DYNAMIC APPROACH TO WIND SENSITIVE OPTIMUM CRUISE PHASE FLIGHT PLANNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜRAY YILDIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
OPERATIONAL RESEARCH

SEPTEMBER, 2012

Approval of the thesis:

DYNAMIC APPROACH TO WIND SENSITIVE OPTIMUM CRUISE PHASE FLIGHT PLANNING

submitted by **GÜRAY YILDIZ** in partial fulfillment of the requirements for the degree of **Master of Science in Operational Research Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Çağlar Güven ______________________
Head of Department, **Operational Research**

Prof. Dr. Sinan Kayalıgil ______________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:
Prof. Dr. Ömer Kırca ______________________
Industrial Engineering Dept., METU

Prof. Dr. Sinan Kayalıgil ______________________
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Halit Oğuztüzün ______________________
Computer Engineering Dept., METU

Asst. Prof. Dr. Cem İyigün ______________________
Industrial Engineering Dept., METU

Özgür Koç, M.Sc. ______________________
Chief Technical Engineer, TUSAŞ

Date: 14.09.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Güray YILDIZ

Signature :

ABSTRACT

DYNAMIC APPROACH TO WIND SENSITIVE OPTİMUM CRUISE PHASE FLIGHT PLANNING

Yıldız, Güray
M.Sc., Operational Research Department
Supervisor: Prof. Dr. Sinan Kayalıgil

September, 2012, 86 pages

A Flight Management System (FMS) performs 4 Dimensional flight planning; Lateral Planning (Calculation of the latitude and longitudes of waypoints), Vertical Planning (Calculation of the altitudes of waypoints) and Temporal Planning (Calculation of Estimated Time of Arrival).

Correct and accurate calculation of 4D flight path and then guiding the pilot/airplane to track the route in specified accuracy limits in terms of lateral (i.e Required Navigational Performance - RNP), vertical (Reduced Vertical Seperation Minima - RVSM), and time (Required Time of Arrival - RTA) is what FMS performs in brief.

Any deviation of planned input values versus actual input values, especially during the emergency cases (i.e burning out one of engines etc.), causes the aircraft to deviate the plan and requires replanning now taking into consideration the current situation.

In emergency situations especially in Oceaning Flights (flights whose cruise phase lasts more than 5 hour is called as “Oceaning Flights”) Optimum Cruise Phase Flight Route Planning plays a vital role.

In avionics domain “Optimum” does not mean “shortest path” mainly due to the effect of weather data as wind speed and direction directly affects the ground speed.

In the scope of the current thesis, an algorithm employing dynamic programming paradigms will be designed and implemented to find the optimum flight route planning. A top down approach by making use of aircraft route planning ontology will be implemented to fill the gap between the flight plan specific domain knowledge and optimization techniques employed. Whereas the algorithm will be generic by encapsulating the aircraft’s performance characteristics; it will be evaluated on C-130 aircraft.

Keywords: Flight planning, Emergency landing, A* algorithm, A star algorithm, Ontology

ÖZ

RÜZGAR HASSAS OPTİMAL SEYİR FAZI UÇUŞ PLANLAMASINA DİNAMİK YAKLAŞIM

Yıldız, Güray
Yüksek Lisans, Yöneylem Araştırması Bölümü
Tez Yöneticisi: Prof. Dr. Sinan Kayalıgil

Eylül 2012, 86 sayfa

Bir Uçuş Yönetim Sistemi (UYS) 4 boyutlu uçuş planlaması gerçekleştirmektedir; Yatay Planlama (Uçuş Kontrol Noktalarının enlem ve boylamlarının hesaplanması), Dikey Planlama (Uçuş Kontrol Noktalarının yüksekliklerinin hesaplanması) ve Zaman Planlaması (Uçuş Kontrol Noktalarına tahmini varış zamanı).

4 boyutlu uçuş planlamasının doğru ve hassas olarak yapılması ve sonrasında uçağın rotayı belirli yatay, dikey ve zaman hassasiyet sınırları içinde takip etmesinde pilotu yönlendirmesi UYS'nin kısa bir özetidir.

Planlanan girdi değerleri ile gerçekleşen değerler arasındaki en ufak bir sapma, özellikle acil durumlarda (motorlardan birinin yanması vb.), uçağın plandan sapmasına mevcut durumu göz önüne alacak şekilde yeniden planlama yapılmasını gerekli kılar.

Acil durumlarda, özellikle Okyanus Uçuşlarında (seyir süresi 5 saatten fazla olan uçuşlar "Okyanus Uçuşu" olarak adlandırılmaktadır) Optimal Seyir Fazı Uçuş Rota Planlaması hayati bir öneme sahiptir.

Havacılık alanında, özellikle rüzgar şiddeti ve yönünün yer hızını doğrudan etkilemesi sebebiyle, “optimal” “en kısa yol” anlamına gelmemektedir.

Mevcut tez kapsamında, optimal uçuş rota planlaması yapan dinamik programlama mantığına dayalı bir algoritma tasarlanacak ve gerçekleştirilecektir. Uçuş planlamasına yönelik alan bilgisi ile uygulanacak en iyileme çözüm teknikleri arasındaki boşluğu uçak rota planlaması varlıkbilim çalışmasından yararlanılarak doldurulacaktır. Geliştirilecek olan algoritma, uçağa özel performans özelliklerinin izole bir paket olarak ele alınması sayesinde, tüm uçaklara uygulanacak özellikte olmakla birlikte sonuçlar C-130 uçağı özelinde elde edilecek ve değerlendirilecektir.

Anahtar kelimeler: Uçuş planlaması, Acil durum iniş, A* algoritması, Varlıkbilim

To My Wife Şule and My Daughter Defne

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to his supervisor Prof. Dr. Sinan Kayalıgil for his guidance, advice, criticism, encouragements and patience throughout the research.

I appreciate the valuable feedback of the examining committee; Dr. Ömer Kırca, Dr. Halit Oğuztüzün, Dr. Cem İyigün, and Özgür Koç.

I am also grateful to Dr. Sedef Meral; she always believed in me completing my thesis and motivated in the whole period.

My last three and a half years as a senior software engineer in TUSAŞ have shaped my area of interest and knowledge in Flight Management Systems. During my thesis work I am greatly indebted to my manager Uğur Çakır and my team members Hakan Yılmaz, Çağrı Toroslu, Seyfullah Arı and Hatice Erat for their encouragement and support.

Last and the most I would also like to express my deepest gratitude to my beatifull wife Şule and my lovely daughteer Defne. Without my wife's endless support I could not afford even intending to continue my thesis.

TABLE OF CONTENTS

ABSTRACT....iv

ÖZ....vi

ACKNOWLEDGEMENTS....ix

TABLE OF CONTENTS....x

LIST OF TABLES....xii

LIST OF FIGURES....xiii

LIST OF ABBREVIATIONS....xiii

CHAPTERS

1. INTRODUCTION 1

1.1 FMS Functionalities and Route Planning 1

1.2 Scope and Objective 5

1.3 Outline of the Study 6

2. BACKGROUND AND THE PROBLEM DOMAIN 7

2.1 Flight Phases 7

2.2 Performance Analysis 9

2.3 Meteorology Data Model and Use Cases 12

2.3.1 Meteorology Data Model 12

2.3.2 Meteorology Use Cases in Route Planning 14

2.4 FMS Cruise Calculations 16

2.4.1 Altitude Calculations: 17

2.4.2 Time and Fuel Calculations 19

2.5 Route Planning Ontology 22

3.LITERATURE SURVEY 25

4. PROBLEM DEFINITION AND MODELING 29

4.1 Modeling the Network and Cost Structure 29
4.2 Assumptions in Modeling 32
4.3 Dynamic Programming Model Definition 33
4.4 Dynamic Programming Algorithm 36
4.4.1 A* Algorithm 36
4.4.2 Coarse to Fine Dynamic Programming (CFDP) Algorithm 38
5. SOLUTION 40
5.1 Calculation Applying CFDP 40
5.2 A* Algorithm Pseudocode 44
5.3 Lower Bound Techniques 46
5.3.1 Take LowerBoundEstimate As 0 46
5.3.2 Use Maximum Wind Technique 47
5.3.3 Spiral Technique 48
5.3.4 CFDP Layers Technique 52
5.4 Development 55
6. CASE DEMONSTRATION 61
6.1 Test Case 1 63
6.2 Test Case 2 66
6.3 Test Case 3 68
6.4 Test Case 4 71
6.5 Test Case 5 73
6.6 Test Case 6 75
6.7 Test Case 7 77
6.8 Test Case 8 78
7. CONCLUSION 81
REFERENCES 83
APPENDIX A. OBJECT ORIENTED APPROACH 85

LIST OF TABLES

TABLES

Table 1 C-130 Flight Phase Characteristics 8

Table 2 Fuel Flow Per Engine Pound Per Hour where Altitude is 28000 feet 11

Table 3 TAS (knots) where Altitude is 28000 feet 12

Table 4 State Dimensions per State Level 42

Table 5 CFDPH State Dimensions per State Level 52

Table 6 Test Cases and Purposes 61

Table 7 Test Case 1 Results 63

Table 8 Test Case 2 Results 66

Table 9 Test Case 3 Results 68

Table 10 Test Case 4 Results 71

Table 11 Test Case 5 Results 73

Table 12 Test Case 6 Results 75

Table 13 Test Case 7 Results 77

LIST OF FIGURES

FIGURES

Figure 1 Navigation System Block Diagram 1
Figure 2 Flight Plan Lateral View 3
Figure 3 Flight Plan Vertical View 3
Figure 4 Flight Phases 7
Figure 5 Distance to Climb Performance Chart 11
Figure 6 Meteorology Grib File Structure 13
Figure 7 Tail Wind versus Head Wind 15
Figure 8 TAS to GS Conversion 16
Figure 9 Altimetry Terminology 18
Figure 10 Time and Fuel Calculations for a Level Leg 20
Figure 11 Time and Fuel Calculations for An Altitude Change Leg (Vertical View) 21
Figure 12 Concepts and Their Relations of ONTRP 24
Figure 13 Gridded Network Structure 30
Figure 14 Types of Neighbor Cells 31
Figure 15 CFDP Iterative Solution Process 39
Figure 16 Original Trellis Graph G = (λ, ξ, C) 40
Figure 17 Search Space Pruning and Orientation 44
Figure 18 Spiral Technique 50
Figure 19 Finding Maximum Wind Using Squares 51
Figure 20 Finding Maximum Wind Using Caches 51
Figure 21 CFDP As a Lower Bound Technique in A* Algorithm 53
Figure 22 Horizontal versus Diagonal Movement 54
Figure 23 Sample Screen Shot of Flight Planner 57

Figure 24 Sample Screen Shot of Flight Planner 58
Figure 25 Path Result Summary 60
Figure 26 testCase1-scenario1 64
Figure 27 testCase1-scenario2 64
Figure 28 testCase1-scenario3 65
Figure 29 testCase1-scenario4 65
Figure 30 testCase2-scenario1 67
Figure 31 testCase2-scenario2 67
Figure 32 testCase2-scenario3 68
Figure 33 testCase3-scenario1 69
Figure 34 testCase3-scenario2 70
Figure 35 testCase3-scenario3 70
Figure 36 testCase3-scenario4 71
Figure 37 testCase4-scenario1 72
Figure 38 testCase4-scenario2 73
Figure 39 testCase4-scenario1 74
Figure 40 testCase5-scenario2 74
Figure 41 testCase5-scenario3 75
Figure 42 testCase6-scenario1 76
Figure 43 testCase6-scenario2 76
Figure 44 testCase7-scenario1 77
Figure 45 testCase7-scenario2 78
Figure 46 testCase8-Legacy$_{ETP}$ 79
Figure 47 testCase8-New$_{ETP}$ 80
Figure 48 Class Diagram 85
Figure 49 Initialization Sequence Diagram 86
Figure 50 Optimization Sequence Diagram 86

LIST OF ABBREVIATIONS

AGL: Above Ground Level

BOD: Bottom of Descent

CFDP: Coarse to Fine Dynamic Programming

COTS: Commercial off the Shelf

DP: Dynamic Programming

DOC: Direct Operating Cost

ETA: Estimated Time of Arrival

ETD: Expected Time of Departure

ETP: Equal Time Point

FMS: Flight Management System

FL: Flight Level,

ft: Feet

lb: Libre (1 Libre = 1 pound = 453 gram)

MSL: Mean Sea Level

NRP: No Return Point

RP: Route Planning

RNP: Required Navigational Performance

RTA: Required Time of Arrival

RVSM: Reduced Vertical Seperation Minima

TAS: True Air Speed

TIT: Turbine Inline Temperature

TOC: Top of Climb

TOD: Top of Descent

CHAPTER 1

INTRODUCTION

This chapter will introduce Flight Management System (FMS) functionalities and route planning problem and give the scope and the outline of the thesis.

1.1 FMS Functionalities and Route Planning

FMS constitutes a crucial part of a Navigation System as shown in Figure 1[1].

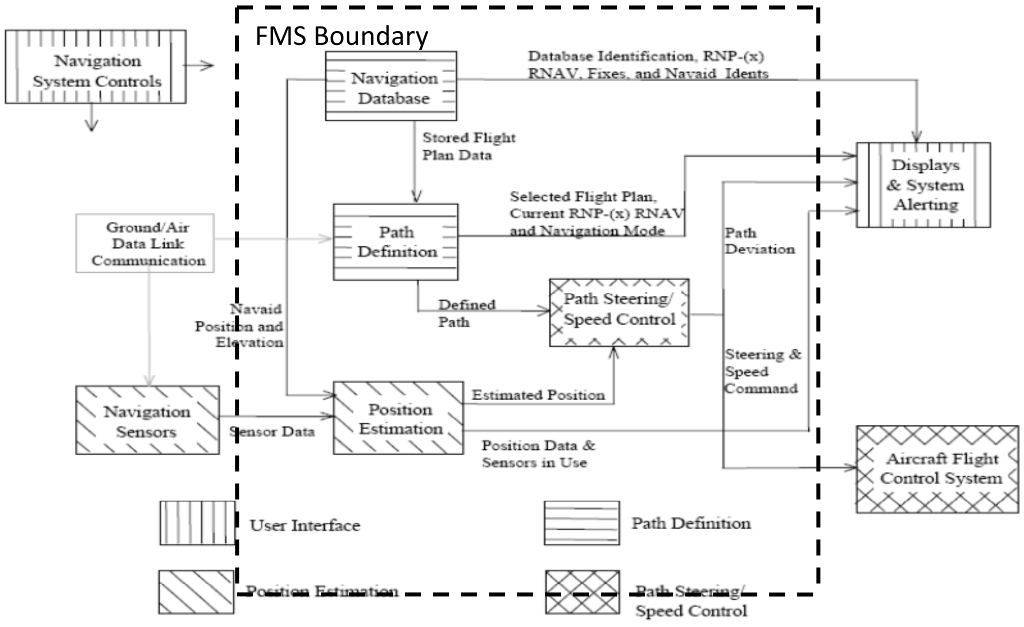

FMS Boundary
Navigation System Controls
Navigation Database
Database Identification, RNP-(x) RNAV, Fixes, and Navaid Idents
Stored Flight Plan Data
Displays & System Alerting
Ground/Air Data Link Communication
Path Definition
Selected Flight Plan, Current RNP-(x) RNAV and Navigation Mode
Path Deviation
Navaid Position and Elevation
Defined Path
Path Steering/ Speed Control
Steering & Speed Command
Navigation Sensors
Sensor Data
Position Estimation
Estimated Position
Position Data & Sensors in Use
Aircraft Flight Control System
User Interface
Path Definition
Position Estimation
Path Steering/ Speed Control

Figure 1 Navigation System Block Diagram

[1] The figure is adapted from DO-236 [1] Figure 1-1.

DO-236B states that [1];

> *"the path definition function computes the defined path to be flown in relation to the vertical, horizontal, and time dimensions. The elements of a defined path include, fixes, path constraints (time, altitude, speed, location), and leg types, such that a seamless horizontal and vertical path between the start and end of the planned flight can be achieved."*

Hence, using the DO-236B terminology, we can define a Flight Plan as the defined path to be flown in relation to the vertical, horizontal, and time dimensions such that a seamless horizontal and vertical path between the departure airport[2] and destination airport can be achieved.

A Flight Management System (FMS) calculates and executes a Flight Plan in 4 Dimension, meaning that;

- Lateral Planning: Calculation of the latitude and longitudes of waypoints,
- Figure 2 illustrates lateral view of a sample flight plan for a C-130 type aircraft.
- Vertical Planning: Calculation of the altitudes of waypoints and the calculation of the vertical trajectory, Figure 3 illustrates vertical view of a sample flight plan for a C-130 type aircraft.
- Temporal Planning: Calculation of leg times and hence, knowing the Expected Time of Departure (ETD), Estimated Time of Arrival (ETA)

[2] It should be noted that a Flight Plan does not have to start from departure airport, in case a replanning situation occurs in air the start position is not the departure point but the current position.

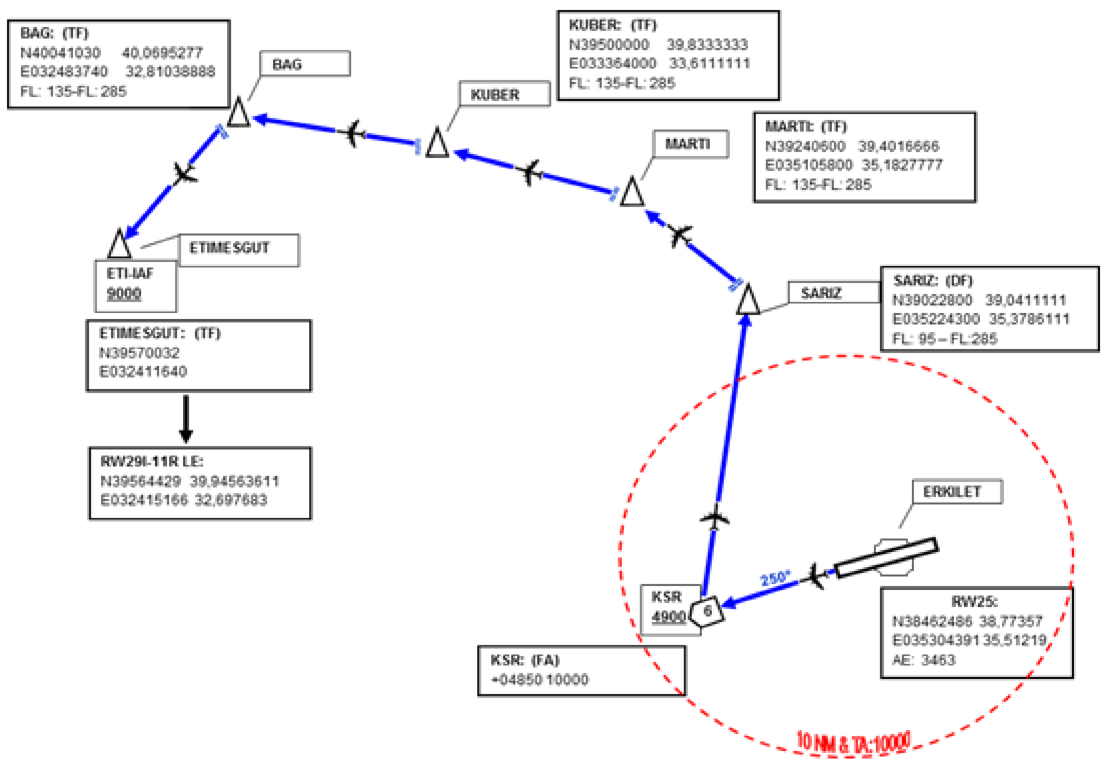


Figure 2 Flight Plan Lateral View

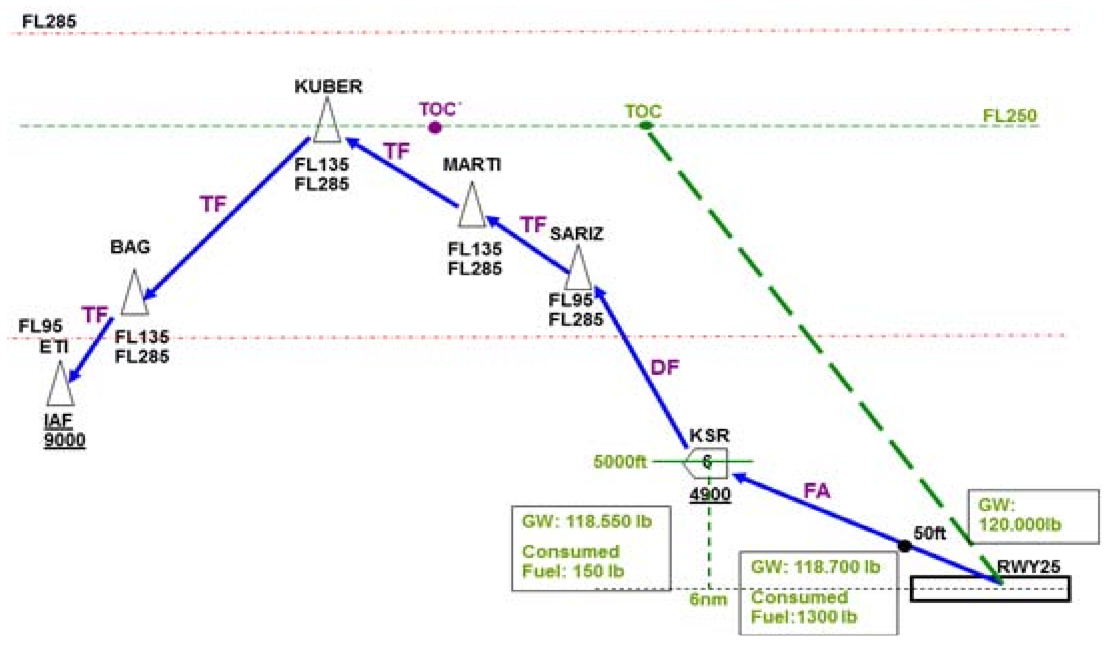


Figure 3 Flight Plan Vertical View

FMS functionalities will be summarized from two perspectives; dynamic and functional views:

1. Dynamic View: FMS supports aircraft crew during both the planning and the flight execution phases:
    - Planning Phase: Covers the definition and calculation of a flight plan applied in 4 dimensions. There may be more than one Flight Plan calculated, where the one is called as *Active Flight Plan* and the others are called as *Secondary* or *Standby Flight Plan*.
    - Execution Phase: FMS guides the pilot (i.e. issues an aural warning in case the altitude of the aircraft goes against the allowable limits) and the sensor(s) (i.e. releases the Autopilot a roll command to follow the desired track) taking the Active Flight Plan as the basis.
2. Functional View: FMS capabilities that are of relevance to the scope of this thesis are summarized where the others are skipped for brevity:
    - Navigation: The navigation function provides the following navigational outputs:
        - Estimated Aircraft Position (latitude, longitude, altitude)
        - Aircraft speed and speed conversions
        - Cross Track Distance, Desired Track and Drift Angle
        - Required Navigation Performance (RNP) and an estimate of actual performance
        - Coordinate conversions
        - Fly-by / Fly-over Turn Anticipations
        - Navigation Database (leg types, procedure and pattern)
        - Search and Rescue (SAR) Patterns
    - Performance: Aircraft performance calculations produce outputs that are either optimum/maximum or expected values with respect to aircraft's structural properties, to name a few:
        - Cruise ceiling: The ideal altitude for fuel efficiency
        - Service ceiling: The maximum altitude where aircraft is still capable to climb at a minimum threshold value (100 feet per minute)
        - Climb/descent time and fuel from an initial to final altitude etc.

- Flight Planning
    - Performance Analysis
    - Threat Analysis
    - Terrain Analysis
    - Notice to Marines (NOTAM), Air Coordination Measures (ACM) Analysis

Flight Planning and *Route Planning* terms are in general used equivalently, where more specifically *Route Planning Problem* means to find out an optimal path which has an objective and subjects to some constraints from the start to the end of a vehicle's journey.

The "best" route differs from flight to flight. Consider a pilot flying a Search and Rescue mission where he aims to maximize flying endurance time at low speeds. The pilot also wants to provide terrain clearance avoiding flying within a specified safety threshold near terrain and man made obstacles. On the other hand for air drop missions the pilot's first priority is to fly a safer route such that minimizing visibility from enemy threats. For longer flights the required fuel on board or the required time of arrival may become the crucial factors.

Factors as sampled above and other factors can be combined and prioritized and either become the optimization criteria and/or constitutes the constraints of the route planning.

1.2 Scope and Objective

In this thesis we try to automate planning the optimum cruise path of an aircraft for a given start to goal node for either the time or fuel required. Although this is a well known *shortest path* problem and has been taken before by other studies some new methods are used in this thesis to set this work apart and provide some contributions.

The first major point of the study is the application of a systematic approach by making use of the existing route planning ontology in the problem solution lifecycle, namely;

- Conceptual analysis,
- Formulation of the model,
- Design of the algorithm,
- Implementation of the algorithm

The real world problem on hand will be evaluated with respect to the existing route planning ontology and the formulation of the problem and the suitable optimization techniques will be determined as a result. Usage of ontology is extended by applying Object Oriented Analysis and Design (OOAD) techniques during the implementation of the algorithm.

The second major contribution is the design of "efficient" heuristics to be able to overcome the *curse of dimensionality* problem. The accurate modeling of the real world problem results in increase in the complexity of the mathematical model which requires application of heuristics trading off optimality to speed up the solution process.

The practical contribution is the use of the developed algorithm in C-130 type aircraft's FMS and to evaluate its effect on emergency replanning case and oceaning calculations and to measure the improvement with the existing situation.

1.3 Outline of the Study

The rest of the thesis will be organized as follows:

- Chapter 2 explains the domain knowledge detailing the background information, concepts and work related to FMS cruise calculations and route planning,
- Chapter 3 covers the literature survey,
- Chapter 4 declares the assumptions and gives the formal definition of the problem,
- Chapter 5 covers the design of the algorithm
- Chapter 6 covers the computational experience and compares the effect of changing parameters (i.e. different heuristics, different edge relationship etc.) on the output
- Chapter 7 discusses on the operational gain on the application of the algorithm and introduces possible areas for the future work

CHAPTER 2

BACKGROUND AND THE PROBLEM DOMAIN

This chapter will detail the background information, concepts and work related to FMS cruise calculations and route planning.

2.1 Flight Phases

A Flight Plan covers the following phases of aircraft's journey:

- Takeoff,
- Climb,
- Cruise,
- Descent,
- Approach and Landing

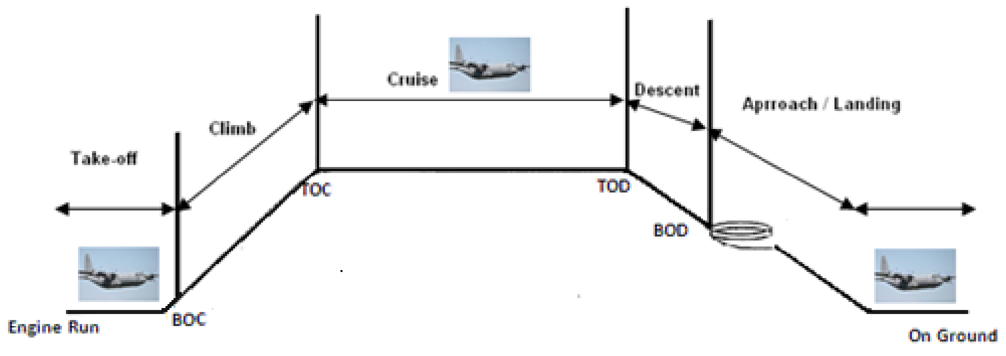


Figure 4 Flight Phases

The start and end of each phase is determined by *pseudo* waypoints, i.e. waypoints not specified by the pilot but auto calculated and inserted into the Flight Plan by FMS.

These pseudo waypoints from start to end of the flight plan, as illustrated in Figure 4, are:

- Bottom of Climb (BOC): The point where aircraft reaches 50 feet (ft) above the runway elevation
- Top of Climb (TOC): The point where aircraft reaches the *Cruise Altitude,* either pilot selected or FMS suggested
- Top of Descent (TOD): The point where the aircraft starts descending and is not allowed to perform any climbing then on
- Bottom of Descent (BOD): The point which is 1200 ft Above Ground Level (AGL) and 4 NM distance far away from destination runway

Each phase has its intrinsic characteristics; taking take off phase as an example the runway specifications (i.e. runway length, surface type etc.) and initial gross weight are the crucial parameters and the main concern is to make "Take off or Abort" decision by applying a formal procedure called the *Take off Decision Tree*.

Table 1 C-130 Flight Phase Characteristics[3]

Flight Phase	Altitude (ft)	Time (minute)	Distance (NM)
Takeoff	[0, 50] (AGL)	0,2	1
Climb	[50 (AGL), [20000, 32000]]	17.5	55 (when cruise altitude is 25000)
Cruise	[20000, 32000]	900 (15 hours)	4000
Descent	[[20000, 32000], 1200 (AGL)]	2.3	11
Approach/Landing	[1200, 0] (AGL)	1.5	8

[3] Table is constructed for C-130 aircraft with the following assumptions:
- Taking initial Gross Weight of 120000 libres (lb),
- No corrections is applied for drag, temperature deviation etc.,
- No terminal area procedure is used,
- No low level flight in Cruise phase,
- Time and Distance values for Climb and Descent phases are calculated for a cruise altitude of 25000 feet,
- Maximum range of aircraft as 4000 NM

As it can be inferred from Table 1 while terrain and obstacle clearances are important for takeoff and climb phases, they are of secondary importance for cruise phase (as long as it is not a low level flight). In a similar manner, comparing the short takeoff duration with long cruise duration, it can be seen that while fuel required is not of a main concern in the Takeoff phase it is of a crucial importance in the cruise phase.

2.2 Performance Analysis

In the flight planning process all lateral, vertical and temporal dimensions are correlated with each other as the simplified list below prevails this dependency:

- Aircraft's performance characteristics (i.e. suggested True Air Speed –TAS-, fuel flow) mainly depends on the gross weight, altitude, and temperature,
- Aircraft's ground speed depends on TAS, wind speed, and tail wind component (which depends on wind direction and the aircraft's course),
- Atmospheric data changes with respect to position, altitude and time,
- Aircraft gets lighter as it consumes fuel

Performance analysis is the process of constructing a consistent Flight Plan in all 4 dimensions complying with aircraft performance[4] and all flight plan specified altitude constraints, speed restrictions and specified gradient constraints reflecting rate of vertical descent speed.

Performance analysis calculates and assures a trajectory that can be flyable by the aircraft. Only flight plans passed from performance analysis process can be activated and becomes the Active Flight Plan.

Performance analysis is built upon performance characteristics of the aircraft. Performance characteristics define;

- The feasibility region of what the aircraft is capable of; i.e. operational flight envelope, maximum speed, minimum speed (stall speed) etc.
- Suggests ideal values for a given input set; i.e. Cruise ceiling (the ideal altitude in terms of fuel efficiency), climb speed etc.

[4] Aircraft performance model is the outcomes that reflect the extent of achievement on many scales of measurement

- Calculates the expected value of an output for a given input set; i.e. required climb time and fuel for a given start to end altitude etc.

There are two approaches in getting the performance characteristics of an aircraft;

- Analytical: Either a mathematical model is constructed,
- Empirical: Flight tests are performed and the results are documented on charts

For C-130 aircrafts over 2000 hours of flight test is performed and the result is documented in C-130 Performance Manual [2]. A sample chart is given in Figure 5[5]. In the sample chart the calculation process of getting the required climb distance from the sea level to 16000 ft will be shown. The gross weight is 138000 lb and temperature deviation is +50 C°:

- Since the upper part of the chart is constructed in International Standard Atmosphere (ISA, where the temperature deviation is 0) conditions; in the first step the gross weight is adjusted to 205,000 lb to account for the temperature deviation effect,
- Then, since there is no specific curve for 16000 ft, the result of 37 NM is calculated by interpolating the intersection of vertical line (GW = 205,000) with "20" and "15" curves for distance to climb.

[5] The output scale of the figure is erased so as not to expose the restricted content to the public. In general, all outputs adapted from C-130 Performance Manual is modified to meet the security clearence level of the document.

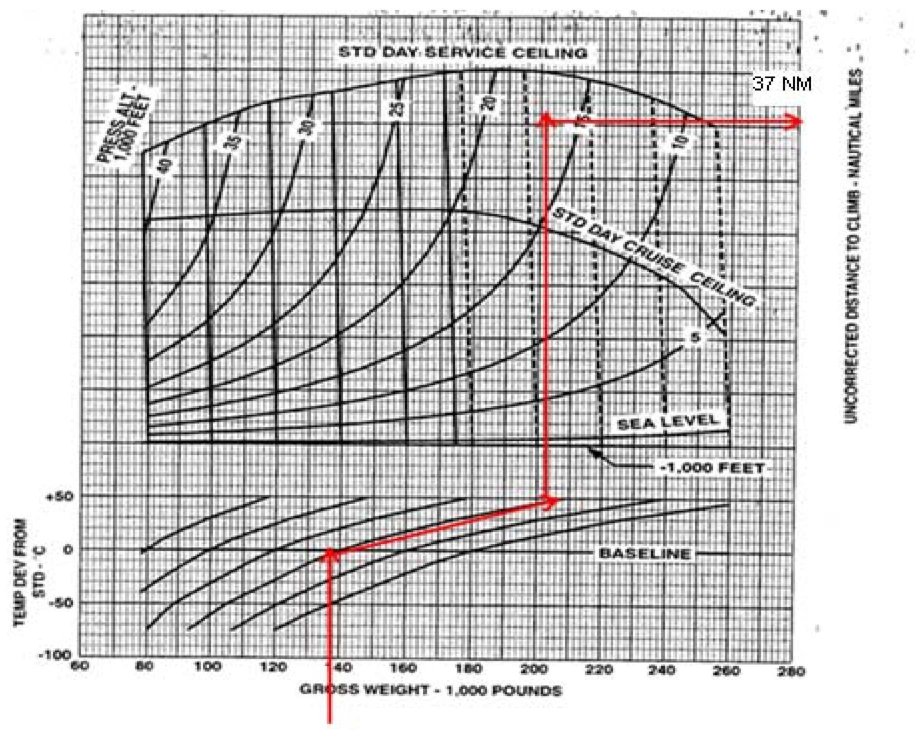
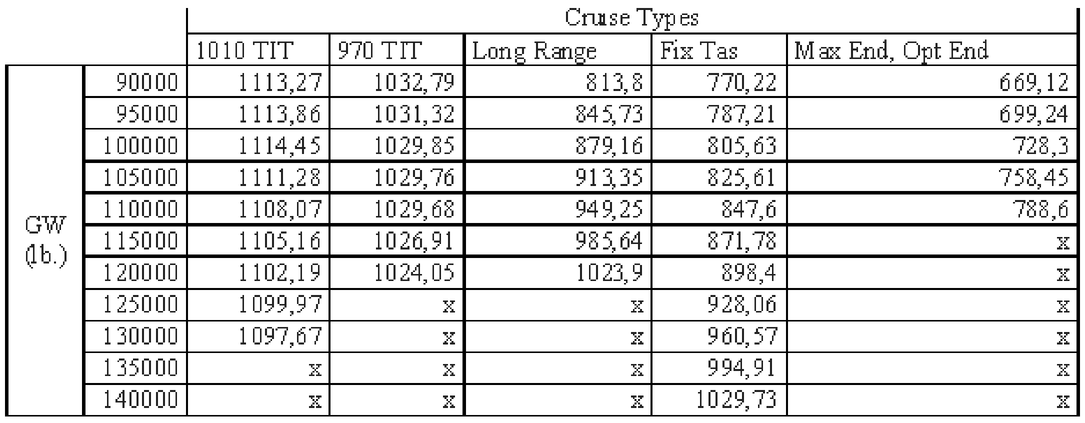


Figure 5 Distance to Climb Performance Chart

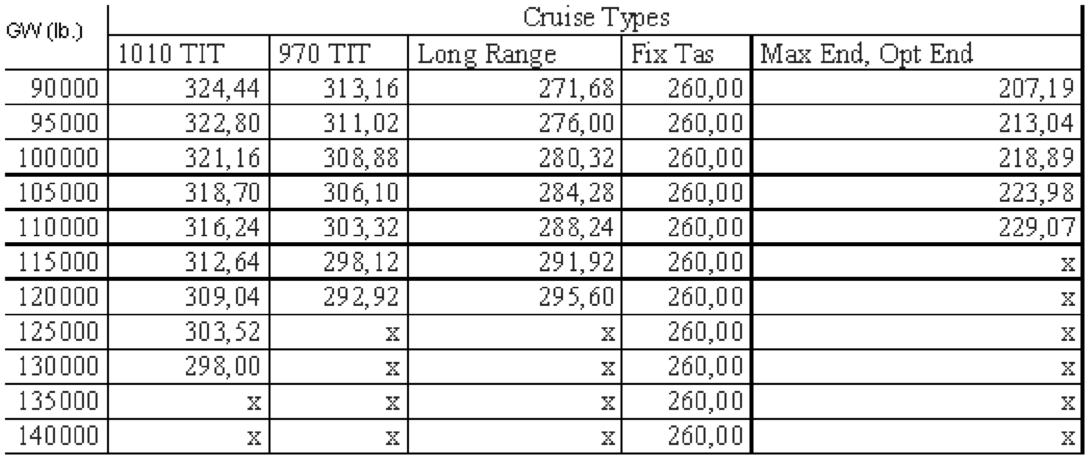
FMS uses a digitized performance database for performance analysis inquiries. In the scope of the thesis, look up tables will be employed for modeling the performance data model of the aircraft. Two samples are given in Table 2 and Table 3.

Table 2 Fuel Flow Per Engine Pound Per Hour where Altitude is 28000 feet



		Cruise Types				
		1010 TIT	970 TIT	Long Range	Fix Tas	Max End, Opt End
GW (lb.)	90000	1113,27	1032,79	813,8	770,22	669,12
	95000	1113,86	1031,32	845,73	787,21	699,24
	100000	1114,45	1029,85	879,16	805,63	728,3
	105000	1111,28	1029,76	913,35	825,61	758,45
	110000	1108,07	1029,68	949,25	847,6	788,6
	115000	1105,16	1026,91	985,64	871,78	x
	120000	1102,19	1024,05	1023,9	898,4	x
	125000	1099,97	x	x	928,06	x
	130000	1097,67	x	x	960,57	x
	135000	x	x	x	994,91	x
	140000	x	x	x	1029,73	x

Table 3 TAS (knots) where Altitude is 28000 feet



GW (lb.)	Cruise Types				
	1010 TIT	970 TIT	Long Range	Fix Tas	Max End, Opt End
90000	324,44	313,16	271,68	260,00	207,19
95000	322,80	311,02	276,00	260,00	213,04
100000	321,16	308,88	280,32	260,00	218,89
105000	318,70	306,10	284,28	260,00	223,98
110000	316,24	303,32	288,24	260,00	229,07
115000	312,64	298,12	291,92	260,00	x
120000	309,04	292,92	295,60	260,00	x
125000	303,52	x	x	260,00	x
130000	298,00	x	x	260,00	x
135000	x	x	x	260,00	x
140000	x	x	x	260,00	x

2.3 Meteorology Data Model and Use Cases

In this section meteorology data model and meteorology use case in route planning will be explained.

2.3.1 Meteorology Data Model

It is stated in ARINC 702 [3] that;

> *"Wind models for use in the cruise segment should allow for the entry of wind (magnitude and bearing) at a waypoint; a single value or multiple wind/altitude pairs. Systems should merge these entries with current winds obtained from sensor data in a method which gives a heavier weighting to sensed winds close to the aircraft.*
>
> *A more advanced representation of wind data in the Flight Management Function is the use of a grid wind model which may be up to a four-dimensional definition of wind. The grid winds would not be tied to waypoints in the flight plan, but associated with latitude longitude regions similar to a magnetic variation model."*

As the ease of getting digitized atmospheric forecast data has improved and volatile/nonvolatile memory capacity of the avionics hardware has grown new generation FMSs supports grid wind model.

In a Flight Management System (FMS), meteorology data (wind direction, wind speed, temperature) is inquired by the following parameters:

- Position: Latitude and longitude of the position where meteorology data will be acquired
- Pressure Altitude: Pressure altitude where meteorology data will be acquired
- Time : Period where meteorology data will be acquired.

For a given position and a time period, cell location is found from a “grib file structure” as shown in Figure 6.

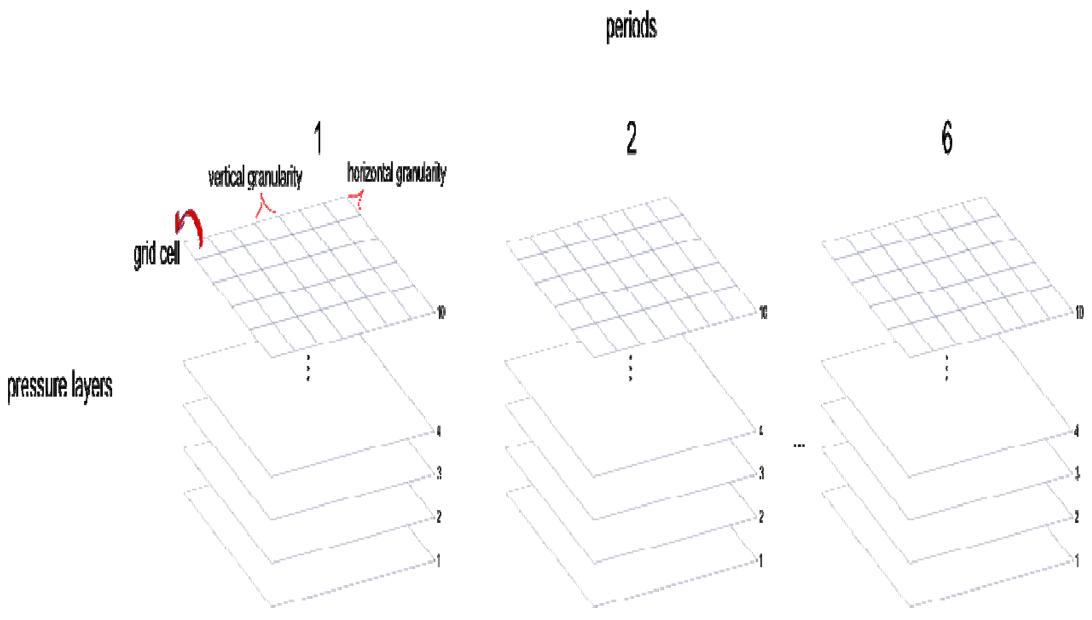


Figure 6 Meteorology Grib File Structure

Figure 6 explains how meteorology file (grib.bin) is formed. The earth surface is divided into grids, where each grid is determined by vertical and horizontal granularities.

It must be noted that near the equator a grid of 1x1 degree is approximately equal to 60*60 NM^2 and the resolution of the raw form of grib weather forecast data is 20 NM. To support 20 NM resolution of the original weather forecast data using geographic units (i.e.

degree of latitude &longitude) the vertical and horizontal granularities are not fixed but varied from 0.33 to 3 degrees depending on the latitude of the lower left corner of the grib file.

Granularity shall be identical in each time period and at each pressure (i.e. altitude) layer. There are 6 time periods of 3 hours granularity and 10 pressure layers in a grib file. Pressure altitude layers are discretized with the following fixed enumerations (in hectoPascal):

1. 1000 hPa ==> (1013.25 - 1000) X 30 (ft/hPa) = 397.5 ft.
2. 850 hPa ==> (1013.25 - 850) X 30 (ft/hPa) = 4897.5 ft.
3. 700 hPa ==> (1013.25 - 700) X 30 (ft/hPa) = 9397.5 ft.
4. 500 hPa ==> (1013.25 - 500) X 30 (ft/hPa) = 15397,5 ft.
5. 400 hPa ==> (1013.25 - 400) X 30 (ft/hPa) = 18397,5 ft
6. 300 hPa ==> (1013.25 - 300) X 30 (ft/hPa) = 21397,5 ft
7. 250 hPa ==> (1013.25 - 250) X 30 (ft/hPa) = 22897,5 ft.
8. 200 hPa ==> (1013.25 - 200) X 30 (ft/hPa) = 24397,5 ft.
9. 150 hPa ==> (1013.25 - 150) X 30 (ft/hPa) = 25897,5 ft.
10. 100 hPa ==> (1013.25 - 100) X 30 (ft/hPa) = 27397,5 ft.

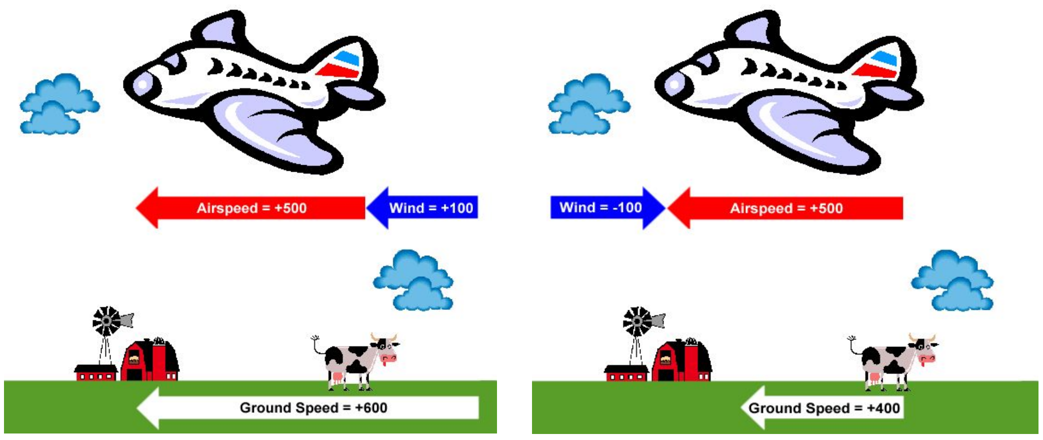
If a pressure altitude is queried different from the above fixed values the altitude of the nearest fixed enumeration value will be returned

2.3.2 Meteorology Use Cases in Route Planning

Use cases of meteorology data is as follows:

1. Planning Phase
   - Calculation of True Air Speed (TAS) due to temperature deviation effect
   - Calculation of planned ground speed via vector summing TAS-Course vector and Wind Speed-Direction vector
2. Execution Phase
   - Calculating current wind direction, speed and Outside Air Temperature (OAT)
   - Calculating waypoint’s wind direction, speed, and OAT

Wind Effect on Ground Speed at the planning phase deserves a special explanation since it outweighs the other uses cases of the meteorology data, and it is a direct factor in this thesis.



A Tail Wind B Head Wind

Figure 7 Tail Wind versus Head Wind

Wind has a direct affect on the ground speed of the aircraft as the two extreme cases shown in Figure 7 illustrate.

Generalizing the situation and switching from illustration to physics the effect of wind on the ground speed of an aircraft is given by a vector sum:

- TAS: Where the magnitude is TAS (in knots) and the direction is the heading[6]
- WIND: Where the magnitude is wind speed (in knots) and the direction is the wind direction (where the wind is blowing to

[6] The longitudinal axis of an airplane (where the nose of the aircraft is pointing to)

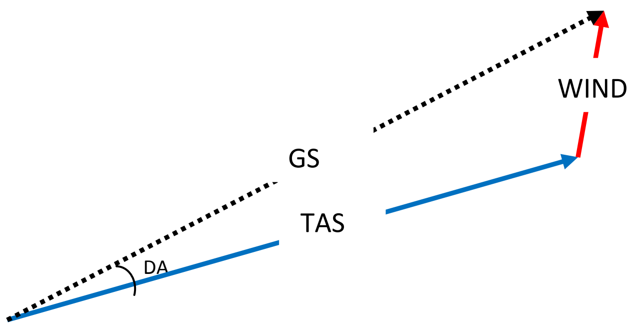


Figure 8 TAS to GS Conversion

The resultant GS vector has Ground Speed as the magnitude and course as the direction. Wind effect on ground speed is due to tail or head wind as shown in Figure 7.

During the execution of the flight plan drift angle ("DA" in Figure 8) is monitored continuously and heading is oriented accordingly so as to follow the course.

2.4 FMS Cruise Calculations

Cruise Types

Following cruise types are used in C-130 aircrafts, which are in general applicable for transport type aircrafts:

1. 970 TIT : Whatever the aircraft configurations and environmental attributes are, the calculations (especially speed) are valid for target 970 °C turbine inlet temperature(TIT). The purpose of 970 °C turbine inlet temperature is, gaining engines with a long life.
2. 1010 TIT : Whatever the aircraft configurations and environmental attributes are, the calculations (especially speed) are valid for target 1010 °C turbine inlet temperature(TIT). The purpose of 1010 °C turbine inlet temperature is gaining optimum performance from engines.
3. Long Range: The calculations which are aimed to realize the longest range. The purpose of this cruise type is to reveal the longest range of the aircraft with a given aircraft configuration and environmental attributes.
4. Fix TAS: The calculations (range, fuel flow etc.) are carried out for a given specific TAS. The advantage of a stable speed is, basically making a fair approximation for

the arrival time with a given distance or vice versa. When engine efficiency is taken at 95%, fix TAS range can be {220, 245, 260, 270, 280, 290, 300, 310} and for the engine efficiency of 100%, fix TAS range can be {220, 245, 260, 270, 280, 290, 300, 310, 320} in knots.

5. Maximum Endurance: The purpose of this calculation type is achieving the maximum endurance given the aircraft configuration and environmental attributes. Maximum Endurance cruise type may be used to plan a Search and Rescue (SAR) operation.
6. Optimum Endurance: Identical calculations are done as in the Maximum Endurance except computing the optimum pressure altitude. The optimum pressure altitude is the altitude which ensures the minimum fuel flow given the aircraft configuration and environmental attributes. The difference from Maximum Endurance is that Maximum Endurance calculations only perform the outputs(Speed, Fuel Flow, Endurance…) with a given pressure altitude and also other configurations. However, Optimum Endurance first calculates the optimum pressure altitude and other calculations are computed with this pressure altitude.

2.4.1 Altitude Calculations:

Altimetry Definitions

Altitude is a relative concept which needs a baseline, literally, to have a precise meaning. Figure 9[7] shows the altimetry terminology where an altimeter is the instrument for measuring the altitude in avionics by sensing the current pressure and converting the difference between the current pressure and the baseline pressure into feet.

[7] The figure is adapted from [4]

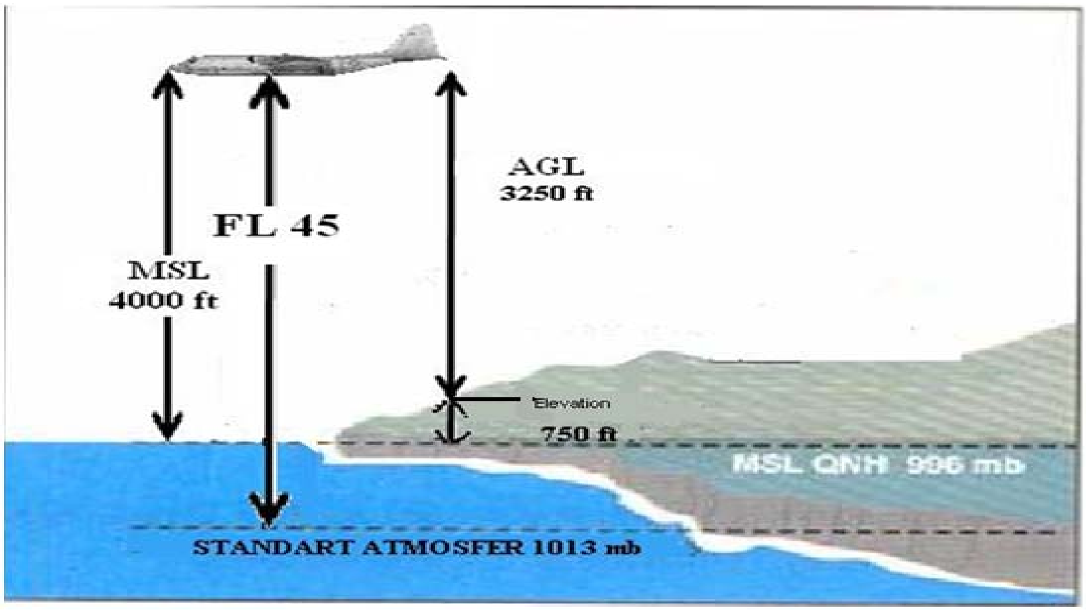


Figure 9 Altimetry Terminology

The terminology is defined as follows [4]:

- Altitude (MSL): Vertical distance above Mean Sea Level (MSL)
- Elevation: Vertical distance of a level or point measured from MSL.
- Flight Level (FL): Surface of constant atmospheric pressure measured from the 1,013.25 datum used for vertical separation by specified pressure intervals (usually 500 or 1,000 ft). Flight Level is measured in hundreds of feets, for example FL350 designates 35,000 ft.
- QNH: Airfield (or generally the ground) pressure converted to MSL.

Service Ceiling

The maximum altitude where aircraft is still capable to climb at a minimum ascend speed (100 feet per minute for C-130). Service ceiling determines the flight envelope of the aircraft[8].

[8] The lower end of flight envelope does not depend on the performance characteristics but on the operating procedures of the aircraft. C-130 aircraft as part of military operations performs low level flights down to the altitude of 500 feet AGL.

Cruise Ceiling

Cruise ceiling is the ideal altitude in terms of fuel efficiency for aircraft to fly enroute. Cruise altitude is calculated using performance model of the aircraft using the following inputs:

- Gross Weight (lb),
- Temperature Deviation (C°),
- TIT (C°),
- Engine efficiency (%),
- Bleed

Among these inputs only gross weight and temperature deviation are considered in the scope of this thesis considering that only these two are variable. All other parameters are assumed steady at their nominal value or configuration specified value.

It must be noted that cruise altitude does not have to be equal to cruise ceiling since pilot may overwrite suggested cruise ceiling and may choose another altitude for operational reasons.

Step Climb

As the aircraft flies and consumes fuel and looses its gross weight, current cruise ceiling is no longer the ideal and needs to be recalculated. The calculation of a new cruise ceiling and the point where to start climbing to the new cruise altitude is called *step climb calculations*.

It is not operationally convenient to continuously follow the cruise ceiling hence a *step size* is specified by the pilot. This enforces performing a step climb when:

cruise ceiling = previously accepted cruise ceiling + step size

2.4.2 Time and Fuel Calculations

Time and Fuel Calculations for Level Legs

The calculation steps of the leg time and the leg fuel for a level leg whose distance is 25 NM is as follows and illustrated in

Figure **10**:

1. Calculate True Air Speed (TAS) using performance model of the aircraft taking GW_{init} and Alt_{init} as inputs
2. Convert TAS to GS using current Wind Vector ($WindSpeed_{init}$ and $WindDirection_{init}$ gathered from forecasted meteorology data at $Position_{init}$, Alt_{init}, and ETA_{init})
3. Calculate leg time (in hours) = 25 NM / GS (in Knots)
4. Calculate Fuel Flow per second (lb per sec) using performance model of the aircraft using GW_{init} and Alt_{init} as inputs
5. Calculate leg fuel (lbs) = Fuel flow (lb/sec) * leg time (in hours) * 3600 (sec/hr)

Update;

$GW_{term} = GW_{init} –$ leg fuel

$ETA_{term} = ETA_{init} –$ leg time

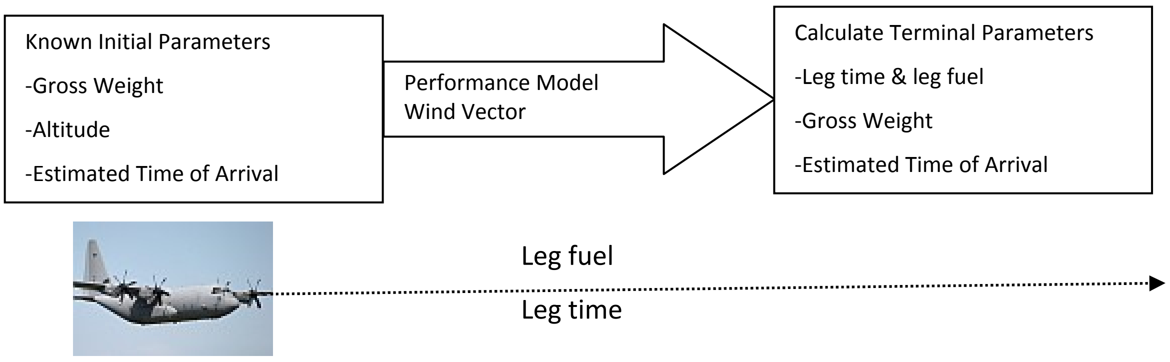


Figure 10 Time and Fuel Calculations for a Level Leg

For legs whose distances are longer than 25 NM the leg is subdivided into sublegs with 25 NM distance and the above process is iterated for each subleg. The benefit of this incremental solution compared to a one shot calculation is twofold both contributing to increasing the accuracy:

- To make perfect use of the meteorology data with a resolution of 20 NM,
- To reflect the effect of decreasing Gross Weight on the performance model with an acceptable accuracy.

<u>**Time and Fuel Calculations for Legs Including Climb/Descent**</u>

The calculation steps of the leg time and the leg fuel for a leg with an altitude change is similar for climb and descent cases, hence only climb case is detailed as follows:

1. Climb time, climb distance, and climb fuel between init to temp_point[9] (called subClimbLeg) is calculated using performance model of the aircraft given GW_{init} and Alt_{init} as inputs, where Figure 11 illustrates the calculation process.
2. $GW_{temp\_point} = GW_{init} –$ climb fuel
3. $ETA_{temp\_point} = ETA_{init} –$ climb time
4. The rest of the calculation between temp_point to term (called subLevelLeg) is the same as the calculations for a level leg case using temp_point's parameters as the initial values
5. The total leg time and leg fuel is the total for the corresponding values subClimbLeg and subLevelLeg's.

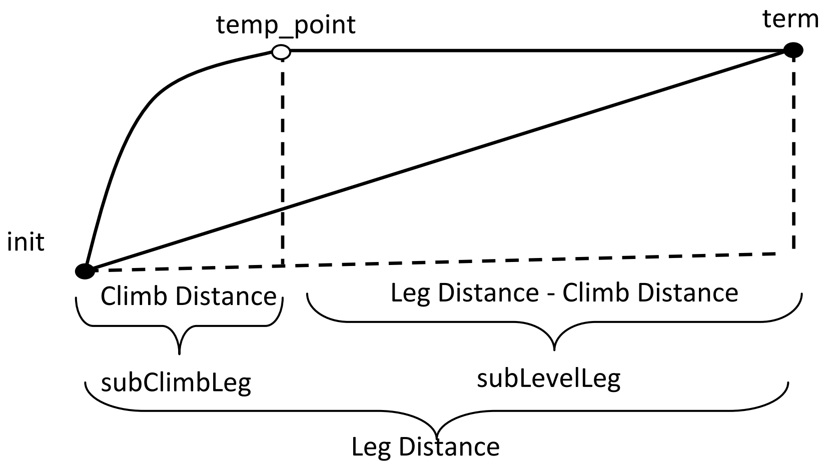


Figure 11 Time and Fuel Calculations for An Altitude Change Leg (Vertical View)

In scope of the thesis leg time and leg fuel of a 25 NM leg including climb is calculated as follows:

- Leg time of a level leg is increased by a factor of 24% per a 4000 feet climb

[9] temp_point's coordinatesis not known in advance. However this does not matter; knowing Alt_{temp_point} suffices to calculate time, distance and fuel values for the climb. Knowing the position init, the bearing between init to term and climb distance temp_point can be calculated.

- Leg fuel of a level leg is increased by a factor of 19% per a 4000 feet climb

In scope of the thesis leg time and leg fuel of a 25 NM leg including descent is calculated as follows:

- Leg time of a level leg is increased by a factor of 10% per a 4000 feet descent
- Leg fuel of a level leg is decreased by a factor of 19% per a 4000 feet descent

2.5 Route Planning Ontology

Concept and Definitions

Ontology is a Greek word meaning study/science (logy) of being (onto). It can be expressed as the study of being, existence and reality. It is a sub-branch of philosophy. Since ontology deals with existence of entities, their hierarchical properties and relations it has been a topic of interest for all science branches and has been used for a variety of engineering applications.

An ontology provides a formal method for identifying and classifying knowledge, while also providing a potential solution to the aforementioned problems and should not be confused with a taxonomy where entities are arranged in a way that only takes the generalization and specialization properties into account. The well known online encyclopedia Wikipedia for example categorizes and sub categorizes the topics in their database in a manner where finding information is easier. The result of this categorization is a taxonomy. In addition to these, an ontology also defines many relations between entities, restrictions and also the way these relations are to be used.

The main components that make up an ontology are:

- Language: A way to formally define ontology concepts. Must provide enough flexibility to allow high level definitions and low level restrictions.

  Example: OWL

- Concept (Class): Building block of the ontology. Every entity belongs to a class. Every property is also a class. Classes allow hierarchical organization.
  Example : NetworkModeller, AStarSolver

- Taxonomy: This is a hierarchical organization of classes. Although an ontology with a single class or multiple independent classes can be defined but most practical ontologies come with a detailed class hierarchy.
- Attribute (Feature): Low level descriptors defined on classes. These can be some primitive type defined on the base language or user defined types.
  Example: OPEN and CLOSED list attributes of AStarSolver class
- Property (Relation): The way classes are connected. These define interclass properties.
  Example: DynamicProgramming->*modelSolutionSpace*->NetworkModeller.
- Restriction (Constraint): Defined mostly on relations. Restricts the way the relation is used.
  Example: *Only* defined on *modelSolutionSpace* and *NetworkModeller* tells that the subject is suspect to using this relation only on NetworkModeller.
- Instance (Object): Realizations of classes. An ontology can act as a database with this ability. While a *RoutePlanner* is a class, "esenbogaToAtaturkFPL" is an instance of that class. Of course all relations and restrictions defined on the *RoutePlanner* class applies to "esenbogaToAtaturkFPL".

Route Planning Ontology

Liu [5] developed an ontology to achieve the domain knowledge reuse and efficient planning process in Route Planning (RP) for Aircraft, which is abbreviated as ONTRP. According to Liu ONTRP is developed to specify the domain knowledge and will fill the gap between conceptual modeling of RP and software implementation of it and he claims that ONTRP help people make an analysis and comparison between two RP algorithms by means of automatic reasoning and establish the models for RP efficiently. Figure 12 reproduced from [5] illustrates the realization of ontology concepts in ONTRP.

In the thesis by making use of ONTRP, a top down approach will be implemented from flight plan specific domain knowledge through the optimization techniques employed.

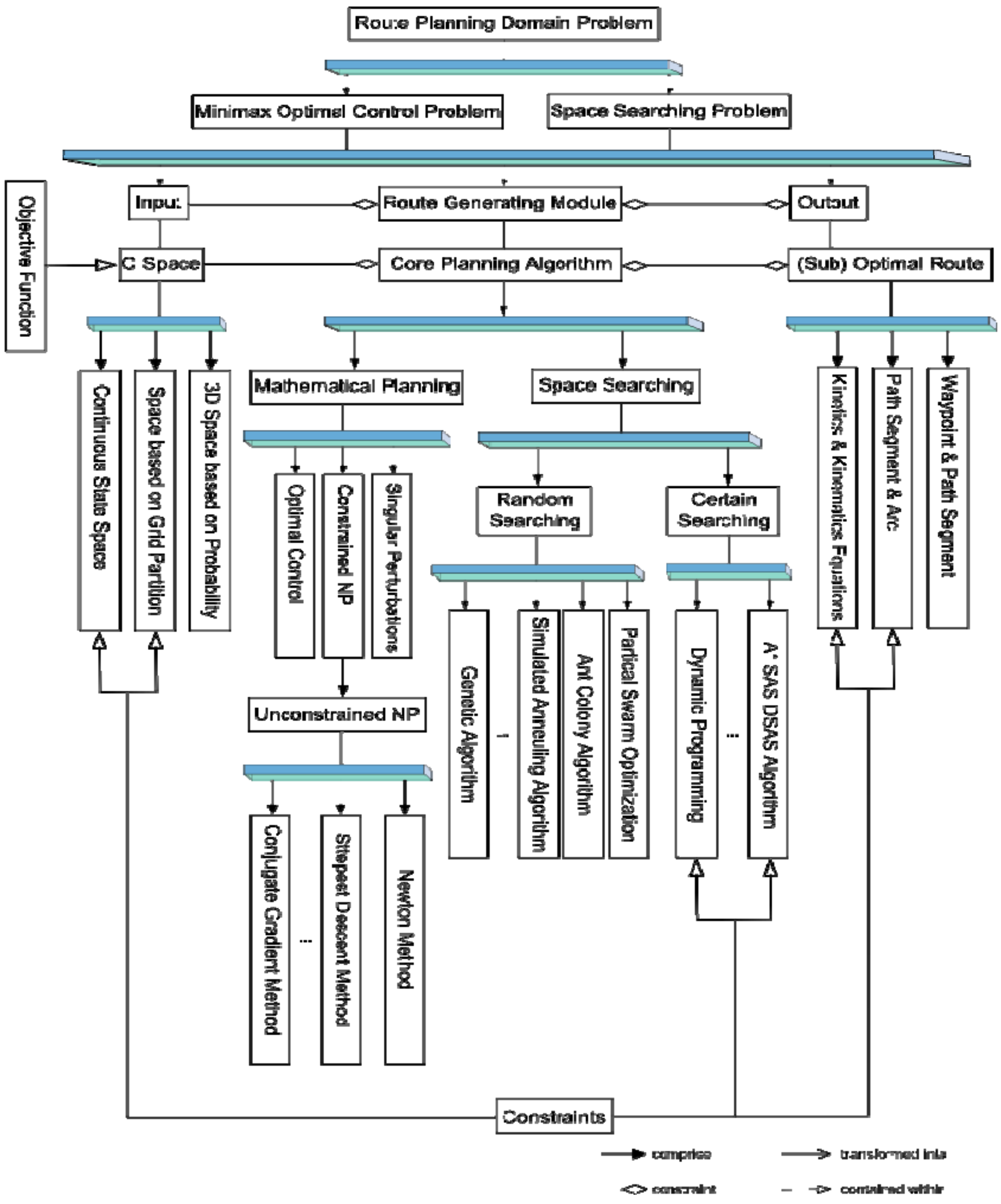


Figure 12 Concepts and Their Relations of ONTRP[10]

[10] The figure is reproduced from [5]

CHAPTER 3

LITERATURE SURVEY

The thesis work has started first surveying the automation in Flight Managements Systems and among the various alternatives automated route planning is selected as the thesis's scope so as to meet the current operational needs of C-130 aircraft. Hence, literature survey is sorted out in two subsections:

- Automation in Flight Management Systems,
- Route Planning

Automation in Flight Management Systems

Many researches have been performed on the automation of flight plans and specifically on the route planning problgem. On flight management systems the trend is in the collaborative management of each individual airplane's flight plans which will require standardization in the way calculations are performed by flight management systems, interoperable data exchange mechanisms, and collaborative algorithms for the conflict resolutions.

Commercial Aircraft FMS's aim to optimize flight paths in terms of both time and fuel required; and especially for "high" speeds these two factors are conflicting each other. Direct Operating Cost (DOC) combine these two conflicting factors, which is defined as the cost of the consumed fuel plus other costs that are proportional to flight time. Time proportional costs are represented by a pilot-selectable *Cost Index* which gives the weight of non-fuel costs to fuel costs. Pilots may change this index per flight or airliners adjust these values per line per season.

Sam P. Liden [6] got a a patent on computing wind-sensitive optimum altitude steps in a FMS. In his work he takes a flight plan which has been defined laterally and proposes a method for calculating the altitude change points and the new flight altitude level He proposes a dynamic approach to solve the problem where the objective function is to minimize Direct Operating Cost and his model also accounts for wind forecast error compensation.

FMS performs correct and accurate calculation of 4D flight path and then guide the pilot/airplane to track the route in specified accuracy limits in terms of lateral (i.e. Required Navigational Performance - RNP), vertical (Reduced Vertical Separation Minima - RVSM), and time (Required Time of Arrival - RTA) .

It is pointed out that, variations in FMS are mainly due to following reasons:

- There are many COTS (Commercial off the Shelf) FMS manufacturers,
- There is no standardization on FMS calculations,
- There is no standard operational procedures; even the same FMS system in the same airline may be used differently by different pilots

Wichman, Carlsson, Lindberg [7] performed flight trials to assess the RTA (Required Time of Arrival) algorithm's performance of FMS used by Scandinavian Airlines and identify necessary improvement for integration into next generation avionics. Excluding the variance in algorithmic behaviors by concentrating on a single FMS system they concluded that the deviation from plan is mainly due to forecasted wind model and how recent wind data is incorporated into the execution of the plan.

Air Traffic Controller in integration with FMSs performs a collaborative planning which is called NEXTGEN; the aim is, by making use of the exchanged 4D plans of individual airplanes, early detection of conflicting routes and issue directives to the aircrafts to change their plans in advance. This early detection will have huge cost savings compared to the last minute savings. On the other hand, due to inaccuracies in the plans, there is a high chance of issuing false alarms. The standardization of FMSs hence plays an important role.

Theunissen, Rademaker, Bleeker, Wichman [8], suggests a network-centric framework for transitioning from "user preferred trajectory" to "business trajectory oriented operation". The new process aims to assure separation and timing.

Sherry [9] makes economic assessment of investment to NEXTGEN and she claims that cost benefit of NEXTGEN is being affected by an aggregate model of the operation including passenger itineraries and itinerary disruptions. She gives the statistical data that a 7-10% increase in load-factor can nullify the reduction in total passenger trip delay gained by a 5% improvement in on-time performance achieved by NEXTGEN.

Route Planning

Route Planning problems can be categorized with respect to several characteristics;

- Solution space:
    - Either vectoral or
    - discrete
        - Road networks,
        - Grid networks,
        - Or combination of both
- Objective Function: Single objective or multi objective function
- Knowledge of the environment data
    - Certainty or uncertainty of the knowledge
    - Static or dynamic data
- Cost function: Uniform or weighted cost functions per grid
- Output: Optimal or (sub) optimal
- Algorithm: A*, Dijkstra, Hierarchical approaches etc.

Liu's [5] route planning ontology categorizes the route planning problem domain (refer to Figure 12) with respect to various aspects.
Independent of in which category the route planning is classified, all suffer from the curse of dimensionality problem. And hence, more research needs to focus on this issue.
Brener and Subrahmanian [10] introduces a new approach for aggregating the grids with traversability characteristics. They claim that this approach supports both grid and road networks at the same time.
Most of the research takes Dijkstra and/or A* algorithms as the base, and tries to make improvements. Shi, Cao, S. Zhu, B. Zhu [11] made improvement on the lower bound when the cost is based on the Euclidean distance and angle between the expanded and end node. [12] made improvement by pruning neighbor grids that are infeasible to fly due to

kinematic constraints of the aircraft (minimum route leg length, maximum turning angles, route distance constraint) and called it as Sparse A*.

Araki and Suzuki [13] introduced a new approach in handling the uncertainty in the environmental factors. In their view environmental data such as weather forecast data contains uncertainty and care should be given when those data is used as a point value. They apply data abstraction and fuzzy reasoning for abstracting the quantitative values into qualitative ones. Then by means of a translation object a quantitave value is mapped into discrete qualitative value(s) with expected probability. An interpretation object evaluates this mapping including the expectations and reassigns a numeric mobility cost index.

CHAPTER 4

PROBLEM DEFINITION AND MODELING

In this chapter the formal definition of the problem is declared and the assumptions are given.

4.1 Modeling the Network and Cost Structure

Stefanakis and Kavouras [14] states that the approach of determination of the optimum flight path in space consists of four steps:

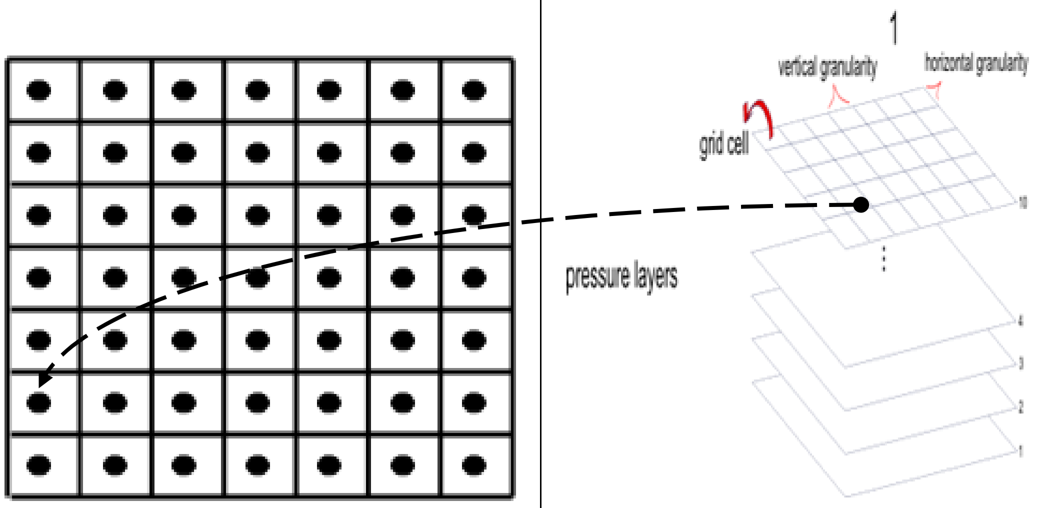
1. Determination of a finite number of spots in space.
2. Establishment of a network connecting these spots.
3. Formation of the travel cost model.
4. Determination of the optimum path(s) from the spot of reference (i.e., the departure or destination spot)

This reveals the basics of a graph based modeling with spots acting on the nodes and the connections as the arcs.

In this section the network structure studied in the thesis will be introduced.

Nodes for spots in space

3 dimensional gridded raster approach will be applied such that the centers of the grids compose the vertices of the underlying network as shown in Figure 13.



a) Grid Structure b) 3 Dimensionality

Figure 13 Gridded Network Structure

The following is the details of the grid space:

- Vertical (latitudinal) and horizontal (longitudinal) grid granularity is 25 NM
- Since latitude and longitude intervals are not of equal length the grid size increments are not uniform in terms of geographical coordinate system
- Pressure altitude granularity is FL2
- Each grid is spatially referenced by its lower left corner in geographical coordinate system (i.e. in latitudes, longitudes) and its altitude

Establishment of a network connecting these spots

The network is constructed by adapting *indirect* neighborhood scheme as shown in Figure 14[11].

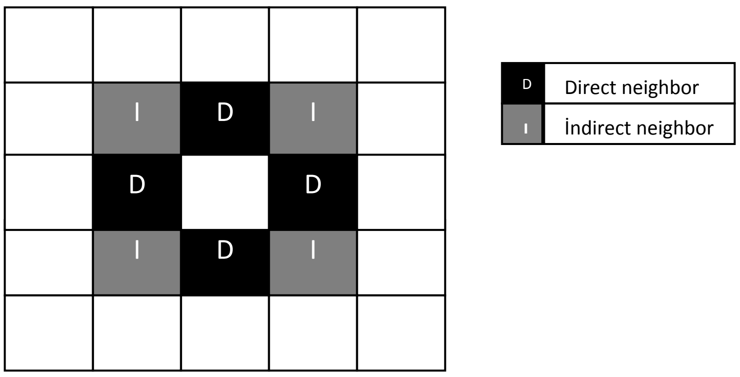


Figure 14 Types of Neighbor Cells

Considering that our network is 3 dimensional, for a given grid there are 24 possible movements where;

- 8 of them are at the same pressure altitude level,
- 8 of them are at the one above pressure altitude level,
- 8 of them are at the one below pressure altitude level,

It is assumed that the aircraft is not a helicopter and does not have the ability of climbing or descending without lateral movement.

Formation of the Travel Cost Model

Travel cost is, denoted as *arcCost*, expresses either time or fuel spent depending on the objective function.

The travel cost model is *dynamic* such that the weight values (time or fuel required for each move) assigned to volume of space under study change over time. The changing

[11] Direct neighbors of a cell are those cells with shared edges, whereas indirect neighbors are those with common vertices

meteorological conditions and the changing gross weight are the reason for the dynamic nature of the travel cost model as explained in section 2.4.2.

4.2 Assumptions in Modeling

1. 25 NM leg distance is sufficient for 90° turns
2. A given Cruise Type is true all the way
3. For a given GW_{init} and Alt_{init} the performance model suggests TAS and Fuel Flow as a couple (i.e. the performance model does not support suggesting the effect of changing TAS on fuel flow and vice versa for a given GW_{init} and Alt_{init})
4. Holding around to wait for improving weather conditions is not allowed
5. Only Track to Fix (TF[12]) legs are allowed.
6. The aircraft is not allowed to make a climb/descent unless it makes a level flight for at least 100 NM.
7. Terrain clearance is not of concern

These assumptions are in accordance with the operational policies and do not have adverse effect on the prospective application of the path solution for the following reasons respectively:

1. Although 25 NM leg distance is normally a very short distance for a cruise leg[13] and 90° turn is very steep[14] the reason is just to establish a baseline configuration to experiment the affect of discretization on the optimal route planning process. In addition to that, "*Path smoothing*" technique is to be applied after optimum path is found to reduce the effect of short leg distances and angular discretization error.
2. Every mission has its own preferred/default cruise type (i.e. for low level flight – less than 5000 ft- and SAR missions maximum endurance type, for VFR flights Fix

[12] TF legs are legs whose starting and ending fix coordinates are known

[13] Average leg distance for cruise legs is 56.36 NM as averaged out from 130107 airway legs defined in DAFIF digital military navigational database, where the minimum leg distance is 1, and maximum leg distance is 918 NM.

[14] Average course change for cruise legs is 6.57 degree as averaged out from 130107 airway legs defined in DAFIF digital military navigational database, where the minimum course change is 0, and maximum course change is 175 degree.

TAS type etc.) and it is always a Pilot's operational requirement to have a single Cruise Type in a Flight Plan.

3. If the pilot prefer not to follow performance model suggested TAS he can choose Fix TAS.
4. Considering that weather forecast data has 3 hours of temporal resolution, abrupt changes in weather conditions are not expected.
5. Cruise phase only includes TF, Hold and PTurn legs where Hold and PTurn legs are used for course reversal. It should also be noted that this assumption is in accordance with the previous assumption.
6. This operational practice is due to meet *vertical separation minima*[15] and to provide an operationally flyable flight path in terms of pilot's convenience.
7. As explained in section 2.1 since cruise phase altitude interval is [20000, 32000] ft ignorance of terrain clearance does not have severe an effect except mountainous terrain.

4.3 Dynamic Programming Model Definition

The problem under study meets the Bellman's *principle of optimality*[16] [15] due to the additive nature of the objective function. In addition, a flight path can be decomposed into stages such that past history of the system, i.e. how we got to the current stage and state, are of no importance and the total cost depends only on the states at the traversed stages. Hence dynamic programming can be applied to the problem evaluated in this thesis.

In the remaining of this section dynamic programming's methodology is used to formally define the problem.

Notation:

Stage: Each move from a grid to the next grid.

x: Longitude of the lower left corner of the grid

[15] Vertical seperation minima is the standard vertical separation required between aircraft flying above specified flight levels.

[16] *An optimal policy has the property that whatever the initial state and the initial decision are, the remaning decisions must constitute an optimal policy with respect to the state which results from the initial decision.* Or, using a simpler and intiuitive definition, *every optimal policy consists only of optimal subpolicies.*

y: Latitude of the lower left corner of the grid

z: Pressure altitude of the grid (ft), z ε [FL16, FL32] and z = k * 2FL where k is integer

μ: # of stages remaining that must be waited before the aircraft can climb/descent, μ ε {0, 1, 2, 3}

GW_{init}: Gross weight of the aircraft (lb) at initial state

ETA_{init}: Expected Time of Arrival at initial state

State Variables

λ_S ε Λ: denotes the state vector at stage s, $\lambda_s = (x, y, z, \mu)^T$ where Λ is the state space.

Initial state is given as $\lambda_0 = (x_{st}, y_{st}, z_{st}, 3)^T$ meaning that the aircraft is not allowed to climb/descent for three stages (i.e. it will make a level flight throughout the first 4 movement)

Final state is given as $\lambda_{n=}$ $(x_{end}, y_{end}, z_{end}, \mu_{end})$ where;

- n ε {1, 2, ..} meaning that the number of stages is not fixed/preknown, n is upper limited by operational constraints such as the fuel availability etc.
- x_{end}, y_{end}, z_{end} are given as input, they represent the coordinates and altitude of TOD point
- μ_{end} ε {0} imposing the operational constraint that after reaching TOD point aircraft will switch from Cruise phase to Descent phase and starts descending.

Decision Variables

$x_S(\lambda_S) = \{\Delta x_S, \Delta y_S, \Delta z_S\}$ meaning that the movement in the lateral&vertical position such that

$x_s+\Delta x_s \rightarrow x_{s+1}$

$y_s+\Delta y_s \rightarrow y_{s+1}$

$z_s+\Delta z_s \rightarrow z_{s+1}$ for all s = 0, 1, 2, ..

$x_s(\lambda_s)$ ε X_{λ_s} represents one of the choices that is available when the system is in state λ_s. The decision set X_{λ_s} consists of the set of all possible choices when the system is in state λ_s [16].

Cost Functions:

Let $C_s(\lambda_s, \lambda_{s+1}) = C_s(x_s(\lambda_s))$ denote the cost of transformation from λ_s to λ_{s+1}

Depending on the formulation of the objective function (i.e. whether to minimize time or minimize fuel flow) cost function $C_s(x_s(\lambda_s))$ is either $Time_s(x_s(\lambda_s))$ or

$FC_s(x_s(\lambda_s))$. Calculation details of $Time(x_s(\lambda_s))$ and $FC(x_s(\lambda_s))$ are described in section 2.4.2 Time and Fuel Calculations.

State Transformation:

Let T_s ($\lambda_{s,}$ $x_s(\lambda_s)$) denote the state transformation

- $\lambda_1 = T_1$ ($\lambda_{0,}$ $x_0(\lambda_0)$) initially. The decision $x_0(\lambda_0)$ already updates 3 of the 4 state variables by the definition:
  - $\lambda_{1.}x$[17] $= \lambda_{0.}x + \Delta x_0$
  - $\lambda_{1.}y = \lambda_{0.}y + \Delta y_0$
  - $\lambda_{1.}z = \lambda_{0.}z$ ($\lambda_{0.}$ μ being equal to 3 causes Δz to be equal to 0)
  - $\lambda_{1.}$ $\mu = 2$
- $\lambda_{s+1} = T_s$ ($\lambda_{s,}$ $x_s(\lambda_s)$) in general:
  - $\lambda_{s+1.}x = \lambda_{s.}x + \Delta x_s$
  - $\lambda_{s+1.}y = \lambda_{s.}y + \Delta y_s$
  - $\lambda_{s+1.}z = \lambda_{s.}z + \Delta z_s$, $\Delta z_s = 0$ if $\lambda_{s.}\mu > 0$
  - $\lambda_{s+1.}\mu = \begin{cases} \max(\lambda_s.\mu - 1, 0) \text{ if } \Delta z_s = 0 \\ 3 \text{ otherwise} \end{cases}$

Objective Function

$g_s(\lambda_s) = \min_x \left(\sum_{t=1}^{s} C_t(x_t(\lambda_t))\right)$

The forward recurrence relationship is as follows;

$g_0(\lambda_0) = 0,$

$g_s(\lambda_s) = \min_{\lambda_{s-1}} {}_{\{\lambda_s \, \varepsilon \, X\}} \left(g_{s-1}(\lambda_{s-1}) + C_{s-1}(\lambda_{s-1}, \lambda_s)\right)$

Since both the initial state λ_0 and the final state λ_n are given it is possible to use either forward or backward recursion to solve the problem. Although they are not a component of the state vector, gross weight and time will have to be treated as data that will be carried from stage to stage due to the dynamic nature of the *arcCost* $C_s(\lambda_s, \lambda_{s+1})$ as defined in section 4.1. The dynamic nature of the *arcCost* combined with the factor that GW_{init} and ETA_{init} are known in advance is the reason why forward recursion is chosen.

[17] $X_{.<a>}$ notation denotes the component <*a*> of the state vector X

In a similar manner the use of backward recursion requires knowing GW_{end} and ETA_{end}. The exact calculation of GW_{end} and ETA_{end} requires the optimum path to be solved causing a cyclic dependency. To break up the dependency the following approach, which is either computationally expensive or inaccurate depending on the *threshold value*, would be applied:

1. Guess an initial estimate of $GW_{end\_estimate}$ and $ETA_{end\_estimate}$
2. Calculate the optimum path backward and get $GW_{init\_estimate}$ and $ETA_{init\_estimate}$ given the optimum path.
3. Compare ($GW_{init\_estimate}$,$ETA_{init\_estimate}$) and (GW_{init},ETA_{init}) if they are within a predefined *threshold value* STOP otherwise update the $GW_{end\_estimate}$ and $ETA_{end\_estimate}$ and loop back.

4.4 Dynamic Programming Algorithm

Dynamic Programming (DP) provides a schema guaranteeing optimal solutions; in the route planning problem context the dynamic programming model defined in section 4.3 is directly formulated into the Dijkstra's famous shortest path algorithm [17].

Dijkstra's shortest path algorithm, in addition to finding an optimum route from *initial* to *final*, it also finds the optimum route starting from any traversed state *s* to the goal.

What Dijkstra's algorithm suffers from is the same for the Dynamic Programming in general; *the curse of dimensionality*. The solution space grows exponentially as the dimension of the state vector and range of state vector elements increases.

In the scope of the thesis to cope with the mentioned problem, two algorithms will be applied in hybrid, which are A* and Coarse to Fine Dynamic Programming (CFDP) of which will be introduced in the subsections below.

4.4.1 A* Algorithm

The following description of A* algorithm is indebted in general outline to the description in Beeker [18] by adapting the notation defined in section 4.3.

The cost of the minimum cost path from state λ_i to state λ_j is represented by $g(\lambda_i, \lambda_j)$. If there is no path from λ_i to λ_j, then $g(\lambda_i, \lambda_j)$ is undefined.

The cost of the minimum cost path from the initial state (λ_0) to final state (λ_{end}) through an arbitrary state λ_i is represented by $f^*(\lambda_i)$. $f^*(i)$ is equal to $g_i(\lambda_i)$[18], the cost of the minimum cost path from the source to λ_i, plus $g(\lambda_i, \lambda_{end})$, the cost of the minimum cost path from λ_i to λ_{end}.

Let $h^*(\lambda_i) = g(\lambda_i, \lambda_{end})$ represent the minimum cost for reaching λ_{end} from λ_i. Then the minimum cost for reaching the λ_{end} with a path through λ_i is $f^*(\lambda_i) = g^*(\lambda_i) + h^*(\lambda_i)$.

A* algorithm keeps a sorted list of states to be expanded. The states are sorted by the total estimated cost of a path from λ_0 to λ_{end} through that state. A* algorithm picks the state that has the lowest estimated total cost to be expanded.

Let $f(\lambda_i)$ be the estimated total cost for reaching λ_{end} from λ_0 through λ_i. It should be noted that $f(\lambda_i)$ is not necessarily equal to $f^*(\lambda_i)$. The estimate does not have to equal the actual cost. Let $g(\lambda_i)$ be the cost of the current path found from λ_0 to λ_i. Because there may be more than one way to reach a state, it is not certain that the actual lowest cost has been found—i.e., $g(\lambda_i)$ does not necessarily equal $g_i(\lambda_i)$. Rather, $g(\lambda_i)$ is the lowest cost found at this point in executing the algorithm. Let $h(\lambda_i)$ be the estimate of how much it will cost to reach λ_{end} from node λ_i. Then $f(\lambda_i) = g(\lambda_i) + h(\lambda_i)$. A* algorithm will always take the node with the lowest $f(\lambda_i)$ from the list to expand.

To expand λ_i, each λ_{next} in the result set of $T(\lambda_i, x(\lambda_i))$, that can be immediately reached from λ_i is examined. For each successor λ_{next}, $g(\lambda_{next}) = g(\lambda_i) + C_i(x_i(\lambda_i))$. In other words, the cost for reaching λ_{next} is equal to the cost for reaching λ_i plus arc cost of λ_i to λ_{next}. Since, it can not be guaranteed that a state will be reached by a lower cost path later; states may be placed on the list to be expanded more than once.

The A* algorithm keeps two lists: OPEN, the list of states that have been reached so far and have yet to be expanded (equivalent to the list in the Dijkstra Algorithm (6)); and CLOSED, the list of states that already have been removed from the OPEN list and expanded. The CLOSED list is necessary in case a state is encountered a second time. If a state is reached with a lower cost then before, it must be reconsidered. The CLOSED list will store the already expanded states and the cost previously calculated for reaching them for comparison purposes.

[18] $g_i(\lambda_i) = g(\lambda_0, \lambda_j)$

A*algortihm is a state pruning algorithm in that is does not perform a thorough search over the state space but apply a form of best first search guided by the applied heuristic function. What prevents A* algorithm to be regarded as a *heuristic algorithm* and guarantees to find the optimum solution is that the heuristic function $h(\lambda_i)$ satisfies the *admissibility* property.

Admissibility property of the heuristic function means that the heuristic function never overestimates the actual minimal cost of reaching λ_{end}. If the heuristic function is also *monotonic*[19] there would be no need to use the CLOSED list. Monotonicity property guarantees that if a state is expanded before the successive visits always have a higher cost [18].

The details of the application of A* algorithm to the problem studied in this thesis will be detailed in section 5.2.

4.4.2 Coarse to Fine Dynamic Programming (CFDP) Algorithm

The following description of CFDP algorithm is indebted in general to the description in Raphael [20].

CFDP technique requires the network structure to be modeled as trellis graph. By trellis, it is meant a graph in which each node has an associated level and arcs can only connect nodes at adjacent levels.

The essential idea of CFDP algorithm is to form a series of coarse approximations to the original trellis network by aggregating trellis states into "superstates." For each coarse approximation, the optimal path is found using DP with optimistic arc costs between the superstates. The superstates along this optimal path are refined and the process is iterated until optimal path contains no "superstates".

Consider the trellis diagram in the upper-left panel of Figure 15 in which minimum cost path from the left-most "state" to the rightmost "state" is calculated. A sequence of approximations to this problem is solved which is guaranteed to result in the optimal path. In the first approximation the states, at each trellis level, are partitioned into a small

[19] A heuristic is monotonic if $h(\lambda_{i-1}) \leq C_{i-1}(\lambda_{i-1}, \lambda_i) + h(\lambda_i)$ for all I = 1, 2, ..
In other words, monotonicity implies that it is impossible to decrease (total cost so far + estimated remaining cost) by extending a path to include a neighboring state. Monotonicity also implies admissibility when $h(\lambda_{end}) = 0$.

collection of subsets or superstates. An “optimistic” graph on the superstates is to be defined using the following two rules:

- Two superstates are connected if any of the pairs of states in their cross product are connected;
- The cost of an arc between two superstates is the minimal cost over all arcs connecting the superstates.

The optimistic graph obtained using these rules are shown in the upper-left panel of Figure 15. The optimal path through this graph will be found using DP; this path is shown in red lines. The second approximation is formed by refining the superstates along the optimal path as shown in the upper-right panel of Figure 15. The graph is then redrawn, using the same two rules as before and the optimal path is found through this somewhat less optimistic graph, again using DP. The process of finding the optimal path and refining the superstates along the optimal path continues until an optimal path composed entirely of singleton superstates is found as in the bottom-right panel of Figure 15.

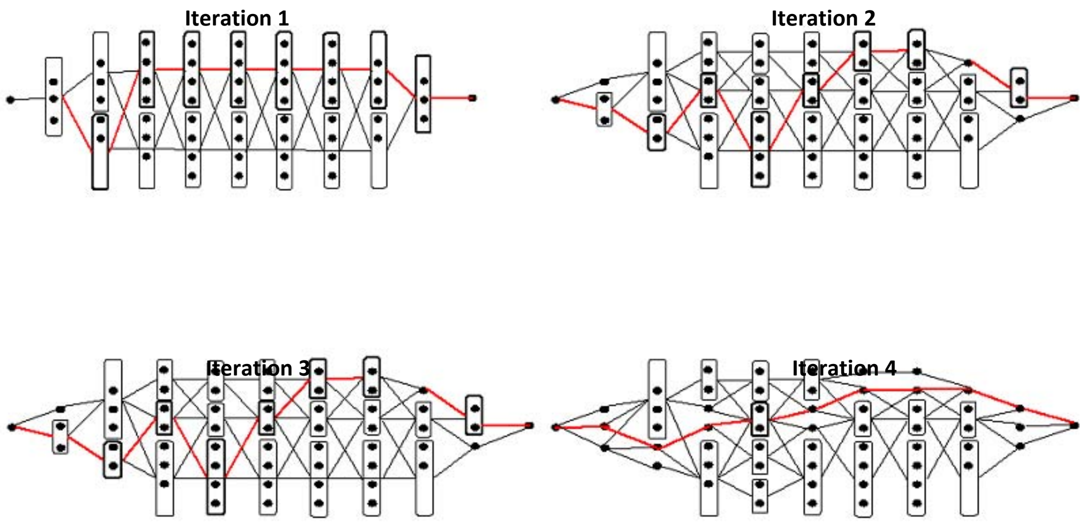


Figure 15 CFDP Iterative Solution Process

The details of the application of CFDP to the current thesis’s scope will be detailed in section 5.1.

CHAPTER 5

SOLUTION

The modeled problem can be solved using two alternative techniques:

1. Apply CFDP and calculate the best found solution (BFS) or alternatively
2. Apply A* algorithm using BFS solution as an upper bound

5.1 Calculation Applying CFDP

CFDP technique requires the network structure to be modeled as trellis graph. It should be noted that by restricting the network structure to be modeled as trellis graph, as shown in Figure 16, optimality is deviated since the movement is restricted in only three lateral directions where in the original network modeling the movement is modeled in 8 lateral directions. Since CFDP is used to establish an upper bound for the original problem optimality is not of a main concern. CFDP algorithm application for upper bound calculation whose details will be given in this section is denoted as CFDP$_{UB}$.

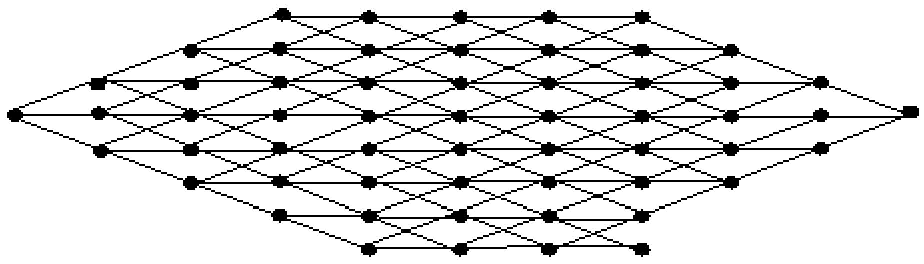


Figure 16 Original Trellis Graph G = (λ, ξ, C)

Raphael formally defines the CFDP algorithm as [20]:

> *$G = (S, \xi, C)$ be a weighted trellis graph where S is a set of states, ξ is a set of edges, and C is a cost function defined on the edges. Where, every state $s \in S$ has a "stage" $v(s) \in \{0,\ldots,N)$ and $\xi \subseteq \{(s, t): s, t \in S, v(s) + 1 = v(t)\}$. The cost of an edge $(s, t) \in \xi$ is given by $C(s, t)$.*
>
> *$S0$ and SN are the unique states having $v(S0) = 0$ and $v(SN) = N$. We define the collection of paths through G as $\pi(G) = \{(S0, S1,.., SN): (Sn, Sn+1) \in \xi, n = 0, 1,\ldots, N-1\}$. The cost of a path is then given by $C((S0, S1,.., SN) = \sum_{n=0}^{N-1} C(Sn, Sn+1)$*
>
> *Initialization:*
>
> *Let P_n be a partition of $\{s \in S: v(s) = n\}$.*
>
> *Let $G^0 = (S^0, \xi^0, C^0)$ where*
>
> *$S^0 = \bigcup_{n=0}^{N} P_n$*
>
> *$\xi^0 = \{(S, T): S, T \in S^0$ and $\exists\ (s, t) \in \xi, s \in S, t \in T\}$*
>
> *$C^0\ (S, T) = \min_{\{(s, t):\, s \in S,\, t \in T,\, (s, t) \in \xi\}} C(s, t)$*
>
> *$(S_0^{\,0}, \ldots, S_N^{\,0}) = \arg\min_{\{(S0, \ldots, SN) \in \pi(G0)\}} C^0\ (S_0,\ldots, S_N)$*
>
> *Iteration: For $k = 1, 2,\ldots,n$ define $G^k = (S^k, \xi^k, C^k)$ where*
>
> *$S^k = \{S^{k-1} \cup\ (\bigcup_{n=0}^{N}$ Partition$(S_n^{\,k-1}))\} \setminus (S_0^{\,k-1}, \ldots, S_N^{\,k-1})$*
>
> *$\xi^k = \{(S, T): S, T \in S^k$ and $\exists\ (s, t) \in \xi, s \in S, t \in T\}$*
>
> *$C^k\ (S, T) = \min_{\{(s, t):\, s \in S,\, t \in T,\, (s, t) \in \xi\}} C(s, t)$*
>
> *$(S_0^{\,k}, \ldots, S_N^{\,k}) = \arg\min_{\{(S0, \ldots, SN) \in \pi(Gk)\}} C^k\ (S_0,\ldots, S_N)$*

CFDP algorithm provides a "*template*" which needs to be concreted by the specific modeling requirements, namely:

- The original trellis graph $G = (S, \xi, C)$ is defined such that
  - $s \subseteq \lambda_S$ *for* every state $s \in S$ and $\lambda_S \in \Lambda$ where Λ is the state vector of the original problem as defined in section 4.3

    - $S \subseteq \Lambda$, meaning that the state space of the CFDP problem is obtained by pruning that of the original problem. Search space is constructed along the λ_0- λ_{end} axis with a 200 NM buffer area as shown in Figure 17
    - $s_0 = \lambda_0$ and $s_N = \lambda_{end}$

- k $\in$ [1, n] where $| S_i^k |$[20] = 1 for i $\in$ [1, N], or equivalently iteration will end when $(S_0^k, \ldots, S_N^k) \in \Lambda$.
- S^0 is a super state such that $S \in S^0$ denotes the state vector S = $(x^0, y^0, z^0, \mu^0)^T$ where
    - $|x^0| = |x|$, where $|x|$ is defined as in paragraph 4.3
    - $|y^0| = 4 * |y|$, where $|y|$ is defined as in paragraph 4.3
    - $|z^0|$ = FL16, and z ε [FL16, FL32]
    - $\mu\ \varepsilon\ \{0\}$

    Which means that $S \in S^0$ does not contain z^0, μ^0 as state vector components.

- As described in section 4.4.2 CFDP aggregates states into "superstates". To formally define the "aggregation level" of each state let define state level "ℓ" as a property of each $S \in S^k$ for k = 1, 2,…,n as given in Table 4:

Table 4 State Dimensions per State Level

State Level	x granularity	y granularity	z granularity	Range of S. μ	Dimension
3	25	4 * 25	FL16[21]	{0}	128
2	25	2 * 25	FL16	{0}	64
1	25	25	FL2	{0, 1, 2, 3}	1

[20] Let $|a|$ denote the discretized range or equivalently grid granularity of a where a is a component of a state vector.

[21] Super state consists of 8 pressure altitude layers where each layer's granularity is 2 FL.

Let define *Partitioning* function as taking an input state (S_{input}) and outputting *m* state(s) as follows:

- If $S_{input}.\ell = 1$ then
  - $\ell_{output} = 1$,
  - m = 1 (partitioning has no effect)
- If $S_{input}.\ell > 1$ then
  - $S_{output}.\ell = S_{input}.\ell - 1$, and
  - m is the result of the division of the corresponding values in Dimension column in Table 4 for row $S_{input}.\ell$ by row $S_{output}.\ell$ (i.e. partitioning a level 3 state produces 128 / 64 = 2 level 2 states and so on)
  - Table 4.Row(ℓ_{input}).Column("Dimension") / Table 4.Row(ℓ_{input}-1).Column("Dimension")

- Note that as specified in (9) "a *lower bound $C^{\sim k}$ (S, T), for the true minimum arc cost, C^k (S, T), between two super states without affecting the correctness of the algorithm, as long as $C^{\sim k}$ (S, T) = C^k (S, T) when |S| = |T| = 1* "
- Calculation of *"C^k (S, T) = $min_{\{(s, t): s \in S, t \in T, (s, t) \in \xi\}}$ C(s, t)"* requires *min* (|S|, |T|) * 9 calculations[22], which is computationally expensive. Raphael specified in (9) that; "One can substitute a *lower bound $C^{\sim k}$ (S, T), for the true minimum arc cost, C^k (S, T), between two super states without affecting the correctness of the algorithm, as long as $C^{\sim k}$ (S, T) = C^k (S, T) when |S| = |T| = 1* "
  Hence, *C^k (S, T)* can also be substituted by *$C^{\sim k}$ (S, T)* which is calculated as below[23];
  - Take *distance* as *$min_{\{(s, t): s \in S, t \in T, (s, t) \in \xi\}}$ GreatCircleDistance(s, t)*
  - Take *TAS* as max $_{\{altitude \in T.z\}}$ (PerfTAS(*altitude*))
  - Take *GS* as *$max_{\{(s, t): s \in S, t \in T, (s, t) \in \xi\}}$ TAS2GS(TAS, Course(s, t), MET(s))*

[22] As shown in Figure 16 for a given state there are 9 possible movements; of which 3 of them are at the same pressure altitude level, the 3 of them are at one above pressure altitude level, and 3 of them are at one below pressure altitude level

[23] *$C^{\sim k}$ (S, T)*calculation is detailed for the case when *arcCost* is time based, the calculation process for the case when *arcCost* is fuel based is near the same except the following additional step and the modification;

Take *FuelFlowperSecond* as min $_{\{altitude \in T.z\}}$ (PerfFFlow(*altitude*))
Take *TAS* as PerfTAS(*altitude*)

  - $C^{\sim k}$ *(S, T) = distance / GS*

- The optimal path at each iteration will be calculated applying A* algorithm

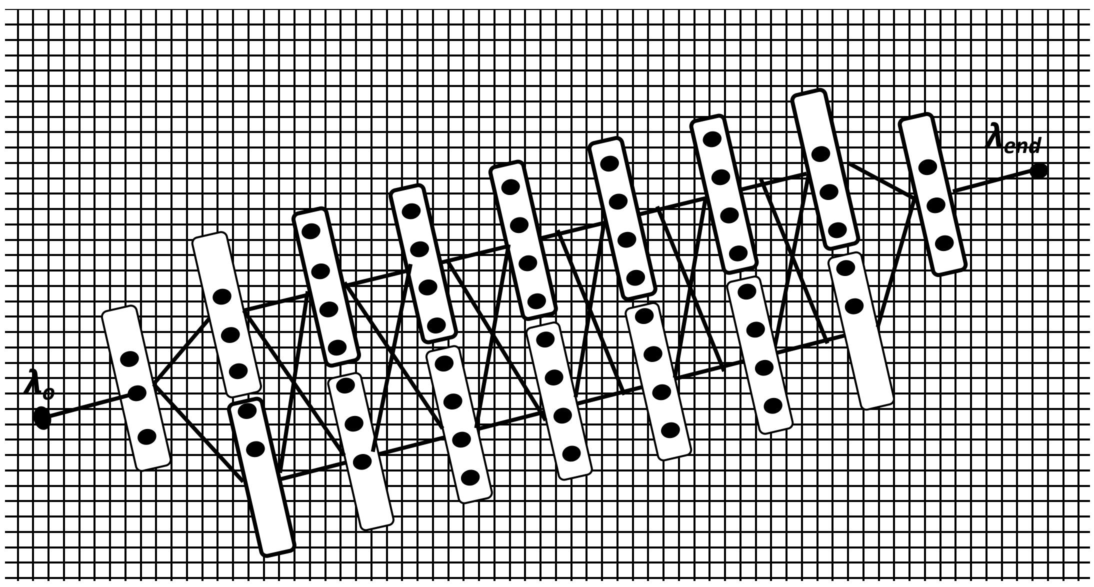


Figure 17 Search Space Pruning and Orientation

5.2 A* Algorithm Pseudocode

```
AStarSearch( $\lambda_0$, $\lambda_{end}$)
{
    clear Open and Closed

    g($\lambda_0$) = 0          //cost from start
    h($\lambda_0$) = PathCostEstimate($\lambda_0$, $\lambda_{end}$)          //cost to goal
    f($\lambda_0$) = h($\lambda_0$)     //total cost
    Parent($\lambda_0$) = null

    insert $\lambda_0$ into Open
    while Open is not empty
    {
        pop $\lambda_{current}$ from Open // $\lambda_{current}$ is the state having lowest TotalCost
        // if at goal, we are done
        if ($\lambda_{current}$ is $\lambda_{end}$)
        {
            construct a path backward from $\lambda_{current}$ to $\lambda_0$
            return success
        }
        else
        {
            for each $\lambda_{next}$ in the result set of T ($\lambda_{current}$, x($\lambda_{current}$))
            {
                newCost = g($\lambda_{current}$)+ arcCost($\lambda_{current}$, $\lambda_{next}$)
                // ignore this node if it exists and no improvement

                if ($\lambda_{next}$ is in Open or Closed) and (g($\lambda_{next}$) <= NewCost)
                {
                    continue
                }
                // store the new or improved information
                else
                {
                Parent($\lambda_{next}$) = $\lambda_{current}$
                    g($\lambda_{next}$) = newCost
                    h($\lambda_{next}$) = LowerBoundEstimate($\lambda_{next}$, $\lambda_{end}$)
                    f($\lambda_{next}$) = g($\lambda_{next}$) + h($\lambda_{next}$)

                    if ($\lambda_{next}$ is in Closed)
                    {
                        remove $\lambda_{next}$ from Closed
                    }
                    if ($\lambda_{next}$ is in Open)
                    {
                        //states in Open are sorted wrt ascending f
                    values
                        adjust $\lambda_{next}$'s location in Open
                    }
                }
            } // now done with Node
        }
        insert Node into Closed
    }
    return failure // no path found and Open is empty CFDP$_{UB}$ solution is the optimum
}
```

Although not defined in the pseudocode a *dominance order* will be defined for dealing with the μ component of the state vector as defined in section 4.3. A state λ_i dominates a state λ_j when all of the following conditions are satisfied:

- λ_i and λ_j are on the same grid (i.e. $\lambda_i.x = \lambda_j.x$, $\lambda_i.y = \lambda_j.y$, $\lambda_i.z = \lambda_j.z$)
- $\lambda_i.\mu < \lambda_j.\mu$

- *$g(\lambda_i) < g(\lambda_j)$*

Since μ having a value different from 0 imposes a constraint the *dominance order* is self explanatory.

5.3 Lower Bound Techniques

The properties of the lower bound technique employed in the A* algorithm (identified as *"$h(\lambda_{next})$ = LowerBoundEstimate(λ_{next}, λ_{end})"*) ensures whether the algorithm guarantees the optimality or not. Apart from this, the quality of *LowerBoundEstimate*[24] will decrease the states to be expanded before reaching the optimal/near optimal solution.

In the scope of the thesis 4 lower bound techniques are implemented of which the last three are custom developed:

- Take *LowerBoundEstimate* as 0
- Max Wind
- Spiral
- CFDP Layers

The following subsections will describe the techniques in detail and investigate whether each technique satisfies admissibility and monotonicity properties.

5.3.1 Take LowerBoundEstimate As 0

Taking the lower bound as 0 leads the A* algorithm directly equivalent to the Dijkstra's famous shortest path algorithm [18]. Although it guarantees optimality it will cause even expand states far from being in optimal path (in this perspective it is not important in what extent a state is "near" the end state to be prioritized; it is impotent how it is "near" the start state)

Computational complexity of "*LowerBoundEstimate As 0"* O(1).

[24] The A Algorithm keeps a sorted list of states to be expanded which are sorted by the "*Lower Bound*" assigned to that state.

"Lower Bound" is identifed as "$f(\lambda_{next}) = g(\lambda_{next})$ + LowerBoundEstimate(λ_{next}, λ_{end})" in paragraph 5.2

5.3.2 Use Maximum Wind Technique

In this lower bound technique *LowerBoundEstimate(λ_i, λ_{end})* is calculated as follows;

- The path from *(λ_i, λ_{end})* is the shortest distance connecting them,
- Through that path in cost function calculations wind speed is taken as "*maximum wind value*"[25] (call it as *maxWindValue*) and wind direction is taken as the same course along λ_{next} - λ_{end}
- TAS is taken as the maximum TAS (call it as *maxTAS*) value that can be achieved given the Gross Weight at λ_i independent of the altitude.

Proposition: *Maximum Wind* lower bound technique satisfies the admissibility property needed by A* algorithm.

Proof (Admissibility): [26]:

- Admissibility: $h(\lambda_i) \leq \min_x (\sum_{t=i}^{end} C_t(x_t(\lambda_t))$[27] (5)
- time = distance / ground speed (6)
- For $h(\lambda_i)$ distance is the direct distance from λ_i to λ_s hence showing that the ground speed used in this technique (call it $GS_{MaxWind}$) is always greater or than equal to that of used on the right hand side of (5) will suffice
- $GS_{MaxWind}$ = *maxTAS + maxWindValue* (7)
- $GS_{MaxWind} \geq GS_{\lambda_t}$ for all t = i, .., end is trivial since;
    - *maxTAS* ≥ TAS_{λ_t} and
    - *maxWindValue* ≥ *WindValue*$_t$

QED

Proposition: *Maximum Wind* lower bound technique satisfies the monotonicity property needed by A* algorithm.

[25] *Maximum wind value is calculated all over the Search Space*

[26] Proof is made taking cost function as time; the same proof steps are also applicable for fuel case with the calculation of *minFuelFLow* as an additional step:

- *minFuelFLow* is calculated as min $_{\{altitude \in T.z\}}$ (FFlow(*altitude*))
- Since Fuel Flow is inversely proportional to gross weight; calculation is performed with ($GW_{\lambda i}$ + EMPTY_AIRCRAFT_WEIGHT) / 2

[27] The cost of the minimum cost path from state λ_i to state λ_s as represented by $g(\lambda_i, \lambda_s)$ (refer to paragraph 4.4.1).

Proof (Monotonicity):

- Monotonicity: $h(\lambda_{i-1}) \leq C_{i-1}(\lambda_{i-1}, \lambda_i) + h(\lambda_i)$ for all i = 1, 2.., end (8)
- $h(\lambda_{i-1}) - h(\lambda_i) \leq C_{i-1}(\lambda_{i-1}, \lambda_i)$ (9)
- [(distance from λ_{i-1} to λ_{end})-(distance from λ_i to λ_{end})] / $GS_{MaxWind}$ ≤ (distance from λ_{i-1} to λ_i) / GS_λ (10)
- Inequality (10) is satisfied since;
  - o $GS_\lambda \leq GS_{MaxWind}$ and
  - o Taking the triangle consisting of the two nodes λ_{i-1}, λ_i and goal node λ_{end}, by the triangle inequality;

    (distance from λ_{i-1} to λ_i) > [(distance from λ_{i-1} to λ_{end})-(distance from λ_i to λ_{end})]

QED

Maximum Wind technique has a major improvement over take *LowerBoundEstimate as 0* technique such that:

- Since direct distance to ending state is used; states being "near" to the ending state has priority compared to the other ones,
- Especially in moderate weather conditions all over the search space *PathCostEstimate* will give reasonable results
- Computational complexity of *PathCostEstimate is O(1); maxWindValue* is just computed once and used every time

5.3.3 Spiral Technique

The drawback of *Maximum Wind* technique is that even a wind vector far from the optimal path and in the opposite direction is used as the tail wind component all over the path and hence producing highly underestimating results. To overcome this situation, without compromising the admissibility and monotonicity properties *spiral* technique is designed and implemented.

As shown in Figure 18 concentric circles are conceived such that;

- The visited state as at center,

- The radius of the smaller circle is grid granularity, which is 25 NM
- The radius increases outwards by grid granularity (i.e. 25 NM)
- The radius of the outermost circle is 25 * k where k is the maximum integer still satisfying "(25 * k) < *Euclidean distance to ending state*

In spiral technique *LowerBoundEstimate*(λ_i, λ_{end})is calculated as follows:

1. Calculate maximum wind value for all circles
   - For each circle C_t for t = 2, .., k the maximum wind value (call it as *maxWind*$_t$) in the area C_t / C_{t-1} is found
   - For the first circle; *maxWind*$_{C_1}$corresponds to the wind value residing in the center
2. For each circle starting from the first circle calculate the minimum cost of traversing from the inner circle (note that by the network model formulation *maxWind*$_{C_t}$is used in movement from C_t to C_{t+1})
   - The movement distance is 25 NM for all except the last one where movement distance is (*direct distance to ending state* - 12.5 + 25 * k)
   - *maxTAS*$_{C_t}$is calculated given the Gross Weight at C_t independent of the altitude (Gross Weight at C_t are maintained at each step considering the fuel used at that step)
   - GS$_{C_t}$ is calculated as *maxTAS*$_{C_t}$ + *maxWind*$_{C_t}$[28]
   - *Time*$_{C_t}$ *is 25 NM /* GS$_{C_t}$
   - *FuelFlow*$_{C_t}$ is calculated given the Gross Weight at C_t independent of the altitude
   - *FuelUsed*$_{C_t}$ *is FuelFlow*$_{C_t}$ * *Time*$_{C_t}$
3. *LowerBoundEstimate* = $\sum_{t=1}^{k} \mathrm{Cost}(C_t)$

[28] Note that the direction of movement and the wind direction are considered as the same hence the vectoral addition is the same as the scalar addition of the magnitudes.

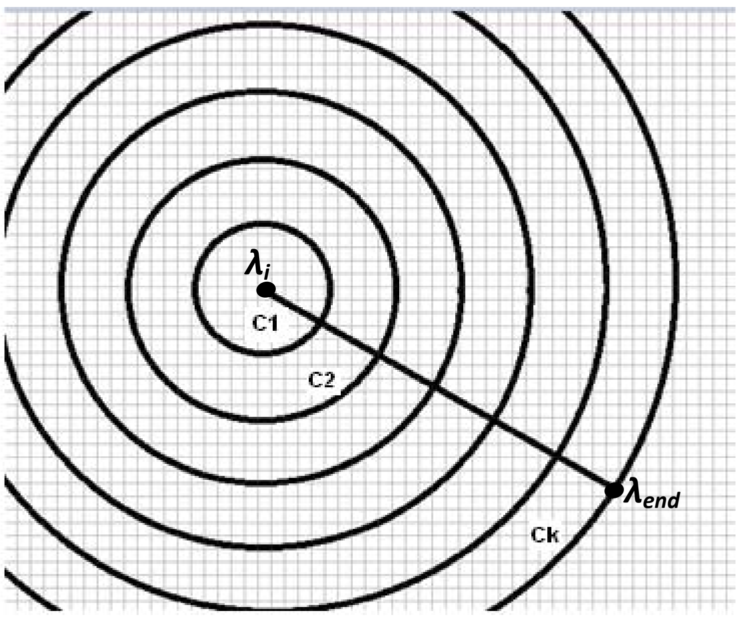


Figure 18 Spiral Technique

The spiral technique satisfies both admissibility and monotonicity properties. The proof is omitted; since the discussion will be similar to that of the "*use maximum wind*" technique.
By use of the the spiral technique the effect of a high magnitude wind is localized into just one area C_t – C_{t+1} hence producing more realistic solutions compared to "use maximum wind" technique. This improvement comes with the additional computational complexity of the lower bound. Computational complexity of PathCostEstimate is O(n * n) where n is the number of grids in the outer most circle.
During the implementation the following optimization is performed trading off counting otherwise not computed grids into the calculation; instead of searching grids in circles/circle segments the search is performed in enclosing squares; i.e. the search for C_1 is performed in S_1, the search for C_2 is performed in S_2 as shown in Figure 19.

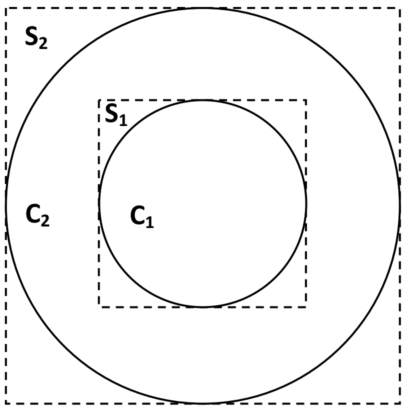


Figure 19 Finding Maximum Wind Using Squares

In addition, although not implemented in the thesis, a cache implementation of which the essentials given below would produce O(n * n / 4) results. Cache implementation depends on the fact that every visited cell has a neighborhood cell which has visited before.

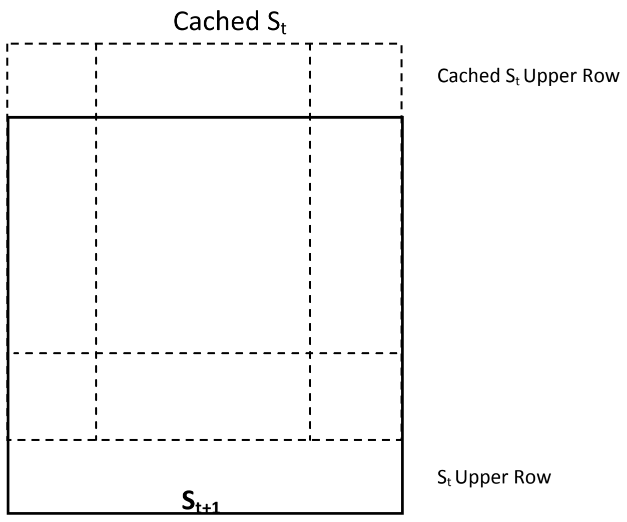


Figure 20 Finding Maximum Wind Using Caches

In this cache implementation for each square segment not just the maximum wind value but also the maximum wind values for the two rows and the two columns constituting the segment should also be stored. Referring to Figure 20 Cache implementation will reduce the problem of finding the maximum wind value in rectangle S_{t+1} (the rectangle with solid edges) into problem of replacing "The maximum wind value in Cached S_t Upper Row with the maximum wind value in S_{t+1} Upper Row".

5.3.4 CFDP Layers Technique

By the state aggregation property of CFDP it is also employed as a lower bound technique. Lower bound value is calculated by just applying one iteration of CFDP, denoted as $CFDP_H$, as shown in Figure 21.

$CFDP_H$ modeling details will be as follows:

- Search Space: 100 NM neighborhood of the axis connecting the state being investigated λ_i and λ_{end}.
- S^0 super state definition:
    - There are 2 + k stages (trellis levels) including the start and the end stages where k is truncated integer value of (distance from λ_i and λ_{end}) / Grid Granularity (25 NM)
    - There are just one superstate at each stage where Table 5 shows superstate properties

Table 5 CFDPH State Dimensions per State Level

State Level	x granularity	y granularity	z granularity	Range of S. µ	Dimension
3	|x|	8 * |y|	FL16[29]	{0}	128
2	N/A	N/A	N/A	N/A	N/A
1	N/A	N/A	N/A	N/A	N/A

[29] Super state consists of 8 pressure altitude layers where each layer's granularity is 2 FL.

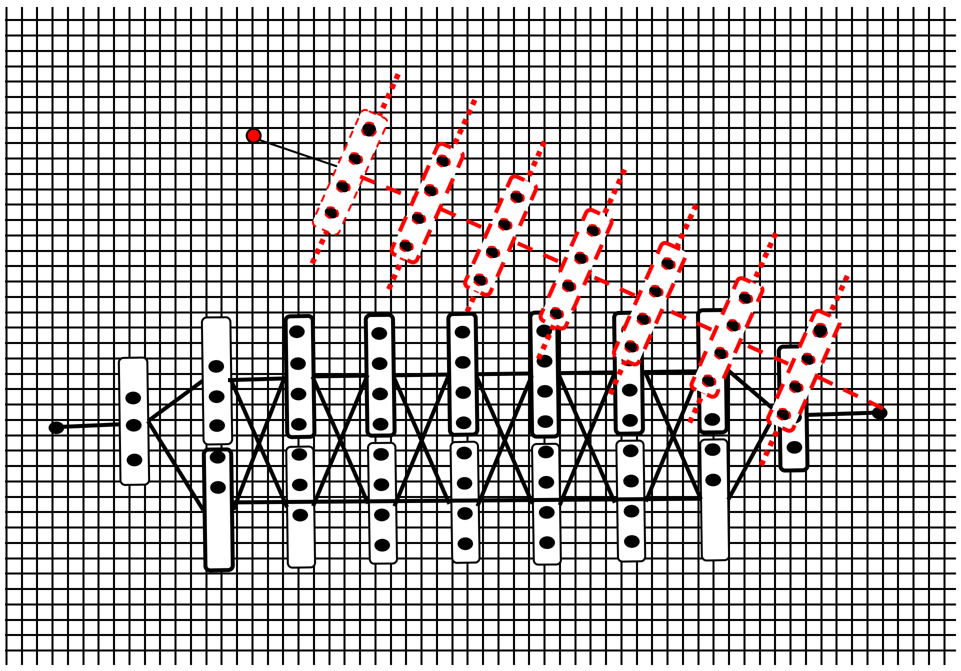


Figure 21 CFDP As a Lower Bound Technique in A* Algorithm

*The calculation of LowerBoundEstimate(λ_i, λ_{end})*is similar to that of the *spiral* technique hence here only the differences will be emphasized:

- *LowerBoundEstimate* is the summation of the cost estimate between the stages,
- Cost estimate at a stage is similar to that of the spiral where max (Tail wind component with respect to the course along λ_i-λ_{end}) is used in the calculations

Computational complexity of PathCostEstimate is O(m * n) where m is the width CFDP layers in girds and n is the Euclidean distance between λ_i and λ_{end} in grids.

CFDP$_H$ technique neither satisfies admissibility nor monotonicity properties. The reason why CFDP$_H$ is not admissible is the same why CFDP does not guarantee the optimal solution; namely:

- CFDP$_H$ prunes the search space,
- CFDP$_H$ restricts the direction of movement in only one direction

Although does not guarantee optimality, this technique provides acceptable results for the following reasons:

- Getting closer to the λ_{end} at each stage is operationally acceptable even preferable decision,
- At each stage the decision of taking maximum wind values over the tail wind components with respect to the course along λ_i-λ_{end} is not a restrictive assumption; on the opposite hand it will increase the quality of the lower bound by decreasing the wind effect and hence decreasing the speed and increasing the time predicted as the following discussion will prevail.

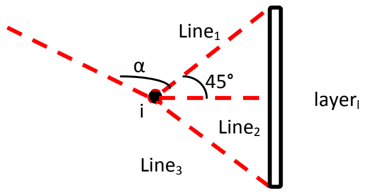


Figure 22 Horizontal versus Diagonal Movement

Referring to Figure 22 the objective is to reach the rectangle in the shortest time and there are three movement alternatives:

- Two diagonal movements (call as M_d) with step length √2
- One horizontal movement (call it as M_h) with step length 1

Proposition: When the objective is to reach layer$_l$ from point i in shortest time the horizontal movement along Line2 outweigh diagonal movements along either Line1 or Line3 (refer to Figure 22) for all operationally acceptable True Air Speeds and Wind Speed/Directions.

Proof: Find the region where "Time(M_h) < Time(M_d)" invariant will hold[30]:

- Distance(M_h) / GroundSpeed(M_h) < Distance(M_d) / GroundSpeed(M_d) (1)
- 1 / (TAS + Tail Wind Component wrt. M_h) < √2 / (TAS + Tail Wind Component wrt. M_d) (2)
- $(\mathrm{TAS} + |\mathrm{Wind}| * \cos\alpha) < \sqrt{2} * (TAS + |Wind| * cos\,(\alpha + 45))$ (3)

[30] M_h stands for the horizontal movement along Line$_2$ where M_d stands for the diagonal movement along either Line$_1$ or Line$_3$. The discussion will compare movement along Line$_1$ with Line$_2$. The discussion will be similar for the comparision between Line$_2$ and Line$_3$

- $(\mathrm{TAS} + |\mathrm{Wind}| * \cos\alpha) < \sqrt{2} * (TAS + |Wind| * (\cos\alpha * \cos 45 - \sin\alpha * \sin 45))$ (4)
- $|\mathrm{Wind}| * \cos\alpha * (1 - \sqrt{2}/2) - |\mathrm{Wind}| * - \sin\alpha * \sqrt{2}/2 < TAS * (\sqrt{2} - 1)$ (5)
- $|\mathrm{Wind}| (0.293 * \cos\alpha - 0.707 \sin\alpha) < 0.414 * TAS$ (6)

For finding the maximizing value of α in the left hand side of the inequality in (6); taking the derivative of $(0.293 * \cos\alpha - 0.707 \sin\alpha)$ and solving it for 0, α is found as -67.490°[31] and putting α into equation 6 results in:

- $0.765 * |\mathrm{Wind}| < 0.414 * TAS$ (7)
- $1.848 * |\mathrm{Wind}| < TAS$ (7)

That is as long as inequality (7) holds the proposition is valid. For C-130 aircraft wind having magnitude greater than 100 knots are considered as a storm that should be avoided to across and C-130 Performance Model suggests TAS values greater than 185 Knot for Cruise phase and hence concluding the proposition.

QED

5.4 Development

The formulated problem on hand whose modeling and high level design is described in the preceding sections is implemented to evaluate the numerical impact of changing parameters (i.e. different lower bound techniques, different edge relationship etc.) on the output.

The software program is developed in C# language and in Microsoft Visual Studio development environment. The reason of selecting C# is that it is an object oriented language and by built in capabilities it allows fast and efficient graphical interface development.

During the design and the development object oriented approach is employed where the conceptual class diagram and the sequence diagrams are given in APPENDIX A – OBJECT ORIENTED APPROACH.

[31] The second derivative of $(0.293 * \cos\alpha - 0.707 \sin\alpha)$ for "α = -67.490" is negative (-0.76531)

The development did not start from scratch; existing projects are searched so as to base the development on existing generic implementation(s) of A* algorithm and on existing graphical user interface(s).

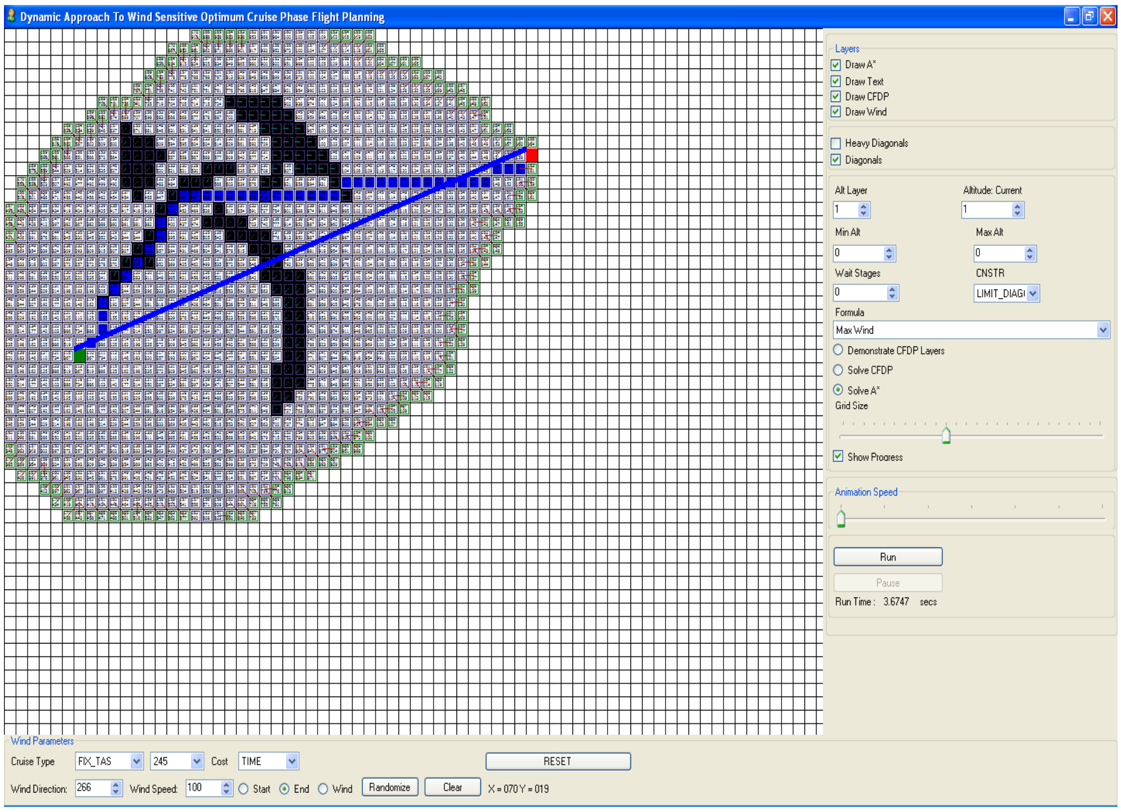
As the above rationale; “A-Star (A*) Implementation in C# (Path Finding, PathFinder)“ is used as the basis [20]. Its graphical user interface, together with its parametric structure and well documentation made its reuse feasible in the thesis.

Whereas the developed algorithm is generic by encapsulating the aircraft’s performance characteristics; it has been evaluated on C-130 aircraft. By just changing the configuration file which holds the performance model as a look up table, the developed software program can be used for any aircraft.

Once weather forecast data is incorporated the software program can be used as an *“what if”* analysis tool for the pilots or navigation officers to answer questions such as “what is the effect of following time optimum path on time and fuel compared to shortest distance path”.

As an another alternative operational usage, automatic route planning functionality may be incorporated into existing Flight Management System(s) or Mission Planning Ground System(s).

<u>Program Functionalities</u>


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Layers
- [✓] Draw A*
- [✓] Draw Text
- [✓] Draw CFDP
- [✓] Draw Wind

- [ ] Heavy Diagonals
- [✓] Diagonals

Alt Layer: 1
Altitude: Current: 1
Min Alt: 0
Max Alt: 0
Wait Stages: 0
CNSTR: LIMIT_DIAGI
Formula: Max Wind

Demonstrate CFDP Layers / Solve CFDP / [✓ Solve A*]

Grid Size

- [✓] Show Progress

Animation Speed

Run Time : 3.6747 secs

Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

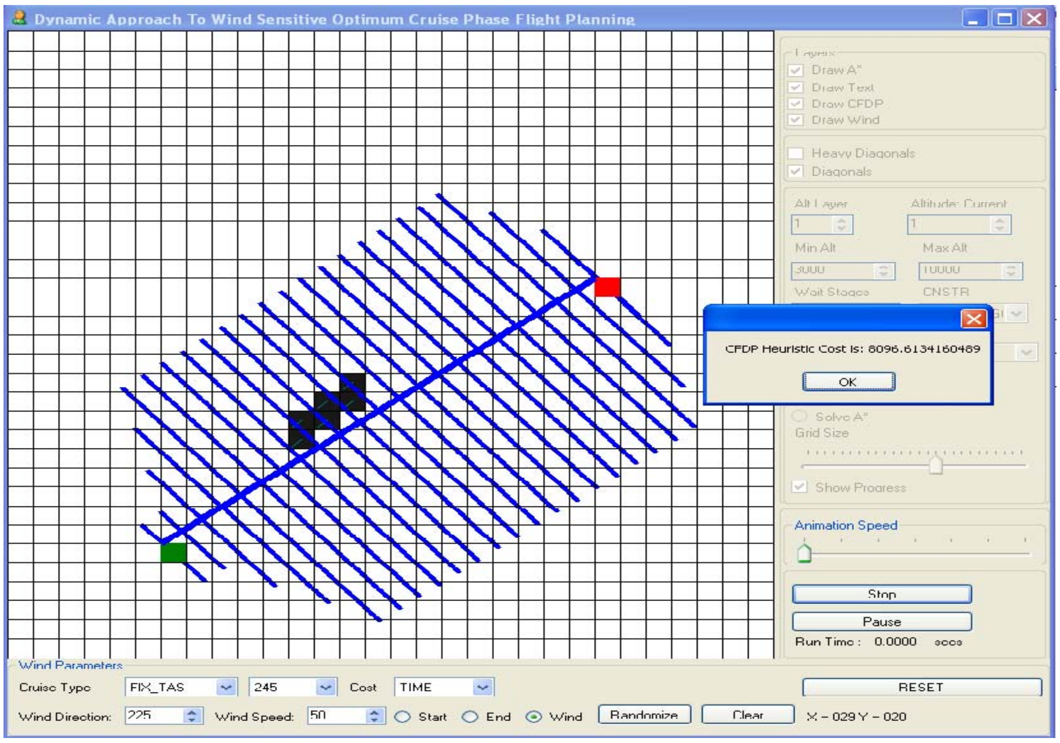
Wind Direction: 266 Wind Speed: 100 Start [✓ End] Wind

X = 070 Y = 019

Figure 23 Sample Screen Shot of Flight Planner

- Wind Parameters
    - For scenario generation user may specify wind forecast values by first selecting wind speed (0, 10, 20, ..100) and editing wind direction and the by clicking
    - User may also specify wind forecast values by randomly assigning wind values between 0 – 100
    - User may reset back to initial wind magnitude of 0
- Form Mode: Specifies the mode of the program; i.e. the action performed when “Run” button is pressed

- Demonstrate CFDP Layers: Demonstrates CFDP Layers lower bound technique by showing the calculated lower bound between the specified start and end nodes and depicting the calculated CFDP layers
- Solve CFDP: Solves and depicts the result of the path finding problem using CFDP algorithm
- Path Finder: Solves and depicts the result of the path finding problem using A* algorithm


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Layers: [✓ Draw A*] [✓ Draw Text] [✓ Draw CFDP] [✓ Draw Wind]

Heavy Diagonals [✓ Diagonals]

Alt Layer: 1 Altitude: Current: 1

Min Alt: 3000 Max Alt: 10000

Wait Stages CNSTR

CFDP Heuristic Cost Is: 8096.6134160489

OK

Solve A*

Grid Size

[✓ Show Progress]

Animation Speed

Stop

Pause

Run Time : 0.0000 secs

Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 225 Wind Speed: 50 Start End [✓ Wind]

Randomize Clear RESET

X – 029 Y – 020

Figure 24 Sample Screen Shot of Flight Planner

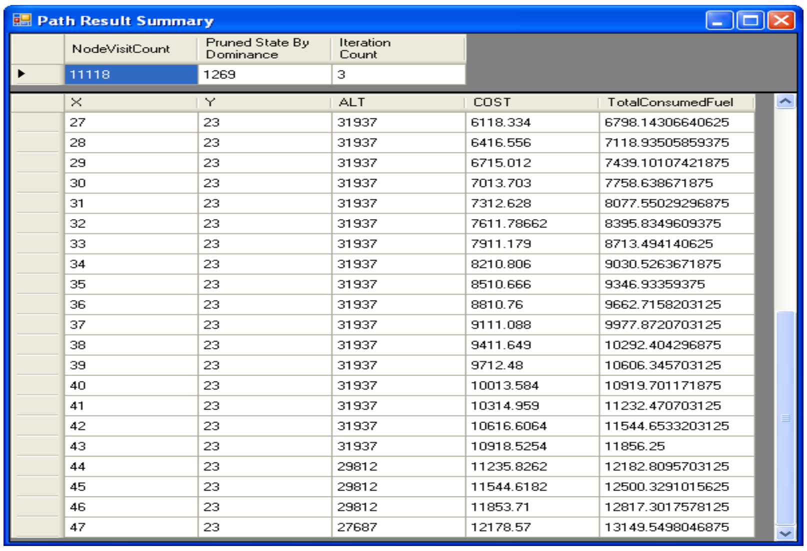
- Grid Size: Adjusts the pixel dimensions of the grids; it does not affect the geometric dimensions of the grid. Decreasing the grid size will decrease the scale and hence increase the geographic area covered in the screen.
- Layers
  - Draw A*: Specifies whether the expanded states will be depicted or not
  - Draw Text: Specifies whether F and G values of the expanded states' will be depicted or not
  - Draw CFDP: Specifies whether the current CFDP network is depicted or not

    - Draw Wind: Specifies whether wind vector of each grid is depicted or not
- Problem Definition Parameters
    - Cost Function: Specifies whether cost function is per time or per fuel.
    - Lower Bound Technique
        - Dijkstra (0)
        - MaxWind
        - CFDPLayers
        - Spiral
    - Neighborhood Schema
        - Direct (only edge neighborhood)
        - Indirect (edge and vertice neighborhood)
    - Constraints: Defines the semantic of μ state variable (refer to section 4.3)
        - Constraint Type
            - LIMIT_VERTICAL: Once the aircraft performs a climb/descent it must perform *Number of Wait Stages* level movement (same meaning as defined in in section 4.3)
            - LIMIT_DIAGONAL: Imposes a constraint on diagonal movement (only applicable for indirect neighborhood)
            - LIMIT_LEFT: Imposes a constraint on left movement
            - LIMIT_RIGHT: Imposes a constraint on diagonal movement
        - Number of Wait Stages
- Route Definition Parameters
    - Cruise Type: Specifies the cruise type on which performance model performs TAS and fuel flow calculations
    - Flight Envelope / Number of altitude layers: Specifies the vertical flight envelope (Min Alt – Max Alt) and number of discrete altitude layers a flight is allowed to be performed.

      Each altitude layer's corresponding altitude is calculated as:
        - layerIncrement = (Flight Envelope.Max Altitude - Flight Envelope.Min Altitude) / Number of altitude layers

        - 1_{st} layer's altitude=Flight Envelope.Min Altitude+layerIncrement / 2
        - i_{th} layer's altitude = 1_{st} layer's altitude + (i – 1) * layerIncrement
    - Start: Specifies the starting position of the path
    - End: Specifies the ending position of the path

      The altitude of the start/end position is determined by the Current Altitude value at the time start/end position is specified
- Path Result Summary
    - Path Result Summary: For each grid in the path shows the tabular summary including (refer to Figure 25):
        - Coordinates and altitude of the grids constituting the path
        - Cost of reaching the grid (either time or fuel consumed)
        - Total consumed fuel
    - Algorithm Metrics:
        - StateVisitCount: Number of states expanded
        - Pruned State By Dominance: Number of states pruned by use of dominance ordering (refer to section 5.2)
        - Iteration Count: In case the path problem is solved using CFDP shows the number of iterations required to reach the solution all composing of primitive states (refer to section 5.1)



Path Result Summary

NodeVisitCount	Pruned State By Dominance	Iteration Count
11118	1269	3

X	Y	ALT	COST	TotalConsumedFuel
27	23	31937	6118.334	6798.14306640625
28	23	31937	6416.556	7118.93505859375
29	23	31937	6715.012	7439.10107421875
30	23	31937	7013.703	7758.638671875
31	23	31937	7312.628	8077.55029296875
32	23	31937	7611.78662	8395.8349609375
33	23	31937	7911.179	8713.494140625
34	23	31937	8210.806	9030.5263671875
35	23	31937	8510.666	9346.93359375
36	23	31937	8810.76	9662.7158203125
37	23	31937	9111.088	9977.8720703125
38	23	31937	9411.649	10292.404296875
39	23	31937	9712.48	10606.345703125
40	23	31937	10013.584	10919.701171875
41	23	31937	10314.959	11232.470703125
42	23	31937	10616.6064	11544.6533203125
43	23	31937	10918.5254	11856.25
44	23	29812	11235.8262	12182.8095703125
45	23	29812	11544.6182	12500.3291015625
46	23	29812	11853.71	12817.3017578125
47	23	27687	12178.57	13149.5498046875

Figure 25 Path Result Summary

CHAPTER 6

CASE DEMONSTRATION

The computational experience and comparison of the effect of changing parameters (i.e. different heuristics, different edge relationship etc.) on the output will be performed and presented.

The sequence of the test cases and test case configurations are as listed in Table **6**.

Table 6 Test Cases and Purposes32

Test Case ID	Control Parameters	Purpose
testCase1	Lower Bound Technique	The effect of applying different lower bound techniques on the number of states expanded before reaching the solution
testCase2	CNSTR Wait Stages	The effect of imposing constraint (i.e. wait stages) on the solution
testCase3	Algorithm	The effect of applying CFDP versus A* algorithm
testCase4	Min – Max Altitude Altitude layers	The effect of expanding the flight envelope on the solution
testCase5	Cruise Type	The effect of changing the cruise type on the solution
testCase6	Cost Type	The effect of changing the cost type on the solution

[32] Unless otherwise stated Cost Tpe is Time (seconds)

Table 6 Test Cases andPurposes cont'd

Test Case ID	Control Parameters	Purpose
testCase7	Realistic scenario	A* algorithm and CFDP are evaluated on a realistic scenario where altitude range is [16000, 32000] and there are 8 altitude layers. In addition for all grids in all altitude layers wind is randomly generated
testCase8	Oceaning Scenario	ETP (Equal Time Point) is calculated both by the existing legacy method and by the algorithm developed in thesis and the results are compared where for all grids in all altitude layers wind is randomly generated

Depending on the test case's purpose, scenarios in each case are compared based on *StateVisitCount* metric or the value of the objective function.

Throughout the thesis as the complexity metric *StateVisitCount* is used while another alternative would be the Run Time. The reason for is that Run Time is affected by many factors not related with the algorithmic complexity such as;

- Depicting the expanded nodes,
- Use of good programming practices (i.e. using binary heaps instead of ordinary arrays etc, multitasking etc.)

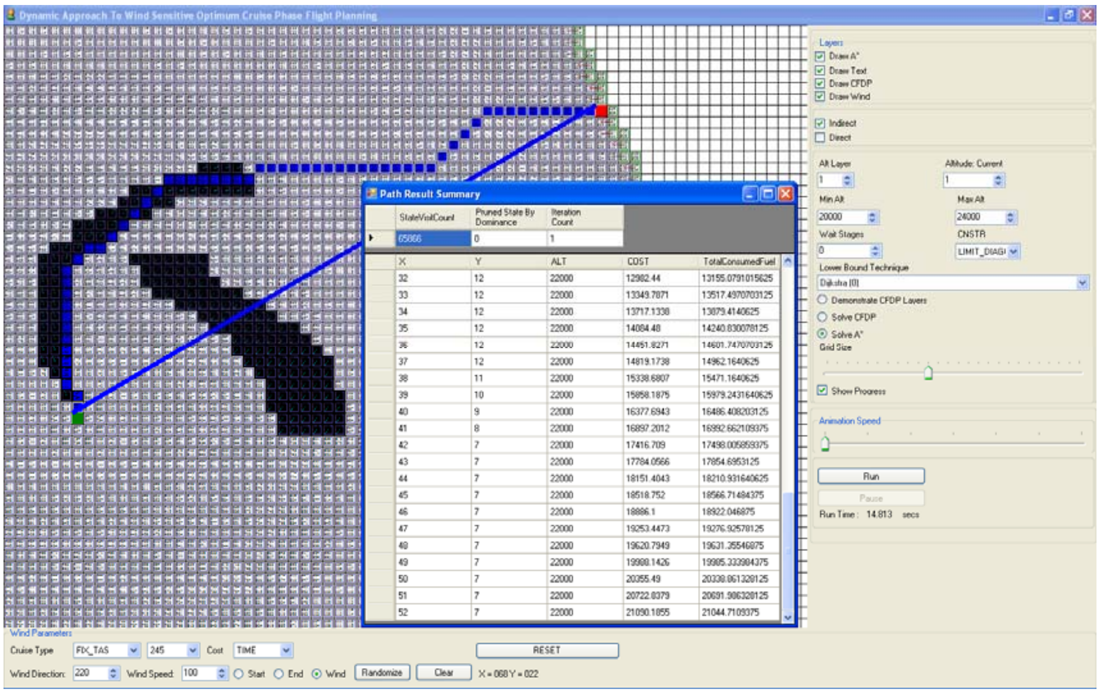
6.1 Test Case 1

Table 7 Test Case 1 Results

Scenario Id	Control Parameter	StateVisitCount	Objective Function
scenario1	Dijkstra	65866	21090.2
scenario2	Max Wind	27551	21090.2
scenario3	Spiral	22378	21090.2
Scenario4	CFDP Layers	13671	21090.2

All lower bound techniques produce the same result (5 hours and 52 minutes) where the Euclidean distance of Start – End is 1333 NM. This must be the case for Dijkstra, Max Wind and Spiral cases since these techniques guarantee optimality (for results refer to Figure 26, Figure 27, Figure 28, Figure 29 respectively).

Improvement gain of the other three techniques over Dijkstra depends on the distribution of wind (Max Wind just depends on the maximum wind value). It should be noted that independent of the distribution Spiral technique must always outweigh, if not being equal, Max Wind technique.



Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Path Result Summary

	StateVisitCount	Pruned State By Dominance	Iteration Count
▸	65866	0	1

X	Y	ALT	COST	TotalConsumedFuel
32	12	22000	12982.44	13155.0791015625
33	12	22000	13349.7871	13517.4970703125
34	12	22000	13717.1338	13879.4140625
35	12	22000	14084.48	14240.830078125
36	12	22000	14451.8271	14601.7470703125
37	12	22000	14819.1738	14962.1640625
38	11	22000	15338.6807	15471.1640625
39	10	22000	15858.1875	15979.2431640625
40	9	22000	16377.6943	16486.408203125
41	8	22000	16897.2012	16992.662109375
42	7	22000	17416.709	17498.005859375
43	7	22000	17784.0566	17854.6953125
44	7	22000	18151.4043	18210.931640625
45	7	22000	18518.752	18566.71484375
46	7	22000	18886.1	18922.046875
47	7	22000	19253.4473	19276.92578125
48	7	22000	19620.7949	19631.35546875
49	7	22000	19988.1426	19985.333984375
50	7	22000	20355.49	20338.861328125
51	7	22000	20722.8379	20691.986328125
52	7	22000	21090.1855	21044.7109375

Layers: [✓] Draw A* [✓] Draw Text [✓] Draw CFDP [✓] Draw Wind

[✓] Indirect [ ] Direct

Alt Layer: 1 Altitude: Current: 1

Min Alt: 20000 Max Alt: 24000

Wait Stages: 0 CNSTR: LIMIT_DIAG

Lower Bound Technique: Dijkstra (0)

○ Demonstrate CFDP Layers ○ Solve CFDP [✓ Solve A*]

[✓] Show Progress

Run Time: 14.813 secs

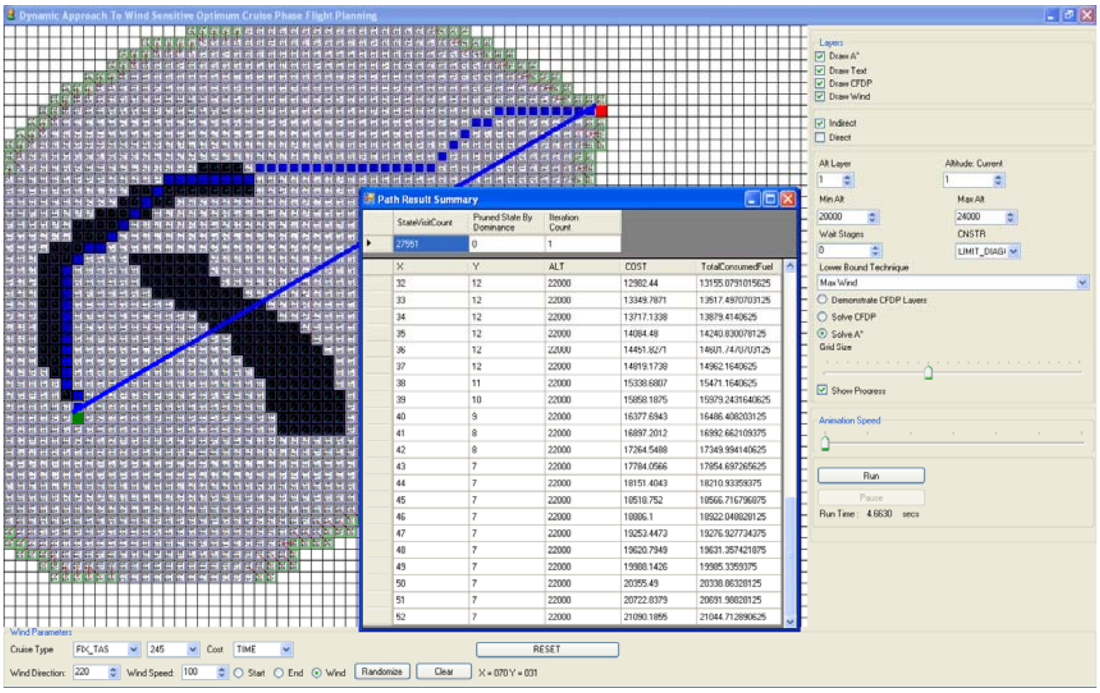
Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 220 Wind Speed: 100 ○ Start ○ End [✓ Wind]

X = 068 Y = 022

Figure 26 testCase1-scenario1



Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Path Result Summary

	StateVisitCount	Pruned State By Dominance	Iteration Count
▸	27951	0	1

X	Y	ALT	COST	TotalConsumedFuel
32	12	22000	12982.44	13155.0791015625
33	12	22000	13349.7871	13517.4970703125
34	12	22000	13717.1338	13879.4140625
35	12	22000	14084.48	14240.830078125
36	12	22000	14451.8271	14601.7470703125
37	12	22000	14819.1738	14962.1640625
38	11	22000	15338.6807	15471.1640625
39	10	22000	15858.1875	15979.2431640625
40	9	22000	16377.6943	16486.408203125
41	8	22000	16897.2012	16992.662109375
42	8	22000	17264.5488	17349.994140625
43	7	22000	17784.0566	17854.697265625
44	7	22000	18151.4043	18210.93359375
45	7	22000	18518.752	18566.716796875
46	7	22000	18886.1	18922.048828125
47	7	22000	19253.4473	19276.927734375
48	7	22000	19620.7949	19631.357421875
49	7	22000	19988.1426	19985.3359375
50	7	22000	20355.49	20338.86328125
51	7	22000	20722.8379	20691.98828125
52	7	22000	21090.1855	21044.712890625

Layers: [✓] Draw A* [✓] Draw Text [✓] Draw CFDP [✓] Draw Wind

[✓] Indirect [ ] Direct

Alt Layer: 1 Altitude: Current: 1

Min Alt: 20000 Max Alt: 24000

Wait Stages: 0 CNSTR: LIMIT_DIAG

Lower Bound Technique: Max Wind

○ Demonstrate CFDP Layers ○ Solve CFDP [✓ Solve A*]

[✓] Show Progress

Run Time: 4.6630 secs

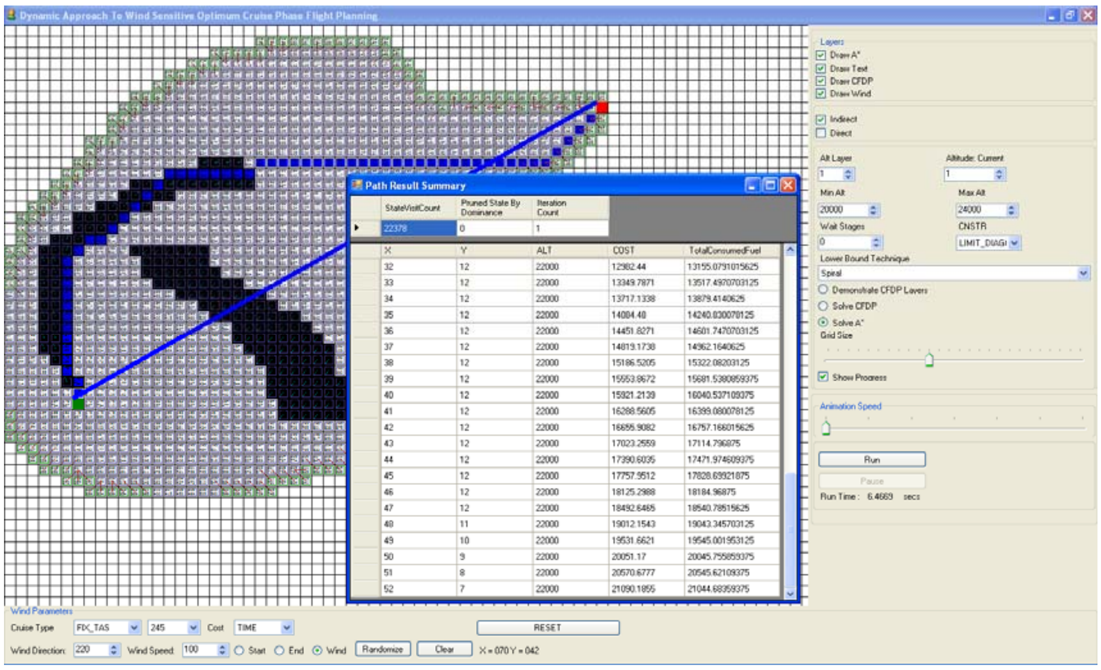
Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 220 Wind Speed: 100 ○ Start ○ End [✓ Wind]

X = 070 Y = 031

Figure 27 testCase1-scenario2



Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Path Result Summary

StateVisitCount	Pruned State By Dominance	Iteration Count
22378	0	1

X	Y	ALT	COST	TotalConsumedFuel
32	12	22000	12982.44	13155.0791015625
33	12	22000	13349.7871	13517.4970703125
34	12	22000	13717.1338	13879.4140625
35	12	22000	14084.48	14240.830078125
36	12	22000	14451.8271	14601.7470703125
37	12	22000	14819.1738	14962.1640625
38	12	22000	15186.5205	15322.08203125
39	12	22000	15553.8672	15681.5380859375
40	12	22000	15921.2139	16040.537109375
41	12	22000	16288.5605	16399.080078125
42	12	22000	16655.9082	16757.166015625
43	12	22000	17023.2559	17114.796875
44	12	22000	17390.6035	17471.974609375
45	12	22000	17757.9512	17828.69921875
46	12	22000	18125.2988	18184.96875
47	12	22000	18492.6465	18540.78515625
48	11	22000	19012.1543	19043.345703125
49	10	22000	19531.6621	19545.001953125
50	9	22000	20051.17	20045.755859375
51	8	22000	20570.6777	20545.62109375
52	7	22000	21090.1855	21044.68359375

Layers
- [✓] Draw A*
- [✓] Draw Text
- [✓] Draw CFDP
- [✓] Draw Wind
- [✓] Indirect
- [ ] Direct

Alt Layer: 1	Altitude: Current: 1
Min Alt: 20000	Max Alt: 24000
Wait Stages: 0	CNSTR: LIMIT_DIAGI

Lower Bound Technique: Spiral

Demonstrate CFDP Layers / Solve CFDP / [✓ Solve A*]

Grid Size

[✓] Show Progress

Animation Speed

Run

Pause

Run Time : 6.4669 secs

Wind Parameters

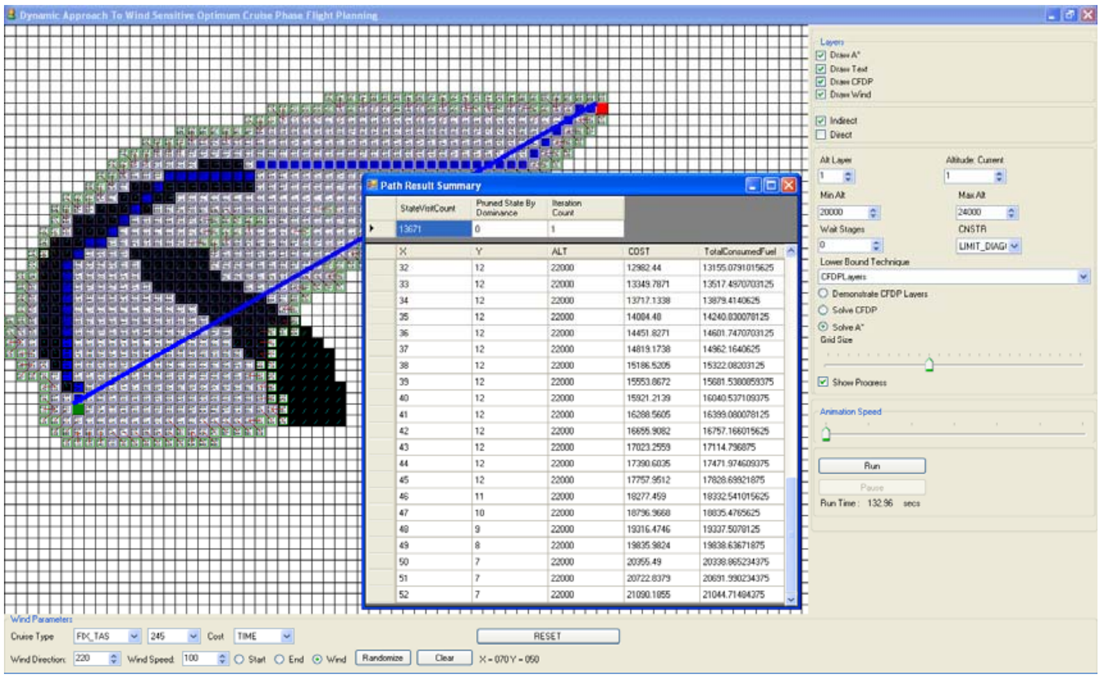
Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 220 Wind Speed: 100 Start / End / [✓ Wind]

Randomize Clear X = 070 Y = 042

RESET

Figure 28 testCase1-scenario3



Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Path Result Summary

StateVisitCount	Pruned State By Dominance	Iteration Count
13671	0	1

X	Y	ALT	COST	TotalConsumedFuel
32	12	22000	12982.44	13155.0791015625
33	12	22000	13349.7871	13517.4970703125
34	12	22000	13717.1338	13879.4140625
35	12	22000	14084.48	14240.830078125
36	12	22000	14451.8271	14601.7470703125
37	12	22000	14819.1738	14962.1640625
38	12	22000	15186.5205	15322.08203125
39	12	22000	15553.8672	15681.5380859375
40	12	22000	15921.2139	16040.537109375
41	12	22000	16288.5605	16399.080078125
42	12	22000	16655.9082	16757.166015625
43	12	22000	17023.2559	17114.796875
44	12	22000	17390.6035	17471.974609375
45	12	22000	17757.9512	17828.69921875
46	11	22000	18277.459	18332.541015625
47	10	22000	18796.9668	18835.4765625
48	9	22000	19316.4746	19337.5078125
49	8	22000	19835.9824	19838.63671875
50	7	22000	20355.49	20338.865234375
51	7	22000	20722.8379	20691.990234375
52	7	22000	21090.1855	21044.71484375

Layers
- [✓] Draw A*
- [✓] Draw Text
- [✓] Draw CFDP
- [✓] Draw Wind
- [✓] Indirect
- [ ] Direct

Alt Layer: 1	Altitude: Current: 1
Min Alt: 20000	Max Alt: 24000
Wait Stages: 0	CNSTR: LIMIT_DIAGI

Lower Bound Technique: CFDPLayers

Demonstrate CFDP Layers / Solve CFDP / [✓ Solve A*]

Grid Size

[✓] Show Progress

Animation Speed

Run

Pause

Run Time : 132.96 secs

Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 220 Wind Speed: 100 Start / End / [✓ Wind]

Randomize Clear X = 070 Y = 050

RESET

Figure 29 testCase1-scenario4

6.2 Test Case 2

Table 8 Test Case 2 Results

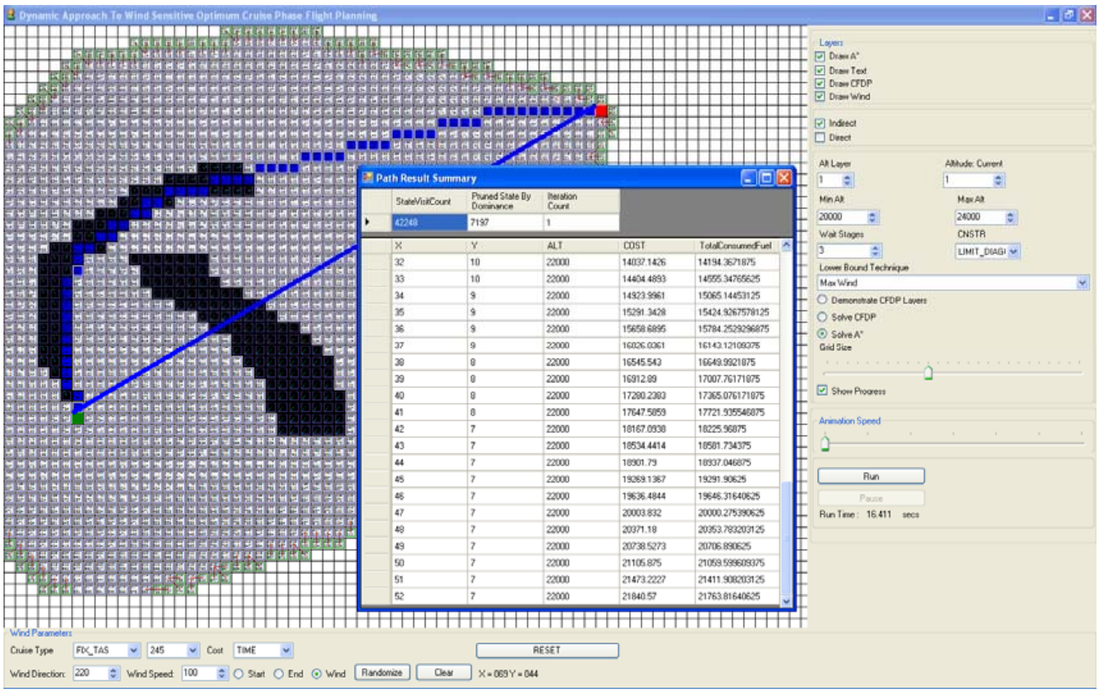
Scenario Id	Control Parameter	StateVisitCount[33]	Objective Function
scenario1	Limit Diagonal Wait Stage 3	42248	21840.6
scenario2	Limit Diagonal Wait Stage 2	39319	21671.1
scenario3	Limit Right Wait Stage 2	103715	48660

In scenario1 imposing a constraint on the same routing problem defined in Test Case 1 worsens the objective function to (6 hours and 4 minutes) and relaxing the constraint by one level in scenario2 the objective function becomes 6 hours and 1 minute.

In scenario3 a tight constraint is imposed such that “once the aircraft makes a Right movement it can not make a right movement before making two other non Right moves”.

Imposing the constraint adds one more dimension to the state variable hence causing StateVisitCount to increase; from 27551 (refer to TestCase2-Scenario2) to values in Table 8. The effect on the lateral path can be seen by comparing Figure 30, Figure 31 and Figure 32 with the figures in section 6.1.

[33] Max Wind lower bound techniques is used in all scenarios



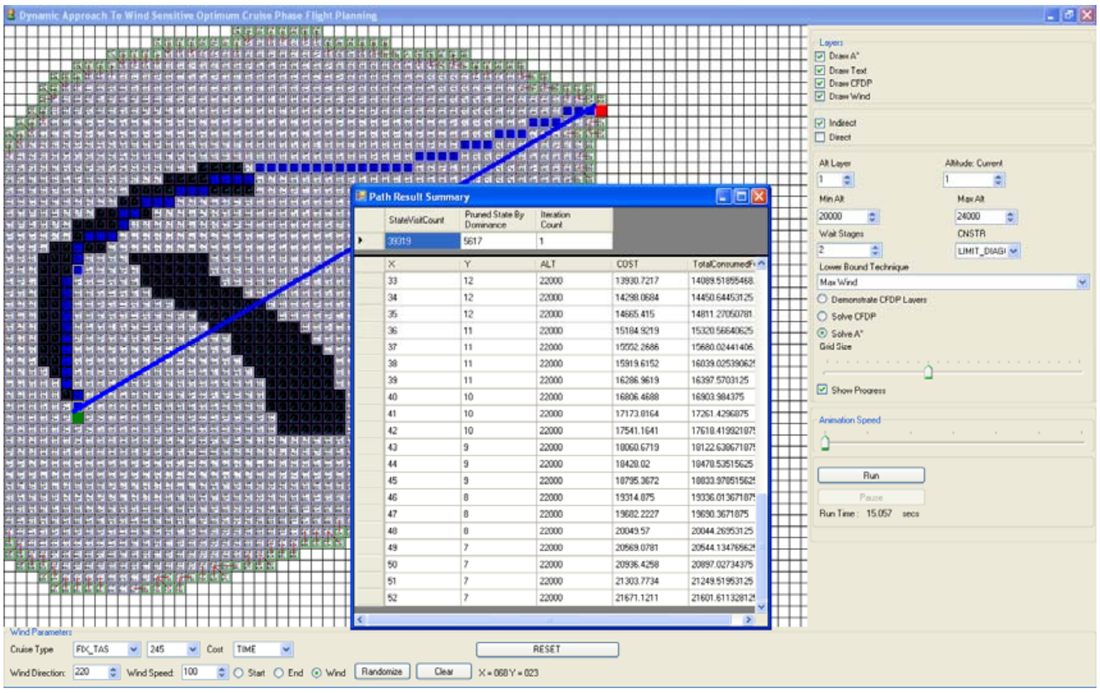
StateVisitCount	Pruned State By Dominance	Iteration Count
42248	7197	1

X	Y	ALT	COST	TotalConsumedFuel
32	10	22000	14037.1426	14194.3671875
33	10	22000	14404.4893	14555.34765625
34	9	22000	14923.9961	15065.14453125
35	9	22000	15291.3428	15424.9267578125
36	9	22000	15658.6895	15784.2529296875
37	9	22000	16026.0361	16143.12109375
38	8	22000	16545.543	16649.9921875
39	8	22000	16912.89	17007.76171875
40	8	22000	17280.2383	17365.076171875
41	8	22000	17647.5859	17721.935546875
42	7	22000	18167.0938	18225.96875
43	7	22000	18534.4414	18581.734375
44	7	22000	18901.79	18937.046875
45	7	22000	19269.1367	19291.90625
46	7	22000	19636.4844	19646.31640625
47	7	22000	20003.832	20000.275390625
48	7	22000	20371.18	20353.783203125
49	7	22000	20738.5273	20706.890625
50	7	22000	21105.875	21059.599609375
51	7	22000	21473.2227	21411.908203125
52	7	22000	21840.57	21763.81640625

☑ Draw A*
☑ Draw Text
☑ Draw CFDP
☑ Draw Wind
☑ Indirect
☐ Direct
Alt Layer: 1 Altitude: Current: 1
Min Alt: 20000 Max Alt: 24000
Wait Stages: 3 CNSTR: LIMIT_DIAG
Lower Bound Technique: Max Wind
○ Demonstrate CFDP Layers
○ Solve CFDP
◉ Solve A*
☑ Show Progress
Run Time : 16.411 secs

Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 220 Wind Speed: 100 ○ Start ○ End ◉ Wind X = 069 Y = 044

Figure 30 testCase2-scenario1



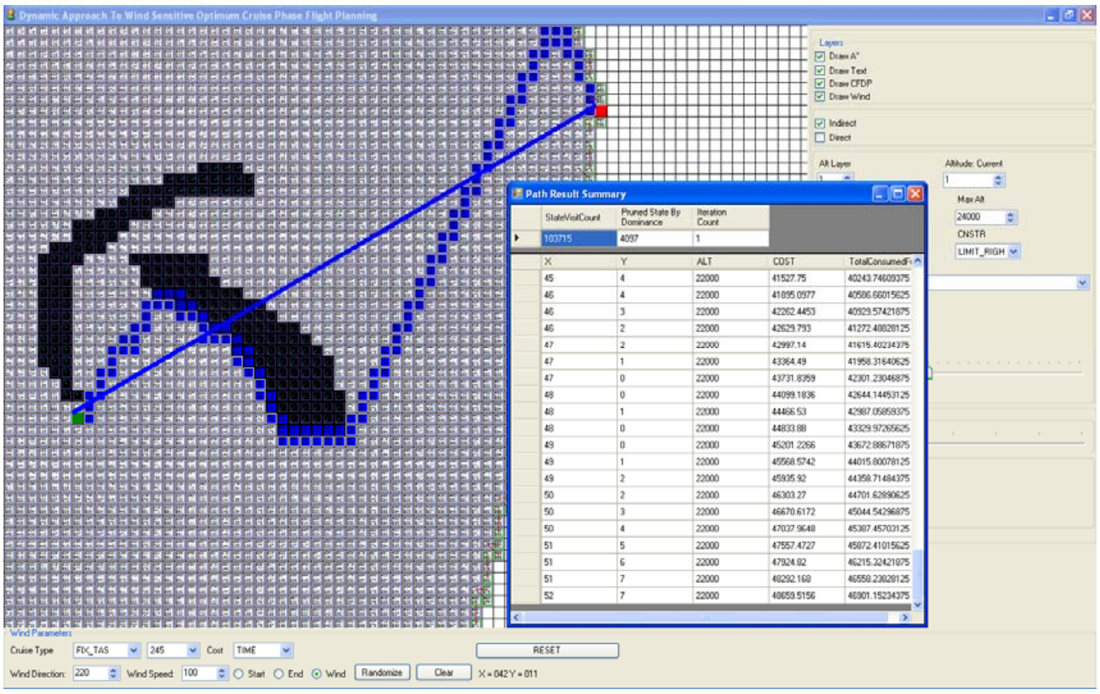
StateVisitCount	Pruned State By Dominance	Iteration Count
39319	5617	1

X	Y	ALT	COST	TotalConsumedF
33	12	22000	13930.7217	14089.51855468
34	12	22000	14298.0684	14450.64453125
35	12	22000	14665.415	14811.27050781
36	11	22000	15184.9219	15320.56640625
37	11	22000	15552.2686	15680.02441406
38	11	22000	15919.6152	16039.02539062
39	11	22000	16286.9619	16397.5703125
40	10	22000	16806.4688	16903.984375
41	10	22000	17173.8164	17261.4296875
42	10	22000	17541.1641	17618.41992187
43	9	22000	18060.6719	18122.63867187
44	9	22000	18428.02	18478.53515625
45	9	22000	18795.3672	18833.97851562
46	8	22000	19314.875	19336.01367187
47	8	22000	19682.2227	19690.3671875
48	8	22000	20049.57	20044.26953125
49	7	22000	20569.0781	20544.13476562
50	7	22000	20936.4258	20897.02734375
51	7	22000	21303.7734	21249.51953125
52	7	22000	21671.1211	21601.61132812

☑ Draw A*
☑ Draw Text
☑ Draw CFDP
☑ Draw Wind
☑ Indirect
☐ Direct
Alt Layer: 1 Altitude: Current: 1
Min Alt: 20000 Max Alt: 24000
Wait Stages: 2 CNSTR: LIMIT_DIAG
Lower Bound Technique: Max Wind
○ Demonstrate CFDP Layers
○ Solve CFDP
◉ Solve A*
☑ Show Progress
Run Time : 15.057 secs

Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 220 Wind Speed: 100 ○ Start ○ End ◉ Wind X = 068 Y = 023

Figure 31 testCase2-scenario2


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning

Layers: Draw A* [✓], Draw Text [✓], Draw CFDP [✓], Draw Wind [✓]

Indirect [✓], Direct [ ]

Alt Layer: 1 | Altitude: Current: 1 | Max Alt: 24000 | CNSTR: LIMIT_RIGH

Path Result Summary

	StateVisitCount	Pruned State By Dominance	Iteration Count
▸	103715	4097	1

X	Y	ALT	COST	TotalConsumedF
45	4	22000	41527.75	40243.74609375
46	4	22000	41895.0977	40586.66015625
46	3	22000	42262.4453	40929.57421875
46	2	22000	42629.793	41272.48828125
47	2	22000	42997.14	41615.40234375
47	1	22000	43364.49	41958.31640625
47	0	22000	43731.8359	42301.23046875
48	0	22000	44099.1836	42644.14453125
48	1	22000	44466.53	42987.05859375
48	0	22000	44833.88	43329.97265625
49	0	22000	45201.2266	43672.88671875
49	1	22000	45568.5742	44015.80078125
49	2	22000	45935.92	44358.71484375
50	2	22000	46303.27	44701.62890625
50	3	22000	46670.6172	45044.54296875
50	4	22000	47037.9648	45387.45703125
51	5	22000	47557.4727	45872.41015625
51	6	22000	47924.82	46215.32421875
51	7	22000	48292.168	46558.23828125
52	7	22000	48659.5156	46901.15234375

Wind Parameters

Cruise Type: FIX_TAS | 245 | Cost: TIME

Wind Direction: 220 | Wind Speed: 100 | Start ( ) End ( ) [✓ Wind]

X = 042 Y = 011

Figure 32 testCase2-scenario3

6.3 Test Case 3

Table 9 Test Case 3 Results

Scenario Id	Control Parameter	StateVisitCount[34]	Objective Function
scenario1	A*	17019	17781.7
scenario2	CFDP	12555[35]	17781.7
scenario3	A*	10603	15158.0
scenario4	CFDP	8711[36]	15782.3

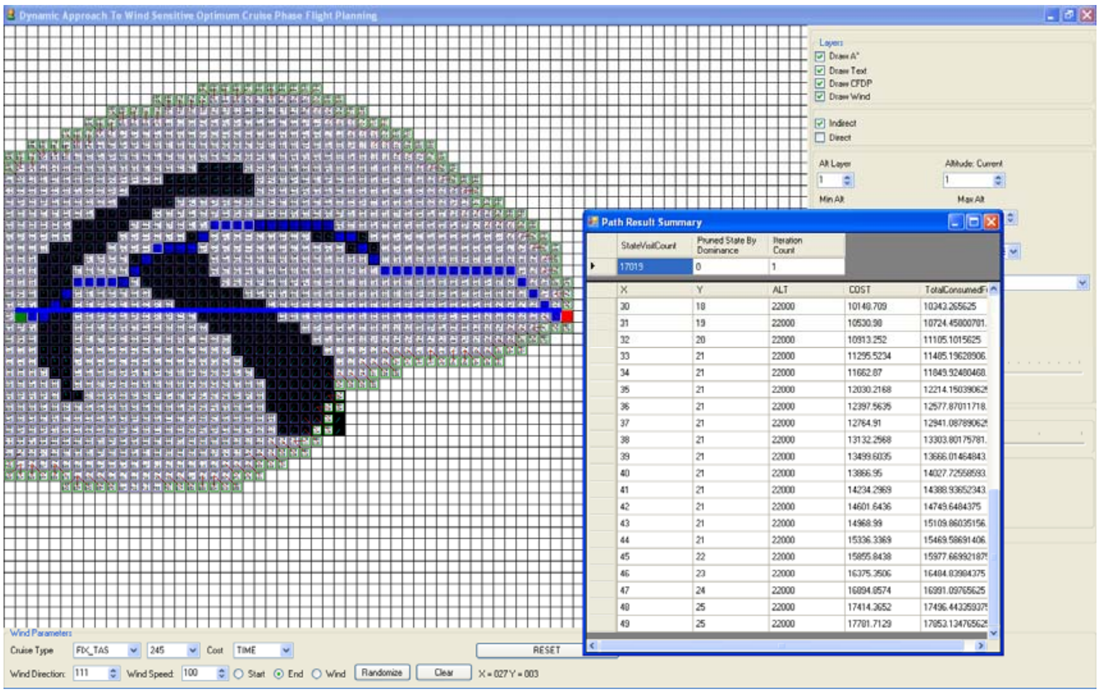
[34] Max Wind technique is employed in all scenarios

[35] Completed in 9 iterations

[36] Completed in 7 iterations

In this test case the same route planning problem is solved by A* algorithm and CFDP and results are evaluated. Note that CFDP employs A* algorithm as path solver including the lower bound technique; aggregating the states and then departitioning the ones in the optimum path of the last iteration into smaller states what CFDP performs on top of A* algorithm.

In scenario 1 and 2 CFDP outperforms A* algorithm (Figure 33, Figure 34); after arranging wind values in favor of following northwards path in scenario3 and scenario4 (refer to Figure 35 and Figure 36) CFDP does not guarantee optimality. This is because CFDP state space is constructed with 200 NM buffer (100 NM along each side) along the *λstart* -λ_{end} axis.


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Layers
Path Result Summary
Wind Parameters
RESET
Randomize
Clear

[✓] Draw A*
[✓] Draw Text
[✓] Draw CFDP
[✓] Draw Wind

[✓] Indirect
[ ] Direct

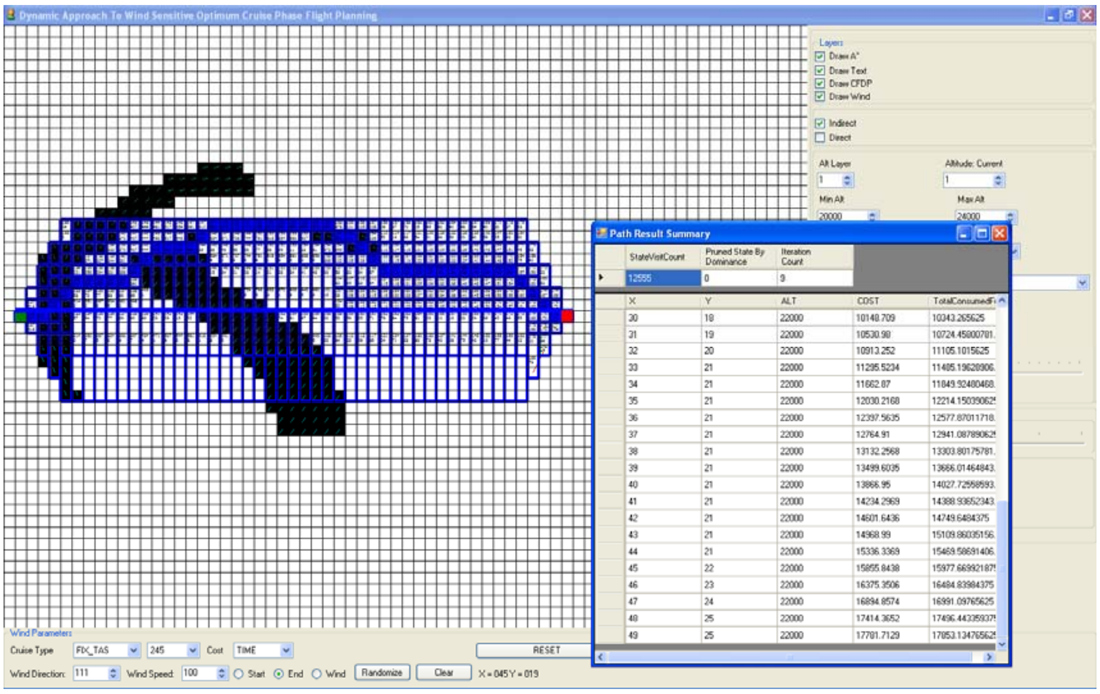
Alt Layer: 1 Altitude: Current: 1
Min Alt Max Alt

StateVisitCount	Pruned State By Dominance	Iteration Count
17019	0	1

X	Y	ALT	COST	TotalConsumedF
30	18	22000	10148.709	10343.265625
31	19	22000	10530.98	10724.45800701.
32	20	22000	10913.252	11105.1015625
33	21	22000	11295.5234	11485.19628906.
34	21	22000	11662.87	11849.92480468.
35	21	22000	12030.2168	12214.15039062
36	21	22000	12397.5635	12577.87011718.
37	21	22000	12764.91	12941.08789062
38	21	22000	13132.2568	13303.80175781.
39	21	22000	13499.6035	13666.01464843.
40	21	22000	13866.95	14027.72558593.
41	21	22000	14234.2969	14388.93652343.
42	21	22000	14601.6436	14749.6484375
43	21	22000	14968.99	15109.86035156.
44	21	22000	15336.3369	15469.58691406.
45	22	22000	15855.8438	15977.66992187
46	23	22000	16375.3506	16484.83984375
47	24	22000	16894.8574	16991.09765625
48	25	22000	17414.3652	17496.44335937
49	25	22000	17701.7129	17853.13476562

Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 111 Wind Speed: 100 ( ) Start [✓ End] ( ) Wind X = 027 Y = 003

Figure 33 testCase3-scenario1



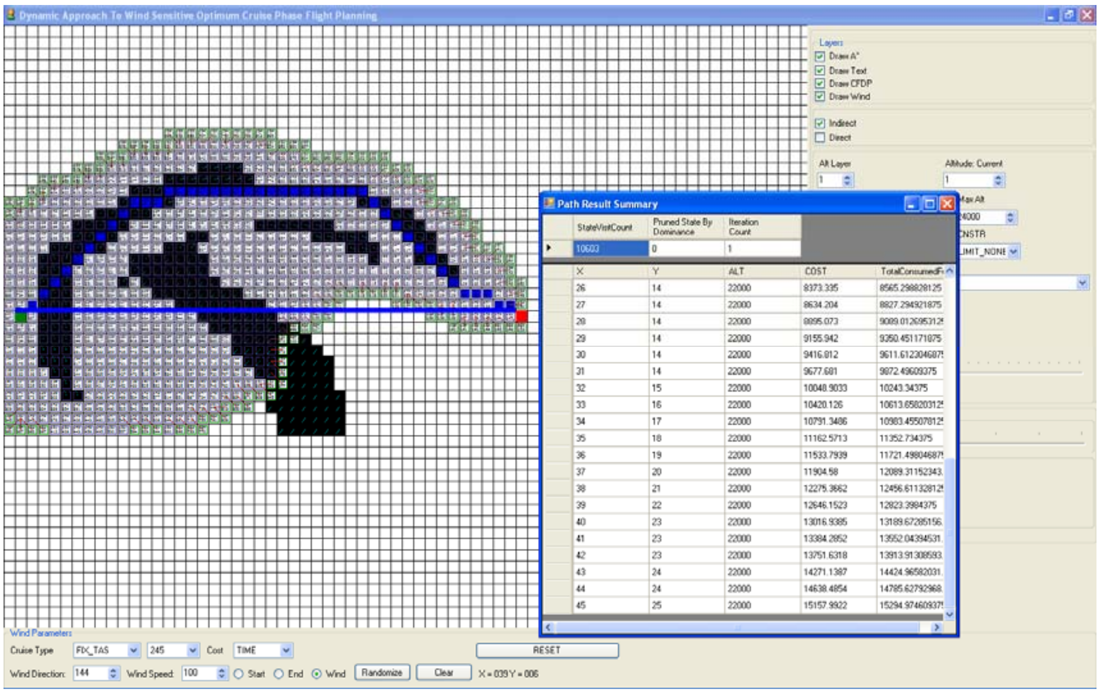
Alt Layer: 1 Altitude: Current: 1
Min Alt: 20000 Max Alt: 24000

StateVisitCount	Pruned State By Dominance	Iteration Count
12555	0	9

X	Y	ALT	COST	TotalConsumedF
30	18	22000	10148.709	10343.265625
31	19	22000	10530.98	10724.45800781.
32	20	22000	10913.252	11105.1015625
33	21	22000	11295.5234	11485.19628906.
34	21	22000	11662.87	11849.92480468.
35	21	22000	12030.2168	12214.15039062
36	21	22000	12397.5635	12577.87011718.
37	21	22000	12764.91	12941.08789062
38	21	22000	13132.2568	13303.80175781.
39	21	22000	13499.6035	13666.01464843.
40	21	22000	13866.95	14027.72558593.
41	21	22000	14234.2969	14388.93652343.
42	21	22000	14601.6436	14749.6484375
43	21	22000	14968.99	15109.86035156.
44	21	22000	15336.3369	15469.58691406.
45	22	22000	15855.8438	15977.66992187
46	23	22000	16375.3506	16484.83984375
47	24	22000	16894.8574	16991.09765625
48	25	22000	17414.3652	17496.44335937
49	25	22000	17781.7129	17853.13476562

Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 111 Wind Speed: 100 ( ) Start (•) End ( ) Wind X = 045 Y = 019

Figure 34 testCase3-scenario2



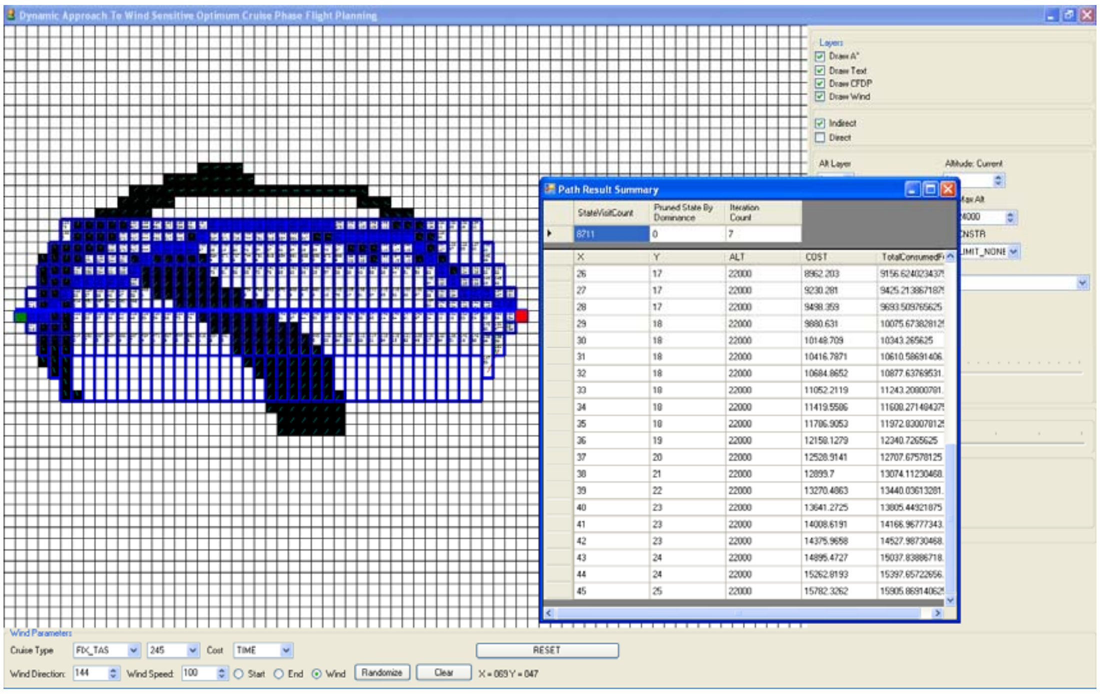
Alt Layer: 1 Altitude: Current: 1
Max Alt: 24000
CNSTR: LIMIT_NONE

StateVisitCount	Pruned State By Dominance	Iteration Count
10603	0	1

X	Y	ALT	COST	TotalConsumedF
26	14	22000	8373.335	8565.29882812 5
27	14	22000	8634.204	8827.294921875
28	14	22000	8895.073	9089.01269531 2
29	14	22000	9155.942	9350.451171875
30	14	22000	9416.812	9611.61230468 7
31	14	22000	9677.681	9872.49609375
32	15	22000	10048.9033	10243.34375
33	16	22000	10420.126	10613.65820312
34	17	22000	10791.3486	10983.45507812
35	18	22000	11162.5713	11352.734375
36	19	22000	11533.7939	11721.49804687
37	20	22000	11904.58	12089.31152343.
38	21	22000	12275.3662	12456.61132812
39	22	22000	12646.1523	12823.3984375
40	23	22000	13016.9385	13189.67285156.
41	23	22000	13384.2852	13552.04394531.
42	23	22000	13751.6318	13913.91308593.
43	24	22000	14271.1387	14424.96582031.
44	24	22000	14638.4854	14785.62792968.
45	25	22000	15157.9922	15294.97460937

Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 144 Wind Speed: 100 ( ) Start ( ) End (•) Wind X = 039 Y = 006

Figure 35 testCase3-scenario3



Path Result Summary

	StateVisitCount	Pruned State By Dominance	Iteration Count
▸	8711	0	7

X	Y	ALT	COST	TotalConsumedF
26	17	22000	8962.203	9156.624023437
27	17	22000	9230.281	9425.213867187
28	17	22000	9498.359	9693.509765625
29	18	22000	9880.631	10075.67382812
30	18	22000	10148.709	10343.265625
31	18	22000	10416.7871	10610.58691406
32	18	22000	10684.8652	10877.63769531
33	18	22000	11052.2119	11243.20800781
34	18	22000	11419.5506	11608.27148437
35	18	22000	11786.9053	11972.83007812
36	19	22000	12158.1279	12340.7265625
37	20	22000	12528.9141	12707.67578125
38	21	22000	12899.7	13074.11230468
39	22	22000	13270.4863	13440.03613281
40	23	22000	13641.2725	13805.44921875
41	23	22000	14008.6191	14166.96777343
42	23	22000	14375.9658	14527.98730468
43	24	22000	14895.4727	15037.83886718
44	24	22000	15262.8193	15397.65722656
45	25	22000	15782.3262	15905.86914062

Max Alt: 24000
CNSTR: LIMIT_NONE

Wind Parameters
Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 144 Wind Speed: 100 Start End [✓ Wind]
X = 069 Y = 047

Figure 36 testCase3-scenario4

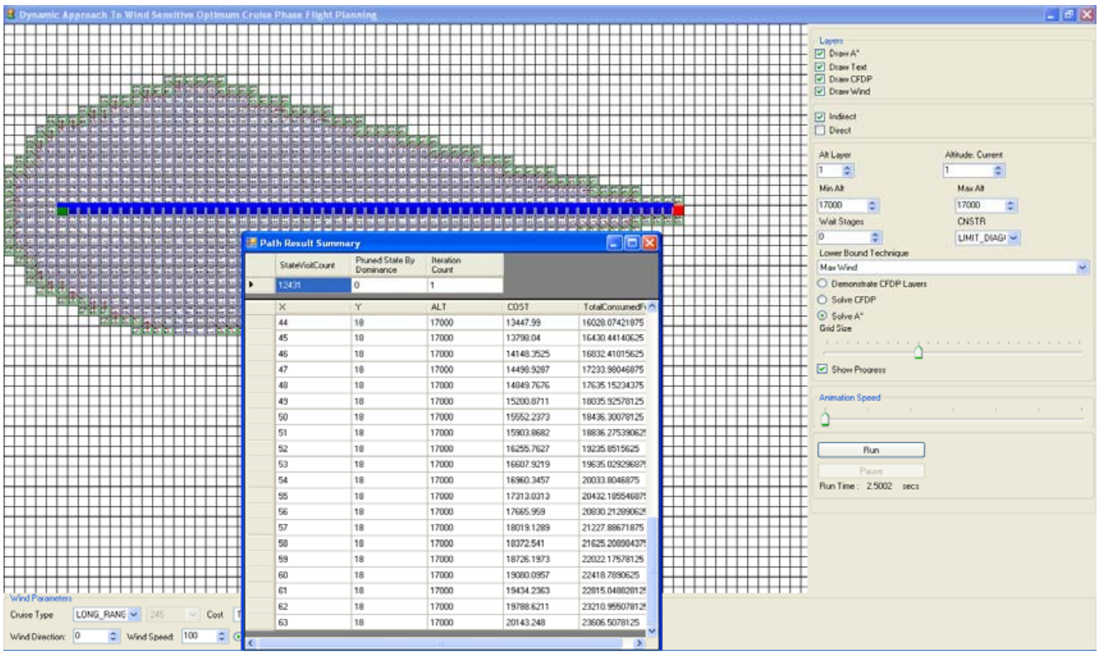
6.4 Test Case 4

Table 10 Test Case 4 Results

Scenario Id	Control Parameter	StateVisitCount	Objective Function[37]
scenario1	Min = Max Altitude = 24000	12431	20143.2
scenario2	Altitude: Min = 17000 Max = 32000 Altitude Layers: 8	102563	18293.9

[37] Long range cruise is selected to evaluate the effect of changing TAS per altitude

Scenarios so far are solved with just one altitude layer; this test case aims to present effect of extending state space by expanding the flight envelope in range [16000, 32000] and specifying 8 altitude layers of: 19000 - 21000 – 23000 – 25000 – 27000 – 29000 – 31000. Specifying 8 altitude layers adds one more dimension to the state variable hence causing StateVisitCount to increase from 12431 to 102563 as shown in Table **10**. Optimal function value is also improved because according to C-130 Performance model TAS increases as altitude increases.


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Path Result Summary
Layers
Animation Speed
Run
Pause
Wind Parameters

☑ Draw A*
☑ Draw Text
☑ Draw CFDP
☑ Draw Wind
☑ Indirect
☐ Direct

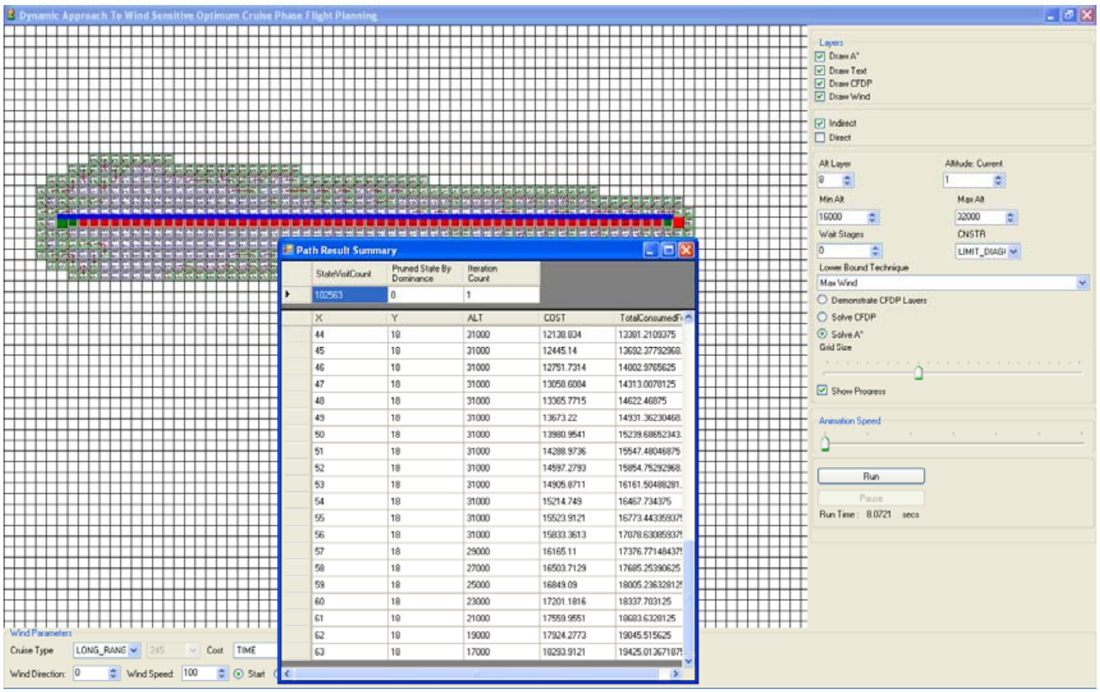
Alt Layer: 1
Altitude: Current: 1
Min Alt: 17000
Max Alt: 17000
Wait Stages: 0
CNSTR: LIMIT_DIAG
Lower Bound Technique: Max Wind
○ Demonstrate CFDP Layers
○ Solve CFDP
[✓ Solve A*]
Grid Size
☑ Show Progress
Run Time : 2.5002 secs

Cruise Type: LONG_RANG
245
Cost
Wind Direction: 0
Wind Speed: 100

StateVisitCount	Pruned State By Dominance	Iteration Count
12431	0	1

X	Y	ALT	COST	TotalConsumedF
44	18	17000	13447.99	16028.07421875
45	18	17000	13798.04	16430.44140625
46	18	17000	14148.3525	16832.41015625
47	18	17000	14498.9287	17233.98046875
48	18	17000	14849.7676	17635.15234375
49	18	17000	15200.8711	18035.92578125
50	18	17000	15552.2373	18436.30078125
51	18	17000	15903.8682	18836.2753906
52	18	17000	16255.7627	19235.0515625
53	18	17000	16607.9219	19635.0292968
54	18	17000	16960.3457	20033.8046875
55	18	17000	17313.0313	20432.1855468
56	18	17000	17665.959	20830.2128906
57	18	17000	18019.1289	21227.88671875
58	18	17000	18372.541	21625.2089843
59	18	17000	18726.1973	22022.17578125
60	18	17000	19080.0957	22418.7890625
61	18	17000	19434.2363	22815.0488281
62	18	17000	19788.6211	23210.9550781
63	18	17000	20143.248	23606.5078125

Figure 37 testCase4-scenario1


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Layers
Draw A*
Draw Text
Draw CFDP
Draw Wind
Run
Pause

Path Result Summary

	StateVisitCount	Pruned State By Dominance	Iteration Count
▶	102563	0	1

X	Y	ALT	COST	TotalConsumedF
44	18	31000	12138.834	13381.2109375
45	18	31000	12445.14	13692.27792968
46	18	31000	12751.7314	14002.9765625
47	18	31000	13058.6084	14313.0078125
48	18	31000	13365.7715	14622.46875
49	18	31000	13673.22	14931.36230468
50	18	31000	13980.9541	15239.68652343
51	18	31000	14288.9736	15547.48046875
52	18	31000	14597.2793	15854.75292968
53	18	31000	14905.8711	16161.50488281
54	18	31000	15214.749	16467.734375
55	18	31000	15523.9121	16773.44335937
56	18	31000	15833.3613	17078.63085937
57	18	29000	16165.11	17376.77148437
58	18	27000	16503.7129	17685.25390625
59	18	25000	16849.09	18005.23632812
60	18	23000	17201.1816	18337.703125
61	18	21000	17559.9551	18683.6328125
62	18	19000	17924.2773	19045.515625
63	18	17000	18293.9121	19425.01367187

[✓ Indirect] Direct
Alt Layer: 0
Altitude: Current: 1
Min Alt: 16000
Max Alt: 32000
Wait Stages: 0
CNSTR: LIMIT_DIAG
Lower Bound Technique: Max Wind
Demonstrate CFDP Layers / Solve CFDP / [✓ Solve A*]
Grid Size
[✓ Show Progress]
Animation Speed
Run Time : 8.0721 secs

Wind Parameters
Cruise Type: LONG_RANG 245 Cost: TIME
Wind Direction: 0 Wind Speed: 100 [✓ Start]

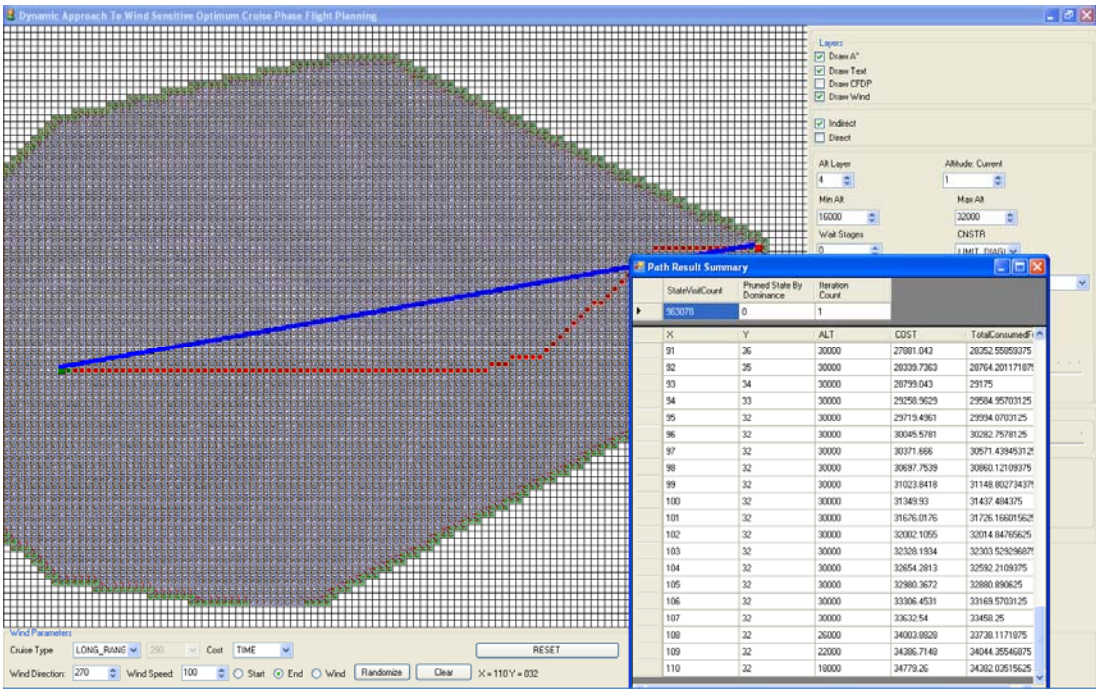
Figure 38 testCase4-scenario2

6.5 Test Case 5

Table 11 Test Case 5 Results

Scenario Id	Control Parameter	StateVisitCount	Objective Function
scenario1	Cruise Type Long Range	963078	34779.3
scenario2	Cruise Type Fix TAS 290	883540	33969.1
Scenario3	Cruise Type Fix TAS 245	970490	40208.3

This test case demonstrates the effect of changing Cruise Type on the solution. As shown in Figure 39, Figure 40, and Figure 41 applying Long Range cruise will cause following higher altitudes compares to Fix TAS.

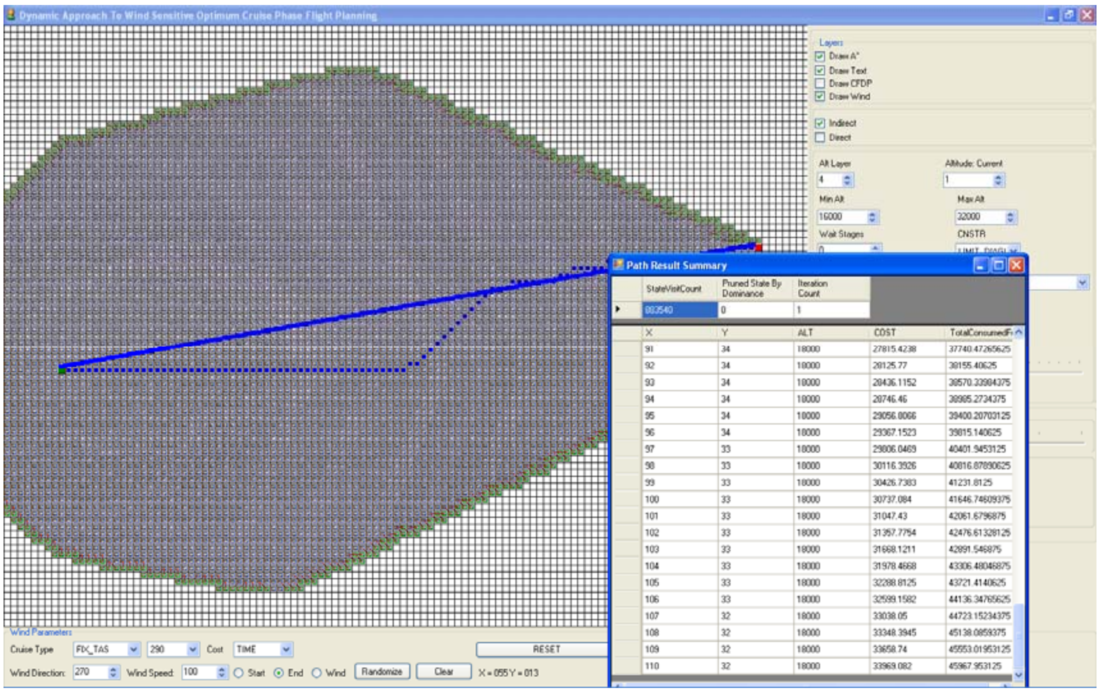

Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Path Result Summary
RESET
Randomize
Clear

Layers: [✓ Draw A*] [✓ Draw Text] Draw CFDP [✓ Draw Wind]
[✓ Indirect] Direct
Alt Layer: 4 Altitude: Current: 1
Min Alt: 16000 Max Alt: 32000
Wait Stages: 0 CNSTR: LIMIT_DIAG
Wind Parameters
Cruise Type: LONG_RANG 290 Cost: TIME
Wind Direction: 270 Wind Speed: 100 Start [✓ End] Wind X = 110 Y = 032

StateVisitCount	Pruned State By Dominance	Iteration Count
963078	0	1

X	Y	ALT	COST	TotalConsumedF
91	36	30000	27881.043	28352.55859375
92	35	30000	28339.7363	28764.2011718
93	34	30000	28799.043	29175
94	33	30000	29258.9629	29584.95703125
95	32	30000	29719.4961	29994.0703125
96	32	30000	30045.5781	30282.7578125
97	32	30000	30371.666	30571.4394531
98	32	30000	30697.7539	30860.12109375
99	32	30000	31023.8418	31148.8027343
100	32	30000	31349.93	31437.484375
101	32	30000	31676.0176	31726.1660156
102	32	30000	32002.1055	32014.84765625
103	32	30000	32328.1934	32303.5292968
104	32	30000	32654.2813	32592.2109375
105	32	30000	32980.3672	32880.890625
106	32	30000	33306.4531	33169.5703125
107	32	30000	33632.54	33458.25
108	32	26000	34003.8828	33738.1171875
109	32	22000	34386.7148	34044.35546875
110	32	18000	34779.26	34382.03515625

Figure 39 testCase4-scenario1

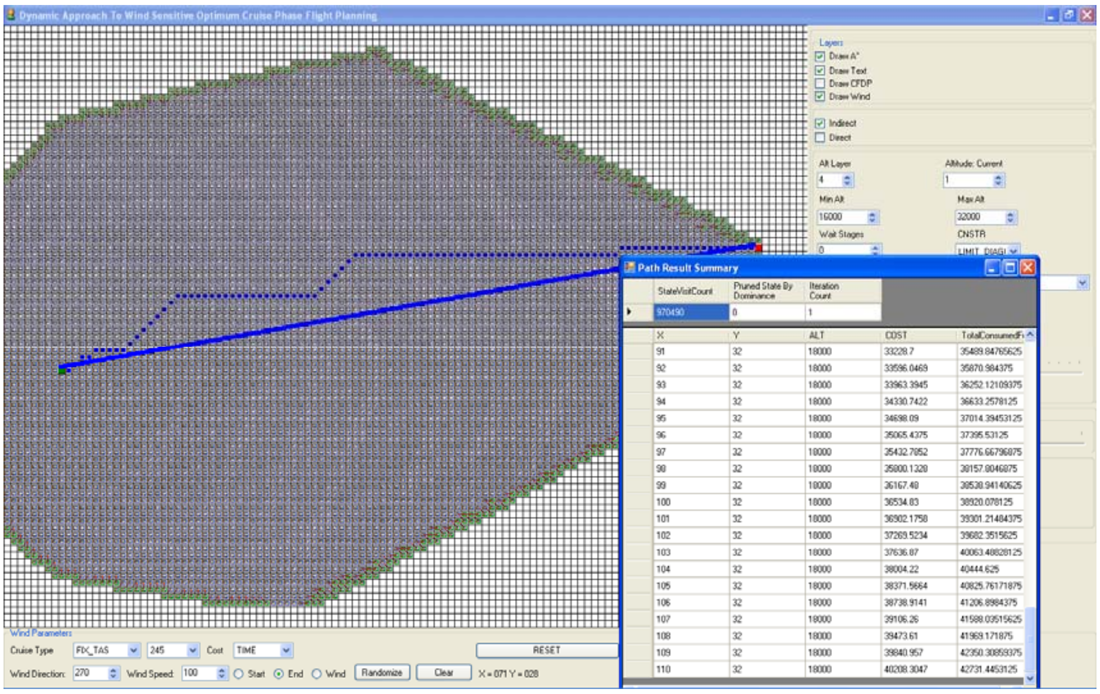

Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Path Result Summary
RESET
Randomize
Clear

Layers: [✓ Draw A*] [✓ Draw Text] Draw CFDP [✓ Draw Wind]
[✓ Indirect] Direct
Alt Layer: 4 Altitude: Current: 1
Min Alt: 16000 Max Alt: 32000
Wait Stages: 0 CNSTR: LIMIT_DIAG
Wind Parameters
Cruise Type: FIX_TAS 290 Cost: TIME
Wind Direction: 270 Wind Speed: 100 Start [✓ End] Wind X = 055 Y = 013

StateVisitCount	Pruned State By Dominance	Iteration Count
883540	0	1

X	Y	ALT	COST	TotalConsumedF
91	34	18000	27815.4238	37740.47265625
92	34	18000	28125.77	38155.40625
93	34	18000	28436.1152	38570.33984375
94	34	18000	28746.46	38985.2734375
95	34	18000	29056.8066	39400.20703125
96	34	18000	29367.1523	39815.140625
97	33	18000	29806.0469	40401.9453125
98	33	18000	30116.3926	40816.87890625
99	33	18000	30426.7383	41231.8125
100	33	18000	30737.084	41646.74609375
101	33	18000	31047.43	42061.6796875
102	33	18000	31357.7754	42476.61328125
103	33	18000	31668.1211	42891.546875
104	33	18000	31978.4668	43306.48046875
105	33	18000	32288.8125	43721.4140625
106	33	18000	32599.1582	44136.34765625
107	32	18000	33038.05	44723.15234375
108	32	18000	33348.3945	45138.0859375
109	32	18000	33658.74	45553.01953125
110	32	18000	33969.082	45967.953125

Figure 40 testCase5-scenario2



Layers: [✓ Draw A*] [✓ Draw Text] Draw CFDP [✓ Draw Wind]

[✓ Indirect] Direct

Alt Layer	Altitude: Current
4	1

Min Alt	Max Alt
16000	32000

Wait Stages	CNSTR
0	LIMIT_DIAG

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 270 Wind Speed: 100 Start [✓ End] Wind X = 071 Y = 028

StateVisitCount	Pruned State By Dominance	Iteration Count
970490	0	1

X	Y	ALT	COST	TotalConsumedF
91	32	18000	33228.7	35489.84765625
92	32	18000	33596.0469	35870.984375
93	32	18000	33963.3945	36252.12109375
94	32	18000	34330.7422	36633.2578125
95	32	18000	34698.09	37014.39453125
96	32	18000	35065.4375	37395.53125
97	32	18000	35432.7852	37776.66796875
98	32	18000	35800.1328	38157.8046875
99	32	18000	36167.48	38538.94140625
100	32	18000	36534.83	38920.078125
101	32	18000	36902.1758	39301.21484375
102	32	18000	37269.5234	39682.3515625
103	32	18000	37636.87	40063.48828125
104	32	18000	38004.22	40444.625
105	32	18000	38371.5664	40825.76171875
106	32	18000	38738.9141	41206.8984375
107	32	18000	39106.26	41588.03515625
108	32	18000	39473.61	41969.171875
109	32	18000	39840.957	42350.30859375
110	32	18000	40208.3047	42731.4453125

Figure 41 testCase5-scenario3

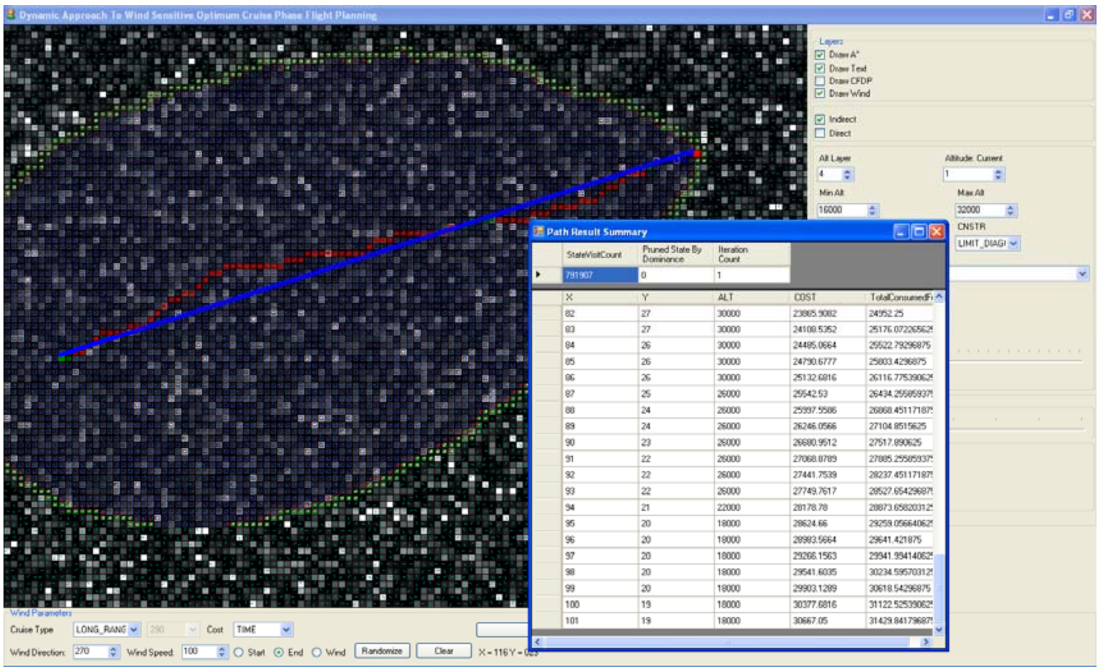
6.6 Test Case 6

Table 12 Test Case 6 Results

Scenario Id	Control Parameter	StateVisitCount	Objective Function
scenario1	Cost Type is Time	791907	30667 (seconds) Consumed Fuel is 31429.8 pound
scenario2	Cost Type is Fuel	810161	30914.5 pound[38]

This test case demonstrates the effect of changing the cost type on the solution for Long Range Cruise type where for all grids in all altitude layers wind is randomly generated. Referring to Figure 42 and Figure 43 paths slightly differs. Also as expected, Consumed fuel for scenario2 is less than that of scenario1 (refer to Table 12).

[38] The calculation of time required is not supported when the objective function is per fuel, it could be implemented as an additional feature.



Alt Layer: 4; Altitude: Current: 1; Min Alt: 16000; Max Alt: 32000; CNSTR: LIMIT_DIAG

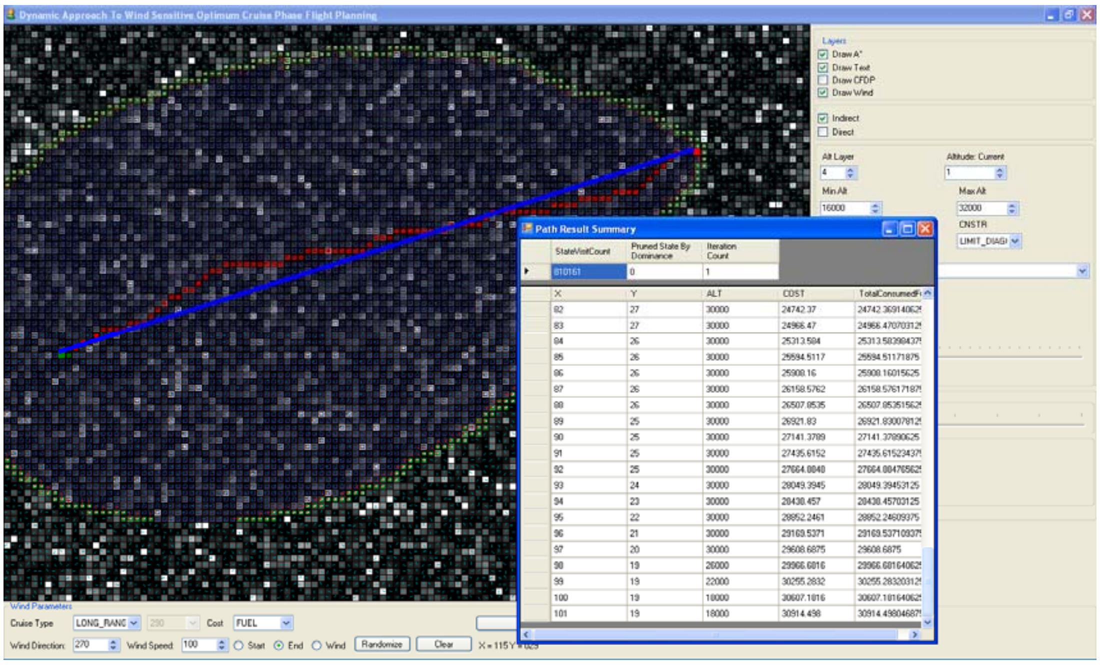
Cruise Type: LONG_RANG; Cost: TIME; Wind Direction: 270; Wind Speed: 100; Start [✓ End] Wind; X = 116 Y = 023

Path Result Summary

StateVisitCount	Pruned State By Dominance	Iteration Count
791907	0	1

X	Y	ALT	COST	TotalConsumedF
82	27	30000	23865.9082	24952.25
83	27	30000	24108.5352	25176.0722656
84	26	30000	24485.0664	25522.79296875
85	26	30000	24790.6777	25803.4296875
86	26	30000	25132.6816	26116.7753906
87	25	26000	25542.53	26434.2558593
88	24	26000	25997.5586	26868.4511718
89	24	26000	26246.0566	27104.8515625
90	23	26000	26680.9512	27517.890625
91	22	26000	27068.8789	27885.2558593
92	22	26000	27441.7539	28237.4511718
93	22	26000	27749.7617	28527.6542968
94	21	22000	28178.78	28873.6582031
95	20	18000	28624.66	29259.0566406
96	20	18000	28983.5664	29641.421875
97	20	18000	29266.1563	29941.9941406
98	20	18000	29541.6035	30234.5957031
99	20	18000	29903.1289	30618.54296875
100	19	18000	30377.6816	31122.5253906
101	19	18000	30667.05	31429.8417968

Figure 42 testCase6-scenario1[39]



Alt Layer: 4; Altitude: Current: 1; Min Alt: 16000; Max Alt: 32000; CNSTR: LIMIT_DIAG

Cruise Type: LONG_RANG; Cost: FUEL; Wind Direction: 270; Wind Speed: 100; Start [✓ End] Wind; X = 115 Y = 023

Path Result Summary

StateVisitCount	Pruned State By Dominance	Iteration Count
810161	0	1

X	Y	ALT	COST	TotalConsumedF
82	27	30000	24742.37	24742.3691406
83	27	30000	24966.47	24966.4707031
84	26	30000	25313.584	25313.5839843
85	26	30000	25594.5117	25594.51171875
86	26	30000	25900.16	25900.16015625
87	26	30000	26158.5762	26158.5761718
88	26	30000	26507.8535	26507.8535156
89	25	30000	26921.83	26921.8300781
90	25	30000	27141.3789	27141.37890625
91	25	30000	27435.6152	27435.6152343
92	25	30000	27664.0840	27664.0847656
93	24	30000	28049.3945	28049.3945312
94	23	30000	28430.457	28430.4570312
95	22	30000	28852.2461	28852.24609375
96	21	30000	29169.5371	29169.5371093
97	20	30000	29608.6875	29608.6875
98	19	26000	29966.6016	29966.6016406
99	19	22000	30255.2832	30255.2832031
100	19	18000	30607.1016	30607.1016406
101	19	18000	30914.498	30914.4980468

Figure 43 testCase6-scenario2

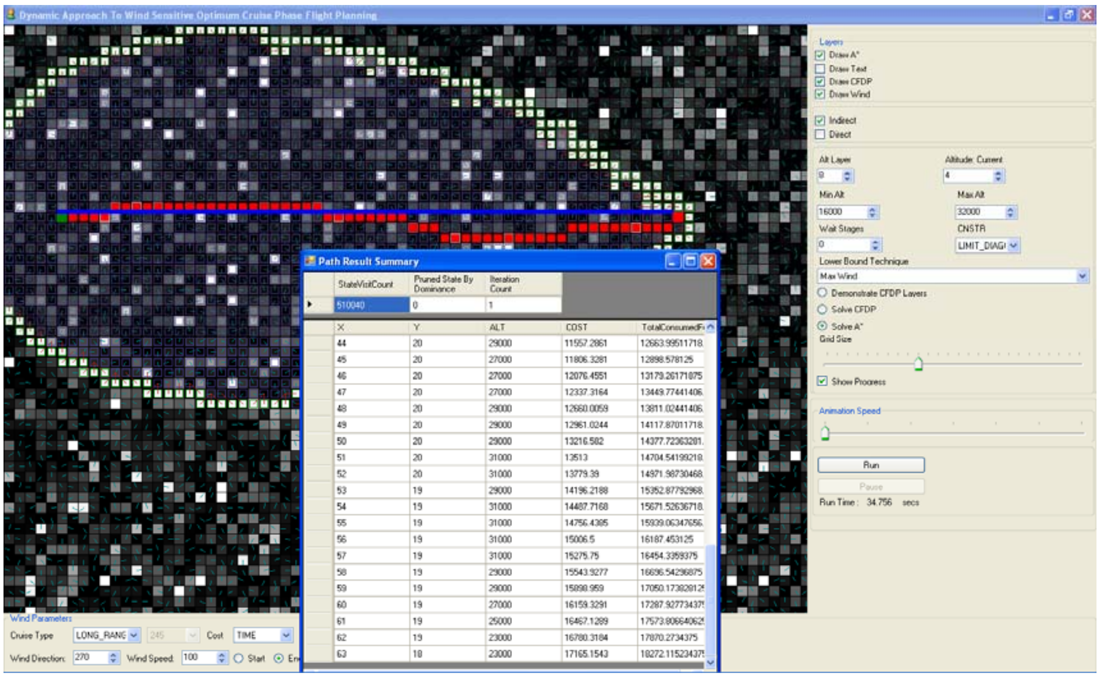
[39] Color codes represent the magnitude of the wind. The color gets darker as the wind magnitude increases

6.7 Test Case 7

Table 13 Test Case 7 Results

Scenario Id	Control Parameter	StateVisitCount	Objective Function[40]
Scenario1	A*	510040	17165.2
Scenario2	CFDP	463569	17165.2

This test case compares A* algorithm and CFDP with a more realistic path of 1500 NM distance and 8 altitude layers as shown in Figure 44 and Figure 45. Although StateVisitCount metric of CFDP is slightly better it required more run time processing because of computations performed for state management between the iterations.


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Path Result Summary
Layers
Run
Pause

StateVisitCount	Pruned State By Dominance	Iteration Count
510040	0	1

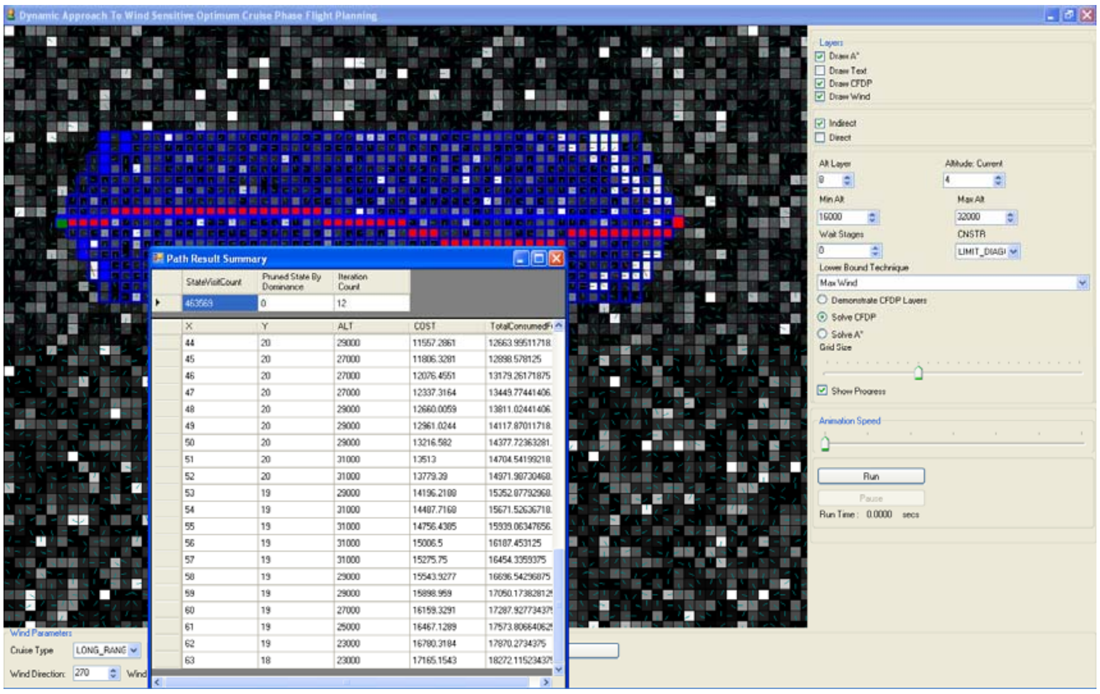
X	Y	ALT	COST	TotalConsumedF
44	20	29000	11557.2861	12663.99511718
45	20	27000	11806.3281	12898.578125
46	20	27000	12076.4551	13179.26171075
47	20	27000	12337.3164	13449.77441406
48	20	29000	12660.0059	13811.02441406
49	20	29000	12961.0244	14117.87011718
50	20	29000	13216.582	14377.72363281
51	20	31000	13513	14704.54199218
52	20	31000	13779.39	14971.98730468
53	19	29000	14196.2188	15352.87792968
54	19	31000	14487.7168	15671.52636718
55	19	31000	14756.4385	15939.06347656
56	19	31000	15006.5	16187.453125
57	19	31000	15275.75	16454.3359375
58	19	29000	15543.9277	16696.54296875
59	19	29000	15890.959	17050.1738281
60	19	27000	16159.3291	17287.9277343
61	19	25000	16467.1289	17573.8066406
62	19	23000	16780.3184	17870.2734375
63	18	23000	17165.1543	18272.1152343

Draw A* [✓]
Draw Text [ ]
Draw CFDP [✓]
Draw Wind [✓]
Indirect [✓]
Direct [ ]
Alt Layer: 8
Altitude Current: 4
Min Alt: 16000
Max Alt: 32000
Wait Stages: 0
CNSTR: LIMIT_DIAGI
Lower Bound Technique: Max Wind
Demonstrate CFDP Layers [ ]
Solve CFDP [ ]
Solve A* [✓]
Grid Size
Show Progress [✓]
Animation Speed
Run Time : 34.756 secs

Wind Parameters
Cruise Type: LONG_RANG
245
Cost: TIME
Wind Direction: 270
Wind Speed: 100
Start [ ] [✓ En]

Figure 44 testCase7-scenario1

[40] Long range cruise is selected to evaluate the effect of changing TAS per altitude


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Path Result Summary
Run
Pause

	StateVisitCount	Pruned State By Dominance	Iteration Count
▶	463569	0	12

X	Y	ALT	COST	TotalConsumedF
44	20	29000	11557.2861	12663.99511718
45	20	27000	11806.3281	12898.578125
46	20	27000	12076.4551	13179.26171875
47	20	27000	12337.3164	13449.77441406
48	20	29000	12660.0059	13811.02441406
49	20	29000	12961.0244	14117.87011718
50	20	29000	13216.582	14377.72363281
51	20	31000	13513	14704.54199218
52	20	31000	13779.39	14971.98730468
53	19	29000	14196.2188	15352.07792968
54	19	31000	14487.7168	15671.52636718
55	19	31000	14756.4385	15939.06347656
56	19	31000	15006.5	16187.453125
57	19	31000	15275.75	16454.3359375
58	19	29000	15543.9277	16696.54296875
59	19	29000	15898.959	17050.1738281
60	19	27000	16159.3291	17287.9277343
61	19	25000	16467.1289	17573.8066406
62	19	23000	16780.3184	17870.2734375
63	18	23000	17165.1543	18272.1152343

Layers
[✓] Draw A*
[ ] Draw Text
[✓] Draw CFDP
[✓] Draw Wind

[✓] Indirect
[ ] Direct

Alt Layer: 0	Altitude: Current: 4
Min Alt: 16000	Max Alt: 32000
Wait Stages: 0	CNSTR: LIMIT_DIAG

Lower Bound Technique: Max Wind
( ) Demonstrate CFDP Layers
(●) Solve CFDP
( ) Solve A*
Grid Size
[✓] Show Progress
Animation Speed
Run Time : 0.0000 secs

Wind Parameters
Cruise Type: LONG_RANG
Wind Direction: 270

Figure 45 testCase7-scenario2

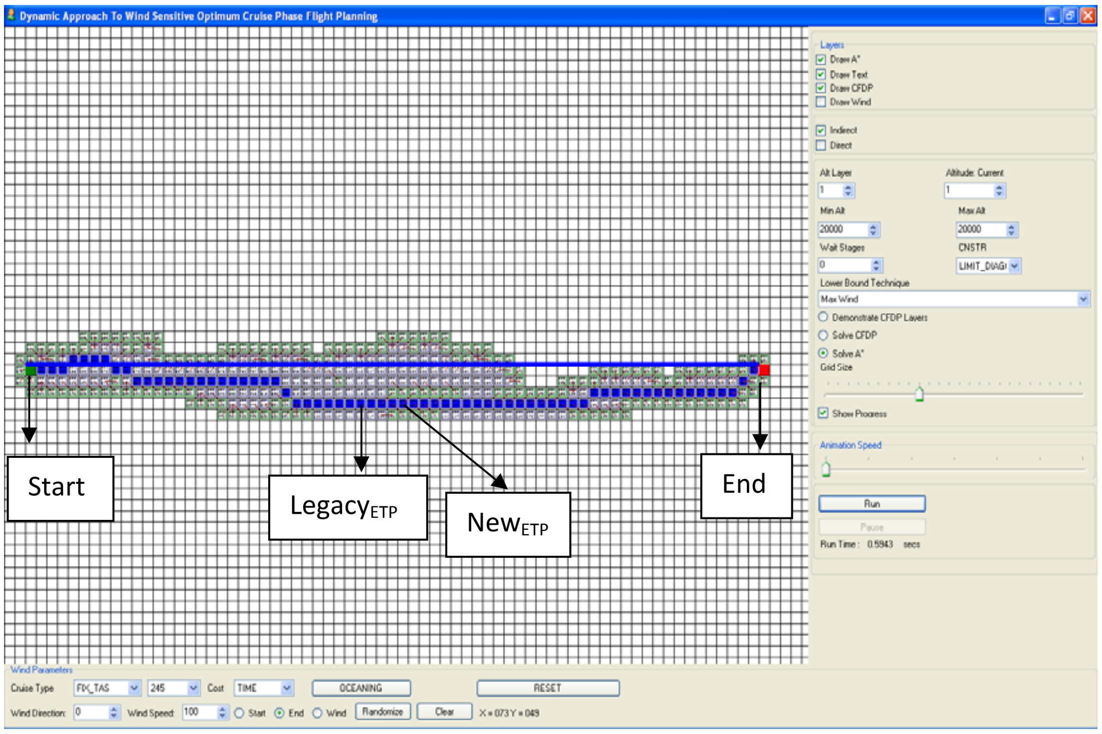
6.8 Test Case 8

Calculation of ETP[41] and NRP[42] in oceaning flights is considered as a CATASTROPHIC function, meaning that the correct and the accurate implementation of this functionality is life crucial. The existing legacy approach for ETP calculation requires tracking back the same flight path that is used in flying straight.

In no wind conditions, the ETP is located halfway between the two aerodromes. On the other hand, considering that the straight route is planned so as to take the maximum benefit of wind speed and direction, we can infer that if the straight route is optimum (in terms of either time or fuel) then returning back following the same course will give the worst case behavior (assuming the wind speed and direction in a point on route do not change in time when passing the same point forward and backward).

[41] *Equal-Time Point (ETP) – is the point in the flight where it would take the same time to continue flying straight, or track back to the departure aerodrome.*

[42] *No Return Point (NRP) - is the point on a flight at which, due to fuel consumption, a plane is no longer capable of returning to its airfield of original takeoff. After passing the point of no return, the plane has no option but to continue to some other destination.*



[✓ Draw A*]
[✓ Draw Text]
[✓ Draw CFDP]
Draw Wind

[✓ Indirect]
Direct

Alt Layer: 1
Altitude: Current: 1
Min Alt: 20000
Max Alt: 20000
Wait Stages: 0
CNSTR: LIMIT_DIAG
Lower Bound Technique: Max Wind

Demonstrate CFDP Layers
Solve CFDP
[✓ Solve A*]

Grid Size

[✓ Show Progress]

Run Time : 0.5943 secs

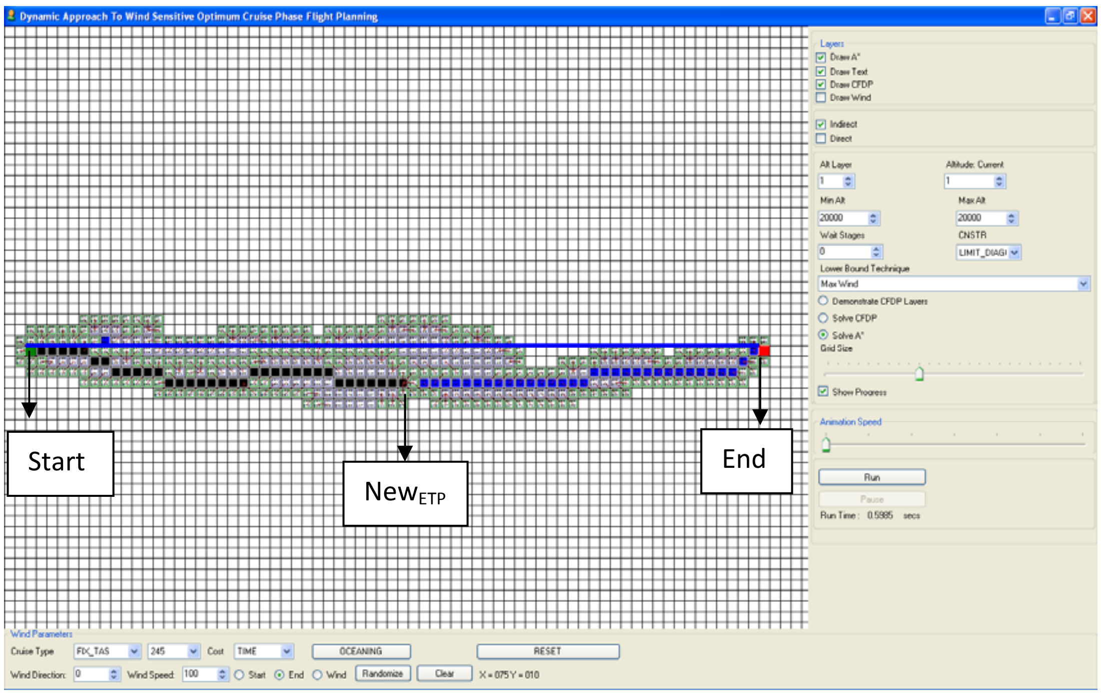
Wind Parameters
Cruise Type: FIX_TAS 245 Cost: TIME
Wind Direction: 0 Wind Speed: 100 Start [✓ End] Wind
X = 073 Y = 049

Figure 46 testCase8-Legacy_{ETP}

ETP calculated by the legacy method (Legacy_{ETP}) is shown in Figure 46. It will take 3 hours and 52 minutes either to go from Legacy_{ETP} to End or to go from Legacy_{ETP} to Start.

By relaxing the constraint of following the same path backward; ETP calculated by the algorithm developed in the thesis[43] (New_{ETP}) is shown in both Figure 46 and Figure 47. In this case, the backward route from New_{ETP} to Start (black squares in Figure 47) is not on the same the same flight path that is used in flying straight (blue squares in Figure 46). It now takes 3 hours and 32 minutes either to go from New_{ETP} to End or to go from New_{ETP} to Start. Comparing the two approaches while New_{ETP} is 100 NM further away from Start compared to Legacy_{ETP} it would take 20 minutes less (3 hours and 32 minute versus 3 hours and 52 minute).

[43] Each backward calculation is treated as a cost minimization for time problem for the backward route to Start


Dynamic Approach To Wind Sensitive Optimum Cruise Phase Flight Planning
Start
New_{ETP}
End
Layers
OCEANING
RESET
Randomize
Clear
Run
Pause

Layers: [✓ Draw A*] [✓ Draw Text] [✓ Draw CFDP] Draw Wind

[✓ Indirect] Direct

Alt Layer: 1 Altitude: Current: 1

Min Alt: 20000 Max Alt: 20000

Wait Stages: 0 CNSTR: LIMIT_DIAG

Lower Bound Technique: Max Wind

Demonstrate CFDP Layers Solve CFDP [✓ Solve A*]

Grid Size

[✓ Show Progress]

Animation Speed

Run Time : 0.5985 secs

Wind Parameters

Cruise Type: FIX_TAS 245 Cost: TIME

Wind Direction: 0 Wind Speed: 100 Start [✓ End] Wind

X = 075 Y = 010

Figure 47 testCase8-New_{ETP}

CHAPTER 7

CONCLUSION

In this thesis we implemented a top down approach by making use of aircraft route planning ontology so as to fill the gap between the flight plan specific domain knowledge and optimization techniques employed. Whereas the developed algorithm is generic by encapsulating the aircraft's performance characteristics as a look up table in a configuration file; it has been evaluated on C-130 aircraft.

On an Intel Core 2 Duo 2.83 Ghz with 2 GB RAM PC, it takes on average about 12 seconds to find the optimal solution for a flight path of 2500 NM start to end distance and 8 altitude layers (between 16000 feet t0 33000 feet). This moderate execution time makes it feasible to use the software program in meeting the operational needs.

The developed software program can be used as an *"what if"* analysis tool for the pilots or navigation officers to find optimum routes per time or fuel and to assess the effect of the changing weather forecast data on the route. An alternative operational usage would be incorporating automatic route planning functionality into existing Flight Management System(s) or Mission Planning Ground System(s).

The first major point of the study has been the application of a systematic approach by making use of the existing route planning ontology in the problem solution lifecycle, namely;

- Conceptual analysis,
- Formulation of the model,
- Design of the algorithm,
- Implementation of the algorithm

The real world problem on hand has been evaluated with respect to the existing route planning ontology and the formulation of the problem and the suitable optimization techniques has determined as a result.

The problem has formulated as a dynamic programming model; and state variables are selected by a through analysis including the problem specific constraints into the model.

Usage of ontology has been extended by applying Object Oriented Analysis and Design (OOAD) techniques during the design and the implementation of the algorithm.

The second major contribution has been the design of “efficient” lower bound techniques and the application of CFDP on route planning problem. Raphael [20] specifies the right choice of superstates as an “art”; in this thesis superstates are designed especially for aggregating the altitude layers.

We have seen that CFDP outweighs A* algorithm especially when there are relatively few paths that give near optimal results and when there are a number of altitude layers.

Quantitatively analyzing the weather forecast distribution and assessing the presence or absence of near optimal paths and then automatically deciding on whether to apply CFDP to current route planning problem can be a issue for a future work.

In a similar way, the way superstates are composed / departitioned can be dynamic depending on the route planning problem (i.e. number of altitude layers, constraints etc.) and environmental factors.

New lower bound techniques can be developed having monotonicity and admissibility properties. Spiral and CFDP Layers techniques could be merged such that not concentric circles are conceived but concentric arcs are conceived. The required parameters of the technique (i.e. the angle of arcs, whether to take tail wind component of the winds i.e.) could be designed so as the technique satisfies monotonicity and admissibility properties.

REFERENCES

[1] DO-236B *Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation.* Washington, DC 20036 USA : Radio Technical Commission for Aeronautics (RTCA) SC-181, 2003, Page(s): 3.

[2] (DoD), U.S. Department of Defense. *USAF Series C-130 Airplanes Performance Data.* s.l. : DoD, 2006. TO 1-C-130H-1-1.

[3] 702A-3, ARINC. *Advanced Flight Management Computer Ssystem.* Annapolis, Maryland : Aeronautical Radio, Inc., 2006. ARINC 702A-3, Page(s): 35.

[4] Services, Oxford Aviation. *Joint Aviation Authorities Airline Transport Pilot's LicenseTheoratical Knowledge Manual Meteorolgy.* Frankfurt, Germany : Jeppesen GmbH, 2001. 0-88487-286-6, Page(s): 6-5.

[5] Juan LIU, Su-yan TANG, Qun LI, Wei-ping WANG, *An Ontology for Aircraft Route Planning* 2009 First International Conference on Advances in System Simulation, Page(s): 68-72.

[6] Sam P. Liden, *Apparatus and Method for Computing Wind-Sensitive Optimum Altitude Steps In a Flight Management System*, November 12, 1996, Patent No: 5,574,647

[7] Keith D. Wichman, Göran Carlsson, Lindberg Flight Trials: *“Runway-To-Runway” Required Time of Arrival Evaluations For Time-Based ATM Environment*, IEEE 2001, 0-7803-7034-1/01 vol. 2, Page(s): 7F6/1 - 7F6/13.

[8] Erik Theunissen, Richard Rademaker and Okko Bleeker, Keith Wichman *Aircraft Trajectory Based Network Centric Applications*, IEEE 2007, 1-4244-1108—4/07, Page(s): 1.E.2-1 - 1.E.2-10.

[9] Lance Sherry, *NAS-Wide Simulation and Passenger Itinerary Performance: Implications for NEXTGEN Benefits Analysis*, Page(s): L7-1 - L7-9.

[10]John Benton, S. S. Lyengar, W. Deng, N. Brener, V.S. Subrahmanian, *Tactical Route Planning: New Algorithms for Decomposing the Map*, Page(s): 268-277.

[11]Hui Shi,Wen Cao, Shulong Zhu, Baoshan Zhu, *Applications of the Improved A* Algorithm for Route Planning*, Page(s): 299-302.

[12]Szczerba R J, Galkowski P, Glicktein I S, et al, *Robust algorithm for real-time route planning,* Aerospace and Electronic Systems, 2000, 36(3): pp. 869-878, doi: 10.1109/7.869506, Page(s): 869-878.

[13]Makoto Suzuki, Dai Araki, Akira Higashide, Teruaki Suzuki, *Geographical Route Planning Based on Uncertain Knowledge*, IEEE 1985, 1082-3409/95, Page(s): 434-441.

[14]M. Kavouras, E. Stefanakis, *Navigating in Space Under Constraints*, 2002, Vol. 1, International Journal of Pure and Applied Mathematics (IJPAM), Page(s): 71-93.

[15]R.E. BELLMAN, *Dynamic Programming*, Princeton University Press, Princeton, 1957

[16]Cooper, Leon Cooper and Mary W., *Introduction to Dynamic Programming,* Dallas : Pergamon Press, 1981. 0-08-025065-3.

[17]Dijkstra, *A note on two problems in connexion with graphs*, E. W. s.l. : Numerische Mathematik, 1959,Vol. 1, Page(s): 269-271.

[18]Beeker, Emmet, *Potential Error in the Reuse of Nilsson's A Algorithm for Path-finding in Military Simulations*, The Journal of Defense Modelling & Simulation., April 2004, Vol. 1, No.2, Page(s): 91-97.

[19]Hart, P. E., N. J. Nilsson, B. Raphael, *A Formal Basis for the Heuristic Determination of Minimum Cost Paths,* IEEE Transactions on Systems, Science, and Cybernetics, 1968, Vol. 4, Issue 2, Page(s): 100-107.

[20]Raphael, Christopher, *Coarse-to-Fine Dynamic Programming*, IEEE Transactions On Pattern Analysis And Machine Intelligence, December 2001, s. Vol. 23, No. 12, Page(s): 1379-1390.

[21]Gustavo Franco. A-Star (A*) Implementation in C# (Path Finding, PathFinder) from http://www.codeguru.com/csharp/csharp/cs_misc/designtechniques/article.php/c12527/AStar-A-Implementation-in-C-Path-Finding-PathFinder.htm Last accessed on March 6th 2012.

APPENDIX A

OBJECT ORIENTED APPROACH

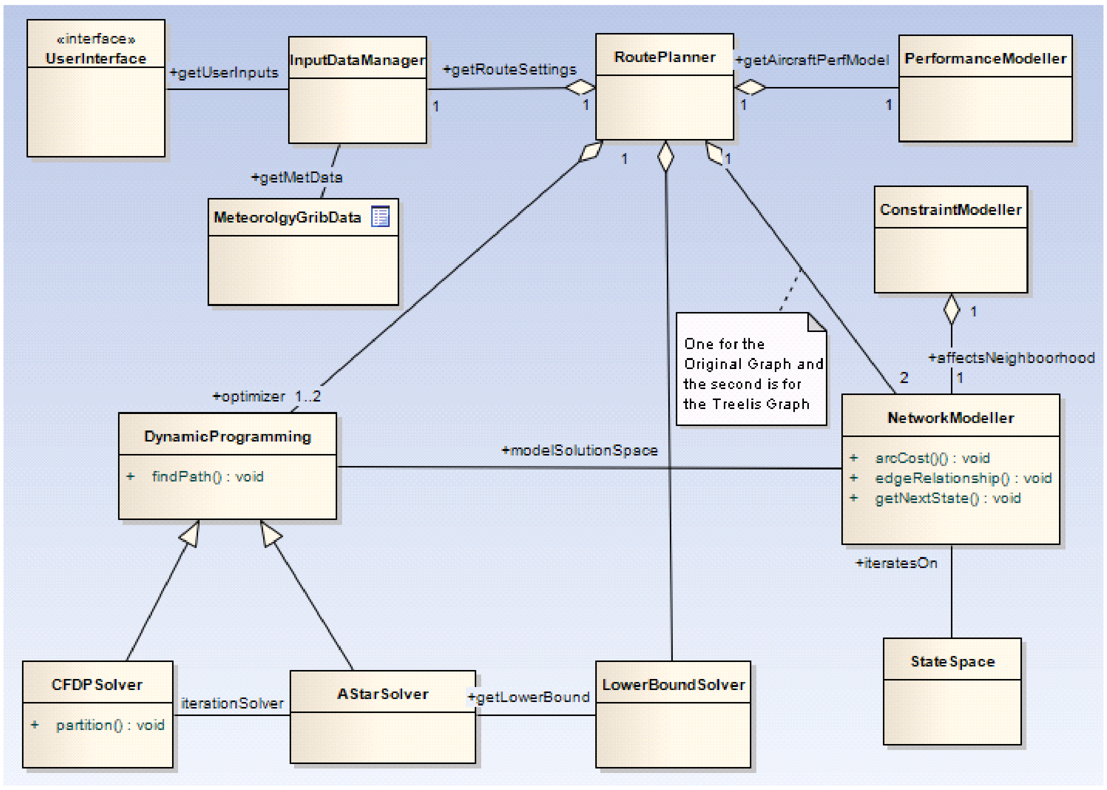


Figure 48 Class Diagram

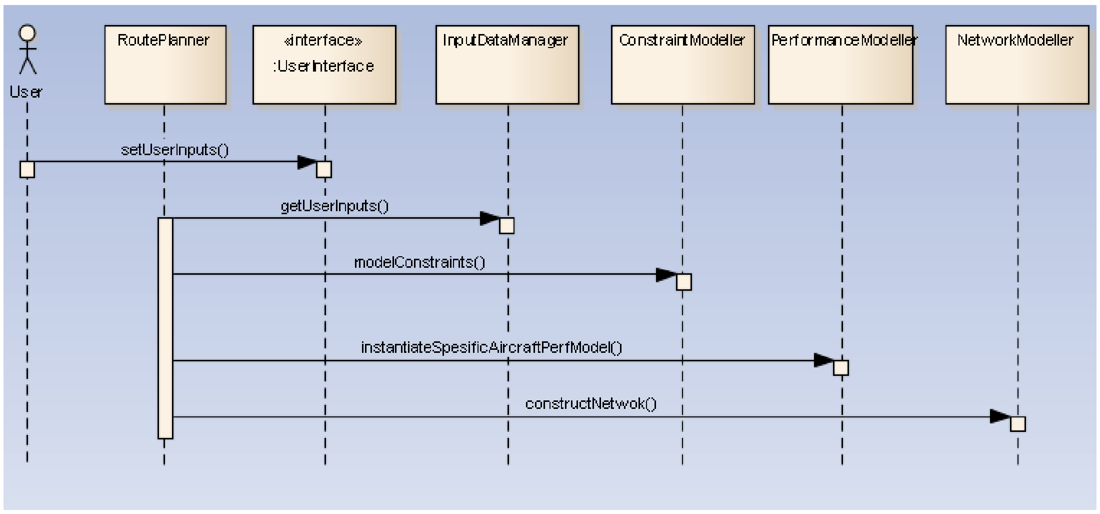


Figure 49 Initialization Sequence Diagram

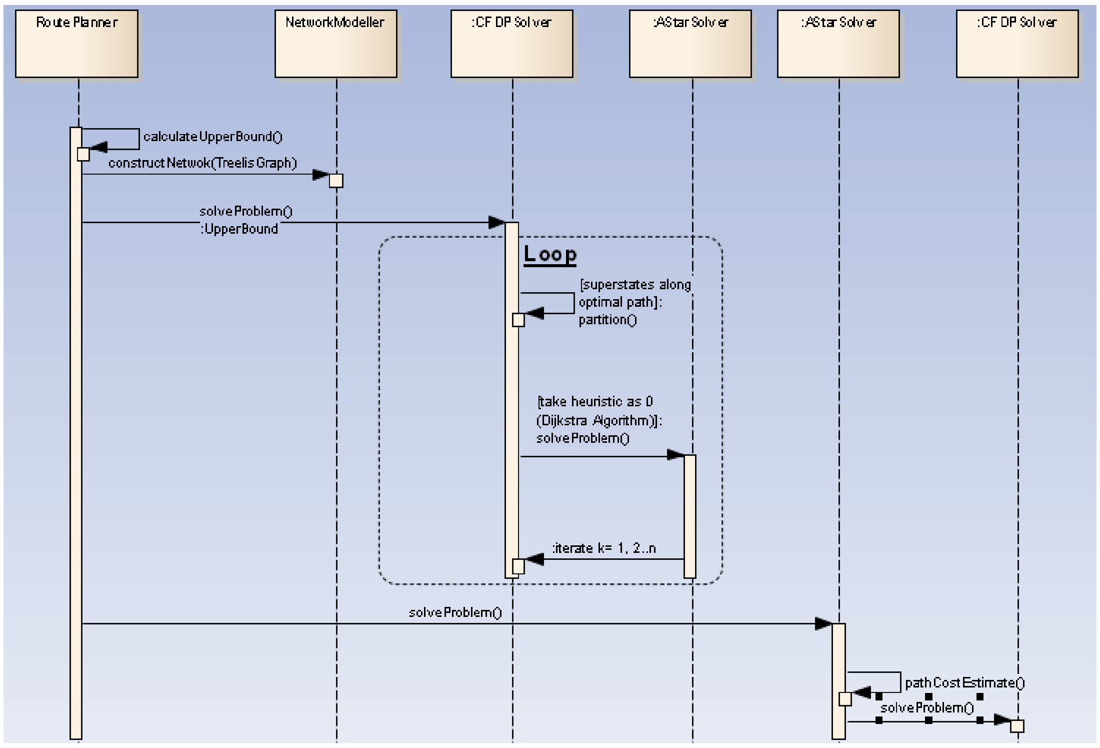


Figure 50 Optimization Sequence Diagram