health problems. Therefore, the developed algorithm, taking advantage of "step climbs" and "weather parameters" to calculate the optimal aircraft trajectory, was conceived by taking into account

ATM constraints such as constant Mach and constant altitude segments following a pre-defined lateral reference route.

The algorithm's ability to reduce the number of possible combinations by up to 86 % in some cases, was clearly demonstrated. This brings us a consequence of a low computation time, which is desirable for low computation power avionics devices such as the Flight Management System (FMS). It was also shown that by correctly selecting the "optimism coefficients", the algorithm found the optimal solution in many different flight cases.

It has been observed that the solutions found by the algorithm for different cost indexes were always more economical than the results obtained using a commercial FMS.

As future work, it is desirable to improve the heuristics when implementing the weather. It will be worthwhile to analyze the flights of a larger number of aircraft operating in the same airspace to obtain the algorithm's performance in the presence of other aircraft or even other constraints such as Non-Fly Zones. It would also be of interest to develop a new technique to determine the best locations to realize step climbs instead of using constant trajectories at fixed time or fixed distance points.

CHAPTER 5

AIRCRAFT VERTICAL ROUTE OPTIMIZATION BY BEAM SEARCH AND INITIAL SEARCH SPACE REDUCTION

Alejandro Murrieta-Mendoza, Laurane Ternisien, Bruce Beuze, Ruxandra Mihaela Botez

École de Technologie Supérieure / Universite du Quebec, Montreal, Canada
Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticty

This article was submitted for publication to the AIAA Journal of Aerospace Information Systems on May 2017

Résumé

Ce chapitre décrit un algorithme d'optimisation fournissant un profil de navigation verticale économique en déterminant les combinaisons de vitesse de montée, de croisière et de descente, ainsi que des altitudes, pour un avion afin de minimiser les coûts de vol. L'algorithme de calcul utilise une méthodologie de réduction de l'espace de recherche pour réduire le nombre initial de combinaisons de vitesse et d'altitude disponibles. La solution optimale a été trouvée en mettant en œuvre l'algorithme de recherche de faisceau. Une fonction de délimitation qui estime correctement le coût du vol en considérant les paliers de montées a été développé pour réduire le nombre de calculs requis par l'algorithme de recherche de faisceau. La quantité de carburant requis pour le vol a été obtenue en utilisant une méthode basée sur une base de données de performance plutôt que sur les équations de mouvement pour calculer la consommation du carburant. La base de données de performance a été développée en utilisant des données expérimentales de vol. Pour valider l'algorithme, ses résultats ont été comparés à ceux de trois autres algorithmes: "une recherche exhaustive", "l'algorithme de recherche de faisceau " et " la réduction de l'espace de recherche ". La solution fournie par l'algorithme a également été comparée à la solution fournie par un système de gestion de vol. À la suite de cette comparaison, l'algorithme a systématiquement trouvé les solutions optimales, en termes de coût de vol, meilleures que celles fournies par le système de gestion de vol.

Abstract

This paper describes an optimization algorithm that provides an economical Vertical Navigation profile by finding the combinations of climb, cruise and descent speeds, as well as altitudes, for an aircraft to minimize flight costs. The computational algorithm takes advantage of a space search reduction methodology to reduce the initial number of available speed and altitude combinations. The optimal solution was found by implementing the beam search algorithm. A bounding function that correctly estimates the flight cost by considering step climbs was developed to reduce the number of calculations required by the Beam Search algorithm. The full flight fuel burn was obtained using a performance database-based method. The algorithm uses a numerical performance model instead of equations of motion to compute fuel burn. The database was developed by using flight experimental data. To validate the algorithm, its results were compared to those of three other algorithms: "an exhaustive search", "Beam Search" and "Search Space Reduction". The solution provided by the algorithm was also compared to the solution provided by a Flight Management System. Following this comparison, the algorithm systematically found the optimal solutions, which were better in terms of flight cost than those provided by the Flight Management System.

5.1 Introduction

The scientific community's concern about global warming has led different industrial sectors to develop highly efficient systems to reduce polluting emissions released to the atmosphere. One of these sectors is the aeronautical industry, which has set the ambitious goal of reducing its carbon dioxide (CO_2) emissions to 50% of those recorded in 2005 by the year 2050 (IATA, 2011). This ambitious goal, was set by considering that the aeronautical industry is responsible for 2% of the annual CO_2 released to the atmosphere (ICAO, 2010), and that air traffic is expected to grow in the coming years. CO_2 is well known for its effects on global warming, and it has been shown that CO_2 emissions affect the mean wind speed by making round transatlantic flights longer than with current CO_2 concentrations. Longer flights require more fuel and therefore release more CO_2 to the atmosphere (Williams, 2016).

In addition to CO_2, other emissions are released to the atmosphere by aircraft. Table 5.1 shows some of the polluting emissions released by aircraft fuel consumption and their effects on the atmosphere and/or directly on the population.

Table 5.1 Polluting emissions generated by fuel burn and their effects

Polluting Emission	**Effect**
Carbon Dioxide (CO2)	Global warming, and wind pattern change
Contrails	Global warming (Green, 2009)
Nitrogen Dioxide (NOx)	Ozone layer depletion (Crutzen, 1970)
Hydrocarbons	Linked to respiratory problems and nervous system shock ((ATSDR), 1999)
Noise	Stress and heart problems (Black, Black, Issarayangyun, & Samuels, 2007)

Models to estimate these emissions are available in the literature such as in (Ashok, Dedoussi, Yim, Balakrishnan, & Barrett, 2014; Sabatini, Gardi, Ramasamy, Kistan, & Marino, 2010).

Different alternatives to reduce fuel consumption have been applied on new generations of aircraft and engines. Promising future technologies aiming to reduce fuel consumption were developed, including new fuel resources such as biofuel (ATAG, 2009; Nojoumi et al., 2009), morphing wing concepts (Vincent & Botez, 2015), and in flight re-fueling (Nangia, 2006).

With the present technology, implementing various measures in airline operations such as engine and aircraft washing, taxiing with one engine, Auxiliary Power Unit (APU) use reduction, and flight planning, among others, can bring significant fuel reduction benefits, as discussed by McConnachie (McConnachie et al., 2013). Reducing fuel consumption is an

ongoing concern for airlines, because of the fact that around 26 % of airlines expenses are related to fuel requirements and expenses (ATAG, 2014). Fuel savings for different air transportation procedures such as airport congestion, air traffic flow management and tools to support a more efficiency decision process were discussed in (Balakrishnan, 2016).

Another way to reduce fuel burn is by optimizing the aircraft's flight reference trajectory. Jensen (Luke Jensen et al., 2014; Luke Jensen et al., 2013; Luke Jensen et al., 2015) suggested the possible gains to be made in improving flight reference trajectories, as many aircraft do not fly at their optimal speeds or altitudes. Another study on fuel flow in flights within Turkey also suggested that aircraft trajectories should be improved to reduce fuel consumption (Turgut et al., 2014).

Based on those studies, several countries acknowledge that the Air Traffic Management (ATM) system should be improved to allow more aircraft to fly at their optimal trajectory while guaranteeing safety. The United States is developing and deploying the Next Generation Air Transportation System (NextGEN) (Jackson, 2008), while Europe is developing and deploying the Single European Sky ATM Research (SESAR). These systems are based on what is called Trajectory Based Operations (TBO), wherein each aircraft is able to negotiate its most economical trajectory with ATM and other (relevant) airborne aircraft.

The future TBO will be time-based which means that trajectories will have a Required Time of Arrival (RTA) constraint at a given destination or at a set of waypoints. This RTA will allow ATM to ensure aircraft separation and will allow a smooth transition for the aircraft flight between the end of cruise to the descent phase. These trajectories are also known as 4D reference trajectories. There have been discussion about automated systems that will manage this future air space (Gardi, Sabatini, Ramsamy, et al., 2015). However, current 3D trajectories (filling a flight plan containing the waypoints to follow, the altitude, and speed of the aircraft) still represent the basis of future trajectories development.

Descent phase optimization in terms of fuel burn and noise reduction has been of great concern due to descents proximity to populated areas and to the health effects that pollution and noise can cause. The Continuous Descent Approach/Operation (CDA/CDO) has brought benefits to fuel burn reduction. In this operation, the aircraft descends following a constant slope, instead of the conventional cruise-step descent. This approach has been successfully tested at different airports (Clarke et al., 2013; Kwok-On et al., 2003; Kwok-On et al., 2006; Novak, Buckai, & Dadisic, 2009; Sprong et al., 2008; Stell, 2009). For this operation to be carried out successfully, the aircraft's weight and position at the beginning of its descent must be estimated correctly (Johnson, 2011; Stell, 2010). A successful CDA/CDO avoids missing the desired descent path and having to execute the Missed Approach (Go-around) procedure, which entails high fuel consumption and thus high CO_2 emissions (R. Dancila et al., 2013; Murrieta-Mendoza et al., 2014). It is important to mention that besides the conventional aircraft, it is expected that UAVs will also share the airspace. Optimization algorithms for these devices have also been developed as in (A. Chamseddine, Zhang, Rabbath, Join, & Theilliol, 2012; Abbas Chamseddine, Zhang, Rabbath, & Theilliol, 2012; Chen et al., 2016)

Concerning operations during the cruise phase, the benefits and consequences of flying at constant Mach have been analyzed (Filippone, 2007; Wieland, Hunter, & Schleicher, 2008), as well as the benefits of reducing cruise speed (Bonnefoy & Hansman, 2010). Lovegren discussed the importance of climb at different altitudes during the cruise phase as a means to reduce fuel consumption (Lovegren, 2011).

For the reference trajectory optimization, the aircraft fuel burn can be modeled using a set of differential equations called the Equations of Motion (EoM), or using a numerical performance model, as shown in this paper. One of the most popular approaches found in the literature to optimize reference trajectories is to use EoMs and optimal control for a mixed linear programming problem. The flight trajectory has been analyzed in its different aspects: by respecting ATM constraints (Valenzuela & Rivas, 2014), by taking advantage of the free flight concept for the cruise phase (Franco & Rivas, 2011; Pargett & Ardema, 2007) and by

considering all the main flight phases (Franco & Rivas, 2015). A dynamic programming algorithm optimizing firstly the lateral reference trajectory and then the vertical reference trajectory by taking winds into account was developed in (Ng et al., 2014).

Graph search theory, such as dynamic programming taking into account weather and RTA, has been implemented (Miyazawa et al., 2013). Fixed Mach and step climbs were considered, accounting for winds. Hagelauer (Hagelauer & Mora-Camino, 1998) used Dynamic Programming with Neural Networks to optimize the trajectory, highlighting that the Flight Management System (FMS) normally uses a database instead of EoM to solve the trajectory optimization problem. Other authors have proposed algorithms that utilize FMS functions and EoM, as shown in (Villarroel & Rodrigues, 2016).

Reference flight trajectories were optimized by taking into account object avoidance using the A* algorithm (Sadovsky, 2014), in which dynamical weather was taken into account, Dijkstra's algorithm (Rippel et al., 2005) for low altitudes, and Particle Swarm Optimization (Cobano et al., 2013). However, these algorithms would be difficult to implement in future Flight Management System (FMS) functions due to memory and processing requirments. They would be more appropriate to use in flight optimization systems prior to airborne flight planning.

Due to processing power issues, the FMS normally uses a numerical aircraft performance database, which contains the aircraft aero-propulsive performance data to compute the fuel burn for the different flight phases. Murrieta *et al.* developed in (Murrieta-Mendoza, Demange, et al., 2015) a numerical performance database by using a Citation X Level D qualified flight simulator equipped with the highest qualification for flight dynamics.

Optimization algorithms using a numerical database must solve a combinatory optimization problem, since the number of available speeds and altitudes is limited. Optimization algorithms have been developed to find the most economical combination of speeds and altitudes for the vertical reference trajectory using a numerical performance database.

Dancila *et al.* (B. Dancila et al., 2013) developed a fuel burn estimator that was used to determine a flight's optimal altitude. Gagné *et al.* developed in (Jocelyn Gagné et al., 2013) a semi-exhaustive optimization algorithm. Félix *et al.* in (R.S. Felix Patron et al., 2013) used the Golden Section Search algorithm to optimize the complete flight trajectory for short flights, and they also implemented the step climbs for long haul flights. Sibidé in (Sidibe & Botez, 2013) used dynamic programming to solve the vertical trajectory of reference problem. In (Roberto Salvador Felix Patron et al., 2013), the genetic algorithms were implemented to find the optimal vertical trajectory of reference. In (Murrieta-Mendoza, Félix-Patrón, & Botez, 2015), genetic algorithms were implemented to find the best trajectory waypoints

Lateral reference trajectory profile optimization using a numerical performance database has been developed in (Murrieta-Mendoza & Botez, 2014a), where the Dijsktra's algorithm was implemented by taking into account winds and temperatures. In (Roberto S. Félix-Patrón et al., 2014), Felix *et al.* implemented genetic algorithms to optimize the flight route. The bee's optimization algorithm was implemented using a dynamic grid to find the optimal reference trajectory (Murrieta Mendoza, Bunel, & Botez, 2015).

Taking advantage of both types of reference trajectories (vertical and lateral) by using a numerical performance database has also been proposed. The optimal vertical trajectory of reference was found for executing step climbs by with a search space reduction technique, and then, five different lateral reference trajectories were evaluated following the same vertical reference trajectory to find the best combinations of both lateral and vertical reference trajectories (Murrieta-Mendoza, 2013). Felix *et al.* in (Roberto S Félix-Patrón et al., 2014; Roberto S. Félix-Patrón & Botez, 2014) implemented genetic algorithms to couple both trajectories of reference in a larger search space. This allowed improved the results as more search space was explored.

In performing the 4D trajectory optimization using a numerical performance database, the most economical combinations of Mach numbers to fulfill the RTA constraint was determined using the ant colony optimization (Murrieta-Mendoza, Hamy, & Botez, 2015).

Most of the algorithms described above require high computation time and/or high memory requirements for a low processing power device such as the FMS, or they require stochastic concepts to generate the optimal trajectory solutions, which are difficult for authorities to certify. For this reason, an algorithm that finds the optimal solution or a *good* sub-optimal solution while requiring only a low computation power would be desired.

The objective of this paper is to expose a new computational algorithm which is able to reduce the trajectory combinations (reduce the memory requirements), converges to the optimal solution in a short time frame, and can be certified, as it is deterministic. All these are desired requirements are desirable for algorithm implementation in an FMS. The solution provided by the developed algorithm was compared to an "Exhaustive Search" algorithm to validate how far the computed solution is from the optimal solution. In the new algorithm, s*tep climbs* during cruise were evaluated and the *CDA/CDO* procedure was implemented for the descent phase. ATM constraints such as "constant altitude segments" and "constant speeds" and weather information were also taken into account.

This paper is organized as follows: Conventional flight is described first, and then the PDB used to compute the flight cost. The method used to compute the flight cost with this PDB is then described, followed by a detailed explanation of the new algorithm's methodology. The results and their interpretation are given next. Finally, the conclusions regarding the results obtained using this new algorithm are presented.

5.2 Methodology

5.2.1 The Studied Flight

A typical commercial flight consists of three main phases: climb to the targeted cruise altitude phase, cruise with the ability of changing flight levels, and descent to the destination airport. In accordance with international regulations, during climb, and descent the aircraft's speed must be at or below 250 Indicated Air Speed (IAS) while at altitudes below 10,000 ft. The studied flight begins at 2,000 ft at 250 IAS; this is called the *initial climb*. After the initial climb, if a speed higher than 250 kts is required, an acceleration is needed to reach the desired speed. A constant IAS climb is then followed until after the crossover altitude (altitude where the True Air Speed (TAS) of the scheduled IAS equals the desired Mach number in terms of TAS) where climb is then executed during a constant Mach. This altitude is called the crossover altitude. The point where the climb ends and the cruise phase begin is called the Top of Climb (ToC). During long flights, changes in altitude during cruise (step climbs) might be performed to improve the flight performance. For the algorithm developed in this paper, when computing the cruise phase costs, step climbs are evaluated at a pre-defined flight time. This is done to avoid executing step climbs that might be too close together. In reality, step climbs executed too often might be denied by ATM. Once at the end of cruise, at the Top of Descent (ToD), the descent phase procedure is executed in a similar way as in climb, and is the CDA.

5.2.2 The Numerical Performance Model

The aircraft fuel information is provided in the form of a numerical performance model which was developed from experimental flight data. This model consists of a set of databases for different flight phases. The typical flight phases with their corresponding inputs and outputs for a commercial aircraft are shown in Table 5.2.

Table 5.2 Numerical Performance Model Sub-databases

Sub-database	**Inputs**	**Outputs**
IAS Climb	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)
Climb acceleration	Gross weight Initial IAS (knots) Acceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burn (kg) Horizontal traveled distance (nm) Altitude needed (ft)
Climb Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)
Cruise Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel flow (kg/hr)
Descent Mach	Mach Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)

Table 5.2 Numerical Performance Model Sub-databases (continue)

Sub-database	Inputs	Outputs
Descent deceleration	Gross weight Initial IAS (knots) Deceleration altitude (ft) Delta speed to accelerate (knots)	Fuel burn (kg) Horizontal traveled distance (nm) Altitude needed (ft)
IAS Descent	IAS (knots) Gross weight (kg) ISA deviation temperature (ºC) Altitude (ft)	Fuel burn (kg) Horizontal traveled distance (nm)

All inputs should be available in order to obtain the desired outputs. Altitudes in the inputs column are multiples of 1,000 ft to comply with regulations; aircraft must fly at flight level (pressure altitudes) multiples of 1,000 ft to guarantee their separation with traffic.

5.2.3 Flight Cost: Fuel burn and Flight Time Computations Using the Numerical Performance Model

During cruise, the aircraft normally flies at constant Mach number and constant altitude segments in order to comply with ATM constraints. This flight can be achieved by selecting one speed and one altitude from the numerical performance model. However, the aircraft's gross weight and the ISA deviation temperature present high variation during the flight. Fuel is burned, and thus its weight diminishes to a value that is not exactly among the ones available within the performance model inputs. ISA deviation temperature has a stochastic behavior as the weather changes during flight. For this reason, interpolations within the available temperatures and gross weights inputs are executed.

A method to compute a vertical trajectory flight cost using a numerical performance model was proposed in (Alejandro Murrieta-Mendoza & Ruxandra M. Botez, 2015), and it is used in this paper. A brief explanation of this method is described below.

5.2.4 Interpolations: Computing the Required Value from the Numerical Performance Model

The numerical performance model presents a linear behavior for short intervals of weight, temperature, speed and altitude. This allows the use of linear interpolation techniques such as the Lagrange interpolation, as given in Eq. (1):

$$Interpolation(v) = \frac{v - l_1}{l_0 - l_1} f_0 + \frac{v - l_0}{l_1 - l_0} f_1 \tag{5.1}$$

where f_0 and f_1 are respectively the minimal and the maximal output values used for interpolation. These values are directly obtained from the numerical performance model. v is the value to interpolate. l_0 and l_1 are the low and the high input values enclosing the desired value v. The interpolation sequence is shown in Figure 5.1.

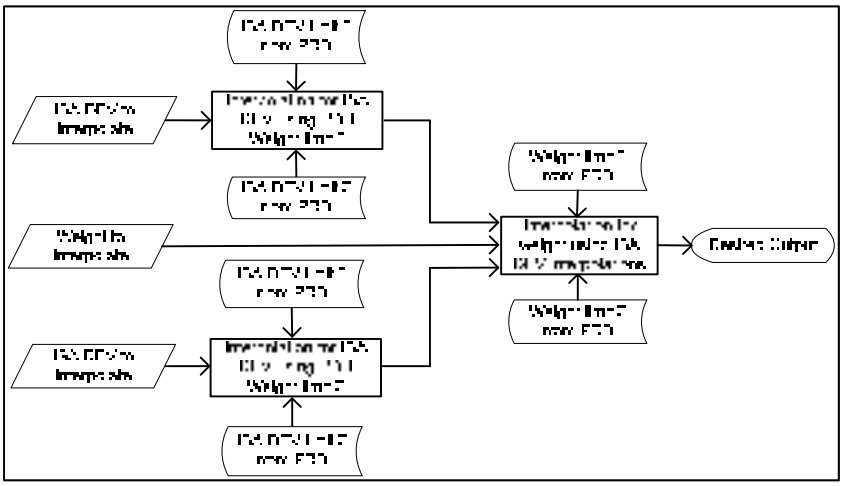

Figure 5.1 General Interpolation Graphic Representation

It is important to identify the input values enclosing the required values. These values are the limits in Figure 5.1. They are required for the gross weight *W*, and the ISA temperature deviation *TMP*. For example, if for *TMP* = 3, the available ISA temperature deviations inputs are defined as […, 0, 5, 10,…], then the lower limit l_0 enclosing *TMP* is equal to 0, and the higher limit l_1 enclosing *TMP* is equal to 5. The same process is followed for the gross weight.

Following the sequence shown in Figure 5.1, if the value of interest is the fuel flow, the process defined from Eq. (5.2) to Eq. (5.8) is executed as follows.

Firstly, two different fuel flows are obtained from the lower weight value ($W\ l_0$), and for the two different ISA temperature deviation values ($TMP\ l_0$ and $TMP\ l_1$) enclosing the required *TMP*.

$$ff_{11} = Cruise\ Mach\ Output\ (Mach\ Number, W\ l_0, TMP\ l_0, Altitude) \tag{5.2}$$

$$ff_{12} = Cruise\ Mach\ Output\ (Mach\ Number, W\ l_0, TMP\ l_1, Altitude) \tag{5.3}$$

Then, the same type of equations is obtained for the higher weight value $W\ l_1$.

$$ff_{21} = Cruise\ Mach\ Output\ (Mach\ Number, W\ l_1, TMP\ l_0, Altitude) \tag{5.4}$$

$$ff_{22} = Cruise\ Mach\ Output\ (Mach\ Number, W\ l_1, TMP\ l_1, Altitude) \tag{5.5}$$

An interpolation is performed to find the fuel flow (*ff*) for the required *TMP* value for each value of *W* (higher and lower):

$$ff_1 = Interpolation\ (ff_{11}, ff_{12}, TMP\ l_0, TMP\ l_1, TMP) \tag{5.6}$$

$$ff_2 = Interpolation\ (ff_{21}, ff_{22}, TMP\ l_0, TMP\ l_1, TMP) \tag{5.7}$$

Finally, interpolations are performed to compute the fuel flow (*ff*) for the required grow weight *W*.

$$ff = Interpolation\ (ff_1, ff_2, W\ l_0, W\ l_1, W) \tag{5.8}$$

The same process can be performed to obtain the fuel burn, the horizontal traveled distance, or the required altitude after an accelerated or decelerated flight.

5.2.5 The Flight Cost Computation: Fuel Burn and Considerations

Fuel burn is provided either as the total fuel burn required to climb (or descend) between two altitudes, as the fuel burn required to accelerate to a given speed, or as a fuel flow during the cruise phase. For the first two cases, the fuel burn is obtained directly. For the case of cruise, a flight time should be provided.

The fuel burn during a cruise phase can thus be computed as shown in Eq. (5.9), where *ff* corresponds to the interpolated value, and $Flight\ Time_{seg}$ corresponds to the time required to travel a given segment. In this paper, this segment is 25 nautical miles because it provides a good compromise between computation time and accuracy.

$$FB_{Cruise} = ff\ \times Flight\ Time_{seg} \tag{5.9}$$

The total fuel burn is then the sum of all the required fuel. Fuel is burned as it is used by the engine to generate thrust, thus diminishing the aircraft's weight. To emulate this behaviour, fuel burn is reduced from the total aircraft's weight at each 1,000 ft during climb or descent and at every 25 nautical miles during the cruise phase.

For the reference trajectory optimization, not only the fuel is of interest, but also the flight time. The latter is computed using the aircraft ground speed defined in Eq.(5.10), where *GS*

refers to the Ground Speed, *TAS* refers to the True Aircraft Speed, *WS* refers to the Wind Speed, *WD* refers to the Wind azimuth, and φ refers to the aircraft's azimuth.

$$GS = TAS + WS\ cos(\varphi - WD) \tag{5.10}$$

Flight time can then be computed using the *GS* and the pre-defined distance for cruise (25 nautical miles) or the horizontal traveled distance provided from the numerical performance model. The *Total Flight Time* is then the sum of all the segments flight times.

Using the *Total Fuel Burn*, and the *Total Flight Time*, it is possible to compute the *Total Flight Cost*. To avoid using currency, which can have important fluctuations, the total flight cost is computed in terms of *kg* of fuel. For this calculation, *Flight Time* is related to *Fuel Burn* terms by using a parameter called the Cost Index (*CI*). This parameter is defined by each airline before each flight and represents the time related costs such as crew salary, maintenance, missing connections, etc. Using *fuel burn* and *flight time*, the *total flight cost* is computed in the FMS as shown in Eq. (5.11).

$$Total\ Flight\ Cost\ (kg) = fuel\ burn + CI * Flight\ Time * 60 \tag{5.11}$$

The number of interpolations required to compute a single flight is obviously quite high, and thus this method, while accurate, is time consuming; therefore, to follow the algorithm described in this paper, it is desirable to avoid executing this total flight cost computation as much as possible.

This accurate, but time consuming method is subsequently referred to as Real Flight Cost Computation (RFCC) in the remainder of this paper.

5.2.6 Problem Definition: The Vertical Reference Trajectory Optimization

The vertical reference trajectory consists of the combination of speeds and altitudes that the aircraft has to follow given a ground track trajectory. Following this definition, and according with the numerical performance model, the design variables are the IAS during climb, the Mach number, the Altitude, and the IAS during descent.

An aircraft's gross weight has an important influence on the fuel cost, as a heavier aircraft requires more fuel to produce enough lift. However, an aircraft's weight is not a control variable, as it might be defined by airline policies, given the pilot's preferences. Temperature also has an influence on engine fuel requirements, as low temperatures are desirable for optimal engine performance. Nevertheless, wind has a stochastic value that is completely not possible to control.

Taking all these hypotheses into account, the conventional vertical reference trajectory optimization solution can be defined as a vector in next Eq. (5.12)

$$\begin{aligned} &Vertical\ Reference\ Trajectory \\ &= [Climb_{IAS}\ \ Mach\ \ \ Altitudes\ \ \ Descent_{IAS}] \end{aligned} \tag{5.12}$$

Each vector element on the right hand side of Eq. (5.12) is defined with discrete value in the numerical performance database. The element called *Altitudes* is a vector containing the cruise altitude per waypoint. $Climb_{IAS}$ and $Descent_{IAS}$ are the scheduled IAS values for climb and descent regimes. The main goal of the algorithm developed in this paper is to find the most economical combination of speeds and altitudes that reduces the flight cost. The problem can be treated as a *combinatorial optimization problem*, which can be defined as follows:

$$\textbf{\textit{Minimize}}\ Total\ Flight\ Cost\ (Climb_{IAS}, Mach, Altitude, Descent_{IAS}) \quad (5.13)$$

$$Climb_{IAS}\ \in\ IAS_{CLIMB\ MODEL} \quad (5.14)$$

$$Mach \in Mach_{MODEL} \quad (5.15)$$

$$Altitudes\ \in Altitude_{MODEL} \quad (5.16)$$

$$Descent_{IAS}\ \in IAS_{DESCENT\ MODEL} \quad (5.17)$$

The discrete design variables defined in Eq. (5.13) to Eq. (5.17) can be modeled in the form of a search tree-graph, as shown in Figure 5.2, where nodes at Level 1 within the tree-graph are created with $Altitude_{MODEL}$, nodes at Level 2 are created with $Mach_{MODEL}$, nodes at the Level 3 are created with $IAS_{CLIMB\ MODEL}$ and nodes at the last level are created with $IAS_{DESCENT\ MODEL}$, where *MODEL* refers to the values available in the numerical performance model.

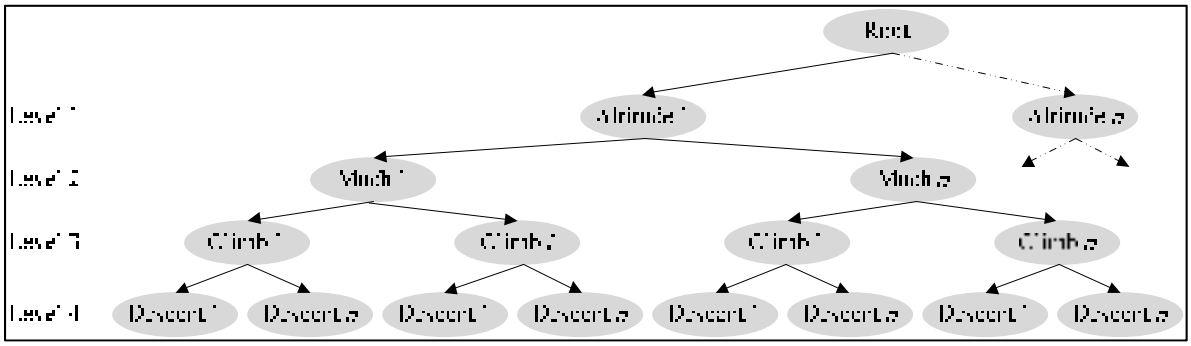

Figure 5.2 Tree-Graph for the available combinations

The solution thus selects one node per level, thus one Mach number, one IASclimb, one IASdescent, and Altitude.

5.3 The Optimization Algorithm

The new optimization algorithm proposed in this paper is based mainly on the *Beam Search* optimization technique, mixed with an initial search space reduction technique. This optimization algorithm consists of three different modules executed one after the other. The first module is the Search Space Reduction Algorithm Module (SSRAM), followed by the Beam Search Optimization Algorithm Module (BSOAM), and finally the Real Cost Computation Module (RCCM). The SSRAM reduces the total search space by helping the BSOAM to quickly converge to the solution, and then the RCCM provides an accurate flight cost.

Combining the SSRMA with the BSOAM brings the advantage of reducing the computation time required to converge to the most economical (optimal) solution. The SSRMA and the BSOAM are described in the following sub-sections. However, a brief overview of the algorithm and some of its generalities are described next.

For long haul flight, the most expensive flight phase is the cruise. The variables that have the highest influence in the cruise phase are *Altitude* and *Mach*. For this reason, in Figure 5.2 the first two levels are composed of the available *Altitude* and *Mach* number nodes. Discarding nodes at these levels, as a large number of combinations can be discarded, would be advantageous to reduce the calculation load. This is the task that the SSRMA accomplishes: it estimates an optimal cruise candidate solution and keeps it only those nodes *close* to this candidate solution. After executing this module, the search space can be reduced up to 50% of its total size.

The tree-graph is then constructed with the remaining nodes. The BSOAM evaluates the remaining nodes to determine if a given nodes combination is worth computing, or if it is better to discard it. Discarding nodes is desirable as it reduces the computation time. Within the BSOAM module, the RCCM module is executed as many times as required to compute the real flight cost.

The algorithm concept is shown below in Figure 5.3. First the search space is reduced with the SSRM where some nodes are rejected (nodes in the left and right extremes), as a consequence the search space is reduced. Then, the BSOA is executed cutting other nodes (dark black shadow rectangles in the middle of the figure). Finally, the optimal solution is shown in black. The RCCM module is executed only at the last level tree (also called leaves).

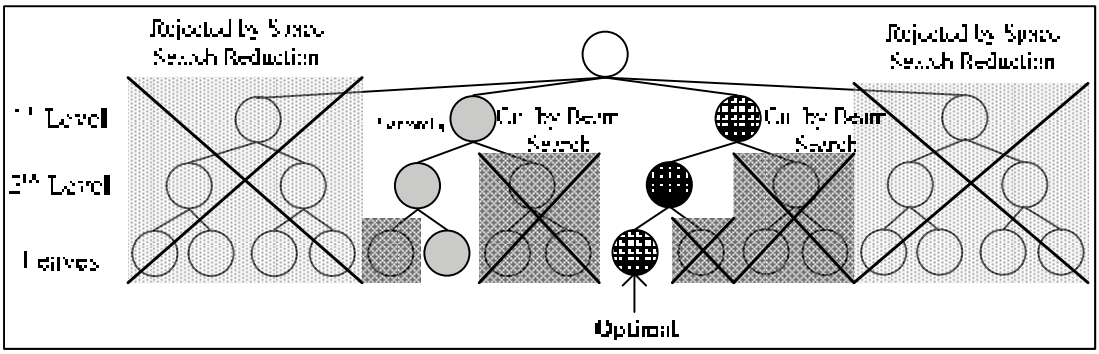

Figure 5.3 The SSRM and the BSOAM effects reducing the combinations.

The algorithm overview is presented in Figure 5.4. Each module is explained in the following paragraphs.

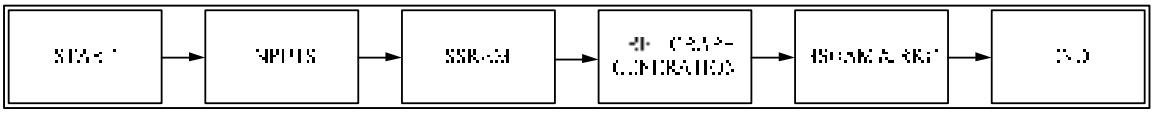

Figure 5.4 Algorithm Modules Execution Order

5.3.1 Algorithm's Input

The algorithm requires as inputs the airport departure and arrival coordinates, the aircraft total weight, a given ground track trajectory, the cost index, and the weather information for the search space. Once these parameters have been provided, the Search Space Reduction (SSRM) module can be executed.

5.3.2 Search Space Reduction Module (SSRM)

The SSRM is based on the algorithm detailed in (Murrieta-Mendoza & Botez, 2014b). The objective of this module is to find a candidate *Mach*/*Altitude* pair that allows the search space to be reduced. This module computes the cruise cost estimation for every available *Mach number*/*altitude* pair for the cruise phase. This estimation is achieved by executing a fast, but inexact cruise cost computation for each combination of *Mach*/*Altitude* available in the numerical performance model. The *Mach*/*Altitude* combination that provides the least expensive cost is defined as the *optimal candidate*. Only those *Mach* numbers and *Altitudes* around the *optimal candidate* are considered in this new search space.

The steps followed to compute this *optimal candidate* are given next. However, it is important to mention that this methodology focuses on medium and long haul flights. If short flights are of interest, modifications should be made to give importance to the climb and the descent phases. Contrary to long haul flights where climb and descent represent only a small percentage of the flight cost, for short flights, these regimes costs could represent over 70% of the flight cost.

Using Figure 5.5 as a guide to all the required parameters, the procedure to find the optimal Mach/Altitude candidate for long and medium haul flights requires the computation of a generic reference distance (D_{Cruise}) and of an estimated weight at the ToC.

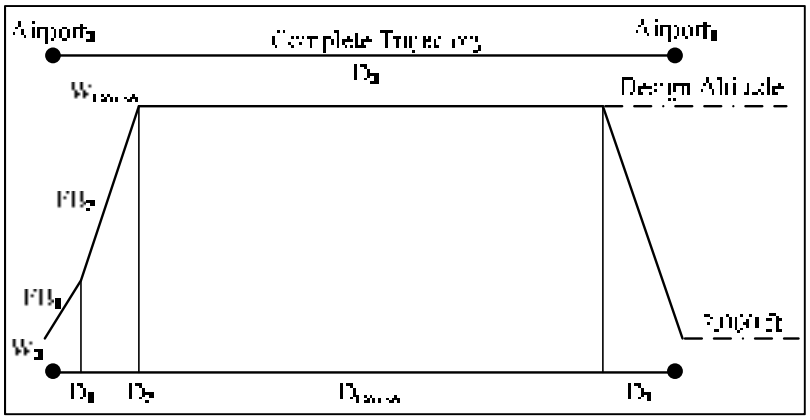

Figure 5.5 Space Reduction Parameters Definition

The process is as follows:

1) The great circle (or geodesic) distance (D_0) between the departing airport ($Airport_0$) and the destination airport ($Airport_1$) is computed.

2) The aircraft takeoff weight (W_0), the fuel burn (FB_1) and the horizontal traveled distance (D_1) required to climb to 10,000 ft at 250 IAS are computed from the numerical performance model Eq.(5.2) - Eq.(5.11).

3) The designated cruise Mach number and altitude for the aircraft of interest should be determined. This information can be found in the Aircraft Flight Manual. If this information is not available, an arbitrary (but coherent) value can be selected.

4) Using the Mach number above obtained, the fuel consumption (FB_2) and the distance traveled (D_2) to arrive at the designated altitude (or the ToC) are obtained directly from the numerical performance model as only an approximation is needed.

5) For the descent phase, as the cruise phase is finished, weight is no longer an important factor. The only parameter of interest is the required distance Distance (D_3) from the ToD to the destination Airport B. This distance is obtained directly from the numerical performance model, assuming that the ToD is at the design cruise altitude. The maximum descent aircraft weight, available in the numerical performance model, and a maximal IAS of 250 kts are taken into account.

6) Using the distance parameter obtained in steps 1 to 5, the estimated D_{cruise} can be computed as shown in Eq. (5.18).

$$D_{cruise} = D_0 - D_1 - D_2 - D_3 \quad (5.18)$$

7) Similarly, using W_0 and the fuel consumption obtained from steps 1 to 4, the W_{cruise} is determined as shown in Eq. (5.19).

$$W_{cruise} = W_0 - FB_1 - FB_2 \quad (5.19)$$

8) Using the W_{cruise} and the D_{cruise} parameters, the cruise cost estimation for every *Mach*/*Altitude* pair available in the numerical performance model is computed with Eq. (5.2) - Eq.(5.11). The distance D_{cruise} is divided into *n* equidistant waypoints (such a 8 for a long haul flight) assuming no step climbs. The number of waypoints is selected to allow a fast computation.

9) The most economical Mach/Altitude pair is identified as the optimal candidate.

Next, the tree-graph is created using the two higher Mach numbers and the two Mach numbers below the optimal Mach number, as well as the two higher altitudes and the two lower altitudes available in the numerical performance database. Two speeds and two altitudes were selected as it is considered that they cover the search space where the optimal solution could be found. This reduces the search space to a maximum of 5 Mach numbers and 5 altitudes for the cruise phase.

5.3.3 The Vertical Reference Trajectory Search Space: The Graph Construction

The tree-graph is created as nodes are expanded. In other words, the whole tree graph is not completely constructed, but is constructed level by level as nodes are expanded. The following hypotheses are taken into account for the graph construction:

- Flight begins at 2,000 ft at or below 250 IAS, so that different Standard Instrument Departure (SID) available for different airports will not be considered.
- Acceleration is allowed only when the aircraft is at 10,000 ft.
- Mach number in climb and descent phases is constant, as it is the same as the Mach number in the cruise phase.

- Step climbs are possible; however they are not shown in the graph defining the search space (see Figure 5.2). Altitudes at the graph represent only the ToC altitude.
- Deceleration is allowed only once during descent.
- IAS must be at or below 250 IAS when the aircraft reaches 10,000 ft.
- The flight ends at 2,000 ft and at 250 IAS. In the same way as in climb, the Standard Terminal Arrival Route (STAR) is not considered as it changes depending on the airport, runaway, and the instruments on board the aircraft.

The nodes at the first level are called $Nodes_{ALT}$, the descendant nodes at the second level are called $Nodes_{Mach}$, the descendant nodes at the third level are called $NodesClimb_{IAS}$, and the last level nodes, called *leaves*, are the $NodesClimb_{DES}$ as shown in Figure 5.2.

5.3.4 The Beam Search Algorithm

The Beam Search algorithm is a variation of the Branch and Bound algorithm. In both approaches, the main idea is to determine a full cost associated with the visited node without knowing all the required parameters to compute the whole flight. An early version of the Beam Search algorithm was defined in (Murrieta-Mendoza, Beuze, Ternisien, & Botez, 2015), and was used as base for this module.

The cost associated to each node is called a *bound*. Since the algorithm is searching to find the most economical combination of nodes, the *bound* is an estimation of the most economical combination of nodes that can be obtained by considering all of the node's descendants. In other words, an estimated cost is computed using an heuristic by considering all the tree levels, even when those levels (and nodes) have not yet been visited. This bound is calculated using what is called a *bounding function*. Since the algorithm's objective is to reduce the flight cost, the lower the bound is, the more promising the node is.

A node is considered to be *promising* when the current *Reference Cost* (most economical *real* cost so far) is higher than the *Bound*. This means that there might be a combination of the

node's descendants that provides a more economical flight cost than the current reference cost. For this reason, the node should be expanded for its evaluation at the higher levels. (see Figure 5.2). On the other hand, when the B*ound* is higher than the *Reference Cost*, it means that expanding the node would not provide a flight cost more economical than the current one. Thus the node is cut along with all its descendants. This means that the connecting nodes along with their children are discarded. Cutting nodes reduces the number of available combinations to compute, and thus the computation time. It is desirable to cut the highest number of nodes as possible in order to obtain the optimal solution with the minimal number of nodes and leaves to be evaluated. However, it is important to avoid cutting potential nodes, as this cutting could lead to sub-optimal solutions. For this reason, it is important to develop a bounding function with a compromise between accuracy and computation time.

5.3.5 The Bounding Function: Heuristics

This bounding function is the most important part of the BSOA module; it should provide an "optimistic bound" to find the global optimal, however, this function should allow the "cutting" of branches to reduce the search space, and thus to reduce the required computations. An "Optimistic bound" means that the computed bound should be less than the real flight cost.

The numerical performance database provides the information to estimate the fuel burn for each flight phase. Depending on the position of the node within the tree-graph, the parameters that define Eq. (5.12) are available. Two different options can be observed: only some of the parameters in Eq. (5.12) are available, or all of them are available. In the first case, when only some parameters are known, the behavior of the bounding function changes depending on them. In the second case, when all parameters are known, the RCCM can be executed.

In the first case, when only some of the parameters values in Eq. (5.12) are known, the bounding function changes as it approaches the leaves. This function becomes less optimistic

since more values from Eq. (5.12) are known, which allows a higher number of nodes and branches to be cut.

The first level, as shown in Figure 5.2, is defined by the discrete available "altitudes". When visiting a node in the first tree-graph level, the "Altitude" (initial cruise altitude) of the solution in Eq. (5.12) is the only one known. All values in the numerical performance model related to that cruise altitude are taken into account. All values related to other altitudes are discarded.

The optimism coefficient (C_{opt}) is used to influence the optimism level in the bounding function computation. C_{opt} can take any value from 0 and 1. When its value is zero, it corresponds to a pessimistic bounding function by taking the C_{opt} into account, the *bound* for a given node where unknowns are available can be computed with Eq. (5.20), as follows.

$$Bound = Database_{Average} - C_{opt} * (Database_{Average} - Database_{min}) \tag{5.20}$$

where *Bound* is the estimated cost of the node, $Database_{min}$ is the minimal cost that the numerical performance database provides, C_{opt} is the optimism coefficient and $Database_{Average}$ is the mean of the maximal and the minimal costs found in the numerical performance database by taking into account the given node restrictions.

After expanding the nodes at the first level "Altitude", the descendent nodes located at the second level are composed by the available Mach numbers. The Mach number knowledge further limits the considered values provided by the numerical performance database, resulting in a more accurate (and less optimistic) bound.

The descent and climb costs estimations are obtained by using Eq. (5.20) for these regimes. At the second level, the 2 cruise parameters, Mach and Altitude, are known.

Cruise step climbs are also considered. However, it is desirable to not enlarge the tree-graph, and thus the maximum number of 2,000 ft step climbs that the aircraft can compute by considering the distance, the TOC, and the maximum altitude are calculated. The fuel flow is obtained from the numerical performance database using Eq. (5.21).

$$FF_{cr} = {}^{1}/_{2} * [FF(M, Alt_0) + FF(M, Alt_1)] \quad (5.21)$$

where FF_{cr} is the fuel flow for cruise, FF (M, Alt_0) is the fuel flow at the first altitude (Alt_0) for a given Mach number, and FF (M, Alt_1) is the fuel at the last available altitude for the same Mach number.

At the third level where the Climb IAS is known, the accuracy of the *Bound* is improved. This improvement takes place because of the fact that the knowledge of the Climb IAS, the Mach number and the initial altitude allows the crossover altitude (the altitude where speed change from IAS to Mach number) to be known. A better climb estimation allows a better cruise estimation.

The last level is composed of the Descent IAS. It is only at this level where the Real Cost Computation Module (RCCM) can be executed. However, before executing the RCCM, a last bound is computed in order to (further) reduce the computation time. At this level, a different C_{opt} called C_{opt2} is used, as the descent has only a small effect on the total flight cost and thus a "pessimistic coefficient" able to cut many nodes. This pessimistic coefficient will be more aggressive at cutting nodes, thus reducing the times the RCCM is executed.

5.3.6 Weather presence in the bounding function

Wind and temperature have a significant influence on flight costs. It is important to introduce these parameters' influence into the bounding function in order to compute its realistic

bounds. Using the weather provided by Environment Canada would result in a time consuming alternative of flight costs computations.

As a general rule, it can be stated that higher winds are found at higher altitudes. Beginning at the initial altitude in the search space, a linear function is used to simulate the wind, as shown in Eq. (5.22).

$$Estimated\ Wind\ Speed = 1.5(kts) + 0.5\left(\frac{kts}{ft}\right) \times \Delta Altitude(ft) \qquad (5.22)$$

This function provides an optimistic wind speed value that gives priority to higher altitudes as wind would be stronger at high altitudes. Wind is considered to be aligned with the aircraft's direction. Temperature is considered to be at the standard atmosphere.

5.4 Results

The testing aircraft has 2 turbofans with a certified ceiling altitude of 39,000 ft, a maximal Mach number of 0.82 and a maximal takeoff weight of 160 tons.

5.4.1 Standalone Algorithms Results Comparison

A set of tests was performed in order to observe the behavior and analyzed the results of the developed algorithm. The Search Space Reduction + Beam Search Optimization algorithm (*BS + SP Red*) results were compared to those of the two algorithms that compose it, the Search space reduction and the Beam Search optimization, and the Exhaustive Search algorithm. The computation time and the fuel savings for all four algorithms were noted and further compared. The Exhaustive Search algorithm was used as the algorithm of reference.

The first set of results cover the algorithm execution time versus the flight distance. Seven different flights were executed to observe the time the exhaustive algorithm took to provide the optimal solution. These tests were executed with a CI = 0; the seven trajectories were Montreal (YUL) – Regina (YQR), Los Angeles (LAX) – Minneapolis (MSP), Vancouver

(YVR) -- Detroit (DTW), Montreal (YUL) – Vancouver (YVR), Montreal (YUL) – Los Angeles (LAX), Montreal (YUL) – London (LHR), and Martinique (FDF) – Paris (ORY). The distances between airports, and the execution time computed for an Exhaustive Search algorithm for all 7 destinations are shown on Figure 5.6.

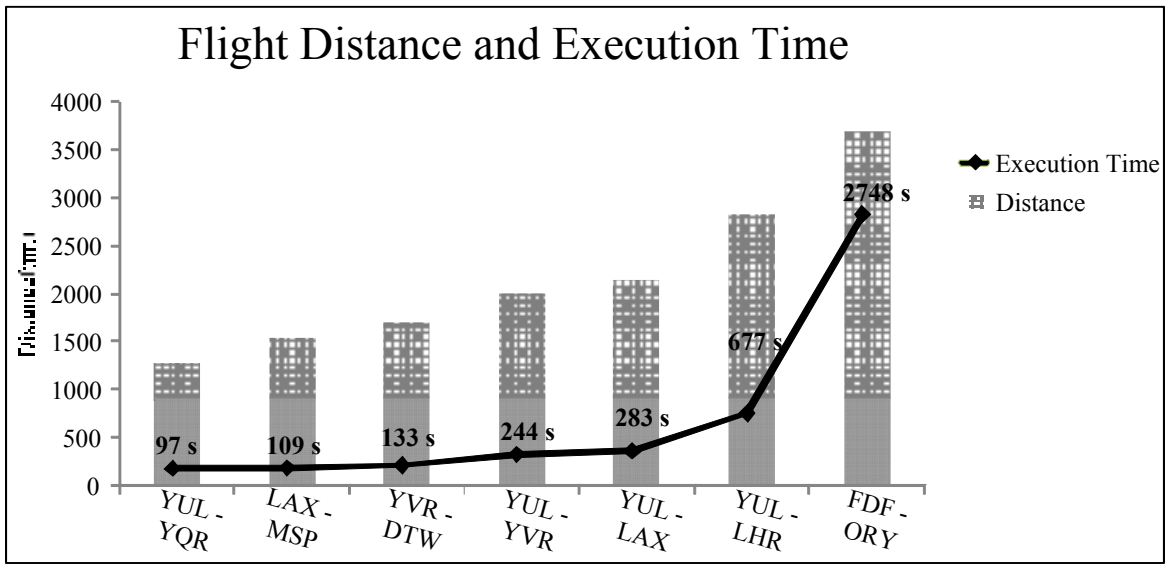

Figure 5.6 Algorithm Execution Time as a Function of Flight Distance

It can be observed that the longer is the flight distance, the longer is the computation time. This observation is evident, as more interpolations are required to compute the flight cost of a longer flight.

The flight trajectories were further optimized with the standalone versions of *Search Space Reduction* (*Sp Red*), the standalone *Beam Search* (*BS*), the *Exhaustive Search* and the algorithm developed in this paper (*BS + Sp Red*). For the algorithm developed here and for the stand alone *Beam Search*, the C_{opt} values were selected to be $C_{opt1} = 0.65$ and $C_{opt2} = 0.1$. Previous research suggests that these values provide a good compromise between the computation time and the quality of the solution (Murrieta-Mendoza, Beuze, et al., 2015).

It was observed that the standalone Beam Search and the Sp Search Reduction algorithms always found the optimal solution for the analyzed flights. The algorithm developed in this

paper was not always able to find the optimal solution. This occurred for three flights out of the 7 flight tests: YUL – LHR, YUL – LAX, and YUL – YVR as seen in Figure 5.7.

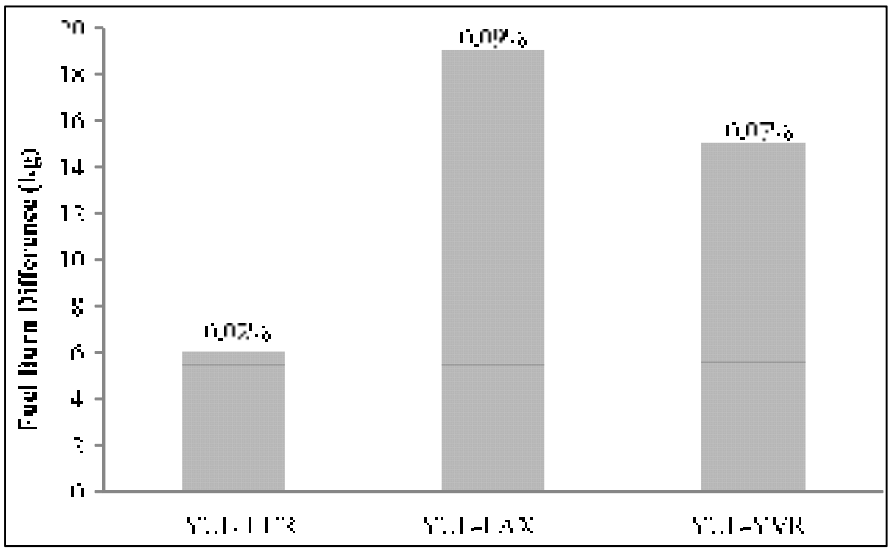

Figure 5.7 Fuel Burn Difference Between the Optimal Trajectory and the Trajectory Found by the Algorithm

The fuel consumption difference (in Kg) between the solution provided by the developed algorithm and the fuel consumption from the global optimal solution found with the exhaustive search algorithm (shown in Figure 5.7) can be considered negligible as these differences are really small, as these differences proved to be less than 0.1% of the optimal total flight cost. This slight difference in the optimal trajectory found by the new algorithm can be attributed to the search space reduction performed by the BOSA, which led the algorithm to cut the optimal node when the BSOAM module was executed. This cutting occurred because the C_{opt} calculated by the bounding function was not as optimistic as it should have been to allow the finding of the optimal solution. However, changing the C_{opt} to a more optimistic value may lead to an increase in the execution time.

In addition to the developed algorithm finding optimal or really good sub-optimal solutions, an interesting improvement in execution time reduction can be seen in Figure 5.8, which shows that the developed algorithm provided its solution more quickly than the other methods. The execution time was recorded for four different algorithms: the exhaustive

search, the Space search standalone algorithm, the Beam Search standalone algorithm, and the developed algorithm (BS + Sp Red). The execution time for the exhaustive search, being the longest one, was selected as the reference time, and then the ratio for the other three algorithms was computed to compare their execution times. For example, a time ratio of 5 in Figure 5.8 means that the algorithm was 5 times faster than the exhaustive search.

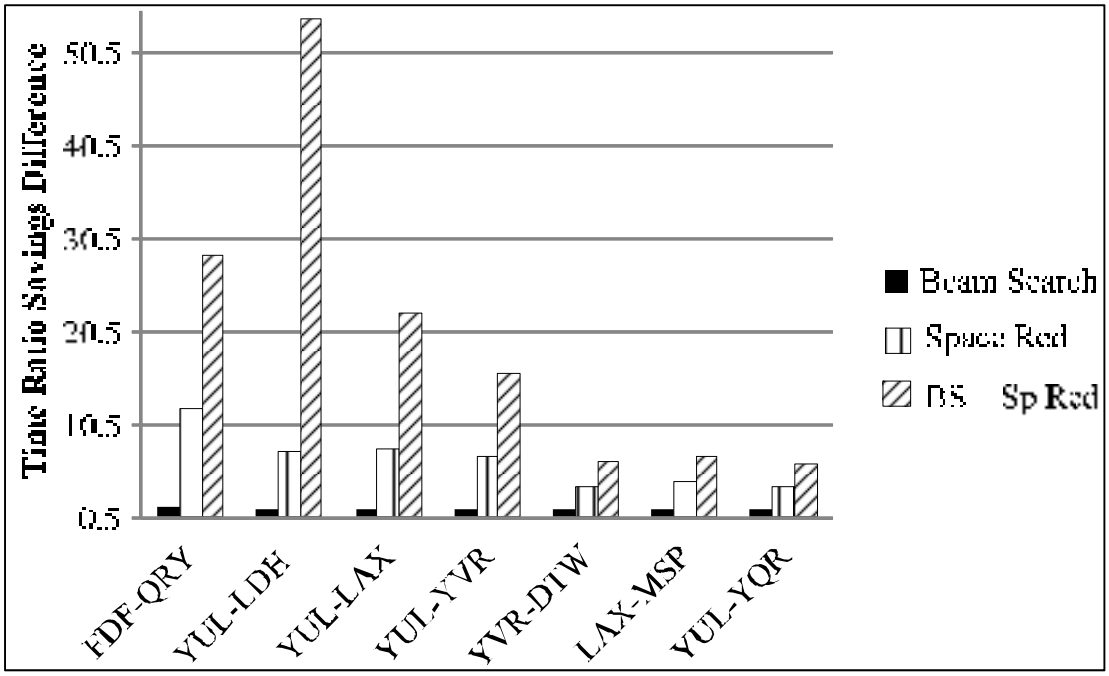

Figure 5.8 Execution time ratio comparison between three different optimization algorithms using the exhaustive search method's time as reference.

Results show that for the FDF-QRY flight, the standalone Beam Search algorithm was 1.5 times faster than the exhaustive search algorithm, while the space search reduction algorithm was around 12 times faster than the exhaustive search algorithm. The proposed algorithm (BS + Sp) was able to provide a solution 28 times faster than the exhaustive search algorithm. Figure 5.8 also shows that for all flights, the proposed algorithm was always faster, to values as up to over 50 times faster than the exhaustive search algorithm. It attains this large gain because the algorithm takes advantage of two different methodologies to reduce the computation time while providing good flight trajectories in terms of fuel reduction.

Comparing the three flights in Figure 5.7, i.e., those that did not provide the optimal solution against the execution time in Figure 5.8 shows that for flight YUL-LDH, the algorithm was almost 52 times faster than the exhaustive search method (and almost 44 times faster than the second-fastest algorithm). This considerable time savings came at a cost of only 0.02% of difference from the global optimal. A similar case can be seen for the YUL – LAX and YUL – YVR flights, where the algorithm was 22 and 15 times faster, respectively, than the exhaustive search, and close to 15 and 8 times faster than the second-fastest algorithm. This time savings came at a cost of only 0.09% and 0.07%, relatively to the global optimal (reference) flight cost.

These results are especially interesting when algorithms are developed for low computation power devices such as the FMS. Although it would be desirable, it was not possible to perform a comparative study using the reference FMS. Due to confidentially reasons, the FMS computation time and the loaded algorithm were not provided. Nevertheless, the optimal flight trajectory can be evaluated compared to those produced by a commercial FMS.

5.4.2 The reference trajectory flight cost between the trajectory provided by a FMS and the trajectory of the developed algorithm

The previous section showed how the developed algorithm was able to provide optimal (or near optimal) solutions compared to solutions provided by an exhaustive search algorithm and two other algorithms. It also showed that the developed algorithm was able to provide a good solution in a much quicker timeframe than the other algorithms. However, it is of great interest to compare the flight cost of the computed trajectories to those of the trajectories proposed by a commercially certified FMS. A commercial FMS benchmark called Past Task Trainer (PTT) was used to perform this comparison. The PTT uses the same numerical performance model as the one used by the developed algorithms. The PTT also requires the same inputs as the algorithm presented in this paper. Once the PTT finishes optimizing the trajectory, it provides an optimized vertical reference trajectory to follow. The same flight

with the same initial conditions was optimized with the developed algorithm, and the flight costs of both algorithms were compared to observe the differences between them.

The flight results for the YUL – YVR trajectory are shown in Figure 5.9, where the trajectory was optimized with the PTT and the developed algorithm for different cost indices from 0 to 100. This trajectory was selected due to the availably of data.

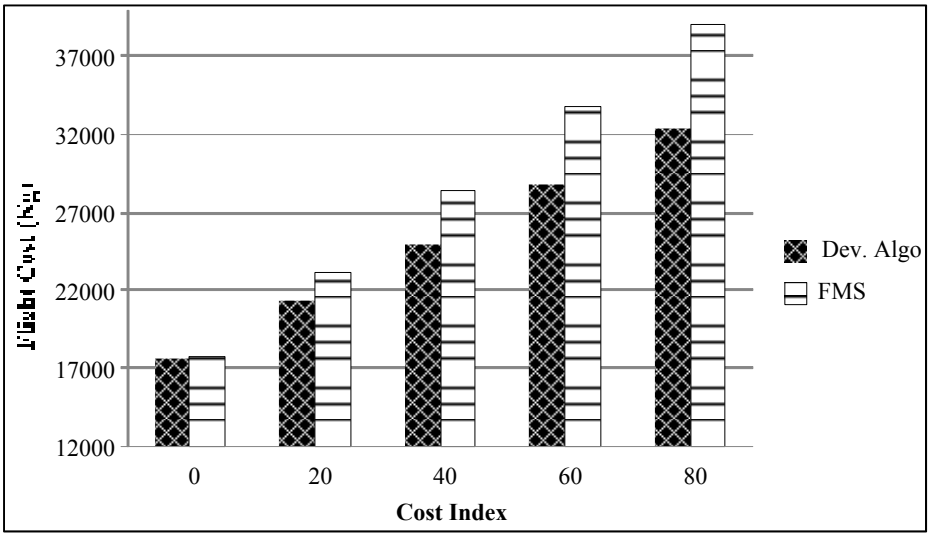

Figure 5.9 Flight Cost Between the FMS and the Developed Algorithm for the YUL – YVR flight

For every case, the developed algorithm was able to provide better results than the FMS of reference. This is explained as the algorithm was able to find a better vertical reference trajectory.

5.4.3 The reference trajectory flight of the developed algorithm compared to a real lateral trajectory.

The algorithm was used to optimize a trajectory following a real lateral trajectory. This test was performed to confirm that the algorithm was able to find the optimal trajectory for a route that does not follow the geodesic trajectory. The exhaustive search trajectory was

computed following the lateral reference trajectory from a Vancouver to Cancun flight taken from *flightaware* ®. The take-off weight was 130,000 kg. The lateral reference trajectory is shown in Figure 5.10.

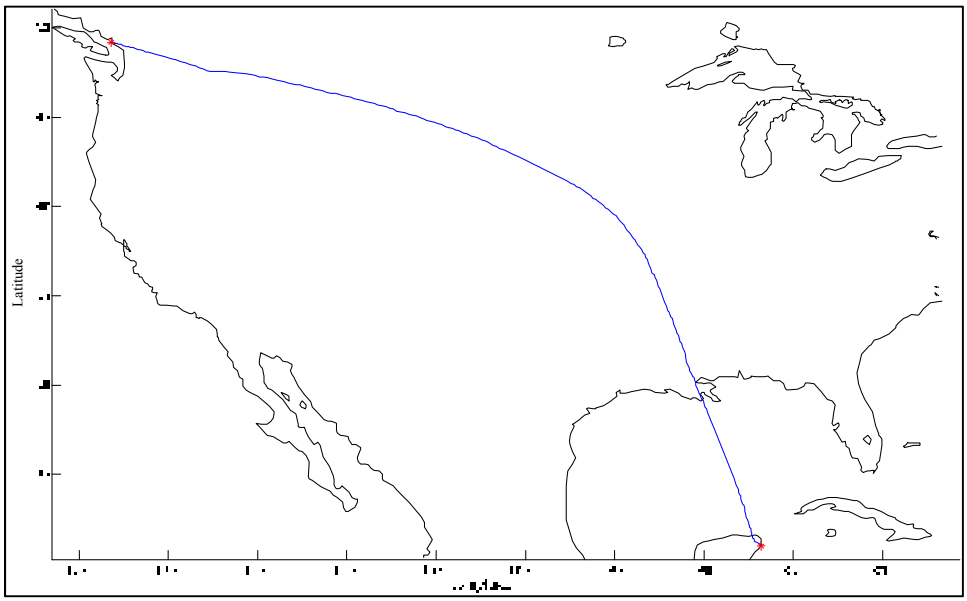

Figure 5.10 Lateral reference trajectory for a Vancouver to Cancun flight

The solution found by the Exhaustive Search algorithm and the solution found by the developed algorithm resulted in the same total fuel burn of around 22,550 kg. Their identical vertical reference trajectories are shown in Figure 5.11.

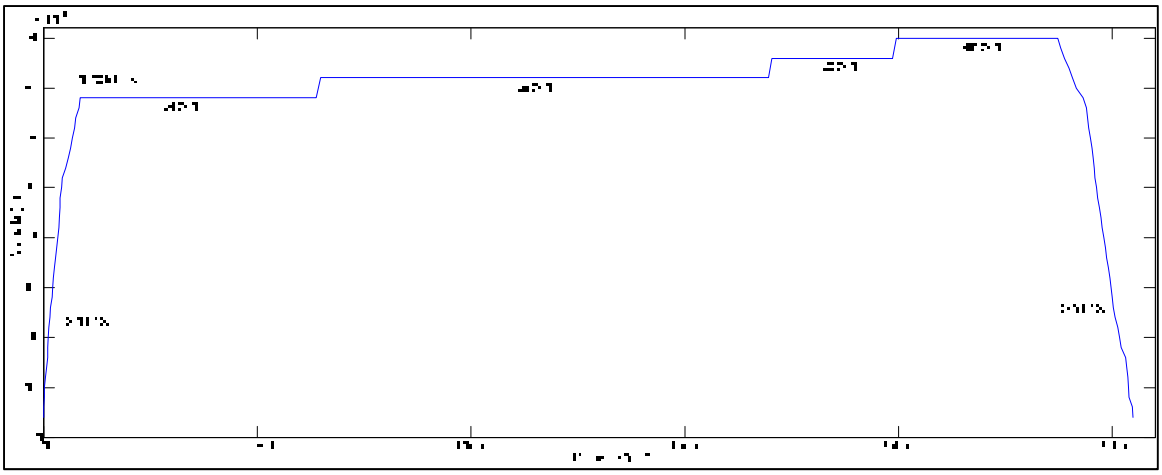

Figure 5.11 Optimal Vertical Reference Trajectory Provided by the Exhaustive Search Algorithm and the Developed Algorithm.

5.5 Conclusion

An optimization algorithm to reduce fuel burn and/or flight cost was developed. This algorithm took into account Air Traffic Management constraints such as constant speeds and constant altitude segments. The possibility of performing altitude changes during the cruise phase was explored to obtain a reduction in the flight cost. To compute the flight cost, both fuel consumption reduction and flight time costs were considered, using the Cost Index. This new algorithm originality is that it uses a numerical performance model with experimental flight data instead of the equations of motion normally used in the literature.

The developed algorithm is also original as it combined two different methodologies, the Search Space Reduction algorithm, which reduces the search space, and the Beam Search algorithm. It was shown that the new developed algorithm always found the optimal or (much more quickly found) a very good sub-optimal solution for several different long haul flights. The new algorithm considerably reduced the execution time compared to that of other algorithms such as the stand-alone Space Reduction, the stand-alone Beam Search, and the Exhaustive Search algorithms. An analysis of the results obtained with the new algorithm with the results provided using a commercial FMS proved that the algorithm developed here provided better optimized trajectories than the commercial FMS.

This new algorithm makes it possible to reduce fuel consumption, which is a very important achievement due to the pollution released to the atmosphere, and to fuel-related costs. It also significantly reduces flight time, which gives valuable savings in terms of maintenance, customer satisfaction, crew salaries and other factors.

CHAPTER 6

4D AIRCRAFT EN-ROUTE OPTIMIZATION ALGORITHM USING THE ARTIFICIAL BEE COLONY

Alejandro Murrieta-Mendoza, Audric Bunel, Ruxandra Mihaela Botez

École de Technologie Supérieure / Université du Québec, Montreal, Canada
Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticty

This article was submitted for publication to the AIAA Journal of Aerospace Information Systems on December 2016

Résumé

La combustion de carburant libère des particules polluantes dans l'atmosphère. Il a été estimé que les avions sont responsables de 2% du total de dioxyde de carbone libéré dans l'atmosphère chaque année. Le carburant est également l'une des dépenses les plus importantes pour les compagnies aériennes. Réduire la quantité requise de carburant des avantages aux aspects environnementaux et économiques de l'industrie aéronautique. L'objectif de ce chapitre est de développer un algorithme d'optimisation qui calcule la trajectoire de référence de l'avion. La trajectoire de vol est optimisée en termes de vitesses, d'altitudes et de positions géographiques, tout en respectant la contrainte de l`heure requise d'arrivée. Les vents et les températures sont pris en compte. La trajectoire optimale sera composée des points de cheminement placés dans chacune des dimensions disponibles (coordonnées, altitudes et vitesses). Ces trajectoires seront améliorées en tenant compte de toutes les dimensions simultanément, au lieu de les améliorer l'une après l'autre. La trajectoire optimale a été trouvée en mettant en œuvre l'algorithme d'optimisation des colonies d'abeilles. La recherche de section d'or a été utilisée pour trouver le premier nombre de référence de Mach qui remplit la contrainte de l'heure d'arrivée requise. Les résultats ont montré que l'algorithme économise environ 5% du coût du vol en comparaison des vols théoriques et aériens.

Abstract

Fuel burn releases polluting particles to the atmosphere. It has been estimated that aircraft are responsible for 2% of the total of carbon dioxide liberated to the atmosphere each year. Fuel is also one of the most important expenses for airlines. Reducing the required amount of fuel to power flights brings benefits to the environmental and economic aspects of the aeronautical industry. The objective of this paper is to develop an optimization algorithm that computes fuel efficient aircraft's reference trajectory. The flight trajectory is optimized in terms of speeds, altitudes and geographical positions, while respecting the Required Time of Arrival constraint. Winds and temperatures are taken into account. The optimal trajectory will be composed of waypoints placed in each of the available dimensions (coordinates, altitudes and speeds). These trajectories will be improved by taking into consideration all dimensions simultaneously, instead of improving them one after the other. The optimal trajectory was found by implementing the Artificial Bee Colony optimization algorithm. The Golden Section Search was used to find the reference Mach number that fulfills the Required Time of Arrival constraint. Results have shown that the algorithm saved around 5% of the flight cost comparing against theoretical and as flown flights.

6.1 Introduction

The release of polluting particles to the atmosphere has been of special interest to many different industries, mostly because of their effect on the environment such as global warming and ozone layer depletion (Crutzen, 1970). The aeronautical industry requires large quantities of fuel to sustain its operations; in 2010, it contributed to 2% of the global carbon dioxide (CO_2) released to the atmosphere (ICAO, 2010). Not only did this pollution affect the environment (in terms of climate change), but other components such as hydrocarbons were also released as polluting particles. Studies have showed that hydrocarbons directly affected public health as short exposure caused discomfort in the form of headache, nausea, drowsiness, and irritation and longer exposure caused immune system and nervous system damage. Noise is another kind of pollution that is of special importance during the terminal flight phases (takeoff, initial climb, and landing) when aircraft are close to population areas.

Noise has been linked to stress, nervous system and cardiovascular diseases (Black et al., 2007).

Despite of these negative factors related to fuel consumption, the aeronautical industry has an important impact on the world economy and development (ATAG, 2014). At the same time, traveling by air is often the most practical way of traveling, especially for long distances and/or to remote areas. The aeronautical industry has set itself a goal of reducing its CO_2 emissions by the year 2020 to half of that recorded in 2005.

Each generation of aircraft takes advantage from new technologies to become more efficient. Materials have been a key factor due to their direct impact on aircraft weight; new alloys and composites have replaced heavier materials. Heavy analogical equipment has been replaced by lighter digital equipment. Bulky equipment has been upgraded to become slimmer, lighter, and more effective. Multiple strategies have been employed to increase engine efficiency, including measuring the vibration of the engine's shaft and reducing the wear (Salvat, Batailly, & Legrand, 2013). Biofuel is another alternative to replace current fossil fuels (ATAG, 2009). Flight tests operating with biofuel have been successfully performed in Canada, Mexico, and Brazil.

In addition to the environmental impacts of burning fuel, there is also its cost factor. Airlines spend around 25% of their budget on fuel, thus it is motivating for them to reduce fuel consumption to improve their profits. Various measures have been implemented to reduce fuel consumption such as washing engines to improve performance, assisting transportation in the taxiway, and etcetera (McConnachie et al., 2013).

A promising operation researched by airlines is the flight plan optimization. Before take-off, the reference flight trajectory would be computed. This trajectory would provide an aircraft with the best flight conditions to reduce fuel consumption. While airborne, there is an avionics device that, among other tasks, might compute the optimal flight trajectory to be suggested to the pilot (Collinson, 2011). This device is the Flight Management System

(FMS), which normally operates with a low computation power processor that would limit its computation capacity.

Despite the efforts taken by airlines and FMSs to promote flight trajectories that reduce fuel consumption, studies in the United States and in Turkey have shown that aircraft do not fly at their optimal trajectories (Luke Jensen et al., 2014; Luke Jensen et al., 2013; Turgut et al., 2014). In order to minimize the flight cost, the flight fuel and time requirements to execute a given mission need to be optimized. To minimize the fuel consumption, the aircraft should fly following the altitudes and speeds that would reduce the flight cost while complying with regulations.

Because of its high exposure to centers of population, the descent phase was one of the first flight phases to be optimized. Thus, the Continuous Descent Approach/Operation (CDA/CDO) was developed. In this procedure, instead of descending in a step fashion (by constantly changing the engine settings), the aircraft performed the descent following a constant slope setting the engines to their lowest configuration: IDLE (Analysis, 1978). This operation has proven its effectiveness in many airports around the world (Clarke et al., 2013; Kwok-On et al., 2003; Novak et al., 2009).

In order to correctly implement CDA/CDO procedure, the aircraft's weight and its location of the Top of Descent (ToD) must be accurately computed (Johnson, 2011; Stell, 2010). The descent operation must be performed with precision in order to avoid to execute the missed approach (or go-around) procedure, which might be expensive both in terms of fuel consumption and pollution as reported in (R. Dancila et al., 2014; Murrieta-Mendoza, Botez, & Ford, 2016).

Another flight phase of interest in reducing fuel consumption is the cruise phase. During long haul flights, it is the flight phase that consumes more fuel. Different aircraft trajectory optimization algorithms, focusing primarily on this flight phase, have been developed with the aim to be implemented in either ground systems or in FMSs.

Many algorithms take advantage of the aircraft Equations of motion (EoM) to simulate the aircraft as a point-mass particle. To solve these equations, researchers have used optimal control and hybrid methodologies (Bonami, Olivares, Soler, & Staffetti, 2014; Franco & Rivas, 2011; Franco & Rivas, 2015; Soler-Arnedo et al., 2013; Valenzuela & Rivas, 2014). Other optimization algorithms such as dynamic programming, shortest path algorithms, and greedy heuristics have also been implemented to solve the EoM to determine the optimal trajectories (Cao, Rathinam, & Sun, 2013; Celis, Sethi, Zammit-Mangion, Singh, & Pilidis, 2014; Cobano et al., 2013; Dicheva & Bestaoui, 2014; Miyazawa et al., 2013; Rippel et al., 2005; Sadovsky, 2014; Sridhar et al., 2013). Although these algorithms have proven to be effective in finding economical reference trajectories, and even to avoid contrail formation zones, they had two drawbacks. The first drawback is that they were mostly based on classical deterministic techniques, which normally required high computational resources that, although available in ground systems, were not yet available in airborne devices. The second drawback was that they might require knowledge of aircraft aerodynamic parameters such as the lift coefficient, drag, and the thrust-specific fuel consumption (TSFC) coefficient.

These values were not easily to be found for different flight phases. There are databases available such as the Base of Aircraft DAta (BADA) or other information found in the literature.

An alternative of using the EoM is to use an aero-performance model called a performance numerical model. These performance numerical models are based on flight test, and they are normally used in the FMS to compute the flight cost. Different algorithms have been created using this type of model; deterministic and metaheuristic algorithms. Metaheuristic algorithms may provide very good results with less computation time than deterministic algorithms, nevertheless, they do not guarantee finding the global optimal solution.

Among the algorithms developed using a numerical performance model, a cruise cost estimation function was developed to compute the optimal cruise altitude (B. Dancila et al., 2013); however, this algorithm has not considered changes in altitudes and the computation

of the optimal Mach number. Dynamic Programming (DP) was implemented in (Sidibe & Botez, 2013) to find the optimal flight trajectories for all flight phases (climb, cruise, step-climbs, and descents); however it was considered too much time consuming. Reducing the search space to find the optimal trajectory was explored in (Jocelyn Gagné et al., 2013; Murrieta-Mendoza & Botez, 2014b) and very good results were obtained. A Branch and Bound variation was also used to reduce the search space and find the optimal trajectory (Murrieta-Mendoza, Beuze, et al., 2015; Alejandro Murrieta-Mendoza & Ruxandra Mihaela Botez, 2015). The Golden Section Search algorithm in short flights, and step climbs in long flights were implemented in (R.S. Felix Patron et al., 2013).

Dijkstra's, Bees' and Genetic Algorithms were implemented to find the winds that offered the optimal cruise trajectory during a fixed altitude flight; the optimization of these algorithms depended directly on the influences of winds (Roberto S. Félix-Patrón et al., 2014; Murrieta-Mendoza & Botez, 2014a). Genetic Algorithms were used to determine the best waypoints positions on the flight trajectories to perform step climbs compared to an airline's reference flight (Murrieta-Mendoza, Félix-Patrón, et al., 2015). Genetic Algorithms were improved to optimize the climb, the descent phases, and simultaneously the vertical and lateral dimensions during the cruise phase (Roberto S Félix-Patrón et al., 2014).

Most of the work mentioned above delivered 3D trajectories (position, altitude, and speed) that could be integrated in a flight plan. The flight plan would be submitted to Air Traffic Management (ATM), which would accept or modify the flight plan before take-off. While airborne, the trajectory can only be changed after having gained the Air Traffic Control (ATC) authorization.

In many parts of the world, the airspace is approaching its capacity limits, and for this reason, new procedures for ATM are being developed. In the case of NextGen in the United States, and SESARS in Europe, it will be possible for the aircraft to modify its trajectory in a time-based system. These trajectories are called TBO/IBO (Trajectory-Based Operations or Intended-Based Operations), and will take place in 4D (latitude, longitude, altitude, and time

dimensions). These 4D trajectories have the novelty that the aircraft will be required to arrive at a certain waypoint at given time; this time is called the Required Time of Arrival (RTA). Another advantage of these new systems is that the automatic negotiation of the aircraft systems with the central ATC systems will allow more aircraft to fly at their most optimal profile without compromising safety (Gardi, Sabatini, Ramsamy, et al., 2015). A number of different algorithms have been proposed for these new trajectories (Cobano et al., 2013; Hagelauer & Mora-Camino, 1998; Korn, Helmke, & Kuenz, 2006; Sam Liden, 1992a; Miyazawa et al., 2013; Soler-Arnedo et al., 2013). These algorithms have used the EoM to obtain the flight cost. Using a numerical performance model, the Ant Colony Optimization algorithm has been applied to find the combinations of Mach numbers that would reduce the flight cost of a transport aircraft to fulfill the RTA constraint for a fixed altitude flight (Murrieta-Mendoza, Hamy, et al., 2015).

The objective of this paper is to expose the methodology followed by a new developed flight optimization algorithm to find the "optimal" reference trajectory for a given flight. In this article, "optimal" refers to the trajectory that delivers the most economical flight in terms of fuel burn, while it respects the RTA constraint. The desired optimal reference trajectory will be obtained in the form of a 4D trajectory. Because of the fact that the algorithm should provide a trajectory in a relatively low computation time, the Artificial Bees Colony (ABC) metaheuristic algorithm was implemented to account for weather information. The ABC provides the optimal trajectory in terms of lateral and longitudinal coordinates, altitudes and speeds. The Golden Section Search (GSS) was employed to find the initial reference Mach number to compute the combination of speeds that fulfill the RTA constraints. The ABC would refine the Mach number to find the most economical one. This paper also proposes a dynamic grid that allows the algorithm to explore all the search space and not pre-defined waypoints as most of the algorithms in the literature.

The paper is arranged as follows: the methodology used to compute the flight cost is described, followed by the search space definition, and the use of the GSS to compute the

initial reference Mach number. The ABC theory and its implementation are then described. Finally, the results, conclusion, and future work are summarized.

6.2 Methodology

6.2.1 Flight Cost

The flight cost function is composed of the costs of two elements: the fuel required to perform the flight, and the required flight time. The relationship between these two variables and the flight cost are given in Eq. (6.1).

$$Flight\ Cost = Fuel\ Required + Cost\ Index * Flight\ Time \tag{6.1}$$

where *Cost Index* (*CI*) is a variable that translates the time in terms of fuel burn. The *CI* is defined by the airline, and is normally constant through the duration of flight. It is obvious that in order to minimize the flight cost, both variables (*Fuel Required*, and *Flight Time*) should be minimized.

6.2.1.1 The Numerical Performance Model

A numerical performance model, created using in-flight experimental performance data, was used to compute the fuel required to execute a given flight. This numerical performance model is composed of a group of tables which provides the fuel consumption for different flight phases. Because of the fact that this paper focuses on the cruise phase with the possibility of performing step climbs at different flight levels, only the information for the *cruise* phase, and the *climb* phase are used. The numerical performance model can be considered as a black box which receives "inputs" to provide the "required output". For the cruise phase, the required inputs are: the aircraft weight (kg), the speed (Mach number), the altitude (ft), and the international standard atmosphere temperature deviation, or ISA_DEV (°C). As an output, the numerical performance model provides information depending on the

flight phase. For the cruise phase, the information provided is the fuel flow (kg/hr). For the climb phase, the information provided is the fuel burn (kg) and the horizontal distance travelled (nm) required to climb or to descend to a given altitude.

Not all the exact inputs to obtain the desired output are available in the numerical performance model. Therefore, interpolations of the outputs using the available inputs are required to obtain the required outputs for the needed inputs. The interpolation process to compute the flight cost using a numerical performance model was discussed and detailed in (Alejandro Murrieta-Mendoza & Ruxandra M. Botez, 2015). A brief explanation of this procedure is defined next.

6.2.1.2 Fuel Burn Computation

The interpolations required to compute the fuel consumption are normally executed for the aircraft's weight (as the plane flies, the consumed fuel weight is reduced from the gross weight), the ISA_DEV (temperature changes along the flight), and the Mach number (a specific Mach number might be needed to fulfill the RTA constraint). The altitude in the numerical performance model is conveniently provided in multiples of 1,000 ft. in order to agree with traffic control regulations. Interpolations in altitude are never required.

The whole interpolation process during the cruise phase is managed differently depending if there is a *fixed altitude cruise* or a *change of altitude* (step-climb).

During a fixed altitude cruise, the distance to be flown is divided in 20 nm segments. At the beginning of each segment, three operations are executed:

1) the fuel burned from the last segment is reduced from the gross weight,
2) local temperature and wind information are used in order to compute the ISA_DEV and the ground speed, and,

3) interpolations in weight, temperature, and Mach number to compute the fuel flow are executed.

With the *Fuel Flow* obtained from the interpolations, and by computing the *Flight Time* required to travel the 20 nm segment at a given Mach number, the fuel burn necessary to fly that 20 nm segment can be computed as shown in Eq. (6.2).

$$Fuel\ Burn(kg) = Fuel\ Flow\left(\frac{kg}{hr}\right) \times Flight\ Time(h) \tag{6.2}$$

When a change in altitude during cruise is necessary (step-climb), the climb cost computation is performed per 1,000 ft steps directly after passing a given waypoint. To evaluate the fuel required for the aircraft to climb to the objective altitude, the flight condition is identified at the initial altitude. The numerical performance model in this algorithm does not provide the fuel to climb from one altitude to another altitude directly. Instead, it provides the fuel consumption to climb to a given altitude by taking the reference altitude at 0 ft. Interpolations are then required to compute the fuel consumption to climb at the *current altitude* and then to climb to the *next altitude* (*current altitude* + 1,000 ft). The fuel required to climb from the *initial altitude* to the *next altitude* is expressed by the fuel consumption difference required to climb between 0 ft to the *initial altitude,* and the fuel required to climb from 0 ft to the *next altitude*. The fuel difference obtained is then subtracted from the total aircraft weight. This process is repeated until the objective altitude is reached. The *horizontal travelled distance* is computed in the same way. Interpolations are required to compute the required horizontal traveled distance needed to climb to the *current altitude* and then to climb to the *next altitude*. The required horizontal traveled distance to climb from the *initial altitude* to the *next altitude* is expressed by the difference from the horizontal traveled distance required to climb between 0 ft to the *initial altitude,* and the horizontal traveled distance required to climb from 0 ft to the *next altitude*. The required horizontal traveled distance is subtracted from the next cruise segment. This process is repeated until the *objective altitude* is reached. After reaching the objective altitude at the end of each step climb, the fixed altitude cruise computations are executed as described before. The *Flight Time* for the climb phase is

computed using the horizontal travelled distance required to perform the desired climb provided by the numerical performance model at a given Mach number.

During the rest of the article, this process of computing the flight cost will be referred to as the *Real Trajectory Cost*. It provides also, in addition to the cost, the time when the aircraft is at every waypoint, as well as the aircraft speed and the fuel consumption for every flight segment.

6.2.1.3 Weather Information

The weather information is obtained using open source information from Weather Canada. This information is provided under the form of a grid spaced by 0.6 x 0.6 km, at different pressure altitudes and at different 3 hours time blocks. The parameters downloaded and used in the algorithm are the wind speeds, wind angles, altitude pressures and temperatures.

Knowing an aircraft's position expressed in terms of its geographical coordinates and its altitude, it is possible to compute the weather parameters. For these different interpolations in time, altitude and geographic coordinates (longitude and latitude) are required as detailed in (Murrieta-Mendoza, 2013).

In Figure 6.1, each node contains the required weather information. The aircraft is also located at a different flight level, at a different time, and at different coordinates. To obtain the weather at the exact aircraft's location, at the exact time, the next interpolation process is executed.

1) Interpolations in time are performed to obtain the four nodes' weather enclosing the aircraft at the *Real Time*.

2) Interpolations in *Flight Level* are performed for each time *Time*. This way, the weather for the four nodes enclosing the aircraft is computed for two different times at the *Real Flight Level*.

3) Bilinear interpolations are performed between the four nodes enclosing the aircraft to obtain the weather at the exact special location.

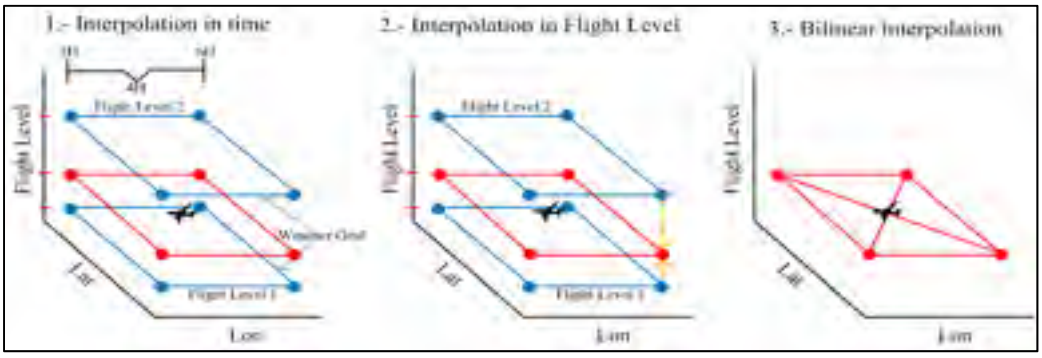

Figure 6.1 Weather Interpolations

6.2.2 The Studied Flight: The Search Space

The studied flight consists only of the *cruise phase*. It begins at the end of the initial climb, the Top of Climb (ToC), and ends at the beginning of the final descent, the Top of Descent (ToD). So the search space is defined only between those points (ToC and ToD). The search space for the algorithm is restricted to lateral, vertical and time-based boundaries. The search space and its boundaries are explained next for each such boundary or dimension.

6.2.2.1 Vertical Dimension and its Boundaries

The vertical search space boundary consists of minimal and maximal altitudes. The search space in the vertical dimension is a graph with pre-defined nodes at the available altitudes. In Figure 6.2, the red line is the maximal available altitude, and the green line is the minimal available altitude. The blue points are the nodes (waypoints) where a change of altitude (step climb) can be executed. The purple line is an hypothetic trajectory that began its cruise phase at 34,000 ft., executed 2 step climbs (of 2,000 ft each) to end at an altitude of 38,000 ft.

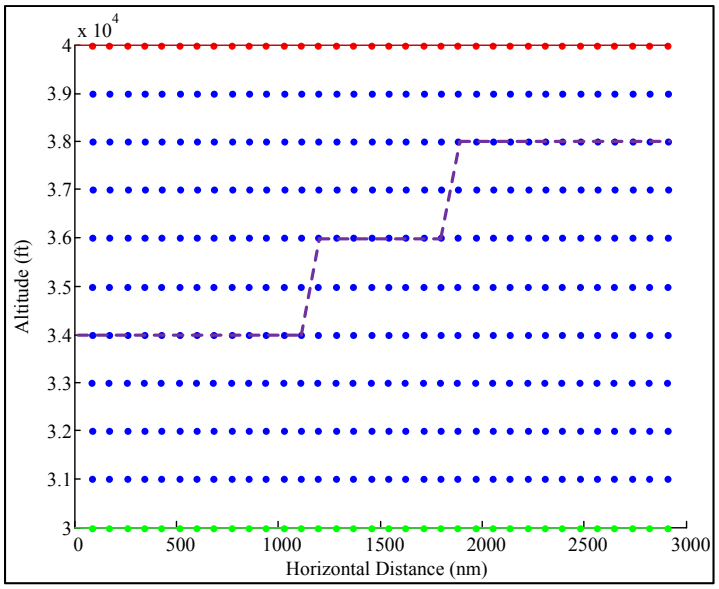

Figure 6.2 Vertical Dimension Search Space

6.2.2.2 Lateral Dimension and its Boundaries

To determine the lateral search space of research, boundaries will be arbitrarily created within a given maximal tolerated distance away from the trajectory of reference. These boundaries start from the ToC, deviate from the reference trajectory at a given angle, remain parallel to the reference trajectory for a certain distance, and finally deviate to reach the ToD.

The deviation is created according to the aircraft's maximum turn angle (Φ), and a maximum offset distance between the boundary and the reference trajectory. The maximum offset distance, determined arbitrarily, is equal to a quarter of the reference trajectory length, multiplied by the tangent of the aircraft's maximum turn angle as shown in Eq. (6.3).

$$MAXDistance = \frac{L}{4} * tan(\Phi) \tag{6.3}$$

where L is the total length of the trajectory of reference and, Φ is the maximum aircraft turn angle. The constructed lateral trajectory boundaries are shown in Figure 6.3.

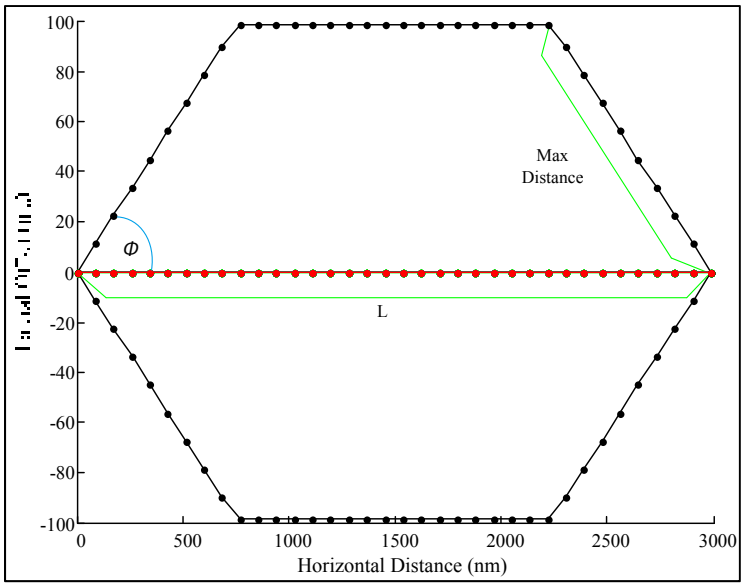

Figure 6.3 Lateral Dimension Boundaries

6.2.2.3 Speed Boundaries

Speed boundaries consist of the minimal and maximal Mach number available for the aircraft. During the algorithm execution, the Mach number can take any required value. However, at the end, the Mach numbers are adjusted to match those available in the numerical performance model (separated in 0.05 Mach number steps). Due to the RTA constraints, the "speed control" is linked to the time boundaries as explained in the next section.

6.2.2.4 Time Boundaries

Time boundaries are created to ensure that the selection of Mach numbers carried out during the optimization, or trajectory creation processes will meet the RTA constraints. However, due to the randomness of weather, it is virtually impossible to predict the exact time boundaries in a 3D space, and thus, time boundaries are only approximations.

If the computed Mach number to reach a given waypoint provides a flight time that is out of the time boundaries, this means that the trajectory cannot fulfill the RTA constraint, and so the trajectory is rejected. In other words, the Mach number selected by the algorithm must provide a flight time to reach the next waypoint that falls within the time boundaries. This computed Mach number must be as well between its minimal and maximal limits.
Figure 6.4 provides an example of the generic trajectories within time boundaries. The outer black lines are the time boundaries and the blue lines are trajectories created that fulfill the RTA constraint

If during the creation of a new trajectory, the speed boundaries at a given waypoint suggest that the required speed is too close or out of bounds, the trajectory is automatically discarded. This is due to the fact that it was observed that Mach numbers near the bounds are uncertain; too much time could be invested in computing these near-the-bound trajectories that at the end will not be able to fulfill the RTA constraint.

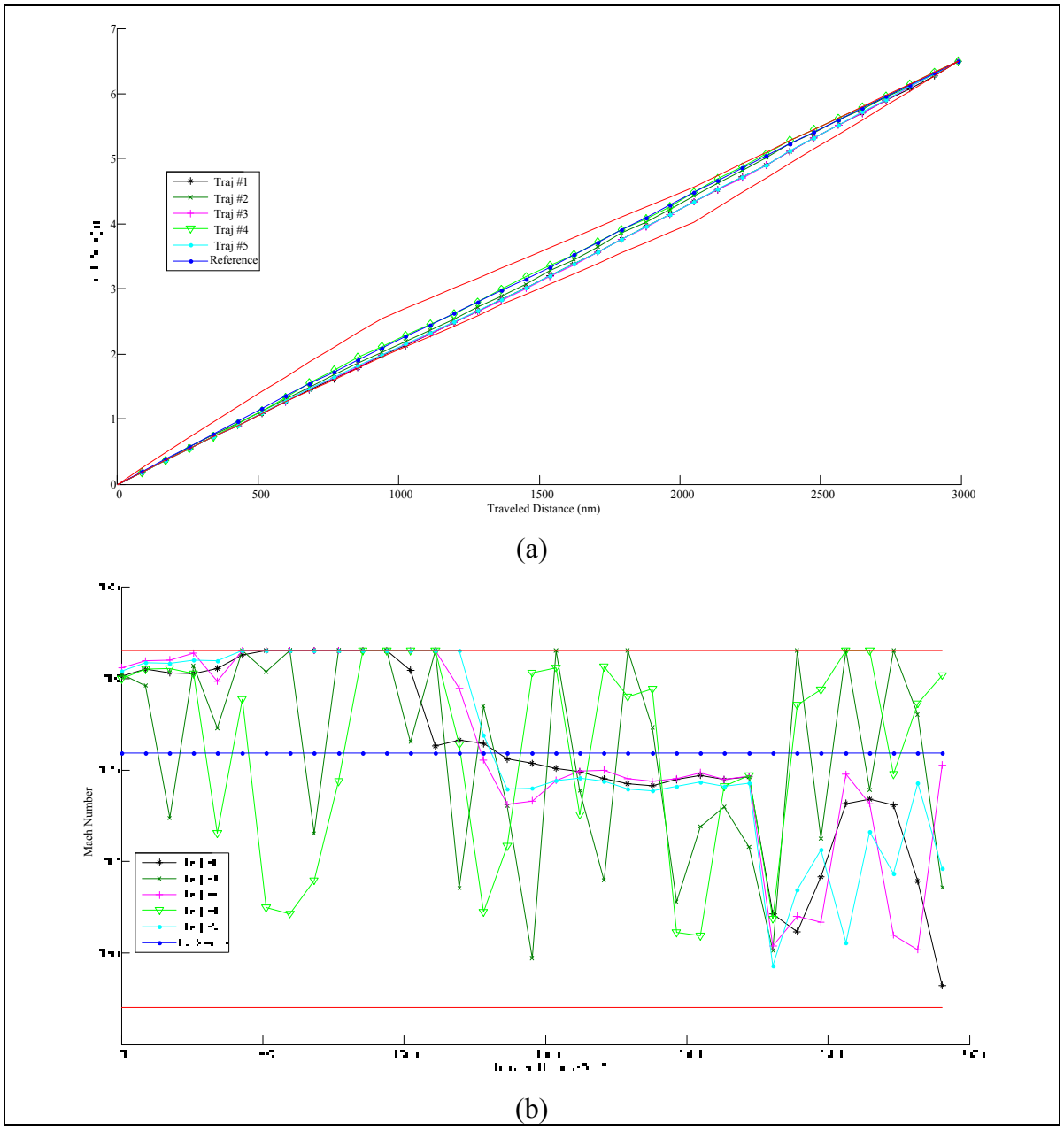

Figure 6.4 Five Generic Trajectories Inside the Time Boundaries and its Mach Variation

6.2.2.5 The RTA Constraint: The First Reference Mach Number

The algorithm requires a unique reference Mach number that fulfills the RTA in order to create trajectories. Due to the weather variation along the route, there is not a direct way to calculate a constant speed for all the waypoints. For this reason, the Golden Section Search

(GSS) method was used to compute this constant Mach number. The GSS is a simple, robust and straight-forward algorithm. A detailed explanation of this algorithm can be found in (Venkataraman, 2009).

The GSS is used to find an extremum (the minimum and closest value to 0) in a strictly unimodal function. A function $f(x)$ is *unimodal* in the interval (a,b) if it contains only one minimum or one maximum on (a,b). The function *fc* used to compute the required Mach number for the entire segment is given in Eq. (6.4). This function is the difference between the flight time and the RTA. In other words, when $fc(mach) = 0$ means that the RTA is fulfilled with an error of 0 seconds.

$$fc(Mach) = |Flight\ Time\ (Mach) - RTA| \tag{6.4}$$

where

$$Flight\ Time(Mach) = \left| \sum \frac{2 * distance}{-V_B + \sqrt{{V_B}^2 - 4 * \left({V_{Ws}}^2 - (Mach * VS_{sound})^2\right)}} - RTA \right| \tag{6.5}$$

and

$$V_B = -2 * W_s * cos(\varphi + W_A) \tag{6.6}$$

where distance is the total length of all the segments, RTA is the required time of arrival, V_{Ws} is a vector containing all the calculated values of W_s, and VS_{sound} is a vector containing all the calculated values of V_{sound}, Mach is the searched Mach number for *fc*. V_B is a vector containing all the calculated values of V_B defined by Eq (6.6) for every segment where W_s is

the local wind speed, W_A is the wind angle, and φ is the aircraft azimuth. This function *fc* is unimodal, as shown in Figure 6.5.

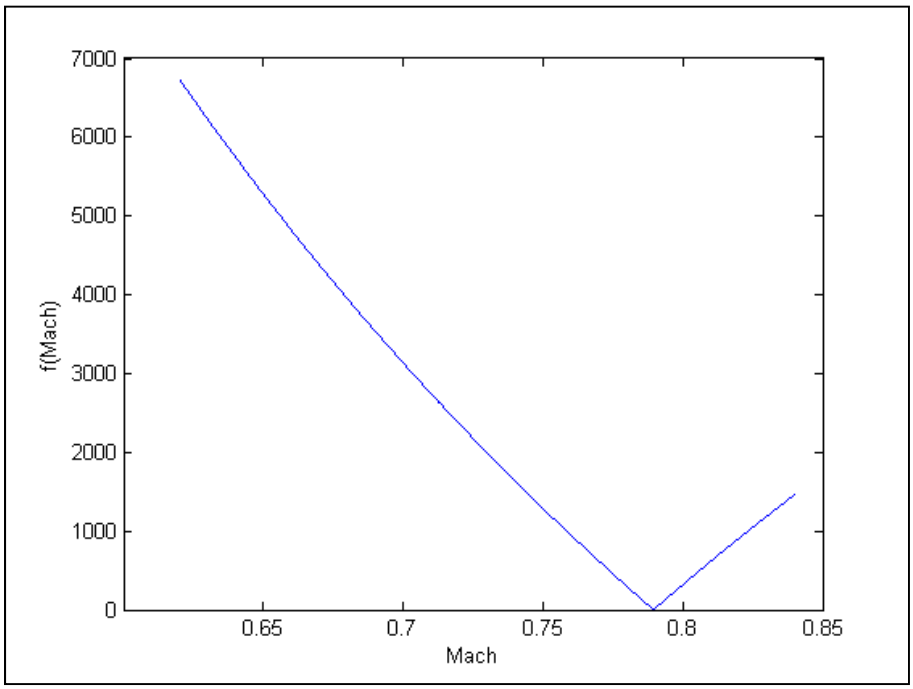

Figure 6.5 Function fc used in the Golden Section Search Algorithm

For a trajectory to be coherent, the required value of Mach must be located between the minimal and maximal Mach numbers available in the aircraft's numerical performance model. The maximal Mach number is first assigned to a variable called *b*, while the minimal Mach number is assigned to a variable called *a*.

Two "new" intermediate points (Mach numbers), called *c* and *d*, are computed inside the interval and calculated with the golden ratio (*gr*) using Eq. (6.7) and Eq. (6.8). The *gr* always has a constant value of approximately 0.618.

$$c = b - gr * (b - a) \tag{6.7}$$

$$d = a + gr * (b - a) \tag{6.8}$$

For example, if the lowest Mach number available is 0.62 and if the highest Mach number available is 0.82, a = 0.62, and b = 0.82. Then, using Eq. (6.7) and Eq. (6.8), two different Mach numbers are computed, these being c = 0.6963 and d = 0.7436. The computed Mach numbers c and d are evaluated with Eq. (6.4). Then, the resulting $f(c)$ and $f(d)$ are compared as in Eq. (6.9), and depending on the results, different procedures are followed as shown in Eq (6.9) – Eq. (6.15).

$$If\ fc(c) < fc(d) \tag{6.9}$$

$$b = d \tag{6.10}$$

$$d = c \tag{6.11}$$

$$c = b - gr * (b - a) \tag{6.12}$$

$$Else$$

$$a = c \tag{6.13}$$

$$c = d \tag{6.14}$$

$$d = a + gr * (b - a) \tag{6.15}$$

In our example, $fc_0(c) \approx 1{,}970$ and $fc_0(d) \approx 409$. This means that $fc_0(c)$ is higher than $fc_0(d)$, so Eq. (6.13) to Eq. (6.15) are followed. As in Eq (6.13), a becomes the current value of c, this is a = 0.6963. Then as in Eq. (6.14), c becomes the current value of d, this is c = 0.7436. Then, as in Eq. (6.15), d is recomputed to become d = 0.7727.

Now, $fc_1(c)$ and $fc_1(d)$ are recomputed with the updated c and d values using Eq. (6.4). The objective is to find a $fc \approx 0$. Following our example, $fc_1(c) \approx 409$ and $fc_1(d) \approx 460$, now fc_1

(c) is lower than fc_1 (d), so Eq. (6.10) to Eq. (6.12) are followed. As in Eq. (6.10), b becomes the current value of c, this is $b = 0.7727$. As in Eq. (6.11) d becomes the current value of c, this is $d = 0.7436$. As in Eq. (6.12) c is recomputed to become $c = 0.7255$.

Now, fc_2 (c) and fc_2 (d) are recomputed with the updated c and d values using Eq. (6.4), and so on. This process is repeated until fc is below a given pre-defined value. In our case, the process stops when a Mach number (d) is found that provides an $fc \approx 0$.

The computed Mach number in this section serves to compute the reference trajectory (blue line) shown in Figure 6.4.

6.2.3 The Optimization Algorithm

Among the metaheuristic algorithms, there is a very large family of algorithms that takes advantage of "swarm intelligence". This term designates a group of systems which replicates the intelligence of several types of swarms (ant, bee, birds, particle, immune systems, etc.). In this large group, the Artificial Bee Colony (ABC) is a very versatile algorithm, efficient with multiple dimensions, able to find an acceptable solution in a large field of research, and thereby capable of efficiently finding the optimal solution. In addition, the ABC algorithm is sometimes considered to be as, or even more, efficient than other algorithms, including "particle swarm optimization" and "evolutionary algorithms" (D. Karaboga & Basturk, 2008). This algorithm has been successfully implemented for its use in solving aeronautical problems, such as the problems explained (Gabor, Simon, Koreanschi, & Botez, 2015) where the drag was reduced from an airfoil.

The ABC algorithm, first proposed by Karaboga (Dervis Karaboga & Basturk, 2007), is a metaheuristic optimization algorithm that mimics the social cooperation and intelligent foraging behaviour of a honey bee swarm. Its general theory and specific implementation for the flight trajectory optimization problem are explained in this section.

6.2.3.1 Description of a Bee's Colony in Nature

The goal of a bees' colony is to gather a maximum amount of food while using a minimum amount of energy. A bee swarm contains different types of bee. Three types are of special interest: the *employed bee*, the *onlooker bee* and the *scout bee*.

Each *employed bee* in the swarm knows a different food source, and the trajectory to link the forge to the food source. Once the *employed bee* has brought back some food, it goes to an area of the hive called "dance floor", to perform the "waggle dance". This dance serves to communicate the location and the importance of the source food to the audience: the *onlooker bees*. The more the *employed bee* "dances" with energy, the more the *onlooker bees* will be impressed, and will be willing to follow the dancing of the *employed bee* to the food source. In this way, the best food sources will have more bees on its route. The last type of bees is the *scout bee*. This bee was previously an *employed bee*, which has abandoned its known food source, and wanders around in the look of new food sources.

Every time any bee returns from the food source, the bee will slightly modify its trajectory. This modification is called a *mutation*. If the *mutation* allows saving a certain amount of energy, this saving will be communicated to the swarm. When a new bee follows the mutated trajectory, this new bee will create its own mutation trying to improve the trajectory. By repeating this mutation process a large number of times, the trajectory to the source food will improve. Since there are more bees (employer & onlookers) on the most promising trajectories, these trajectories tend to improve fast, converging rapidly to their optimal solution (either "local" or "global").

6.2.3.2 The Artificial Bee's Colony Implementation to the Flight Reference Trajectory Optimization

The food source exploited by the bees is located at the aircraft arrival point: the ToD. For the ABC case, this means that all food sources contain the same amount of food; it is minimizing energy (or "flight cost") that is of interest to the algorithm described in this paper.

Within the ABC algorithm, any mutation is numerically based on a neighboring trajectory, which means that the more efficient neighboring trajectories there are, the more efficient the mutation of the current trajectory will be.

The main ABC modules are: Initialization, Employed bees, On-looker bees, Scout bees and Finalization.

6.2.3.3 Module 1: Initialization Phase

This module consists in describing the required inputs for the algorithm to operate and in describing the generation of the initial random trajectories.

Inputs

The required inputs for the algorithm are the aircraft's initial weight, the initial ToC coordinates, the destination ToD coordinates, the initial time at the ToC, the RTA constraint at the ToD, the flight plan (if available), the maximal aircraft's turning angle Φ, the max/min altitudes, and the max/min Mach numbers.

Initial Trajectories Generation

The ABC algorithm requires random trajectories to be assigned to the different "bees". These trajectories are created by placing waypoints within the search space (lateral and vertical

dimensions) while respecting the RTA. The new trajectories' waypoints are placed under the influence of an initial trajectory (either the flight plan or the geodesic trajectory). In the trajectory creation process, the lateral dimension is created first, followed by the vertical dimension, and finally the Mach number to respect the RTA is computed.

The lateral dimension

Trajectories in the lateral dimension are created following two different methods. Some trajectories are created with the first method, and the others are independently created with the second method. The first method is to place the trajectories parallel to the reference trajectory. Trajectories created with this method are placed towards the search space limits. The second method consists in arbitrarily selecting some waypoints from the reference trajectory, and modifying their positions. Trajectories created with this method are closer to the reference trajectory. These methods are stochastic; the generated trajectories are never the same.

The initial trajectories can be seen in Figure 6.6 where the red and green lines delimit the search space, the dark blue line represents the trajectory of reference, and the light blue lines represent the generated trajectories. In this figure, the trajectories generated with the two methods can be clearly identified. The parallel trajectories created with the first procedure are closer to the search space limits (red and green lines), while closer to the trajectory of reference (blue line) are the trajectories created with the second method. The trajectories created with the second method are more chaotic than the ones determined with the first method since waypoints are arbitrarily selected and modified.

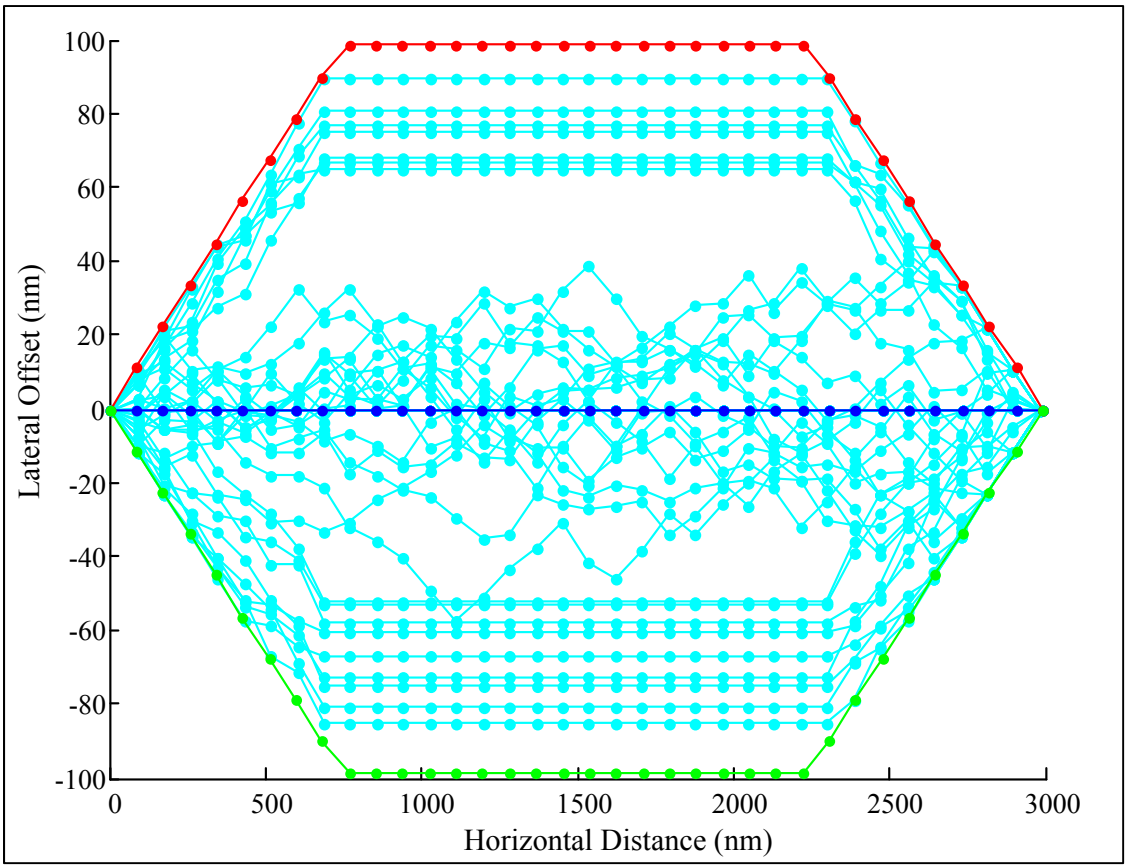

Figure 6.6 Creation of the lateral trajectories with respect to the reference trajectory

Although in Figure 6.6 it might appear that trajectories have abrupt changes of directions, it is not the case when trajectories are traced on a map as shown in Figure 6.7, where trajectories after the optimization process are shown. It can be seen that all trajectories are rather smooth.

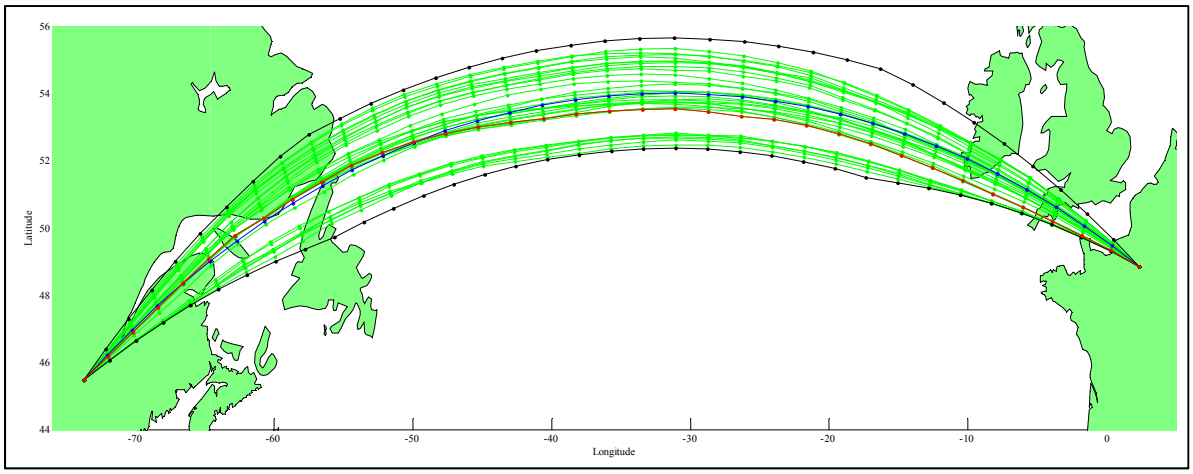

Figure 6.7 Lateral trajectories representation as placed on the Earth

The algorithms described in the literature require to define a classical pre-defined grid which limits the search space. The aircraft can only fly towards the waypoints defined in that pre-defined grid. The trajectory generation process for lateral trajectories in this algorithm has the particularity that it does not require the knowledge of this classical grid. The algorithm can place a waypoint anywhere in the search space by only respecting the trajectory of reference longitude. This type of grid is called a *dynamic grid*. It gives the advantage of exploring search space areas that are impossible to be explored with the classical grid approach.

A comparison between the classical grid and the dynamic grid is shown in Figure 6.8.

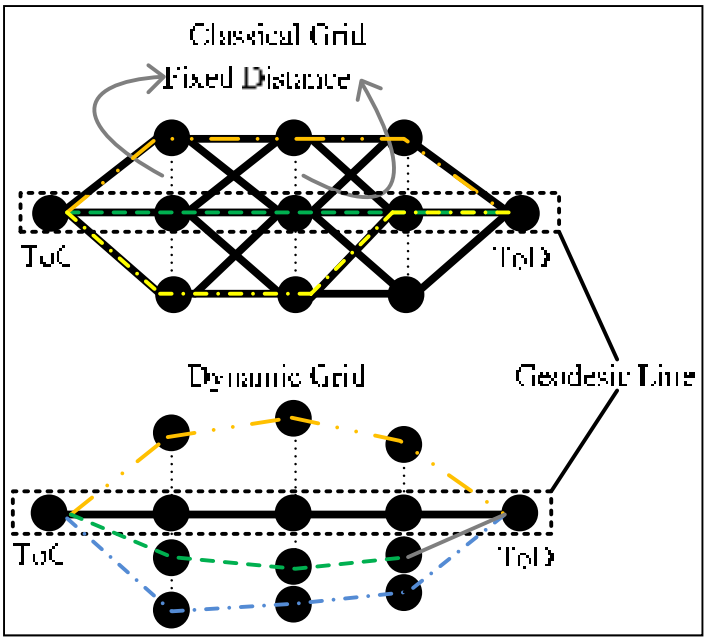

Figure 6.8 Fixed grid and dynamic grid comparison

The classical grid on the top of allows the hypothetical trajectories to move only between the pre-defined waypoints at fixed distances. The space between the waypoints is not explored as the waypoints are fixed. The *dynamic grid* shown below of Figure 6.8, on the other hand allows moving the waypoints to different parts of the search space so that a higher part of the search space can be explored.

The vertical dimension (Altitude)

Creating the trajectories in the vertical dimension requires them to change in altitude in multiples of 2,000 ft to agree with current traffic control conventions. Since the vertical space search is small (around six possible altitudes are available by the legislation), a random function to define the waypoint's altitude would be enough to cover all the search space. Since the altitude changes are imposed, a classical fixed grid was used in this vertical dimension, and was shown earlier in Figure 6.8.

The Mach number selection: the time dimension

At this point, a trajectory has already been created in a 3D space (latitude, longitude, and altitude). Now, the methodology followed to compute the combination of Mach numbers that fulfill the RTA is detailed.

Using the reference Mach number computed with the GSS mentioned beforehand, the Estimated Time of Arrival (ETA) at each waypoint is known. The ETA at each waypoint is arbitrarily modified to times between the time boundaries. For example, in Figure 6.9 the dark blue line represents the ETA's at each waypoint when flying at the reference Mach number. The light blue line represents the ETA modification at each waypoint. The black lines represent the time boundaries; again the modified times should fall within these two black lines.

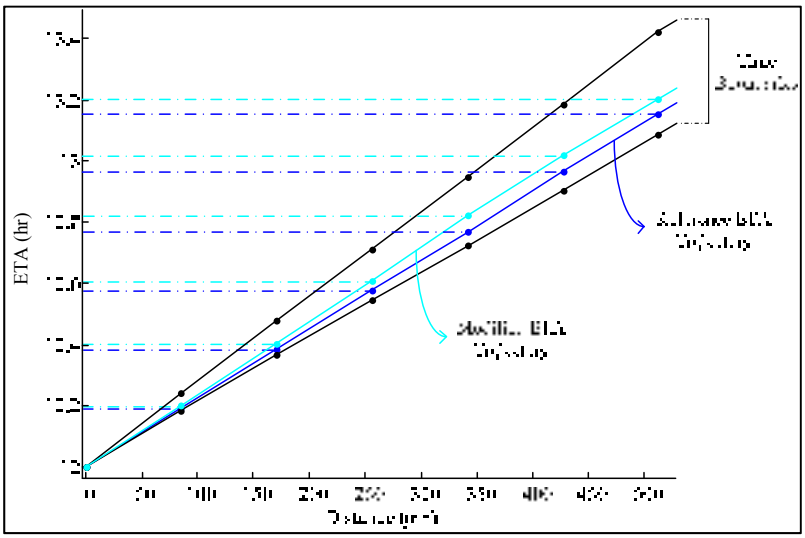

Figure 6.9 Flight Time Limits

As the ETA changed, the Mach number has to be re-computed. The required Ground Speed (GS) to reach the new ETA is computed using Eq. (6.16):

$$GS = \frac{segment\ distance}{ETA\ at\ a\ given\ waypoint} \tag{6.16}$$

where *segment distance* is the distance that the aircraft has to travel between two waypoints.

The *TAS* can then be determined taking the wind into account by using the wind triangle as in Eq. (6.17), where W_s is the wind speed and V_B was defined in Eq. (6.6)

$$TAS = \sqrt{GS^2 + GS * V_B + {W_s}^2} \tag{6.17}$$

Finally, the Mach number is found by using Eq. (6.18).

$$Mach = \frac{TAS}{V_{sound}} \tag{6.18}$$

There are rare cases when the resulting Mach number is found outside the aircraft's Mach numbers limits. In these cases, the maximal (or minimal) Mach number is used to compute and update the ETA value to the next waypoint.

When the modified ETA does not fall within the time boundaries, even at Maximal or Minimal Mach numbers, it means that the RTA condition cannot be respected. The whole trajectory is then discarded, and the whole trajectory creation process is re-executed.

The time dimension for different trajectories generated in 4D that respect the RTA are shown in Figure 6.10.

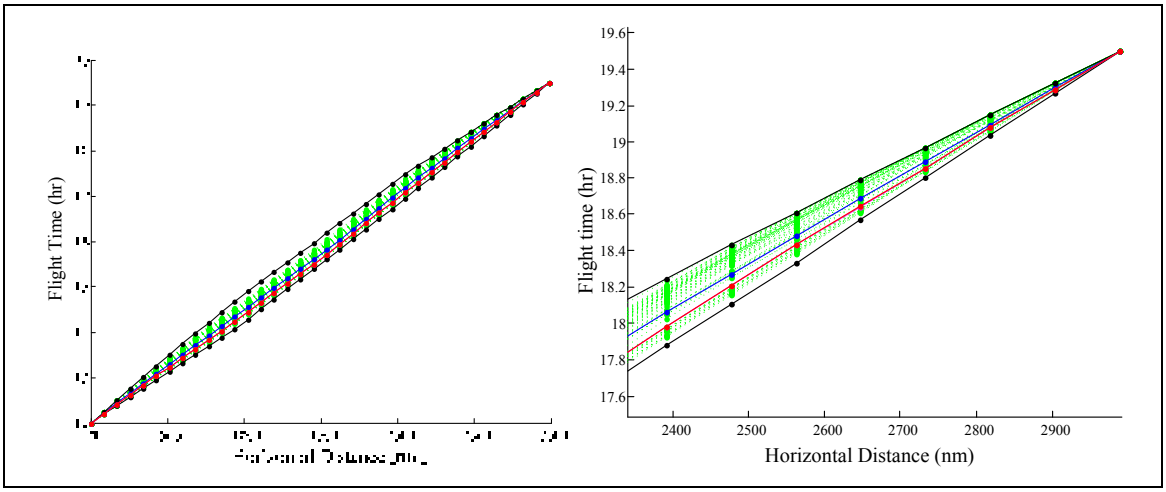

Figure 6.10 Left: Trajectories in the time dimension.
Right: Convergence to the RTA time

Once all the initial trajectories are created, their cost is evaluated. The most economical trajectory is selected as the *current global best trajectory*.

6.2.3.4 Module 2: Employed Bee Phase

In this module, every trajectory generated is assigned to an *employed bee*. The main task of the employed bee is to modify its assigned trajectory in order to attempt to reduce its cost. This modification is carried out via a mutation. If the mutated trajectory is more economical than the current trajectory, the mutated trajectory takes the place of the original trajectory. This is called a *successful mutation.* Otherwise, if the mutation fails to improve the trajectory (*fail mutation)*, the mutated trajectory is discarded, and a *fail counter* is incremented. When the fail counter overflows, as it will be explained later in Module 4, the trajectory assigned to that employed bee is discarded and a new trajectory is created.

The steps executed in the *employed bee* module are the following:

1) The first time this module is executed, every trajectory created during the *initialization* process is assigned to a different *employed bee*. For the following iterations, each *employed bee* keeps its trajectory.

2) Every *employed bee* mutates its assigned trajectory.

3) A trajectory cost comparison is performed. If the mutation reduces the trajectory's cost, this mutated trajectory is kept, otherwise, it is discarded and a *fail counter* is incremented.

As all trajectories have to mutate, this process is repeated until all trajectories are mutated either with a successful or a fail mutation. The mutation process is explained next.

Mutation

A mutation is a slight change of the trajectory which hopefully would reduce the flight cost. Bees can mutate their assigned trajectory in 3 different dimensions: the lateral dimension, the vertical dimension (altitude) and the time dimension (Mach number selection).

When trajectories are equally distributed in the search space, successful mutations will place the selected waypoint in a wide range to better explore the search space. After several iterations, when many trajectories are close to the most efficient trajectory, successful mutations will place the waypoint closer to the most efficient trajectory to search convergence. It is thus important to regularly improve every trajectory, even the less efficient ones, as they influence the most efficient ones.

Every mutated trajectory needs to be verified in all dimensions (time, altitude, and lateral) to guarantee that it does not cross the search space boundaries. If the trajectory mutation in any dimension gives a waypoint of the trajectory outside the space boundaries, the trajectory keeps mutating until a valid trajectory is found.

The mutation process is as follows:

1) The dimension on which the trajectory is going to be mutated is arbitrary selected.

2) An arbitrary waypoint from the trajectory is selected. The trajectory will be mutated only on this waypoint.

3) The mutation is performed.

4) The complete mutated trajectory is evaluated.

Depending on the dimension, mutations take place in a different way.

Mutation in the lateral dimension

This dimension has countless mutation possibilities due to the implementation of the dynamic grid (Section 6.2.3.3). A mutation in this dimension refers to change the waypoint location. This dimension looks for favorable tailwinds and to avoid headwinds.

In order to obtain an efficient mutation, a waypoint from an arbitrary neighbor trajectory is taken as reference. The selected waypoint can only be placed between the reference waypoint neighbor trajectory position and the position of the waypoint from the trajectory being mutated. The lateral mutation is computed with Eq. (6.19).

$$\text{L}ateral\ Mutation_W = round\left[\frac{Position_W + (Position_W - Position_N) * random}{step}\right] * step \tag{6.19}$$

where *Lateral Mutation*$_W$ provides the new waypoint location, *Position*$_W$ is the initial waypoint position, *Position*$_N$ is the neighbor waypoint position, *random* is a random value between (-1...1), and $step$ is the step of the grid for the selected dimension. *Step* is given in the order of fractions of nautical miles.

The time dimension

A mutation in this dimension consists in changing the ETA to the selected waypoint. As a consequence, the Mach number required to fly that segment is recomputed in order to reach the selected waypoint at the new ETA.

Similar as with the lateral dimension, an arbitrary neighbor trajectory is selected to influence the mutation as expressed in Eq. (6.20).

$$New\ ETA_W\ Mutation = ETA_W + (ETA_W - ETA_N) * (random - 0.5) * 2 \quad (6.20)$$

where *New ETAW* is the ETA at the waypoint after the mutation, ETA_w is the current ETA for the selected waypoint, ETA_N is the ETA for the waypoint found in the neighbor trajectory of reference, and *random* is an arbitrary value between 0 and 1.

The vertical dimension

The altitude is the dimension that influences the most the fuel burn. However, its mutation is the simplest one as the selected waypoint altitude is modifying its altitude either in 2,000 ft or 4,000 ft steps. Since the algorithm does not allow descents, the rest of the trajectory is studied to verify that the rest of the altitudes are the same or higher as the ones considered in the mutation. Figure 6.11 shows some examples of vertical mutations.

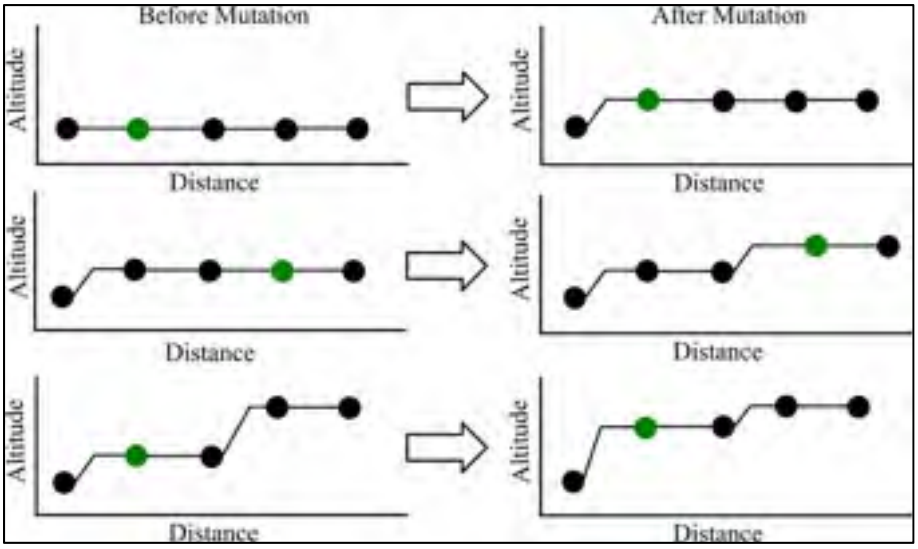

Figure 6.11 Trajectory Mutation on the Vertical Dimension

In Figure 6.11 the green waypoint position on its left hand side is modified to obtain the trajectories on the right hand side. Note how all the following waypoints are adjusted, so that

the next nodes are at the same altitude or at higher altitudes. This operation is done to avoid descending during cruise.

6.2.3.5 Module 4: On-Looker Bee Module

This module is executed right after the *employed* bee Module 2. It intends to mimic the elitist trajectory selection process followed by the *on-looker* bees.

On-looker bees mutate the trajectory in the same way as the *employed* bees. Following the real bees' behavior and systematically assigning the most fitted trajectories to the *on-looker* bees could bring as a consequence finding the local optimal instead of the global optimal solution. For this reason, a trajectory should pass an "efficiency test" (economy test) before being assigned to an *on-looker* bee. The more efficient a trajectory is, the greater its chances are to pass the test. This *efficiency* test uses the parameter called *Efficiency Weight* (*EW*) shown in Eq.(6.21).

$$EW = \frac{0.5 * fitness - Min_{fitness}}{Max_{fitness} - Min_{fitness}} + 0.5 \tag{6.21}$$

where *fitnes*s is a parameter that depends on the trajectory cost. The more economical is the trajectory, the greater is its fitness. $Max_{fitness}$ is the maximal fitness available (most economical trajectory), and $Min_{fitness}$ is the minimal fitness available (most expensive trajectory). The *EW* typical value range is from 0.5 to 1.

A trajectory is defined as efficient when the *EW* value is higher than a generated random number (from 0 to 1). In this way, less efficient trajectories have their chances to be selected. If the current trajectory fails the test, then it is not selected by the on-looker bee, and another trajectory is evaluated. If the selected trajectory succeeds the test, it is assigned to an on-looker bee in order to be mutated.

On-looker bees do not keep trajectories as their own as the employed bees. For this reason, the trajectory can be selected by different *on-looker* bees. The *on-looker* bee is not attached to its assigned trajectory. The next time this module is executed, the *on-looker* bee can select a different trajectory.

This module ends when the number of successful trajectory mutations executed is equal to the number of onlooker bees. The most economical trajectory is selected as the "global optimal".

The on-looker bee's module steps can be described as follows:

1) The first available trajectory is selected.
2) The *EW* test is performed.
    a) If the test is not successful, the trajectory is rejected. The current on-looker bee selects the next available trajectory. Step 2 is executed.
    b) If the test is successful, the mutation process is executed. After a successful mutation is executed, the next on-looker bee selects the next available trajectory, and Step 2 is executed. When there are no more *on-looker* bees available, Step 3 is executed.
3) The most economical trajectory is selected as the current global optimal.
4) Module Ends

When this module ends, either the *employed bee module* or the *scout bee module* is executed.

6.2.3.6 Module 4: Scout bee module

This module mimics the bees that are always looking for new food sources.

Every trajectory mutated by the employed and the on-looker bees has an associated "fail counter". This counter is incremented every time a mutation (either by the employed or the on-looker bees) created fails to improve the trajectory. When this counter reaches a pre-

defined value, the associated trajectory is deleted and a new trajectory is generated. The "fail counter" for the new generated trajectory will be set to '0'.

In this way, spending of computational resources are avoided on trajectories that may not be the global optimal, but could be "local optimal".

6.2.3.7 Stopping Criteria

The stopping criteria could be reached after the maximum number of iterations, or after meeting the required accuracy optimization level.

6.2.3.8 Module 5: Finalization

Once the algorithm has fulfilled the stopping criteria and the "optimal" trajectory has been found, a last task remains to be achieved. The Mach number schedule computed to fulfill the RTA constraint might be "too accurate" to be flown. In other words, its variation would be too small to be respected, and could require a level of speed control that current aircraft systems may not be able to provide. The computed Mach numbers are then adjusted to the Mach numbers values available in the numerical performance model (0.05 Mach number steps). The adjustment process is as follows.

The Mach number used to fly the first segment (wpt_1 to wpt_2) is adjusted to its closest value available in the numerical performance model. The ETA_{new} to reach the next waypoint (wpt_2) with the adjusted Mach number is then computed. The Mach number selection to fly the next segment (wpt_2 to wpt_3) depends on the ETA_{new} at wpt_2. If ETA_{new} is lower than ETA_{old} (ETA computed with the original Mach numbers), then the selected Mach number to fly the second segment (wpt_2 to wpt_3) is adjusted to the closest higher Mach number available in the numerical performance model. If the ETA_{new} is higher than the ETA_{old}, the Mach number for the second segment (wpt_2 to wpt_3) is adjusted to the closest lower Mach number available

within the numerical performance model. The ETA_{new} to reach the next waypoint (wpt_3) is then recomputed. This process is repeated until the last waypoint is reached.

Evidently, at the end of the Mach number adjustment, time difference between the ETA_{new} at the last waypoint and the required RTA could be significant. A new waypoint, between the ToD and the TOD (n-1), is created in order to allow a last Mach number adjustment, and to respect the required RTA. This new point is called "Transition Waypoint", and is shown in Figure 6.12.

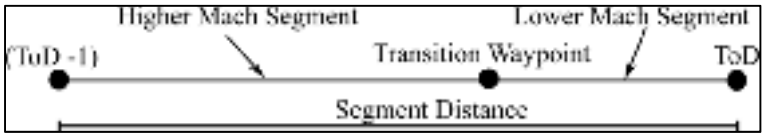

Figure 6.12 Last Waypoint Mach Number Selection

The transition waypoint is placed as follows. Firstly, the exact Mach number ($Mach_{exact}$) between (ToD-1) and the ToD that fulfills the RTA is computed. Secondly, the closest lower and closest higher Mach numbers available in the numerical performance model are identified. For example, for a computed Mach number of 0.7535, the lower Mach number would be 0.74, and the maximal high Mach number would be 0.76. Thirdly, the flight time to reach from (ToD – 1) to ToD is computed with each one of the identified speeds. If the segment between (ToD – 1) to the *transition waypoint* segment is flown with the identified higher Mach number, and the segment between the *transition waypoint* to the ToD is flown with the identified lower Mach number. Then, the distance (d_1) from (ToD-1) where the *transition waypoint* would be placed is computed with Eqs. (6.22) – (6.26) where *Flight* $Time_{opt}$ corresponds to the flight time delivered by the algorithm, and *Segment Distance* corresponds to the distance between the (ToD -1) to the ToD.

$$\begin{aligned} &Higher\ Mach\ Number\ Time \\ &= Flight\ Time\ with\ Higher\ Mach\ Number \\ &- Flight\ time_{opt} \end{aligned} \tag{6.22}$$

$$\Delta_{time} = RTA - Flight\ time_{opt} \tag{6.23}$$

$$t_1 = \frac{\Delta_{time} - LM\ time}{HM\ time - LM\ time} * \Delta_{time} \tag{6.24}$$

$$v_1 = \frac{Segment\ Distance}{HM\ time} \tag{6.25}$$

$$d_1 = v_1 * t_1 \tag{6.26}$$

Finally, the trajectory flight cost is recomputed with this new transition waypoint.

6.2.4 Algorithm Summary

The algorithm described above can be summarized in its different modules as shown in Figure 6.13.

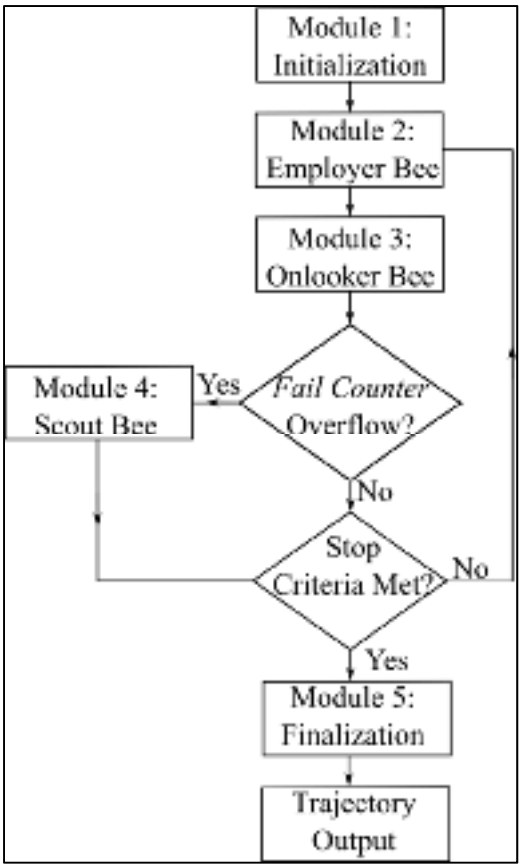

Figure 6.13 ABC Algorithm Flowchart

Initialization (Module 1): Random trajectories are created. Each trajectory is assigned to an employed bee.

REPEAT

Employed bee module (Module 2): Each employed bee goes to its food source, creates a mutation on the trajectory and evaluates it. If the mutated trajectory is more efficient than the old one, it replaces the old one.

Onlooker bee module (Module 3): Each trajectory is rated according to its efficiency. A random function will determine which trajectory each on-looker bee will follow, and the higher chances that will be given to follow the best trajectories. After the selection of the trajectory, the on-looker bee follows it and creates a mutation. In the same way as the employed bee, if the mutated trajectory is more efficient than the old one, it replaces it. The best trajectory is memorized.

IF Fail Counter overflow == true

- Scout bee module (Module 4): The number of mutations created on every trajectory, without any improvement on it, is known. If this number reaches the pre-defined maximum of mutations, it is assumed that the associated trajectory cannot be improved anymore without a too-high cost in time. The trajectory is forgotten, and a new one is created for the concerned employed bee.

UNTIL (Stopping criteria is fulfilled)

Finalization.

6.3 Results

Different tests[5] were performed to evaluate the efficiency and robustness of the algorithm. The first set of tests measured the ABC algorithm's efficiency for different iterations. The second set of tests measured the algorithm's robustness (how often the algorithm provided the optimal trajectory). For the third test, a case study from a real flight obtained from *FlightAware*® was analyzed, and potential fuel savings were obtained. For the last set of tests, the fuel savings potential for different real flights were presented.

The studied aircraft is a long haul model with 2 turbo fan engines, a maximal TO weight of 160 tons, a range of Mach numbers from 0.68 to.0.82, and a maximal certified ceiling of 39,000 ft.

6.3.1 Number of Iterations' Influence on the Resulting Trajectory

The first study case is a flight from Montreal to Paris, the 13th of June starting at 12.00 am and finishing at 06.30 pm, for an initial aircraft weight of 120 Tons. The reference trajectory is a direct, geodesic line for a Mach number of around 0.761. The reference trajectory consumes close to 25,493 kg of fuel.

With this flight, the number of iterations is modified to see how the optimal reference trajectory is affected in terms of time of arrival and fuel savings. Table 6.1 shows the results.

---

[5] All tests were performed in a personal computer with a AMD Phenom™ II X4 B93 2.80 GHz Processor; 6.00 GB of RAM; 64-bit Windows™ 7 enterprise, and Matlab™ 2013b

Table 6.1 Solution Evolution when Varying the Number of Iterations

Test	Iterations	Computation Time (s)	RTA	Time of Arrival difference (s)	Fuel savings (kg)
1	745	58	6h30	0.31	1199 (4.70%)
2	1500	127	6h30	0.24	1216 (4.77%)
3	3000	161	6h30	0.19	1224 (4.80%)
4	6000	314	6h30	0.02	1240 (4.86%)
5	8000	587	6h30	0.14	1252 (4.91%)

Following the analysis of results obtained for different number of iterations shown in Table 6.1, it can be seen that the number of iterations reduced the fuel burn, but it is evident that at the same time, the computation time increased, as more trajectories were computed. For all the flight tests shown in Table 6.1, the RTA was fulfilled with a maximal difference of 0.31 seconds (Table 6.1), and a minimal difference of 0.02 seconds (test 4). The time of arrival differences is not proportional with the number of iterations as the algorithm does not seek to reduce this difference, but to find the most economical flight cost while fulfilling the RTA constraint.

It was noticed that with fewer iterations, the trajectories presented more abrupt variations, especially in the lateral reference trajectory. Figure 6.14 to Figure 6.16 show the trajectories results for Test 1 in Table 6.1, and Figure 6.17 to Figure 6.19 show the trajectory for Test 5 in Table 6.1.

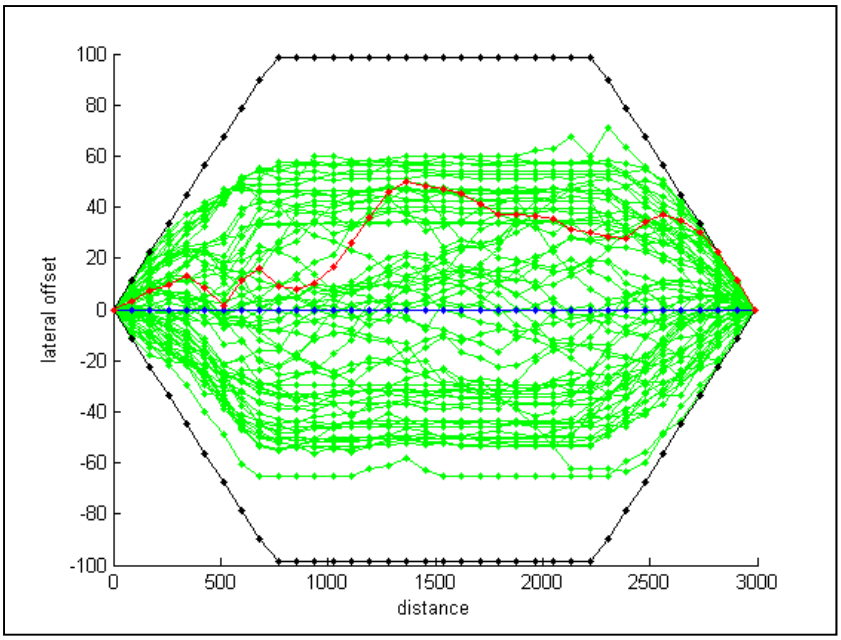

Figure 6.14 Last candidate trajectories (green), and the optimal trajectory (red) for Flight 1

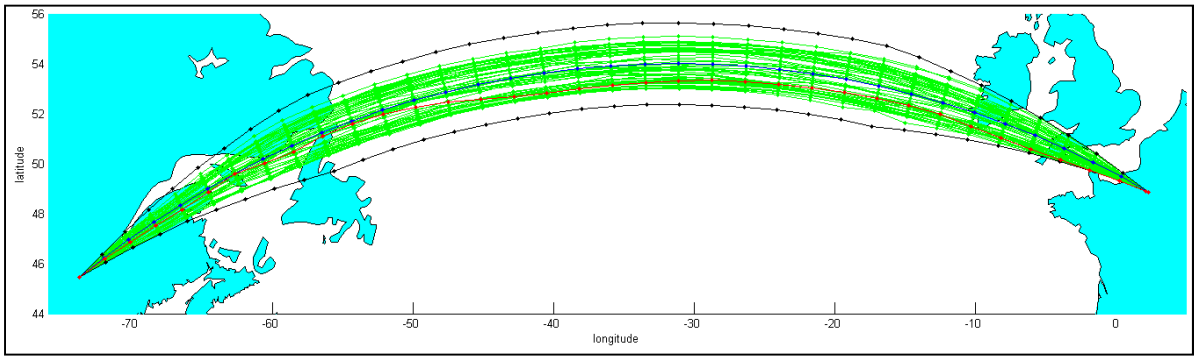

Figure 6.15 Lateral Trajectories: Optimized trajectory (red), the geodesic trajectory (blue), and remaining candidate trajectories (green).

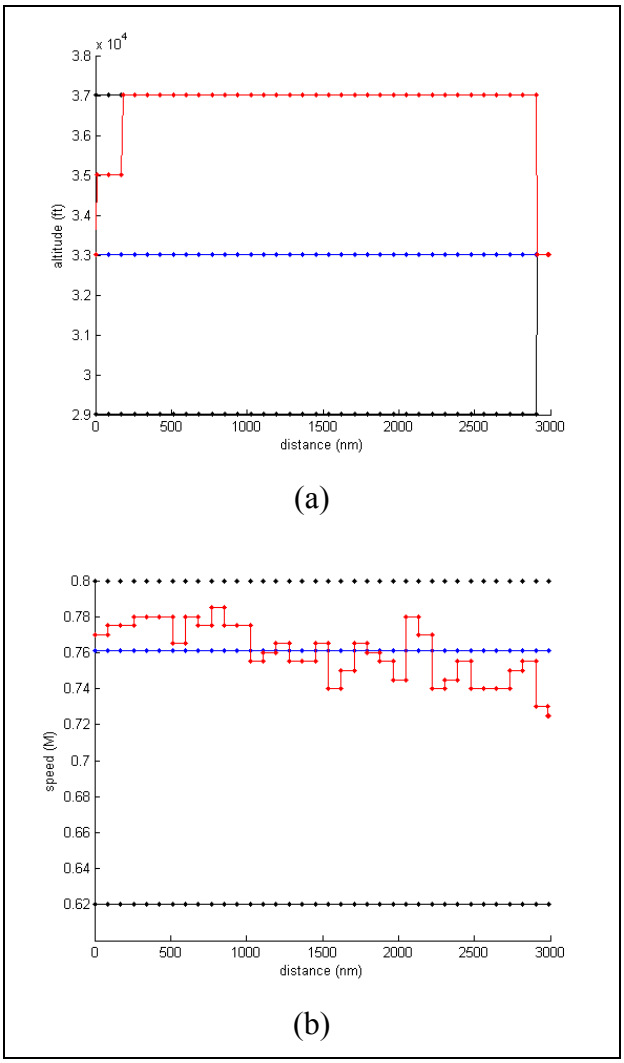

Figure 6.16 Altitude versus Distance (a). Speed versus Distance (b) for Flight 1

Figure 6.14 and Figure 6.15 show the lateral trajectories. Two types of representations are enclosed in these figures: the optimized lateral trajectory is represented with the red line, and the rejected trajectories being mutated by bees can be observed in green inside the search space area delimited by the black lines. Figure 6.16 (a) shows the optimal altitudes to be followed with the red line. The blue line is the constant initial altitude. It can be seen that the initial altitude of reference was 33,000 ft. As seen in Figure 6.16 (a), the algorithm suggests starting the flight at 35,000 ft, and then performing a step climb. The Mach number schedule

changes along the trajectory is seen in Figure 6.16 (b). The Mach number variation is small; its minimal value was 0.72 and its maximal value was 0.78.

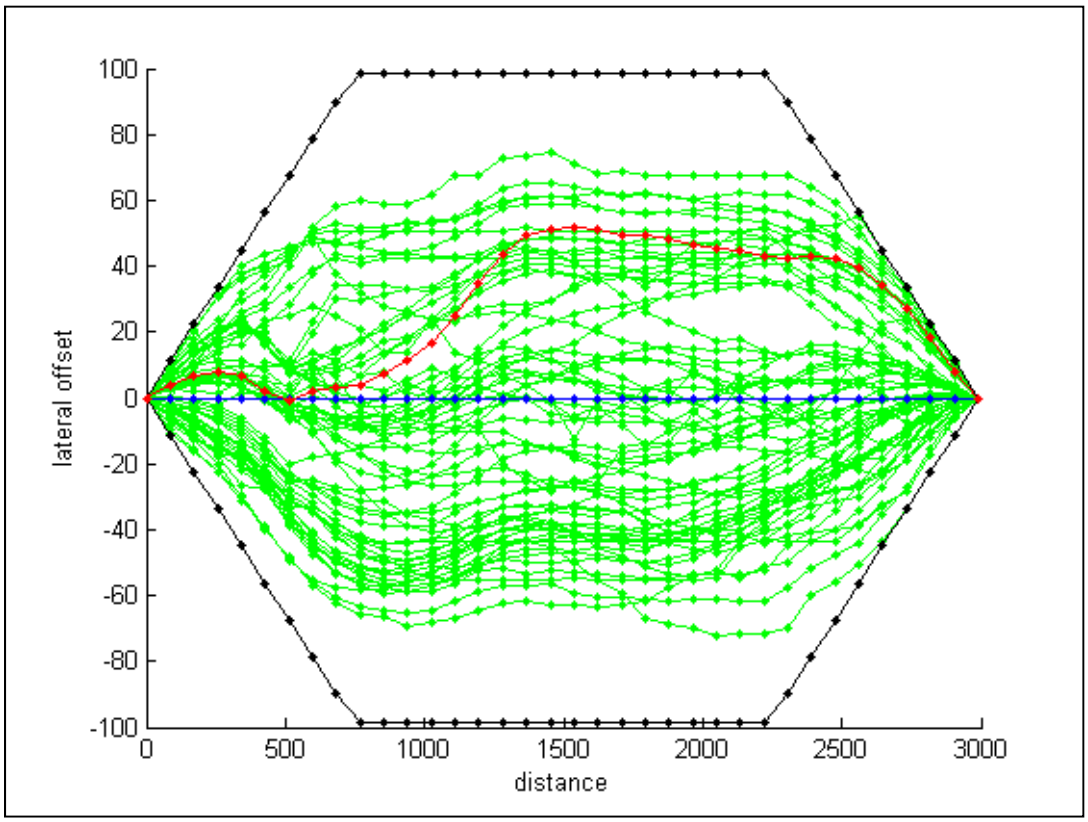

Figure 6.17 Lateral offset depending on the travelled distance (8000 iterations – for test 5)

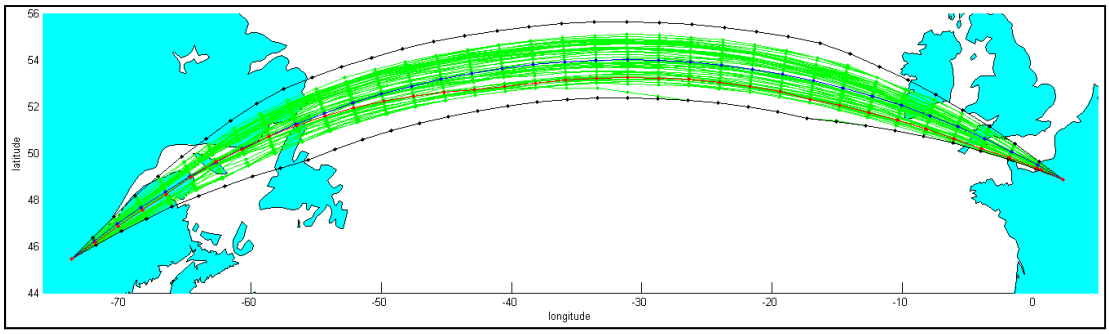

Figure 6.18 Optimized lateral trajectory (red) nearby the reference (blue), and other candidate trajectories (green) for test 5

In Figure 6.17 and Figure 6.18, a smoother lateral trajectory was found with respect to the lateral trajectory shown in Figure 6.14 & Figure 6.15; this was due to an increment in the number of iterations. Figure 6.19 shows the same type of altitude and Mach numbers variations with the distance as the ones shown in Figure 6.16 (a). The Mach number schedule in Figure 6.19 shows fewer variations than the one shown in Figure 6.16 (b) ; in practice it is desired to fly at a constant Mach number as much as possible. As expected, if the computation time is not a constraint, a higher number of iterations are preferred to obtain a better trajectory as in the case of Test 5 (needing more iterations than Test 1) which provided better results in terms of the optimized trajectory than Test 1.

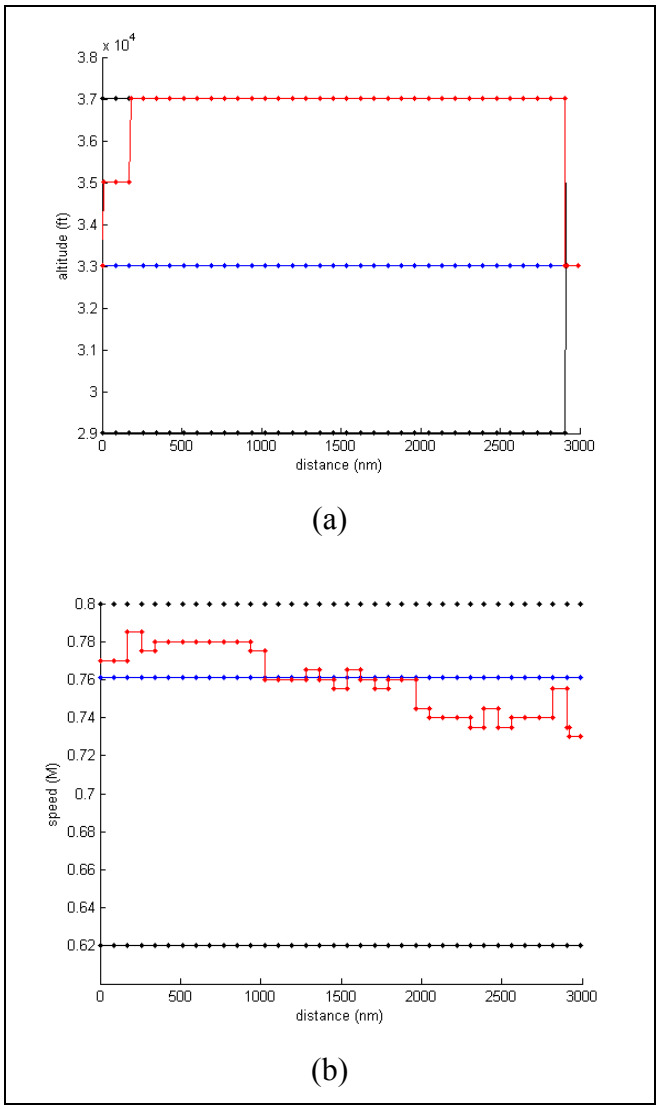

Figure 6.19 Optimized altitude and speed (red) around the reference (blue)

6.3.2 The ABC´s Robustness

A set of tests was performed to determine the ability of the algorithm to find the same solution every time it is executed. In other words, to evaluate the algorithm's robustness.

The evaluated trajectory was a flight from Paris to Montreal the 11th November of 2015. The flight cost for this trajectory following the geodesic, with a constant Mach of 0.8 at a constant

altitude of 33,000 ft was of 29,225 kg. This cost was used as reference to determine the ability of the algorithm to find the same optimal solution.

This reference flight was optimized by executing the algorithm an arbitrary number of 800 times with the same initial conditions such as the same weather, and the same aircraft's weight at the ToC. Thus, 800 "optimized trajectories" were obtained. The amount of fuel saved by the 800 trajectories provided by the algorithm is shown in Figure 6.20.

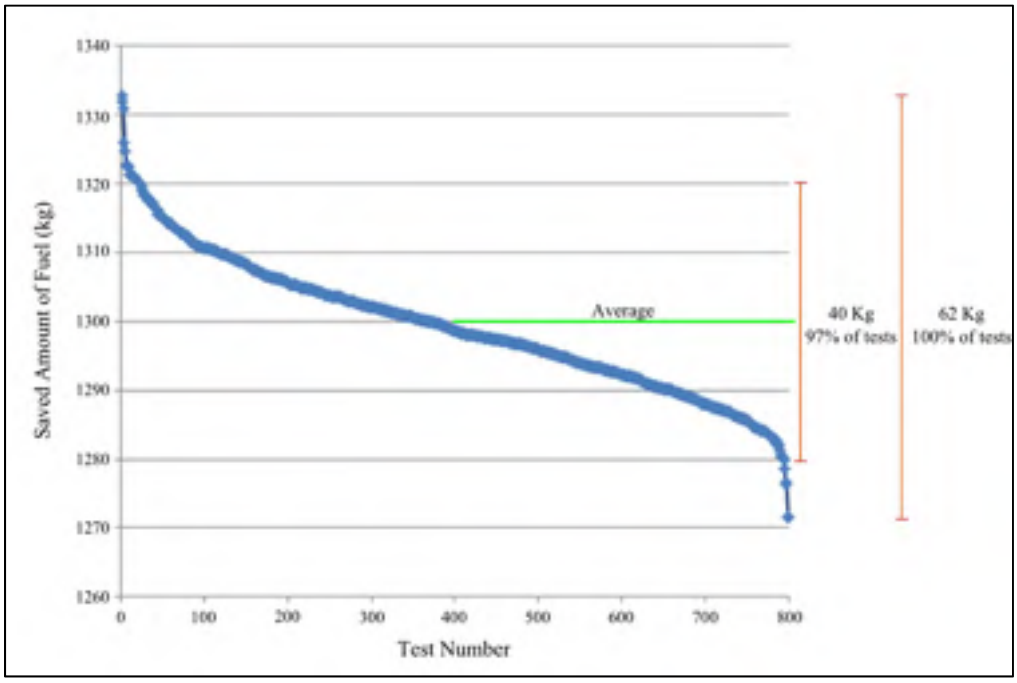

Figure 6.20 Saved amount of fuel for 800 trajectories

The amount of saved fuel was between 1,271 kg, and 1,333 kg, thus the average was found to be of 1,300 kg (To total average fuel consumption of 27,925 kg). That is, 97% of the results were found to be between 1,281 kg and 1,321 kg, which represents a difference of only 40 kg, or close to 0.14% of the average fuel consumption. Due to the fact that the algorithm uses a dynamic grid in the lateral trajectory, making it is almost impossible to find the same lateral trajectory twice, a difference of 40 kg is considered to be a very good result as it is only a small fraction of fuel burn. It can be concluded that the algorithm is stable to a range of 40 kg

for 97% of cases. In the same way, following the maximal and minimal savings obtained, it can be concluded that the algorithm is stable to a range of 62 kg for 100% of the tests (or close to 0.21%).

6.3.3 Real Flights Study

The last set of tests was done to study the algorithm's results compared to a real flight plan flown by the same type of aircraft. The flight plan was obtained via *FlightAware*®. The selected flight took place between Edmonton (YEG) and Punta Caña (PUJ), the 13th of November 2015. The flight reached its ToC at 13h38, and its ToD at 19h06, with a total flight time of 5h27min41s. The flown flight plan in the lateral dimension is shown in Figure 6.21.

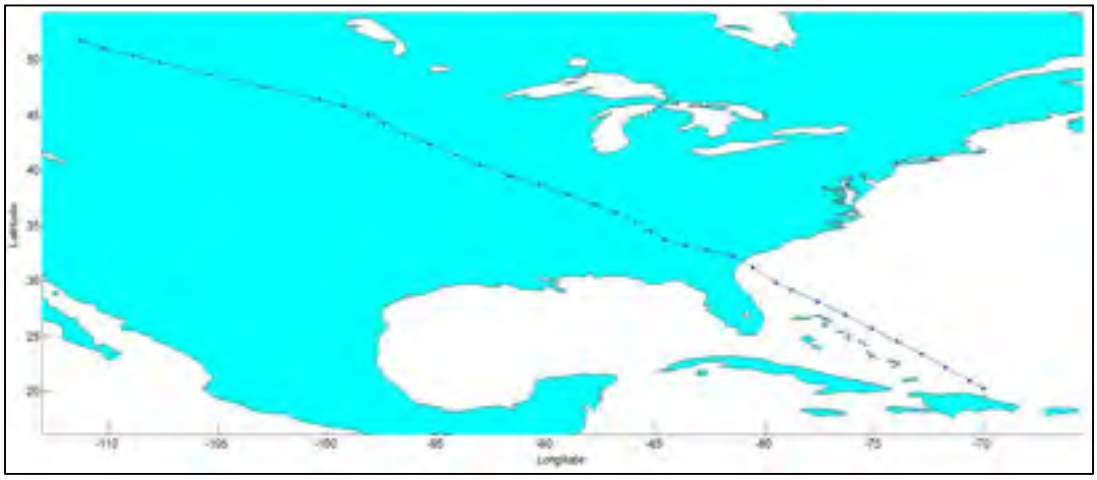

Figure 6.21 Flight Plan From Edmonton to Punta Caña.

An aircraft's weight at the ToC of 110,000 kg was assumed in order to compute the flight plan's fuel burn. This aircraft's weight was considered a normal cruise weight for this aircraft on the flying distance. With these considerations, the estimated fuel burn for the cruise phase provided in the flight plan was close to 21,454 kg.

This flight trajectory was optimized with the algorithm developed in this paper. The same flight conditions were set for the optimization algorithm, and the RTA was set to occur at the same time at which the real flight reached the ToD, thus at 19h06. The optimal reference trajectory output provided by the algorithm only required around 20,483 kg of fuel, representing an improvement of around 4.52 % (savings close to 971 kg) compared to the initial flight plan.

The different dimensions of the output trajectory are shown in Figure 6.22 to Figure 6.25.

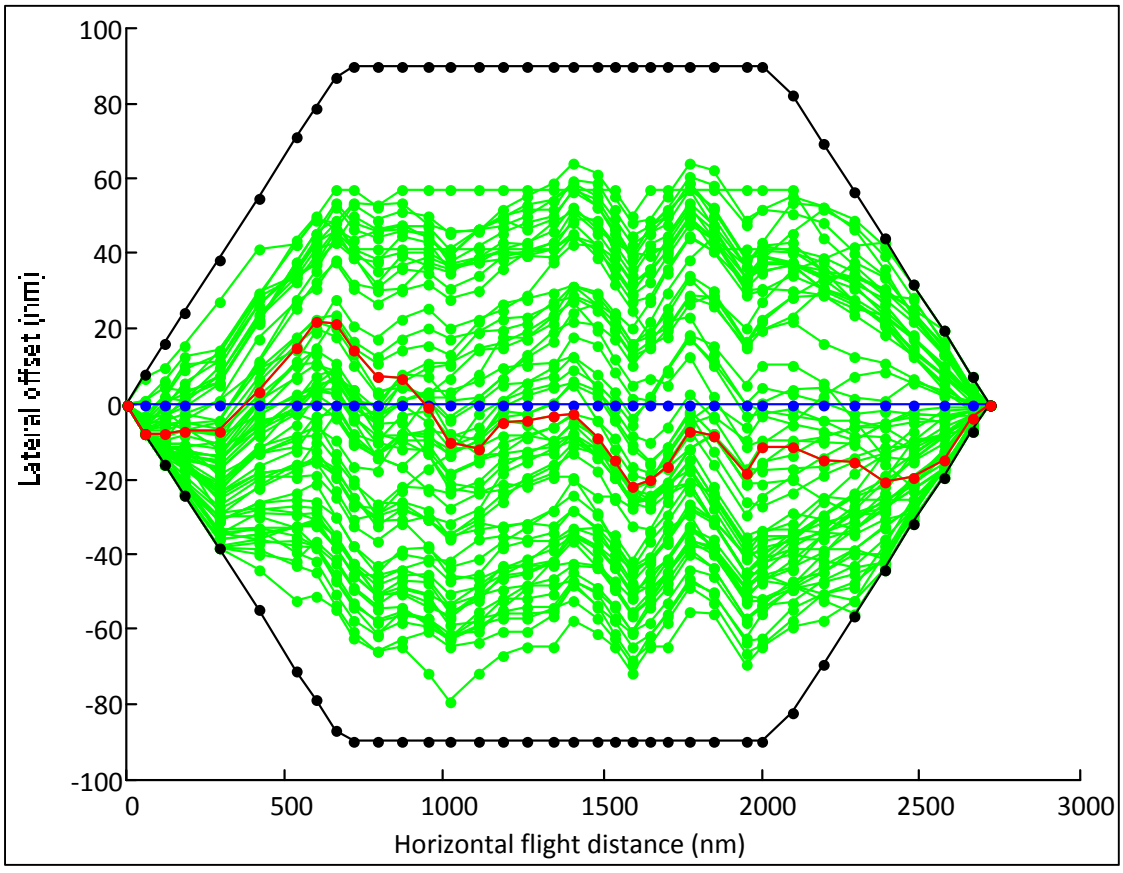

Figure 6.22 Lateral offset of the optimized trajectory (red), around the reference trajectory (blue) and the other candidate trajectories (green)

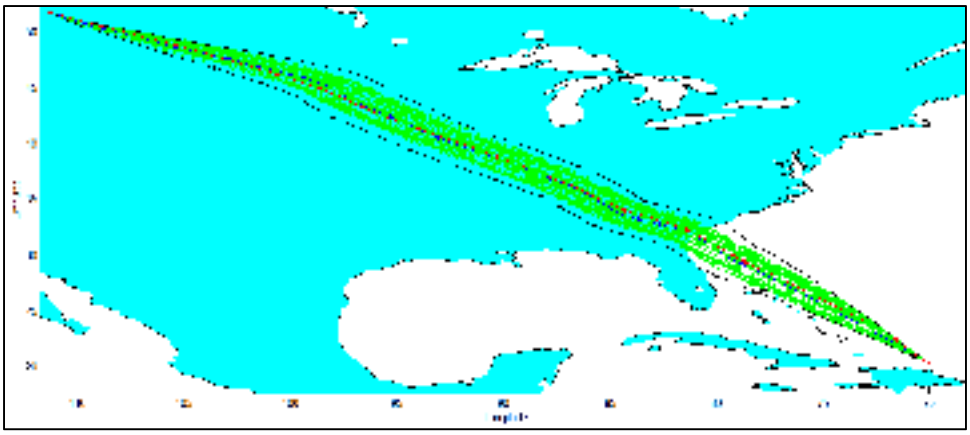

Figure 6.23 Lateral optimized trajectory (red), around the reference one (blue) from the ToC to the ToD

Figure 6.22 shows that the optimal lateral trajectory (in red) separated from the reference trajectory (in blue) to a maximal distance of around 70 nautical miles. This separation can be explained as the existence of more favorable winds in that area. The rest of the candidate trajectories exploring the search space can be seen in green. These same trajectories can be seen in Figure 6.23 over the map. It can be seen that the optimal trajectory proposed the flight to be shifted towards the South.

Figure 6.24 shows a comparison between the altitudes followed by the reference flight plan and the optimized trajectory provided by the algorithm. It can be seen that three step climbs were executed in the initial flight plan in blue. The optimized trajectory in red only required one step climb. It can also be observed that the optimization algorithm (the change of altitude from 33,000 ft to 35,000 ft suggests that the ToC should be at an altitude of 39,000 ft instead of 33,000 ft).

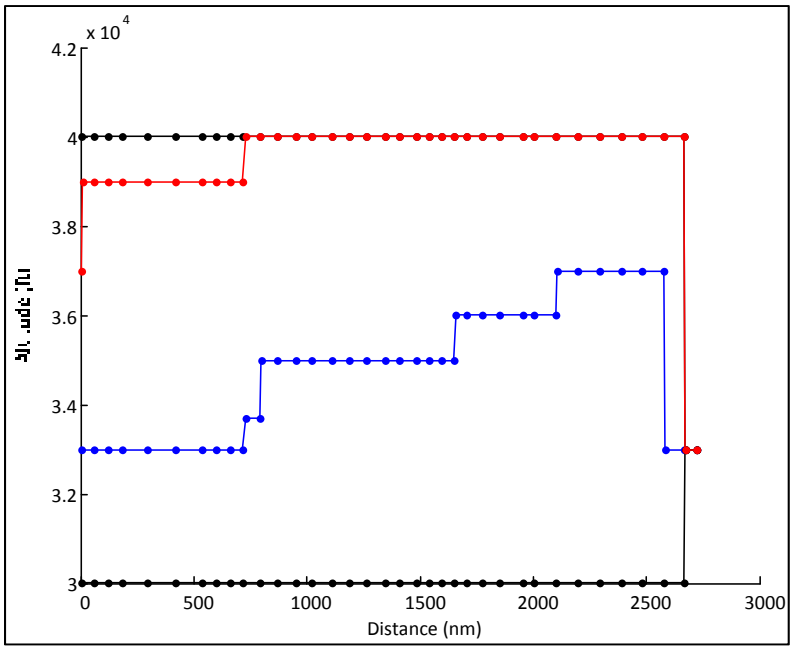

Figure 6.24 Optimized vertical trajectory (red), and original trajectory (blue)

Figure 6.25 provides the Mach number variation within 0.05 steps as in the numerical performance model. The Mach number remains fairly constant and close to the reference Mach number (blue line). It can be noticed that the Mach number tends to increase with a high deceleration at the end. This deceleration is proposed by the algorithm due to the RTA constraint.

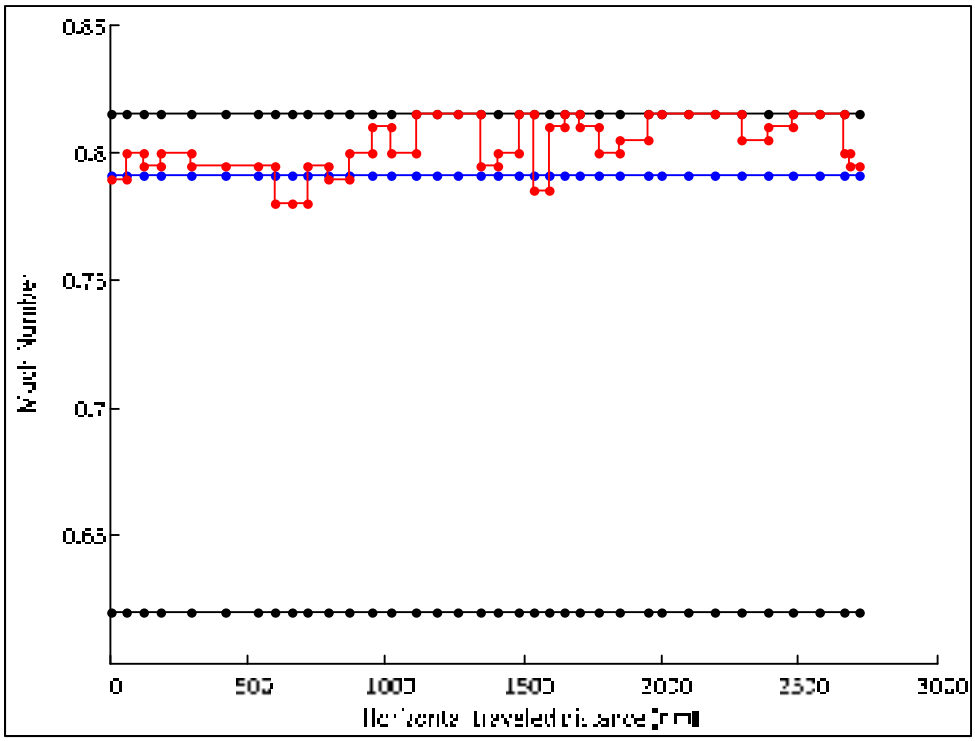

Figure 6.25 Speed versus distance variation for the optimized speed (red), and original speed (blue)

6.3.4 Multiple Flights Fuel Reduction

The fuel consumption of 10 real flight trajectories was compared against the fuel consumption of 10 trajectories delivered by the optimization algorithm. All flights were evaluated using the same commercial aircraft. Results can be found in Table 6.2.

For these flights, the number of iterations was selected to be 6,000, the number of employed bees was of 64, the number of on-looker bees was of 64, and the fail counter was set to be 200. The RTA for each flight was selected to be the time at which the original flight reached the ToD. The RTA tolerance was arbitrary selected to be +/- 30 seconds.

As seen in Table 6.2 Fuel consumption for different optimized flights, the total average fuel consumption was of 3.9% while respecting the imposed RTA. These reductions were obtained for a better selection of altitudes, a better selection of trajectories in the lateral

dimension (avoiding headwinds and taking advantage of tailwinds), and a good selection of Mach numbers that fulfilled the RTA constraint.

Table 6.2 Fuel consumption for different optimized flights

Departure	**Arrival**	**Date**	**Optimized Flight Cost (kg)**	**Original Flight Cost (kg)**	**Flight Cost Reduction (%)**	**Optimized Flight Time (s)**	**Original Flight Time (s)**	**RTA (s)**
Edmonton	Punta Cana	13 Nov. 2015	20,172.6	20,620.4	2.22%	22,361	22,360	1
London	Toronto	23 Nov. 2015	25,729.9	27,761.1	7.32%	28,103.1	28,102	1.1
Edmonton	Punta Cana	20 Nov. 2015	22,138.9	22,602.7	2.05%	22,758.1	22,758	0.1
Istanbul	Madrid	1 Nov. 2015	89,15.57	90,16.04	1.13%	12,117.5	12,130	12.5
London Gatwick	Toronto	13 Apr. 2016	22,978.5	24,203.3	5.33%	25,858	25,857	1
Lisbon	Boston	13 Apr. 2016	19,128.8	19,769.9	3.35%	24,228.8	24,248	19.2
Ponta Delgada	Toronto	13 Apr. 2016	17,333.4	17,460.6	0.73%	22,106.3	22,104	2.3

Table 6.2 Fuel consumption for different optimized flights (continue)

Departure	Arrival	Date	Optimized Flight Cost (kg)	Original Flight Cost (kg)	Flight Cost Reduction (%)	Optimized Flight Time (s)	Original Flight Time (s)	RTA (s)
Ponta Delgada	Toronto	18 Nov. 2015	19,457.6	20,159.1	3.61%	21,960.8	21,964	3.2
Calgary	Cancun	18 Nov. 2015	13,888.2	14,921.3	7.44%	15,011.5	15,006	5.5
London	Toronto	18 Nov. 2015	25,324.2	26,827.9	5.94%	27,165.1	27,177	11.9

6.4 Conclusion

In this paper, the Artificial Bee Colony (ABC) metaheuristic algorithm was implemented to find the optimal 4D reference trajectories by taking into account the influence of the weather. Fuel savings were found for all flight tests. The ABC algorithm adapted to the problem of fuel consumption reduction provided very good results expressed in terms of fuel burn. Comparing against a flown flight plan, the algorithm was able to provide a trajectory that optimized around 5% of the flight cost in terms of fuel burn. For this trajectory, the Required Time of Arrival (RTA) constraint was respected within the imposed accuracy of ± 30 seconds. The delivered trajectories did not present high variation in the lateral and the vertical dimensions, i.e. no sudden changes in direction or altitudes occurred.

The algorithm required a short computation time of maximum of 5 minutes to find a really good trajectory. This fast time is due to the efficiency of the ABC algorithm, and to its optimization. If the calculation time is an issue, the number of iterations can be reduced, the amount of fuel saved would also be reduced, and the trajectory would not be as “smooth”. However, if executing computing time is not an issue, the ABC algorithm proposed in this

paper is able to calculate an optimal trajectory that is as smooth as possible, but the number of iterations would be higher.

The aeronautical certification authorities are not yet ready to accept the use of metaheuristics algorithms on airborne devices due to their stochastic nature. For this reason, the ABC algorithms will not be implemented on FMSs in the short term. However, this article shows the potential of computation time reduction offered by the ABC algorithm in comparison to deterministic methodologies, which in some cases may take many hours to solve. Probably, over time, metaheuristic algorithms will gain enough confidence by the authorities and they will be implemented in airborne devices. Meanwhile, this algorithm may also be able to be implemented in auxiliary devices such as electronic tablets, and/or in ground systems.

As future work, more traffic data could be considered in trajectories optimization in order to be able to negotiate the possible optimized route with other aircraft routes. The terminal flight phases such as taxi and take-off have not yet been considered in this algorithm. As future work, after the consideration of the terminal flight phases; an algorithm could be developed for finding the shortest path for the aircraft from gate to gate.

CHAPTER 7

3D AND 4D AIRCRAFT REFERENCE TRAJECTORY OPTIMIZATION USING THE ANT COLONY OPTIMIZATION

Alejandro Murrieta-Mendoza, Antoine Hamy, Ruxandra Mihaela Botez

École de Technologie Supérieure / Universite du Quebec, Montreal, Canada
Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticty

This article was accepted for publication to the AIAA Journal of Aerospace Information Systems on August 2017

Résumé

Une méthodologie d'optimisation de la trajectoire de référence de l'avion inspirée de l'optimisation des colonies de fourmis est utilisée dans cet algorithme pour trouver la trajectoire la plus efficace en termes de consommation et de coût du vol pendant la phase de croisière. Les conditions météorologiques sont prises en compte dans le calcul de la trajectoire la plus économique. L'algorithme est conçu en deux étapes consécutives. Tout d'abord, la trajectoire de référence est optimisée en 3D. Ensuite, la combinaison la plus économique de nombres de Mach répondant à la contrainte de temps d'arrivée requis sur cette trajectoire 3D est trouvée créant une trajectoire de référence 4D. Différents tests de simulation consistent à créer des trajectoires suivant une trajectoire géodésique à altitude fixe, ainsi que des vols réels. Les simulations ont révélé que l'algorithme développé était capable de trouver la trajectoire la plus efficace et que le coût du vol était 6,82% plus économique que la trajectoire de référence géodésique. En outre, les tests ont montré que l'algorithme de la colonie des fourmis pouvait trouver une trajectoire 4D proche de la trajectoire du plan de vol réel, offrant une moyenne d'optimisation de 0,91%. Des études ont montré que la réalisation d'une trajectoire 3D satisfaisant à la contrainte d'arrivée requise entraînait une perte importante de 2,4% d'optimisation en raison des changements de nombre de Mach.

Abstract

A methodology of aircraft reference trajectory optimization inspired by the Ant Colony Optimization is used in this paper to find the most efficient trajectory in terms of fuel burn and flight cost during the cruise phase. Weather conditions are taken into account in computing the most economical trajectory. The algorithm is designed in two consecutive stages. First, the reference trajectory is optimized in 3D. Then, the most economical combination of Mach numbers that fulfills the Required Time of Arrival constraint over that 3D trajectory is found, creating a 4D reference trajectory. Different simulation tests consisted of trajectories following a fixed altitude geodesic trajectory, as well as of real as flown flights. Simulations revealed that the Ant Colony Algorithm was able to find the most efficient trajectory and the flight cost was 6.82% more economical than the geodesic reference trajectory. Moreover, tests showed that the Ant Colony Algorithm was able to find a 4D trajectory close to the real flight plan trajectory providing an optimization average of 0.91%. Studies showed that making a 3D trajectory fulfilling the Required Time of Arrival constraint led to an important loss of 2.4% of optimization due to the Mach number changes.

7.1 Introduction

Air transportation has become more and more important to modern society. Traveling long distances is impractical by land or sea as it might take several days or even weeks to arrive at a given destination. Aircraft, in the other hand, provides the comfort of traveling those same distances in a smaller amount of time. This is especially important not only for people, but also for goods with a short expiration date, or other products that, due to their importance (such as medicines, biological products, and some machines/machine parts), should get to their destinations quickly. In the year 2014, around $ 15.3 billion of goods were transported by air (ATAG, 2016). Due to these and to other reasons beyond the scope of this paper, air traffic has increased and is expected to increase even more in the coming years. This growth is expected to be especially pronounced in regions under development such as Asia and Latin America.

This air traffic increase, while good for the aeronautical industry, brings new challenges to the already crowed air space. The Air Traffic Management (ATM) is responsible for providing each aircraft enough separation between itself and other aircraft to safely execute its flight, thus it must be find a way to improve the procedures for managing airspace, as the current arrangements are already near their capacities.

To provide this required separation, ATM requires each aircraft's crew to submit its flight plan intentions, which should contains the aircraft's trajectory in terms of geographical coordinates, altitudes, and speeds.

For commercial aircraft, this flight plan is not submitted directly by the crew. Airlines have specialized teams which compute the most economical flight plan in terms of cost (fuel and time) before take-off and then submits it to ATM. These flight plans are normally based on pre-defined routes called *airways*. Airways for commercial aircraft are normally defined above 18,000 ft. According to regulations, westbound flights are normally flown at odd altitudes, and eastbound flights are flown at pair altitudes. If this flight plan is accepted by ATM, it is then given to the crew and it becomes the flight reference trajectory that the aircraft should fly. The crew introduces this flight plan to an avionics device called a Flight Management System (FMS).

The FMS manages the flight and, in some cases, is able to optimize the flight reference trajectory while airborne (Collinson, 2011). The optimization carried out in a FMS normally regards the vertical reference trajectory. These trajectories provide the speeds and altitudes where the aircraft should fly to minimize the fuel requirements.

The flight plans can be changed while airborne due to obstacles, weather changes, or if the aircraft would need to access a more economical trajectory. Any changes (e.g. in altitude, indicated air speed, direction, destination airport, etc.) while airborne should only be executed after the aircraft crew has received vocal authorization from Air Traffic Control (ATC). Before providing such authorization, a flight controller analyzes the airspace and verifies the safety of such a flight plan change (Field, 1985).

In spite of the airlines' interest in reducing flight costs, regulations, traffic, and airline policies do not guarantee that aircraft fly at their optimal speed, altitudes, and trajectories (Luke Jensen et al., 2014; Luke Jensen et al., 2013; Luke Jensen et al., 2015; Turgut et al., 2014). This leads to unnecessary fuel burn which not only increases flight costs, but it adds to environmental degradation. Carbon dioxide (CO_2), oxide nitrogen (NOx), and hydrocarbons are among the polluting particles emitted by aircraft engines. During the cruise phase at high altitudes, NOx emissions are believed to deplete the ozone layer (Crutzen, 1970), and CO_2 is believed to contribute to global warming. Different emissions models were presented in (Sabatini et al., 2010).

A recent study suggests that CO_2 has caused a negative change in global wind patterns, thereby incrementing flight times and thus fuel requirements (Williams, 2016). Today, the aeronautical industry releases 2% of the global CO_2 emissions (ICAO, 2010). The expected increase in air traffic will potentially increase this pollution share. To address this issue, the aeronautical industry has set the goal of reducing CO_2 emissions by 2050 to half of those recorded in 2005.

Improvements in engine performance, aircraft weight reduction, aerodynamic morphing analyses, and aircraft operations have proved to reduce fuel consumption (McConnachie et al., 2013). Reference trajectory optimization has also been shown to reduce fuel consumption. One notable case of success in operations and reference trajectory optimization is the Continuous Descent Approach (CDA). For the CDA, the descent is performed following a constant descent slope with engines set at IDLE. According to different studies, the CDA has proved to reduce both fuel consumption and noise levels at several airports (Kwok-On et al., 2003; Kwok-On et al., 2006; Sprong et al., 2008). Reducing executing the missed approach procedure is also of importance as it has been shown that it is an expensive procedure in terms of fuel and pollution (Murrieta-Mendoza, Botez, et al., 2016).

The CDA success has encouraged the development of optimization algorithms for different flight phases, while respecting current ATM constraints. Some of these algorithms are designed to be used by ground equipment before takeoff, and others while airborne.

Some of the efforts to develop algorithms are listed here. Using the single optimal control technique, the throttle setting was used as the control input to compute the optimal speed for a given fixed cruise altitude to optimize the flight path (Pargett & Ardema, 2007). Optimal control following current ATC constraints such as step-climbs and constant Mach numbers was explored in (Valenzuela & Rivas, 2014). The golden search section was applied in (R.S. Felix Patron et al., 2013) to find the optimal altitude for short flights. A fuel burn estimator was used to compute the optimal speed and altitude for a fixed altitude flight in (B. Dancila et al., 2013). An algorithm based on reducing the search space to quickly converge to the optimal solution was developed in (Jocelyn Gagné et al., 2013), to which a heuristic algorithm applied on the cruise phase was later added to further reduce the search space (Murrieta-Mendoza & Botez, 2014b). An algorithm based on branch & bound was developed in (Murrieta-Mendoza, Beuze, et al., 2015) to compute the vertical reference trajectory for a complete flight. Search space reduction techniques and a branch and bound approximation were developed in (Alejandro Murrieta-Mendoza & Ruxandra Mihaela Botez, 2015) to reduce the computation time while providing good reference trajectories. Genetic Algorithms were implemented for the same purpose in (Roberto Salvador Felix Patron et al., 2013), and later, used to find the most convenient waypoints at which to execute the step-climbs or step-descents (Murrieta-Mendoza, Félix-Patrón, et al., 2015).

Other algorithms have been developed that reduce the flight cost by not following airways, but instead by using a concept called “free flight”. This new concept could impact FMS functions, including memory and processor requirements (Schoemig, Armbruster, Boyle, Haraldsdottir, & Scharl, 2006). These algorithms are based on looking for wind currents that will shorten the flight time, as in (Roberto S. Félix-Patrón et al., 2014) where Genetic Algorithms proved to be effective in finding the shortest trajectory in time for a reduced number of waypoints by taking into account the weather. Dijkstra’s algorithm implemented

in (Murrieta-Mendoza & Botez, 2014a) was able to find the optimal trajectory, by taking into account not only flight time, but also the fuel consumption. Dijkstra's algorithm has also been used to avoid obstacles for a general aviation aircraft (Rippel et al., 2005). Floyd-Warshall algorithm was also implemented to reduce the flight time (Murrieta-Mendoza, Romain, & Botez, 2016). Trajectory optimization are also of interest for the emerging unmanned aerial vehicle industry as many algorithms have been developed to optimize the trajectory of these new aircraft as in (Wilburn et al., 2013a) where the Dijkstra's algorithm was used to avoid high terrain.

Trajectories can be optimized first in the lateral dimension, and then in vertical dimension over the previously found lateral dimension in order to obtain an optimization in 3D (latitude, longitude, and altitude) as in (Murrieta-Mendoza, 2013; Ng et al., 2014). Trajectories can also be simultaneously optimized using Genetic Algorithms as shown in (Roberto S Félix-Patrón et al., 2014; Roberto Salvador Félix-Patrón & Botez, 2015). The A* algorithm was used to find the optimal trajectory simultaneously in both dimensions and it took also into account object avoidance (Dicheva & Bestaoui, 2014). The Dijkstra's algorithm was later used to optimize the vertical and the lateral reference trajectory (Murrieta Mendoza, Mugnier, & Botez, 2017).

Future commercial aircraft systems will be based on what it is called Intended Based Operations (IBO) or Trajectory Based Operations (TBO). During these operations, aircraft communicate with each other and with a central ATC to negotiate their most economical trajectories. To be able to do this operations and communications, future trajectories will require a new dimension in addition to the 3D (latitude, longitude, and altitude) in the form of a Requited Time of Arrival (RTA) constraint assigned to a set of waypoints. These trajectories are also known as 4D trajectories. The systems and the procedures are already being deployed in different programs such as the Next Generation Air Transportation System (NextGEN) in North America and SESARS in Europe. Trajectory optimization algorithms and air traffic management systems focusing in these concepts have been proposed paying special attention to the whole flight and the Terminal Maneuvering Area (Gardi, Sabatini,

Kistan, Lim, & Ramasamy, 2015; Gardi, Sabatini, & Ramasamy, 2016; Ramasamy, Sabatini, Gardi, & Liu, 2013).

Algorithms have been developed to optimize the 4D trajectory. Lidén was one of the first researchers in this field (Sam Liden, 1992a) . Dynamic programming with neural networks were later developed to solve this problem in (Hagelauer & Mora-Camino, 1998). This concept was utilized in (Franco & Rivas, 2011) as an optimal control problem where the RTA was taken into account as a cost for arriving too late or too early at the destination. Optimal Control considering the constraint of avoiding contrail formation with the goal of finding the most economical flight was developed in (Soler-Arnedo et al., 2013). Multi objective optimization was implemented taking into account future communication, navigation, surveillance and future avionics. A new dynamic programming concept to solve this problem was proposed in (Miyazawa et al., 2013). Artificial bee colony was implemented for the vertical reference trajectory optimization fulfilling the RTA constraint (Murrieta-Mendoza, Bunel, & Botez, 2016).

The originality of this paper lies in its study of the optimization potential of a reference flight trajectory optimization algorithm inspired by the metaheuristic Ant Colony Optimization (ACO) approach for 4D trajectory optimization. In fact, the ACO has not been explored in the literature for a commercial long-haul aircraft during the cruise phase using a numerical performance model and an imposed RTA constraint. This paper also provides an analysis of the flight cost optimization consequence of fulfilling the RTA constraint, as less efficient Mach numbers have to be selected for the 4D optimal trajectory compared to the 3D optimal trajectory, where speeds can be *freely* selected.

The ACO was selected for this investigation because metaheuristic algorithms may be capable of finding good solutions for dense search spaces such as those required for a good flight trajectory optimization. The objective of this paper is to develop a 4D trajectory optimization algorithm using the ACO. The fuel cost reduction potential of the trajectories

studied by the algorithm was studied for fixed altitude flights and compared to real flight plans obtained from the open source site *flightaware* ®.

This paper is organized as follows: the fuel consumption model is explained, then a description of the methodology is given for the computation the flight cost using this model; after, the weather information model is defined. The general ACO concept is then presented, followed by the ACO implementation in the 4D trajectory optimization. Finally, the optimization potential is analyzed, as well as the results obtained from simulations and the paper ends with conclusions.

7.2 Numerical Models and the Search Space

The algorithm developed in this paper only considers the cruise phase of a commercial flight. This phase begins at the Top of Climb (ToC) and ends at the Top of Decent (ToD). Different parameters such as the aircraft's altitude, the aircraft`s direction, and the Mach number can change along the flight in order to reduce the fuel burn and fulfill the RTA constraint. The free flight concept is considered, which means that the aircraft can change its directions and altitudes at any waypoint.

7.2.1 Fuel Consumption: The Numerical Performance Model

The fuel consumption was obtained from a numerical performance model created from experimental flight data. This model consists of different tables for each different flight phases. For this paper, the fuel burn information for cruise and climb phases (due to step-climbs) are of interest. Figure 7.1 represents the model for the cruise phase. The required inputs are the Mach number, the aircraft`s gross weight (kg), the international standard atmosphere (ISA) temperature deviation, and the flying altitude (ft.). The output is the fuel flow (kg/h).

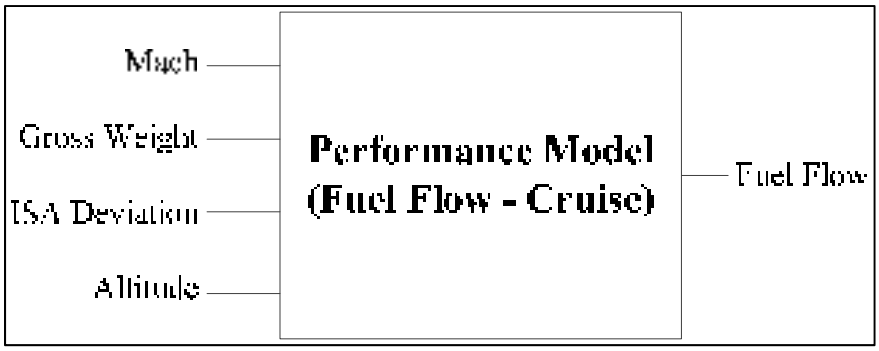

Figure 7.1 Inputs required to obtain fuel flow during cruise from the performance numerical model

Figure 7.2 represents the model for the climb phase. The inputs are the same as in the cruise phase in Figure 7.1, with the addition of the center of gravity. The outputs for the climb phase are the fuel burn required (kg and the required horizontal distance (nm) to climb at the destination altitude.

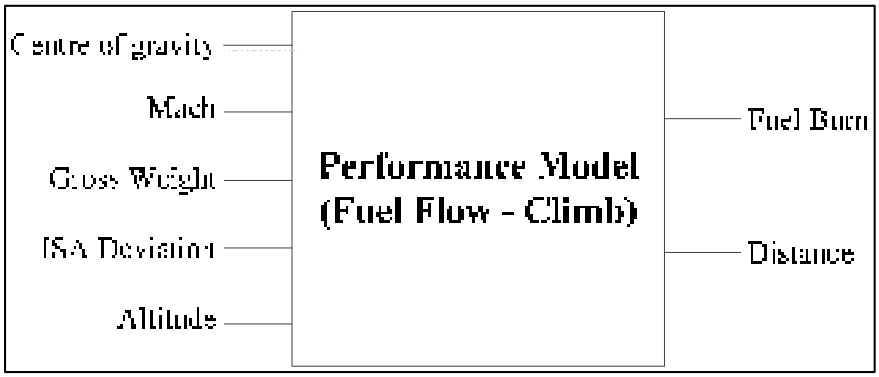

Figure 7.2 Inputs required to obtain fuel burn during climb from the performance numerical model

For the cruise phase (Figure 7.1), the *Fuel Flow* is directly provided by the numerical performance model for the aircraft's flight condition. However, the fuel burn and the horizontal distance traveled during the climb phase (when executing a step-climb) are given differently. The fuel burn required to climb to a given altitude is given by the numerical performance model from a reference altitude. For example, if the current cruise altitude is 33,000 ft, and if a step climb of 2,000 ft is required (to reach 35,000 ft), the numerical

performance model does not provide the fuel burn required to climb from 33,000 ft to 35,000 ft, but from the reference altitude to 35,000 ft. The procedure to compute the fuel burn required to climb from a reference altitude to a desired altitude is explained next using Figure 7.3.

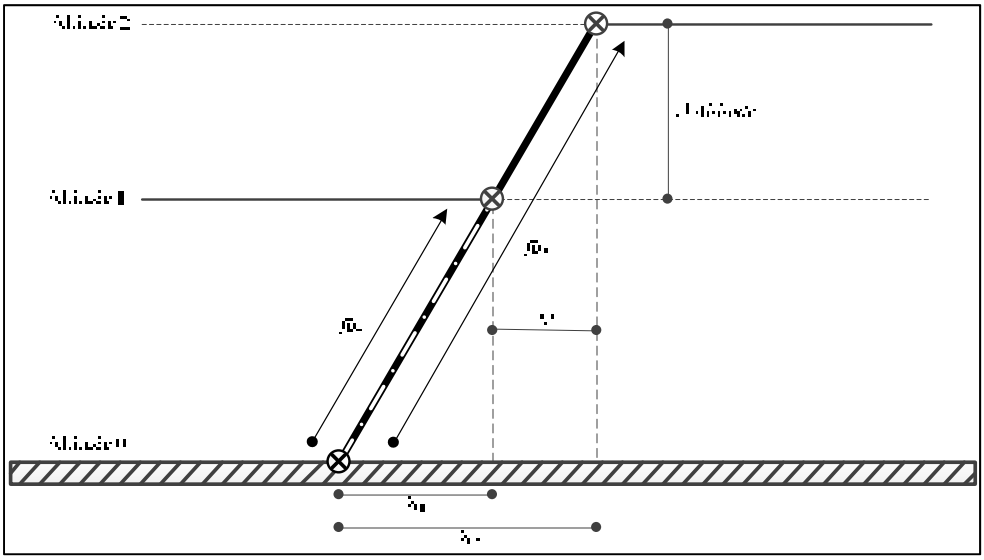

Figure 7.3 Fuel burn and horizontal travelled distance

In Figure 7.3, *Altitude 1* is the current cruise altitude and *Altitude 2* is the targeted altitude (altitude after the step-climb). *Altitude 0* is the reference altitude. The values fb_1 and fb_2 are directly obtained from the performance model. They represent the required fuel burn to climb from *Altitude 0* to *Altitude 1* and from *Altitude 0* to *Altitude 2*, respectively. The difference provides the fuel required to climb from *Altitude 1* to *Altitude 2*, as expressed by Eq. (7.1).

$$Fuel\ Burn_{clb} = fb_2 - fb_1 \tag{7.1}$$

A similar operation is executed with the required horizontal distance (D) to climb from *Altitude 1 to Altitude 2*. X_1 and X_2 are the horizontal distances required to climb from *Altitude 0* to *Altitude 1*, and from *Altitude 0* to *Altitude 2,* respectively. These distances are obtained from the performance model, and thus the D required to climb from the current altitude to the desired altitude can be computed as shown in Eq (7.2).

$$D = X_2 - X_1 \tag{7.2}$$

7.2.2 Fuel Burn Computation

The discrete nature of the numerical performance model means that information is only provided for some specific flight conditions. However, flight conditions are given within the information available in the numerical performance model. For this reason, linear interpolations in the information provided by the numerical performance model are executed to obtain its required outputs.

Interpolations are normally executed for Gross Weight and ISA temperature deviation, as these parameters tend to be different than the exact available inputs. The Gross Weight diminishes as the aircraft burns fuel, and the ISA temperature changes as weather changes along the route. A full explanation of the interpolation process utilized to compute the flight cost can be found in (Alejandro Murrieta-Mendoza & Ruxandra M. Botez, 2015). However, a short explanation is provided next.

The Mach number and the altitude are already known, as they always correspond to the values available in the numerical performance model. At each waypoint, the Gross Weight and the Temperature around the aircraft are obtained. With this information, the values in-between the real ISA deviation temperature inputs and the input values in-between the real gross weights available in the performance numerical model are identified. With these values, four different fuel flows are obtained: $FF\_ISAT_1\_W_1$, $FF\_ISAT_2\_W_1$, $FF\_ISAT_1\_W_2$, and $FF\_ISAT_2\_W_2$. These variables are read as follows: $FF\_ISAT_1\_W_1$ is the fuel flow for the lower ISA deviation temperature identified in the numerical performance model for the lower Gross Weight identified in the numerical performance model. $FF\_ISAT_2\_W_2$ is the fuel flow for the higher ISA deviation temperature identified in the numerical performance model for the higher weight identified. Interpolations are then performed, as shown in Figure 7.4.

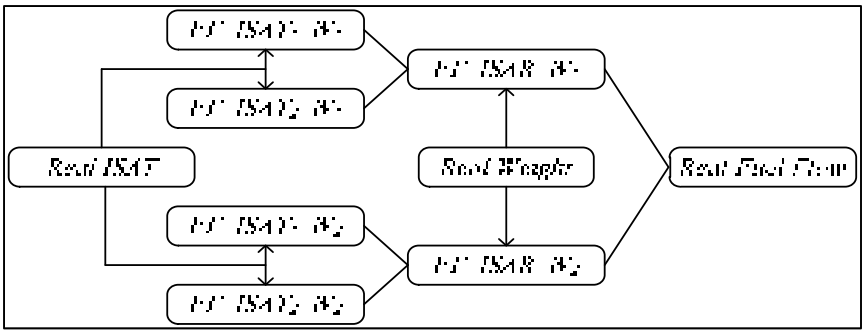

Figure 7.4 Interpolation schema for the fuel flow interpolation

Figure 7.4 shows how the fuel flow for the ISA Temperatures Deviation ($FF\_ISAT_1\_W_1$ and $FF\_ISAT_2\_W_1$) for Weight 1 are interpolated for the real ISA Temperature Deviation (*Real ISAT*). The same is done for the fuel flow for the ISA Temperature Deviations ($FF\_ISAT_1\_W_2$ and $FF\_ISAT_2\_W_2$) for Weight 2. The fuel flow obtained for the real ISA deviation temperature for each weight are interpolated for the *Real Weight*, thereby obtaining the Real *Fuel Flow*. A similar procedure is executed to compute the fuel burn and the horizontal distance traveled during climb.

Depending on the flight phase, the fuel consumption and the flight time are computed differently. During the fixed cruise altitude, the model provides the fuel flow, and the segment distance is proposed by the user. For this algorithm, the *total distance* is normally divided into equidistant segments of 25 nm.

The *Flight Time* between two waypoints is computed using Eq. (7.3), where *Distance* is the length between two waypoints and Ground Speed (*GS*) is the aircraft speed relative to the ground.

$$Flight\ Time = \frac{Distance}{GS} \tag{7.3}$$

$Fuel\ Burn_{cr}$ during cruise is computed using Eq. (7.4). The *Fuel Flow* is obtained from the numerical performance model, and the *Flight Time* is computed with Eq. (7.3)

$$Fuel\ Burn_{cr} = Fuel\ Flow\ \times Flight\ Time \tag{7.4}$$

When a step climb is executed, the fuel required for the climb is obtained using Eq. (7.1). The time required for the climb is obtained using the distance computed in Eq.(7.2) and the *GS* between the two altitudes, thus Eq. (7.3) is used.

7.2.3 Flight Cost

For commercial aircraft, fuel consumption is not the only important parameter needed to minimize costs. *Flight Time* has real effect on expenses, as there are many operations related to time such as crew salaries, aircraft maintenance, and the need to arrive on time at a given destination to avoid causing costumers to miss connecting flights, as well as to avoid airport penalties.

The *Flight Cost* is computed using Eq. (7.5).

$$Flight\ Cost = Fuel\ Burn + CI \times Flight\ Time \tag{7.5}$$

where *Fuel Burn* corresponds to the fuel needed to perform a flight. The Cost Index (*CI*) is a parameter used to provide a cost of time in terms of fuel. This number has different values, which are normally defined by each airline. Finally, *Flight Time* refers to the time required for the aircraft to fly during the cruise phase.

7.2.4 Weather Information

To obtain a realistic trajectory of reference, the weather's influence on a trajectory has to be taken into account. Weather has an important effect on engine behavior, on the speeds of sound, and on the winds an aircraft encounters. Low temperatures make the engines more efficient in terms of fuel consumption, while high temperatures are required at higher speed

of sound. Obviously, flying with tailwinds is much more advantageous than flying with headwinds.

Typically, an airline use different weather prediction sources and creates different optimal trajectories for each prediction. However, for this research, only the prediction provided by Weather Canada was used.

Weather data was obtained from Environment Canada. The forecast is provided under the form of a grid for different isobaric pressures (pressure altitudes) and several time blocks. For the aircraft reference trajectory problem, the parameters of interest are the wind speed, azimuth, altitude pressure, and temperature. Since the aircraft is rarely at the exact available grid points, interpolations are required to obtain the weather at the exact geographical and temporal aircraft positions. Figure 7.5 graphically shows the grid with respect of the aircraft localization for a given hour, at a given set of coordinates and a given altitude.

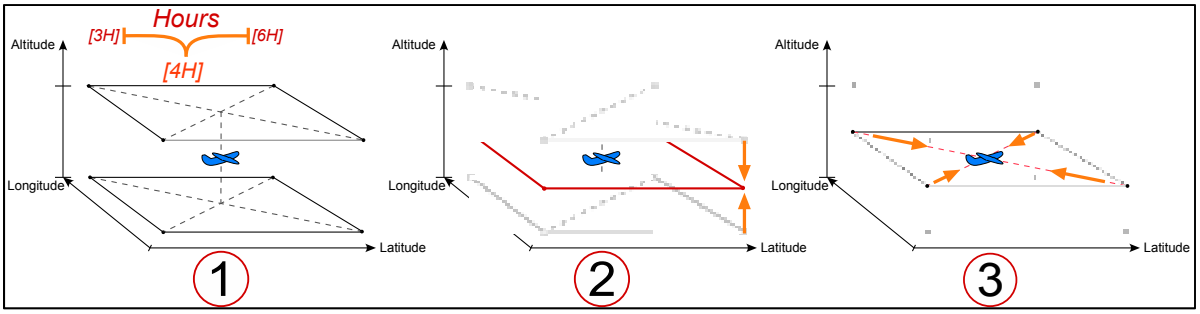

Figure 7.5 Weather information interpolation around a plane

Three steps are shown in Figure 7.5. Firstly, the weather information is interpolated in hours to find the required time (Figure 7.5-1). Secondly, an interpolation is done in terms of altitudes (Figure 7.5-2). Finally, the third step and last step represents the geographical (latitude and longitude) bilinear coordinates interpolation (Figure 7.5-3).

Once the weather at the aircraft location (latitude, longitude, altitude and time) has been determined, the temperature is used to compute the local sound speed, and by using the Mach number, the True Air Speed (*TAS*) is obtained as indicated in Eq. (7.6).

$$TAS = Mach\ Number\ \times Sound\ Speed_{Alt} \tag{7.6}$$

With the computed *TAS*, the Wind Speed (*WS*) and the Wind Azimuth (W_{Az}) obtained from the weather forecast, it is possible to compute the Ground Speed (*GS*) between two waypoints as shown in Eq.(7.7).

$$GS = TAS + WS * cos\ \times (R_{AZ} - W_{AZ}) \tag{7.7}$$

where R_{AZ} is the trajectory azimuth between two consecutive waypoints, and W_{AZ} is the wind azimuth obtained from the weather forecast. This can be seen in Figure 7.6.

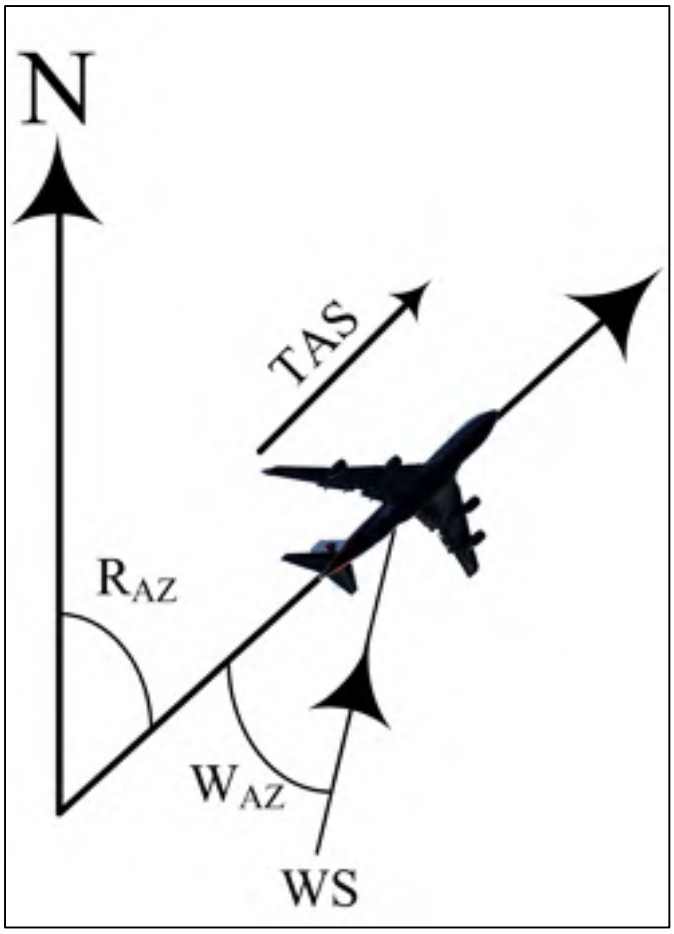

Figure 7.6 Wind effect on the aircraft ground speed

7.2.5 The Search Space

7.2.5.1 The 3D trajectory

The algorithm focuses on the lateral, vertical and time dimensions (the latter expressed by the selection of Mach number). The search space for the algorithm developed in this paper is modeled under the form of a graph $G(E,V)$ with nodes E connected with edges V. In this algorithm, the nodes (E) can be interpreted as waypoints while the edges (V) can be interpreted as connections between two consecutive waypoints. The edges' weights can be interpreted as the cost to fly between two consecutive connected nodes. Following aeronautical principles, G is unidirectional as aircraft do not fly backwards. The edge's

weights are always positive. A negative weight would mean that the aircraft is either generating fuel or time, both of which are impossible. The graph (G) is created from the beginning of the cruise phase, the Top of Climb (ToC), to the end of the cruise phase, or Top of Descent (ToD). The graph is built in 3 dimensions (3D) using reference trajectory as shown in Figure 7.7. This is in order to capture the variation of the trajectory in the lateral and in the vertical dimensions.

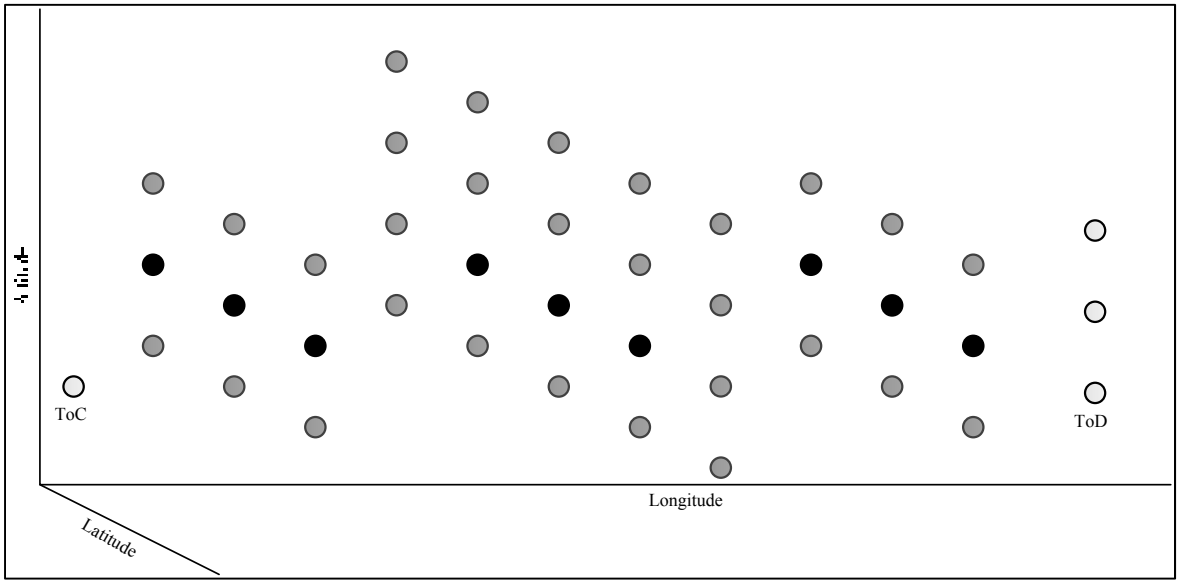

Figure 7.7 3D graph for trajectory optimization, every circle is a node (or a waypoint)

Some parameters such as altitude and turning angle must be limited in order to avoid too much deviation from the trajectory of reference, to comply with ATC constraints, and to reduce the required computation time.

For the vertical dimension, the aircraft is not allowed to descend during the cruise phase. The only possibilities in this dimension are to stay at the current altitude or to climb from it to either 2,000 ft or 4,000 ft (or to the maximal available altitude). For the lateral dimension, consecutives nodes are placed in the same direction and at +/- 15° from the actual route. The aircraft can only fly to consecutive waypoints. Figure 7.8 shows the different nodes available from the current waypoint.

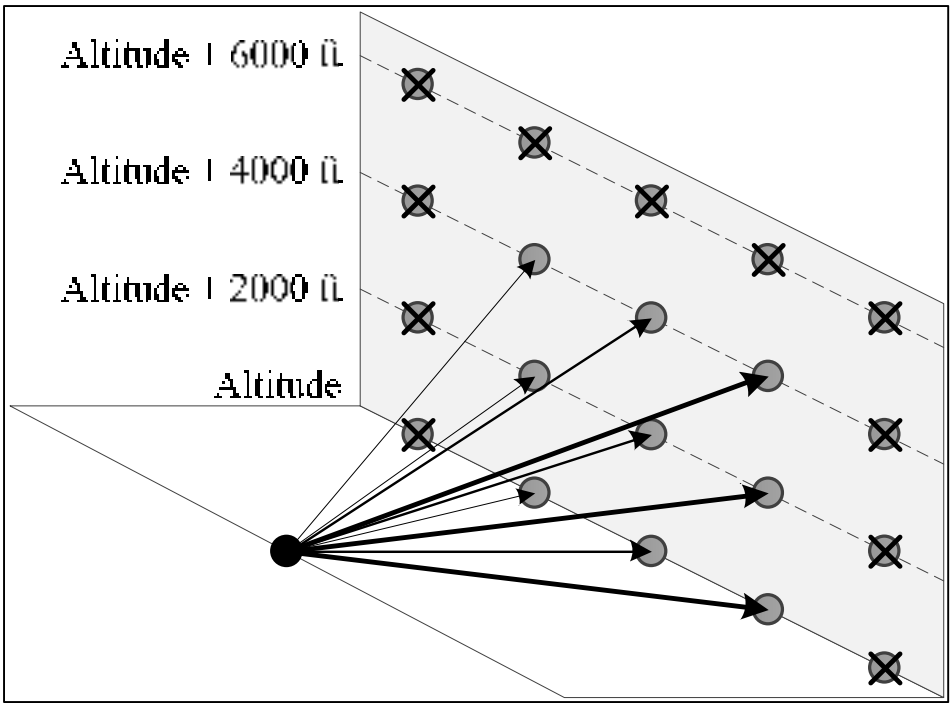

Figure 7.8 Consecutive nodes available to create a trajectory from the current waypoint

Starting from the waypoint at the left shown in Figure 7.8, there are considered only nine possible consecutive waypoints to continue the trajectory, at three different altitudes and in three possible directions. Nodes at angles greater than 15 degrees and at altitudes higher than 4,000 ft with respect to the current altitudes are not considered.

7.2.5.2 Mach number selection optimization

Once a 3D trajectory has been optimized, the algorithm will select the most economical combination of Mach numbers for a flight that fulfills the RTA.

The search space lies in a graph where the nodes are the available Mach numbers, and the edges are the cost of flying between two waypoints at that Mach number. The search space for the Mach numbers is not symmetric with respect to Mach numbers as the aircraft begins to fly at a given Mach number, and the flight can end at any other Mach numbers, as shown in Figure 7.9.

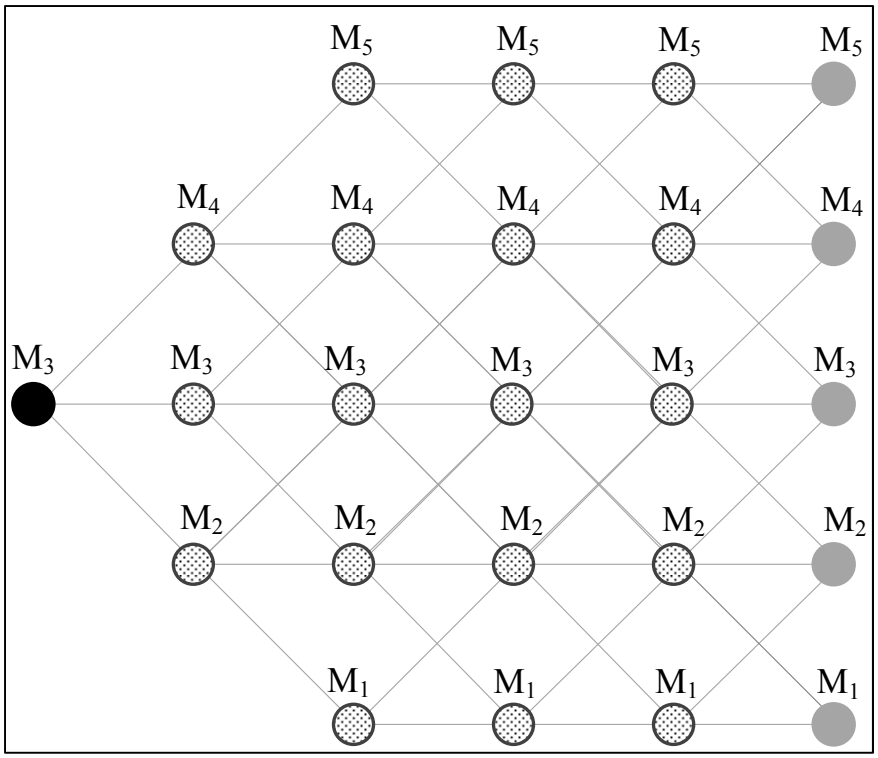

Figure 7.9 Graph of available Mach numbers in the search space

The Mach number search space also needs some computing rules. The algorithm can select from among a maximum of three different Mach numbers between two consecutive waypoints. Figure 7.10 indicates the possible Mach number selection.

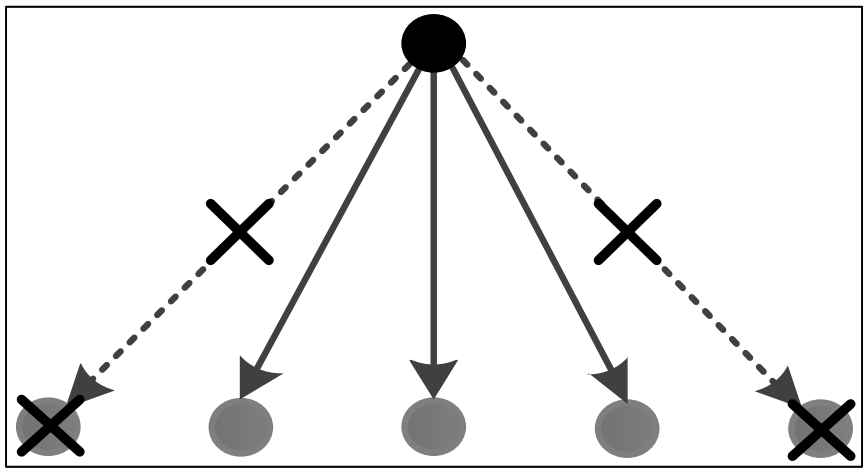

Figure 7.10 Only the three closest Mach numbers can be selected.

7.3 Introduction to the Ant Colony Optimization Algorithm

7.3.1 Bio Mimicry and Metaheuristic algorithms

Bio mimicry uses existing nature optimization solutions to solve a number of complex problems. Nature solutions are often remarkably efficient in terms of energy. Many species have developed simple, structured, and optimized solutions to certain problems as part of their adaptation to the environment. These solutions can be coded to develop an algorithm that mimics their behavior and that can be used to solve different problems related to the shortest path, population organization, scheduling, and so on. Most of these algorithms are of a metaheuristic nature. This means that there is a random behavior, and thus there is not a guarantee that they will find the same solution twice. These algorithms are considered to be metaheuristics, for this reason they have the particularity that they do not guarantee finding the optimal solution. Ants are one of the species that uses such solutions.

7.3.2 The Ant Colony In Nature

In nature, ants wander around in large open areas looking for food sources. Whenever they find a food source, they come back to the hive, stock the food they return, and come back to the food source. The trajectory found by that ant from the hive to the food source could be inefficient. It is unlikely that a single ant will find the most efficient route between the hive and the food source, but through cooperation an ant colony is able to define the most efficient trajectory.

Ants use a chemical substance called pheromone to mark the places they have visited. Ants from the same colony recognize the pheromone and are more likely to follow trajectories where they or other ants from their colony have previously visited. Pheromone has the particularity that it evaporates over time. This evaporation is useful, because if a given part of a trajectory is not used for a certain time, the pheromone lingers its concentration, and for this reason that part of the trajectory is less attractive. In the opposite sense, if a path is highly

used, the pheromone concentration increases. The more pheromone in a given trajectory, the more likely ants will follow that trajectory. It is this property that allows ants in a colony to find the most efficient trajectory between the nest and the food source. Figure 7.11 shows the ant colony organization needed to find the most efficient trajectory between the nest (*N*) and the food source (*F*). In this figure, the black dots are the *ants*, and the continuous lines represent the *pheromone*.

Firstly, ants locate a food source. Multiple ants might find different paths from the nest to the food source. These ants spread a certain pheromone on their trajectories that is shown as a continuous line in Figure 7.11 a. Next, within a certain time frame, ants on the shortest path make more round trips than others, depositing pheromone more often. Therefore, during the same time frame, the pheromone concentrations on the shortest paths are higher than on the longer trajectories as shown in Figure 7.11-b. When other ants are wandering around and find these high pheromone trajectories, they are more likely to follow them, increasing the number of ants following that trajectory, and thus increasing the pheromone concentration. Consequently, after some time, most of the ants are following the shortest trajectory, and the pheromone concentration in that trajectory is higher than in any of the other trajectories, as shown in Figure 7.11-c. However, there are still ants that explore different paths trying to find a shorter solution. This exploration is important when adapting the ants' behavior to an optimization algorithm in order to avoid staying in a local minimum (it will also increase the ants chances of finding other food sources).

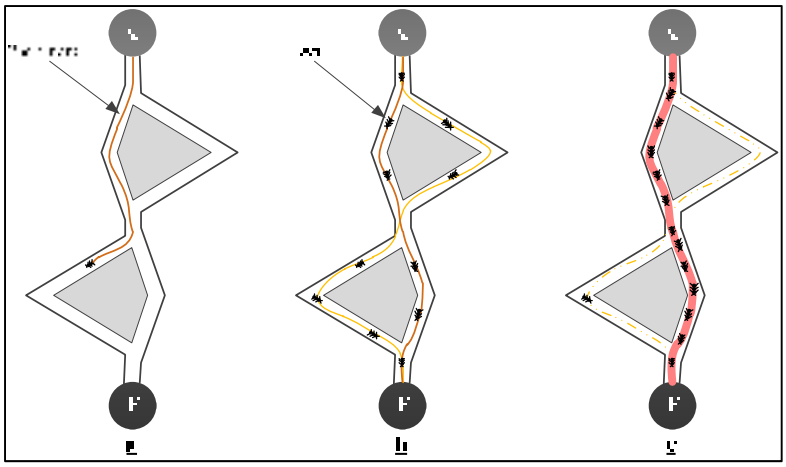

Figure 7.11 The ant colony organization in nature

7.3.3 The ACO algorithm implementation for trajectory optimization

The ant colonies' behavior can be mimicked to develop an optimization algorithm (Yang, 2010). The Ant Colony Optimization (ACO) is adapted in this research to the flight trajectory optimization problem to find the most economical trajectory in 4D for a commercial flight during the cruise phase.

The developed algorithm optimizes the trajectory in two different phases. Firstly, the ACO is used to find the most economical combination of waypoints that provides the 3D trajectory (latitude, longitude, altitude) in terms of flight cost and close to the RTA constraint. Secondly, an economical combination of Mach numbers that fulfills the RTA is determined for the 3D trajectory. The two stages of the algorithm are shown in Figure 7.12.

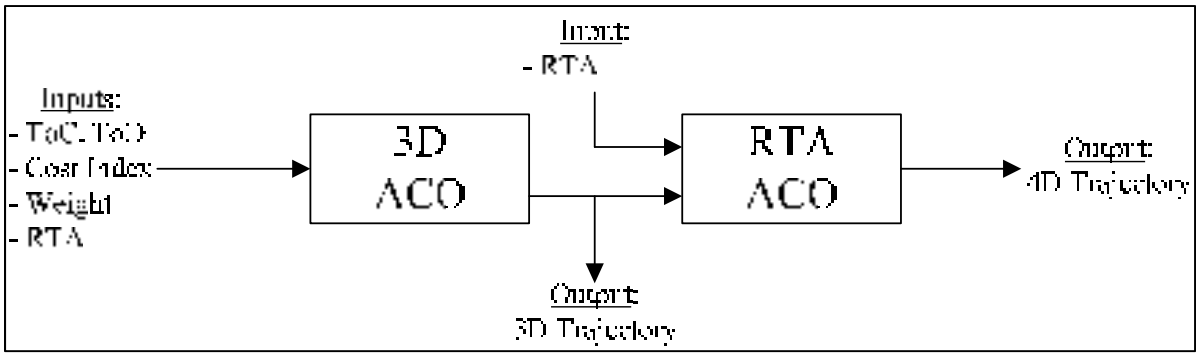

Figure 7.12 Algorithm stages: First, the ACO is used to find the optimal 3D trajectory, then the combination of Mach numbers that fulfill the RTA constraint

These stages are composed of 3 different modules which mimic the ant's behavior: explore the search space (Module 1), add pheromone in the trajectories and compute the probability that a given ant would take a given trajectory (Module 2), and the behavior of ants to select the most convenient path (Module 3). We should clarify that these three modules are executed in a different order.

7.4 3D Reference Trajectory Optimization

While computing the optimal 3D trajectory using the Ant Colony Optimization, the Mach number for each waypoint is selected from a pre-computed table available in the numerical performance model. This pre-computed Mach number provides the most economical Mach (also called "Mode Economy" or mode ECON) number taking into account parameters such as Cost Index, Gross Weight, Altitude, and ISA Temperature Deviation.

The first stage of the optimization algorithm consists of three different parts.

7.4.1 ACO First Stage: 3D – Module 1 (M1)

This first part mimics the ant behavior of creating a random trajectory to link the ant's nest (*N*) to the food source (*F*). In the optimization algorithm context, many different 3D random trajectories linking the ToC to the ToD are created. The ToC represents the nest (*N*) and the

ToD represents the food source (*F*). A certain quantity of pheromone is deposited at each edge linking two points. A random trajectory is created its 3D; the lateral and the vertical dimensions are shown in Figure 7.13.

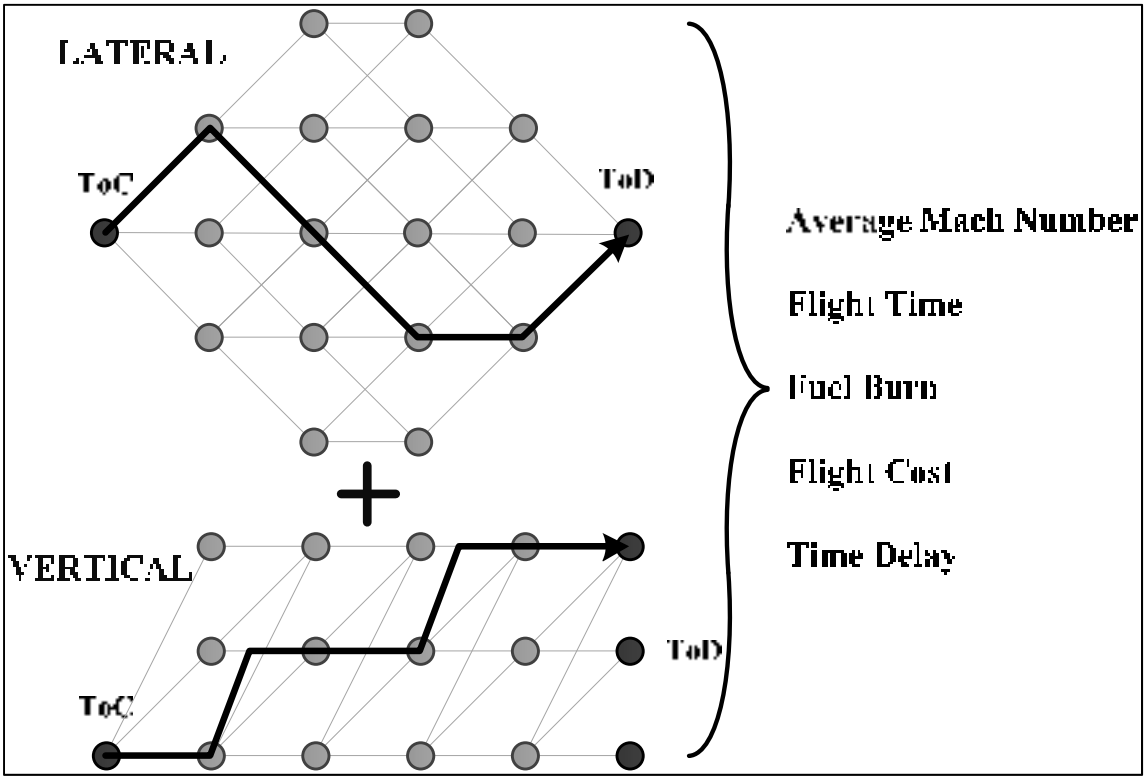

Figure 7.13 3D random trajectory creation and its parameters per trajectory

For this trajectory, different parameters are computed to be used in next Part 2 (P2) and Part 3 (P3). These parameters are: the required *Flight Time*, the required *Fuel Burn*, the *Flight Cost,* and two special parameters used later for the RTA, the *Time Delay* and the *Average Mach Number in 3D*. These last two parameters are explained next.

At every waypoint, the Mode ECON table provides the most economical Mach number for the current conditions. The average Mach number for every trajectory is computed - *Average Mach Number* (*AMN_{3D}*). This *AMN_{3D}* is the initial speed value for the RTA fulfillment part described in Section 7.3.3.2.

The *Time Delay* is the difference between the trajectory's flight time and the objective RTA, computed with Eq. (7.8). This parameter is later used Ito influence the ants' decision to select a path.

$$Time\ Delay = |RTA - Flight\ Time| \tag{7.8}$$

This module steps can be summarized as follows:

1. A random trajectory is created using the 3D trajectory.

2. Parameters of interest such as *Flight Time*, *Fuel Burn*, *Flight Cost*, *Time Delay* and *Average Mach Number in 3D* are computed for the created trajectory.

3. Module 1 ends

7.4.2 ACO First Stage: 3D – Module 2 (M2)

During a time frame, ants on the most efficient trajectory (the most economical one) will make more round trips than ants on other trajectories (more expensive trajectories). While making these trips, ants deposit pheromone, thus a higher pheromone concentration will be found on the most efficient trail. This part mimics this pheromone addition to trajectories.

The most promising trajectories are given preference by adding a significant amount of pheromone to the consecutive edges that compose these trajectories. It is possible to select the most economical trajectory (ies) so far in terms of *Flight Cost*; this parameter was computed in the above sub-section *3D - Part 1* for each created trajectory.

However, if the algorithm systematically selects only the more economical trajectories in terms of *Flight Cost*, it might spend more resources on trajectories that have difficulties fulfilling (or may not fulfill at all) the RTA constraint.

Therefore, a compromise between the *Flight Cost*, and the *likeness* to fulfill the *RTA constraint* (this *likeness* is obtained with the *Time Delay* parameter) is required. This compromise is achieved by using the parameter *P* computed in Eq. (7.9). The higher the parameter *Prob* is, the best is the compromise between the *Flight Cost* and the chances of fulfilling the RTA constraint (*Time Delay)*.

$$Prob(i) = \frac{Flight\ Cost^{-\alpha} + Time\ Delay^{-\beta}}{\sum_i (Flight\ Cost_i^{-\alpha} + Time\ Delay_i^{-\beta})} \tag{7.9}$$

In Eq. (7.9), the coefficients α and β are used to give higher importance to one parameter than to another. The algorithm's objective is to find the most economical trajectory in terms of *Flight Cost*, and therefore the parameters' relationship should be expressed as $\alpha > \beta$. The trajectories with the highest parameter *P* are selected, and therefore, a notable amount of pheromone is added to them.

The pheromone concentration rate evolves with time, which allows less promising trajectories to disappear and to benefit the most promising ones. This pheromone-level is updated using Eq. (7.10), where *Pherom3D* is a data matrix in which the pheromone rate for each edge (connection between two nodes) in the 3D graph *G* is stored, and γ_{pher} is the evaporation rate.

$$Pherom3D = Pherom3D * (1 - \gamma_{pher}) \tag{7.10}$$

This module steps can be summarized as follows:

1) The parameter *Prob* for each available trajectory is evaluated using Eq. (7.9).
2) The highest *Prob* trajectories are selected.
3) An important quantity of pheromone is added to the edges composing each trajectory in *Pherom3D*.

4) The pheromone concentration in *Pheromn3D* is updated using Eq. (7.10).

5) Module 2 ends

7.4.3 ACO First Stage: 3D – Module 3 (M3)

Ants tend to follow the path where the pheromone concentration is the highest. The algorithm emulates this behavior by building a new trajectory using only the pheromone concentration available in *Pherom3D*.

Firstly, the pheromone concentration is updated using Eq. (7.10). Then, the algorithm analyses the pheromone concentrations exclusively. This is carried out in a greedy manner. Beginning at the ToC, the consecutive edges with the highest pheromone concentration all the way up to the ToD are selected. This concept is expressed in Figure 7.14, where at each waypoint; the decision of where to go is selected based on the highest pheromone concentration (shown by the line). This new trajectory is called the *composed trajectory*.

The parameters AMN_{3D}, *flight time*, *fuel burn*, *flight cost*, and *time delay* are computed for the *composed trajectory*. Pheromone is added to the waypoints' edges of this composed trajectory, and if this trajectory is the most economical, then is selected as the "optimal trajectory".

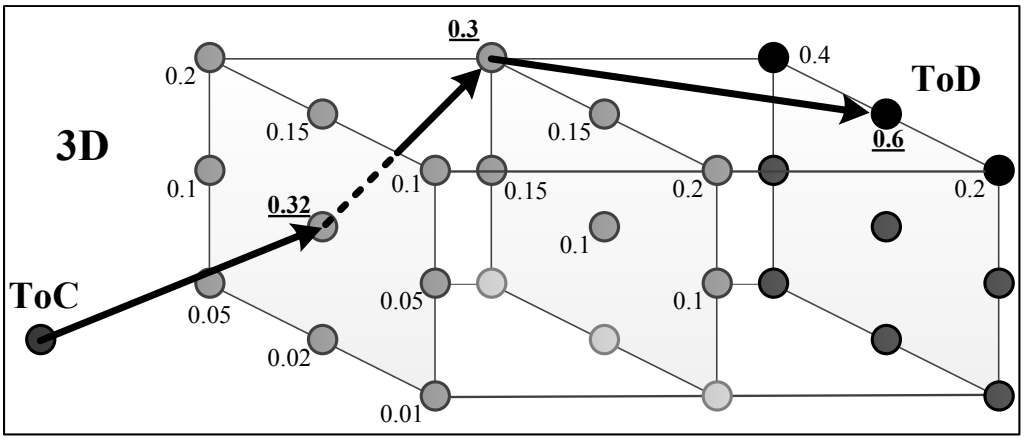

Figure 7.14 Module 3 – Route building on a 3D simplified model, following pheromone concentration only

This module can be summarized with the next steps:

1) The pheromone concentration in *Pheromn3D* is updated using Eq. (7.10).

2) Beginning at the ToD, the algorithms selects the nodes with the highest quantity of pheromone in a greedy way.

3) The composed trajectory is evaluated and it might be selected as the optimal.

4) Module 3 ends

7.4.4 Functioning

The algorithm is executed in a non-linear way. This means that modules described are not executed one after the other. The sequence is shown as a flowchart in Figure 7.15.

Firstly, Module 1(*a* in Figure 7.15) is computed *n* times in order to create a set of random base trajectories to join the ToC to the ToD. Next, Parts two and three (*b* in Figure 7.15) are used to find the first set of candidate trajectories. At this point, there is a set of trajectories to work on, and there exists the first optimal trajectory.

Then, the module one (*c* in Figure 7.15) mimics an ant looking for a new trajectory without being influenced by pheromone. This is done in order to explore more search space and avoid

local minimum. Please note that c and d (explained in the next paragraph) are executed *m* times.

Module two and three (*d* in Figure 7.15) are executed after each new ant. They are executed 4 times to add an important quantity of pheromone. This is done as it was observed that a lot of pheromone should be added to quickly converge to the optimal 3D trajectory.

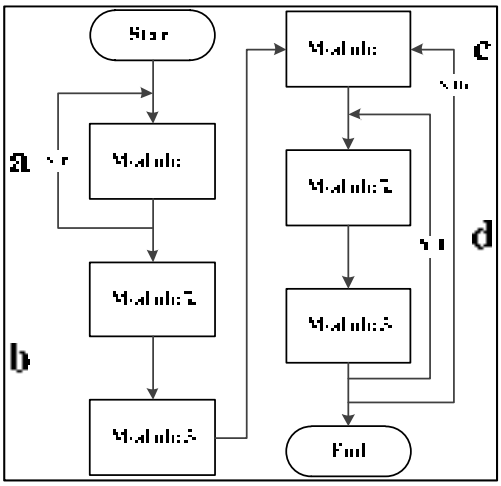

Figure 7.15 ACO functioning

The three dimension optimal trajectory is found through this ACO method. The combination of Mach numbers that fulfill the RTA constraint are then computed for the identified optimal 3D trajectory. Notice that the flowchart shown in Figure 7.15 is executed once again for the 4D trajectory described next.

7.5 4D Reference Trajectory Optimization: RTA Fulfillment

The same methodology used to compute this combination of Mach numbers is practically the same as the one used in Section 7.4 with the differences explained here. First, two important conceptual differences are explained: an initial Mach number and a different graph search, followed by slight differences in the methodology itself.

For this part of the algorithm to properly work, an initial Mach number is required to begin this part of the algorithm. An average Mach number was computed from the previous section. However, the Mach numbers (and thus the Average Mach Number in 3D - AMN_{3D}) normally do not satisfy the RTA constraint.

For this reason, the AMN_{3D} should therefore be adjusted to reduce the difference between the 3D trajectory *Flight Time* and the required RTA. This is achieved using Eq. (7.11)

$$AMN = AMN_{3D} \times \left(2 - \frac{RTA}{Flight\ Time}\right) \tag{7.11}$$

where the AMN_{3D} was obtained as the average of the Mach numbers composing the 3D trajectory, the *RTA* was imposed, and the 3D *Flight Time* was known from the previously computed 3D trajectory.

For this equation to be true, $RTA \approx Flight\ Time$, this way the quotient remains close to 1. The best case scenario for Eq. (7.11) would be when *Flight Time* equals *RTA*; when this happens, $AMN = AMN_{3D}$. If $RTA / Flight\ Time > 1$, the resulting *AMN* increments, the contrary is true when $RTA/ Flight\ Time < 1$ the resulting *AMN* decrements. This new *AMN* is a second and better approximation of the speed required to fulfill the RTA, with a precision.

This new *AMN* is the initial Mach number used to find the combination of Mach numbers that satisfies the RTA constraint. This is done using a similar methodology to find the 3D trajectory. The main difference is that the search space consists of the available Mach numbers, and not of the position in space. Each waypoint that composes the 3D trajectory is mapped to the *AMN* selection graph. A reduced Mach selection graph for each waypoint with 0.05 Mach separations up to ± 0.1 Mach is shown in Figure 7.16.

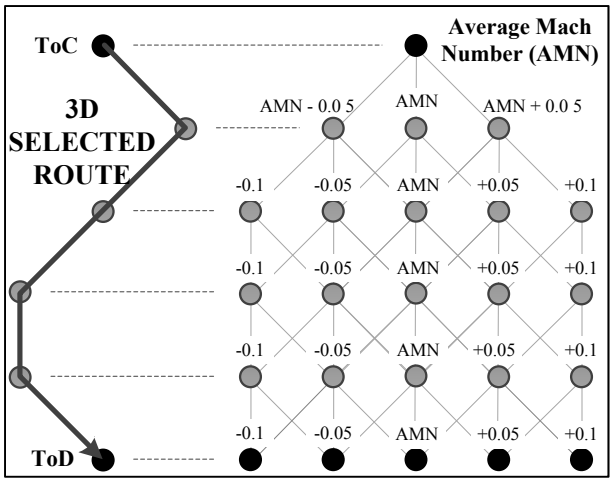

Figure 7.16 Mach number selection

Note that the ToC is mapped to the AMN as the initial Mach number. The next waypoint in the 3D trajectory is mapped to the Mach number available selections. This is repeated for the next 3D waypoints until the ToD. The ToD is mapped to many different Mach numbers, as there is no imposed Mach number at the end of the flight.

The particularities of the Mach number selection process for the three different parts that compose the ACO applied to the RTA are explained below.

7.5.1 RTA (4D) First Module – M1

The algorithm creates a random combination of Mach Numbers from the ToC to the ToD. One speed is selected for each waypoint composing the optimal 3D trajectory (see Figure 7.14) while respecting the grid rules described in Section 7.2.5.2.

Similarly, as in the 3D trajectory optimization, different parameters such as *Fuel Burn*, *Flight Time*, *Time Delay* and *Flight Cost* are computed for the randomly-generated trajectory, using Equations (7.5) to (7.8). A random trajectory with this computed parameter is shown in Figure 7.17.

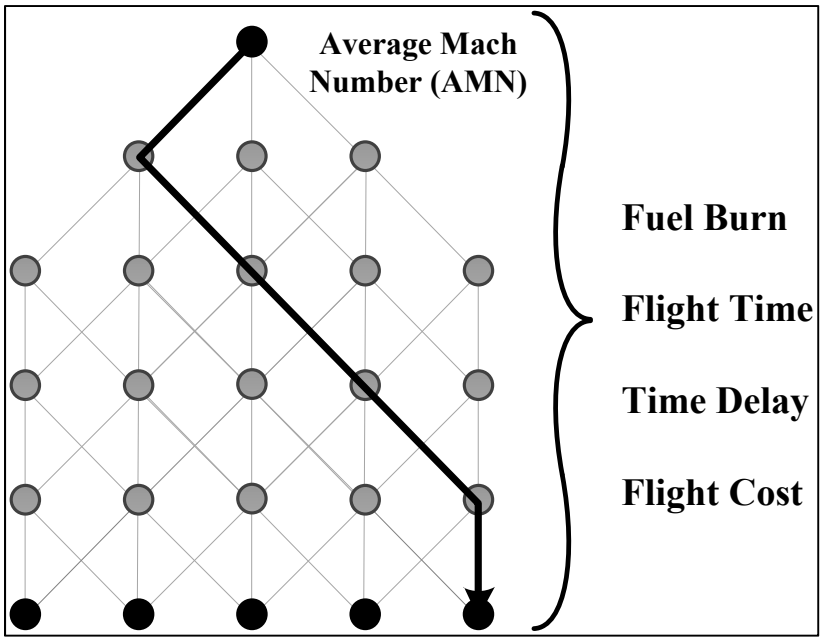

Figure 7.17 Mach number selection with a random trajectory

This module steps can be summarized as follows:

1) A random combination of Mach number is created using the *AMN* as the initial Mach number.

2) Parameters of interest such as *Flight Time*, *Fuel Burn*, *Flight Cost*, and *Time Delay* are computed for the created combinations of Mach numbers.

3) Module 1 for RTA ends

7.5.2 RTA Second Module – M2:

As in the 3D trajectory optimization, firstly, the parameter *Prob* for the available trajectory is computed taking into account the parameters such as the *Flight Cost* and the *Time Delay* using Eq. (7.9). The main difference in computing the coefficient *Prob* in this RTA fulfillment is that $\alpha > \beta$, because in this stage, fulfilling the *RTA* constrains (in terms of *Time Delay*) is more important than the *Flight Cost* value.

The most promising trajectories are then identified and pheromone is added to the identified trajectories using in the *PheromRTA* matrix. Finally, the pheromone rate is updated using Eq. (7.12).

$$PheromRTA = PheromRTA * (1 - \gamma_{pher}) \tag{7.12}$$

This module steps can be summarized as follows:

1) The parameter Prob for each available trajectory is evaluated using Eq. (7.9).
2) The highest Prob trajectories are selected.
3) An important quantity of pheromone is added to the edges composing each trajectory in PheromRTA.
4) The pheromone concentration in is updated using Eq.(7.12).
5) Module 2 ends

7.5.3 RTA Third Module – P3:

Firstly, the algorithm updates the pheromone concentration using Eq. (7.9). Next, using the greedy concept (similarly as in the 3D trajectory selection), a combination of Mach numbers is constructed for different trajectories wherein the highest pheromone concentration is found in consecutive waypoints from the three previously-selected trajectories using the parameter *Prob*.

Figure 7.18 is an example of this, showing how a trajectory is constructed by selecting the highest pheromone concentration.

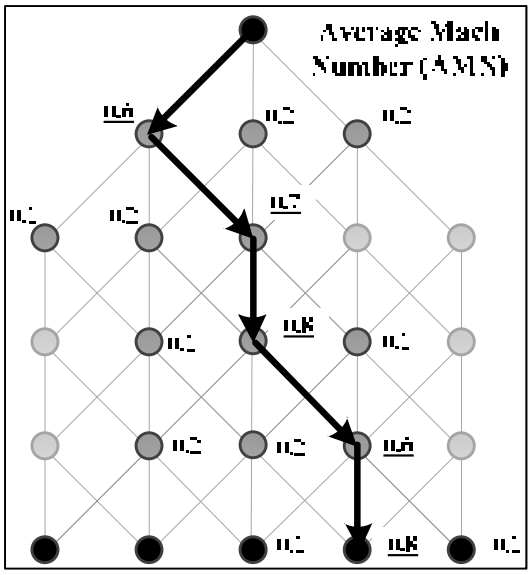

Figure 7.18 Mach number ant selection

Similarly, the required parameters such as *fuel burn*, the *AMN*, *Flight time*, and *time delay* are once again computed. The pheromone concentration is updated from the Module 3 Mach number selection. The current optimal combination of Mach numbers that fulfill the RTA is then determined. Finally, this selection is saved in an RTA data matrix.

This module can be summarized with the next steps:

1) The pheromone concentration is updated using Eq. (7.12).
2) Beginning at the ToD, the algorithms selects the nodes with the highest quantity of pheromone in a greedy way.
3) The composed trajectory is evaluated and it might be selected as the optimal.
4) Module 3 ends

7.5.4 RTA Functioning

This RTA algorithm shown in the 3 above modules is conducted in the same way as in the 3D, this is the flowchart shown in Figure 7.15 is executed once again for the RTA

fulfillment. At the end of this stage, the algorithm delivers the optimal 3D trajectory and the combinations of waypoints that fulfill the RTA constraint (4D).

7.6 Results

The aircraft used to evaluate the algorithm's savings potential was a long haul, 2-engine wide body commercial aircraft from an European manufacturer. For all the test cases, unless otherwise noted, the Gross Weight at the ToC was 130,000 kg.

Two different types of tests were performed to evaluate the algorithm's behavior: The first set of tests was conducted by taking the geodesic line (shortest path in a sphere) between the ToC and the ToD, at a constant altitude of 30 000 ft. The second set of tests was conducted using real flight plans from the website *flightaware*®. The flight plans from this website provide the departure time at the ToC, the ToC geographical coordinates, the ToD geographical coordinates, the waypoints to follow, and the initial altitude.

7.6.1 The ACO algorithm trajectory comparison with the geodesic trajectory

In this section, the flight cost savings for the algorithm in 3D and in 4D are shown by comparing the flight cost of following the geodesic trajectory at a fixed altitude and a fixed Mach number against the flight cost of the trajectories delivered by the algorithm. The fuel burn for the optimal 3D ACO trajectory (blue) and for the optimal 4D ACO trajectory (red) are compared in Figure 7.19. Figure 7.20 shows a comparison of the 3D trajectory and the 4D trajectory for the same flights in terms of the Flight Costs. Those graphics provide information for a cruise flight at the cost index 50.

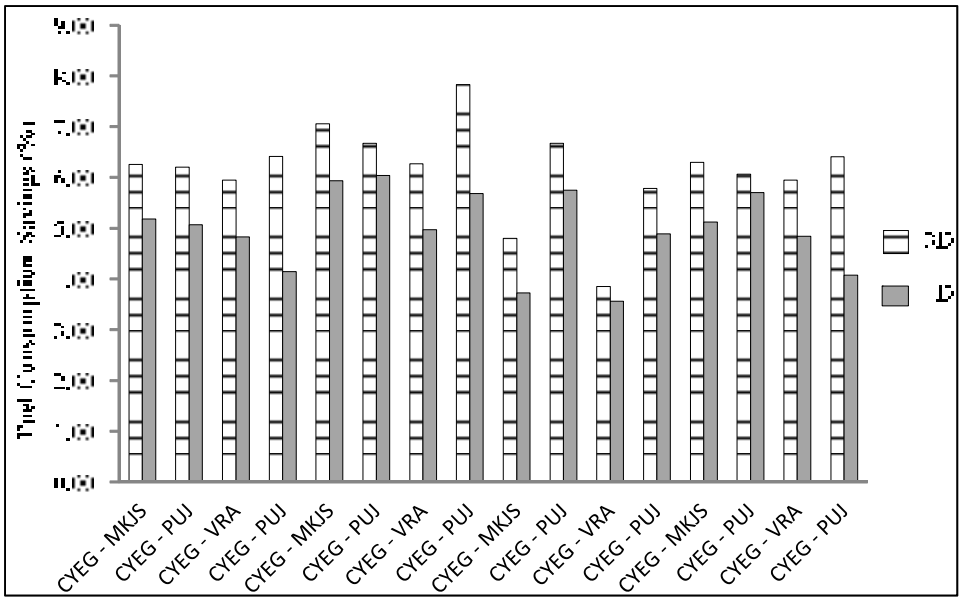

Figure 7.19 Fuel burn saving percentage for different flights

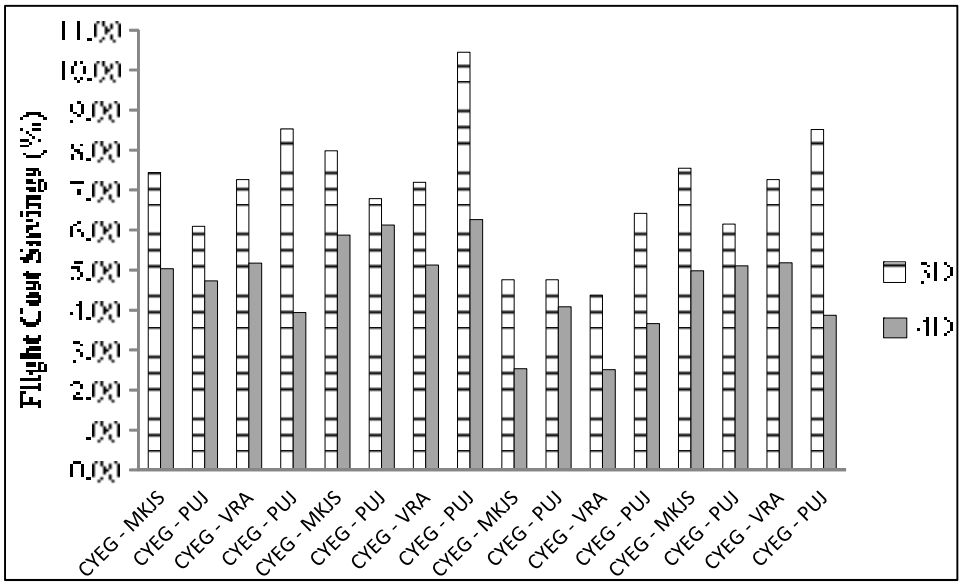

Figure 7.20 Flight cost saving percentage for several flights

As expected, by analyzing the results provided in Figure 7.19 and in Figure 7.20, it can be seen that the ACO algorithm gives very good results when comparing the fuel burn and the flight cost of the trajectories provided by the algorithm against the fixed altitude cruise flight costs. This serves to validate that the step climbs and the aircraft heading changes to where

wind is more favorable proposed by the algorithm would reduce the fuel burn and the flight cost compared to those obtained by fixed Mach number, altitude, and direction trajectories.

By comparing the 3D and the 4D trajectories to the geodesic trajectory results in Figure 7.19, the average fuel burn saving is around 5.6% for the 3D and 4.8% for the 4D trajectory, while the savings average in flight cost is 6.82 % for 3D and 4.62 % for the 4D trajectory. It is clear that the 3D trajectories provide better results than the 4D trajectories. This is evident because the 3D trajectories option has more freedom than the 4D option in selecting its Mach number, as there is no RTA constraint. The selected Mach number helps to further reduce the fuel burn. However, the results obtained for the 4D trajectories are still considered to be good, as the average fuel burn difference between the 3D and the 4D trajectories is less than 1%. This fact is considered to be a very good compromise to fulfill the RTA constraint.

To graphically show the solution for the flight tests in 4D, Figure 7.21 shows the lateral reference trajectory for a flight from Edmonton to Punta Cana. The blue line represents the geodesic trajectory at 30,000 ft, while the green line represents the lateral reference trajectory proposed by the ACO algorithm.

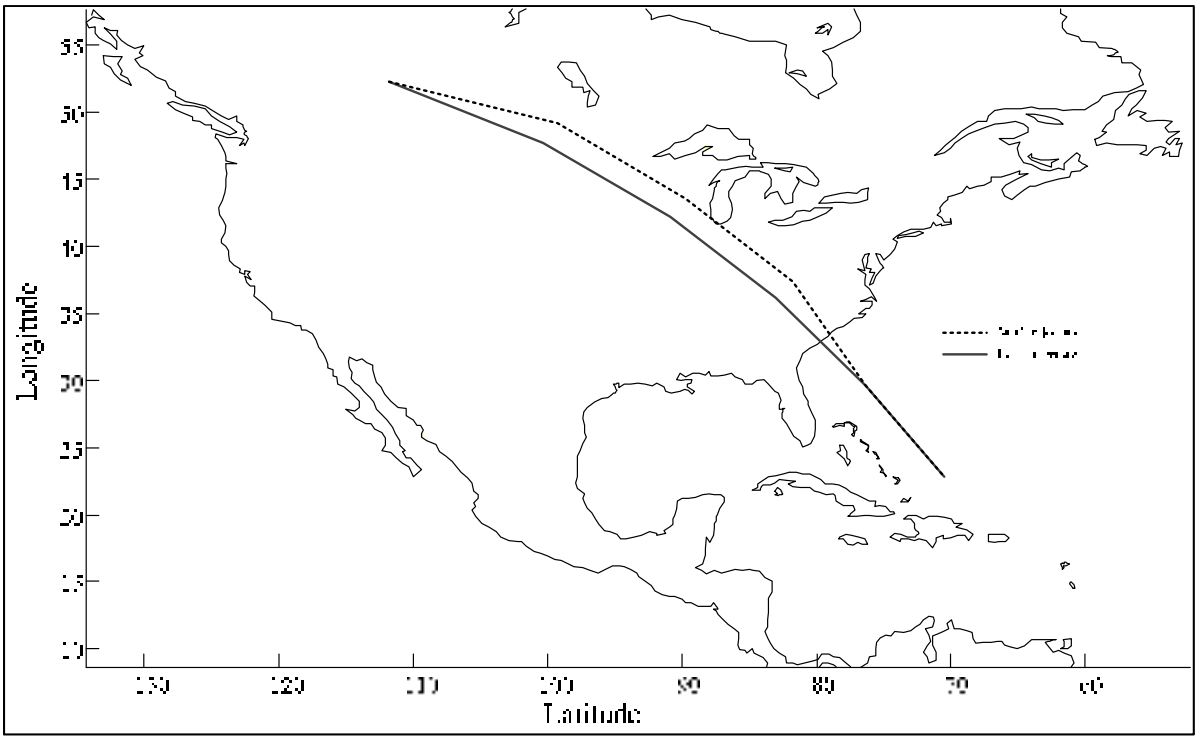

Figure 7.21 ACO trajectory and geodesic lateral reference trajectory.

The executed step climbs are shown on Figure 7.22. It can be seen that the aircraft begins at an altitude of 30,000 ft and executes 4 step climbs until it reaches an altitude of 38,000 ft and stays there until it reaches the ToD.

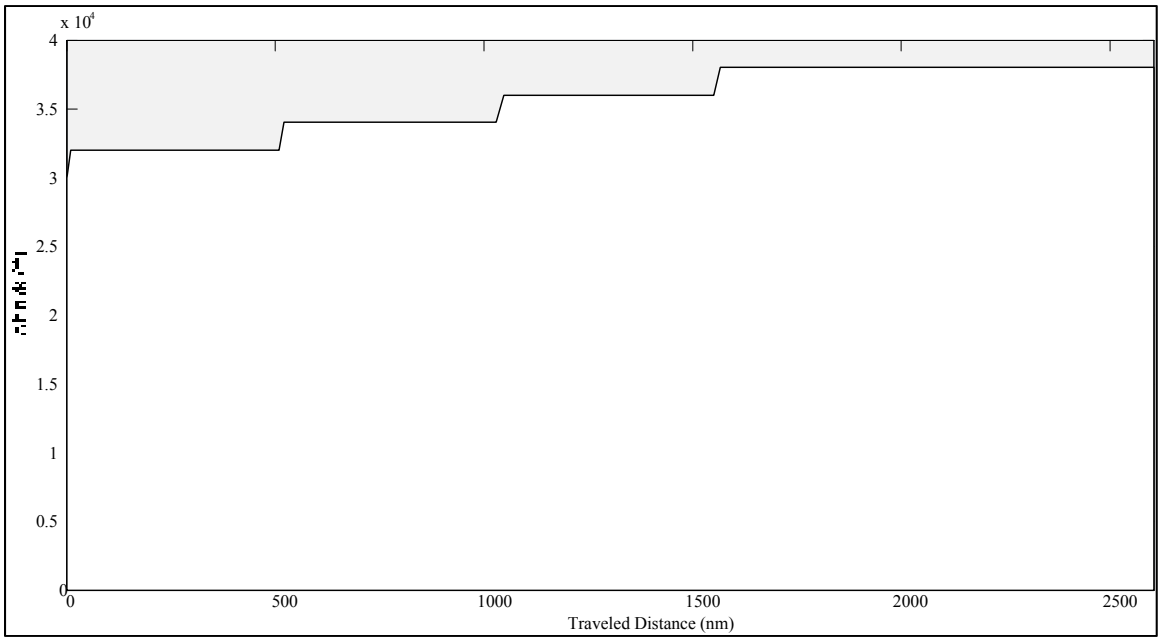

Figure 7.22 ACO trajectory step climb

The Mach number variations are provided in Figure 7.23. It can be observed that the aircraft increments its Mach number, and it keeps it constant every time a step climb is executed.

This observation is logical, as the real speed for a Mach number is higher at lower altitudes. Under the ISA, while below the tropopause, (alt ~< 36000 ft), for a given Mach number, the aircraft is flying faster at lower altitudes, and after the tropopause, the aircraft flies at the same speed. Therefore, as the aircraft gains altitude it is actually flying "slower". For this reason, the speed should be increased so that the RTA constraint can be respected. The last speed increase, occurring above the tropopause, can be explained as a change in weather conditions. It is interesting to observe that the Mach number variation was fairly low, and that after this change, the Mach number was kept constant.

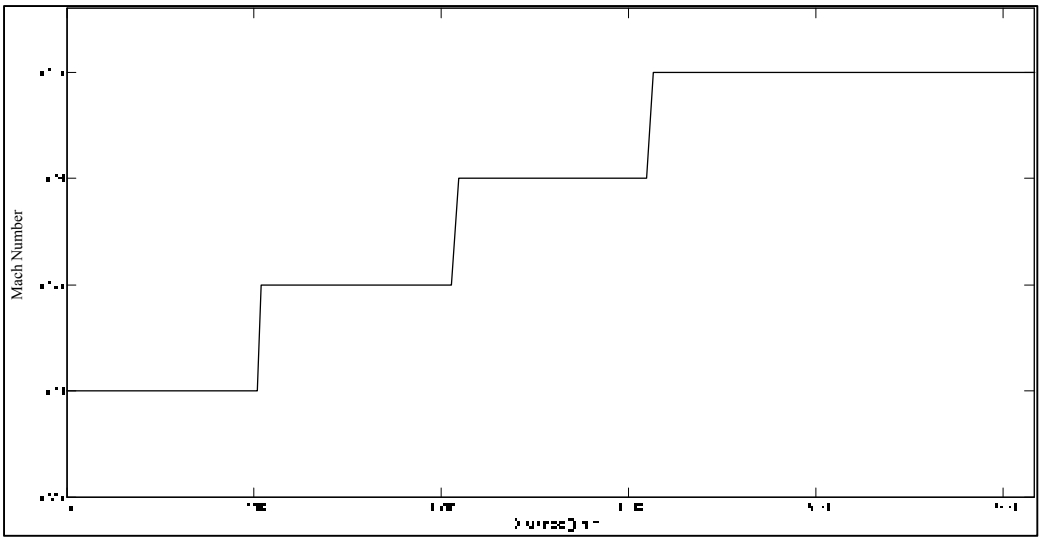

Figure 7.23 Mach number selection

It is worthwhile analyzing the results when changing the *Cost Index*. Thus, additional tests were conducted for two different cost index values (50 & 100). These tests were only conducted for the 3D trajectory. They were not performed for the 4D trajectory as the flight time does not change because the RTA constraints should be respected. Table 7.1 shows the average fuel burn and flight cost percentages of sixteen flight tests at three different cost index values.

Table 7.1 Average fuel burn and flight cost optimization for three different cost index values

Cost Index	**ACO 3D trajectory**		
	0	50	**100**
Fuel burn (%)	6.12	6.15	4.57
Flight Cost (%)	6.12	6.97	7.37

As expected, increasing the Cost Index diminishes the *fuel burn* optimization, as *flight time* becomes more important. This occurs because the algorithm focuses on reducing the *flight time,* as it is the factor that gives the highest cost. For this same reason, the *flight cost*

reduction tends to increase as the aircraft begins to fly at higher speeds by shorting the *flight time,* and thus by reducing the *flight cost*.

7.6.2 The ACO algorithm versus a real as flown flight plan results

The ACO algorithm can reduce the fuel cost compared to the costs for *as flown* flights. The *as flown* flight information was taken from the website *flightaware* ®.

From Figure 7.24 to Figure 7.26, the real flight is shown in magenta and the flight optimized by the ACO algorithm is shown in green. For the real flight optimization process, the selected RTA constraint that should be fulfilled by the ACO algorithm was the real flight's arrival time.

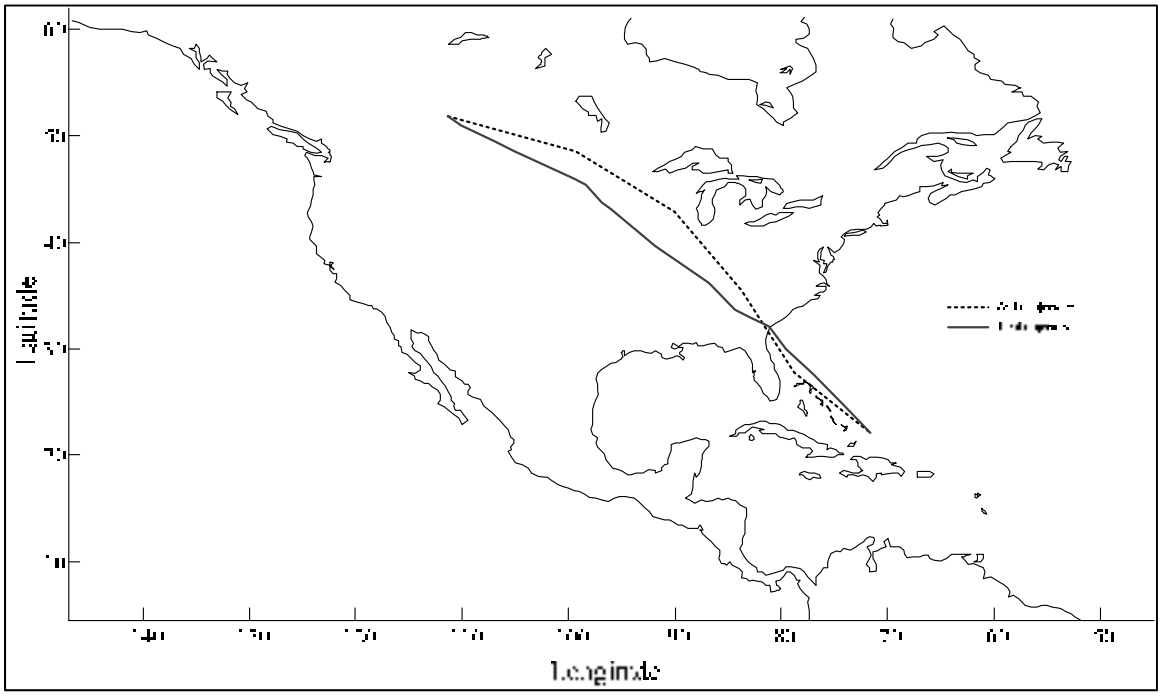

Figure 7.24 Lateral reference trajectory for the ACO algorithm and for a real flight plan

Figure 7.25 shows the vertical profile, and indicates that the ACO algorithm proposed a two-step climb while the *as flown* flight executed three step climbs. It is interesting to observe

that the *as flown* flight executed a 1,000 ft step climb. It can also be seen that the algorithm proposed the step first step climb later than the *as flown* flight; this can be attributed to the weather changes caused by the lateral reference trajectory change.

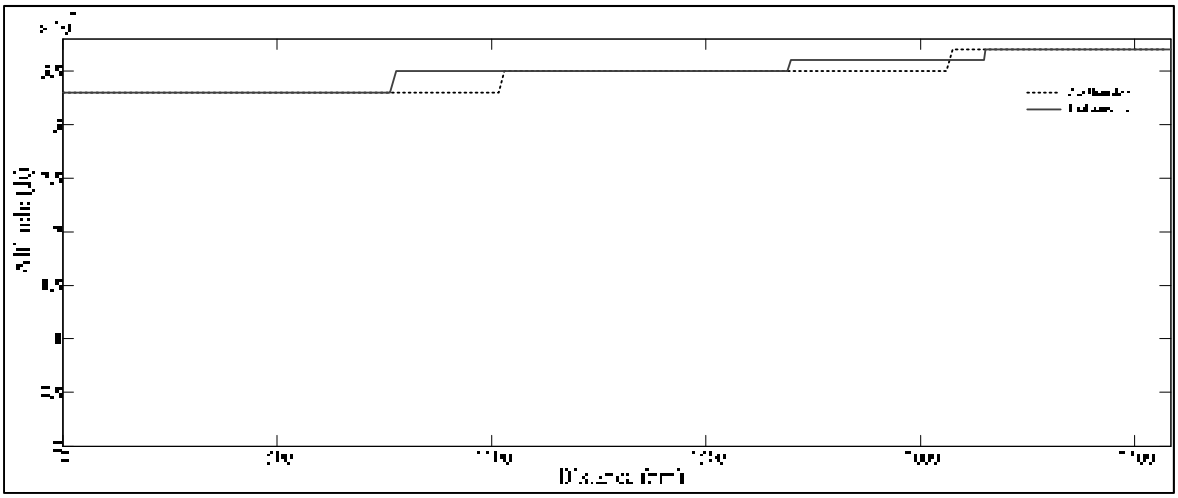

Figure 7.25 ACO vertical trajectory step climb and the as flown vertical trajectory

Figure 7.26 shows the Mach number variations throughout the flight. Again, the Mach numbers for the *as flown* flight is shown in magenta, and the Mach numbers provided by the ACO algorithm is shown in green. The Mach number variations provided by the algorithm are stable, as the Mach number of 0.78 is maintained for almost the whole flight. After the second change of altitude, the aircraft speeds up a little to 0.785.

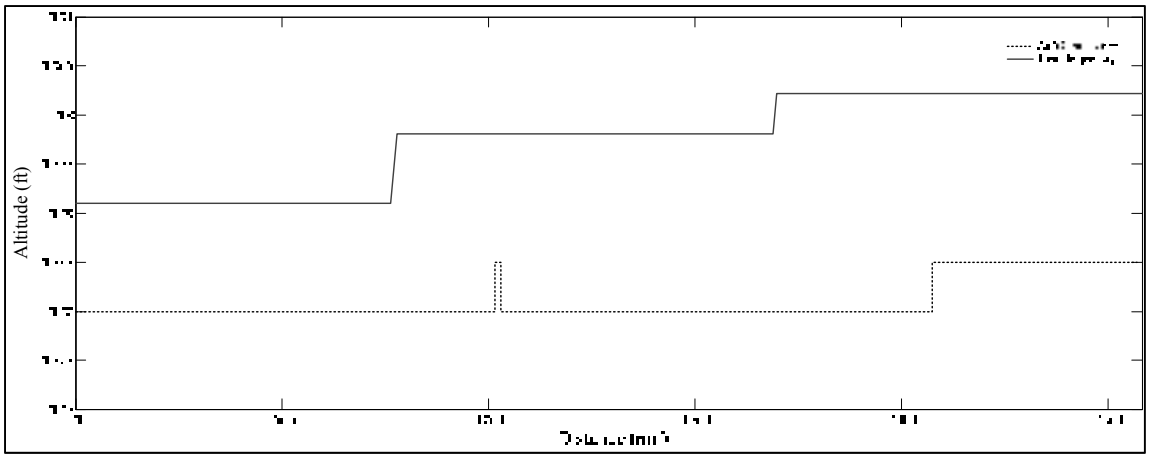

Figure 7.26 Mach number selection

7.6.3 The ACO algorithm versus different as flown flights

It is worthwhile to observe the ACO's optimization capacities for many different *as flown* flights. 16 different flights (some are the same city pairs at different dates, thus different weather conditions) flown by the same aircraft were captured from *flightaware*®. The optimization was performed for 3D and 4D trajectories. For these flights, the was CI = 50 kg/hr.

Fuel burn and flight cost are shown on Figure 7.27 and Figure 7.28, respectively, for the 3D and 4D ACO trajectories are compared to the flight plan trajectories.

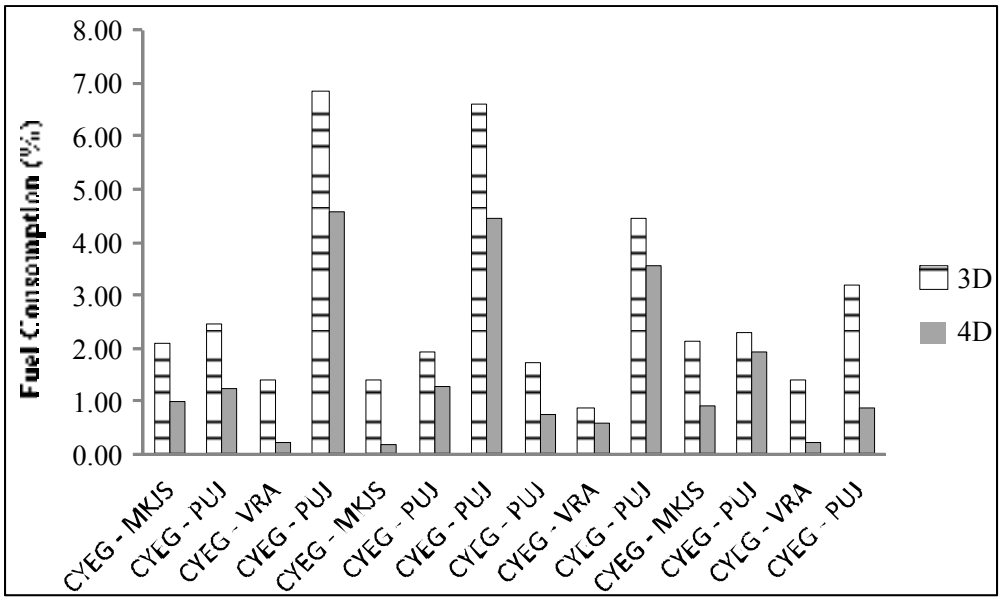

Figure 7.27 Fuel burn savings percentages for different flights

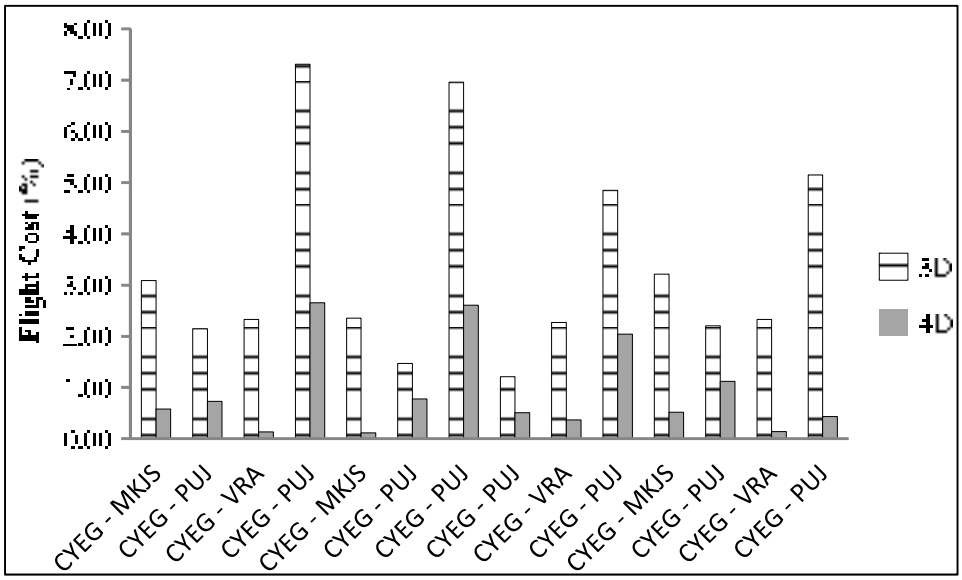

Figure 7.28 Flight cost savings percentage for several flights

The savings obtained from these different flights have close values to those of the preceding savings; thus the 3D trajectories provide more savings than the 4D trajectories. In the 3D trajectories, for the flight CYCG – PUJ (flight 3) the maximum where fuel savings of almost 7% (and maximum flight cost savings of over 7.3%) was obtained. For the same flight in 4D,

fuel savings were close to 4.6% (and the flight cost savings were of 2.65%). The average savings in terms of flight cost for the 3D trajectories was 3.35%, while the average flight cost optimization for the 4D trajectories was 0.91%.

However, there were cases where the 4D optimization savings was not as high (less than 1% for both fuel burn and flight cost) as for the *as flown* flight, such as in the case of flights CYEG – VRA, CYEG – MKJS, and for the second CYEG-VRA, where the 4D optimization was less than 1%.

A quick observation is that fulfilling the RTA constraints brings as a consequence low optimization percentages. This is due partially because flying at the Mach numbers that fulfill the RTA makes the aircraft to sometime fly at speeds where the engines' performance is less efficient in terms of fuel burn.

Flying at different speeds bring as a consequence missing what could be influential weather conditions. For example, it is possible that good tailwinds were missed because the aircraft was flying at speeds that do not allow it to catch those weather conditions.

For these same flights shown in Figure 7.27 for 3D trajectories, it is interesting to observe their optimization results obtained by modifying the Cost Index. These results are presented in Table 7.2.

Table 7.2 Average savings percentages of fuel burn and flight cost for three different cost indexes.

Cost Index	**ACO 3D Algorithm**		
	0	**50**	**100**
Fuel burn (%)	2.74	2.55	1.13
Flight Cost (%)	2.74	3.19	3.63

By comparing the optimizations results shown n Table 7.1 and Table 7.2, it is clear that the obtained optimization is considerably less in Table 7.2. This fact is because the comparison was evaluated against real pre-optimized flights in Table 7.2, while in Table 7.1 the optimization was evaluated against real non-optimized trajectories.

The results shown in Table 7.2 have the same type of behavior as the results shown in Table 7.1. As the Cost Index was increased, the fuel burn optimization was reduced, while the flight cost optimization was improved. This fact occurs because as the Cost Index increases, the algorithm's focus is to reduce the flight time, since it becomes a more important factor than the fuel burn.

7.6.4 The Required Time of Arrival

Flight information was obtained from *flightaware*®. The flight times for the optimized flights were recorded for the 3D and the 4D trajectories. For the 4D trajectory, the RTA was selected as the flight time recorded by the flights obtained from *flightaware*® at the ToD. In other words, if *flightaware*® recorded 17h00 at the ToD, the RTA constraint for the 4D trajectory was for the aircraft to arrive at the ToD at 17h00 +/- tolerance. For these tests, this tolerance was arbitrarily selected to be 40 seconds.

In Table 7.3, the time differences from the RTA for many different flights on different days were evaluated in 3D and 4D for three different Cost Index values.

Table 7.3 Absolute flight time differences from the RTA for the optimized trajectories in 3D and 4D for different Cost Index values

Flight	Time Objective hh:mm:ss	CI = 0		C=50		CI=100	
		3D hh:mm:ss (s)	4D hh:mm:ss (s)	3D hh:mm:ss (s)	4D hh:mm:ss (s)	3D hh:mm:ss (s)	4D hh:mm:ss (s)
CYEG - MKJS	5:27:17	5:19:51 (445)	5:27:24 (8)	5:12:33 (884)	5:27:17 (0)	5:08:30 (1126)	5:27:05 (11)
CYEG - PUJ	5:52:20	5:57:56 (336)	5:52:27 (7)	5:46:18 (362)	5:52:28 (8)	5:40:55 (685)	5:52:15 (5)
CYEG - VRA	4:53:27	4:48:39 (288)	4:53:28 (1)	4:42:41 (646)	4:53:27 (0)	4:38:29 (898)	4:53:28 (1)
CYEG - PUJ	6:19:23	5:57:19 (1324)	6:19:15 (8)	5:49:04 (1819)	6:19:55 (32)	5:44:46 (2077)	6:19:27 (4)
CYEG - MKJS	5:27:56	5:22:38 (318)	5:27:55 (1)	5:15:44 (731)	5:27:57 (1)	5:11:50 (965)	5:27:46 (9)
CYEG - PUJ	5:56:50	6:02:30 (341)	5:56:43 (6)	5:54:02 (168)	5:56:40 (10)	5:48:17 (513)	5:56:41 (9)
CYEG - PUJ	4:52:00	4:47:32 (268)	4:51:58 (2)	4:42:08 (1732)	4:51:58 (8)	4:37:27 (873)	4:52:19 (19)

Table 7.3 Absolute flight time differences from the RTA for the optimized trajectories in 3D and 4D for different Cost Index values (continue)

CYEG - VRA	5:16:29	5:11:24 (305)	5:16:42 (12)	5:04:01 (727)	5:16:30 (7)	5:00:35 (954)	5:16:39 (10)
CYEG - PUJ	5:42:04	5:46:27 (263)	5:42:23 (19)	5:40:26 (1187)	5:41:30 (25)	5:35:04 (420)	5:42:00 (4)
CYEG - MKJS	4:44:42	4:42:25 (137)	4:44:36 (6)	4:32:35 (931)	4:44:35 (10)	4:29:23 (919))	4:44:36 (6)
CYEG - PUJ	6:04:56	5:53:00 (716)	6:04:53 (3)	5:45:09 (439)	6:05:21 (10)	5:40:36 (1460)	6:04:52 (4)
CYEG - VRA	5:27:17	5:21:05 (371)	5:27:22 (5)	5:11:45 (646)	5:27:27 (1)	5:08:20 (1137)	5:27:11 (6)
CYEG - PUJ	5:52:20	5:53:52 (92)	5:52:09 (11)	5:45:01 (1816)	5:52:30 (10)	5:40:36 (704)	5:52:30 (10)

Table 7.3 shows that the absolute time differences between the flight time for the trajectories in 3D range from 1.5 minutes (or 92 seconds) up to 35 minutes (or 2,077 seconds), as in the flight CYEG – PUJ, *CI* = 100. This result is normal, because in 3D the algorithm does not put much effort on reducing the flight time. In the 3D context, the only interest of the algorithm is to reduce the *Flight Time* to reduce the *Flight Cost* as the Cost Index increases. This type of result can be observed by analyzing the flight time for each flight on Table 7.3.

On the other hand, the 4D algorithm always stayed close to the flight time reduction objective. In almost all cases, a small increase of speed, of the order of seconds can be

observed as the Cost Index increases, while the time difference (number on parenthesis) always stays within the imposed limits.

Figure 7.29 to Figure 7.31 show a visualization by bars of these comparisons of flight times obtained for the different CIs in Table 7.3.

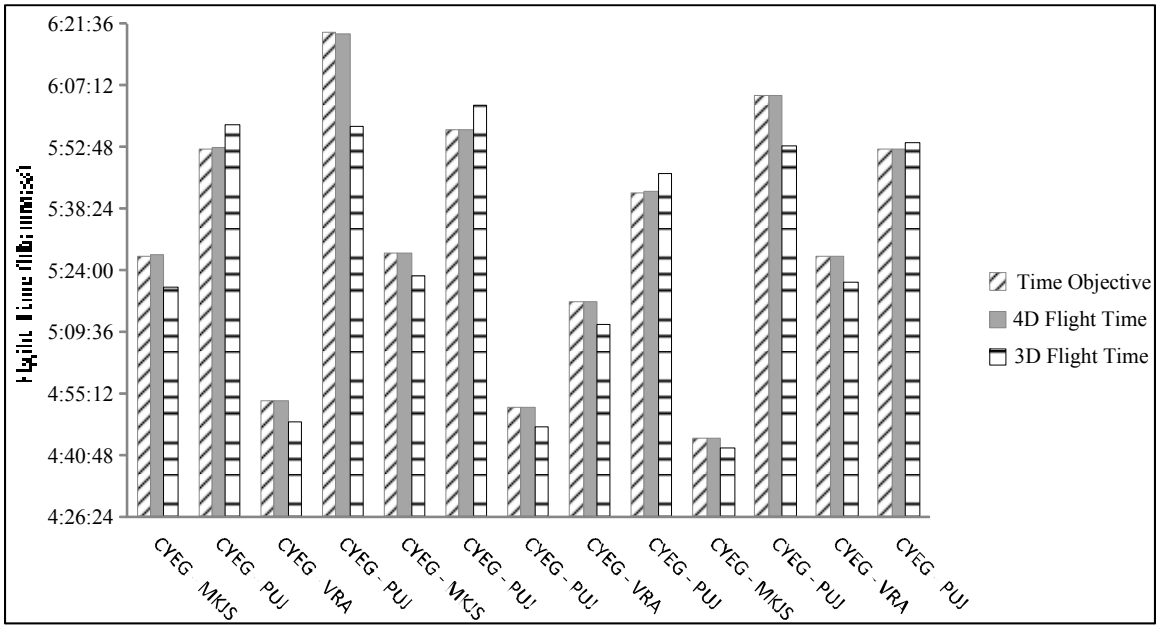

Figure 7.29 Flight time comparison between the 3D flight time and the 4D flight time for a CI = 0

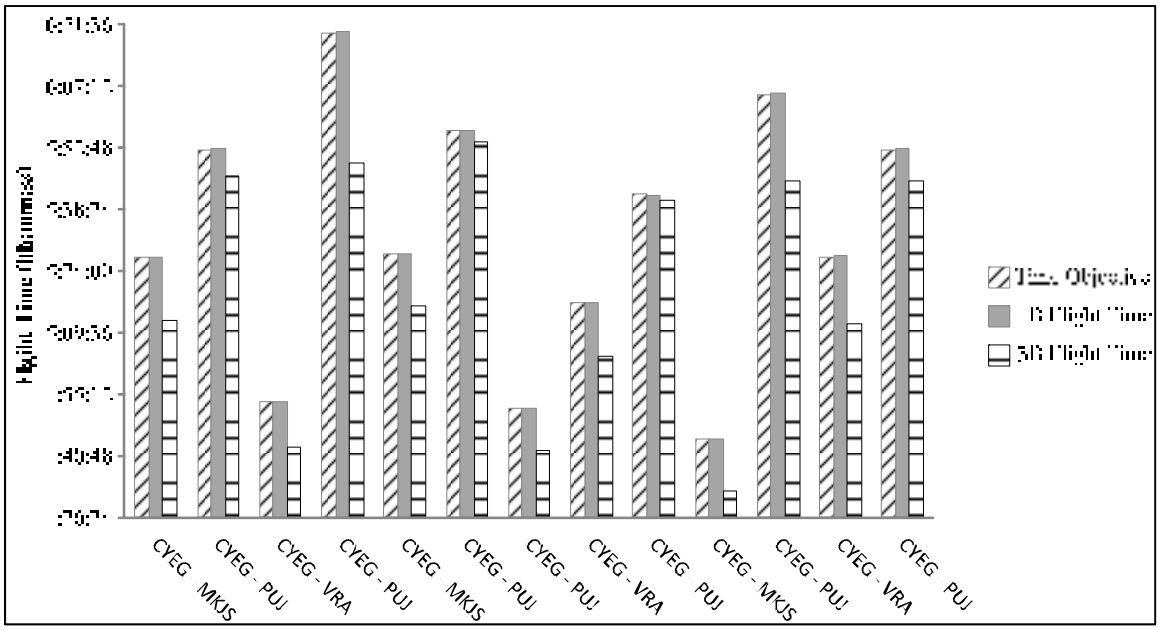

Figure 7.30 Flight time comparison between the 3D flight time and the 4D flight time for a CI = 50

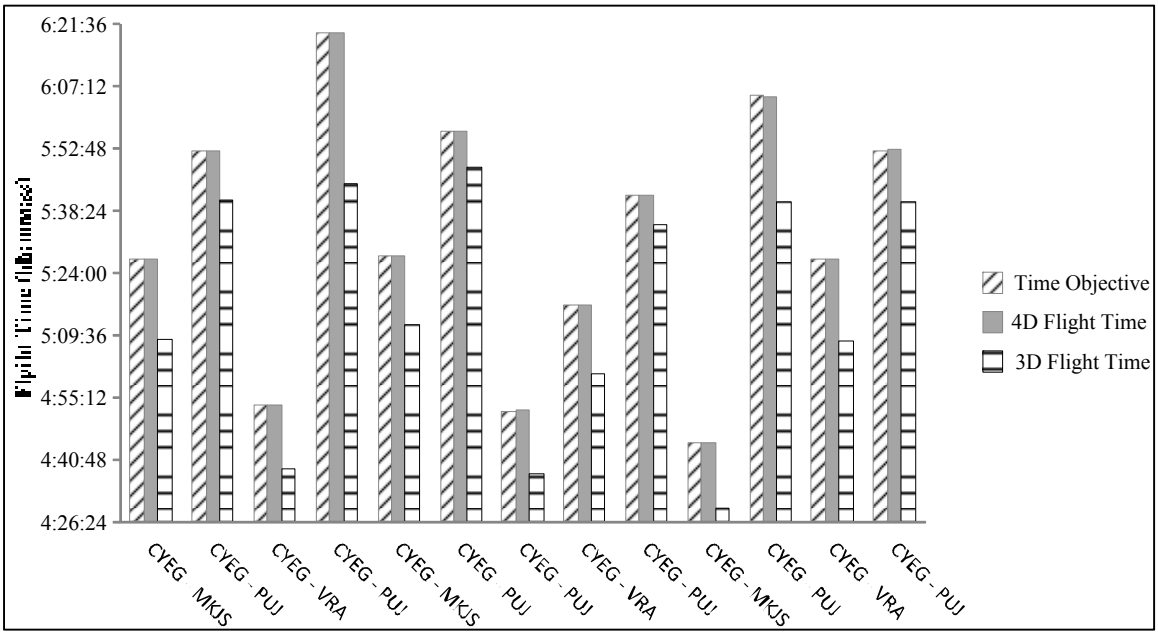

Figure 7.31 Flight time comparison between the 3D flight time and the 4D flight time for a CI = 100

It can be observed that the computed flight time for the 3D trajectories in Figure 7.29 is always higher than the flight time objective (RTA) result; this is obtained because the algorithm selects Mach numbers that reduce the fuel burn, which occurs normally at low speeds. Figure 7.30 shows that for a CI = 50, the flight time obtained for the 3D trajectories diminishes as flight time becomes a part of the flight cost, and thus higher speeds are selected. Finally, Figure 7.31 shows that for a CI = 100 the flight time for the 3D trajectories is at its lowest value as flight time becomes a more important part of the flight cost than in the cases for lower CIs. On the other hand, for the 4D trajectories, the flight time always stays close to the time objective, thus fulfilling the RTA constraint.

7.7 Conclusion

This paper presents an algorithm that finds an economical trajectory in 3D taking into account current airspace constraints and in 4D for the next generation airspace systems. The Ant Colony Optimization (ACO) methodology was used as inspiration to build this algorithm, which is capable of providing the 3D trajectory to be used in the current airspace. The algorithm only accounts for the cruise phase of a flight; it is capable of performing step climbs, and of changing an aircraft's heading in the direction of more favorable weather such as winds and temperature.

A three-dimensional trajectory was first found from the Top of Climb to the Top of Descent in the cruise regime, and then the combination of Mach numbers that fulfilled the Required Time of Arrival constraint was implemented for that trajectory.

Only one set of weather obtained from Weather Canada was used, thus the proposed trajectory could provide the computed saving benefits only if flown under that weather prediction.

In all cases the algorithm was able to provide a more economical trajectory than the flew one. However, this algorithm cannot guarantee that the optimal solution was found.

Following the analysis of different results, it was possible to conclude that the ACO algorithm was able to find an efficient trajectory while fulfilling the required time of arrival constraints. It was also shown that the ACO algorithm was able to find similar solutions than to those computed by airlines, which led to the conclusion that the algorithm proposed coherent trajectories.

The results provided an overview of the 4D influence on the flight cost. In geodesic fixed altitude trajectories, the average 3D optimization in terms of fuel burn was 5.8%; by adapting these trajectories to a 4D objective, the optimization gave lower results such as to 4.8%, for a loss of 1%. For more complex trajectories, such as flow flights, the average optimization obtained for 3D trajectories was 3.35%; however, by adapting this trajectory to a RTA objective (4D), only a 0.91% optimization was achieved. This result means that adapting "optimal" 3D trajectories to the 4D constraint would diminish 2.44% from the optimization opportunities. Another case was reported in which the optimization went down from around 2.33% in 3D to close to 0.14% in 4D. In other words, the 4D optimization is necessary, but it might be achieved with a significant reduction in the optimization capabilities.

The cost index value selection did not affect the 4D trajectory result, as the RTA objectives were always fulfilled. However a small increase of speed was observed as the Cost Index increased. Trajectories in 3D showed flight time decreasing as the Cost Index increased.

In future work, it would be desirable to find a Cost Index that could fulfill the RTA constraint for a 3D trajectory. The algorithm could then consist of only one stage, as the second stage (the one used to fulfill the RTA) could be discarded as the 3D trajectory would provide the 4D trajectory along with the desired Cost Index value.

This research evaluated the flight cost reduction using as flown flights. It is desirable to obtain different the methodology used by airlines in order to compare against the developed algorithm. It is also desirable to compare the developed algorithm against different algorithms developed by other research groups.

CHAPTER 8

DISCUSSION OF RESULTS

Each chapter presents a detailed results section with their depth analysis and discussion. This section presents a summary of these results.

In Chapter 3, a new methodology to compute the flight cost was developed. The flight cost for three different aircrafts was evaluated. Aircraft A was evaluated using FlightSIM®. For this aircraft, the maximum flight burn difference was of **2.52%** and the maximal flight time difference was of **1.45%.** For Aircraft B and Aircraft C, the fuel burn and the flight time were compared with those obtained with the PTT provided by our industrial partner. For aircraft B, the maximal fuel burn difference was of **0.94%** and the maximal flight time difference was of **1.74%.** For Aircraft C, the maximal fuel burn difference was of **2.27%** and the maximal flight time relative error was of **1.37 %**.

In Chapter 4, the vertical reference trajectory was optimized by implementing the Beam Search algorithm. The algorithm was able to find the optimal trajectory while comparing it with the optimal trajectory provided by an exhaustive algorithm. The algorithm was able to find the optimal solution by computing only **33%** of the search space in the best case scenario for the Aircraft A, **28%** of the search space for the Aircraft B, and **10%** of the search space for the Aircraft C. The algorithm was able to find extremely good sub-optimal solutions when weather was taken into account. The maximal difference between the optimal solution and the found solution was of only **0.34%**

In Chapter 5, the Beam Search was improved with a pre-search space selection in order to improve the computation time. Results showed that the computation time was reduced to half of the time in average. The algorithm was normally able to find the optimal solution or extremely good solutions with a maximal difference of only **0.09%** from the optimal solution

with respect to an exhaustive search. This flight was compared against the PTT flight and it was found that the developed algorithm was able to give savings of up to **6000 kg**.

Chapter 6 presented the implementation of the 4D reference trajectory optimization using the Artificial Bee's Colony (ABC) optimization algorithm. This algorithm was able to reduce the flight cost to up to **7.44 %** for a long haul flight with a RTA difference of **5 seconds**. The lowest optimization found was of **0.73 %** of fuel burn with a RTA difference of **2 seconds**. The average fuel savings evaluating many different flights was of **3.9 %** for the evaluated flights.

Chapter 7 presented the implementation of the Ants Colony Optimization (ACO) algorithm. Fuel savings of over **6.82 %** were found when comparing the algorithm trajectories with geodesic constant altitude trajectories, and fuel savings of up to **4.5%** were found while comparing the algorithm trajectories with *as flown* flights. This study also showed that flying 4D trajectories with an imposed RTA are in average **2.4%** more expensive than flying 3D trajectories.

CONCLUSION AND RECOMMENDATIONS

Different versions of flight reference trajectory optimization algorithms were presented in this manuscript-based thesis.

It can be concluded that for a numerical performance model, a series of linear interpolations provides fast and accurate results. As this methodology is of a deterministic nature, it can be implemented in airborne devices such as the FMS. This is also true for the heuristic developed in this thesis.

While computing the fuel burn during cruise, it was observed that reducing the fuel burn from the aircraft's total weight every 25 nm to 35 nm segments is a good compromise between computation time and accuracy of results. Updating the aircraft's gross weight for distances less than 25 nm provides more accurate results, but the computation time is significantly higher making algorithms too slow. Updating the aircraft's total weight in distances longer than 35 nm would provide a faster computation time, but it will degrade the accuracy of results at the point of providing *false optimal* trajectories.

The vertical reference trajectory can also be modeled as a decision graph and search algorithms such as Beam Search can be implemented to find the optimal trajectory. For this type of algorithms, it is of interest to find methods to reduce the search space as the optimal solution will be found faster by the algorithm find an optimal solution faster. It was concluded that reliable heuristics able to estimate the flight cost can be design using numerical models such as the one used in this thesis.

It was concluded that finding a good 3D trajectory, then optimizing its speeds to achieve an RTA (4D) is a better strategy than directly computing a 4D trajectory. This is found because trying to find directly the 4D trajectory would require more iterations. If an unrealistic RTA objective is set, the algorithm is not able to provide any solution at all.

It was confirmed that weather plays a major role in aircraft trajectory optimization. Changing any parameter (such as speed, altitude or geographical position) would change the future aircraft behaviour in terms of geographical position or time. This means that weather can dramatically change between similar trajectories.

Linked to the conclusion above, it was also observed that the required time of arrival constraint dramatically decreases the optimization potential. The required time of arrival could lead to the decision of avoiding tailwinds as they might complicate fulfilling the time constraint. These requirements evidently bring as a consequence more expensive flights.

Among the developed algorithms, it was observed that the modified Beam Search algorithm (third paper) was better as it provided almost as good results in only a fraction of the time. It can be concluded that metaheuristic algorithms were able to provide good solutions.

Future work on this research could take the direction explained in the next paragraphs.

All flights were optimized using the *free flight* concept and it was assumed that the computed trajectory was the one flown. However, once airborne, there are many aircraft in the sky that might request the same, or parts of the same trajectory computed by the developed algorithms. It could be of interest to develop an algorithm able to take into account traffic, and negotiation with conflicting aircraft, so both aircraft can reach an agreement and each one can fly its most optimal trajectory.

Weather was taken into account for this research, however, storms and dangerous conditions were not taken into account. There might be zones where interesting whether parameters can be found such as good tailwinds; however, these flight zones might be zones including storms or even No-Fly zones (such as military complex, government building, landmarks, etcetera). Constraints such as No-Fly zones should be introduced to achieve most realistic trajectories.

ATC might negate the computed trajectory; then, the algorithms able to compute different trajectories, used as ‘back-up’ trajectories are as well desirable. Different RTA constraints during flight could be of interest. This research took into consideration only a RTA at the end of cruise. The airspace contains different control zones; RTAs might be required every time an aircraft enters a new control zone or if heavy traffic is present.

This research explored 4 different optimization algorithms. However, there are many different algorithms in the literature such as optimal control, cuckoo search, bat algorithm, etcetera. These algorithms should also be studied to investigate if they provide better results than the algorithms studied in this thesis or the ones available in the literature.

BIBLIOGRAPHY

(ATSDR),. (1999). Toxicological profile for total petroleum hydrocarbons (TPH). Dans *U.S. Department of Health and Human Services, Public Health Service*.

Airbus. (2011). Airbus 321 demonstrates first RNP-AR arrival combined with a transition to an ILS approach. Blagnac Cedex, France.

Analysis, D.O.R.A. (1978). *The Noise Benefits Associated with Use of Continuous Descent Approach and Low Power / Low Drag Approach Procedures at Heathrow Airport* (nº 7807). London: Civil Aviation Authority.

Ashok, A., Dedoussi, I. C., Yim, S. H. L., Balakrishnan, H., & Barrett, S. R. H. (2014). Quantifying the air quality-CO2 tradeoff potential for airports. *Atmospheric Environment, 99*, 546-555. doi: http://dx.doi.org/10.1016/j.atmosenv.2014.10.024.

ATAG. (2005). *The economic & social benefits of air transport*. Geneva, Switzerland: Air Transportation Action Group.

ATAG. (2009). *Beginner's Guide to Aviation Biofuels*. Geneva, Switzerland Air Transportation ACtion Group.

ATAG. (2010). *Beginner's Guide to Aviation Efficiency*. Geneve, Switzerland: Air Transportation Action Group.

ATAG. (2014). *Aviation Benefits Beyond Borders*. Geneva, Switzerland: Air Transport Action Group.

ATAG. (2016). *Aviation Benefits Beyond Borders*. Geneva, Switzerland: Air Transport Action Group.

Balakrishnan, H. (2016). Control and optimization algorithms for air transportation systems. *Annual Reviews in Control, 41*, 39-46. doi: http://dx.doi.org/10.1016/j.arcontrol.2016.04.019.

Black, D. A., Black, J. A., Issarayangyun, T., & Samuels, S. E. (2007). Aircraft Noise Exposure and Resident's Stress and Hypertension: A public Health Perspective for

Airport Environmental Management. *Journal of Air Transport Management, 13*(5), 264-276. doi: http://dx.doi.org/10.1016/j.jairtraman.2007.04.003.

Boeing. (2015). *Precise Approaches at Seattle - Tacoma International airport enhance route and fuel efficency*. BOEING. Repéré à http://splash.alaskasworld.com/Newsroom/ASnews/ASstories/Boeing-Report-SEATAC-Alaska-Airlines-RNAV-RNP-summary.pdf

Boeing. (2016a). Improving efficiency with RNP and GLS.

Boeing. (2016b). *World Air Cargo Forecast 2016-2017*.

Bonami, P., Olivares, A., Soler, M., & Staffetti, E. (2014). Multiphase Mixed-Integer Optimal Control Approach to Aircraft Trajectory Optimization. *Journal of Guidance, Control, and Dynamics, 36*(5), 1267-1277. doi: 10.2514/1.60492. Repéré à http://dx.doi.org/10.2514/1.60492

Bonnefoy, P., & Hansman, R. J. (2010). Operational Implications of Cruise Speed Reductions for Next Generation Fuel Efficient Subsonic Aircraft. Dans *27TH Congress Of The International Council Of The Aeronautical Sciences*.

Bronsvoort, J., McDonald, G., Boucquey, J., Garcia-Avello, J., & Besada, J. A. (2013). *Impact of Data-Link on Ground-Based Trajectory Prediction Accuracy for Continuous Descent Arrivals* présentée à AIAA Modeling and Simulation Technologies (MST) Conference. doi: http://dx.doi.org/10.2514/6.2013-5068

Campbell, S. E., Bragg, M. B., & Neogi, N. A. (2013). Fuel-Optimal Trajectory Generation for Persistent Contrail Mitigation. *Journal of Guidance, Control, and Dynamics, 36*(6), 1741-1750. doi: http://dx.doi.org/10.2514/1.55969

Cao, Y., Kotegawa, T., & Post, J. (2011). Evaluation of Continuous Descent Approach as a Standard Terminal Airspace Operation. Dans *9th USA/Europe Air Traffic Management R&D Seminar*.

Cao, Y., Rathinam, S., & Sun, D. (2013). Greedy-Heuristic-Aided Mixed-Integer Linear Programming Approach for Arrival Scheduling. *Journal of Aerospace Information Systems, 10*(7), 323-336. doi: http://dx.doi.org/10.2514/1.I010030.

Celis, C., Sethi, V., Zammit-Mangion, Singh, R., & Pilidis, P. (2014). Theoretical Optimal Trajectories for Reducing the Environmental Impact of Commercial Aircraft Operations. *Journal of Aerospace Technology And Management, 6*(1), 29-42. doi: http://dx.doi.org/10.5028/jatm.v6i1.288

Chakravarty, A. (1985). Four-dimensional fuel-optimal guidance in the presence of winds. *Journal of Guidance, Control, and Dynamics, 8*(1), 16-22. doi: http://dx.doi.org/10.2514/3.19929.

Chamseddine, A., Zhang, Y., Rabbath, C. A., Join, C., & Theilliol, D. (2012). Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle. *IEEE Transactions on Aerospace and Electronic Systems, 48*(4), 2832-2848. doi: http://dx.doi.org/10.1109/TAES.2012.6324664

Chamseddine, A., Zhang, Y., Rabbath, C. A., & Theilliol, D. (2012). Trajectory Planning and Replanning Strategies Applied to a Quadrotor Unmanned Aerial Vehicle. *Journal of Guidance, Control, and Dynamics, 35*(5), 1667-1671. doi: http://dx.doi.org/10.2514/1.56606.

Chen, Y., Yu, J., Mei, Y., Zhang, S., Ai, X., & Jia, Z. (2016). Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task. *Chinese Journal of Aeronautics, 29*(1), 184-201. doi: http://dx.doi.org/10.1016/j.cja.2015.12.008.

Clarke, J. P., Brooks, J., Nagle, G., Scacchioli, A., White, W., & Liu, S. R. (2013). Optimized Profile Descent Arrivals at Los Angeles International Airport. *Journal of Aircraft, 50*(2), 360-369. doi: http://dx.doi.org/10.2514/1.c031529.

Cobano, J. A., Alejo, D., Heredia, G., & Ollero, A. (2013). 4D Trajectory Planning in ATM With an Anytime Stochastic Approach. Dans ATACCS (Éd.), *Proceedings of the 3rd International Conference on Application and Theory of Automation in Command and Control Systems*. ACM. doi: http://dx.doi.org/10.1145/2494493.2494494

Collinson, R. P. G. (2011). *Introduction to Avionics Systems* (3rd éd.).

Crutzen, P. J. (1970). The influence of nitrogen oxides on the atmospheric ozone content. *Q.J.R. Meteorol. Soc, 96*(408), 320-325. doi: http://dx.doi.org/10.1002/qj.49709640815

Dancila, B., Botez, R., & D, L. (2012, 13-16 August). *Altitude Optimization Algorithm for Cruise, Constant Speed and Level Flight Segments* présentée à AIAA Guidance, Navigation, and Control Conference. doi: http://dx.doi.org/10.2514/6.2012-4772.

Dancila, B., Botez, R., & Labour, D. (2012, 13 August 2012 - 16 August 2012). *Altitude Optimization Algorithm for Cruise, Constant Speed and Level Flight Segments* présentée à AIAA Guidance, Navigation, and Control Conference. doi: doi: http://dx.doi.org/10.2514/6.2012-4772.

Dancila, B., Botez, R. M., & Labour, D. (2013). Fuel Burn Prediction Algorithm for Cruise, Constant Speed and Level Flight Segments. *The Aeronatuical Journal, 117*(1191).

Dancila, R., Botez, R. M., & Ford, S. (2013). *Fuel Burn and Emissions Evaluation for a Missed Approach Procedure Performed by a B737-400* présentée à 2013 Aviation Technology, Integration, and Operations Conference, Los Angeles, USA. doi: doi: http://dx.doi.org/10.2514/6.2013-4387.

Dancila, R., Botez, R. M., & Ford, S. (2014). Fuel Burn and Emissions Evaluation for a Missed Approach Procedure Performed by a B737-400. *The Aeronautical Journal, 118*(1209), 20.

Dicheva, S., & Bestaoui, Y. (2014). Three-Dimensional A* Dynamic Mission Planning for an Airborne Launch Vehicle. *Journal of Aerospace Information Systems, 11*(2), 98-106. doi: http://dx.doi.org/10.2514/1.I010070.

du Puy de Goyne, T., Plays, Y., Lepourry, P., & Besse, J. (1995). *Initiation à l'aéronautique*. Toulouse, France: Cépaduès - Éditions.

Fays, J., & Botez, R. (2013). Aircraft trajectories generation by use of no fly zones self-management for a flight management system. Dans *AIAC15: 15th Australian International Aerospace Congress*. (pp. 60-73). AIAC.

Felix-Patron, R., Botez, R. M., & Labour, D. (2013). *Low Calculation Time Interpolation Method on the Altitude Optimization Algorithm for the FMS CMA-9000 Improvement on the A310 and L-1011 Aircraft* présentée à 2013 Aviation Technology, Integration, and Operations Conference, Los Angeles. USA. doi: http://dx.doi.org/10.2514/6.2013-4256.

Felix-Patron, R., Kessaci, A., & Botez, R. M. (2013, August 19-22, 2013). *Flight Trajectories Optimization Under the Influence of Winds Using Genetic Algorithms* présentée à AIAA Guidance, Navigation, and Control (GNC) Conference, Boston USA. doi: http://dx.doi.org/10.2514/6.2013-4620.

Félix-Patrón, R. S., Berrou, Y., & Botez, R. M. (2014). New Methods of Optimization of the Flight Profiles for Performance Database-Modeled Aircraft. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. doi: http://dx.doi.org/10.1177/0954410014561772

Félix-Patrón, R. S., & Botez, R. M. (2014). Flight Trajectory Optimization Through Genetic Algorithms Coupling Vertical and Lateral Profiles. Dans *International Mechanical Engineering Congress & Exposition*.

Félix-Patrón, R. S., & Botez, R. M. (2015). Flight Trajectory Optimization Through Genetic Algorithms for Lateral and Vertical Integrated Navigation. *Journal of Aerospace Information Systems, 12*(8), 533-544. doi: http://dx.doi.org/10.2514/1.I010348.

Felix-Patron, R. S., Botez, R. M., & Labour, D. (2012, 25-28 Oct. 2012). *Vertical Profile Optimization for the Flight Management System CMA-9000 Using the Golden Section Search Method* présentée à IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society. doi: http://dx.doi.org/10.1109/iecon.2012.6389517

Félix-Patrón, R. S., Kessaci, A., & Botez, R. (2014). Horizontal Flight Trajectories Optimisation for Commercial Aircraft Through a Flight Management System *The Aeronautical Journal, 118*(1210), 20.

Félix Patrón, R., Berrou, Y., & Botez, R. M. (2014). Climb, Cruise and Descent 3D Trajectory Optimization Algorithm for a Flight Management System. Dans *AIAA/3AF Aircraft Noise and Emissions Reduction Symposium*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2014-3018

Felix Patron, R. S., Botez, R. M., & Labour, D. (2013). New Altitude Optimisation Algorithm for the Flight Management System CMA-9000 Improvement on the A310 and L-1011 Aircraft. *The Aeronautical Journal, 117*(1194), 787-805.

Felix Patron, R. S., Oyono Owono, A. C., Botez, R. M., & Labour, D. (2013). Speed and Altitude Optimization on the FMS CMA-9000 for the Sukhoi Superjet 100 Using Genetic Algorithms. Dans *2013 Aviation Technology, Integration, and Operations*

Conference. American Institute of Aeronautics and Astronautics. doi: doi: http://dx.doi.org/10.2514/6.2013-4257.

Field, A. (1985). *International air traffic control : management of the world's airspace / by Arnold Field*. Oxford ; New York: Pergamon Press.

Filippone, A. (2007). On the Benefits of Lower Mach Number Aircraft Cruise. *The Aeronautical Journal, 111*(1122), 531-542.

Franco, A., & Rivas, D. (2011). Minimum-Cost Cruise at Constant Altitude of Commercial Aircraft Including Wind Effects. *Journal of Guidance, Control, and Dynamics, 34*(4), 1253-1260. doi: http://dx.doi.org/10.2514/1.53255.

Franco, A., & Rivas, D. (2015). Optimization of Multiphase Aircraft Trajectories Using Hybrid Optimal Control. *Journal of Guidance, Control, and Dynamics*, 1-16. doi: http://dx.doi.org/10.2514/1.G000688.

Freitag, W., Terry, Shulze E. (2009). Blended Winglets Improve Performance. *Aeromagazine Boeing,* (35), 9-12.

Gabor, O. Ş., Simon, A., Koreanschi, A., & Botez, R. M. (2015). Improving the UAS-S4 Éhecal airfoil high angles-of-attack performance characteristics using a morphing wing approach. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 230*(1), 118-131. doi: http://dx.doi.org/10.1177/0954410015587725

Gagné, J. (2013). *Nouvelle méthode d'optimisation du coût d'un vol par l'utilisation d'un système de gestion de vol et sa validation sur un avion Lockheed L-1011 TriStar* (École de technologie supérieure, Montreal).

Gagné, J., Murrieta-Mendoza, A., Botez, R., & Labour, D. (2013). New Method for Aircraft Fuel Saving Using Flight Management System and Its Validation on the L-1011 aircraft. *2013 Aviation Technology, Integration, and Operations Conference*. doi: http://dx.doi.org/10.2514/6.2013-4290.

Gagné, J., Murrieta, A., Botez, R., & Labour, D. (2013). *New method for aircraft fuel saving using Flight Management System and its validation on the L-1011 aircraft* présentée à 2013 Aviation Technology, Integration, and Operations Conference, Los Angeles USA. doi: http://dx.doi.org/10.2514/6.2013-4290.

Gardi, A., Sabatini, R., Kistan, T., Lim, Y., & Ramasamy, S. (2015). 4 Dimensional trajectory functionalities for air traffic management systems. Dans *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)* (pp. N3-1-N3-11). doi: http://dx.doi.org/10.1109/ICNSURV.2015.7121246

Gardi, A., Sabatini, R., & Ramasamy, S. (2016). Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context. *Progress in Aerospace Sciences, 83*, 1-36. doi: http://dx.doi.org/10.1016/j.paerosci.2015.11.006.

Gardi, A., Sabatini, R., Ramsamy, S., Marino, M., & Kistan, T. (2015). Automated ATM System Enabling 4DT-Based Operations. *SAE 2015 AeroTech Congress & Exhibition, 2015-01-2539*, 7. doi: http://dx.doi.org/10.4271/2015-01-2539

Godbole, P., J.,, Ranade, A. G., & Pant, R. S. (2014). Branch & Bound Global-Search Algorithm for Aircraft Ground Movement Optimization. Dans *14th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2014-2155.

Godbole, P. J., Ranade, A. G., & Pant, R. S. (2014). Branch & Bound Global-Search Algorithm for Aircraft Ground Movement Optimization. Dans *14th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2014-2155

Green, J. E. (2009). The potential for reducing the impact of aviation on climate. *Technology Analysis & Strategic Management, 21*(1), 39-59. doi: http://dx.doi.org/10.1080/09537320802557269.

Guo, X., & Zhu, M. (2013). Direct trajectory optimization based on a mapped Chebyshev pseudospectral method. *Chinese Journal of Aeronautics, 26*(2), 401-412. doi: http://dx.doi.org/10.1016/j.cja.2013.02.018.

Hagelauer, P., & Mora-Camino, F. (1998). A Soft Dynamic Programming Approach for On-Line Aircraft 4D-Trajectory Optimization. *European Journal of Operational Research, 107*(1), 87-95. doi: http://dx.doi.org/10.1016/S0377-2217(97)00221-X.

Hoekstra, J. M. (2001). *Designing for Safety: The Free Flight Air Traffic Management Concept* (Delft University of Technology).

Houacine, M., & Khardi, S. (2010). Gauss Pseudospectral Method for Less Noise and Fuel Consumption of Aircraft Operations. *Journal of Aircraft, 47*(6), 2152-2158. doi: http://dx.doi.org/10.2514/1.C031007.

IATA. (2011). *Vision 2050*. Singapore: International Air Transport Association.

ICAO. (2010). *Aviation's contribution to climate change*. Montreal: International Civil Aviation Organization.

Jackson, M. R. C. (2008). Role of Avionics in Trajectory Based Operations. Dans *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (pp. 3.A.1-1-3.A.1-9). doi: http://dx.doi.org/10.1109/dasc.2008.4702792

Jensen, L., Hansman, J. R., Venuti, J., & Reynolds, T. (2014). Commercial Airline Altitude Optimization Strategies for Reduced Cruise Fuel Consumption. Dans *14th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2014-3006

Jensen, L., Hansman, J. R., Venuti, J. C., & Reynolds, T. (2013, August 12-14, 201). *Commercial Airline Speed Optimization Strategies for Reduced Cruise Fuel Consumption* présentée à 2013 Aviation Technology, Integration, and Operations Conference, Los Angeles, USA. doi: http://dx.doi.org/10.2514/6.2013-4289.

Jensen, L., Tran, H., & Hansman, J. R. (2015). *Cruise Fuel Reduction Potential from Altitude and Speed Optimization in Global Airline Operations* présentée à Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal.

Jin, L., Cao, Y., & Sun, D. (2013). Investigation of Potential Fuel Savings Due to Continuous-Descent Approach. *Journal of Aircraft, 50*(3), 807-816. doi: 10.2514/1.C032000.

Johnson, C. M. (2011). Analysis of Top of Descent (TOD) uncertainty. Dans *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th* (pp. 2E3-1-2E3-10). doi: http://dx.doi.org/10.1109/dasc.2011.6096041

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459-471. doi: http://dx.doi.org/10.1007/s10898-007-9149-x.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8*(1), 687-697. doi: http://dx.doi.org/10.1016/j.asoc.2007.05.007.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Dans *Neural Networks, 1995. Proceedings., IEEE International Conference on* (Vol. 4, pp. 1942-1948). doi: http://dx.doi.org/10.1109/ICNN.1995.488968

Koreanschi, A., Sugar Gabor, O., & Botez, R. M. (2014). *New Numerical Study of Boundary Layer Behavior on A Morphing Wing-with-Aileron System* présentée à 32nd AIAA Applied Aerodynamics Conference, Atlanta, Georgia. doi: http://dx.doi.org/10.2514/6.2014-3170

Korn, B., Helmke, H., & Kuenz, A. (2006). 4D Trajectory Managemetn In The Extended TMA: Coupling AMAN And 4D FMS For Optimized Approach Trajectoreis. Dans P. I. Grant (Éd.), *25th Congress of International Council of the Aeronautical Sciences* (Vol. Paper ICAS 2006-8.8.2). Optimage Ltd.

Kwok-On, T., Anthony, W., & John, B. (2003). Continuous Descent Approach Procedure Development for Noise Abatement Tests at Louisville International Airport, KY. Dans *AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2003-6772.

Kwok-On, T., Daniel, B., & Anthony, W. (2006). *Development of Continuous Descent Arrival (CDA) Procedures for Dual-Runway Operations at Houston Intercontinental* présentée à 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO). doi: http://dx.doi.org/10.2514/6.2006-7750.

Liden, S. (1985). Practical Considerations in Optimal Flight Management Computations. Dans *American Control Conference, 1985* (pp. 675-681).

Liden, S. (1986). Practical considerations in optimal flight management computations. *Journal of Guidance, Control, and Dynamics, 9*(4), 427-432. doi: http://dx.doi.org/10.2514/3.20128.

Liden, S. (1992a). Optimum 4D guidance for long flights. Dans *Digital Avionics Systems Conference, 1992. Proceedings., IEEE/AIAA 11th* (pp. 262-267). doi: http://dx.doi.org/10.1109/dasc.1992.282146

Liden, S. (1992b). Optimum cruise profiles in the presence of winds. Dans *Digital Avionics Systems Conference, 1992. Proceedings., IEEE/AIAA 11th* (pp. 254-261). doi: http://dx.doi.org/10.1109/dasc.1992.282147

Liden, S. (1994). The evolution of Flight Management Systems. Dans *Digital Avionics Systems Conference, 1994. 13th DASC., AIAA/IEEE* (pp. 157-169).

Lovegren, J. A. (2011). *Estimation of potential aircraft fuel burn reduction in cruise via speed and altitude optimization strategies* (Massachusetts Institute of Technology).

Marasa, J. (2010). RNP is revolutionizing the instrument approach: Wings, NAVCanada.

McConnachie, D., Wollersheim, C., & Hansman, R. J. (2013). The Impact of Fuel Price on Airline Fuel Efficiency and Operations. Dans *2013 Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-429

McEnteggart, Q., & Whidborne, J. (2012). A multiobjective trajectory optimisation method for planning environmentally efficient trajectories. Dans *Control (CONTROL), 2012 UKACC International Conference on* (pp. 128-135). doi: http://dx.doi.org/10.1109/CONTROL.2012.6334618

Miyazawa, Y., Wickramasinghe, N. K., Harada, A., & Miyamoto, Y. (2013). Dynamic Programming Application to Airliner Four Dimensional Optimal Flight Trajectory. Dans *AIAA Guidance, Navigation, and Control (GNC) Conference*. Boston USA: American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-4969. R

Murrieta-Mendoza, A. (2013). *Vertical and Lateral Flight Optimization Algorithm and Missed Approach Cost Calculation.* (École de Technologie Supérieure, Montreal).

Murrieta-Mendoza, A., Beuze, B., Ternisien, L., & Botez, R. (2015). Branch & Bound-Based Algorithm for Aircraft VNAV Profile Reference Trajectory Optimization. Dans *15th AIAA Aviation Technology, Integration, and Operations Conference, Aviation Forum*. AIAA. doi: http://dx.doi.org/10.2514/6.2015-2280

Murrieta-Mendoza, A., Botez, R., & Ford, S. (2014). *Estimation of Fuel Consumption and Polluting Emissions Generated during the Missed Approach Procedure* présentée à The 33nd IASTED International Conference on Modelling, Identification, and Control (MIC 2014) Innsbruck, Austria. doi: http://dx.doi.org/10.2316/P.2014.809-040

Murrieta-Mendoza, A., & Botez, R. M. (2014a). Lateral Navigation Optimization Considering Winds And Temperatures For Fixed Altitude Cruise Using The Dijkstra's Algorithm. Dans *International Mechanical Engineering Congress & Exposition*. doi: http://dx.doi.org/10.1115/IMECE2014-37570

Murrieta-Mendoza, A., & Botez, R. M. (2014b). Vertical Navigation Trajectory Optimization Algorithm For A Commercial Aircraft. *AIAA/3AF Aircraft Noise and Emissions Reduction Symposium*. doi: http://dx.doi.org/10.2514/6.2014-3019.

Murrieta-Mendoza, A., & Botez, R. M. (2015). *Aircraft Vertical Route Optimization Deterministic Algorithm for a Flight Management System* présentée à SAE 2015 AeroTech Congress & Exhibition, Seattle, USA. doi: http://dx.doi.org/10.4271/2015-01-2541

Murrieta-Mendoza, A., & Botez, R. M. (2015). Methodology for Vertical-Navigation Flight-Trajectory Cost Calculation Using a Performance Database. *Journal of Aerospace Information Systems, 12*(8), 519-532. doi: http://dx.doi.org/10.2514/1.I010347.

Murrieta-Mendoza, A., Botez, R. M., & Ford, S. (2016). New method to compute the missed approach fuel consumption and its emissions. *The Aeronautical Journal, 120*(1228), 18. doi: http://dx.doi.org/10.1017/aer.2016.37

Murrieta-Mendoza, A., Bunel, A., & Botez, R. M. (2016). Aircraft Vertical Reference Trajectory Optimization With a RTA Constraint Using the ABC Algorithm. Dans *16th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2016-4208.

Murrieta-Mendoza, A., Demange, S., George, F., & Botez, R. M. (2015). Performance Database Creation Using a Flight D Simulator For Cessna Citation X Aircraft in Cruise Regime. Dans *The 34th IASTED International Conference on Modelling, Identification, and Control (MIC2015)*. IASTED. doi: http://dx.doi.org/10.2316/P.2015.826-028

Murrieta-Mendoza, A., Félix-Patrón, R. S., & Botez, R. M. (2015). *Flight Altitude Optimization Using Genetic Algorithms Considering Climb and Descent Costs in Cruise with Flight Plan Information* présentée à SAE 2015 AeroTech Congress & Exhibition, Seattle, USA. doi: http://dx.doi.org/10.4271/2015-01-2542

Murrieta-Mendoza, A., Hamy, A., & Botez, R. M. (2015). Mach Number Selection for Cruise Phase using the Ant Colony Optimization Algorithm With RTA Constrains. Dans *International Conference on Air Transport*.

Murrieta-Mendoza, A., Hamy, A., & Botez, R. M. (2016). Lateral Reference Trajectory Algorithm Using Ant Colony Optimization. Dans *16th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2016-4209.

Murrieta-Mendoza, A., Mugnier, P., & Botez, R. M. (2016, August 22-24). *Vertical Reference Trajectory Optimization and Simulation for a Commercial Aircraft* présentée à 11th International Conference on Modeling, Optimization and Simulation - MOSIM'16, Montreal, QC, Canada.

Murrieta-Mendoza, A., Romain, C., & Botez, R. M. (2016). Commercial Aircraft Lateral Flight Reference Trajectory Optimization. Dans *20th IFAC Symposium on Automatic Control in Aerospace ACA 2016* (Vol. 49, pp. 1-6). doi: http://dx.doi.org/10.1016/j.ifacol.2016.09.001.

Murrieta Mendoza, A., & Botez, R. (2014). Vertical Navigation Trajectory Optimization Algorithm For A Commercial Aircraft. Dans *AIAA/3AF Aircraft Noise and Emissions Reduction Symposium*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2014-3019

Murrieta Mendoza, A., Bunel, A., & Botez, R. (2015, November 13-14). *Aircraft Lateral Flight Optimization Using Artificial Bees Colony* présentée à International Conference on Air Transport INAIR 2015, Amsterdam, the Netherlands.

Murrieta Mendoza, A., Mugnier, P., & Botez, R. M. (2017). Vertical and Horizontal Flight Reference Trajectory Optimization for a Commercial Aircraft. Dans *Sci-Tech 2017 AIAA Guidance, Navigation, and Control Conference*. doi: http://dx.doi.org/10.2514/6.2017-1241

Nangia, R. (2006). Operations and aircraft design towards greener civil aviation using air-to-air refuelling. *Aeronautical Journal, 110*(1113), 705-722.

Ng, H. K., Sridhar, B., & Grabbe, S. (2014). Optimizing Aircraft Trajectories with Multiple Cruise Altitudes in the Presence of Winds. *Journal of Aerospace Information Systems, 11*(1), 35-47. doi: http://dx.doi.org/10.2514/1.I010084.

Nojoumi, H., Dincer, I., & Naterer, G. F. (2009). Greenhouse gas emissions assessment of hydrogen and kerosene-fueled aircraft propulsion. *International Journal of Hydrogen Energy, 34*(3), 1363-1369. doi: http://dx.doi.org/10.1016/j.ijhydene.2008.11.017.

Nolan, M. S. (2010). *Fundamentals of Air Traffic Control* (5th éd.). Delmar Publishers Inc.

Novak, D., Buckai, T., & Dadisic, T. (2009). Development, Design and Flight Test Evaluation of Continuous Descent Approach Procedure in FIR Zagreb. *Scientific Journal on Traffic and Transportation Research, 21*(5), 319-329. doi: http://dx.doi.org/10.7307/ptt.v21i5.247

Ojha, S. K. (1995). Optimization of Cruising Flights of Turbojet Aircraft. Dans *Flight Performance of Aircraft* (pp. 237-258). American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/5.9781600868535.0237.0258.

Oliveira, Í. R., Quachio, R., & Cugnasca, P. S. (2014). Improving Computation of Simulated Wind-Prediction Error for Air Traffic Applications. *Journal of Aerospace Information Systems, 11*(7), 423-432. doi: http://dx.doi.org/10.2514/1.I010121.

Pargett, D. M., & Ardema, M. D. (2007). Flight Path Optimization at Constant Altitude. *Journal of Guidance, Control, and Dynamics, 30*(4), 1197-1201. doi: http://dx.doi.org/10.2514/1.28954.

Ramasamy, S., Sabatini, R., Gardi, A. G., & Liu, Y. (2013). Novel Flight Management System for Real-Time 4-Dimensional Trajectory Based Operations: American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-4763.

Randt P., N., Jessberger, C., & Ploetner, K. O. (2015). Estimating the Fuel Saving Potential of Commercial Aircraft in Future Fleet-Development Scenarios. Dans AIAA (Éd.), *15th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2015-2435.

Ribeiro, V. F., Pamplona, D. A., Fregnani, J. A. T. G., Oliveira, Í. R. d., & Li, W. (2016). Modeling the swarm optimization to build effective Continuous Descent Arrival sequences. Dans *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 760-765). doi: http://dx.doi.org/10.1109/ITSC.2016.7795640

Rippel, E., Bar-Gill, A., & Shimkin, N. (2005). Fast Graph-Search Algorithms for General-Aviation Flight Trajectory Generation. *Journal of Guidance, Control, and Dynamics, 28*(4), 801-811. doi: http://dx.doi.org/10.2514/1.7370.

Robinson, E. (1978). Hydrocarbons in the atmosphere. *Pure and applied geophysics, 116*(2-3), 372-384. doi: http://dx.doi.org/10.1007/bf01636892.

Ruiz, S., Piera, M. A., Nosedal, J., & Ranieri, A. (2014). Strategic de-confliction in the presence of a large number of 4D trajectories using a causal modeling approach. *Transportation Research Part C: Emerging Technologies, 39*, 129-147. doi: http://dx.doi.org/10.1016/j.trc.2013.12.002.

Sabatini, R., Gardi, A., Ramasamy, S., Kistan, T., & Marino, M. (2010). Modern Avionics and ATM Systems for Green Operations. *Encyclopedia of Aerospace Engineering*. doi: http://dx.doi.org/10.1002/9780470686652.eae1064.

Sabuncuoglu, I., & Bayiz, M. (1999). Job shop scheduling with beam search. *European Journal of Operational Research, 118*(2), 390-412. doi: http://dx.doi.org/10.1016/S0377-2217(98)00319-1.

Sadovsky, A. V. (2014). Application of the Shortest-Path Problem to Routing Terminal Airspace Air Traffic. *Journal of Aerospace Information Systems, 11*(3), 118-130. doi: http://dx.doi.org/10.2514/1.I010074.

Salvat, N., Batailly, A., & Legrand, M. (2013). Modeling of Abradable Coating Removal in Aircraft Engines Through Delay Differential Equations. *Journal of Engineering for Gas Turbines and Power, 135*(10), 102102.

Schoemig, E. G., Armbruster, J., Boyle, D., Haraldsdottir, A., & Scharl, J. (2006). 3D Path Concept and Flight Management System (FMS) Trades. Dans *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA* (pp. 1-12). Portland, OR: IEEE. doi: http://dx.doi.org/10.1109/dasc.2006.313689

Sidibe, S., & Botez, R. M. (2013, April 30th - May 2nd). *Trajectory optimization of FMS-CMA 9000 by dynamic programming* présentée à ASI AÉRO 2013 conference, 60th Aeronautics Conference and AGM, Toronto, Canada.

Soler-Arnedo, M., Hansen, M., & Zou, B. (2013). Contrail Sensitive 4D Trajectory Planning with Flight Level Allocation Using Multiphase Mixed-Integer Optimal Control. Dans *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-5179.

Soler, M., Zou, B., & Hansen, M. (2014). Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach. *Transportation Research Part C: Emerging Technologies, 48*, 172-194. doi: http://dx.doi.org/10.1016/j.trc.2014.08.009.

Sprong, K. R., Klein, K. A., Shiotsuki, C., Arrighi, J., & Liu, S. (2008). Analysis of AIRE Continuous Descent Arrival Operations at Atlanta and Miami. Dans *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (pp. 3.A.5-1-3.A.5-13). doi: http://dx.doi.org/10.1109/dasc.2008.4702796

Sridhar, B., Ng, H., & Chen, N. (2013). Aircraft Trajectory Optimization and Contrails Avoidance in the Presence of Winds. *Journal of Guidance, Control, and Dynamics, 34*(5), 1577-1584. doi: http://dx.doi.org/10.2514/1.53378.

Stell, L. (2009). Flight Management System Prediction and Execution of Idle-Thrust Descents. Dans *Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th* (pp. 1.C.4-1-1.C.4-12). doi: http://dx.doi.org/10.1109/dasc.2009.5347570

Stell, L. (2010, 13 September 2010 - 15 September 2010). *Predictability of Top of Descent Location for Operational Idle-Thrust Descents* présentée à 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth USA. doi: doi: http://dx.doi.org/10.2514/6.2010-9116.

Theunissen, E., Rademaker, R., & Lambregts, T. (2011). Navigation system autonomy and integration in NextGen: Challenges and solutions. Dans *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th* (pp. 1-19). doi: http://dx.doi.org/10.1109/dasc.2011.6096268

Turgut, E. T., Cavcar, M., Usanmaz, O., Canarslanlar, A. O., Dogeroglu, T., Armutlu, K., & Yay, O. D. (2014). Fuel flow analysis for the cruise phase of commercial aircraft on

domestic routes. *Aerospace Science and Technology, 37*(0), 1-9. doi: http://dx.doi.org/10.1016/j.ast.2014.04.012.

Valenzuela, A., & Rivas, D. (2014). Optimization of Aircraft Cruise Procedures Using Discrete Trajectory Patterns. *Journal of Aircraft, 51*(5), 1632-1640. doi: http://dx.doi.org/10.2514/1.C032041.

Venkataraman, P. (2009). *Applied Optimization with MATLAB Programming*. Hoboken, New Jersey: John Wiley & Sons.

Villarroel, J., & Rodrigues, L. (2016). Optimal Control Framework for Cruise Economy Mode of Flight Management Systems. *Journal of Guidance, Control, and Dynamics*, 1-12. doi: http://dx.doi.org/10.2514/1.G001373.

Vincent, J.-B., & Botez, R. (2015). Systemic Modeling and Design Approach for Morphing Wing Aileron Controller Using Matlab/Simulink. Dans *AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2015-0904

Wang, H., Lyu, W., Yao, P., Liang, X., & Liu, C. (2015). Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. *Chinese Journal of Aeronautics, 28*(1), 229-239. doi: http://dx.doi.org/10.1016/j.cja.2014.12.031.

Wieland, F., Hunter, G., & Schleicher, D. (2008). The Implication of New Aircraft Types on the Next Generation Air Transportation System. Dans *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (pp. 2.C.2-1-2.C.2-8). St. Paul, MN doi: http://dx.doi.org/10.1109/dasc.2008.4702781

Wilburn, J. N., Perhinschi, M. G., & Wilburn, B. K. (2013a). Implementation of a 3-Dimensional Dubins-Based UAV Path Generation Algorithm. Dans *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-5232

Wilburn, J. N., Perhinschi, M. G., & Wilburn, B. K. (2013b). Implementation of Composite Clothoid Paths for Continuous Curvature Trajectory Generation for UAVs. Dans *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics. doi: http://dx.doi.org/10.2514/6.2013-5230

Williams, P., D. . (2016). Transatlantic flight times and climate change. *Environmental Research Letters, 11*(2), 024008. doi: http://dx.doi.org/10.1088/1748-9326/11/2/024008.

Yang, X.-S. (2010). *Nature-Inspired Metaheuristic Algorithms* (2nd éd.). Luniver Press.